Politecnico di Torino

ScuDo
Scuola di Dottorato ~ Doctoral School
WHAT YOU ARE, TAKES YOU FAR

Doctoral Dissertation

Doctoral Program in Electrical, Electronics and Communications Engineering
XXXVIII cycle

Advancements in Model Predictive Control for Real-Time Applications With Economic and Conflicting Control Objectives

by

Lorenzo Calogero

Supervisors
Prof. Alessandro Rizzo, Supervisor
Dr. Michele Pagone, Co-supervisor

Doctoral Examination Committee
Prof. Antonio Ferramosca, Referee, Università di Bergamo, Italy
Prof. Valentina Breschi, Referee, Eindhoven University of Technology, The Netherlands
Prof. Teodoro Rafael Alamo Cantarero, Universidad de Sevilla, Spain
Prof. Sophie Marie Fosson, Politecnico di Torino, Italy
Prof. Carlo Novara, Politecnico di Torino, Italy

Politecnico di Torino
2025

Declaration

I hereby declare that, the contents and organization of this dissertation constitute my own original work and does not compromise in any way the rights of third parties, including those relating to the security of personal data.

Lorenzo Calogero
2025

Abstract

Model Predictive Control (MPC) is an advanced optimal control strategy, widely recognized for its remarkable versatility in controlling a wide range of nonlinear, multivariable dynamical systems, and explicitly handling constraints.

Despite its desirable features, the application of MPC faces some inherent limitations in practice. The first is real-time feasibility, as MPC requires to repeatedly solve an optimal control problem online, which entails a considerable computational effort, compromising its use in fast application scenarios. The second concerns the lack, in existing MPC formulations, of a systematic methodology to handle multiple, conflicting, and time-varying control objectives, which typically arise in several practical applications.

This thesis aims to address these two challenges by proposing novel theoretical and methodological MPC frameworks, capable of operating in real time, under strict timing constraints and limited computational resources, while also extending their formulation to handle multiple, potentially conflicting control objectives, ensuring consistent performance and systematic closed-loop stability guarantees.

To address the challenge of real-time feasibility, we pursue two different directions. The first one focuses on developing a comprehensive analytical framework to fast solve Nonlinear MPC (NMPC) problems in the Koopman lifted space. Leveraging the Koopman operator theory, we propose a general procedure to analytically represent NMPC problems in a higher-dimensional Koopman space, and a complementary method to arbitrarily reduce the dimensionality of such a lifted space, which can be, in some cases, infinite-dimensional. This strategy yields an equivalent quadratic program (QP) in the Koopman lifted space (called Koopman NMPC, or K-NMPC), that closely approximates the original NMPC solution and can be solved with reduced computational effort. The K-NMPC framework showcases proficient control performance and a ten-fold reduction in computation times, as confirmed through extensive simulations and experimental validations.

The second direction achieves fast MPC operation through direct hardware implementation. Specifically, we present a general methodology to design fully-analog electronic circuits implementing Linear-Quadratic MPC policies (QP-MPC), leveraging Explicit MPC (Ex-MPC). This approach, referred to as Analog Circuital Ex-MPC, eliminates the need for online optimization on digital hardware, and ensures a remarkably fast evaluation of the MPC control action, with computation times in the order of few microseconds. This methodology is validated on the control of DC-DC Buck converters, a class of electronic devices with an inherently fast dynamics, demonstrating its real-time feasibility, robustness, and solid control performance.

To address the second challenge, we first direct our efforts towards Economic Nonlinear MPC (E-NMPC), a noteworthy variant of classic NMPC for systems where achieving a profitable economic performance is of primary concern. However, a conflict between control tasks naturally arises if the system is also tasked with regulation or tracking objectives. In this context, we propose a novel E-NMPC formulation, capable of handling such conflicting objectives, together with a general constructive procedure to guarantee closed-loop stability with minimal impact on the economic performance. This framework is validated on the case study of adaptive cruise control (ACC) for electric vehicles, achieving the optimal trade-off between vehicle control performance and energy efficiency, while also ensuring closed-loop stability.

For handling conflicting control tasks, we also propose an alternative framework, called Neural Adaptive Model Predictive Control (NA-MPC). NA-MPC extends standard MPC formulations by introducing an online metaheuristic tuning strategy, which dynamically adapts the weights of the MPC cost function, to achieve multiple conflicting objectives concurrently, and a neural emulation of the MPC policy, providing an equivalent neural controller that exhibits negligible computation time. The NA-MPC framework is applied to the case study of power management in fuel cell hybrid electric vehicles (FCHEVs), demonstrating solid control performance, adaptability, and superiority over conventional strategies.

Acknowledgments

The author gratefully acknowledges the financial support received for his doctoral research, which was funded in part by the European Union NextGenerationEU (NGEU)–Piano Nazionale di Ripresa e Resilienza (PNRR), Mission 4, Component 2, Investment 3.3, and the Italian Ministry of University and Research (MUR), under Ministerial Decree n. 352 of April 9, 2022; in part by Dumarey Softronix S.r.l., with the involvement of Dr. Francesco Cianflone and Dr. Edoardo Gandino as industrial advisors.

Contents

Glossary

List of Abbreviations

ACC Adaptive Cruise Control
ADC Analog-to-Digital Converter
A-ECMS Adaptive Equivalent Consumption Minimization Strategy
APV Affine Parameter-Varying
AQC Adiabatic Quantum Computing
CG Center of Gravity
CT Continuous-Time
CTG Constant Time Gap
DAC Digital-to-Analog Converter
DARE Discrete Algebraic Riccati Equation
DC-DC Direct Current to Direct Current
DT Discrete-Time
ECMS Equivalent Consumption Minimization Strategy
EDMD Extended Dynamic Mode Decomposition
EKF Extended Kalman Filter
E-NMPC Economic Nonlinear Model Predictive Control
EM Electric Motor
ESR Equivalent Series Resistance
EV Electric Vehicle
Ex-MPC Explicit Model Predictive Control
FC Fuel Cell
FCHEV Fuel Cell Hybrid Electric Vehicle
FGS Finite-Gain Stability
FLC Fuzzy-Logic Control
FNN Feedforward Neural Network
FPGA Field-Programmable Gate Array
GA Genetic Algorithm
HEV Hybrid Electric Vehicle
HV Hybrid Vehicle
HVAC Heating, Ventilation, and Air Conditioning
i.i.d. independent and identically distributed
KKT Karush-Kuhn-Tucker

K-NMPC Koopman Nonlinear Model Predictive Control
LAN Local Area Network
LiDAR Light Detection and Ranging
LP Linear Program/Programming
LPV Linear Parameter-Varying
LQR Linear Quadratic Regulator
LS Least Squares
LTI Linear Time-Invariant
MHE Moving Horizon Estimation
MOSFET Metal-Oxide-Semiconductor Field-Effect Transistor
MPC Model Predictive Control
mp-QP Multiparametric Quadratic Programming
MUX Multiplexer
NA-MPC Neural Adaptive Model Predictive Control
NLP Nonlinear Program/Programming
NMPC Nonlinear Model Predictive Control
OP-AMP Operational Amplifier
PEM Proton-Exchange Membrane
PSO Particle Swarm Optimization
PWA Piecewise Affine
PWM Pulse Width Modulation
QA Quantum Annealing
QC Quantum Computing
QP Quadratic Program/Programming
QP-MPC Linear-Quadratic Model Predictive Control
QUBO Quadratic Unconstrained Binary Optimization
RK Runge-Kutta
ROS Robot Operating System
RTI Real-Time Iteration
S/H Sample and Hold
SOC State of Charge
SQP Sequential Quadratic Programming
VMC Voltage Mode Control
WLTC Worldwide Harmonized Light Vehicles Test Cycle
wrt with respect to
ZOH Zero-Order Hold

List of Notations

The following section gives an overview of the notation used throughout this thesis. Note that auxiliary or specific notations are not displayed here.

General Notations

Sets and Spaces

$\mathbb{N}$, $\mathbb{Z}_{\geq 0}$ Natural numbers (including 0), non-negative integer numbers
$\mathbb{R}$ Real numbers
$\mathbb{R}_{>0}$ Positive real numbers
$\mathbb{R}_{\geq 0}$ Non-negative real numbers
$\mathcal{A} = \{a_1,\ a_2,\ \ldots,\ a_N\} = \{a_i\}_{i=1}^N$ Set of N elements a_i (non-ordered, non-repeatable)
$\mathcal{C} = (c_1,\ c_2,\ \ldots,\ c_N) = (c_i)_{i=1}^N$ Collection of N elements c_i (ordered, repeatable)
$\mathcal{B}_r(x)$ Closed ball centered at x with radius r
C^n Class of functions with continuous derivative of order n ($n \geq 0$)
L^p Space of measurable functions equipped with the p-norm ($1 \leq p \leq \infty$)
$\mathcal{K}$ Class of continuous functions $\alpha : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$ which are strictly increasing with $\alpha(0) = 0$
$\mathcal{K}_\infty$ Class of functions $\alpha \in \mathcal{K}$ which are unbounded

Vectors and Matrices

$x = [x_1,\ \ldots,\ x_n]^\top = [x_i]_{i=1}^n$ Column vector with n components x_i
$\boldsymbol{x} = (x_1,\ \ldots,\ x_N) = (x_i)_{i=1}^N$ Collection of N vectors x_i
$(x)_i$ i-th component of the vector x
$(x)_{\mathcal{I}}$ Vector collecting the components of x, indexed by the set $\mathcal{I} \subset \{1, \ldots, n\}$
$A = \begin{bmatrix} a_{1,1} & \cdots & a_{1,n} \\ \vdots & \ddots & \vdots \\ a_{m,1} & \cdots & a_{m,n} \end{bmatrix} = [a_{i,j}]_{1 \leq i \leq m,\, 1 \leq j \leq n}$ Matrix with m rows, n columns (i.e., size $m \times n$), and elements $a_{i,j}$
$(A)_{i,j}$ Element of the matrix A in position (i, j)
$(A)_{i,\cdot}$ i-th row of the matrix A
$(A)_{\cdot,j}$ j-th column of the matrix A
$(A)_{\mathcal{I},\cdot}$ Matrix composed by the rows of A, indexed by the set $\mathcal{I} = \{1, \ldots, m\}$
$(A)_{\cdot,\mathcal{J}}$ Matrix composed by the columns of A, indexed by the set $\mathcal{J} = \{1, \ldots, n\}$
$(A)_{\mathcal{I},\mathcal{J}}$ Matrix composed by the elements of A, belonging to a row indexed by the set $\mathcal{I} = \{1, \ldots, m\}$ and a column indexed by the set $\mathcal{J} = \{1, \ldots, n\}$
$(A_1,\ \ldots,\ A_N) = (A_i)_{i=1}^N$ Collection of N matrices A_i
$\mathbf{0}_n$ Null vector of dimension n
$\mathbf{1}_n$ All-1 vector of dimension n
$\mathbf{0}_{m \times n}$ Null matrix of size $m \times n$
I_n Identity matrix of size $n \times n$
$e_n^{(m)}$ m-th vector of the standard Euclidean basis of $\mathbb{R}^n$
$E_n^{(m)}$ Matrix of size $n \times n$, with 1s on the m-th subdiagonal and 0s elsewhere

Operators and Functions

$\mathrm{conv}(\cdot)$ Convex hull of a set of points
$\mathrm{diag}(A)$ Diagonal of the matrix A
$\mathrm{diag}(x)$ Matrix with the vector x as its diagonal
$\mathrm{diag}(X_1, \ldots, X_N)$ Block diagonal matrix, with blocks $X_1, \ldots, X_N$
$\mathrm{int}(\cdot)$ Interior of a set
$\mathrm{Lip}(\cdot)$ Lipschitz constant of a function
$O(\cdot)$ Landau's big O
$\mathrm{Pre}(\cdot)$ Predecessor set
$\mathrm{Proj}_x(\mathcal{A}) = \{x \in \mathbb{R}^{n_x} : \exists\, y \in \mathbb{R}^{n_y},\ (x, y) \in \mathcal{A}\}$ Projection of the set $\mathcal{A}$ onto the lower-dimensional space $\mathbb{R}^{n_x}$
$\mathrm{ReLU}(\cdot)$ Rectified linear unit
$\mathrm{sign}(\cdot)$ Sign function
$\mathrm{span}(\cdot)$ Linear combination
$|\cdot|$ Set cardinality, absolute value
$\|x\|$, $\|x\|_2$ Euclidean norm (or 2-norm) of the vector x
$\|x\|_p$ p-norm of the vector x $(1 \leq p \leq \infty)$
$\|x\|_M = \|M^{1/2}x\| = \sqrt{x^\top M x}$ Weighted norm of the vector x, with weighting matrix M
$\|A\|_p$ Induced p-norm of the matrix A
$\|A\|_M = \|M^{1/2} A M^{-1/2}\|$ Induced weighted norm of the matrix A, with weighting matrix M
$\otimes$ Kronecker product
$\odot$ Element-wise product, logical AND
$\oplus$ Logical OR
$\overline{\star}$ Logical NOT
$\star^\dagger$ Pseudoinverse matrix
$L_\star = \star \cdot \partial$ Lie derivative
$\lambda_{\min}(\cdot)$, $\lambda_{\max}(\cdot)$ Minimum and maximum eigenvalue of a matrix
$\mathbb{P}[\,\cdot\,]$ Probability
$\mathbb{E}[\,\cdot\,]$ Expected value
$U(\cdot)$ Uniform probability distribution
$N(\cdot)$ Normal probability distribution

Dynamical Systems

x State vector
u Input vector
y Output vector
ν Vector of exogenous disturbances
$\mathcal{X}$ State constraint set, state-space manifold
$\mathcal{U}$ Input constraint set
$\mathcal{Y}$ Output constraint set, output manifold
$\mathscr{U}$ Space of piecewise continuous input signals
$f(x, u)$, $f(x, u, \nu)$ State dynamics function (for continuous-time systems), state transition function (for discrete-time systems)

$g(x,u)$, $g(x,u,\nu)$ Output function
$\varphi^t(x_0)$, $\varphi^k(x_0)$ Flow of the system, parameterized by t (continuous-time) or k (discrete-time), with initial state x_0

Model Predictive Control

$z=(x,u)=[x^\top,u^\top]^\top$ Combined state-input vector
$\mathcal{Z}=\mathcal{X}\times\mathcal{U}$ Combined state-input constraint set
$\mathcal{Z}_e=\{(x_e,u_e)\in\mathcal{Z}: f(x_e,u_e)=0 \text{ or } x_e=f(x_e,u_e)\}$ Manifold of admissible equilibrium points (continuous-time and discrete-time)
$\mathcal{U}^N$ Set of admissible input sequences for horizon N ($N\in\mathbb{N}_{\geq 1}\cup\{\infty\}$)
$\mathcal{X}^N$ Set of admissible state sequences for horizon N
N_p Prediction horizon
N_c Number of free input decision variables under the move blocking strategy
$\hat{\boldsymbol{x}}_k=(\hat{x}_{i|k})_{i=0}^{N_p}$ Predicted state trajectory, i steps ahead of the current time k
$\hat{\boldsymbol{u}}_k=(\hat{u}_{i|k})_{i=0}^{N_p-1}$ Input sequence for prediction, i steps ahead of the current time k
$\hat{\boldsymbol{y}}_k=(\hat{y}_{i|k})_{i=0}^{N_p-1}$ Predicted output trajectory, i steps ahead of the current time k
$J(\boldsymbol{x},\boldsymbol{u})$ Cost function
$\ell(x,u)$ Stage cost function
$\ell_r(x,u)$ Stage cost function encoding a regulation or tracking task
$\ell_e(x,u)$ Stage cost function encoding an economic criterion
$V_f(x)$ Terminal cost function
$\mathcal{X}_f$ Terminal constraint set
$\hat{\boldsymbol{x}}_k^*=(\hat{x}_{i|k}^*)_{i=0}^{N_p}$ Optimal predicted state trajectory
$\hat{\boldsymbol{u}}_k^*=(\hat{u}_{i|k}^*)_{i=0}^{N_p-1}$ Optimal input sequence
$\hat{\boldsymbol{y}}_k^*=(\hat{y}_{i|k}^*)_{i=0}^{N_p-1}$ Optimal predicted output trajectory
$J_N^*(x)$ Optimal cost value function with prediction horizon N
$\pi(x)$ Optimal control policy
$\mathscr{U}_N(x)$ Set of feasible input sequences for horizon N and initial state x
$\mathcal{X}_N$ Set of feasible states for horizon N, region of attraction
$\kappa_f(x)$ Terminal control law

Koopman Operator

$\mathcal{K}^{t,\boldsymbol{u}}$, $\mathcal{K}^{k,\boldsymbol{u}}$ Koopman operator, parameterized by t (continuous-time) or k (discrete-time), and the input signal $\boldsymbol{u}$
$\phi(x)$ Observable function (or observable)
$\phi(x)=[\phi_i(x)]_{i=1}^{n_z}$ Observables arranged in a vector
$\Phi=\{\phi_i(x)\}_{i=1}^{n_z}$ Basis of observables
$\mathscr{F}$ Banach space of continuously differentiable observables
$\mathscr{F}_N$ N-dimensional linear subspace of $\mathscr{F}$
$\mathcal{Z}\subseteq\mathbb{R}^{n_z}$ Koopman lifted state space
$z\in\mathbb{R}^{n_z}$ Lifted state vector
$N_o=n_z=|\Phi|$ Number of observables in the basis, dimensionality of the lifted space
$\overline{N}_o$ Number of observables in the reduced basis, dimensionality of the reduced lifted space

$\mathcal{L}$ Infinitesimal generator of the Koopman operator

$\psi_{\text{in}}(x)$ Initial hand-picked observable

$\psi_{\text{in}}(x) = [\psi_{\text{in},i}(x)]_{i=1}^{N_{\text{in}}}$ Initial hand-picked observables arranged in a vector

$\Phi_{\text{in}} = \{\psi_{\text{in},i}(x)\}_{i=1}^{N_{\text{in}}}$ Initial set of hand-picked observables

$\psi(x)$ Observable obtained through analytical generation

$\psi(x) = [\psi_i(x)]_{i=1}^{N_\psi}$ Observables obtained through analytical generation, arranged in a vector

$\Psi = \{\psi_i(x)\}_{i=1}^{N_\psi}$ Set of observables obtained through analytical generation

$h(x)$ Nonlinear inequality constraint function

$\sigma(x)$ Elementary nonlinear term composing h as a linear combination

$\sigma(x) = [\sigma_i(x)]_{i=1}^{N_\sigma}$ Elementary nonlinear terms composing h as a linear combination, arranged in a vector

$z_x \in \mathbb{R}^{n_x}$ Lifted states related to the observables x (i.e., original system states)

$z_{\text{in}} \in \mathbb{R}^{N_{\text{in}}}$ Lifted states related to the initial hand-picked observables $\psi_{\text{in}}(x)$

$z_\sigma \in \mathbb{R}^{N_\sigma}$ Lifted states related to the observables $\sigma(x)$ (i.e., elementary nonlinear terms of h)

$z_\psi \in \mathbb{R}^{N_\psi}$ Lifted states related to the generated observables $\psi(x)$

$\hat{\boldsymbol{z}}_k = (\hat{z}_{i|k})_{i=0}^{N_p}$ Predicted lifted state trajectory, i steps ahead of the current time k

$\hat{\boldsymbol{z}}_k^* = (\hat{z}_{i|k}^*)_{i=0}^{N_p}$ Optimal predicted lifted state trajectory

1

Introduction

Automatic control constitutes one of the foundational disciplines of modern engineering and technology. Among its possible definitions, the essence of automatic control can be grasped from the etymology of the word "automatic", which derives from the ancient Greek *autómatos*, meaning "self-acting" or "acting of one's own will". Hence, the main purpose of automatic control is to enable real-world systems to perform their intended tasks without direct human intervention. In other words, automatic control endows a system with the capability to act on its own: the system evaluates its own state and determines the necessary actions to modify its current behavior, with the aim of attaining its prescribed objective.

The earliest known control system dates back to around 270 B.C., when Ctesibius of Alexandria, Ptolemaic Egypt, designed an improved water clock, which, by means of a float-valve feedback mechanism, allowed to maintain constant water pressure, ensuring accurate measurement of time without manual adjustments [91].

Over the centuries, the progressive improvement in our ability to design control devices has allowed us to entrust automated systems with tasks of increasing complexity. As this complexity has grown, so too has the need for more advanced control methodologies. In particular, nonlinear control has gained increasing importance, as most real-world systems exhibit nonlinear behaviors that must be explicitly considered in order to deliver an effective control action.

Among these control strategies, a most promising one, widely recognized by both the academic and the industrial community, is Model Predictive Control (MPC) [64]. Its widespread diffusion has arisen from its remarkable versatility in controlling a wide range of nonlinear, multivariable dynamical systems, explicitly handling constraints, and possibly accounting for exogenous disturbances and changing operating conditions.

MPC is an optimization-based control strategy, belonging to the family of optimal control methods. Its working principle relies on a mathematical model of the system to control, which is employed to predict its future evolution over a finite horizon under different sequences of control inputs. These predictions are then leveraged to compute the optimal control action, which is obtained by solving, at each sampling time instant, a finite-horizon constrained optimal control problem [64]. In this optimization problem, the cost function encodes the control objective to be attained by MPC. This optimization-based

approach resembles, to some extent, human decision-making, in which actions are often taken by anticipating possible future outcomes, derived from observation and reasoning.

Owing to its desirable features, MPC has found extensive use in numerous industrial and technological fields, including process control, robotics, automotive, aerospace, energy systems, and advanced manufacturing [128, 129]. To address the diversity of these application contexts, several MPC formulations have been developed to accommodate different classes of dynamical systems. Among them, Linear-Quadratic MPC (QP-MPC) [22] and Nonlinear MPC (NMPC) [64] are the most prominent, designed for linear and nonlinear multivariable systems, respectively. Extensions to these schemes have been proposed, allowing to handle, e.g., hybrid, time-delayed, and event-triggered systems [16, 111, 94]. In addition, robust MPC frameworks have been developed for uncertain and stochastic systems [81]. Further formulations also encompass networked, distributed, and multi-agent systems [138].

1.1. Motivation

MPC stands as a versatile and effective control strategy across a wide variety of scenarios. However, its application faces inherent limitations in practice.

Among these limitations, the most significant one is real-time feasibility. At each sampling time instant, MPC requires solving online a constrained optimal control problem. Although closed-form solutions can be found in some particular cases of Linear-Quadratic MPC (going by the name of Explicit MPC [14]), solving such an optimization problem is generally computationally intensive, especially for Nonlinear MPC. For slow systems, such as industrial processes, the available computational resources are typically enough to solve the optimal control problem in real time. However, for systems characterized by a fast dynamics or subject to strict timing constraints, standard MPC schemes are, in general, computationally prohibitive.

Another relevant limitation of MPC can be found in the context of control performance, rather than computational aspects. Specifically, MPC is inherently well suited to account for multiple and potentially conflicting control objectives within its cost function. However, despite this intrinsic capability, a systematic methodology to handle conflicting control tasks in MPC, while also ensuring closed-loop stability and consistent control performance, is still lacking. Moreover, in real-world applications, control objectives and operating conditions are often time-varying. In these situation, introducing an online adaptation of the MPC control action could be advantageous for maintaining a consistent control performance. However, MPC schemes commonly rely on offline tuning, typically based on trial-and-error procedures, while only few strategies have been proposed for online adaptation.

Among the various MPC formulations, the most relevant one in which conflicting and time-varying objectives naturally arise is Economic Nonlinear Model Predictive Control (E-NMPC) [134], which is aimed at delivering an economically profitable control action, rather than attaining pure regulation or tracking. Despite its potential, conventional E-NMPC schemes typically account for economic objectives only, disregarding other adversarial tasks, lack systematic closed-loop stability guarantees, and do not employ online adaptation strategies.

This thesis aims to address these two key challenges, namely, the development of novel fast and real-time feasible MPC frameworks, capable of operating under strict timing constraints and limited computational resources, and the extension of Nonlinear Economic MPC to handle multiple, potentially conflicting, and time-varying control objectives, while ensuring consistent control performance and systematic closed-loop stability guarantees.

1.2. Contributions and Thesis Outline

In the following, we report the outline of this thesis and provide a comprehensive overview of its main contributions.

- **Chapter 2** reviews the theoretical foundations of MPC, focusing on its main formulations, namely Nonlinear MPC (NMPC), Linear-Quadratic MPC (QP-MPC), and Explicit MPC (Ex-MPC), with focus on their fundamental properties and stability results.

- **Part I** covers the contributions of this thesis to the development of fast and real-time feasible MPC frameworks, introducing novel formulations and strategies that enable MPC to operate under strict timing constraints, without compromising the control performance.

    - **Chapter 3** presents a comprehensive analytical framework to efficiently solve Nonlinear MPC problems in the Koopman lifted space.

      The Koopman operator is a powerful system-theoretic approach that is capable of transforming nonlinear dynamical systems into equivalent linear (or bilinear) ones, evolving within a different state space, named Koopman lifted space, which, compared to the original one, is higher-dimensional (and, possibly, infinite-dimensional). The shift between these state spaces is performed by means of a suitable basis of static functions, called observable functions (or, more briefly, observables).

      The Koopman operator framework can be extended to nonlinear optimal control problems, enabling a fast and efficient solution of them in the Koopman lifted space. However, a systematic methodology for analytically finding a suitable basis of Koopman observables and handling the operator infinite-dimensionality is still lacking.

      To address these limitations, in this chapter we formulate a general procedure to analytically derive a basis of observables that lifts both the nonlinear prediction model and nonlinear constraints of NMPC, obtaining a quadratic program in the lifted space (called Koopman NMPC, or K-NMPC) that closely approximates the original NMPC solution, and can be solved with superior computational performance. Additionally, we develop a general method to arbitrarily reduce the dimensionality of the Koopman lifted space, lowering the K-NMPC complexity and handling the infinite-dimensional case.

      Our K-NMPC approach is validated on real-world case studies, through both simulations and experimental validations, assessing control performance

and execution times.

- **Chapter 4** presents a general methodology to design fully-analog electronic circuits implementing Linear-Quadratic MPC policies, leveraging Explicit MPC.

  Linear-Quadratic MPC is a compelling control strategy for many applications, but its hardware implementation remains a challenge in practice. Its reliance on digital supports introduces significant costs and the need for real-time optimization limits its applicability for fast systems. Explicit MPC mitigates these issues by removing the need for online optimization. However, digital implementations of Ex-MPC can still be costly and complex when involving many polytopic regions.

  In this scenario, a promising alternative is brought by analog implementations, which is the path that we follow in this chapter. Specifically, starting from a general QP-MPC problem, we begin by deriving its Ex-MPC form. Then, we introduce a set of tailored complexity-reduction techniques, through which we greatly simplify the Ex-MPC policy. Finally, the Ex-MPC is implemented as an electronic circuit, using only commercially-available low-latency analog components, by which we achieve a cheap design and a high-speed circuit operation, ensuring effective control of systems with very fast dynamics or stringent timing constraints.

  We validate our analog circuital Ex-MPC approach on the case study of DC-DC Buck converters control, assessing control performance, robustness, and disturbance rejection capability, along with a comparison with conventional control strategies.

- **Part II** covers the contributions of this thesis on Economic Nonlinear MPC (E-NMPC), focusing on control applications where multiple, conflicting, and time-varying control objectives must be simultaneously attained.

  - **Chapter 5** presents a novel Economic NMPC formulation for conflicting control objectives, together with a general constructive procedure to design suitable stabilizing terms, ensuring the closed-loop stability of E-NMPC with minimal impact on the economic performance.

    Within the domain of optimal control, Economic NMPC has emerged as a noteworthy variant of classic NMPC, for systems where economic performance is of primary concern. Specifically, E-NMPC is designed to deliver an economically optimal control action, optimizing the economic profit of the plant with respect to a given economic criterion. However, its applicability is hindered by two main limitations: E-NMPC does not allow to include additional adversarial tasks, such as tracking, and its closed-loop stability is not easy to guarantee.

    To address these limitations, in this chapter we develop a novel formulation of E-NMPC, which handles conflicting control objectives, such as tracking and economic tasks, allowing to attain the optimal trade-off between them. Furthermore, we propose a general constructive procedure

to design suitable stabilizing terms for E-NMPC, ensuring its closed-loop stability with minimal impact on the economic performance.

The effectiveness of the proposed E-NMPC strategy is validated on the case study of Adaptive Cruise Control (ACC) in electric vehicles (EVs). Optimizing the energy consumption of EVs during operation is a key factor in mitigating their overall environmental impact. Autonomous vehicle functions, such as ACC, typically disregard economic criteria, such as energy optimization, being in general not trivial to conciliate tracking and economic control tasks.

- **Chapter 6** presents a novel advanced control framework, named Neural Adaptive Model Predictive Control (NA-MPC), which augments standard MPC problems with three key features. First, an online metaheuristic tuning strategy dynamically adapts the MPC cost function weights, to achieve multiple conflicting control objectives at once. Second, through neural emulation, the MPC control policy is replaced by an equivalent neural controller, which faithfully delivers the original optimal control action, thanks to universal approximation guarantees, and also ensures real-time feasibility. Third, a neural black-box model of the plant is considered, serving also as MPC prediction model, so to ensure applicability when an accurate, white-box model of the plant is not readily available.

  The generality and versatility of NA-MPC make it suitable for a broad range of control applications. In this chapter, we validate it on the case study of power management in fuel cell hybrid electric vehicles (FCHEVs), a topic of growing interest within the frame of sustainable transportation. In this domain, current state-of-the-art strategies exhibit several shortcomings: they typically fail in achieving an effective trade-off between accurate power tracking and supply saving, proving a merely suboptimal control action; their adaptation capability is limited, relying either on offline tuning or simplistic non-optimal adaptation policies; moreover, only few basic optimal control approaches have been proposed in the literature, with little focus on their real-time feasibility.

  NA-MPC overcomes these limitations by providing optimal power allocation, effectively attaining multiple control objectives concurrently, and, thanks to its neural embedding, ensuring real-time feasibility and a straightforward implementation on hardware with limited computational resources.

2

Model Predictive Control

2.1. Introduction

Model predictive control (MPC) stands as one of the most established and widely adopted control strategies in modern control theory. Its broad recognition in both academia and industry stems from its remarkable versatility in controlling a wide range of nonlinear, multivariable dynamical systems, explicitly handling constraints, and possibly accounting for exogenous disturbances and changing operating conditions.

Its key working principle consists of three main steps: first, given the current system state, a finite-horizon constrained optimal control problem is solved, obtaining a sequence of optimal control inputs; then, only the first sample of the optimal input sequence is applied to the system (this step is called receding horizon); finally, the system evolves under the given control input, attaining its next state that is used to solve a new optimal control problem, closing the loop.

The widespread success of MPC over conventional control techniques arises from a set of distinctive features that make it both effective and versatile.

First, MPC relies on model-based predictions of the plant evolution, over a given prediction horizon. Specifically, within the optimal control problem, a prediction model of the plant is inserted as a set of nonlinear equality constraints. The free input decision variables serve to evaluate the evolution of such a model. Then, MPC seeks the optimal input sequence minimizing a given cost function, which encodes the control task to be attained. Finally, only the first optimal input of the sequence is applied to the plant.

The predictive nature of MPC allows it to effectively control a wide variety of systems, provided that we can derive a suitable mathematical model of them for prediction. In particular, existing MPC formulations are capable of handling linear and nonlinear multivariable systems (called, respectively, *Linear-Quadratic MPC* and *Nonlinear MPC*) [22, 64], as well as hybrid, time-delayed, and event-triggered systems [16, 111, 94]. Further extensions of MPC have been developed for stochastic and uncertain systems [81], as well as networked, distributed, and multi-agent systems [138].

Additionally, by its inherent formulation as an optimization problem, MPC is capable of seamlessly handling constraints on the plant states, inputs, and outputs. These are directly

inserted in the optimal control problem, typically as nonlinear inequality constraints. Such constraints arise in many practical applications, in which they can represent actuation and physical limits, as well as regions of operation, obstacles to be avoided, and safety or comfort requirements [31, 34].

As we can see, the working principle of MPC is particularly effective and, to some extent, it takes inspiration from human decision-making. A classic example on this aspect is the analogy between MPC and a human playing chess.

For both MPC and playing chess, decisions are made through a process of foresight. In MPC, the future evolution of the plant is predicted over a finite horizon, and the optimal input sequence is selected to minimize the given cost function. Similarly, a chess player mentally explores possible future moves and countermoves, assessing their consequences several steps ahead to identify the most advantageous strategy. In both contexts, predictions are inherently limited to a finite horizon: MPC considers only a finite prediction interval, just as a player can evaluate only a few moves in advance before the complexity becomes intractable.

Also constraints play a comparable role in defining feasible decisions. In MPC, they restrict the admissible values for the input sequence and the predicted state and output trajectories. In chess, the rules of the game, among which how each piece can move and the finite size of the board, constrain how the player can plan its moves and strategy.

Finally, the receding horizon in MPC, for which only the first sample of the optimal input sequence is applied to the system, closely mirrors what happens in chess, where the player plans several moves ahead, but only a single piece can be moved at a time.

Along with its ability to handle various classes of constrained systems, MPC can be employed to attain a wide range of control tasks. The most relevant ones are the following: *regulation*, consisting in steering the system towards one of its equilibrium states; *path following*, which involves following a predefined sequence of states, without explicit timing constraints; *tracking*, which is similar to path following, but with prescribed timing, meaning that the sequence of states is time-parameterized and, in general, corresponds to a trajectory of the system.

The first studies introducing the fundamental concepts of MPC appeared in the 1960s, addressing both linear and nonlinear systems [127, 88]. However, it was only in the late 1970s that MPC began to gain wider popularity within control engineering [148, 44]. Its initial success emerged primarily in the field of process control, where the relatively slow dynamics of industrial processes were compatible with the computational capabilities available at the time for solving the optimization problems underlying MPC [64]. Academic interest in the theoretical foundations of MPC, particularly its stability properties, developed somewhat later, with the first studies on stability of linear and nonlinear MPC formulations appearing during the 1980s and early 1990s [55, 103].

Modern stability theory in MPC largely relies on the inclusion of the so-called *terminal ingredients*, namely a terminal cost function and a terminal state constraint, within the MPC optimal control problem [64]. Both of these ingredients are "terminal", in the sense that they act upon the state predicted at the end of the prediction horizon, which is known as terminal state. The foundations of these concepts date back to the 1990s, when the first studies on MPC stability with terminal ingredients have emerged [47, 38, 104]

In the following years, alternative schemes were also proposed, relaxing or even omitting

terminal ingredients, thereby broadening the applicability of MPC while maintaining stability guarantees [65, 62, 64].

In the early 2000s, another significant advancement in MPC was achieved with the introduction of *Explicit Model Predictive Control* (Ex-MPC) [14], first formalized in the pioneering work [18]. This approach was developed to overcome the high computational demand inherent to MPC, which requires solving an optimal control problem online at each time step. Explicit MPC obviates the need for online optimization by pre-solving Linear-Quadratic MPC problems over the entire set of feasible states. The result is an explicit solution to the original MPC problem, in the form of a static state-feedback control policy, that can be evaluated with negligible computational effort. The Ex-MPC policy is proved to have a closed-form expression, exhibiting a piecewise-affine (PWA) structure, defined over a collection of polytipic regions [18]. As a result, instead of solving online the MPC problem, Ex-MPC only requires to locate the polytopic region containing the current system state and evaluating the related affine function, thereby achieving a substantial reduction in computational complexity.

2.1.1. Outline

Building on this background, in the remainder of this chapter we will delve into the mathematical foundations of MPC theory.

In particular, we will begin by analyzing its two principal variants: Nonlinear MPC (Section 2.2) and Linear-Quadratic MPC (Section 2.3). For both schemes, we will focus on their two main formulations: MPC with stabilizing terminal ingredients (Sections 2.2.1 and 2.3.1) and MPC without stabilizing terminal ingredients (Sections 2.2.2 and 2.3.2), discussing in detail the conditions required to guarantee closed-loop stability in both cases. Finally, we will address Explicit MPC (Section 2.4), focusing on the derivation of its explicit solution.

We remark that most of the results presented in this chapter are classical and can be found in several works within the MPC literature. In view of the huge amount of this literature, our aim here is to provide a concise and coherent compendium that uniformly present these foundational results in MPC theory. Below, we provide a selected list of references, that served as the main sources we have drawn upon in writing this chapter.

The theoretical results on MPC with terminal ingredients, presented Sections 2.2 and 2.3, are adapted from [47, 38, 104, 64, 133, 22]. The results on MPC without terminal ingredients, presented in Sections 2.2.2 and 2.3.2, are based on [65, 62, 64]. Section 2.4 on Explicit MPC collects the results presented in [18, 14, 149, 61].

2.2. Nonlinear Model Predictive Control

Let us consider a discrete-time (DT) nonlinear dynamical system, i.e.,

$$x_{k+1} = f(x_k, u_k), \quad k \in \mathbb{Z}_{\geq 0}, \tag{2.1}$$

subject to the constraints

$$x_k \in \mathcal{X} = \{x \in \mathbb{R}^{n_x} : c_x(x) \leq 0\} \subseteq \mathbb{R}^{n_x}, \tag{2.2a}$$

$$u_k \in \mathcal{U} = \{u \in \mathbb{R}^{n_u} : c_u(u) \leq 0\} \subseteq \mathbb{R}^{n_u}, \tag{2.2b}$$

at all time instants $k \geq 0$. In Eqs. (2.1) and (2.2), $x \in \mathbb{R}^{n_x}$ and $u \in \mathbb{R}^{n_u}$ are the state and input vectors, respectively, $f : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \to \mathbb{R}^{n_x}$ is the state transition function, $c_x : \mathbb{R}^{n_x} \to \mathbb{R}^{N_{c_x}}$ and $c_u : \mathbb{R}^{n_u} \to \mathbb{R}^{N_{c_u}}$ are nonlinear functions, which define the constraint sets $\mathcal{X}$ and $\mathcal{U}$ by imposing N_{c_x} and N_{c_u} inequality constraints on the state and input, respectively. Hereafter, let $\mathcal{Z} = \mathcal{X} \times \mathcal{U}$.

If the functions c_x and c_u are affine, i.e.,

$$c_x(x) = H_x x - h_x, \quad c_u(u) = H_u u - h_u, \tag{2.3}$$

then $\mathcal{X}$ and $\mathcal{U}$ are convex polytopes, with the following half-space representation:

$$\mathcal{X} = \{x \in \mathbb{R}^{n_x} : H_x x \leq h_x\}, \quad \mathcal{U} = \{x \in \mathbb{R}^{n_x} : H_x x \leq h_x\}. \tag{2.4}$$

We introduce the following additional assumptions on system (2.1):

Assumption 2.1

The state x_k of system (2.1) is known and available at each time instant $k \geq 0$.

Assumption 2.1 is a standard one in the MPC literature, as it allows to focus the analysis and design workflow on the control problem itself, decoupling it from an additional state estimation problem. While such an assumption can be often viable in practice, for several real-world systems the full state is either not available or affected by disturbances. As a consequence, in such cases it is necessary to combine MPC with a state estimator. Several approaches have been proposed in this respect, ranging from classic state estimation methods, based on nonlinear observers or the Extended Kalman Filter (EKF), or optimization-based ones, such as the Moving Horizon Estimation (MHE). For further details, we refer the interested reader to [133, Section 1.4, Chapters 4, 5].

Assumption 2.2

The function f is continuous on $\mathcal{X} \times \mathcal{U}$.

Assumption 2.3

System (2.1) admits a manifold of equilibrium points, given by

$$\mathcal{Z}_r = \{(x_r, u_r) \in \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} : x_r = f(x_r, u_r)\}. \tag{2.5}$$

Definition 2.1 (Admissibility)

a) The states $x \in \mathcal{X}$ are called *admissible states* and the inputs $u \in \mathcal{U}$ are called *admissible inputs*.

b) For $N \in \mathbb{N}_{\geq 1} \cup \{\infty\}$, we call an input sequence $\boldsymbol{u} = (u_i)_{i=0}^{N-1}$ *admissible up to time* N if $u_i \in \mathcal{U}$, $\forall\, i = 0, \ldots, N-1$. $\mathcal{U}^N$ is the set of admissible input sequences.

Similarly, we call a state sequence $\boldsymbol{x} = (x_i)_{i=0}^{N-1}$ admissible up to time N if $x_i \in \mathcal{X}$, $\forall\, i = 0, \ldots, N-1$. $\mathcal{X}^N$ is the set of admissible state sequences.

For system (2.1), we consider the following two control tasks:

1) *Regulation* towards an admissible state $x_r \in \mathcal{X}$ (called *reference state*), for which exists an admissible input $u_r \in \mathcal{U}$ such that (x_r, u_r) is an equilibrium point of system (2.1), i.e.,

$$x_r = f(x_r, u_r). \tag{2.6}$$

2) *Tracking* of an admissible state sequence $\boldsymbol{x}_r = (x_{r,k})_{k\geq 0} \in \mathcal{X}^\infty$, for which exists an admissible input sequence $\boldsymbol{u}_r = (u_{r,k})_{k\geq 0} \in \mathcal{U}^\infty$ such that

$$x_{r,k+1} = f(x_{r,k}, u_{r,k}), \quad \forall\, k \geq 0, \tag{2.7}$$

i.e., $\boldsymbol{x}_r$ is a trajectory of system (2.1) (called *reference state trajectory*).

Nonlinear Model Predictive Control (NMCP) serves as our control strategy to attain the above tasks.

The NMPC strategy provides, at each discrete time instant $k \geq 0$, an optimal control input u_k^* to system (2.1), obtained by solving the following finite-horizon constrained optimal control problem:

$$\min_{\hat{\boldsymbol{x}}_k, \hat{\boldsymbol{u}}_k} \quad J(\hat{\boldsymbol{x}}_k, \hat{\boldsymbol{u}}_k) = \sum_{i=0}^{N_p-1} \ell(\hat{x}_{i|k}, \hat{u}_{i|k}) + V_f(\hat{x}_{N_p|k}) \tag{2.8a}$$

$$\text{s.t.} \quad \hat{x}_0 = x_k, \tag{2.8b}$$

$$\hat{x}_{i+1|k} = f(\hat{x}_{i|k}, \hat{u}_{i|k}), \quad i = 0, \dots, N_p - 1, \tag{2.8c}$$

$$\hat{x}_{i|k} \in \mathcal{X}, \quad i = 0, \dots, N_p - 1, \tag{2.8d}$$

$$\hat{u}_{i|k} \in \mathcal{U}, \quad i = 0, \dots, N_p - 1, \tag{2.8e}$$

$$\hat{x}_{N_p|k} \in \mathcal{X}_f. \tag{2.8f}$$

In the NMPC problem (2.8):

- $(\hat{u}_{0|k}, \hat{u}_{1|k}, \dots, \hat{u}_{N_p-1|k}) = (\hat{u}_{i|k})_{i=0}^{N_p-1} = \hat{\boldsymbol{u}}_k$ and $(\hat{x}_{0|k}, \hat{x}_{1|k}, \dots, \hat{x}_{N_p|k}) = (\hat{x}_{i|k})_{i=0}^{N_p} = \hat{\boldsymbol{x}}_k$ are sequences of N_p inputs and $N_p + 1$ states, respectively, acting as decision variables of problem (2.8). N_p is called *prediction horizon*.

  The notation "$i|k$" has the following meaning: i is the prediction time instant, with $i = 0, 1, \dots, N_p$; k is the current time instant of system (2.1).

  Therefore, $\hat{x}_{i|k}$ and $\hat{u}_{i|k}$ represent state and input samples that are predicted i steps ahead of the current time k, respectively.

  For notational convenience, we can omit the explicit dependence on k when clear from context.

- Eq. (2.8a) reports the cost function J, encoding the given control task. It is composed by the *stage cost function* $\ell : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \to \mathbb{R}$ and the *terminal cost function* $V_f : \mathbb{R}^{n_x} \to \mathbb{R}$; the latter is a key ingredient to establish closed-loop stability guarantees for the NMPC problem (2.8).

- Eq. (2.8b) imposes that the initial state $\hat{x}_{0|k}$ is equal to the current state x_k of system (2.1).

- Eq. (2.8c) imposes that the state and input sequences are trajectories of system (2.1). In this way, $\hat{\boldsymbol{x}}_k$ and $\hat{\boldsymbol{u}}_k$ are actual predictions of the dynamics of system (2.1), with initial condition given by Eq. (2.8b), over the prediction time $i = 0, \dots, N_p$. Hence, Eq. (2.8c) is called *prediction model*.

- Eqs. (2.8d) and (2.8e) imposes that the predicted state trajectory $(\hat{x}_{i|k})_{i=0}^{N_p-1}$ and input sequence $\hat{\boldsymbol{u}}_k$ are admissible up to time N_p, i.e., they respect the constraints in Eq. (2.2).
- Eq. (2.8f) imposes a constraint on the terminal predicted state $\hat{x}_{N_p|k}$, where $\mathcal{X}_f \subseteq \mathcal{X}$ is called *terminal set*. Eq. (2.8f) is called *terminal constraint* and is a key ingredient to establish closed-loop stability guarantees for the NMPC problem (2.8).

The optimal input sequence and predicted state trajectory, resulting from the solution of the NMPC problem (2.8), are denoted by $\hat{\boldsymbol{u}}_k^* = (\hat{u}_{i|k}^*)_{i=0}^{N_p-1}$ and $\hat{\boldsymbol{x}}_k^* = (\hat{x}_{i|k}^*)_{i=0}^{N_p}$, respectively.

The *optimal cost value* is given by $J(\hat{\boldsymbol{x}}_k^*, \hat{\boldsymbol{u}}_k^*)$. Since the only parameter of the NMPC problem (2.8) is the current state x_k of system (2.1) in Eq. (2.8b), we can express the optimal cost value as a function of x_k only, i.e., $J(\hat{\boldsymbol{x}}_k^*, \hat{\boldsymbol{u}}_k^*) = J^*(x_k)$.

Also, let $J_N^*(x)$ denote the optimal cost value function of the NMPC problem (2.8) with prediction horizon $N \geq 1$.

In order to ensure that the NMPC problem (2.8) admits a solution, we introduce the following additional assumption:

Assumption 2.4

a) The stage cost function ℓ is continuous on $\mathcal{X} \times \mathcal{U}$; the terminal cost function V_f is continuous on $\mathcal{X}_f$.

b) The terminal set $\mathcal{X}_f$ is compact.

As proved in [133, Proposition 2.4], under Assumptions 2.2 and 2.4, together with $\mathcal{X}$ and $\mathcal{U}$ being closed sets by Eq. (2.2), the NMPC problem (2.8) admits a solution.

Moreover, Assumptions 2.2 and 2.4a also ensure that the cost value function J is continuous [133, Proposition 2.1]. Instead, the optimal cost value function J^* is not necessarily continuous, even if J is continuous [133]. We shall see how to deal with this aspect in Section 2.2.1.

System (2.1) is controlled by NMPC under the so-called *one-step receding horizon policy*: at each time instant $k \geq 0$, only the first optimal input sample $\hat{u}_{0|k}^*$ is applied to system (2.1); the remainder of $\hat{\boldsymbol{u}}_k^*$ is discarded.

Then, the NMPC problem (2.8) can be represented by a static state-feedback control policy π, as follows:

$$u_k = \hat{u}_{0|k}^* = \pi(x_k), \tag{2.9}$$

and the closed-loop system (2.1), (2.9) evolves as

$$x_{k+1} = f(x_k, \pi(x_k)), \quad k \geq 0. \tag{2.10}$$

Controlling Continuous-Time Nonlinear Systems

So far, we have assumed system (2.1) to be specified in discrete time. The main reason is that the solution of discrete-time optimal control problems is much more practical than their continuous counterpart and allows to make ready use of powerful mathematical programming software (see, e.g., [96, 8, 108, 153]). Moreover, advanced MPC controllers are almost always implemented on digital devices, by sampling the system variables and transmitting the control action to the system at discrete time instants.

However, in practical settings, the system to control is typically a continuous-time (CT) nonlinear plant, i.e.,

$$\dot{x}(t) = f_c(x(t), u(t)), \quad t \in \mathbb{R}_{\geq 0}. \tag{2.11}$$

where $f_c : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \to \mathbb{R}^{n_x}$ is the state dynamics function.

We introduce the following assumptions on system (2.11):

Assumption 2.5

The state $x(t)$ of system (2.11) is known and available at each time instant $t \geq 0$.

Assumption 2.6

a) The function $f(\cdot, u)$ is Lipschitz continuous on $\mathcal{X}$.

b) The function $f(x, \cdot)$ is continuous on $\mathcal{U}$.

Assumption 2.7

The input signal $\boldsymbol{u} = (u(t))_{t\in[0,+\infty)}$ belongs to the space of piecewise continuous signals $\mathscr{U}$.

Remark 2.1

Under Assumptions 2.6 and 2.7, the Picard-Lindelöf theorem guarantees the existence of a unique solution $x(t)$ of system (2.11) in the time interval $[0, +\infty)$, starting from x_0 [77].

Assumption 2.8

System (2.11) admits a manifold of equilibrium points, given by

$$\mathcal{Z}_r^c = \{(x_r, u_r) \in \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} : f_c(x_r, u_r) = 0\}. \tag{2.12}$$

In order to control the CT plant (2.11) using the same DT formulation of the NMPC problem (2.8), we adopt the following measures:

1) We construct a DT model of the CT plant (2.11), to be used as prediction model (2.8c) within the NMPC problem (2.8).

   To this end, we *discretize* the CT system (2.11), with discrete time step $T_s > 0$, obtaining a DT system matching Eq. (2.1), i.e., $x_{k+1} = f(x_k, u_k)$, where $x_k \approx x(kT_s)$ and $u_k = u(kT_s)$, $k \geq 0$.

   To obtain a meaningful discretization, we introduce a more tight assumption on the input signal:

Assumption 2.9

The input signal $\boldsymbol{u} = (u(t))_{t\in[0,+\infty)}$ is piecewise constant over the time intervals $[kT_s, (k+1)T_s)$, $k \geq 0$, i.e.,

$$u(t) = u_k, \quad \forall\, t \in [kT_s, (k+1)T_s). \tag{2.13}$$

Indeed, this assumption is a particular case of Assumption 2.7.

In most cases, the discretization introduces an approximation error in the state transition function f, due to which the states x_k and $x(kT_s)$ do not match exactly over time $k \geq 0$.

Among the existing discretization methods, the most relevant ones are the following:

- *Exact discretization.* This methods applies to linear/affine CT systems, i.e.,

$$\dot{x}(t) = A_c x(t) + B_c u(t) + b_c. \tag{2.14}$$

The discretized system is derived *exactly* through the following convolution integral:

$$\begin{aligned} x((k+1)T_s) &= e^{A_c T_s} x(kT_s) + \int_{kT_s}^{(k+1)T_s} e^{A_c((k+1)T_s-\tau)}(B_c u(\tau) + b_c)d\tau \\ &= e^{A_c T_s} x(kT_s) + e^{A_c(k+1)T_s} \int_{kT_s}^{(k+1)T_s} e^{-A_c\tau} d\tau\ (B_c u_k + b_c) \\ &= e^{A_c T_s} x(kT_s) + (e^{A_c T_s} - I)A_c^{-1}(B_c u_k + b_c) \end{aligned} \tag{2.15}$$

$$\Rightarrow \quad x_{k+1} = Ax_k + Bu_k + b, \tag{2.16}$$

where $x_k = x(kT_s)$, $u_k = u(kT_s)$, $A = e^{A_c T_s}$, $B = (e^{A_c T_s} - I)A_c^{-1}B_c$, and $b = (e^{A_c T_s} - I)A_c^{-1}b_c$.

We see that no approximations have been introduced in the discretized system, ensuring the exact match between x_k and $x(kT_s)$ for all $k \geq 0$.

- *Forward Euler method.* This method relies on approximating the time derivative with a finite difference, i.e.,

$$\begin{aligned} &\dot{x}(kT_s) = f_c(x(kT_s), u(kT_s)) \approx \frac{x((k+1)T_s) - x(kT_s)}{T_s}, \\ &x((k+1)T_s) \approx x(kT_s) + T_s f_c(x(kT_s), u(kT_s)) \end{aligned} \tag{2.17}$$

$$\Rightarrow \quad x_{k+1} = x_k + T_s f_c(x_k, u_k) = f_{\mathrm{E}}(x_k, u_k), \tag{2.18}$$

where $x_k \approx x(kT_s)$ and $u_k = u(kT_s)$.

- *Explicit Runge-Kutta methods.* An explicit Runge-Kutta method of order n (RKn) advances the state $x_k \approx x(kT_s)$ via n stage evaluations, i.e.,

$$k_i = f_c\Big(x_k + T_s \sum_{j=1}^{i-1} a_{ij} k_j, u_k\Big), \quad i = 1, \ldots, n, \tag{2.19a}$$

$$x_{k+1} = x_k + T_s \sum_{i=1}^{n} b_i k_i(x_k, u_k) = f_{\mathrm{RK}n}(x_k, u_k). \tag{2.19b}$$

The coefficients $(a_{ij})_{1\le j<i\le n}$, $(b_i)_{i=1}^n$, and $(c_i)_{i=2}^n$ are summarized in a Butcher tableau [29].

The most widely known member of the Runge-Kutta family is the fourth-order method, i.e., Runge-Kutta 4 (RK4). Also, the RK1 method is equivalent to the forward Euler method.

Remark 2.2

It is worth noticing that all the above discretization methods preserve the equilibrium manifold (2.5), (2.12), i.e., for the CT system (2.11) and its discretization (2.1), it holds that

$$\mathcal{Z}_r^c = \mathcal{Z}_r. \tag{2.20}$$

2) At each time instant $k \ge 0$, we pass the current state $x(kT_s) = x_k$ to the NMPC problem (2.8).

Then, the CT plant (2.11) is controlled by NMPC under the one-step receding horizon policy as follows: at each $k \ge 0$, the first optimal input sample $\hat{u}_{0|k}^*$ is applied to the plant (2.11) over the continuous time interval $[kT_s, (k+1)T_s)$.

Hence, the control input signal $\boldsymbol{u} = (u(t))_{t\in[0,+\infty)}$ is piecewise constant, i.e.,

$$u(t) = u_k = \hat{u}_{0|k}^*, \quad \forall\, t \in [kT_s, (k+1)T_s), \tag{2.21}$$

satisfying Assumption 2.9.

Remark 2.3

With the exception of exact discretization, all the other methods described above introduce a mismatch between the NMPC prediction and the actual closed-loop state trajectory of system (2.11). This mismatch can, in principle, compromise the closed-loop stability guarantees that are going to be introduced in the next section.

Such an issue can be formally tackled by robustifying the NMPC problem (2.8) (see, e.g., [64, Theorems 7.36, 7.41, 11.10]). In practice, however, this is hardly ever rigorously ensured. The reason for this is that, for "good" discretization methods, numerical errors are usually very small compared to other error sources like model errors, exogenous disturbances, etc. Although even very small errors may, in the worst case, be destabilizing, this is very unlikely to happen and such phenomena are hardly ever observed in simulations or practical examples [64].

Hence, for most practical cases involving a CT plant with non-exact DT prediction model, stability guarantees are retained, provided that the numerical errors are sufficiently small.

2.2.1. Nonlinear MPC With Stabilizing Terminal Ingredients

In the following, we analyze the NMPC problem (2.8) with the inclusion of terminal ingredients, namely the terminal cost function V_f in Eq. (2.8a) and the terminal constraint

set $\mathcal{X}_f$ in Eq. (2.8f). These terminal ingredients are used to establish closed-loop stability and recursive feasibility guarantees for the NMPC problem (2.8), in the case of regulation and tracking control tasks.

Before delving into this topic, we introduce some important concepts that are needed in the subsequent analysis:

Definition 2.2 (Feasibility)

We say that the NMPC problem (2.8) is *feasible* for the current state $x_k \in \mathcal{X}$ if there exists an admissible input sequence $\hat{\boldsymbol{u}} \in \mathcal{U}^{N_p}$ such that the state sequence $\hat{\boldsymbol{x}}$ given by

$$\hat{x}_0 = x_k, \tag{2.22a}$$
$$\hat{x}_{i+1} = f(x_i, u_i), \quad i = 0, \dots, N_p - 1, \tag{2.22b}$$

is admissible, i.e., $\hat{\boldsymbol{x}} \in \mathcal{X}^{N_p+1}$, and $\hat{x}_{N_p} \in \mathcal{X}_f$.

The state x_k is called *feasible state* of problem (2.8); the input sequence $\hat{\boldsymbol{u}}$ is called *feasible input sequence*; the state sequence $\hat{\boldsymbol{x}}$ given by $\hat{\boldsymbol{u}}$ is called *feasible state sequence*.

Definition 2.3 (Recursive feasibility)

The NMPC problem (2.8) is *recursively feasible* for the initial state $x_0 \in \mathcal{X}$ if x_0 is feasible and the closed-loop successor state $x_{k+1} = f(x_k, \pi(x_k))$ is again feasible for all $k \geq 0$.

The state x_0 is called *recursively feasible state* of problem (2.8).

Definition 2.4 (Feasible sets)

Given the NMPC problem (2.8), we define:

a) the *set of feasible input sequences* $\mathscr{U}_N(x)$ for horizon $N \in \mathbb{N}$, $0 \leq N \leq N_p$, and initial state $x \in \mathcal{X}$ by

$$\begin{aligned} \mathscr{U}_N(x) = \{\hat{\boldsymbol{u}} \in \mathcal{U}^N : \exists\, \hat{\boldsymbol{x}} \in \mathcal{X}^{N+1},\ & \hat{x}_0 = x, \\ & \hat{x}_{i+1} = f(\hat{x}_i, \hat{u}_i),\ i = 0, \dots, N-1, \\ & \hat{x}_N \in \mathcal{X}_f\}, \end{aligned} \tag{2.23}$$

b) the *set of feasible states* $\mathcal{X}_N$ for horizon $N \in \mathbb{N}$, $0 \leq N \leq N_p$, by

$$\mathcal{X}_N = \{x \in \mathcal{X} : \exists\, \hat{\boldsymbol{u}} \in \mathscr{U}_N(x)\}. \tag{2.24}$$

Remark 2.4

By Definition 2.4, for horizon $N = 0$ we have that $\mathcal{X}_0 = \mathcal{X}_f$.

To start our analysis on how the inclusion of terminal ingredients can ensure the closed-loop stability of the NMPC problem (2.8), we begin by outlining the standard approach that is usually employed in the literature to prove the stability of a closed-loop system under the control of a generic NMPC scheme.

Since system (2.1) is controlled by NMPC through the receding horizon policy, the first key condition to ensure is that the NMPC problem (2.8) is recursively feasible (Definition 2.3). This property guarantees that, for a given set of initial states $x_0 \in \mathcal{X}$, the NMPC problem (2.8) is solvable at all subsequent time instants $k \geq 0$, thereby always providing an optimal control input to the system.

Then, we can proceed with the stability analysis. The MPC literature usually resorts to Lyapunov stability theory to establish closed-loop stability guarantees. From a practical point of view, what is typically done is proving that the optimal cost value $J^*(x)$ is a Lyapunov function of the closed-loop system (2.10).

Regulation Control Task

First, let us focus on the control task of regulation towards an admissible equilibrium state $x_r \in \mathcal{X}$ of system (2.1), with admissible input $u_r \in \mathcal{U}$ such that (x_r, u_r) is an equilibrium point of system (2.1).

Assumption 2.10 (Stage cost)

The stage cost function $\ell : \mathcal{X} \times \mathcal{U} \to \mathbb{R}_{\geq 0}$ in Eq. (2.8a) is positive definite in $\mathcal{X} \times \mathcal{U}$ wrt (x_r, u_r), i.e.,

$$\ell(x_r, u_r) = 0, \tag{2.25a}$$

$$\ell(x, u) > 0, \quad \forall\, (x, u) \in (\mathcal{X} \times \mathcal{U}) \setminus \{(x_r, u_r)\}. \tag{2.25b}$$

Remark 2.5 (Quadratic stage cost)

A "popular" choice for the stage cost ℓ meeting Assumption 2.10 is given by the following quadratic function:

$$\ell(x, u) = \|x - x_r\|_Q^2 + \|u - u_r\|_R^2. \tag{2.26}$$

Assumption 2.11 (Terminal ingredients)

a) The terminal set $\mathcal{X}_f$ in Eq. (2.8f) is compact subset of $\mathcal{X}$ and contains the reference state x_r, i.e., $x_r \in \mathcal{X}_f$.

b) The terminal set $\mathcal{X}_f$ is control invariant for system (2.1), i.e.,

$$\forall\, x \in \mathcal{X}_f,\ \exists\, u \in \mathcal{U} :\ f(x, u) \in \mathcal{X}_f. \tag{2.27}$$

More specifically, there exists a terminal control law $\kappa_f : \mathcal{X}_f \to \mathcal{U}$ such that $u_r = \kappa_f(x_r)$ and

$$\forall\, x \in \mathcal{X}_f,\ f(x, \kappa_f(x)) \in \mathcal{X}_f. \tag{2.28}$$

c) The terminal cost $V_f : \mathcal{X}_f \to \mathbb{R}_{\geq 0}$ in Eq. (2.8a) is a positive semidefinite function in $\mathcal{X}_f$ wrt x_r.

d) The terminal cost V_f is such that

$$V_f(f(x, \kappa_f(x))) - V_f(x) \leq -\ell(x, \kappa_f(x)), \quad \forall\, x \in \mathcal{X}_f. \tag{2.29}$$

Remark 2.6

The "simplest" terminal set satisfying Assumptions 2.11a and 2.11b is given by the singleton

$$\mathcal{X}_f = \{x_r\}, \tag{2.30}$$

since $f(x_r, u_r) = x_r \in \mathcal{X}_f$. As a consequence, the related terminal cost, by Assumption 2.11c, degenerates into the identically-null function, i.e.,

$$V_f(x) = 0. \tag{2.31}$$

Recursive Feasibility As outlined at the beginning of this section, we start by ensuring the recursive feasibility of the NMPC problem (2.8). For further details on the following results, we refer the reader to [64].

Lemma 2.1

If Assumption 2.11b holds, then, for each $N \geq 2$, we have that:

(i) For each feasible state $x_0 \in \mathcal{X}_{N-1}$, with related feasible sequences $\hat{\boldsymbol{u}} \in \mathscr{U}_{N-1}(x_0)$ and $\hat{\boldsymbol{x}}$, the prolonged sequence

$$\hat{u}'_i = \hat{u}_i, \quad i = 0, \ldots, N-2, \quad \hat{u}'_{N-1} = \kappa_f(\hat{x}_{N-1}) \tag{2.32}$$

is also feasible, i.e., $\hat{\boldsymbol{u}}' \in \mathscr{U}_N(x_0)$.

(ii) The inclusion $\mathcal{X}_{N-1} \subseteq \mathcal{X}_N$ holds.

Proof

(i) In order to prove $\hat{\boldsymbol{u}}' \in \mathscr{U}_N(x_0)$, we have to verify that $(\hat{x}'_{N-1}, \hat{u}'_{N-1}) \in \mathcal{X} \times \mathcal{U}$ and $\hat{x}'_N \in \mathcal{X}_f$.

By Eq. (2.32), $\hat{x}'_{N-1} = \hat{x}_{N-1} \in \mathcal{X}_f \subseteq \mathcal{X}$; $\hat{u}'_{N-1} = \kappa_f(\hat{x}_{N-1}) \in \mathcal{U}$ by Assumption 2.11b; $\hat{x}'_N = f(\hat{x}'_{N-1}, \hat{u}'_{N-1}) = f(\hat{x}'_{N-1}, \kappa(\hat{x}_{N-1})) = f(\hat{x}'_{N-1}, \kappa(\hat{x}'_{N-1})) \in \mathcal{X}_f$ by Assumption 2.11b. Thus, $\hat{\boldsymbol{u}}' \in \mathscr{U}_N(x_0)$.

(ii) Let $x_0 \in \mathcal{X}_{N-1}$, for which there exists $\hat{\boldsymbol{u}} \in \mathscr{U}_{N-1}(x_0)$. By (i), there also exists $\hat{\boldsymbol{u}}' \in \mathscr{U}_N(x_0)$. Thus, $\mathscr{U}_N(x_0) \neq 0$, from which $x_0 \in \mathcal{X}_N$ follows.

Lemma 2.2

The set $\mathcal{X}_N$ is forward invariant for the closed-loop system (2.10), i.e.,

$$\forall\, x \in \mathcal{X}_N,\ f(x, \pi(x)) \in \mathcal{X}_N. \tag{2.33}$$

Proof

First, we show that $\forall\, x \in \mathcal{X}_N$, $f(x, \pi(x)) \in \mathcal{X}_{N-1}$. Let us consider the feasible sequences $\hat{\boldsymbol{u}}$ and $\hat{\boldsymbol{x}}$ associated with the feasible state x. It holds that $x = \hat{x}_0$, $\pi(x) = \hat{u}_0$ (by the receding horizon policy), and $f(x, \pi(x)) = f(\hat{x}_0, \hat{u}_0) = \hat{x}_1$. Then, the shortened

sequence

$$(\hat{x}'_{i-1}, \hat{u}'_{i-1}) = (\hat{x}_i, \hat{u}_i), \quad i = 1, \dots, N-1, \quad \hat{x}'_N = \hat{x}_{N-1}, \tag{2.34}$$

is feasible and, thus, $\hat{x}'_0 = \hat{x}_1 = f(x, \pi(x)) \in \mathcal{X}_{N-1}$.

Then, the claim follows immediately from Lemma 2.1(ii), as $f(x, \pi(x)) \in \mathcal{X}_{N-1} \subseteq \mathcal{X}_N$.

Theorem 2.1 (Recursive feasibility)

Consider the NMPC problem (2.8) and let Assumptions 2.11a and 2.11b hold.

Then, the NMPC problem (2.8) is recursively feasible for all initial states $x_0 \in \mathcal{X}_{N_p}$.

Proof

Follows immediately from Lemma 2.2 by setting $N = N_p$.

With Theorem 2.1, we have shown that, by including the terminal constraint (2.8f), with terminal set $\mathcal{X}_f$ satisfying Assumptions 2.11a and 2.11b, we are able to ensure the recursive feasibility of the NMPC problem (2.8) for the regulation control task.

As noted in Remark 2.6, the simplest choice for the terminal set, ensuring recursive feasibility, is the singleton $\mathcal{X}_f = \{x_r\}$. This fact is established in the following Corollary:

Corollary 2.1 (Recursive feasibility with singleton terminal set)

Under the statement of Theorem 2.1, let the terminal set be $\mathcal{X}_f = \{x_r\}$.

Then, the NMPC problem (2.8) is recursively feasible for all initial states $x_0 \in \mathcal{X}_{N_p}$.

Proof

Follows immediately from Theorem 2.1 and Remark 2.6.

Remark 2.7

The straightforward choice $\mathcal{X}_f = \{x_r\}$ comes at the price of obtaining a smaller set $\mathcal{X}_{N_p}$, i.e., the set of recursively feasible states [102]. This effect is illustrated for the case of Linear-Quadratic MPC in Section 2.3, Example 2.1.

Indeed, by setting $\mathcal{X}_f = \{x_r\}$, we are implicitly requiring system (2.1) to be exactly controllable to x_r in N_p steps (i.e., in finite time), so to ensure that the feasible set $\mathcal{X}_{N_p}$ is non-empty and contains a neighborhood of x_r.

Thus, the singleton $\mathcal{X}_f = \{x_r\}$ cannot be employed for systems (2.1) that are merely stabilizable but not controllable to x_r [64], since, for these systems, $\mathcal{X}_{N_p} = \emptyset$.

In such cases, we have to construct a larger terminal set $\mathcal{X}_f$, satisfying Assumptions 2.11a and 2.11b. This point is going to be analyzed in the following (p. 24).

Closed-Loop Stability After having ensured recursive feasibility in Theorem 2.1, we can now proceed with the stability analysis of the NMPC problem (2.8). For further details on the following results, we refer the reader to [64].

Lemma 2.3

Consider the NMPC problem (2.8), with stage cost ℓ satisfying Assumption 2.10 and terminal cost V_f satisfying Assumption 2.11c. Let π be the related control policy as in Eq. (2.9). Then, it holds that

$$u_r = \pi(x_r). \tag{2.35}$$

Additionally, x_r is an equilibrium of the closed-loop system (2.10).

Proof

By Assumptions 2.10 and 2.11c, the cost function $J(\hat{\boldsymbol{x}}_k, \hat{\boldsymbol{u}}_k)$ can only take values in $\mathbb{R}_{\geq 0}$. Specifically, we can see that J is globally minimized to 0 by the feasible state and input sequences

$$(\hat{x}_i, \hat{u}_i) = (x_r, u_r), \;\; i = 0, \ldots, N_p - 1, \quad \hat{x}_{N_p} = x_r. \tag{2.36}$$

Any other feasible sequence must deviate at least once from (x_r, u_r), yielding a cost value $J > 0$. Thus, Eq. (2.36) is the unique, global minimizer of J.

Therefore, by the receding horizon policy, it holds that $u_r = \pi(x_r)$.

Also, it immediately follows that x_r is an equilibrium of the closed-loop system (2.10), as $f(x_r, \pi(x_r)) = f(x_r, u_r) = x_r$.

As outlined at the beginning of this section, to establish closed-loop stability guarantees, we would like to resort to Lyapunov stability theory, employing the optimal cost value $J^* : \mathcal{X}_{N_p} \to \mathbb{R}_{\geq 0}$ as a Lyapunov function for the closed-loop system (2.10).

In order for $J^*(x)$ to be a Lyapunov function, the following well-known conditions must hold:

(i) $J^*(x)$ is continuous on $\mathcal{X}_{N_p}$.

(ii) $J^*(x)$ is positive definite on $\mathcal{X}_{N_p}$ wrt x_r, i.e.,

$$J^*(x_r) = 0, \quad J^*(x) > 0, \;\; \forall\, x \in \mathcal{X}_{N_p} \setminus \{x_r\}. \tag{2.37}$$

(iii) $J^*(f(x, \pi(x))) - J^*(x)$ is negative definite on $\mathcal{X}_{N_p}$ wrt x_r, i.e.,

$$J^*(f(x_r, \pi(x_r))) - J^*(x_r) = 0, \tag{2.38a}$$

$$J^*(f(x, \pi(x))) - J^*(x) < 0, \;\; \forall\, x \in \mathcal{X}_{N_p} \setminus \{x_r\}. \tag{2.38b}$$

Among these conditions, we see that (i), as noted in Section 2.2, is not necessarily satisfied by $J^*(x)$. However, in Lyapunov stability theory, the continuity condition (i) is sufficient, but it can be more restrictive than necessary. Indeed, such a condition can be relaxed by requiring an additional state-dependent upper bound on $J^*(x)$, as suggested in [64, Theorem 2.19].

Definition 2.5 (Class $\mathcal{K}$ functions [77])

a) A function $\alpha : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$ is said to belong to *class* $\mathcal{K}$ if

(i) it is continuous;

(ii) it is strictly increasing, i.e., $\alpha(r) > \alpha(s)$ if $r > s$;

(iii) $\alpha(0) = 0$.

b) A function $\alpha : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$ is said to belong to *class* $\mathcal{K}_\infty$ if it belongs to class $\mathcal{K}$ and $\alpha(r) \to +\infty$ as $r \to +\infty$.

Then, condition (i) can be relaxed by requiring that there exists a function $\alpha \in \mathcal{K}_\infty$ such that

$$J^*(x) \leq \alpha(\|x - x_r\|), \quad \forall\, x \in \mathcal{X}_{N_p}. \tag{2.39}$$

A sufficient condition for Eq. (2.39) to hold is given in [64, Proposition 5.14(ii)], which introduces a further assumption on the terminal set $\mathcal{X}_f$ and terminal cost V_f:

Assumption 2.12

a) The terminal set $\mathcal{X}_f$ contains a ball $\mathcal{B}_\delta(x_r)$, $\delta > 0$.

b) There exists a function $\alpha_f \in \mathcal{K}_\infty$ such that

$$V_f(x) \leq \alpha_f(\|x - x_r\|), \quad \forall\, x \in \mathcal{B}_\delta(x_r). \tag{2.40}$$

By [64, Lemma 5.12], it can be shown that $J^*(x) \leq V_f(x) \leq \alpha_f(\|x - x_r\|)$, for all $x \in \mathcal{B}_\delta(x_r)$. Then, [64, Proposition 5.14(ii)] proves that such a local bound implies the existence of a function $\alpha \in \mathcal{K}_\infty$, such that $J^*(x) \leq \alpha(\|x - x_r\|)$, for all $x \in \mathcal{X}_{N_p}$, verifying Eq. (2.39).

Assumption 2.12b represents a non-restrictive condition on the terminal cost V_f since, as we shall see on p. 24, V_f is typically constructed as a quadratic function.

Theorem 2.2 (Closed-loop stability)

Consider the NMPC problem (2.8). Let the stage cost ℓ, the terminal set $\mathcal{X}_f$, and the terminal cost V_f satisfy Assumptions 2.10-2.12.

Then, x_r is an asymptotically stable equilibrium of the closed-loop system (2.10), with region of attraction $\mathcal{X}_{N_p}$.

Proof

Consider the optimal cost value function $J^* : \mathcal{X}_{N_p} \to \mathbb{R}_{\geq 0}$ of the NMPC problem (2.8). We want to show that $J^*(x)$ is a Lyapunov function on $\mathcal{X}_{N_p}$, wrt x_r, of the closed-loop system (2.10).

Eq. (2.37) follows under the same arguments used to prove Lemma 2.3.

Eq. (2.38a) follows from Lemma 2.3, since $J^*(f(x_r, \pi(x_r))) = J^*(x_r)$.

We now prove that Eq. (2.38b) holds. Let $\hat{\boldsymbol{x}}^* = (\hat{x}_i^*)_{i=0}^{N_p}$ and $\hat{\boldsymbol{u}}^* = (\hat{u}_i^*)_{i=0}^{N_p-1}$ be the optimal state trajectory and input sequence, solving the NMPC problem (2.8), with initial state $\hat{x}_0^* = x$ by Eq. (2.8b).

The optimal cost value is given by

$$J^*(x) = J(\hat{\boldsymbol{x}}^*, \hat{\boldsymbol{u}}^*) = \sum_{i=0}^{N_p-1} \ell(\hat{x}_i^*, \hat{u}_i^*) + V_f(\hat{x}_{N_p}^*). \tag{2.41}$$

By the receding horizon policy, $\pi(x) = \hat{u}_0^*$ and $x^+ = f(x, \pi(x)) = f(\hat{x}_0^*, \hat{u}_0^*) = \hat{x}_1^*$.

Now, we overestimate the optimal cost value $J^*(x^+)$ by considering a feasible, but suboptimal, shifted state trajectory $\tilde{\boldsymbol{x}} = (\tilde{x}_i)_{i=0}^{N_p}$ and input sequence $\tilde{\boldsymbol{u}} = (\tilde{u}_i)_{i=0}^{N_p-1}$ of the NMPC problem (2.8), with initial state $\tilde{x}_0 = x^+ = \hat{x}_1^*$, i.e.,

$$\tilde{u}_i = \hat{u}_{i+1}^*, \;\; i = 0, \dots, N_p - 2, \quad \tilde{u}_{N_p-1} = \kappa_f(\tilde{x}_{N_p-1}), \tag{2.42}$$

$$\tilde{x}_i = \hat{x}_{i+1}^*, \;\; i = 0, \dots, N_p - 1, \quad \tilde{x}_{N_p} = f(\tilde{x}_{N_p-1}, \tilde{u}_{N_p-1}). \tag{2.43}$$

To show that $\tilde{\boldsymbol{x}}$ and $\tilde{\boldsymbol{u}}$ are feasible, we verify that $(\tilde{x}_{N_p-1}, \tilde{u}_{N_p-1}) \in \mathcal{X} \times \mathcal{U}$ and $\tilde{x}_{N_p-1} \in \mathcal{X}_f$: by Eq. (2.42), $\tilde{x}_{N_p-1} = \hat{x}_{N_p}^* \in \mathcal{X}_f \subseteq \mathcal{X}$; $\tilde{u}_{N_p-1} = \kappa_f(\tilde{x}_{N_p-1}) \in \mathcal{U}$ by Assumption 2.11b; $\tilde{x}_{N_p} = f(\tilde{x}_{N_p-1}, \tilde{u}_{N_p-1}) = f(\hat{x}_{N_p}^*, \kappa(\tilde{x}_{N_p-1})) = f(\hat{x}_{N_p}^*, \kappa(\hat{x}_{N_p}^*)) \in \mathcal{X}_f$ by Assumption 2.11b.

Then, we have that

$$\begin{aligned}
J^*(x^+) \leq J(\tilde{\boldsymbol{x}}, \tilde{\boldsymbol{u}}) &= \sum_{i=0}^{N_p-1} \ell(\tilde{x}_i, \tilde{u}_i) + V_f(\tilde{x}_{N_p}) \\
&= \sum_{i=1}^{N_p-1} \ell(\hat{x}_i^*, \hat{u}_i^*) + \ell(\hat{x}_{N_p}^*, \kappa(\hat{x}_{N_p}^*)) + V_f(f(\hat{x}_{N_p}^*, \kappa(\hat{x}_{N_p}^*))) \\
&= J^*(x) - \ell(\hat{x}_0^*, \hat{u}_0^*) - V_f(\hat{x}_{N_p}^*) + \ell(\hat{x}_{N_p}^*, \kappa(\hat{x}_{N_p}^*)) + V_f(f(\hat{x}_{N_p}^*, \kappa(\hat{x}_{N_p}^*))) \\
&\leq J^*(x) - \ell(\hat{x}_0^*, \hat{u}_0^*),
\end{aligned} \tag{2.44}$$

since $\hat{x}_{N_p}^* \in \mathcal{X}_f$ and, by Assumption 2.11d,

$$V_f(f(\hat{x}_{N_p}^*, \kappa(\hat{x}_{N_p}^*))) - V_f(\hat{x}_{N_p}^*) + \ell(\hat{x}_{N_p}^*, \kappa(\hat{x}_{N_p}^*)) \leq 0. \tag{2.45}$$

Then, recalling that $\hat{x}_0^* = x$ and $\hat{u}_0^* = \pi(x)$,

$$J^*(f(x, \pi(x))) - J^*(x) \leq -\ell(x, \pi(x)). \tag{2.46}$$

By Assumption 2.10 and Lemma 2.3, $-\ell(x, \pi(x)) < 0$, $\forall\, x \in \mathcal{X}_{N_p} \setminus \{x_r\}$, yielding

$$J^*(f(x, \pi(x))) - J^*(x) < 0, \;\; \forall\, x \in \mathcal{X}_{N_p} \setminus \{x_r\}, \tag{2.47}$$

which verifies Eq. (2.38b).

Finally, by Assumption 2.12, Eq. (2.39) holds, leveraging [64, Proposition 5.14(ii)].

Therefore, we conclude that, by Lyapunov stability theory, x_r is an asymptotically stable equilibrium of the closed-loop system (2.10), with region of attraction $\mathcal{X}_{N_p}$.

With Theorem 2.2, we have shown that, by including the terminal constraint (2.8f), with terminal set $\mathcal{X}_f$ satisfying Assumptions 2.11a, 2.11b, and 2.12a, and the terminal cost V_f in Eq. (2.8a), satisfying Assumption 2.11c, 2.11d, and 2.12b, we are able to ensure the closed-loop stability of the NMPC problem (2.8) for the regulation control task.

By Remark 2.6, the simplest choice for the terminal set and terminal cost, ensuring closed-loop stability, is the singleton $\mathcal{X}_f = \{x_r\}$ and the identically-null function $V_f(x) = 0$, respectively.

Differently from the previous case, since now $\mathcal{X}_f = \{x_r\}$ is a singleton, Assump-

tion 2.12a cannot hold anymore, thus not allowing to satisfy Eq. (2.39) through [64, Proposition 5.14(ii)]. As a consequence, we need to introduce a modified assumption, according to [64, Proposition 5.7(ii)]:

Assumption 2.13

a) The state constraint set $\mathcal{X}$ contains a ball $\mathcal{B}_\delta(x_r)$, $\delta > 0$.

b) There exists a function $\alpha_\ell \in \mathcal{K}_\infty$ such that, for all $x \in \mathcal{B}_\delta(x_r)$, an admissible input $u \in \mathcal{U}$ exists for which

$$f(x,u) = x_r, \quad \ell(x,u) \leq \alpha_\ell(\|x - x_r\|). \tag{2.48}$$

By Assumption 2.13, $u \in \mathscr{U}_1(x)$ for $x \in \mathcal{B}_\delta(x_r)$, meaning that $\mathcal{B}_\delta(x_r) \subseteq \mathcal{X}_1$ and $J_1^*(x) \leq \alpha_\ell(\|x - x_r\|)$. Then, by [64, Lemma 5.4], it holds that $J^*(x) = J_{N_p}^*(x) \leq J_1^*(x) \leq \alpha_\ell(\|x - x_r\|)$, for all $N_p \geq 1$ and for all $x \in \mathcal{B}_\delta(x_r)$. Finally, [64, Proposition 5.7(ii)] proves that such a local bound implies the existence of a function $\alpha \in \mathcal{K}_\infty$, such that $J^*(x) \leq \alpha(\|x - x_r\|)$, for all $x \in \mathcal{X}_{N_p}$, verifying Eq. (2.39).

Assumption 2.13b enforces that system (2.1) must be locally controllable in 1-step to x_r, requiring also that the cost of this 1-step move is upper bounded by some function of class $\mathcal{K}_\infty$.

Such an assumption can be further relaxed by considering a k-step controllability condition on system (2.1), i.e., for all $x \in \mathcal{B}_\delta(x_r)$, there exists an input sequence $\boldsymbol{u} = (u_i)_{i=0}^{k-1} \in \mathscr{U}_k(x)$, with related state sequence $\boldsymbol{x} = (x_i)_{i=0}^{k}$, $x_0 = x$, such that $\sum_{i=0}^{k-1} \ell(x_i, u_i) \leq \alpha_\ell(\|x - x_r\|)$. Under the same arguments in [64, Proposition 5.7(ii)], this choice also verifies Eq. (2.39).

Corollary 2.2 (Closed-loop stability with point terminal ingredients)

Under the statement of Theorem 2.2, let the terminal ingredients be

$$\mathcal{X}_f = \{x_r\}, \quad V_f(x) = 0. \tag{2.49}$$

Let Assumption 2.13 hold.

Then, x_r is an asymptotically stable equilibrium of the closed-loop system (2.10), with region of attraction $\mathcal{X}_{N_p}$.

Proof

The proof unfolds as in Theorem 2.2, with the following differences.

The optimal cost value $J^*(x)$ is given by

$$J^*(x) = J(\hat{\boldsymbol{x}}^*, \hat{\boldsymbol{u}}^*) = \sum_{i=0}^{N_p-1} \ell(\hat{x}_i^*, \hat{u}_i^*). \tag{2.50}$$

The suboptimal shifted state trajectory $\tilde{\boldsymbol{x}}$ and input sequence $\tilde{\boldsymbol{u}}$ are given by

$$\tilde{u}_i = \hat{u}_{i+1}^*, \quad i = 0, \ldots, N_p - 2, \quad \tilde{u}_{N_p-1} = u_r, \tag{2.51}$$

$$\tilde{x}_i = \hat{x}_{i+1}^*, \quad i = 0, \ldots, N_p - 1, \quad \tilde{x}_{N_p} = f(\tilde{x}_{N_p-1}, \tilde{u}_{N_p-1}) = x_r, \tag{2.52}$$

whose feasibility is verified, as $\tilde{x}_{N_p-1} = \hat{x}^*_{N_p} \in \mathcal{X}_f \subseteq \mathcal{X}$, $\tilde{u}_{N_p-1} = u_r \in \mathcal{U}$ and $\tilde{x}_{N_p} = f(\tilde{x}_{N_p-1}, \tilde{u}_{N_p-1}) = f(x_r, u_r) = x_r \in \mathcal{X}_f$.

Then, we compute

$$\begin{aligned} J^*(x^+) \leq J(\tilde{\boldsymbol{x}}, \tilde{\boldsymbol{u}}) &= \sum_{i=0}^{N_p-1} \ell(\tilde{x}_i, \tilde{u}_i) \\ &= \sum_{i=1}^{N_p-1} \ell(\hat{x}^*_i, \hat{u}^*_i) + \underbrace{\ell(\tilde{x}_{N_p-1}, \tilde{u}_{N_p-1})}_{=\ell(x_r,u_r)=0} \\ &= J^*(x) - \ell(\hat{x}^*_0, \hat{u}^*_0) \end{aligned} \tag{2.53}$$

$$\Rightarrow \quad J^*(f(x, \pi(x))) - J^*(x) \leq -\ell(x, \pi(x)) < 0, \quad \forall\, x \in \mathcal{X}_{N_p} \setminus \{x_r\}. \tag{2.54}$$

Finally, by Assumption 2.13, Eq. (2.39) holds, leveraging [64, Proposition 5.7(ii)].

Remark 2.8

The straightforward choice $\mathcal{X}_f = \{x_r\}$ and $V_f(x) = 0$, as also highlighted in Remark 2.7, leads to a smaller feasible set $\mathcal{X}_{N_p}$, which, in the case of closed-loop stability, is also the region of attraction of the asymptotically stable equilibrium x_r. This effect is illustrated for the case of Linear-Quadratic MPC in Section 2.3, Example 2.1.

As a consequence, Corollary 2.2 does not hold for systems (2.1) that are stabilizable, but not controllable, to x_r, since $\mathcal{X}_{N_p} = \emptyset$. This aspect is further highlighted by Assumption 2.13b.

In such cases, we have to construct a larger terminal set $\mathcal{X}_f$ and a related terminal cost V_f, satisfying Assumption 2.11. This point is going to be analyzed in the following.

Constructing the Terminal Ingredients For nonlinear systems whose linearization at (x_r, u_r) is stabilizable, $\mathcal{X}_f$ and V_f can be constructed by a linear-quadratic approach via the corresponding Riccati equation [64]. Further details on the following construction can be found in [64].

Let us start by considering system (2.1), i.e., $x^+ = f(x, u)$, and assuming that f is at least C^1-smooth near (x_r, u_r). Then, we can compute

$$A = \frac{\partial f}{\partial x}(x_r, u_r), \quad B = \frac{\partial f}{\partial u}(x_r, u_r), \tag{2.55}$$

and write

$$f(x, u) = \underbrace{f(x_r, u_r)}_{=x_r} + A(x - x_r) + B(u - u_r) + \Delta(x, u), \tag{2.56}$$

where for the reminder $\Delta(x, u)$, by Taylor's theorem, there exist constants $\delta_f > 0$ and $c_f > 0$ such that

$$\|\Delta(x, u)\| \leq c_f(\|x - x_r\| + \|u - u_r\|)^2, \quad \forall\, (x, u) : \|(x - x_r, u - u_r)\| \leq \delta_f, \tag{2.57}$$

Then, let us consider a stage cost ℓ satisfying Assumption 2.10, and assume also that ℓ is at least C^2-smooth near (x_r, u_r). Hence, it holds that

$$\ell(x,u) = \underbrace{\ell(x_r,u_r)}_{=0} + \underbrace{\nabla\ell(x_r,u_r)}_{=\mathbf{0}^\top}\begin{bmatrix}x-x_r\\u-u_r\end{bmatrix} + \begin{bmatrix}x-x_r\\u-u_r\end{bmatrix}^\top \underbrace{H_\ell(x_r,u_r)}_{=\left[\begin{smallmatrix}Q & S\\ S^\top & R\end{smallmatrix}\right]}\begin{bmatrix}x-x_r\\u-u_r\end{bmatrix} + r(x,u), \tag{2.58}$$

where H_ℓ is the Hessian matrix of ℓ. Without loss of generality, we consider $Q \succ 0$, $R \succ 0$, and $S = \mathbf{0}_{n_x \times n_u}$. Then, we have that

$$\ell(x,u) = \|x - x_r\|_Q^2 + \|u - u_r\|_R^2 + r(x,u), \tag{2.59}$$

where for the reminder $r(x,u)$, by Taylor's theorem, there exist $\delta_\ell, c_\ell > 0$ such that

$$|r(x,u)| \le c_\ell(\|x - x_r\| + \|u - u_r\|)^3, \quad \forall\,(x,u) : \|(x - x_r, u - u_r)\| \le \delta_\ell. \tag{2.60}$$

Additionally, let us assume that the constraint sets $\mathcal{X}$ and $\mathcal{U}$ (2.2) contain a neighborhood of x_r and u_r, respectively, i.e., there exist $\delta_x, \delta_u > 0$ such that

$$B_{\delta_x}(x_r) = \{x \in \mathbb{R}^{n_x} : \|x - x_r\| \le \delta_x\} \subseteq \mathcal{X}, \tag{2.61a}$$
$$B_{\delta_u}(u_r) = \{u \in \mathbb{R}^{n_u} : \|u - u_r\| \le \delta_u\} \subseteq \mathcal{U}. \tag{2.61b}$$

Consider the couple (A, B) in Eq. (2.56) and assume that it is stabilizable (which holds if, e.g., system (2.1) is stabilizable to x_r). Then, there exists a unique $P \succ 0$ solving the *discrete algebraic Riccati equation* (DARE) with A, B, Q, R in Eqs. (2.56) and (2.59), i.e.,

$$P = A^\top PA - A^\top PB(R + B^\top PB)^{-1}B^\top PA + Q. \tag{2.62}$$

From Eq. (2.62), we can also find the feedback matrix of the infinite-horizon DT linear-quadratic regulator (LQR), which is given by

$$K = -(R + B^\top PB)^{-1}B^\top PA. \tag{2.63}$$

In particular, such a matrix K renders $A + BK$ Schur stable.

Then, let us set the terminal control law as

$$\kappa_f(x) = K(x - x_r) + u_r \tag{2.64}$$

and define the function

$$V(x) = \|x - x_r\|_P^2. \tag{2.65}$$

Lemma 2.4

The following identity holds:

$$\|(A + BK)(x - x_r)\|_P^2 - \|x - x_r\|_P^2 = -(x - x_r)^\top(Q + K^\top RK)(x - x_r). \tag{2.66}$$

Proof

We can expand the left-hand side of Eq. (2.66) as

$$\|(A+BK)(x-x_r)\|_P^2 - \|x-x_r\|_P^2 = (x-x_r)^\top [(A+BK)^\top P(A+BK) - P](x-x_r). \tag{2.67}$$

From the DARE (2.62), $A^\top PA - P = -Q + A^\top PBS^{-1}B^\top PA$, with $S = R + B^\top PB$. Also, from the expression of the LQR matrix K (2.63), $K^\top S = -A^\top PB$ and $K^\top SK = A^\top PBS^{-1}B^\top PA$.

Then, we have that

$$\begin{aligned}(A+BK)^\top P(A+BK) - P &= (A^\top PA - P) + A^\top PBK + K^\top B^\top PA + K^\top B^\top PBK \\ &= -Q + A^\top PBS^{-1}B^\top PA - A^\top PBS^{-1}B^\top PA \\ &\quad - A^\top PBS^{-1}B^\top PA + A^\top PBS^{-1}B^\top PA - K^\top RK \\ &= -Q - K^\top RK. \end{aligned} \tag{2.68}$$

Therefore, $\|(A+BK)(x-x_r)\|_P^2 - \|x-x_r\|_P^2 = -(x-x_r)^\top (Q + K^\top RK)(x-x_r)$.

Lemma 2.5

For any $\sigma > 1$, there exists $\delta > 0$, such that

$$V(f(x,\kappa_f(x))) - V(x) \leq -\frac{1}{\sigma}\ell(x,\kappa_f(x)), \quad \forall\, x : \|x - x_r\| \leq \delta. \tag{2.69}$$

Proof

Let us start by computing the closed-loop successor state of system (2.56) under the terminal law κ_f (2.64), i.e.,

$$f(x,\kappa_f(x)) = (A+BK)(x-x_r) + \Delta(x,\kappa_f(x)) + x_r. \tag{2.70}$$

Evaluating Eq. (2.70) through function V (2.65) yields

$$\begin{aligned} V(f(x,\kappa_f(x))) &= \|(A+BK)(x-x_r) + \Delta(x,\kappa_f(x))\|_P^2 \\ &= \|(A+BK)(x-x_r)\|_P^2 \\ &\quad + \underbrace{2(x-x_r)^\top (A+BK)^\top P\Delta(x,\kappa_f(x)) + \|\Delta(x,\kappa_f(x))\|_P^2}_{r_V(x)} \\ &= \|(A+BK)(x-x_r)\|_P^2 + r_V(x). \end{aligned} \tag{2.71}$$

Also, by Lemma 2.4 and Eq. (2.71), we have that

$$V(f(x,\kappa_f(x))) - V(x) = -\|x-x_r\|_{\overline{Q}}^2 + r_V(x), \tag{2.72}$$

where $\overline{Q} = Q + K^\top RK \succ 0$.

Now, recall the bounds on Δ and r in Eqs. (2.57) and (2.60), respectively; letting $0 < \delta_0 \leq \min(\delta_f, \delta_\ell)$, both bounds hold for all $(x,u) : \|(x-x_r, u-u_r)\| \leq \delta_0$.

Considering the bound on Δ in Eq. (2.57), by replacing in it the terminal law κ_f and using the properties of induced matrix norms, there exist $\delta_1, c_f' > 0$ such that

$$\begin{aligned}|\Delta(x,\kappa_f(x))| &\leq c_f(\|x-x_r\|+\|K(x-x_r)\|)^2 \leq c_f(\|x-x_r\|+\|K\|\|x-x_r\|)^2 \\ &= \underbrace{c_f(1+\|K\|)^2}_{c_f'}\|x-x_r\|^2, \quad \forall\, x : \|x-x_r\| \leq \delta_1. \end{aligned} \tag{2.73}$$

Specifically, δ_1 is related to δ_0 as follows:

$$\begin{aligned}\|(x-x_r,\kappa_f(x)-u_r)\| &= \|(x-x_r,K(x-x_r))\| = \sqrt{\|x-x_r\|^2+\|K(x-x_r)\|^2} \\ &\leq \sqrt{\|x-x_r\|^2+\|K\|^2\|(x-x_r)\|^2} \\ &= \sqrt{1+\|K\|^2}\|x-x_r\| \leq \delta_0 \Rightarrow \delta_1 \leq \frac{\delta_0}{\sqrt{1+\|K\|^2}}. \end{aligned} \tag{2.74}$$

Similarly, for the bound on r in Eq. (2.57), there exists $c_\ell' > 0$ such that

$$|r(x,\kappa_f(x))| \leq \underbrace{c_\ell(1-\|K\|)^3}_{c_\ell'}\|x-x_r\|^3, \quad \forall\, x : \|x-x_r\| \leq \delta_1. \tag{2.75}$$

By analyzing the term r_V in Eq. (2.71), we have that

$$\begin{aligned}|2(x-x_r)^\top(A+BK)^\top P\Delta(x,\kappa_f(x))| &\leq 2\|P\|\|(A+BK)(x-x_r)\|\|\Delta(x,\kappa_f(x))\| \\ &\leq 2\|P\|\|(A+BK)\|\|\Delta(x,\kappa_f(x))\|\|x-x_r\| \\ &\leq 2c_f'\|P\|\|(A+BK)\|\|x-x_r\|^3, \end{aligned}$$

$$\|\Delta(x,\kappa_f(x))\|_P^2 \leq \|P\|\|\Delta(x,\kappa_f(x))\|^2 \leq c_f'^2\|P\|\|x-x_r\|^4.$$

Then, there exists $c_V > 0$ such that

$$\begin{aligned}|r_V(x)| &\leq 2c_f'\|P\|\|(A+BK)\|\|x-x_r\|^3 + c_f'^2\|P\|\|x-x_r\|^4 \\ &\leq c_V\|x-x_r\|^3, \quad \forall\, x : \|x-x_r\| \leq \delta_1. \end{aligned} \tag{2.76}$$

Now, consider the stage cost ℓ (2.59) and replace in it the terminal law κ_f, obtaining

$$\begin{aligned}\ell(x,\kappa_f(x)) &= \|x-x_r\|_Q^2 + \|K(x-x_r)\|_R^2 + r(x,\kappa_f(x)) \\ &= (x-x_r)^\top(Q+K^\top RK)(x-x_r) + r(x,\kappa_f(x)) \\ &= \|x-x_r\|_{\overline{Q}}^2 + r(x,\kappa_f(x)). \end{aligned} \tag{2.77}$$

Combining Eqs. (2.72), (2.77), and introducing the constant $\sigma > 1$, we obtain

$$V(f(x,\kappa_f(x))) - V(x) + \frac{1}{\sigma}\ell(x,\kappa_f(x)) = -\left(1-\frac{1}{\sigma}\right)\|x-x_r\|_{\overline{Q}}^2 + r_V(x) + \frac{1}{\sigma}r(x,\kappa_f(x)). \tag{2.78}$$

By Eqs. (2.75) and (2.76), there exists $c > 0$ such that

$$r_V(x) + \frac{1}{\sigma}r(x,\kappa_f(x)) \leq |r_V(x)| + \frac{1}{\sigma}|r(x,\kappa_f(x))|$$

$$\leq \underbrace{\left(c_V + \frac{c'_\ell}{\sigma}\right)}_{c} \|x - x_r\|^3, \quad \forall\, x : \|x - x_r\| \leq \delta_1. \tag{2.79}$$

Since $1 - \frac{1}{\sigma} > 0$ for all $\sigma > 1$, the following bound holds globally:

$$-\left(1 - \frac{1}{\sigma}\right) \|x - x_r\|_{\overline{Q}}^2 \leq -\left(1 - \frac{1}{\sigma}\right) \lambda_{\min}(\overline{Q}) \|x - x_r\|^2, \tag{2.80}$$

where $\lambda_{\min}(\overline{Q}) > 0$ is the smallest eigenvalue of $\overline{Q} \succ 0$.

Then, by Eq. (2.79), for any $\beta \in (0, 1)$ and $\sigma > 1$, there exists $0 < \delta \leq \delta_1$ such that

$$r_V(x) + \frac{1}{\sigma} r(x, \kappa_f(x)) \leq \beta \left(1 - \frac{1}{\sigma}\right) \lambda_{\min}(\overline{Q}) \|x - x_r\|^2, \quad \forall\, x : \|x - x_r\| \leq \delta. \tag{2.81}$$

In the end, combining Eqs. (2.78), (2.80), and (2.81), we obtain

$$\begin{aligned} V(f(x, \kappa_f(x))) - V(x) + \frac{1}{\sigma} \ell(x, \kappa_f(x)) \leq & \underbrace{-(1 - \beta)\left(1 - \frac{1}{\sigma}\right) \lambda_{\min}(\overline{Q})}_{>0} \|x - x_r\|^2 \\ & \leq 0, \quad \forall\, x : \|x - x_r\| \leq \delta \end{aligned}$$

$$\Rightarrow \quad V(f(x, \kappa_f(x))) - V(x) \leq -\frac{1}{\sigma} \ell(x, \kappa_f(x)), \quad \forall\, x : \|x - x_r\| \leq \delta. \tag{2.82}$$

Remark 2.9

The value of δ in Lemma 2.5 satisfies $0 < \delta \leq \frac{\min(\delta_f, \delta_\ell)}{\sqrt{1 + \|K\|^2}}$.

Additionally, it depends on the chosen value of $\sigma > 1$, which represents a degree of freedom. This can be noticed in Eq. (2.81): the term $\beta(1 - \frac{1}{\sigma}) \in (0, 1)$ decreases (i.e., tends to 0) as σ decreases (i.e., $\sigma \to 1$); therefore, δ will be smaller as σ decreases.

Lemma 2.6 (Construction of the quadratic terminal ingredients)

Consider the following quadratic terminal ingredients for the NMPC problem (2.8):

$$V_f(x) = \sigma V(x) = \sigma \|x - x_r\|_P^2, \quad \sigma > 1, \tag{2.83a}$$

$$\mathcal{X}_f = \{x \in \mathbb{R}^{n_x} : V(x) \leq \alpha\}, \quad \alpha > 0. \tag{2.83b}$$

Then, there exists $\alpha > 0$ and $\sigma > 1$ such that the terminal ingredients (2.83) satisfy Assumption 2.11.

Proof

First, we show that there exists $\alpha_0 > 0$ such that, by choosing $0 < \alpha \leq \alpha_0$,

$$\mathcal{X}_f \subseteq B_\delta(x_r) = \{x \in \mathbb{R}^{n_x} : \|x - x_r\| \leq \delta\} \subseteq \mathcal{X}, \tag{2.84}$$

where $\delta > 0$ and $\sigma > 1$ in Eq. (2.83a) are defined in Lemma 2.5.

By Eq. (2.61a), there exists $\delta_x > 0$ such that $B_{\delta_x}(x_r) \subseteq \mathcal{X}$. Then, we can select $\delta \leq \delta_x$ so that $B_\delta(x_r) \subseteq B_{\delta_x}(x_r)$.

Then, since $P \succ 0$, there surely exists $\alpha_0 > 0$ such that $\{x \in \mathbb{R}^{n_x} : \|x - x_r\|_P^2 \leq \alpha_0\} \subseteq B_\delta(x_r) \subseteq B_{\delta_x}(x_r) \subseteq \mathcal{X}$.

Thus, under the choice $0 < \alpha \leq \alpha_0$, Eq. (2.84) holds. Importantly, also Eq. (2.69) holds for all $x \in \mathcal{X}_f$.

Then, let us analyze each Assumption 2.11a-2.11d.

(a) By construction, $\mathcal{X}_f = \{x \in \mathbb{R}^{n_x} : (x - x_r)^\top P(x - x_r) \leq \alpha\}$ (2.83b), with $P \succ 0$ and $\alpha > 0$, is compact and contains x_r. Also, by Eq. (2.84), it is a subset of $\mathcal{X}$.

(b) Let us consider the terminal control law $\kappa_f(x) = K(x - x_r) + u_r$ (2.64). First, we have to ensure that, for all $x \in \mathcal{X}_f$, $\kappa_f(x) \in \mathcal{U}$.

By Eq. (2.61b), there exists $\delta_u > 0$ such that $B_{\delta_u}(u_r) \subseteq \mathcal{U}$. Since $P \succ 0$, it holds

$$\|x - x_r\|_P^2 \geq \lambda_{\min}(P)\|x - x_r\|^2 \Rightarrow \|x - x_r\| \leq \sqrt{\frac{\alpha}{\lambda_{\min}(P)}}, \quad \forall\, x \in \mathcal{X}_f. \tag{2.85}$$

Hence,

$$\|\kappa(x) - u_r\| = \|K(x - x_r)\| \leq \|K\|\|x - x_r\| \leq \|K\|\sqrt{\frac{\alpha}{\lambda_{\min}(P)}}, \quad \forall\, x \in \mathcal{X}_f. \tag{2.86}$$

Then, if we choose $0 < \alpha \leq \min\left(\frac{\delta_u^2 \lambda_{\min}(P)}{\|K\|^2}, \alpha_0\right)$, we have that

$$\|\kappa(x) - u_r\| \leq \delta_u \Rightarrow \kappa(x) \in B_{\delta_u}(u_r) \subseteq \mathcal{U}, \quad \forall\, x \in \mathcal{X}_f. \tag{2.87}$$

Now, by Eq. (2.69), it holds that

$$\forall\, x \in \mathcal{X}_f, \;\; V(f(x, \kappa_f(x))) \leq V(x) \leq \alpha \Rightarrow f(x, \kappa_f(x)) \in \mathcal{X}_f. \tag{2.88}$$

(c) The terminal cost $V_f(x) = \sigma V(x) = \sigma\|x - x_r\|_P^2$, with $\sigma > 1$ and $P \succ 0$, is positive definite in $\mathcal{X}_f$ wrt x_r.

(d) By Eq. (2.69), it holds that

$$V_f(f(x, \kappa_f(x))) - V_f(x) \leq -\ell(x, \kappa_f(x)), \quad \forall\, x \in \mathcal{X}_f. \tag{2.89}$$

Thus, V_f satisfies Eq. (2.29).

Remark 2.10

By Lemma 2.6 and recalling Remark 2.9, the value of δ now satisfies $0 < \delta \leq \min\left(\frac{\delta_f}{\sqrt{1+\|K\|^2}}, \frac{\delta_\ell}{\sqrt{1+\|K\|^2}}, \delta_x\right)$.

The value of $\sigma > 1$ remains a degree of freedom. In particular, the set $\mathcal{X}_f$ becomes smaller as σ decreases (i.e., $\sigma \to 1$).

In this regard, it is interesting to notice that, for $\sigma \to 1$, $\mathcal{X}_f$ collapses to the singleton $\{x_r\}$ and $V_f : \mathcal{X}_f \to \mathbb{R}_{\geq 0}$ degenerates into the identically-null function $V_f(x) = 0$, yielding the terminal ingredients in Remark 2.6.

Finally, with the terminal ingredients constructed in Lemma 2.6, we can state the following theorems:

Theorem 2.3 (Recursive feasibility with ellipsoidal terminal set)

Consider the NMPC problem (2.8). Assume that the linearization at (x_r, u_r) of system (2.1) is stabilizable, i.e., for $A = \frac{\partial f}{\partial x}(x_r, u_r)$ and $B = \frac{\partial f}{\partial u}(x_r, u_r)$, (A, B) is stabilizable. Assume also that the constraint sets $\mathcal{X}$ and $\mathcal{U}$ (2.2) contain a neighborhood of x_r and u_r, respectively.

Let the terminal set $\mathcal{X}_f$ be given by Eq. (2.83b).

Then, the NMPC problem (2.8) is recursively feasible for all initial states $x_0 \in \mathcal{X}_{N_p}$.

Proof

This follows immediately from Theorem 2.1 and Lemma 2.6.

Theorem 2.4 (Closed-loop stability with quadratic terminal ingredients)

Consider the NMPC problem (2.8). Assume that the linearization at (x_r, u_r) of system (2.1) is stabilizable, i.e., for $A = \frac{\partial f}{\partial x}(x_r, u_r)$ and $B = \frac{\partial f}{\partial u}(x_r, u_r)$, (A, B) is stabilizable. Assume also that the constraint sets $\mathcal{X}$ and $\mathcal{U}$ (2.2) contain a neighborhood of x_r and u_r, respectively.

Let the stage cost ℓ satisfy Assumption 2.10 and let the terminal set $\mathcal{X}_f$ and terminal cost V_f be given by Eq. (2.83).

Then, x_r is an asymptotically stable equilibrium of the closed-loop system (2.10), with region of attraction $\mathcal{X}_{N_p}$.

Proof

This follows immediately from Theorem 2.2 and Lemma 2.6.

Remark 2.11

The choice $\mathcal{X}_f = \{x \in \mathbb{R}^{n_x} : \|x - x_r\|_P^2 \leq \alpha\}$ and $V_f(x) = \sigma\|x - x_r\|_P^2$ enlarges the feasible set $\mathcal{X}_{N_p}$ with respect to the case $\mathcal{X}_f = \{x_r\}$ and $V_f(x) = 0$ (see Remarks 2.7 and 2.8).

It also enables stability and recursive feasibility guarantees for systems (2.1) that are stabilizable but not controllable to x_r.

Tracking Control Task

Now, let us focus on the control task of tracking an admissible state trajectory $\boldsymbol{x}_r = (x_{r,k})_{k\geq 0} \in \mathcal{X}^\infty$ of system (2.1), with admissible input sequence $\boldsymbol{u}_r = (u_{r,k})_{k\geq 0} \in \mathcal{U}^\infty$.

Since the reference couple $(x_{r,k}, u_{r,k})$ is time varying, the stage cost ℓ, the terminal cost V_f, and the terminal set $\mathcal{X}_f$ acquire an explicit dependence on the time instant k, i.e., for each $k \geq 0$, we define $\ell_k : \mathcal{X} \times \mathcal{U} \to \mathbb{R}_{\geq 0}$, $\mathcal{X}_{f,k} \subseteq \mathcal{X}$, and $V_{f,k} : \mathcal{X}_{f,k} \to \mathbb{R}_{\geq 0}$.

As a consequence, the NMPC problem (2.8) modifies as follows:

$$\min_{\hat{\boldsymbol{x}}_k, \hat{\boldsymbol{u}}_k} \quad J_k(\hat{\boldsymbol{x}}_k, \hat{\boldsymbol{u}}_k) = \sum_{i=0}^{N_p-1} \ell_{k+i}(\hat{x}_{i|k}, \hat{u}_{i|k}) + V_{f,k+N_p}(\hat{x}_{N_p|k}) \tag{2.90a}$$

$$\text{s.t.} \quad \hat{x}_0 = x_k, \tag{2.90b}$$

$$\hat{x}_{i+1|k} = f(\hat{x}_{i|k}, \hat{u}_{i|k}), \quad i = 0, \dots, N_p - 1, \tag{2.90c}$$

$$\hat{x}_{i|k} \in \mathcal{X}, \quad i = 0, \dots, N_p - 1, \tag{2.90d}$$

$$\hat{u}_{i|k} \in \mathcal{U}, \quad i = 0, \dots, N_p - 1, \tag{2.90e}$$

$$\hat{x}_{N_p|k} \in \mathcal{X}_{f,k+N_p}. \tag{2.90f}$$

The static state-feedback control policy of the NMPC problem (2.90) is now given by π_k,

$$u_k = \hat{u}^*_{0|k} = \pi_k(x_k), \tag{2.91}$$

and the closed-loop system (2.1), (2.91) evolves as

$$x_{k+1} = f(x_k, \pi_k(x_k)), \quad k \geq 0. \tag{2.92}$$

With the modified NMPC problem (2.90), we should also slightly adjust Definition 2.4 on feasible sets:

- For each $k \geq 0$, the set of feasible input sequences $\mathscr{U}_{N,k}(x)$ for horizon N and initial state $x \in \mathcal{X}$ is defined by

$$\begin{aligned} \mathscr{U}_{N,k}(x) = \{\hat{\boldsymbol{u}} \in \mathcal{U}^N : \exists\, \hat{\boldsymbol{x}} \in \mathcal{X}^{N+1},\ & \hat{x}_0 = x, \\ & \hat{x}_{i+1} = f(\hat{x}_i, \hat{u}_i),\ i = 0, \dots, N-1, \\ & \hat{x}_N \in \mathcal{X}_{f,k+N_p}\}. \end{aligned} \tag{2.93}$$

- For each $k \geq 0$, the set of feasible states $\mathcal{X}_{N,k}$ for horizon N is defined by

$$\mathcal{X}_{N,k} = \{x \in \mathcal{X} : \exists\, \hat{\boldsymbol{u}} \in \mathscr{U}_{N,k}(x)\}. \tag{2.94}$$

Assumption 2.14 (Stage cost)

For each $k \geq 0$, the stage cost function $\ell_k : \mathcal{X} \times \mathcal{U} \to \mathbb{R}_{\geq 0}$ is positive definite in $\mathcal{X} \times \mathcal{U}$ wrt $(x_{r,k}, u_{r,k})$, i.e.,

$$\ell_k(x_{r,k}, u_{r,k}) = 0, \tag{2.95a}$$

$$\ell_k(x, u) > 0, \quad \forall\, (x, u) \in (\mathcal{X} \times \mathcal{U}) \setminus \{(x_{r,k}, u_{r,k})\}. \tag{2.95b}$$

Remark 2.12 (Quadratic stage cost)

A "popular" choice for the stage cost ℓ_k meeting Assumption 2.14 is given by the following quadratic function:

$$\ell_k(x, u) = \|x - x_{r,k}\|^2_Q + \|u - u_{r,k}\|^2_R. \tag{2.96}$$

Assumption 2.15 (Terminal ingredients)

For each $k \geq 0$:

a) The terminal set $\mathcal{X}_{f,k}$ in Eq. (2.90f) is compact and contains the reference state $x_{r,k}$, i.e., $x_{r,k} \in \mathcal{X}_{f,k}$.

b) For the terminal set $\mathcal{X}_{f,k}$, it holds that

$$\forall\, x \in \mathcal{X}_{f,k},\ \exists\, u \in \mathcal{U} :\ f(x,u) \in \mathcal{X}_{f,k+1}. \tag{2.97}$$

More specifically, there exists a terminal control law $\kappa_{f,k} : \mathcal{X}_{f,k} \to \mathcal{U}$ such that $u_{r,k} = \kappa_{f,k}(x_{r,k})$ and

$$\forall\, x \in \mathcal{X}_{f,k},\ f(x, \kappa_{f,k}(x)) \in \mathcal{X}_{f,k+1}. \tag{2.98}$$

c) The terminal cost $V_{f,k} : \mathcal{X}_{f,k} \to \mathbb{R}_{\geq 0}$ in Eq. (2.90a) is a positive semidefinite function in $\mathcal{X}_{f,k}$ wrt $x_{r,k}$.

d) The terminal cost $V_{f,k}$ is such that

$$V_{f,k+1}(f(x, \kappa_{f,k}(x))) - V_{f,k}(x) \leq -\ell_k(x, \kappa_{f,k}(x)), \quad \forall\, x \in \mathcal{X}_f. \tag{2.99}$$

Remark 2.13

For each $k \geq 0$, the "simplest" terminal set satisfying Assumptions 2.15a and 2.15b is given by the singleton

$$\mathcal{X}_{f,k} = \{x_{r,k}\}. \tag{2.100}$$

As a consequence, the related terminal cost, by Assumption 2.15c, degenerates into the identically-null function, i.e.,

$$V_{f,k}(x) = 0. \tag{2.101}$$

Recursive Feasibility In the following, we establish recursive feasibility guarantees for the NMPC problem (2.90). All subsequent results are extensions of those derived for the regulation case in Section 2.2.1, p. 18, leveraging the modified Assumptions 2.15a and 2.15b. Consequently, we report only the statements of formal results, as proofs directly follow. For further details, we refer the reader to [64].

Lemma 2.7

If Assumption 2.11b holds, then, for each $N \geq 2$ and $k \geq 0$, we have that:

(i) For each feasible state $x_0 \in \mathcal{X}_{N-1,k}$, with related feasible sequences $\hat{\boldsymbol{u}} \in \mathscr{U}_{N-1,k}(x_0)$ and $\hat{\boldsymbol{x}}$, the prolonged sequence

$$\hat{u}'_i = \hat{u}_i,\ \ i = 0, \ldots, N-2, \quad \hat{u}'_{N-1} = \kappa_{f,k}(\hat{x}_{N-1}) \tag{2.102}$$

is also feasible, i.e., $\hat{\boldsymbol{u}}' \in \mathscr{U}_{N,k}(x_0)$.

(ii) The inclusion $\mathcal{X}_{N-1,k} \subseteq \mathcal{X}_{N,k}$ holds.

Lemma 2.8

For each $k \geq 0$, it holds that

$$\forall\, x \in \mathcal{X}_{N,k},\ f(x, \pi_k(x)) \in \mathcal{X}_{N,k+1}. \tag{2.103}$$

Theorem 2.5 (Recursive feasibility)

Consider the NMPC problem (2.90) and let Assumptions 2.15a and 2.15b hold.

Then, the NMPC problem (2.90) is recursively feasible for all initial states $x_0 \in \mathcal{X}_{N_p,0}$.

Corollary 2.3 (Recursive feasibility with singleton terminal set)

Under the statement of Theorem 2.5, for each $k \geq 0$, let the terminal set be $\mathcal{X}_{f,k} = \{x_{r,k}\}$.

Then, the NMPC problem (2.90) is recursively feasible for all initial states $x_0 \in \mathcal{X}_{N_p,0}$.

Closed-Loop Stability After having ensured recursive feasibility, we now establish closed-loop stability guarantees for the NMPC problem (2.90). As for recursive feasibility, all subsequent results are extensions of those derived for the regulation case in Section 2.2.1, p. 19, leveraging the modified Assumption 2.15. Consequently, we report only the statements of formal results, as proofs directly follow. For further details, we refer the reader to [64].

Lemma 2.9

Consider the NMPC problem (2.90), with stage cost ℓ_k satisfying Assumption 2.14 and terminal cost $V_{f,k}$ satisfying Assumption 2.15c. Let π_k be the related control policy as in Eq. (2.91). Then, for each $k \geq 0$, it holds that

$$u_{r,k} = \pi_k(x_{r,k}). \tag{2.104}$$

Additionally, the state sequence $\boldsymbol{x}_r = (x_{r,k})_{k\geq 0}$ is a trajectory of the closed-loop system (2.92).

Assumption 2.16

There exist $\delta > 0$ and a function $\alpha_f \in \mathcal{K}_\infty$ such that, for each $k \geq 0$:

a) The terminal set $\mathcal{X}_{f,k}$ contains the ball $\mathcal{B}_\delta(x_{r,k})$.

b) For all $x \in \mathcal{B}_\delta(x_{r,k})$, it holds that

$$V_{f,k}(x) \leq \alpha_f(\|x - x_{r,k}\|). \tag{2.105}$$

Under Assumption 2.16, there exists a function $\alpha \in \mathcal{K}_\infty$ such that, for each $k \geq 0$,

$$J_k^*(x) \leq \alpha(\|x - x_{r,k}\|), \quad \forall\, x \in \mathcal{X}_{N_p,0}, \tag{2.106}$$

which is the time-varying generalization of Eq. (2.39).

Theorem 2.6 (Closed-loop stability)

Consider the NMPC problem (2.90). Let the stage cost ℓ_k, the terminal set $\mathcal{X}_{f,k}$, and the terminal cost $V_{f,k}$ satisfy Assumptions 2.14-2.16.

Then, $\boldsymbol{x}_r$ is an asymptotically stable trajectory of the closed-loop system (2.92), with region of attraction $\mathcal{X}_{N_p,0}$.

Assumption 2.17

There exist $\delta > 0$ and a function $\alpha_\ell \in \mathcal{K}_\infty$ such that, for each $k \geq 0$:

a) The state constraint set $\mathcal{X}$ contains the ball $\mathcal{B}_\delta(x_{r,k})$.

b) For all $x \in \mathcal{B}_\delta(x_{r,k})$, there exists an admissible input $u \in \mathcal{U}$ such that

$$f(x,u) = x_{r,k+1}, \quad \ell_k(x,u) \leq \alpha_\ell(\|x - x_{r,k}\|). \tag{2.107}$$

Under Assumption 2.17, Eq. (2.106) is verified.

Corollary 2.4 (Closed-loop stability with point terminal ingredients)

Under the statement of Theorem 2.6, for each $k \geq 0$, let the terminal ingredients be

$$\mathcal{X}_{f,k} = \{x_{r,k}\}, \quad V_{f,k}(x) = 0. \tag{2.108}$$

Let Assumption 2.17 hold.

Then, $\boldsymbol{x}_r$ is an asymptotically stable trajectory of the closed-loop system (2.92), with region of attraction $\mathcal{X}_{N_p,0}$.

Constructing the Terminal Ingredients With respect to the regulation case, constructing the terminal ingredients $\mathcal{X}_{f,k}$ and $V_{f,k}$ for tracking is more complicated because, in the time-varying case, the linear-quadratic approach does not lead to an algebraic Riccati equation that can be easily solved [64].

2.2.2. Nonlinear MPC Without Stabilizing Terminal Ingredients

As thoroughly described in the previous sections, the inclusion of terminal ingredients in the NMPC problem (2.8), namely the terminal cost function V_f and the terminal constraint set $\mathcal{X}_f$, allows to guarantee asymptotic stability and recursive feasibility over the set $\mathcal{X}_{N_p}$.

However, to obtain a stabilizing NMPC control action over a large set $\mathcal{X}_{N_p}$, a large prediction horizon N_p is typically required. Moreover, the straightforward choice of employing point terminal ingredients $\mathcal{X}_f = \{x_r\}$ and $V_f(x) = 0$ may be too restrictive, yielding a small set $\mathcal{X}_{N_p}$, and, thus, requiring the design of a larger terminal set and non-null terminal cost. In such a scenario, the terminal ingredients must satisfy Assumption 2.10 (or 2.14), which poses a considerable additional difficulty in the NMPC design, notably (but not exclusively) for tracking control tasks (i.e., time-varying references).

In contrast to this, practical experience shows us that, typically, NMPC without terminal ingredients can asymptotically stabilize the closed-loop system, over the whole state constraint set $\mathcal{X}$, if a sufficiently large prediction horizon N_p is chosen. In some

cases, the NMPC problem (2.8) without terminal ingredients can stabilize the system on $\mathcal{X}$ already for a very small N_p, while, instead, including terminal ingredients requires a larger N_p to obtain a smaller region of attraction $\mathcal{X}_{N_p} \subseteq \mathcal{X}$. In this scenario, it is crucial to formally characterize the system properties that permit such stabilization and to determine quantitatively when a prediction horizon is "sufficiently large".

Therefore, in the following, we analyze the conditions under which the NMPC problem (2.8) without terminal ingredients is proved to be recursively feasible and to asymptotically stabilize the closed-loop system (2.10).

Regulation Control Task

Let the stage cost ℓ satisfy Assumption 2.10 and define $\ell^*(x) = \inf_{u \in \mathcal{U}} \ell(x, u)$.

Assumption 2.18

The state constraint set $\mathcal{X}$ is control invariant for system (2.1), i.e.,

$$\forall\, x \in \mathcal{X},\ \exists\, u \in \mathcal{U} :\ f(x, u) \in \mathcal{X}. \tag{2.109}$$

Assumption 2.19 (Asymptotic controllability wrt ℓ)

The system (2.1) is *asymptotically controllable with respect to ℓ with rate β*, i.e., for all $x \in \mathcal{X}$, there exists an admissible input sequence $\boldsymbol{u} = (u_i)_{i \geq 0} \in \mathcal{U}^\infty$, with related state trajectory $\boldsymbol{x} = (x_i)_{i \geq 0}$ having $x_0 = x$, such that

$$\ell(x_i, u_i) \leq \beta(\ell^*(x), i), \quad \forall\, i \geq 0. \tag{2.110}$$

The rate $\beta : \mathbb{R}_{\geq 0} \times \mathbb{Z}_{\geq 0} \to \mathbb{R}_{\geq 0}$ is such that:

1) $\beta(r, \cdot)$ is continuous, strictly increasing, and $\beta(r, \cdot) = 0$;
2) $\beta(\cdot, i)$ is continuous, strictly decreasing, and $\beta(\cdot, i) \to 0$ as $i \to +\infty$.

Relevant cases of β are:

- $\beta(r, i) = c\sigma^i r$, with $c > 0$ and $\sigma \in (0, 1)$ (*exponential controllability*);
- $\beta(r, i) = c_i r$, with $c_i \geq 0$ and $c_i = 0$ for $i \geq i_0$ (*finite-time controllability*).

Theorem 2.7 ([64, Chapter 6])

Consider the NMPC problem (2.8) with prediction horizon N_p and without terminal ingredients, i.e.,

$$\mathcal{X}_f = \mathcal{X}, \quad V_f(x) = 0. \tag{2.111}$$

Let the stage cost ℓ satisfy Assumption 2.10 and let Assumptions 2.18 and 2.19 hold.

Then, there exists $\overline{N}_p \geq 1$ such that, for any prediction horizon $N_p \geq \overline{N}_p$, the NMPC problem (2.8) is recursively feasible for all initial states $x_0 \in \mathcal{X}$, and x_r is an asymptotically stable equilibrium of the closed-loop system (2.10), with region of attraction $\mathcal{X}$.

Theorem 2.7 is mainly aimed at finding a sufficiently large prediction horizon $\overline{N}_p$ for which closed-loop stability is ensured, without terminal ingredients. Consequently, we do not place emphasis on recursive feasibility; we introduce Assumption 2.18, whose role is simply to guarantee recursive feasibility over $\mathcal{X}$. Such an assumption, however, is quite strong and is not satisfied by arbitrary constraint sets. Nonetheless, in some cases, one can find a subset of $\mathcal{X}$ that is control invariant for system (2.1), whose size increases with the prediction horizon N_p [64].

To provide some insight, we briefly outline the intuition behind the proof of Theorem 2.7. For more details, we refer the reader to [65, 64].

By Assumption 2.19, it equivalently holds that

$$J_N^*(x) \leq B_N(\ell^*(x)), \quad B_N(r) = \sum_{i=0}^{N-1} \beta(r,i), \quad \forall\, x \in \mathcal{X},\ \forall\, N \geq 1. \tag{2.112}$$

In relevant cases, $B_j(r) = \gamma_j r$, i.e., it is linear in r and $\gamma_j \geq 1$ for all $j \geq 1$. For example, in the case of exponential controllability,

$$\beta(r,i) = c\sigma^i r \ \ \Rightarrow \ \ B_j(r) = c\frac{1-\sigma^j}{1-\sigma} r,\ \ \gamma_j = \frac{1-\sigma^j}{1-\sigma}. \tag{2.113}$$

Through Eq. (2.112), it can be shown that there exists $\alpha \leq 1$ for which the following Lyapunov inequality on the optimal cost value J_N^* holds:

$$J_N^*(f(x,\pi(x))) - J_N^*(x) \leq -\alpha \ell(x,\pi(x)), \quad \forall\, x \in \mathcal{X}. \tag{2.114}$$

According to Theorem 2.2, to ensure closed-loop stability, it must hold that $\alpha \in (0,1]$. While the exact value of α is typically hard to compute, it is ensured that such a value falls within the interval $[\underline{\alpha}_\gamma(N), 1]$. The lower bound $\underline{\alpha}$ depends on the prediction horizon N and the terms $(\gamma_j)_{j\geq 1}$ of the function B_j, as follows:

$$\underline{\alpha}_\gamma(N) = 1 - \frac{(\gamma_N - 1)\prod_{j=2}^{N}(\gamma_j - 1)}{\prod_{j=2}^{N}\gamma_j - \prod_{j=2}^{N}(\gamma_j - 1)}. \tag{2.115}$$

In order to ensure that $\alpha > 0$, it is sufficient to find a sufficiently large $\overline{N} \geq 2$ for which $\underline{\alpha}_\gamma(\overline{N}) > 0$. Such a value $\overline{N}$ is ensured to exist by the fact that $\underline{\alpha}_\gamma(N) \to 1$ as $N \to +\infty$.

In some cases, the expression (2.115) can be inverted to find $\overline{N}$. For example, assume that $\gamma_j = \gamma$ for all j. Then, Eq. (2.115) becomes

$$\underline{\alpha}_\gamma(N) = 1 - \frac{(\gamma-1)^N}{\gamma^{N-1} - (\gamma-1)^{N-1}}. \tag{2.116}$$

By imposing $\underline{\alpha}_\gamma(\overline{N}) > 0$, we obtain

$$\overline{N} \geq \left\lceil 2 + \frac{\ln(\gamma-1)}{\ln\gamma - \ln(\gamma-1)} \right\rceil. \tag{2.117}$$

Tracking Control Task

All results developed in the previous section remain valid for the tracking control task.

For each $k \geq 0$, let the stage cost ℓ_k satisfy Assumption 2.14 and define $\ell_k^*(x) = \inf_{u \in \mathcal{U}} \ell_k(x,u)$. Also, let $J_{k,N}^*(x)$ be the optimal cost value of the NMPC problem (2.8) at time k, with prediction horizon $N \geq 1$.

In the time-varying case, the asymptotic controllability assumption is uniform in time, as follows:

Assumption 2.20 (Uniform asymptotic controllability wrt ℓ)

The system (2.1) is *uniformly asymptotically controllable with respect to ℓ with rate β*, i.e., for all $x \in \mathcal{X}$ and for all $k \geq 0$, there exists an admissible input sequence $\boldsymbol{u} = (u_i)_{i \geq 0} \in \mathcal{U}^\infty$, with related state trajectory $\boldsymbol{x} = (x_i)_{i \geq 0}$ having $x_0 = x$, such that

$$\ell_{k+i}(x_i, u_i) \leq \beta(\ell_k^*(x), i), \quad \forall\, i \geq 0. \tag{2.118}$$

Under these modified assumptions, all results in the previous section carry over to the time-dependent setting [64].

2.3. Linear-Quadratic Model Predictive Control

Linear-Quadratic Model Predictive Control is particular case of the more general Nonlinear MPC, in which the system under control and prediction model are linear (or affine), the state and output constraints (including the terminal one) are linear, and the cost function (composed by stage cost and terminal cost) is quadratic. Therefore, the finite-horizon constrained optimal control problem associated with Linear Quadratic MPC is a Quadratic Program (QP). For this reason, Linear-Quadratic MPC is typically called, in short, QP-MPC.

Let us consider a DT affine dynamical system, i.e.,

$$x_{k+1} = Ax_k + Bu_k + b, \tag{2.119}$$

subject to the linear constraints

$$x_k \in \mathcal{X} = \{x \in \mathbb{R}^{n_x} : H_x x \leq h_x\} \subseteq \mathbb{R}^{n_x}, \tag{2.120a}$$

$$u_k \in \mathcal{U} = \{u \in \mathbb{R}^{n_u} : H_u u \leq h_u\} \subseteq \mathbb{R}^{n_u}, \tag{2.120b}$$

at all time instants $k \geq 0$. In Eq. (2.119), $A \in \mathbb{R}^{n_x \times n_x}$, $B \in \mathbb{R}^{n_x \times n_u}$, and $b \in \mathbb{R}^{n_x}$. Note that, in Eq. (2.120), $\mathcal{X}$ and $\mathcal{U}$ are convex polytopes.

For system (2.119), we consider the regulation and tracking control tasks, as described in Section 2.2.

The QP-MPC strategy provides, at each discrete time instant $k \geq 0$, an optimal control input u_k^* to system (2.119), obtained by solving the following finite-horizon constrained QP optimal control problem:

$$\min_{\hat{\boldsymbol{x}}_k, \hat{\boldsymbol{u}}_k} \quad J(\hat{\boldsymbol{x}}_k, \hat{\boldsymbol{u}}_k) = \sum_{i=0}^{N_p-1} \underbrace{\left(\|\hat{x}_{i|k} - x_r\|_Q^2 + \|\hat{u}_{i|k} - u_r\|_R^2 \right)}_{\ell(\hat{x}_{i|k}, \hat{u}_{i|k})} + \underbrace{\|\hat{x}_{N_p|k} - x_r\|_P^2}_{V_f(\hat{x}_{N_p|k})} \tag{2.121a}$$

$$\text{s.t.} \quad \hat{x}_0 = x_k, \tag{2.121b}$$

$$\hat{x}_{i+1|k} = A\hat{x}_{i|k} + B\hat{u}_{i|k} + b, \quad i = 0, \dots, N_p - 1, \tag{2.121c}$$
$$H_x\hat{x}_{i|k} \leq h_x, \quad i = 0, \dots, N_p - 1, \tag{2.121d}$$
$$H_u\hat{u}_{i|k} \leq h_u, \quad i = 0, \dots, N_p - 1, \tag{2.121e}$$
$$H_f\hat{x}_{N_p|k} \leq h_f. \tag{2.121f}$$

In the QP-MPC problem (2.121):

- Eq. (2.121a) reports the quadratic cost function J, encoding the given control task.
  The stage cost $\ell : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \to \mathbb{R}$ is given by the quadratic function $\ell(x, u) = \|x - x_r\|_Q^2 + \|u - u_r\|_R^2$, where $Q \succeq 0$, $R \succ 0$, and (x_r, u_r) is the reference (either for regulation of tracking). The stage cost matches the form in Remarks 2.5 and 2.12 and, thus, satisfies Assumptions 2.10 and 2.14.
  The terminal cost $V_f : \mathbb{R}^{n_x} \to \mathbb{R}$ is given by the quadratic function $V_f(x) = \|x - x_r\|_P^2$, with $P \succeq 0$.
- Eqs. (2.121b) and (2.121c) is the affine prediction model, given by system (2.119).
- Eqs. (2.121d) and (2.121e) imposes the linear state and input constraints in Eq. (2.120).
- Eq. (2.121f) imposes the terminal state constraint, where the terminal set is a convex polytope, given by $\mathcal{X}_f = \{x \in \mathbb{R}^n : H_f x \leq h_f\} \subseteq \mathcal{X}$.

System (2.119) is controlled by QP-MPC under the one-step receding horizon policy. Then, the QP-MPC problem (2.121) can be represented by a static state-feedback control policy π, as follows:

$$u_k = \hat{u}_{0|k}^* = \pi(x_k), \tag{2.122}$$

and the closed-loop system (2.119), (2.122) evolves as

$$x_{k+1} = f(x_k, \pi(x_k)), \quad k \geq 0. \tag{2.123}$$

2.3.1. Linear-Quadratic MPC With Stabilizing Terminal Ingredients

In the following, we establish closed-loop stability guarantees for the QP-MPC problem (2.121), in the case of regulation and tracking control tasks.

As done for NMPC in Section 2.2.1, we start by ensuring the recursive feasibility of the QP-MPC problem (2.121); then, we proceed with the stability analysis. For further details, we refer the reader to [22, 64]

Regulation Control Task

Recalling Section 2.2.1, p. 24, let us assume that the couple (A, B) of system (2.119) is stabilizable. Then, there exists a unique $P \succeq 0$ solving the discrete algebraic Riccati equation (DARE) with A, B, Q, R in Eqs. (2.119) and (2.121a),

$$P = A^\top PA - A^\top PB(R + B^\top PB)^{-1}B^\top PA + Q, \tag{2.124}$$

with LQR feedback matrix given by

$$K = -(R + B^\top PB)^{-1}B^\top PA, \tag{2.125}$$

which renders $A + BK$ Schur stable.

Lemma 2.10

Let $V(x) = \|x - x_r\|_P^2$, where $P \succeq 0$ is the unique solution of the DARE (2.124), and define $\kappa(x) = K(x - x_r) + u_r$, where K is the LQR matrix (2.125). Also, let $f(x, u) = Ax + Bu + b$.

Then, it holds that

$$V(f(x, \kappa(x))) - V(x) = -(x - x_r)^\top (Q + K^\top R K)(x - x_r). \tag{2.126}$$

Proof

The proof unfolds as in Lemma 2.4, only adding that, by the equilibrium condition $x_r = Ax_r + Bu_r + b$, we have $Ax + Bu + b - x_r = A(x - x_r) + B(u - u_r)$, yielding

$$V(f(x, \kappa(x))) - V(x) = \|(A + BK)(x - x_r)\|_P^2 - \|x - x_r\|_P^2. \tag{2.127}$$

In the following Lemma, we characterize and construct the terminal ingredients of the QP-MPC problem (2.121).

Lemma 2.11 (Construction of the terminal ingredients)

Consider the terminal ingredients of the QP-MPC problem (2.121), i.e.,

$$V_f(x) = \|x - x_r\|_P^2, \tag{2.128a}$$
$$\mathcal{X}_f = \{x \in \mathbb{R}^{n_x} : H_f x \leq h_f\}. \tag{2.128b}$$

Assume that the couple (A, B) of system (2.119) is stabilizable. For the terminal cost (2.128a), let $P \succeq 0$ be the unique solution of the DARE (2.124), with LQR matrix K (2.125).

As terminal set (2.128b), consider the largest polytope within $\mathcal{X}$ that is control invariant for system (2.119) under the terminal control law $\kappa_f(x) = K(x - x_r) + u_r$,

$$\forall\, x \in \mathcal{X}_f, \quad x \in \mathcal{X},\ \ \kappa_f(x) \in \mathcal{U},\ \ f(x, \kappa_f(x)) \in \mathcal{X}_f. \tag{2.129}$$

Then, the terminal ingredients (2.128) satisfy Assumption 2.11.

Proof

By construction, the terminal set $\mathcal{X}_f$ satisfies Assumptions 2.11a and 2.11b.

The terminal cost $V_f(x) = \|x - x_r\|_P^2$, being $P \succeq 0$, is positive semidefinite in $\mathcal{X}_f$ wrt x_r. Then, we compute

$$\begin{aligned}\ell(x, \kappa_f(x)) &= \|x - x_r\|_Q^2 + \|\kappa_f(x) - u_r\|_R^2 = \|x - x_r\|_Q^2 + \|K(x - x_r)\|_R^2 \\ &= (x - x_r)^\top (Q + K^\top R K)(x - x_r).\end{aligned} \tag{2.130}$$

By Lemma 2.10, combining Eqs. (2.126) and (2.130) yields

$$V(f(x, \kappa_f(x))) - V(x) \overset{(=)}{\leq} -\ell(x, \kappa_f(x)), \quad \forall\, x \in \mathcal{X}_f. \tag{2.131}$$

Therefore, V_f satisfies Assumptions 2.11c and 2.11d.

Recursive Feasibility In the following, we establish recursive feasibility guarantees for the QP-MPC problem (2.121), leveraging the results derived for NMPC in Section 2.2.1, p. 18, and employing the terminal set defined in Lemma 2.11.

Theorem 2.8 (Recursive feasibility)

Consider the QP-MPC problem (2.121). Let the terminal set $\mathcal{X}_f$ be given by Lemma 2.11 in Eq. (2.128b).

Then, the QP-MPC problem (2.121) is recursively feasible for all initial states $x_0 \in \mathcal{X}_{N_p}$.

Proof

This follows immediately from Theorem 2.1 and Lemma 2.11.

Remark 2.14

Remark 2.6 and Corollary 2.1 do hold also for QP-MPC, being it a particular case of the more general NMPC. Hence, recursive feasibility is also guaranteed by employing the singleton $\mathcal{X}_f = \{x_r\}$.

However, as noted in Remark 2.7, this choice leads to a smaller feasible set $\mathcal{X}_{N_p}$, compared to the one obtained with the terminal set defined in Lemma 2.11. This effect is illustrated in Example 2.1.

Closed-Loop Stability Now, we establish closed-loop stability guarantees for the QP-MPC problem (2.121), leveraging the results derived for NMPC in Section 2.2.1, p. 19, and employing the terminal ingredients defined in Lemma 2.11.

Theorem 2.9 (Closed-loop stability)

Consider the QP-MPC problem (2.121). Let the terminal set $\mathcal{X}_f$ and terminal cost V_f be given by Lemma 2.11 in Eqs. (2.128b) and (2.128a), respectively.

Then, x_r is an asymptotically stable equilibrium of the closed-loop system (2.123), with region of attraction $\mathcal{X}_{N_p}$.

Proof

This follows immediately from Theorem 2.2 and Lemma 2.11.

Remark 2.15

Remark 2.6 and Corollary 2.2 do hold also for QP-MPC, being it a particular case of the more general NMPC. Hence, closed-loop stability is also guaranteed by employing the point terminal ingredients $\mathcal{X}_f = \{x_r\}$ and $V_f(x) = 0$.

However, as noted in Remark 2.8, this choice leads to a smaller region of attraction $\mathcal{X}_{N_p}$, compared to the one obtained with the terminal set and terminal cost defined in Lemma 2.11. This effect is illustrated in Example 2.1.

Constructing the Terminal Set The terminal set $\mathcal{X}_f$ defined in Lemma 2.11 is a

maximal control invariant set and can be constructed in an iterative way [22].

First, note that, by Assumptions 2.11a, 2.11b, and Eq. (2.129), for any $x \in \mathcal{X}_f$, the following three conditions must be satisfied: (i) $x \in \mathcal{X}$, (ii) $u = \kappa_f(x) = K(x - x_r) + u_r \in \mathcal{U}$, (iii) $f(x, \kappa_f(x)) = Ax + B(K(x - x_r) + u_r) + b = (A + BK)(x - x_r) + x_r \in \mathcal{X}_f$.

Conditions (i) and (ii) are satisfied by requiring that

$$\begin{aligned}\mathcal{X}_f \subseteq & \left\{ x \in \mathbb{R}^{n_x} : \begin{array}{r} H_x x \leq h_x \\ H_u(K(x - x_r) + u_r) \leq h_u \end{array} \right\} \\ = & \left\{ x \in \mathbb{R}^{n_x} : \begin{array}{c} H_x x \leq h_x \\ H_u K x \leq h_u + H_u(K x_r - u_r) \end{array} \right\} = \mathcal{X}_f^{(0)}. \end{aligned} \tag{2.132}$$

Then, to ensure condition (iii), we proceed iteratively as follows:

1) Initialize $\mathcal{X}_f^{(0)}$ as in Eq. (2.132).

2) For $i \geq 0$, iteratively shrink $\mathcal{X}_f^{(i)} = \{x \in \mathbb{R}^{n_x} : H_f^{(i)} x \leq h_f^{(i)}\}$ by enforcing the forward invariance constraint $(A + BK)(x - x_r) + x_r \in \mathcal{X}_f^{(i)}$. This translates into

$$\mathcal{X}_f^{(i+1)} = \mathcal{X}_f^{(i)} \cap \mathrm{Pre}(\mathcal{X}_f^{(i)}), \tag{2.133}$$

where

$$\begin{aligned}\mathrm{Pre}(\mathcal{X}_f^{(i)}) &= \{x \in \mathbb{R}^{n_x} : (A + BK)(x - x_r) + x_r \in \mathcal{X}_f^{(i)}\} \\ &= \{x \in \mathbb{R}^{n_x} : H_f^{(i)}(A + BK)x \leq h_f^{(i)} + H_f^{(i)}((A + BK)x_r - x_r)\} \end{aligned} \tag{2.134}$$

is the predecessor of set $\mathcal{X}_f^{(i)}$ under the forward invariance law.

3) Iterate until convergence, i.e., $\mathcal{X}_f^{(i+1)} = \mathcal{X}_f^{(i)}$, or a maximum number of iterations N_{iter} is reached, and set $\mathcal{X}_f = \mathcal{X}_f^{(i)}$.

The above procedure is encoded in the following Algorithm:

Algorithm 2.1 (Construction of the terminal set $\mathcal{X}_f$ (2.128b))

Input: A, B, Q, R, H_x, h_x, H_u, h_u, x_r, u_r, N_{iter}
Output: $\mathcal{X}_f$

Solve the DARE $P = A^\top P A - A^\top P B(R + B^\top P B)^{-1} B^\top P A + Q$ for P
$K \leftarrow -(R + B^\top P B)^{-1} B^\top P A$
$\mathcal{X}_f^{(0)} \leftarrow \left\{ x \in \mathbb{R}^{n_x} : \begin{array}{c} H_x x \leq h_x \\ H_u K x \leq h_u + H_u(K x_r - u_r) \end{array} \right\}$
for $i = 0, \ldots, N_{\text{iter}} - 1$ **do**
    $\mathcal{X}_f^{(i+1)} \leftarrow \left\{ x \in \mathbb{R}^{n_x} : \begin{array}{c} H_f^{(i)} x \leq h_f^{(i)} \\ H_f^{(i)}(A + BK)x \leq h_f^{(i)} + H_f^{(i)}((A + BK)x_r - x_r) \end{array} \right\}$, where
    $\mathcal{X}_f^{(i)} = \{x \in \mathbb{R}^{n_x} : H_f^{(i)} x \leq h_f^{(i)}\}$
    if $\mathcal{X}_f^{(i+1)} = \mathcal{X}_f^{(i)}$ **then break**
return $\mathcal{X}_f \leftarrow \mathcal{X}_f^{(i)}$

Computing the Feasible Set In QP-MPC, since the terminal set $\mathcal{X}_f$ (2.128b) is a polytope and system (2.119) is affine, we are able to compute with relative ease the feasible

set $\mathcal{X}_{N_p}$, which is both the set of recursively feasible states (Theorem 2.8) and the region of attraction of the closed-loop equilibrium x_r (Theorem 2.9).

Computing $\mathcal{X}_{N_p}$ is particularly insightful, as highlighted in Remarks 2.7 and 2.8: it enables a visual assessment of how large the recursively feasible region is compared to the entire admissible set $\mathcal{X}$, and it reveals how this region expands as the prediction horizon N_p increases.

Recalling Definition 2.4 of feasible states set $\mathcal{X}_N$, $0 \leq N \leq N_p$, and Remark 2.4, we can write

$$\mathcal{X}_0 = \mathcal{X}_f, \tag{2.135a}$$
$$\mathcal{X}_1 = \{x \in \mathcal{X} : \exists\, u \in \mathcal{U},\ \ f(x,u) \in \mathcal{X}_0\}. \tag{2.135b}$$

Eq. (2.135) can be recursively extended as follows:

$$\mathcal{X}_{i+1} = \{x \in \mathcal{X} : \exists\, u \in \mathcal{U},\ f(x,u) \in \mathcal{X}_i\}, \quad i = 0, \ldots, N_p - 1. \tag{2.136}$$

Therefore, we can compute $\mathcal{X}_{N_p}$ iteratively, starting from $\mathcal{X}_f$, as follows [22]:

1) For $0 \leq i \leq N_p - 1$, to build $\mathcal{X}_{i+1}$ from $\mathcal{X}_i$, we have to enforce three conditions, i.e.,

$$\begin{aligned}
x \in \mathcal{X} \quad &\Rightarrow \quad H_x x \leq h_x, && \text{(2.137a)}\\
u \in \mathcal{U} \quad &\Rightarrow \quad H_u u \leq h_u, && \text{(2.137b)}\\
f(x,u) = Ax + Bu + b \in \mathcal{X}_i \quad &\Rightarrow \quad H_i(Ax + Bu + b) \leq h_i, \\
&\qquad\ \ H_i A x + H_i B u \leq h_i - H_i b. && \text{(2.137c)}
\end{aligned}$$

To this end, let us first construct the following polytope on $\mathbb{R}^{n_x} \times \mathbb{R}^{n_u}$:

$$\mathcal{Z}_i = \left\{ (x,u) \in \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} : \begin{array}{c} H_x x \leq h_x \\ H_u u \leq h_u \\ H_i A x + H_i B u \leq h_i - H_i b \end{array} \right\}. \tag{2.138}$$

2) Recalling the definition of projection of polytopes, given $\mathcal{Z} \subseteq \mathbb{R}^{n_x} \times \mathbb{R}^{n_u}$,

$$\mathrm{Proj}_x(\mathcal{Z}) = \{x \in \mathbb{R}^{n_x} : \exists\, u \in \mathbb{R}^{n_u},\ (x,u) \in \mathcal{Z}\}. \tag{2.139}$$

Therefore, we precisely have that

$$\mathcal{X}_{i+1} = \mathrm{Proj}_x(\mathcal{Z}_i) = \{x \in \mathcal{X} : \exists\, u \in \mathcal{U},\ f(x,u) \in \mathcal{X}_i\}. \tag{2.140}$$

3) Iterate until convergence, i.e., $\mathcal{X}_{i+1} = \mathcal{X}_i$, or reaching $i = N_p - 1$.

The above procedure is encoded in the following Algorithm:

Algorithm 2.2 (Construction of the feasible set $\mathcal{X}_{N_p}$)

Input: $\mathcal{X}_f$, A, B, b, H_x, h_x, H_u, h_u, N_p
Output: $\mathcal{X}_{N_p}$

$\mathcal{X}_0 \leftarrow \mathcal{X}_f$
for $i = 0, \dots, N_p - 1$ **do**
  $\mathcal{Z}_i \leftarrow \left\{ (x,u) \in \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} : \begin{matrix} H_x x \le h_x \\ H_u u \le h_u \\ H_i A x + H_i B u \le h_i - H_i b \end{matrix} \right\}$, where $\mathcal{X}_i = \{x \in \mathbb{R}^{n_x} : H_i x \le h_i\}$
  $\mathcal{X}_{i+1} \leftarrow \mathrm{Proj}_x(\mathcal{Z}_i)$
  if $\mathcal{X}_{i+1} = \mathcal{X}_i$ **then**
    $\mathcal{X}_{i+2}, \dots, \mathcal{X}_{N_p} \leftarrow \mathcal{X}_i$
    break
return $\mathcal{X}_{N_p}$

Example 2.1 Consider the following linear system:

$$x_{k+1} = \underbrace{\begin{bmatrix} 1.2 & 1 \\ 0 & 1.2 \end{bmatrix}}_{A} x_k + \underbrace{\begin{bmatrix} 0.5 \\ 0.3 \end{bmatrix}}_{B} u_k, \tag{2.141}$$

subject to state and input constraints

$$\mathcal{X} = \left\{ x \in \mathbb{R}^2 : \begin{bmatrix} -5 \\ -2 \end{bmatrix} \le x \le \begin{bmatrix} 5 \\ 2 \end{bmatrix} \right\}, \quad \mathcal{U} = \{u \in \mathbb{R} : -1 \le u \le 1\}. \tag{2.142}$$

We take the equilibrium $(x_r, u_r) = (\mathbf{0}_2, 0)$ of system (2.141) as the reference for regulation. In the QP-MPC problem (2.121), we set $N_p = 10$, $Q = \mathrm{diag}([1,1]^\top)$, and $R = 1$.

By solving the DARE (2.124), we obtain

$$P = \begin{bmatrix} 4.4452 & 1.0238 \\ 1.0238 & 7.7691 \end{bmatrix}, \quad K = \begin{bmatrix} -0.9737 & -1.9056 \end{bmatrix}. \tag{2.143}$$

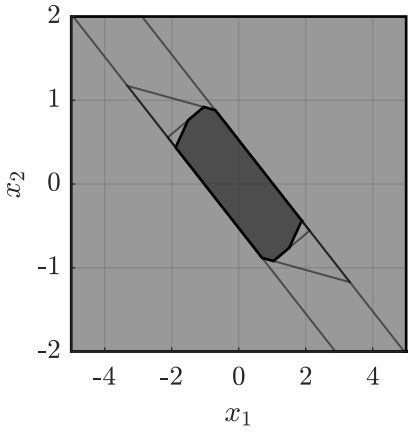
Through the obtained matrix P, we are able to construct the terminal cost $V_f(x) = \|x - x_r\|_P^2 = \|x\|_P^2$, according to Lemma 2.11.

Then, let us construct the terminal set $\mathcal{X}_f$ defined in Lemma 2.11, i.e., the maximal control invariant set of system (2.141) under the terminal control law $\kappa_f(x) = K(x - x_r) + u_r = Kx$. To compute it, we follow Algorithm 2.1. The algorithm converges after 4 iterations to the set

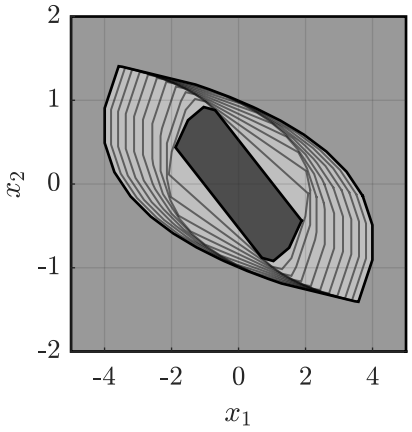
$$\mathcal{X}_f = \left\{ x \in \mathbb{R}^2 : \begin{bmatrix} -0.9737 & -1.9056 \\ 0.9737 & 1.9056 \\ -0.1378 & -1.2433 \\ 0.1378 & 1.2433 \\ 0.2649 & -0.7877 \\ -0.2649 & 0.7877 \\ 0.4190 & -0.4824 \\ -0.4190 & 0.4824 \end{bmatrix} x \le \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} \right\}. \tag{2.144}$$

The construction of the terminal set $\mathcal{X}_f$ is visually reported in Figure 2.1a.

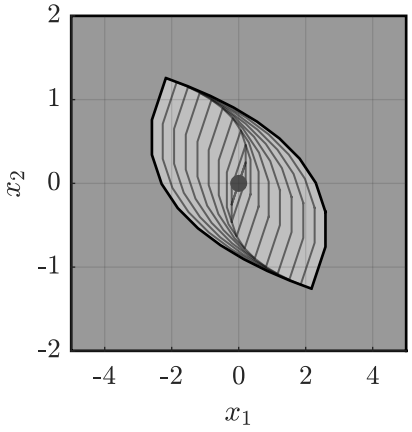
Now, let us compute the feasible set $\mathcal{X}_{N_p}$. To do so, we employ Algorithm 2.2, initializing it with the terminal set $\mathcal{X}_f$ in Eq. (2.144). The algorithm terminates after $10 = N_p$ iterations



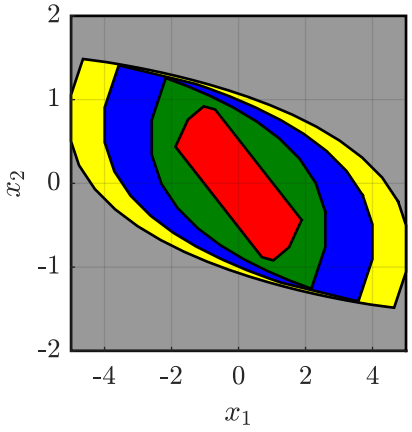
(a) Construction of the terminal set $\mathcal{X}_f$: state constraint set $\mathcal{X}$; terminal set $\mathcal{X}_f$ (2.144); $\mathcal{X}_f^{(i)}$, $i = 0, \ldots, 2$.



(b) Construction of the feasible set $\mathcal{X}_{N_p}$ with $\mathcal{X}_f$ in Eq. (2.144): state constraint set $\mathcal{X}$; terminal set $\mathcal{X}_f = \mathcal{X}_0$; feasible set $\mathcal{X}_{N_p}$; $\mathcal{X}_i$, $i = 1, \ldots, N_p - 1$.



(c) Construction of the feasible set $\mathcal{X}_{N_p}$ with $\mathcal{X}_f = \{x_r\}$: state constraint set $\mathcal{X}$; terminal set $\mathcal{X}_f = \mathcal{X}_0$; feasible set $\mathcal{X}_{N_p}$; $\mathcal{X}_i$, $i = 1, \ldots, N_p - 1$.



(d) State constraint set $\mathcal{X}$; Terminal set $\mathcal{X}_f$ (2.144); Feasible set $\mathcal{X}_{N_p}$ ($\mathcal{X}_f = \{x_r\}$, $N_p = 10$); Feasible set $\mathcal{X}_{N_p}$ ($\mathcal{X}_f$ in Eq. (2.144), $N_p = 10$); Feasible set $\mathcal{X}_{N_p}$ ($\mathcal{X}_f$ in Eq. (2.144), $N_p = 20$).

Figure 2.1. Terminal set $\mathcal{X}_f$ and feasible set $\mathcal{X}_{N_p}$ of Example 2.1.

without converging. This means that, by further increasing the prediction horizon N_p, we would obtain a larger feasible set. We verify so by also computing $\mathcal{X}_{N_p}$ with $N_p = 20$. The construction of the feasible set $\mathcal{X}_{N_p}$ (with $\mathcal{X}_f$ in Eq. (2.144)) up to $N_p = 10$ is reported in Figure 2.1b.

Furthermore, for comparison, we compute the feasible set $\mathcal{X}_{N_p}$ in the case that the terminal set is chosen as the singleton $\mathcal{X}_f = \{x_r\} = \{\mathbf{0}_2\}$, in accordance with Corollaries 2.1, 2.2, and Remark 2.6. Thus, we employ Algorithm 2.2, initializing it with $\mathcal{X}_f = \{\mathbf{0}_2\}$. The algorithm terminates after $10 = N_p$ iterations without converging. The construction of the feasible set $\mathcal{X}_{N_p}$ (with $\mathcal{X}_f = \{x_r\}$) up to $N_p = 10$ is reported in Figure 2.1c.

Figure 2.1d compares the sets in terms of size. We can see that, in accordance with Remarks 2.14 and 2.15, employing the terminal set $\mathcal{X}_f$ defined in Lemma 2.11 yields a larger feasible set $\mathcal{X}_{N_p}$ compared to using $\mathcal{X}_f = \{x_r\}$. Moreover, we see that, as established by Lemma 2.1(ii), as the prediction horizon N_p increases, the feasible set $\mathcal{X}_{N_p}$ becomes larger (up to, possibly, converging to a maximal feasible set, see Remark 2.16).

Remark 2.16

Algorithm 2.2 may converge after $\overline{N}_p < N_p$ iterations. When this happens, the set $\mathcal{X}_{\overline{N}_p}$ is the *maximal feasible set* of the given QP-MPC problem (2.121). This means that $\overline{N}_p$ provides an upper limit on the prediction horizon N_p, over which the feasible set does not further increase.

$\overline{N}_p$ is typically called *determinedness index* [22].

We can observe this considering, e.g., the double integrator

$$x_{k+1} = \underbrace{\begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}}_{A} x_k + \underbrace{\begin{bmatrix} 0 \\ 1 \end{bmatrix}}_{B} u_k, \tag{2.145}$$

subject to state and input constraints

$$\mathcal{X} = \left\{ x \in \mathbb{R}^2 : \begin{bmatrix} -5 \\ -5 \end{bmatrix} \leq x \leq \begin{bmatrix} 5 \\ 5 \end{bmatrix} \right\}, \quad \mathcal{U} = \{u \in \mathbb{R} : -1 \leq u \leq 1\}, \tag{2.146}$$

and taking $(x_r, u_r) = (\mathbf{0}_2, 0)$, $Q = \mathrm{diag}([10, 1]^\top)$, $R = 1$. By solving the DARE (2.124), we obtain

$$P = \begin{bmatrix} 22.1080 & 12.8841 \\ 12.8841 & 15.6000 \end{bmatrix}, \quad K = \begin{bmatrix} -0.7762 & -1.7159 \end{bmatrix}. \tag{2.147}$$

The terminal set $\mathcal{X}_f$ defined in Lemma 2.11 is computed through Algorithm 2.1, which converges after 2 iterations to the set

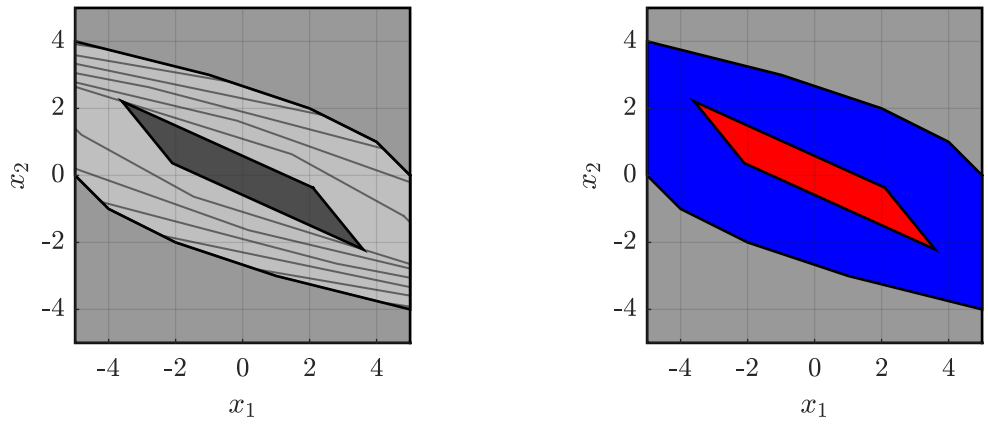
$$\mathcal{X}_f = \left\{ x \in \mathbb{R}^2 : \begin{bmatrix} -0.7762 & -1.7159 \\ 0.7762 & 1.7159 \\ 0.5557 & 0.4523 \\ -0.5557 & -0.4523 \end{bmatrix} x \leq \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} \right\}. \tag{2.148}$$

The feasible set $\mathcal{X}_{N_p}$ is computed through Algorithm 2.2 (initialized with $\mathcal{X}_f$ in

Eq. (2.148)), which converges after 7 iterations to the set

$$\mathcal{X}_7 = \left\{ x \in \mathbb{R}^2 : \begin{bmatrix} -0.4472 & -0.8944 \\ -0.2425 & -0.9701 \\ -0.3162 & -0.9487 \\ 0.4472 & 0.8944 \\ 0.2425 & 0.9701 \\ 0.3162 & 0.9487 \\ 0.7071 & 0.7071 \\ -0.7071 & -0.7071 \\ -1 & 0 \\ 1 & 0 \end{bmatrix} x \leq \begin{bmatrix} 2.6833 \\ 2.6679 \\ 2.5298 \\ 2.6833 \\ 2.6679 \\ 2.5298 \\ 3.5355 \\ 3.5355 \\ 5 \\ 5 \end{bmatrix} \right\}. \tag{2.149}$$

Hence, $\mathcal{X}_7$ is the maximal feasible set and the determinedness index is $\overline{N}_p = 7$.



(a) Construction of the feasible set $\mathcal{X}_{N_p}$ with $\mathcal{X}_f$ in Eq. (2.148): state constraint set $\mathcal{X}$ ▭; terminal set $\mathcal{X}_f = \mathcal{X}_0$ ▭; maximal feasible set $\mathcal{X}_7$ ▭; $\mathcal{X}_i$, $i = 1, \ldots, 6$ ▭.

(b) State constraint set $\mathcal{X}$ ▭; Terminal set $\mathcal{X}_f$ (2.148) ▭; Maximal feasible set $\mathcal{X}_7$ ($\mathcal{X}_f$ in Eq. (2.148)) ▭.

Figure 2.2. Terminal set $\mathcal{X}_f$ and maximal feasible set $\mathcal{X}_{\overline{N}_p}$ for the double integrator case (2.145).

Tracking Control Task

For the tracking control task, we refer the reader to the results developed in Section 2.2.1, p. 30, which equivalently hold for QP-MPC, being it a particular case of NMPC [22, 64].

It is worth noticing that, as for the nonlinear case, constructing the terminal ingredients for tracking is more complicated than for regulation, because, in the time-varying setting, the linear-quadratic approach does not lead to an algebraic Riccati equation that can be easily solved [64].

2.3.2. Linear-Quadratic MPC Without Stabilizing Terminal Ingredients

For what concerns the closed-loop stability and recursive feasibility of QP-MPC when we do not include the terminal ingredients $\mathcal{X}_f$ and V_f, designed in Lemma 2.11, we refer the reader to the results reported in Section 2.2.2, which equivalently hold for QP-MPC, being it a particular case of NMPC [65, 64].

Nonetheless, for affine systems (2.119), we can find a more strong result on asymptotic controllability, whose definition is reported in Assumption 2.19. Without loss of generality, in the following we focus on the regulation control task only.

Proposition 2.1

Consider the affine system (2.119), i.e., $x_{k+1} = Ax_k + Bu_k + b$, and the stage cost

$$\ell(x, u) = \|x - x_r\|_Q^2 + \|u - u_r\|_R^2, \tag{2.150}$$

where $Q \succ 0$, $R \succ 0$, and (x_r, u_r) is an equilibrium point of system (2.119). Assume that the couple (A, B) of system (2.119) is stabilizable.

Then, system (2.119) is exponentially controllable with respect to ℓ (2.150).

Proof

First, given the stage cost $\ell(x, u)$ in Eq. (2.150), we have that $\ell^*(x) = \min_u \ell(x, u) = \|x - x_r\|_Q^2$.

Since system (2.119) is stabilizable, there exists a matrix K such that $A + BK$ is Schur stable. Then, let us consider the stabilizing control policy

$$u_k = u_r + K(x_k - x_r), \tag{2.151}$$

yielding the closed-loop system $x_{k+1} - x_r = (A + BK)(x_k - x_r)$. Under the equilibrium condition for (x_r, u_r), we have that

$$x_k - x_r = (A + BK)^k (x_0 - x_r). \tag{2.152}$$

Now, replacing Eqs. (2.151) and (2.152) into the stage cost ℓ (2.150), it holds that

$$\ell(x_k, u_k) = (x_0 - x_r)^\top (A + BK)^{k\top} (Q + K^\top RK)(A + BK)^k (x_0 - x_r), \tag{2.153}$$

where $Q + K^\top RK = S \succ 0$.

Being $A + BK$ Schur, there exists $m > 0$ and $\rho \in (0, 1)$ such that $\|(A + BK)^k\| \leq m\,\rho^k$, for all $k \geq 0$. Therefore, it holds that

$$\ell(x_k, u_k) \leq m^2 \rho^{2k} \lambda_{\max}(S) \|x_0 - x_r\|^2. \tag{2.154}$$

Recalling the expression of $\ell^*(x)$,

$$\ell^*(x_0) = \|x_0 - x_r\|_Q^2 \geq \lambda_{\min}(Q) \|x_0 - x_r\|^2. \tag{2.155}$$

Combining Eqs. (2.154) and (2.155) yields

$$\ell(x_k, u_k) \leq \frac{m^2 \lambda_{\max}(S)}{\lambda_{\min}(Q)} \rho^{2k} \ell^*(x_0), \quad \forall\, k \geq 0. \tag{2.156}$$

Then, by the definition of asymptotic controllability in Assumption 2.19, we conclude that system (2.119) is exponentially controllable with respect to the stage cost ℓ in Eq. (2.150), with rate

$$\beta(r, k) = c\sigma^k r, \quad c = \frac{m^2 \lambda_{\max}(S)}{\lambda_{\min}(Q)} > 0, \quad \sigma = \rho^2 \in (0, 1). \tag{2.157}$$

Therefore, stabilizable affine systems (2.119) satisfy the asymptotic controllability

condition required by Theorem 2.7, which ensures the existence of a sufficiently large prediction horizon $\overline{N}_p$ such that, for $N_p \geq \overline{N}_p$, the QP-MPC problem (2.121), without terminal ingredients, is recursively feasible and ensures closed-loop stability.

2.3.3. Approximated Nonlinear MPC

In several practical applications, we may find ourselves with a nonlinear system (2.1), subject to linear state and input constraints (2.120).

In such cases, since nonlinear constraints are not present in the problem formulation, we may want to resort to QP-MPC, rather than NMPC, for controlling the system with improved computational performance. Consequently, we aim to *approximate* the NMPC problem and solve it by means of quadratic programming.

To achieve this approximation, we only need to address the nonlinearity of system (2.1), which serves as the prediction model in the MPC problem. Among the possible ways, a conventional, widely-employed approach consists in transforming the nonlinear system (2.1) into an *affine parameter-varying* (APV) system. To derive such an APV system, we compute the first-order Taylor expansion of system (2.1) around a given point $(\overline{x}, \overline{u}) \in \mathcal{X} \times \mathcal{U}$, which acts as the parameter of the APV system. Specifically,

$$\begin{aligned} x_{k+1} &= f(x_k, u_k) \\ &\approx f(\overline{x}, \overline{u}) + \frac{\partial f}{\partial x}(\overline{x}, \overline{u})(x_k - \overline{x}) + \frac{\partial f}{\partial u}(\overline{z}, \overline{u})(u_k - \overline{u}) \\ &= \underbrace{\frac{\partial f}{\partial x}(\overline{x}, \overline{u})}_{A(\overline{x},\overline{u})} x_k + \underbrace{\frac{\partial f}{\partial u}(\overline{x}, \overline{u})}_{B(\overline{x},\overline{u})} u_k + \underbrace{f(\overline{x}, \overline{u}) - \frac{\partial f}{\partial x}(\overline{x}, \overline{u})\overline{x} - \frac{\partial f}{\partial u}(\overline{x}, \overline{u})\overline{u}}_{b(\overline{x},\overline{u})} \\ &= A(\overline{x}, \overline{u})x_k + B(\overline{x}, \overline{u})u_k + b(\overline{x}, \overline{u}). \end{aligned} \tag{2.158}$$

Then, we employ the APV system in Eq. (2.158) as prediction model of the QP-MPC problem (2.121).

Importantly, to counteract the modeling error introduced by the linearization in Eq. (2.158), we set the parameters $(\overline{x}, \overline{u})$ in a time-varying way, i.e., we consider, at each $k \geq 0$, $(\overline{x}_k, \overline{u}_k) = \big((\overline{x}_{k,i}, \overline{u}_{k,i})\big)_{i=0}^{N_p-1}$. Specifically, we set them equal to the shifted optimal trajectory computed by the QP-MPC at the previous time instant $k-1$, i.e.,

$$\begin{aligned} (\overline{x}_k, \overline{u}_k) &= \big((\overline{x}_{k,i}, \overline{u}_{k,i})\big)_{i=0}^{N_p-1} \\ &= \begin{cases} \big((x_0, \mathbf{0}_{n_u}),\ \ldots,\ (x_0, \mathbf{0}_{n_u})\big) & \text{if } k = 0, \\ \big((x_k, \hat{u}^*_{1|k-1}),\ (\hat{x}^*_{2|k-1}, \hat{u}^*_{2|k-1}),\ \ldots,\ (\hat{x}^*_{i+1|k-1}, \hat{u}^*_{i+1|k-1}),\ \ldots, \\ \ (\hat{x}^*_{N_p-1|k-1}, \hat{u}^*_{N_p-1|k-1}),\ (\hat{x}^*_{N_p|k-1}, \mathbf{0}_{n_u})\big) & \text{if } k \geq 1. \end{cases} \end{aligned} \tag{2.159}$$

In the end, we obtain the following QP-MPC problem, approximating the NMPC one, at each $k \geq 0$:

$$\min_{\hat{\boldsymbol{x}}_k, \hat{\boldsymbol{u}}_k} \quad J(\hat{\boldsymbol{x}}_k, \hat{\boldsymbol{u}}_k) = \sum_{i=0}^{N_p-1} \|\hat{x}_{i|k} - x_r\|_Q^2 + \|\hat{u}_{i|k} - u_r\|_R^2 \tag{2.160a}$$

$$\text{s.t.} \quad \hat{x}_0 = x_k, \tag{2.160b}$$

$$\hat{x}_{i+1|k} = A(\overline{x}_{k,i}, \overline{u}_{k,i})\hat{x}_{i|k} + B(\overline{x}_{k,i}, \overline{u}_{k,i})\hat{u}_{i|k} + b(\overline{x}_{k,i}, \overline{u}_{k,i}), \quad i = 0, \dots, N_p - 1, \tag{2.160c}$$

$$H_x \hat{x}_{i|k} \leq h_x, \quad i = 0, \dots, N_p - 1, \tag{2.160d}$$

$$H_u \hat{u}_{i|k} \leq h_u, \quad i = 0, \dots, N_p - 1. \tag{2.160e}$$

Remark 2.17

The method described above, to approximate NMPC as a QP-MPC problem, is a simplified version of the sequential quadratic programming (SQP) approach with real-time iteration (RTI), which is employed as a fast solution method for general NMPC problems [63].

2.4. Explicit Model Predictive Control

Consider the QP-MPC problem introduced in Section 2.3, i.e.,

$$\min_{\hat{\boldsymbol{x}}, \hat{\boldsymbol{u}}} \quad J(\hat{\boldsymbol{x}}, \hat{\boldsymbol{u}}) = \sum_{i=0}^{N_p-1} \left(\|\hat{x}_i - x_r\|_Q^2 + \|\hat{u}_i - u_r\|_R^2 \right) + \|\hat{x}_{N_p} - x_r\|_P^2 \tag{2.161a}$$

$$\text{s.t.} \quad \hat{x}_0 = x_k, \tag{2.161b}$$

$$\hat{x}_{i+1} = A\hat{x}_i + B\hat{u}_i + b, \quad i = 0, \dots, N_p - 1, \tag{2.161c}$$

$$H_x \hat{x}_i \leq h_x, \quad i = 0, \dots, N_p - 1, \tag{2.161d}$$

$$H_u \hat{u}_i \leq h_u, \quad i = 0, \dots, N_p - 1, \tag{2.161e}$$

$$H_f \hat{x}_{N_p} \leq h_f. \tag{2.161f}$$

and its related static state-feedback control policy π, i.e.,

$$u_k = \hat{u}^*_{0|k} = \pi(x_k). \tag{2.162}$$

The QP-MPC problem (2.161) can be rewritten in a simpler and more compact form, comprising only $\hat{\boldsymbol{u}}$ as decision variables. We achieve this by eliminating the prediction model constraints (2.161b), (2.161c) and rewriting the cost function in Eq. (2.161a) so to depend on the variables $\hat{\boldsymbol{u}}$ and x_k only.

In the following, let $\hat{\boldsymbol{x}}$ and $\hat{\boldsymbol{u}}$ denote, interchangeably, the sequences $(\hat{x}_i)_{i=0}^{N_p}$, $(\hat{u}_i)_{i=0}^{N_p-1}$ and the block column vectors $[\hat{x}_i]_{i=0}^{N_p} = [\hat{x}_0^\top, \cdots, \hat{x}_{N_p}^\top]^\top$, $[\hat{u}_i]_{i=0}^{N_p-1} = [\hat{u}_0^\top, \cdots, \hat{u}_{N_p-1}^\top]^\top$.

First, to eliminate the variables $\hat{\boldsymbol{x}}$ from problem (2.161), by recursively applying Eq. (2.161c), we have

$$\hat{x}_i = A^i x_k + \sum_{j=0}^{i-1} A^{i-j-1}(B\hat{u}_i + b), \quad i = 0, \dots, N_p. \tag{2.163}$$

Then, $\hat{\boldsymbol{x}}$ is expressed as a function of $\hat{\boldsymbol{u}}$ and x_k by

$$\hat{\boldsymbol{x}} = \Phi x_k + \Gamma \hat{\boldsymbol{u}} + \gamma, \tag{2.164}$$

where

$$\Phi = \begin{bmatrix} I_{n_x} \\ \sum_{i=1}^{N_p} e_{N_p}^{(i)} \otimes A^i \end{bmatrix}, \tag{2.165a}$$

$$\Gamma = \begin{bmatrix} \mathbf{0}_{n_x \times N_p n_u} \\ I_{N_p} \otimes B + \sum_{i=1}^{N_p-1} E_{N_p}^{(i)} \otimes A^i B \end{bmatrix}, \tag{2.165b}$$

$$\gamma = \begin{bmatrix} \mathbf{0}_{n_x \times n_x} \\ \sum_{i=1}^{N_p} \left(e_{N_p}^{(i)} \otimes \sum_{j=0}^{i-1} A^j \right) \end{bmatrix} b, \tag{2.165c}$$

and $\otimes$ is the Kronecker product operator. Next, by defining

$$\begin{aligned} &\overline{Q} = \begin{bmatrix} I_{N_p} \otimes Q & \mathbf{0} \\ \mathbf{0} & P \end{bmatrix}, \quad \overline{R} = I_{N_p} \otimes R, \\ &\boldsymbol{x}_r = \mathbf{1}_{N_p+1} \otimes x_r, \quad \boldsymbol{u}_r = \mathbf{1}_{N_p} \otimes u_r, \end{aligned} \tag{2.166}$$

we can rewrite the cost function $J(\hat{\boldsymbol{x}}, \hat{\boldsymbol{u}})$ in Eq. (2.161a) in the following compact form:

$$J(\hat{\boldsymbol{x}}, \hat{\boldsymbol{u}}) = \|\hat{\boldsymbol{x}} - \boldsymbol{x}_r\|_{\overline{Q}}^2 + \|\hat{\boldsymbol{u}} - \boldsymbol{u}_r\|_{\overline{R}}^2. \tag{2.167}$$

Replacing Eq. (2.164) into Eq. (2.167) yields

$$J(\hat{\boldsymbol{x}}, \hat{\boldsymbol{u}}) = J_u(\hat{\boldsymbol{u}}) = \frac{1}{2} \hat{\boldsymbol{u}}^\top H \hat{\boldsymbol{u}} + (F x_k + c)^\top \hat{\boldsymbol{u}}, \tag{2.168}$$

where

$$H = \Gamma^\top \overline{Q} \Gamma + \overline{R}, \tag{2.169a}$$

$$F = \Gamma^\top \overline{Q} \Phi, \tag{2.169b}$$

$$c = \Gamma^\top \overline{Q} (\gamma - \boldsymbol{x}_r) - \overline{R} \boldsymbol{u}_r. \tag{2.169c}$$

Finally, we also express the linear constraints (2.161d)-(2.161f) as a function of $\hat{\boldsymbol{u}}$ and x_k only. First, we compact Eqs. (2.161d)-(2.161f) as

$$\overline{H}_x \hat{\boldsymbol{x}} \leq h_x, \quad \overline{H}_u \hat{\boldsymbol{u}} \leq h_u, \tag{2.170}$$

where

$$\overline{H}_x = \begin{bmatrix} I_{N_p} \otimes H_x & \mathbf{0} \\ \mathbf{0} & H_f \end{bmatrix}, \quad \overline{h}_x = \begin{bmatrix} \mathbf{1}_{N_p} \otimes h_x \\ h_f \end{bmatrix}, \tag{2.171a}$$

$$\overline{H}_u = I_{N_p} \otimes H_u, \qquad \overline{h}_u = \mathbf{1}_{N_p} \otimes h_u. \tag{2.171b}$$

Then, replacing Eq. (2.164) into Eq. (2.170) yields the constraints

$$G \hat{\boldsymbol{u}} \leq w + K x_k, \tag{2.172}$$

where

$$G = \begin{bmatrix} \overline{H}_x \Gamma \\ \overline{H}_u \end{bmatrix}, \quad w = \begin{bmatrix} \overline{h}_x - \overline{H}_x \gamma \\ \overline{h}_u \end{bmatrix}, \quad K = -\overline{H}_x \Phi. \tag{2.173}$$

Overall, the original QP-MPC problem (2.161) is equivalent to the following compact optimization problem:

$$\min_{\hat{\boldsymbol{u}}} \quad \frac{1}{2} \hat{\boldsymbol{u}}^\top H \hat{\boldsymbol{u}} + (F x_k + c)^\top \hat{\boldsymbol{u}} \tag{2.174a}$$

$$\text{s.t.} \quad G\hat{\boldsymbol{u}} \leq w + Kx_k. \tag{2.174b}$$

We notice that, thanks to the positive definiteness assumptions on the matrices $Q \succeq 0$ and $R \succ 0$, it holds that $H = H^\top \succ 0$ [14]. Therefore, for any feasible state x_k, the QP-MPC problem (2.161), (2.174) is strongly convex and admits a unique and global optimum $\hat{\boldsymbol{u}}^*$ [23, 19, 35]. Consequently, the policy π (2.162) is well-defined and establishes a bijection between each feasible state $x_k \in \mathcal{X}$ and the optimal control input $u_k = \hat{u}^*_{0|k}$.

Within this setting, we consider the following key idea: rather than using a numerical QP solver online to compute the optimizer $\hat{u}^*_{0|k}$ of the QP-MPC problem (2.161), (2.174) for each given current state x_k, we can try to presolve the QP offline for the entire set of feasible states $x_k \in \mathcal{X}$ to get the QP-MPC control policy $u_k = \hat{u}^*_{0|k} = \pi(x_k)$ (2.162) *explicitly* as a function of x_k.

The main tool to get such an explicit solution is *multiparametric quadratic programming* (mp-QP). The QP-MPC control policy π, obtained by means of mp-QP, goes by the name of *Explicit Model Predictive Control* (Ex-MPC) [14].

For mp-QP problems of the form in Eq. (2.174), the pioneering work [18] proved that the Explicit MPC policy $\pi : \mathcal{P} \to \mathcal{U}$ is continuous and piecewise affine over the set of states $\mathcal{P} = \mathcal{X}_{N_p} \subseteq \mathcal{X}$ for which the problem is recursively feasible, where $\mathcal{P}$ is a polytope (see Section 2.3, p. 41), and that optimal cost value function $J^* : \mathcal{P} \to \mathbb{R}_{\geq 0}$ is continuous, convex, and piecewise quadratic. Moreover, the Ex-MPC policy π is defined over a partition of the set $\mathcal{P}$ into R polytopic regions $(\mathcal{R}_i)_{i=1}^R$.

Definition 2.6 (Polytopic partition)

Let $\mathcal{P} \subseteq \mathbb{R}^n$ be a polytope. The collection of polytopes $(\mathcal{R}_i)_{i=1}^R \subseteq \mathbb{R}^n$ is a *polytopic partition* of $\mathcal{P}$ if

$$\bigcup_{i=1}^{R} \mathcal{R}_i = \mathcal{P}, \tag{2.175a}$$

$$\operatorname{int}(\mathcal{R}_i) \cap \operatorname{int}(\mathcal{R}_j) = \emptyset, \quad \forall\, 1 \leq i, j \leq R, \;\; i \neq j. \tag{2.175b}$$

Theorem 2.10 (Explicit Model Predictive Control [18])

The optimal solution of the QP-MPC problem (2.161), (2.174), at each $k \geq 0$, is given by the following explicit, static state-feedback control policy:

$$u_k = \hat{u}^*_{0|k} = \pi(x_k), \tag{2.176}$$

where:

(i) $\pi : \mathcal{P} \to \mathcal{U}$ is a continuous, piecewise affine (PWA) function, defined over R regions $\mathcal{R}_i$, $i = 1, \ldots, R$, i.e.,

$$u_k = \pi(x_k) = K_i x_k + l_i = \kappa_i(x_k) \quad \text{if} \;\; x_k \in \mathcal{R}_i. \tag{2.177}$$

The domain $\mathcal{P} \subseteq \mathcal{X}$ is the set of recursively feasible states of problem (2.161), (2.174) and is a full-dimensional polytope, i.e.,

$$\mathcal{R}_i = \{x \in \mathbb{R}^{n_x} : H_i x \leq h_i\}. \tag{2.178}$$

Each region $\mathcal{R}_i \subseteq \mathcal{X}$, $i = 1, \dots, R$, is also a full-dimensional polytope.

(ii) The regions $(\mathcal{R}_i)_{i=1}^R$ form a polytopic partition of $\mathcal{P}$.

2.4.1. Generating the Ex-MPC Regions and Affine Functions

The polytopic regions $(\mathcal{R}_i)_{i=1}^R$ and the piecewise-affine structure of the Ex-MPC policy π (2.176) arise by considering the different combinations of active constraints for the QP-MPC problem (2.174) [14].

Specifically, let us consider the Karush-Kuhn-Tucker (KKT) conditions for optimality of problem (2.174), i.e.,

$$H\hat{\boldsymbol{u}} + Fx + c + G^\top \lambda = 0, \tag{2.179a}$$
$$\lambda^\top (G\hat{\boldsymbol{u}} - w - Kx) = 0, \tag{2.179b}$$
$$G\hat{\boldsymbol{u}} \leq w + Kx, \tag{2.179c}$$
$$\lambda \geq 0, \tag{2.179d}$$

where $\lambda \in \mathbb{R}^q$ is the vector of Lagrange multipliers and $G \in \mathbb{R}^{q \times n_u N_p}$. Being problem (2.174) a strictly convex QP, conditions (2.179) are necessary and sufficient for optimality [23].

Let us consider a feasible state $x \in \mathcal{P}$. By solving the related QP-MPC problem (2.174), we obtain the optimal solution $\hat{\boldsymbol{u}}^*$ and we can also retrieve the subset of active and inactive constraints (2.174b) at $\hat{\boldsymbol{u}}^*$. Let $\mathcal{I} = \{1, \dots, q\}$ be a set of indices for all constraints in Eq. (2.174b); let $\mathcal{I}_A \subset \mathcal{I}$ and $\mathcal{I}_I \subset \mathcal{I}$, with $\mathcal{I}_A \cup \mathcal{I}_I = \mathcal{I}$ and $\mathcal{I}_A \cap \mathcal{I}_I = \emptyset$, denote the active and inactive constraints, respectively. Then,

$$G_A \hat{\boldsymbol{u}}^* = w_A + K_A x, \tag{2.180a}$$
$$G_I \hat{\boldsymbol{u}}^* \leq w_I + K_I x, \tag{2.180b}$$

where $G_A = (G)_{\mathcal{I}_A,\cdot}$, $w_A = (w)_{\mathcal{I}_A}$, $K_A = (K)_{\mathcal{I}_A,\cdot}$, $G_I = (G)_{\mathcal{I}_I,\cdot}$, $w_I = (w)_{\mathcal{I}_I}$, and $K_I = (K)_{\mathcal{I}_I,\cdot}$. In view of the complementarity condition (2.179b), we can also split the Lagrange multipliers λ into two subvectors, i.e.,

$$\lambda_A \geq 0, \tag{2.181a}$$
$$\lambda_I = 0, \tag{2.181b}$$

where $\lambda_A = (\lambda)_{\mathcal{I}_A}$ and $\lambda_I = (\lambda)_{\mathcal{I}_I}$.

By Eqs. (2.179a) and (2.181b), we have that

$$\hat{\boldsymbol{u}}^* = -H^{-1}(Fx + c + G_A^\top \lambda_A), \tag{2.182}$$

which, when substituted into Eq. (2.180a), yields

$$\lambda_A = -\underbrace{G_A^\top (G_A H^{-1} G_A^\top)^{-1}}_{=M_A}\Big(w_A + (K_A + G_A H^{-1} F)x + G_A H^{-1} c\Big), \tag{2.183}$$

and, by substitution in Eq. (2.182),

$$\hat{\boldsymbol{u}}^* = H^{-1}\Big(M_A w_A + (M_A(K_A + G_A H^{-1} F) - F)x + (M_A G_A H^{-1} - I)c\Big). \tag{2.184}$$

The solution $\hat{\boldsymbol{u}}^*$ provided by Eq. (2.184) is affine in x. If we multiply both sides of Eq. (2.184) by the matrix $[I_{n_u},\ \mathbf{0}_{n_u \times n_u(N_p-1)}]$, we can extract the optimal control input $\hat{u}_0^*$ from $\hat{\boldsymbol{u}}^*$ (by the receding horizon policy). Then, by affinity of Eq. (2.184), we can express it as

$$\hat{u}_0^* = K_{\mathcal{I}_A} x + l_{\mathcal{I}_A} = \kappa_{\mathcal{I}_A}(x). \tag{2.185}$$

The affine relation (2.185) holds for all $x \in \mathcal{P}$ for which the given combination of active constraints $\mathcal{I}_A$ remains optimal. Such states x are identified by imposing constraints (2.180b) and (2.181a) on $\hat{\boldsymbol{u}}^*$ and λ_A, respectively, that leads to constructing a polytopic set, i.e.,

$$\begin{aligned} \mathcal{R}_{\mathcal{I}_A} &= \{x \in \mathcal{P} : \lambda_A(x) \geq 0,\ G_I \hat{\boldsymbol{u}}^*(x) \leq w_I + K_I x\} \\ &= \{x \in \mathbb{R}^{n_x} : H_{\mathcal{I}_A} x \leq h_{\mathcal{I}_A}\}. \end{aligned} \tag{2.186}$$

Therefore, we conclude that the affine function $\kappa_{\mathcal{I}_A} : \mathcal{R}_{\mathcal{I}_A} \to \mathcal{U}$ is a piece of the overall piecewise-affine Ex-MPC policy π (2.176). The polytope $\mathcal{R}_{\mathcal{I}_A} \subseteq \mathcal{P}$ is the corresponding Ex-MPC region, i.e, an element of the polytopic partition of $\mathcal{P}$.

In order to generate the Ex-MPC regions $\mathcal{R}_i = \{x \in \mathbb{R}^{n_x} : H_i x \leq h_i\}$ and affine functions $\kappa_i(x) = K_i x + l_i$, $i = 1, \ldots, R$, an mp-QP algorithm starts by choosing a feasible state $x_0 \in \mathcal{X}$ and solves the related QP-MPC problem (2.174) to obtain the subset of active and inactive constraints at the optimal solution $\hat{\boldsymbol{u}}^*$.

A good choice for x_0 is the center of the largest ball contained in $\mathcal{X} = \{x \in \mathbb{R}^{n_x} : H_x x \leq h_x\}$, where $H_x \in \mathbb{R}^{r_x \times n_x}$, for which there exists $\hat{\boldsymbol{u}}$ satisfying constraint (2.174b), determined by solving the following linear program (LP) [18]:

$$\max_{x, \hat{\boldsymbol{u}}, \varepsilon} \quad \varepsilon \tag{2.187a}$$

$$\text{s.t.} \quad (H_x)_{i,\cdot}\, x + \varepsilon \|(H_x)_{i,\cdot}\| \leq (h_x)_{i,\cdot}, \quad i = 1, \ldots, r_x, \tag{2.187b}$$

$$G\hat{\boldsymbol{u}} \leq w + Kx. \tag{2.187c}$$

By solving the feasibility problem (2.187), if $\varepsilon \leq 0$, then the QP-MPC problem (2.174) is infeasible for all $x \in \text{int}(\mathcal{X})$; otherwise, we set $x_0 = x$.

This allows to construct, through Eqs. (2.185) and (2.186), the first region $\mathcal{R}_1$ (which is also called *critical region*) and affine function κ_1.

Now, we want to cover the remainder $\mathcal{P} \setminus \mathcal{R}_1$ with other regions corresponding to new combinations of active constraints. A possible approach is described in the following [18].

Assume that $\mathcal{R}_1$ has r_1 facets, meaning that it is defined by r_1 inequalities, i.e., $H_1 \in \mathbb{R}^{r_1 \times n_x}$ and $h_1 \in \mathbb{R}^{r_1}$. We partition the set $\mathcal{X} \setminus \mathcal{R}_1$ by changing, one at a time, the sign of each inequality defining $\mathcal{R}_1$. In this way, we obtain r_1 polytopic sets $(\mathcal{S}_j)_{j=1}^{r_1}$, as follows:

$$\mathcal{S}_j = \mathcal{X} \cap \left\{ x \in \mathbb{R}^{n_x} : \begin{array}{c} (H_1)_{1,\cdot}\, x \leq (h_1)_1 \\ \vdots \\ (H_1)_{j-1,\cdot}\, x \leq (h_1)_{j-1} \\ (H_1)_{j,\cdot}\, x \geq (h_1)_j \end{array} \right\}, \quad j = 1, \ldots, r_1. \tag{2.188}$$

The collection $(\mathcal{R}_1, \mathcal{S}_1, \ldots, \mathcal{S}_{r_1})$ is a partition of $\mathcal{X}$.

For each $j = 1, \dots, r_1$, we determine a new feasible state $x_j \in \mathcal{S}_j$ by solving the feasibility LP (2.187). If no feasible state x_j could be found, it means that $\mathcal{S}_j$ is outside the feasible set $\mathcal{P}$ and, hence, $\mathcal{S}_j$ is dropped.

Instead, if a feasible x_j exists, we solve the related QP-MPC problem (2.174) to obtain a new subset of active and inactive constraints, defining a new Ex-MPC region $\mathcal{R}_j \subseteq \mathcal{S}_j$ and affine function κ_j.

The same procedure is iterated over the newly-generated regions until convergence, i.e., $R-1$ new regions $(\mathcal{R}_i)_{i=2}^{N}$ are generated, partitioning the set $\mathcal{P} \setminus \mathcal{R}_1$, with related affine functions κ_i, $i = 2, \dots, R$.

Implementations of the above algorithm to compute the Ex-MPC policy are available in several software tools, such as the Multi-Parametric Toolbox (MPT) for MATLAB [71].

Example 2.2 Consider the double integrator

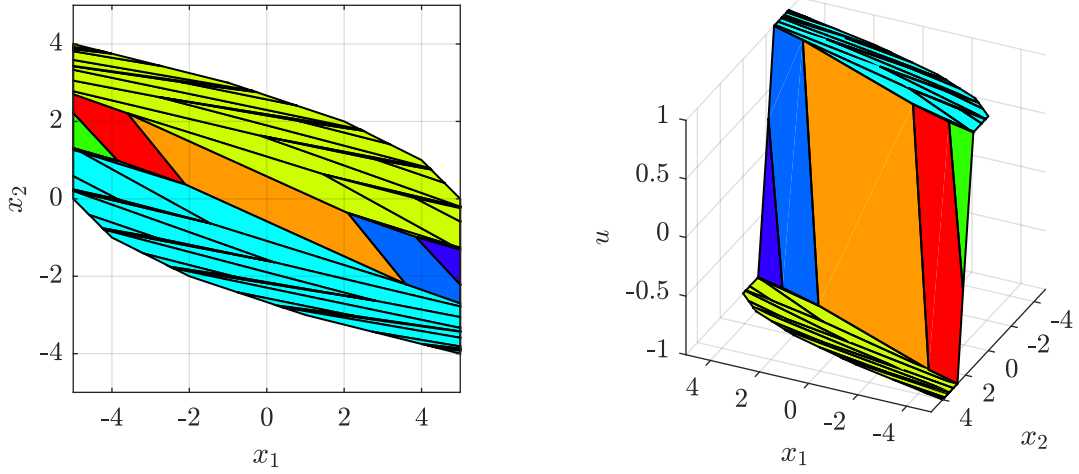
$$x_{k+1} = \underbrace{\begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}}_{A} x_k + \underbrace{\begin{bmatrix} 0 \\ 1 \end{bmatrix}}_{B} u_k, \tag{2.189}$$

subject to state and input constraints

$$\mathcal{X} = \left\{ x \in \mathbb{R}^2 : \begin{bmatrix} -5 \\ -5 \end{bmatrix} \leq x \leq \begin{bmatrix} 5 \\ 5 \end{bmatrix} \right\}, \quad \mathcal{U} = \{ u \in \mathbb{R} : -1 \leq u \leq 1 \}, \tag{2.190}$$

and take $(x_r, u_r) = (\mathbf{0}_2, 0)$, $Q = \mathrm{diag}([10, 1]^\top)$, $R = 1$, $N_p = 10$.

The related Ex-MPC policy π (2.176) has been generated, following the approach described in Section 2.4.1, and is composed by $R = 63$ polytopic regions, with 9 unique affine functions. All regions, partitioning the feasible set $\mathcal{P}$, are reported in Figure 2.3a. The piecewise-affine Ex-MPC policy is reported in Figure 2.3b.



(a) Ex-MPC polytopic regions, partitioning the feasible states set.

(b) Ex-MPC piecewise-affine policy.

Figure 2.3. Ex-MPC policy of Example 2.2. Each color relates to a different affine function.

It is worth noting that, in Figure 2.3a, the feasible set $\mathcal{P}$ is precisely equal to the

maximal feasible set $\mathcal{X}_{\overline{N}_p}$, with $\overline{N}_p = 7$ being the determinedness index, that was computed in Remark 2.16.

Also, it is interesting to observe that the orange region (▭) in Figure 2.3a coincides with the terminal set $\mathcal{X}_f$, which was found in Remark 2.16, and the corresponding affine function is $u_k = \kappa(x_k) = Kx_k$, where K is the LQR matrix (2.125). This happens since $\mathcal{X}_f$ is a control invariant set for system (2.189), as per Assumption 2.11b, with terminal control law $u_k = Kx_k$.

In a neighborhood of the reference $x_r = \mathbf{0}_2$, which is contained in $\mathcal{X}_f$, the QP-MPC problem (2.161) behaves as its unconstrained version, i.e.,

$$\min_{\hat{\boldsymbol{x}},\hat{\boldsymbol{u}}} \quad J(\hat{\boldsymbol{x}},\hat{\boldsymbol{u}}) = \sum_{i=0}^{N_p-1} \|\hat{x}_i - x_r\|_Q^2 + \|\hat{u}_i - u_r\|_R^2 \tag{2.191a}$$

$$\text{s.t.} \quad \hat{x}_0 = x_k, \tag{2.191b}$$

$$\hat{x}_{i+1} = A\hat{x}_i + B\hat{u}_i + b, \quad i = 0, \dots, N_p - 1. \tag{2.191c}$$

Moreover, for a sufficiently large prediction horizon N_p, the unconstrained problem (2.161) behaves like its infinite-horizon version, i.e.,

$$\min_{\hat{\boldsymbol{x}},\hat{\boldsymbol{u}}} \quad J(\hat{\boldsymbol{x}},\hat{\boldsymbol{u}}) = \sum_{i=0}^{\infty} \|\hat{x}_i - x_r\|_Q^2 + \|\hat{u}_i - u_r\|_R^2 \tag{2.192a}$$

$$\text{s.t.} \quad \hat{x}_0 = x_k, \tag{2.192b}$$

$$\hat{x}_{i+1} = A\hat{x}_i + B\hat{u}_i + b, \quad i \geq 0. \tag{2.192c}$$

Problem (2.192) is the infinite-horizon DT LQR, whose optimal control input has the well-known closed-form solution as the static state-feedback $u_k = \hat{u}^*_{0|k} = Kx_k$, where K is the matrix in Eq. (2.125), obtained by solving the discrete algebraic Riccati equation (2.124).

Therefore, by the control invariance of $\mathcal{X}_f$, we can conclude that $\mathcal{X}_f$ itself is the neighborhood of x_r in which the QP-MPC problem (2.161), and thus the Ex-MPC policy (2.176), behaves like the infinite-horizon DT LQR, i.e., with control law $u_k = Kx_k$.

Note that the above behavior holds even if we did not include, in the QP-MPC problem (2.161) for Example 2.2, the terminal ingredients $\mathcal{X}_f = \{x \in \mathbb{R}^{n_x} : H_f x \leq h_f\}$ and $V_f(x) = \|x - x_r\|_P^2$, as designed in Lemma 2.11; it is sufficient that N_p is large enough so that the unconstrained problem (2.191) behaves like its infinite-horizon version (2.192).

2.4.2. Explicit MPC Complexity

Explicit MPC has the obvious advantage, with respect to QP-MPC, of not requiring any online optimization, since the optimal control input is computed by means of the static policy π (2.176). However, when the policy (2.176) is too complex, i.e., contains a very large number of polytopic regions, its static evaluation may become even more costly, in computational terms, than solving online the original QP-MPC problem.

The complexity of Ex-MPC is quantified by the number of regions R that form the explicit solution (2.176), which dictates [14]:

- the amount of memory to store the terms (H_i, h_i, K_i, l_i), representing each i-th region $\mathcal{R}_i$ and affine function κ_i, $i = 1, \dots, R$;

- the execution time required to find the index i of the region $\mathcal{R}_i = \{x \in \mathbb{R}^{n_x} : H_i x \leq h_i\}$ containing the state x (*point location problem*) and evaluating $\kappa_i(x) = K_i x + l_i$.

The point location problem is the major bottleneck of Ex-MPC. While some methods have been proposed to solve it more efficiently than searching linearly through the regions [149], a high number R of regions may cause the point location problem to become more computationally expensive than solving the QP-MPC problem (2.161) online with a conventional QP solver.

The number of regions R is equal to the number of possible combinations of active constraints for the QP-MPC problem (2.174) [61]. Therefore, a rough upper bound on R is given by the number of all such combinations, i.e.,

$$R \leq \sum_{i=0}^{q} \binom{q}{i} = 2^q \tag{2.193}$$

where q is the total number of linear inequality constraints imposed in problem (2.174), i.e., $G \in \mathbb{R}^{q \times n_u N_p}$.

A tighter bound can be found by recalling that, for strictly convex QPs like problem (2.174), at most $n_u N_p$ constraints can be active at the optimum, being $n_u N_p$ the number of decision variables of the problem. Hence,

$$R \leq \sum_{i=0}^{n_u N_p} \binom{q}{i}, \tag{2.194}$$

and, typically, $n_u N_p \ll q$.

In practice, R is much smaller, as most combinations of constraints are never active at optimality (e.g., upper-lower bound constraints cannot be active at the same time, unless they coincide).

In general, the complexity of Ex-MPC has:

- strongest dependence on the number q of linear inequality constraints (2.174b);
- strong dependence on the number N_{var} of free decision variables in problem (2.174) (in our case, $N_{\text{var}} = n_u N_p$);
- weak dependence on the number n_p of parameters in problem (2.174) (in our case, the only parameter is x_k and $n_p = n_x$).

I

Advancements in Real-Time Model Predictive Control

3

Solving Nonlinear MPC Problems in the Koopman Lifted Space: A Comprehensive Analytical Framework

3.1. Introduction

The Koopman operator has gained traction in the realm of system modeling and control [126], due to its ability to encapsulate the dynamics of nonlinear systems into a "lifted" state space with linear dynamics. Original and lifted spaces are mapped through a basis of observable functions, which span an invariant subspace of the Koopman operator [101, 26, 59].

The Koopman operator can be used to tackle nonlinear optimal control problems, including Nonlinear Model Predictive Control (NMPC) [79]. Specifically, the nonlinear prediction model and nonlinear state constraints of NMPC can be lifted into the Koopman space and transformed into their equivalent linear counterparts, yielding nonlinear programs (NLP) to be replaced by quadratic programs (QP), which can be solved with superior computational performance [35].

The Koopman operator unleashes its full potential when the basis of observables can be derived analytically. While analytical approaches have been explored in the literature, they face inherent challenges that make their application quite restricted. Eigenfunction-based methods [59, 27] find closed-form observables only for few classes of systems and, in general, Koopman eigenfunctions cannot be inverted to recover the original system state (which is crucial to enable control) [26]. Methods like Carleman linearization [82] derive observables by expanding in series the system dynamics, limiting to the polynomial space and not effectively exploiting the original system structure. Moreover, analytical methods often struggle to systematically address the case of infinite-dimensional bases of observables [26, 27]. As a result, data-driven techniques have emerged to obtain an approximated finite-dimensional Koopman lifting; yet, they also come with their own inherent shortcomings. Extended Dynamic Mode Decomposition (EDMD) [157] – the leading data-driven Koopman approach – relies on system measurements and a handpicked dictionary of observables [26]. However, to achieve a satisfactory accuracy, a very large

dictionary is typically required [79], and, in general, the resulting lifted system cannot generalize beyond the initial data set. On the other hand, automated techniques to learn the dictionary [98, 5] lack interpretability and cannot incorporate human expertise in selecting potentially useful observables. Notably, data-driven methods directly identify an approximated finite-dimensional lifted system, rather than performing a tailored reduction on the infinite-dimensional one [27].

In this chapter, we address the limitations of the existing methods discussed above, namely, the systematic derivation of suitable Koopman observables and the handling of infinite-dimensional Koopman lifted systems. We formulate a systematic procedure to analytically lift generic nonlinear systems, starting from an initial set of handpicked observables and iteratively generating the whole basis. We also present a general approach to reduce infinite-dimensional Koopman lifted systems to a finite dimension of choice. To determine the lowest dimension achieving the best accuracy for the lifted system, we evaluate its multi-step prediction error for increasing dimensions.

The proposed method is employed to lift a general NMPC problem, including both the nonlinear prediction model and nonlinear state constraints, obtaining an equivalent QP-MPC in the Koopman space – referred to as Koopman NMPC (in short, K-NMPC) [31] – which closely approximates the original NMPC solution. In this context, reducing the dimensionality of the Koopman lifted space directly reduces also the computational complexity of the K-NMPC problem. Some works have applied the Koopman operator to NMPC [54, 140]; yet, they all rely on data-driven methods to lift the prediction model only. By converse, our method features a fully analytical and general framework, lifting also the NMPC constraints, and incorporating dimension reduction.

To validate our K-NMPC strategy, we carry out simulations and experimental validations on two real-world case studies: mobile robot navigation in cluttered environments and autonomous parallel parking. In both scenarios, K-NMPC manages to proficiently attain the control tasks, while always respecting the original NMPC constraints. We demonstrate that K-NMPC closed-loop trajectories closely approximate those given by NMPC, with K-NMPC outperforming NMPC in terms of computation time by over an order of magnitude.

3.1.1. Outline

This chapter is organized as follows. Section 3.2 introduces the Koopman operator theory and derives the Koopman lifted form of nonlinear dynamical systems, considering both the autonomous and the input-dependent case, in continuous and discrete time. Section 3.3 presents our algorithmic procedure to analytically derive the basis of Koopman observables in a systematic way. Section 3.4 describes the general method to reduce the dimensionality of the Koopman lifted space. Section 3.5 details the process of lifting a general NMPC problem into the Koopman space, yielding the quadratic K-NMPC problem. Section 3.6 validates the K-NMPC approach through simulations and experimental validations on real-world case studies.

3.1.2. Related Works

This chapter is based, in part, on the following works:

- L. Calogero, M. Boggio, C. Novara, and A. Rizzo, “A General Analytical Framework

for Fast Solving Nonlinear MPC Problems in the Linear Koopman Space," in *Proceedings of the European Control Conference*, 2025, pp. 743–748.

- L. Calogero, A. Usai, and A. Rizzo, "Solving Nonlinear MPC Problems in the Koopman Lifted Space: The Case Study of Mobile Robot Navigation in Cluttered Environments," in *Proceedings of the "Automatica.it" Conference*, 2025.

3.2. Koopman Operator Theory

The *Koopman operator* offers a powerful framework for the analysis and control of nonlinear dynamical systems. Its key aspect consists in transforming a generic nonlinear system (evolving in the state space $\mathcal{X} \subseteq \mathbb{R}^{n_x}$) into an equivalent linear (or bilinear) one that evolves in a different state space $\mathcal{Z} \subseteq \mathbb{R}^{n_z}$; the latter is named *lifted state space* and is, in general, higher-dimensional (and, typically, infinite-dimensional) compared to the original one (i.e., $n_z \gg n_x$, $n_z \in \mathbb{R} \cup \{+\infty\}$).

The shift between these state spaces is performed by a basis of static functions $z = \phi(x) = [\phi_i(x)]_{i=1}^{n_z} : \mathcal{X} \to \mathcal{Z}$, called *observable functions* (or, more briefly, *observables*), which map each original state $x \in \mathcal{X}$ to its lifted counterpart $z \in \mathcal{Z}$.

Thus, rather than linearizing the original system through local approximations within the same state space $\mathcal{X}$, the Koopman operator constructs a (possibly infinite-dimensional) lifted version of the system, evolving in a different lifted state space $\mathcal{Z}$. In this way, the trajectories $x(t)$ and $z(t)$ of the original and lifted system, respectively, do always globally coincide without approximation, through the map $z(t) = \phi(x(t))$, $\forall\, t \geq 0$.

In this framework, the key challenge resides in analytically finding a suitable basis of observables ϕ. Specifically, if one manages to identify a countable (possibly infinite) basis of observables $\Phi = \{\phi_i(x)\}_{i=1}^{n_z}$ whose span forms an *invariant subspace of the Koopman operator*, then the dynamics of the original nonlinear system is captured exactly by the linear (or bilinear) lifted system.

A second key challenge comes into play when the basis of observables (spanning the Koopman invariant subspace) is countable, yielding an infinite-dimensional lifted system.

In the remainder of this section, we derive the Koopman lifted form of nonlinear dynamical systems, considering both the autonomous and the input-dependent case, in continuous and discrete time. These derivations are analogous to the general results that can be found in, e.g., [101, 59, 74].

3.2.1. Continuous-Time Autonomous Systems

Let us consider a continuous-time (CT) autonomous system[1], i.e.,

$$\dot{x}(t) = f(x(t)), \quad x(0) = x_0, \quad t \in \mathbb{R}_{\geq 0}, \tag{3.1}$$

where $x \in \mathcal{X} \subseteq \mathbb{R}^{n_x}$ is the state vector, $\mathcal{X}$ is the state-space manifold, and $f : \mathcal{X} \to \mathcal{X}$ is the state dynamics function.

[1]Hereafter, for notational convenience, we shall omit the explicit time dependence when clear from context.

If f is Lipschitz continuous on $\mathcal{X}$, the Picard-Lindelöf theorem guarantees the existence of a unique solution $x(t)$ of system (3.1) in the time interval $[0, +\infty)$, starting from x_0 [77],

$$x(t) = x_0 + \int_0^t f(x(\tau))d\tau = \varphi^t(x_0). \tag{3.2}$$

The family of one-parameter maps $\varphi^t : \mathcal{X} \to \mathcal{X}$ represents the *flow* of system (3.1), which generates the system trajectories $x(t)$.

The Koopman operator allows to transform system (3.1), which evolves in the state space $\mathcal{X}$, into an equivalent linear system, evolving in a higher-dimensional state space $\mathcal{Z} \subseteq \mathbb{R}^{n_z}$, $n_z \gg n_x$, called *Koopman lifted space*, i.e.,

$$\dot{z} = Az, \quad z(0) = \phi(x_0). \tag{3.3}$$

System (3.3) is called *Koopman lifted system* and $z \in \mathcal{Z}$ is the *lifted state*.

Systems (3.1) and (3.3) are put in relation through a suitable basis of functions, called observable functions (or, more briefly, *observables*), i.e.,

$$\Phi = \{\phi_i(x)\}_{i=1}^{N_o} \subset \mathscr{F}, \quad \phi(x) = [\phi_i(x)]_{i=1}^{N_o}, \tag{3.4}$$

where Φ denotes the basis of observables $\phi_i : \mathcal{X} \to \mathbb{R}$, $i = 1, \dots, N_o = n_z$, $\phi(x)$ is the basis arranged in a vector, and $\mathscr{F}$ is a suitable vector space of functions.

Definition 3.1 (Koopman operator – CT autonomous systems)

Let $\mathscr{F} \subseteq \mathcal{C}^1$ be a Banach space of continuously differentiable observables $\phi : \mathcal{X} \to \mathbb{R}$. The family of Koopman operators $\mathcal{K}^t : \mathscr{F} \to \mathscr{F}$ associated with the family of maps φ^t (3.2) is defined as follows:

$$\mathcal{K}^t\phi = \phi \circ \varphi^t, \quad \forall\, \phi \in \mathscr{F}. \tag{3.5}$$

From Definition 3.1, we observe that the Koopman operator (3.5) provides the evolution of the observables ϕ along the trajectories $x(t)$ of system (3.1), i.e.,

$$\mathcal{K}^t\phi(x_0) = \phi \circ \varphi^t(x_0) = \phi \circ x(t) = \phi(x(t)), \quad \forall\, \phi \in \mathscr{F}. \tag{3.6}$$

The trajectories $z(t) = \phi(x(t))$, $\forall\, \phi \in \mathscr{F}$, are called *lifted trajectories*.

Then, we can compute the dynamics of the lifted trajectories $z(t)$ (*lifted dynamics*) – as a counterpart of the dynamics of the original trajectories $x(t)$ – by evaluating the time derivative of the lifted trajectories $\dot{z}(t)$, i.e.,

$$\begin{aligned} \dot{z} = \dot{\phi}(x) &= \frac{d}{dt}\phi(x) = \frac{\partial \phi}{\partial x}\frac{dx}{dt} = \frac{\partial \phi}{\partial x} f(x) \\ &= L_f \phi(x), \end{aligned} \tag{3.7}$$

where $L_\star = \star \cdot \partial$ denotes the Lie derivative.

Remark 3.1

The result in Eq. (3.7) can be also obtained by resorting to semigroup theory.

We can notice that the family of maps $\{\varphi^t\}_{t\geq 0}$ in Eq. (3.2) (i.e., the flow of

system (3.1)) is a semigroup, since it exhibits the following properties:

$$\begin{aligned}&\varphi^0(x_0) = x_0,\\ &\varphi^{t+s}(x_0) = \varphi^s \circ \varphi^t(x_0), \quad t, s \geq 0.\end{aligned}$$

Then, the family of Koopman operators $\{\mathcal{K}^t\}_{t\geq 0}$ (3.5) is a semigroup as well, since

$$\begin{aligned}&\mathcal{K}^0\phi = \phi \circ \varphi^0 = \phi,\\ &\mathcal{K}^{t+s}\phi = \phi \circ \varphi^{t+s} = \phi \circ (\varphi^s \circ \varphi^t) = \mathcal{K}^s\mathcal{K}^t\phi, \quad t, s \geq 0.\end{aligned}$$

Moreover, the Koopman operator semigroup is linear, i.e., $\mathcal{K}^t(c_1\phi_1 + c_2\phi_2) = c_1\mathcal{K}^t\phi_1 + c_2\mathcal{K}^t\phi_2$, for all $\phi_1, \phi_2 \in \mathscr{F}$, $c_1, c_2 \in \mathbb{R}$, and $t \geq 0$.

Under the assumption of Lipschitz continuity of f, $\{\varphi^t\}_{t\geq 0}$ is uniformly Lipschitz continuous with respect to t. This implies that the semigroup $\{\mathcal{K}^t\}_{t\geq 0}$ is strongly continuous. This property ensures that the limit

$$\lim_{t\to 0^+} \frac{\mathcal{K}^t\phi - \phi}{t} = \mathcal{L}\phi \tag{3.8}$$

exists $\forall\, \phi \in \mathscr{F}$. The operator $\mathcal{L} : \mathscr{F} \to \mathscr{F}$ is the *infinitesimal generator* of the Koopman operator semigroup. Specifically,

$$\begin{aligned}\lim_{t\to 0^+} \frac{\mathcal{K}^t\phi(x_0) - \phi(x_0)}{t} &= \lim_{t\to 0^+} \frac{\phi \circ \varphi^t(x_0) - \phi(x_0)}{t} = \lim_{t\to 0^+} \frac{\phi(x(t)) - \phi(x_0)}{t}\\ = \frac{d\phi(x)}{dt}(0) &= \frac{\partial\phi}{\partial x}(x_0)\,\frac{dx}{dt}(0) = \frac{\partial\phi}{\partial x}(x_0)\,f(x_0) = [L_f\phi](x_0) = [\mathcal{L}\phi](x_0),\end{aligned} \tag{3.9}$$

yielding $\mathcal{L} = L_f$. By applying $\mathcal{K}^t$ to both sides of Eq. (3.9), and leveraging the semigroup and linearity properties, we obtain

$$\mathcal{K}^t\frac{d\phi(x)}{dt}(0) = \frac{d\phi(x)}{dt} = \dot{\phi}(x) = \mathcal{K}^t[\mathcal{L}\phi](x_0) = \mathcal{L}\phi(x) = L_f\phi(x), \tag{3.10}$$

which coincides with the lifted dynamics in Eq. (3.7).

Now, let us restrict our analysis to a N_o-dimensional linear subspace of $\mathscr{F}$, i.e., $\mathscr{F}_{N_o} \subseteq \mathscr{F}$, spanned by a countable (and possibly infinite) basis Φ of N_o observables, as in Eq. (3.4).

Assumption 3.1

For system (3.1) and the basis of observables Φ (3.4), it holds that

$$L_f\phi(x) \in \operatorname{span}(\Phi), \quad \forall\, \phi \in \Phi. \tag{3.11}$$

Under Assumption 3.1, $\mathscr{F}_{N_o}$, spanned by the basis Φ (3.4), is an invariant subspace of the Koopman operator [59]. This allows to establish a direct relation between the original system (3.1) and the Koopman lifted system (3.3) through the lifted dynamics (3.7), as stated in the following theorem:

Theorem 3.1 (Koopman lifted system – CT autonomous systems)

Let system (3.1) and a basis of observables $\Phi = \{\phi_i(x)\}_{i=1}^{N_o}$ (3.4) be given, satisfying Assumption 3.1. Let $x(t)$ be the solution of system (3.1) and $z(t)$ the solution of the Koopman lifted system (3.3).

Then, the two solutions are equivalent through the mapping defined by ϕ, i.e.,

$$z(t) = \phi(x(t)), \quad x(t) = \phi^{-1}(z(t)), \quad \forall\, t \geq 0. \tag{3.12}$$

Proof

By Assumption 3.1, $\forall\, i = 1, \ldots, N_o$ we can write

$$L_f \phi_i(x) = a_i^\top \phi(x), \quad a_i \in \mathbb{R}^{N_o}. \tag{3.13}$$

Then, recalling the lifted dynamics (3.7), we have that

$$\dot{\phi}_i(x) = a_i^\top \phi(x), \quad i = 1, \ldots, N_o \tag{3.14a}$$

$$\Rightarrow \dot{\phi}(x) = A\phi(x), \tag{3.14b}$$

where $A = [a_1, \ldots, a_{N_o}]^\top$.

By applying the map $z = \phi(x)$ to Eq. (3.14b) and adding the initial condition $z(0) = \phi(x_0)$, we obtain system (3.3).

Then, it holds that $z(t) = \phi(x(t))$, $\forall\, t \geq 0$, being $x(t)$ and $z(t)$ the solutions of systems (3.1) and (3.3), respectively. If ϕ admits an inverse map ϕ^{-1}, then it also holds $x(t) = \phi^{-1}(z(t))$.

Remark 3.2 (Invertible observables map)

A sufficient condition for the existence of the inverse map ϕ^{-1} in Eq. (3.12) is that the states x are included as the first n_x observables, i.e., $(\phi(x))_{\{1,\ldots,n_x\}} = x$. In this way, $x = \phi^{-1}(z) = (z)_{\{1,\ldots,n_x\}}$.

To sum up, we have transformed the CT autonomous system (3.1) into its equivalent linear Koopman lifted system (3.3); these two systems are put in relation through the basis of observables (3.4), as follows:

$$\dot{x} = f(x) \quad \overset{\substack{z = \phi(x) \\ x = \phi^{-1}(z)}}{\Longleftrightarrow} \quad \dot{z} = Az. \tag{3.15}$$

3.2.2. Continuous-Time Systems With Inputs

Let us now consider a CT system with inputs, i.e.,

$$\dot{x}(t) = f(x(t), u(t)), \quad x(0) = x_0, \quad t \geq 0, \tag{3.16}$$

where $x \in \mathcal{X} \subseteq \mathbb{R}^{n_x}$ and $u \in \mathcal{U} \subseteq \mathbb{R}^{n_u}$ are the state and input vectors, respectively, $\mathcal{U}$ is the set of admissible input values, and $f : \mathcal{X} \times \mathcal{U} \to \mathcal{X}$ is the state dynamics function.

If $f(\cdot, u)$ is Lipschitz continuous on $\mathcal{X}$, $f(x, \cdot)$ is continuous on $\mathcal{U}$, and the input signal $\boldsymbol{u} = (u(t))_{t\in[0,+\infty)}$ belongs to the space of piecewise continuous signals $\mathscr{U}$, the Picard-Lindelöf theorem guarantees the existence of a unique solution $x(t)$ of system (3.16) in the time interval $[0, +\infty)$, starting from x_0 [77], i.e.,

$$x(t) = x_0 + \int_0^t f(x(\tau), u(\tau))d\tau = \varphi^t(x_0, \boldsymbol{u}). \tag{3.17}$$

The family of one-parameter maps $\varphi^t : \mathcal{X} \times \mathscr{U} \to \mathcal{X}$ is the flow of system (3.16).

Definition 3.2 (Koopman operator – CT systems with inputs)

Let $\mathscr{F} \subseteq \mathcal{C}^1$ be a Banach space of continuously differentiable observables $\phi : \mathcal{X} \to \mathbb{R}$. The family of Koopman operators $\mathcal{K}^{t,\boldsymbol{u}} : \mathscr{F} \to \mathscr{F}$ associated with the family of maps $\varphi_t(\cdot, \boldsymbol{u})$ (3.17) is defined as follows:

$$\mathcal{K}^{t,\boldsymbol{u}}\phi(\cdot) = \phi \circ \varphi^t(\cdot, \boldsymbol{u}), \quad \forall\, \phi \in \mathscr{F}. \tag{3.18}$$

Remark 3.3

In Eq. (3.18), the family of Koopman operators is parameterized by the continuous time instant t and the input signal $\boldsymbol{u} \in \mathscr{U}$.

From Definition 3.2, the Koopman operator (3.18) provides the evolution of the observables ϕ along the trajectories $x(t)$ of system (3.16), i.e., the lifted trajectories $z(t) = \phi(x(t))$,

$$\mathcal{K}^{t,\boldsymbol{u}}\phi(x_0) = \phi \circ \varphi^t(x_0, \boldsymbol{u}) = \phi \circ x(t) = \phi(x(t)), \quad \forall\, \phi \in \mathscr{F}. \tag{3.19}$$

The lifted dynamics is obtained by computing the time derivative $\dot{z}(t)$, i.e.,

$$\dot{z} = \dot{\phi}(x) = \frac{d}{dt}\phi(x) = \frac{\partial \phi}{\partial x}\frac{dx}{dt} = \frac{\partial \phi}{\partial x} f(x, u). \tag{3.20}$$

Now, consider the basis of observables Φ in Eq. (3.4).

Assumption 3.2

a) System (3.16) can be separated into its autonomous and input-dependent parts,

$$\dot{x} = f(x, u) = f_0(x) + g(x, u). \tag{3.21}$$

b) For system (3.21) and the basis of observables Φ (3.4), it holds that

$$L_{f_0}\phi(x) \in \operatorname{span}(\Phi), \quad \forall\, \phi \in \Phi. \tag{3.22}$$

By Assumption 3.2a, the lifted dynamics in Eq. (3.20) becomes

$$\dot{\phi}(x) = \frac{\partial \phi}{\partial x} \cdot (f_0(x) + g(x, u)) = L_{f_0}\phi(x) + \frac{\partial \phi}{\partial x} g(x, u). \tag{3.23}$$

Theorem 3.2 (Koopman lifted system – CT systems with inputs)

Let system (3.16) and a basis of observables $\Phi = \{\phi_i(x)\}_{i=1}^{N_o}$ (3.4) be given, satisfying Assumption 3.2. Let $x(t)$ be the solution of system (3.16) and $z(t)$ the solution of the following Koopman lifted system:

$$\dot{z} = Az + \mathcal{B}(z,u), \quad z(0) = \phi(x_0), \tag{3.24}$$

where $\mathcal{B}$ is defined through the equality $\mathcal{B}(\phi(x),u) = \mathcal{B}_x(x,u)$, with

$$\mathcal{B}_x(x,u) = \frac{\partial \phi}{\partial x} g(x,u). \tag{3.25}$$

Then, the two solutions are equivalent through the mapping defined by ϕ, i.e.,

$$z(t) = \phi(x(t)), \quad x(t) = \phi^{-1}(z(t)), \quad \forall\, t \geq 0.$$

Proof

By Assumption 3.2b, $\forall\, i = 1, \ldots, N_o$ we can write

$$L_{f_0}\phi_i(x) = a_i^\top \phi(x), \quad a_i \in \mathbb{R}^{N_o}. \tag{3.26}$$

Then, by Assumption 3.2a, recalling the lifted dynamics (3.23), we have that

$$\dot{\phi}_i(x) = a_i^\top \phi(x) + \frac{\partial \phi_i}{\partial x} g(x,u), \;\; i = 1, \ldots, N_o \tag{3.27a}$$

$$\Rightarrow \dot{\phi}(x) = A\phi(x) + \frac{\partial \phi}{\partial x} g(x,u) = A\phi(x) + \mathcal{B}_x(x,u), \tag{3.27b}$$

where $A = [a_1, \ldots, a_{N_o}]^\top$.

Now, by enforcing the equality $\mathcal{B}_x(x,u) = \mathcal{B}(\phi(x),u)$, we can obtain $\mathcal{B}(z,u)$.

Then, by applying the map ϕ to Eq. (3.27b) and adding the initial condition $z(0) = \phi(x_0)$, we obtain system (3.24).

Finally, it holds that $z(t) = \phi(x(t))$, $\forall\, t \geq 0$, being $x(t)$ and $z(t)$ the solutions of systems (3.16) and (3.24), respectively. If ϕ^{-1} exists, then it also holds $x(t) = \phi^{-1}(z(t))$.

Remark 3.4

If $\mathcal{B}_x(x,u)$ in Eq. (3.25) is affine in the observables Φ, then the term $\mathcal{B}(z,u)$ can be suitably chosen to be affine in z.

Additionally, the simplest choice for $\mathcal{B}$ that satisfies the equality $\mathcal{B}(\phi(x),u) = \mathcal{B}_x(x,u)$ is given by $\mathcal{B}(z,u) = \mathcal{B}_x(\phi^{-1}(z),u)$ (assuming that ϕ^{-1} exists). In this case, however, $\mathcal{B}(z,u)$ is, in general, nonlinear in z.

To sum up, we have transformed the CT system with inputs (3.16), (3.21) into its equivalent Koopman lifted system (3.24); these two systems are put in relation through the basis of observables (3.4), as follows:

$$\dot{x} = f(x,u) = f_0(x) + g(x,u) \quad \overset{\substack{z=\phi(x) \\ x=\phi^{-1}(z)}}{\Longleftrightarrow} \quad \dot{z} = Az + \mathcal{B}(z,u). \tag{3.28}$$

We can observe that, however, for systems with a general nonlinearity as in Eq. (3.16), the Koopman lifted system is, in general, still nonlinear, due to the term $\mathcal{B}(z,u)$ in Eq. (3.24).

Linear/Affine Parameter-Varying Form

To address the nonlinear dependence of $\mathcal{B}$ on the lifted state z and input u, we can try to express system (3.24) in a linear or affine parameter-varying (LPV or APV) form. We opt for such a formulation since LPV/APV systems are widely-employed in the domain of Model Predictive Control (MPC). Specifically, nonlinear prediction models in MPC can be replaced by more tractable LPV/APV representations, wherein the model parameter is set to the shifted optimal trajectory computed at the previous time step. This approach enables an effective approximation of the original system nonlinearity along the prediction horizon. For more details, we refer the reader to Section 2.3.3.

To express system (3.24) in LPV or APV form, we can proceed in two ways:

1) We can rewrite the input-dependent term as $\mathcal{B}(z,u) = B(z,u)u$, so that the input u enters both nonlinearly through B and linearly via multiplication. To do so, we resort to the following lemma [74], under the assumption that $\mathcal{B}(z,0) = 0$:

Lemma 3.1 (Factorization lemma [74, Lemma 1])

Let $\mathcal{B} : \mathcal{Z} \times \mathcal{U} \to \mathcal{Z}$ be continuously differentiable in u, continuous in z, and satisfying $\mathcal{B}(z,0) = 0$. Then,

$$B(z,u) = \int_0^1 \frac{\partial \mathcal{B}}{\partial u}(z, \lambda u) d\lambda \tag{3.29}$$

provides a factorization of $\mathcal{B}$ such that $\mathcal{B}(z,u) = B(z,u)u$.

Remark 3.5

To factorize $\mathcal{B}(z,u)$ through Lemma 3.1, it is required that $\mathcal{B}(z,0) = 0$. This holds under the sufficient condition $g(x,0) = 0$. In fact, $\mathcal{B}(\phi(x),0) = \mathcal{B}_x(x,0)$ and, by Eq. (3.25), $\mathcal{B}_x(x,0) = \frac{\partial \phi}{\partial x} g(x,0) = 0 \Leftarrow g(x,0) = 0$.

By Lemma 3.1, system (3.24) becomes

$$\dot{z} = Az + B(z,u)u.$$

Then, by introducing the parameter $p = (\overline{z}, \overline{u}) \in \mathcal{Z} \times \mathcal{U}$ and defining the matrix $\tilde{B}(p) = \tilde{B}(\overline{z}, \overline{u})$, where $\tilde{B} = B$, the LPV form of the Koopman lifted system (3.24) is given by

$$\dot{z} = Az + \tilde{B}(\overline{z}, \overline{u})u. \tag{3.30}$$

2) If $\mathcal{B}(z,0) \neq 0$ and, thus, Lemma 3.1 cannot be applied, the input-dependent term $\mathcal{B}(z,u)$ can be replaced by its first-order Taylor expansion around a given point $(\overline{z}, \overline{u}) = p$, acting as parameter. Specifically,

$$\dot{z} = Az + \mathcal{B}(z,u)$$

$$
\begin{aligned}
&\approx Az + \mathcal{B}(\overline{z},\overline{u}) + \frac{\partial \mathcal{B}}{\partial z}(\overline{z},\overline{u})(z-\overline{z}) + \frac{\partial \mathcal{B}}{\partial u}(\overline{z},\overline{u})(u-\overline{u}) \\
&= \underbrace{\left(A + \frac{\partial \mathcal{B}}{\partial z}(\overline{z},\overline{u})\right)}_{\tilde{A}(\overline{z},\overline{u})} z + \underbrace{\frac{\partial \mathcal{B}}{\partial u}(\overline{z},\overline{u})}_{\tilde{B}(\overline{z},\overline{u})} u + \underbrace{\mathcal{B}(\overline{z},\overline{u}) - \frac{\partial \mathcal{B}}{\partial z}(\overline{z},\overline{u})\overline{z} - \frac{\partial \mathcal{B}}{\partial u}(\overline{z},\overline{u})\overline{u}}_{\tilde{b}(\overline{z},\overline{u})} \\
&= \tilde{A}(\overline{z},\overline{u})z + \tilde{B}(\overline{z},\overline{u})u + \tilde{b}(\overline{z},\overline{u}),
\end{aligned} \tag{3.31}
$$

yielding an APV form for the Koopman lifted system (3.24).

Input-Affine Systems

Let us now consider a continuous-time input-affine system, which is a particular case of system (3.16), (3.21), i.e.,

$$
\dot{x}(t) = f(x(t),u(t)) = f_0(x(t)) + \sum_{j=1}^{n_u} g_j(x(t))u_j(t), \quad x(0) = x_0 \in \mathcal{X}, \tag{3.32}
$$

where $x \in \mathcal{X}$, $u = [u_j]_{j=1}^{n_u} \in \mathcal{U}$, and $f_0, (g_j)_{j=1}^{n_u} : \mathcal{X} \to \mathcal{X}$.

The lifted dynamics in Eq. (3.20) becomes

$$
\dot{\phi}(x) = \frac{\partial \phi}{\partial x} f(x,u) = L_{f_0}\phi(x) + \sum_{j=1}^{n_u} L_{g_j}\phi(x)u_j. \tag{3.33}
$$

Now, consider the basis of observables Φ in Eq. (3.4).

Assumption 3.3

For system (3.32) and the basis of observables Φ (3.4), it holds that

$$
L_{f_0}\phi(x) \in \operatorname{span}(\Phi), \quad L_{g_j}\phi(x) \in \operatorname{span}(\Phi), \tag{3.34}
$$

with $j = 1,\dots,n_u$ and $\forall\, \phi \in \Phi$.

Theorem 3.3 (Koopman lifted system – CT input-affine systems)

Let system (3.32) and a basis of observables $\Phi = \{\phi_i(x)\}_{i=1}^{N_o}$ (3.4) be given, satisfying Assumption 3.3. Let $x(t)$ be the solution of system (3.32) and $z(t)$ the solution of the following bilinear Koopman lifted system:

$$
\dot{z} = Az + \sum_{j=1}^{n_u} B_j z u_j, \quad z(0) = \phi(x_0). \tag{3.35}
$$

Then, the two solutions are equivalent through the mapping defined by ϕ, i.e.,

$$
z(t) = \phi(x(t)), \quad x(t) = \phi^{-1}(z(t)), \quad \forall\, t \geq 0.
$$

Proof

By Assumption 3.3, $\forall\, i = 1,\dots,N_o$ we can write

$$
L_{f_0}\phi_i(x) = a_i^\top \phi(x), \quad a_i \in \mathbb{R}^{N_o},
$$

$$L_{g_j}\phi_i(x) = b_{i,j}^\top \phi(x), \quad b_{i,j} \in \mathbb{R}^{N_o}, \quad j = 1, \dots, n_u. \tag{3.36}$$

Then, recalling the lifted dynamics (3.33), we have that

$$\dot{\phi}_i(x) = a_i^\top \phi(x) + \sum_{j=1}^{n_u} b_{i,j}^\top \phi(x) u_j, \quad i = 1, \dots, N_o \tag{3.37a}$$

$$\Rightarrow \dot{\phi}(x) = A\phi(x) + \sum_{j=1}^{n_u} B_j \phi(x) u_j, \tag{3.37b}$$

where $A = [a_1, \dots, a_{N_o}]^\top$ and $B_j = [b_{1,j}, \dots, b_{N_o,j}]^\top$, $j = 1, \dots, n_u$.

By applying the map ϕ to Eq. (3.37b) and adding the initial condition $z(0) = \phi(x_0)$, we obtain system (3.35).

Then, it holds that $z(t) = \phi(x(t))$, $\forall\, t \geq 0$, being $x(t)$ and $z(t)$ the solutions of systems (3.32) and (3.35), respectively. If ϕ^{-1} exists, then it also holds $x(t) = \phi^{-1}(z(t))$.

In several practical cases, system (3.32) may also contain an input-linear term, i.e.,

$$\dot{x} = f_0(x) + \sum_{i=1}^{n_u} g_j(x) u_j = f_0(x) + B_0 u + \sum_{j=1}^{n_u} \gamma_j(x) u_j, \tag{3.38}$$

where $g_j(x) = (B_0)_{\cdot,j} + \gamma_j(x)$, $j = 1, \dots, n_u$. To obtain the related Koopman lifted system, we start by introducing a first corollary of Theorem 3.3:

Corollary 3.1

Given the results of Theorem 3.3, assume that the observable $\phi_{i^*}(x) = 1$ belongs to the basis $\Phi = \{\phi_i(x)\}_{i=1}^{N_o}$ ($i^* \in I = \{1, \dots, N_o\}$).

Construct the reduced basis $\breve{\Phi} = \Phi \setminus \{\phi_{i^*}(x)\}$, $|\breve{\Phi}| = N_o - 1$, with $\breve{\phi}(x) = [\phi_i(x)]_{i \in \breve{I}}$ ($\breve{I} = I \setminus \{i^*\}$), and consider the following bilinear Koopman lifted system:

$$\dot{\breve{z}} = \breve{A}\breve{z} + \breve{B}_0 u + \sum_{j=1}^{n_u} \breve{B}_j \breve{z} u_j + \breve{b}, \quad \breve{z}(0) = \breve{\phi}(x_0), \tag{3.39}$$

where $\breve{z} \in \mathbb{R}^{n_{\breve{z}}}$, $n_{\breve{z}} = N_o - 1$,

$$\breve{A} = (A)_{\breve{I},\breve{I}}, \tag{3.40a}$$

$$\breve{B}_0 = [(B_1)_{\breve{I},i^*}, \dots, (B_{n_u})_{\breve{I},i^*}], \tag{3.40b}$$

$$\breve{B}_j = (B_j)_{\breve{I},\breve{I}}, \quad j = 1, \dots, n_u, \tag{3.40c}$$

$$\breve{b} = (A)_{\breve{I},i^*}, \tag{3.40d}$$

and the matrices A, $(B_j)_{j=1}^{n_u}$ are those of system (3.35).

Then, the solutions $x(t)$, $z(t)$, and $\breve{z}(t)$ of systems (3.32), (3.35) and (3.39), respectively, are equivalent through the following relations:

$$\breve{z}(t) = \breve{\phi}(x(t)), \quad x(t) = \breve{\phi}^{-1}(\breve{z}(t)), \quad \breve{z}(t) = (z(t))_{\breve{I}}, \quad \forall\, t \geq 0. \tag{3.41}$$

Proof

We can replace $\phi_{i^*}(x) = 1$ into Eq. (3.37a) and remove the related dynamical equation (i.e., $\dot{\phi}_{i^*}(x) = 0$); defining $\breve{a}_i = (a_i)_{\breve{I}}$ and $\breve{b}_{i,j} = (b_{i,j})_{\breve{I}}$, we obtain

$$\dot{\phi}_i(x) = \breve{a}_i^\top \breve{\phi}(x) + (a_i)_{i^*} + \sum_{j=1}^{n_u} \breve{b}_{i,j}^\top \breve{\phi}(x) u_j + \sum_{j=1}^{n_u} (b_{i,j})_{i^*} u_j, \quad i \in \breve{I} \tag{3.42a}$$

$$\Rightarrow \dot{\breve{\phi}}(x) = \breve{A}\breve{\phi}(x) + \breve{b} + \sum_{j=1}^{n_u} \breve{B}_j \breve{\phi}(x) u_j + \breve{B}_0 u, \tag{3.42b}$$

where $\breve{A}$, $\breve{B}_0$, $\{\breve{B}_j\}_{j=1}^{n_u}$, and $\breve{b}$ are given by Eq. (3.40).

By applying the map $\breve{\phi}$ to Eq. (3.42b) and adding the initial condition $\breve{z}(0) = \breve{\phi}(x_0)$, we obtain system (3.39) and it holds that $\breve{z}(t) = \breve{\phi}(x(t))$, $x(t) = \breve{\phi}^{-1}(\breve{z}(t))$, $\forall\, t \geq 0$. By construction, it also holds that $\breve{z}(t) = (z(t))_{\breve{I}}$, $\forall\, t \geq 0$.

Then, we characterize the Koopman lifted system in this second corollary:

Corollary 3.2

Let system (3.38) and a basis of observables $\Phi = \{\phi_i(x)\}_{i=1}^{N_o}$ (3.4) be given, satisfying Assumption 3.3. Assume also that Φ contains the system states as the first n_x observables, i.e., $\{x_i\}_{i=1}^{n_x} \subset \Phi$ and $\phi(x) = [x^\top, \cdots]^\top$.

Then, building on the results of Theorem 3.3:

(i) The basis Φ contains the observable $\phi(x) = 1$.

(ii) With the reduced basis $\breve{\Phi} = \Phi \setminus \{1\}$, consider the following bilinear Koopman lifted system:

$$\dot{\breve{z}} = \breve{A}\breve{z} + \breve{B}_0 u + \sum_{i=1}^{n_u} \breve{B}_j \breve{z} u_j, \quad \breve{z}(0) = \breve{\phi}(x_0), \tag{3.43}$$

where $\breve{z} \in \mathbb{R}^{n_{\breve{z}}}$, $n_{\breve{z}} = n_z - 1$.

Then, the solutions $x(t)$ and $\breve{z}(t)$ of systems (3.32) and (3.43), respectively, are equivalent through the mapping defined by $\breve{\phi}$, i.e.,

$$\breve{z}(t) = \breve{\phi}(x(t)), \quad x(t) = \breve{\phi}^{-1}(\breve{z}(t)), \quad \forall\, t \geq 0. \tag{3.44}$$

Proof

Let us rewrite $B_0 u = \sum_{j=1}^{n_u} (B_0)_{\cdot,j} u_j$. The lifted dynamics of system (3.38) is given by

$$\dot{\phi}(x) = \frac{\partial \phi}{\partial x} f(x,u) = L_{f_0}\phi(x) + \sum_{j=1}^{n_u} \frac{\partial \phi}{\partial x} \cdot (B_0)_{\cdot,j} u_j + \sum_{j=1}^{n_u} L_{\gamma_j}\phi(x) u_j. \tag{3.45}$$

Then, Assumption 3.3 can be equivalently restated as

$$L_{f_0}\phi(x) \in \operatorname{span}(\Phi), \quad L_{\gamma_j}\phi(x) \in \operatorname{span}(\Phi), \quad \frac{\partial \phi}{\partial x} \cdot (B_0)_{\cdot,j} \in \operatorname{span}(\Phi), \tag{3.46}$$

with $j = 1, \ldots, n_u$ and $\forall\, \phi \in \Phi$.

Since the system states $\{x_i\}_{i=1}^{n_x}$ are observables within Φ, by Eq. (3.46) it holds that

$$\frac{\partial(x_i)}{\partial x} \cdot (B_0)_{\cdot,j} = e_i^\top (B_0)_{\cdot,j} = (B_0)_{i,j} \in \text{span}(\Phi), \quad i = 1, \dots, n_x. \tag{3.47}$$

where e_i is the i-th vector of the standard Euclidean basis of $\mathbb{R}^{n_x}$. Being $(B_0)_{i,j}$ a constant term, Eq. (3.47) implies that $\phi(x) = 1$ is an observable within the basis Φ, proving statement (i).

Since $f_0(x)$ and $\gamma_j(x)$, $j = 1, \dots, n_u$, do not contain any constant affine term, it holds that

$$L_{f_0}\phi(x) \in \text{span}(\breve{\Phi}), \quad L_{\gamma_j}\phi(x) \in \text{span}(\breve{\Phi}), \quad \forall\, \phi \in \Phi. \tag{3.48}$$

Then, following Corollary 3.1, the Koopman lifted system (3.35) can be rewritten under the reduced basis $\breve{\Phi}$, obtaining

$$\dot{\breve{z}} = \breve{A}\breve{z} + \breve{B}_0 u + \sum_{j=1}^{n_u} \breve{B}_j \breve{z} u_j + \breve{b}, \quad \breve{z}(0) = \breve{\phi}(x_0), \tag{3.49}$$

where, by Eqs. (3.40), (3.47) and (3.48),

$$\breve{A} = (A)_{\breve{I},\breve{I}}, \qquad \breve{B}_0 = \begin{bmatrix} B_0 \\ \mathbf{0}_{n_{\breve{z}} - n_x \times n_u} \end{bmatrix},$$
$$\breve{B}_j = (B_j)_{\breve{I},\breve{I}}, \qquad \breve{b} = \mathbf{0}_{n_{\breve{z}}},$$

which proves statement (ii).

To sum up, we have transformed the CT input-affine system (3.32) into its equivalent bilinear Koopman lifted system (3.35); these two systems are put in relation through the basis of observables (3.4), as follows:

$$\dot{x} = f_0(x) + \sum_{j=1}^{n_u} g_j(x) u_j \quad \overset{\substack{z = \phi(x) \\ x = \phi^{-1}(z)}}{\Longleftrightarrow} \quad \dot{z} = Az + \sum_{j=1}^{n_u} B_j z u_j. \tag{3.50}$$

In the case that an input-linear term is present, the following particular case holds:

$$\dot{x} = f_0(x) + B_0 u + \sum_{j=1}^{n_u} \gamma_j(x) u_j \quad \overset{\substack{z = \breve{\phi}(x) \\ x = \breve{\phi}^{-1}(z)}}{\Longleftrightarrow} \quad \dot{z} = \breve{A}z + \breve{B}_0 u + \sum_{j=1}^{n_u} \breve{B}_j z u_j. \tag{3.51}$$

We see that, by considering an input-affine structure rather than a general nonlinearity for system (3.16), we are able to obtain a bilinear Koopman lifted system (3.35), rather than a nonlinear one as in Eq. (3.24).

Linear Parameter-Varying Form The Koopman lifted system (3.35) can be expressed in LPV form. Specifically, we observe that the system is already in factorized form, i.e.,

$$\dot{z} = Az + \sum_{j=1}^{n_u} B_j z u_j = Az + \underbrace{[B_1 z, \cdots, B_{n_u} z]}_{B(z)} u = Az + B(z)u.$$

Then, by introducing the parameter $p = \overline{z} \in \mathcal{Z}$, the LPV form is given by

$$\dot{z} = Az + \tilde{B}(\overline{z})u, \quad \tilde{B} = B. \tag{3.52}$$

Remark 3.6

For Assumptions 3.1, 3.2b, and 3.3 to hold, an infinite-dimensional basis Φ (3.4) may be needed, yielding an infinite-dimensional Koopman lifted system (3.3), (3.24), (3.35).

3.2.3. Discrete-Time Autonomous Systems

Let us consider a discrete-time (DT) autonomous system, i.e.,

$$x_{k+1} = f(x_k), \quad k \in \mathbb{Z}_{\geq 0}, \tag{3.53}$$

where $x \in \mathcal{X} \subseteq \mathbb{R}^{n_x}$ is the state vector and $f : \mathcal{X} \to \mathcal{X}$ is the state transition function.

For any well-defined f, there exists a unique solution x_k of system (3.53) in the time interval $[0, +\infty) \cap \mathbb{Z}_{\geq 0}$, starting from x_0, i.e.,

$$x_k = \underbrace{f(\cdots f(f}_{k \text{ times}}(x_0))\cdots) = \varphi^k(x_0). \tag{3.54}$$

The family of one-parameter maps $\varphi^k : \mathcal{X} \to \mathcal{X}$ is the flow of system (3.53).

Definition 3.3 (Koopman operator – DT autonomous systems)

Let $\mathscr{F} \subseteq \mathcal{C}^1$ be a Banach space of continuously differentiable observables $\phi : \mathcal{X} \to \mathbb{R}$. The family of Koopman operators $\mathcal{K}^k : \mathscr{F} \to \mathscr{F}$ associated with the family of maps φ^k (3.54) is defined as follows:

$$\mathcal{K}^k \phi = \phi \circ \varphi^k, \quad \forall\, \phi \in \mathscr{F}. \tag{3.55}$$

The Koopman operator (3.55) provides the lifted trajectories $z_k = \phi(x_k) = \mathcal{K}^k \phi(x_0)$, $\forall\, \phi \in \mathscr{F}$. The lifted dynamics is obtained by computing the successor state z_{k+1}, i.e.,

$$z_{k+1} = \phi(x_{k+1}) = \phi(f(x_k)). \tag{3.56}$$

Recalling Eq. (3.7), the lifted dynamics for CT and DT systems compare as follows:

$$z^+ = \phi(x)^+ = \begin{cases} \frac{\partial \phi}{\partial x} f(x) & \text{in CT,} \\ \phi(f(x)) & \text{in DT,} \end{cases}$$

where $(\star)^+$ is the successor operator (i.e., $\dot{\star}(t)$ in CT, $\star_{k+1}$ in DT).

Now, consider the basis of observables Φ in Eq. (3.4).

Assumption 3.4

For system (3.53) and the basis of observables Φ (3.4), it holds that

$$\phi(f(x)) \in \operatorname{span}(\Phi), \quad \forall\, \phi \in \Phi. \tag{3.57}$$

Theorem 3.4 (Koopman lifted system – DT autonomous systems)

Let system (3.53) and a basis of observables $\Phi = \{\phi_i(x)\}_{i=1}^{N_o}$ (3.4) be given, satisfying Assumption 3.4. Let x_k be the solution of system (3.53) and z_k the solution of the following linear Koopman lifted system:

$$z_{k+1} = Az_k, \quad z_0 = \phi(x_0). \tag{3.58}$$

Then, the two solutions are equivalent through the mapping defined by ϕ, i.e.,

$$z_k = \phi(x_k), \quad x_k = \phi^{-1}(z_k), \quad \forall\, k \geq 0. \tag{3.59}$$

Proof

By Assumption 3.4, $\forall\, i = 1, \ldots, N_o$ we can write

$$\phi_i(f(x_k)) = a_i^\top \phi_i(x_k), \quad a_i \in \mathbb{R}^{N_o}. \tag{3.60}$$

Then, recalling the lifted dynamics (3.56), we have that

$$\phi_i(x_{k+1}) = a_i^\top \phi_i(x_k), \;\; i = 1, \ldots, N_o \tag{3.61a}$$

$$\Rightarrow \phi(x_{k+1}) = A\phi(x_k), \tag{3.61b}$$

where $A = [a_1, \ldots, a_{N_o}]^\top$.

By applying the map ϕ to Eq. (3.61b) and adding the initial condition $z_0 = \phi(x_0)$, we obtain system (3.58).

Then, it holds that $z_k = \phi(x_k)$, $\forall\, k \geq 0$, being x_k and z_k the solutions of systems (3.53) and (3.58), respectively. If ϕ^{-1} exists, then it also holds $x_k = \phi^{-1}(z_k)$.

To sum up, we have transformed the DT autonomous system (3.53) into its equivalent linear Koopman lifted system (3.58); these two systems are put in relation through the basis of observables (3.4), as follows:

$$x_{k+1} = f(x_k) \quad \begin{matrix} z_k = \phi(x_k) \\ x_k = \phi^{-1}(z_k) \\ \Longleftrightarrow \end{matrix} \quad z_{k+1} = Az_k. \tag{3.62}$$

3.2.4. Discrete-Time Systems With Inputs

Let us now consider a DT system with inputs, i.e.,

$$x_{k+1} = f(x_k, u_k), \quad k \geq 0, \tag{3.63}$$

where $x \in \mathcal{X} \subseteq \mathbb{R}^{n_x}$ and $u \in \mathcal{U} \subseteq \mathbb{R}^{n_u}$ are the state and input vectors, respectively, and $f : \mathcal{X} \times \mathcal{U} \to \mathcal{X}$ is the state transition function.

Let $\boldsymbol{u} = (u_k)_{k \in [0,+\infty) \cap \mathbb{Z}_{\geq 0}}$ be the input signal, belonging to the space $\mathscr{U} = \mathcal{U}^\infty$. For any well-defined f, there exists a unique solution x_k of system (3.63) in the time interval $[0, +\infty) \cap \mathbb{Z}_{\geq 0}$, starting from x_0, i.e.,

$$x_k = \underbrace{f(\cdots f(f}_{k \text{ times}}(x_0, u_0), u_1) \cdots u_{k-1}) = \varphi^k(x_0, \boldsymbol{u}). \tag{3.64}$$

The family of one-parameter maps $\varphi^k : \mathcal{X} \times \mathscr{U} \to \mathcal{X}$ is the flow of system (3.63).

Definition 3.4 (Koopman operator – DT systems with inputs)

Let $\mathscr{F} \subseteq \mathcal{C}^1$ be a Banach space of continuously differentiable observables $\phi : \mathcal{X} \to \mathbb{R}$. The family of Koopman operators $\mathcal{K}^{k,\boldsymbol{u}} : \mathscr{F} \to \mathscr{F}$ associated with the family of maps $\varphi^k(\cdot, \boldsymbol{u})$ (3.64) is defined as follows:

$$\mathcal{K}^{k,\boldsymbol{u}}\phi(\cdot) = \phi \circ \varphi^k(\cdot, \boldsymbol{u}), \quad \forall\, \phi \in \mathscr{F}. \tag{3.65}$$

The Koopman operator (3.65) provides the lifted trajectories $z_k = \phi(x_k) = \mathcal{K}^{k,\boldsymbol{u}}\phi(x_0)$, $\forall\, \phi \in \mathscr{F}$. The lifted dynamics is obtained by computing the successor state z_{k+1}, i.e.,

$$z_{k+1} = \phi(x_{k+1}) = \phi(f(x_k, u_k)). \tag{3.66}$$

Recalling Eq. (3.20), the lifted dynamics for CT and DT systems compare as follows:

$$z^+ = \phi(x)^+ = \begin{cases} \frac{\partial \phi}{\partial x} f(x,u) & \text{in CT,} \\ \phi(f(x,u)) & \text{in DT.} \end{cases}$$

Now, consider the basis of observables Φ in Eq. (3.4).

Assumption 3.5

a) System (3.63) can be separated into its autonomous and input-dependent parts,

$$x_{k+1} = f(x_k, u_k) = f_0(x_k) + g(x_k, u_k). \tag{3.67}$$

b) For system (3.67) and the basis of observables Φ (3.4), it holds that

$$\phi(f_0(x)) \in \operatorname{span}(\Phi), \quad \forall\, \phi \in \Phi. \tag{3.68}$$

By Assumption 3.5a, the lifted dynamics in Eq. (3.66) becomes

$$\phi(x_{k+1}) = \phi(f_0(x_k) + g(x_k, u_k)). \tag{3.69}$$

In contrast to the CT case in Eq. (3.23), the expression $\phi(f_0(x_k) + g(x_k, u_k))$ in Eq. (3.69) cannot be linearly expanded due to the nonlinearity of $\phi(\cdot)$. This makes it hard, in the DT case, to directly separate the autonomous and input-dependent parts of the lifted dynamics. Nonetheless, an analytically exact representation for the Koopman lifted system can be found by resorting to the fundamental theorem of calculus [74].

Theorem 3.5 (Koopman lifted system – DT systems with inputs)

Let system (3.63) and a basis of observables $\Phi = \{\phi_i(x)\}_{i=1}^{N_o}$ (3.4) be given, satisfying Assumption 3.5. Let x_k be the solution of system (3.63) and z_k the solution of the following Koopman lifted system:

$$z_{k+1} = Az_k + \mathcal{B}(z_k, u_k), \quad z_0 = \phi(x_0), \tag{3.70}$$

where $\mathcal{B}$ is defined through the equality $\mathcal{B}(\phi(x_k), u_k) = \mathcal{B}_x(x_k, u_k)$, with

$$\mathcal{B}_x(x_k, u_k) = \int_0^1 \frac{\partial \phi}{\partial x}(f_0(x_k) + \lambda g(x_k, u_k))d\lambda \, g(x_k, u_k). \tag{3.71}$$

Then, the two solutions are equivalent through the mapping defined by ϕ, i.e.,

$$z_k = \phi(x_k), \quad x_k = \phi^{-1}(z_k), \quad \forall\, k \geq 0.$$

Proof

For each $\phi_i(x)$, $i = 1, \ldots, N_o$, the fundamental theorem of calculus [74] states that

$$\phi_i(q) - \phi_i(p) = \int_p^q \frac{\partial \phi_i}{\partial x}(x)dx = \int_0^1 \frac{\partial \phi_i}{\partial x}(x(\lambda))\frac{\partial x}{\partial \lambda}(\lambda)d\lambda,$$

where the change of variable $x \leftrightarrow x(\lambda)$ has been performed, with $x(0) = p$ and $x(1) = q$. We can set $x(\lambda) = p + \lambda(q - p)$, for which $\frac{\partial x}{\partial \lambda}(\lambda) = q - p$ and

$$\phi_i(q) - \phi_i(p) = \int_0^1 \frac{\partial \phi_i}{\partial x}(p + \lambda(q - p))d\lambda \, (q - p). \tag{3.72}$$

Now, let $q = x_{k+1} = f_0(x_k) + g(x_k, u_k)$ by Assumption 3.5a, and $p = f_0(x_k)$. Replacing them into Eq. (3.72) yields

$$\begin{aligned} &\phi_i(x_{k+1}) - \phi_i(f_0(x_k)) \overset{\text{Ass. 3.5b}}{=} \phi_i(x_{k+1}) - a_i^\top \phi_i(x_k) \\ &= \int_0^1 \frac{\partial \phi_i}{\partial x}(f_0(x_k) + \lambda g(x_k, u_k))d\lambda \, g(x_k, u_k) \end{aligned}$$

$$\begin{aligned} \Rightarrow \; \phi(x_{k+1}) &= A\phi(x_k) + \int_0^1 \frac{\partial \phi}{\partial x}(f_0(x_k) + \lambda g(x_k, u_k))d\lambda \, g(x_k, u_k) \\ &= A\phi(x_k) + \mathcal{B}_x(x_k, u_k), \end{aligned} \tag{3.73}$$

where $A = [a_1, \ldots, a_{N_o}]^\top$.

Now, by enforcing the equality $\mathcal{B}_x(x_k, u_k) = \mathcal{B}(\phi(x_k), u_k)$, we can obtain $\mathcal{B}(x_k, u_k)$.

Then, by applying the map ϕ to Eq. (3.73) and adding the initial condition $z_0 = \phi(x_0)$, we obtain system (3.70).

Then, it holds that $z_k = \phi(x_k)$, $\forall\, k \geq 0$, being x_k and z_k the solutions of systems (3.63) and (3.70), respectively. If ϕ^{-1} exists, then it also holds $x_k = \phi^{-1}(z_k)$.

To sum up, we have transformed the DT system with inputs (3.63), (3.67) into its equivalent Koopman lifted system (3.70); these two systems are put in relation through the basis of observables (3.4), as follows:

$$x_{k+1} = f(x_k, u_k) = f_0(x_k) + g(x_k, u_k) \quad \overset{\substack{z_k = \phi(x_k) \\ x_k = \phi^{-1}(z_k)}}{\Longleftrightarrow} \quad z_{k+1} = Az_k + \mathcal{B}(z_k, u_k). \tag{3.74}$$

As for the CT case in Eq. (3.24), we can observe that, for systems with a general nonlinearity as in Eq. (3.63), the Koopman lifted system is, in general, still nonlinear, due to the term $\mathcal{B}(z_k, u_k)$ in Eq. (3.70).

Linear/Affine Parameter-Varying Form

Also in the DT case, to address the nonlinear dependence of $\mathcal{B}$ on the lifted state z_k and input u_k, we can express system (3.70) in LPV or APV form:

1) If $\mathcal{B}(z_k, 0) = 0$, we can apply Lemma 3.1 to factorize the input-dependent term as $\mathcal{B}(z_k, u_k) = B(z_k, u_k)u_k$, so that system (3.70) can be rewritten as

$$z_{k+1} = Az_k + B(z_k, u_k)u_k.$$

Then, by introducing the parameter $p = (\overline{z}, \overline{u}) \in \mathcal{Z} \times \mathcal{U}$ and defining the matrix $\tilde{B}(p) = \tilde{B}(\overline{z}, \overline{u})$, where $\tilde{B} = B$, the LPV form of the Koopman lifted system (3.70) is given by

$$z_{k+1} = Az_k + \tilde{B}(\overline{z}, \overline{u})u_k. \tag{3.75}$$

Remark 3.7

To factorize $\mathcal{B}(z_k, u_k)$ through Lemma 3.1, it is required that $\mathcal{B}(z_k, 0) = 0$. This holds under the sufficient condition $g(x_k, 0) = 0$. In fact, $\mathcal{B}(\phi(x_k), 0) = \mathcal{B}_x(x_k, 0)$ and, by Eq. (3.71), $\mathcal{B}_x(x_k, 0) = \int_0^1 \frac{\partial \phi}{\partial x}(f_0(x_k) + \lambda g(x_k, 0))d\lambda\, g(x_k, 0) = 0 \Leftarrow g(x_k, 0) = 0$.

2) If $\mathcal{B}(z_k, 0) \neq 0$ and, thus, Lemma 3.1 cannot be applied, the input-dependent term $\mathcal{B}(z_k, u_k)$ can be replaced by its first-order Taylor expansion around a given point $(\overline{z}, \overline{u}) = p$, acting as parameter. Specifically,

$$\begin{aligned} z_{k+1} &= Az_k + \mathcal{B}(z_k, u_k) \\ &\approx Az_k + \mathcal{B}(\overline{z}, \overline{u}) + \frac{\partial \mathcal{B}}{\partial z}(\overline{z}, \overline{u})(z_k - \overline{z}) + \frac{\partial \mathcal{B}}{\partial u}(\overline{z}, \overline{u})(u_k - \overline{u}) \\ &= \underbrace{\left(A + \frac{\partial \mathcal{B}}{\partial z}(\overline{z}, \overline{u})\right)}_{\tilde{A}(\overline{z}, \overline{u})} z_k + \underbrace{\frac{\partial \mathcal{B}}{\partial u}(\overline{z}, \overline{u})}_{\tilde{B}(\overline{z}, \overline{u})} u_k + \underbrace{\mathcal{B}(\overline{z}, \overline{u}) - \frac{\partial \mathcal{B}}{\partial z}(\overline{z}, \overline{u})\overline{z} - \frac{\partial \mathcal{B}}{\partial u}(\overline{z}, \overline{u})\overline{u}}_{\tilde{b}(\overline{z}, \overline{u})} \\ &= \tilde{A}(\overline{z}, \overline{u})z_k + \tilde{B}(\overline{z}, \overline{u})u_k + \tilde{b}(\overline{z}, \overline{u}), \end{aligned} \tag{3.76}$$

yielding an APV form for the Koopman lifted system (3.70).

Remark 3.8

Due to the nonlinearity of $\phi(\cdot)$ in Eq. (3.69), a DT input-affine system would still retain a nonlinear Koopman lifted system – and not bilinear, as in the CT case.

For this reason, it is typically preferable to lift CT systems in the Koopman space, and then discretize the resulting Koopman lifted system to obtain a DT representation.

This approach is particularly convenient for input-affine systems (3.32), whose Koopman lifted system (3.35) is bilinear and can be directly factorized in LPV form, i.e., $\dot{z} = Az + \tilde{B}(\overline{z})u$ as in Eq. (3.52). Under Assumption 2.7, the LPV lifted system admits an exact discretization, which can be obtained through the following convolution

integral (Section 2.2, p. 12):

$$\begin{aligned} z_{k+1} &= e^{AT_s} z_k + \int_{kT_s}^{(k+1)T_s} e^{A((k+1)T_s-\tau)} \tilde{B}(\overline{z}) u(\tau) d\tau \\ &= \underbrace{e^{AT_s}}_{A_d} z_k + \underbrace{(e^{AT_s} - I) A^{-1} \tilde{B}(\overline{z})}_{B_d(\overline{z})} u_k = A_d z_k + B_d(\overline{z}) u_k, \end{aligned} \tag{3.77}$$

where T_s is the discrete time step.

This approach is going to be applied when extending the Koopman operator framework to NMPC problems, in order to derive a linear DT version of the CT Koopman lifted system, to serve as the MPC prediction model (see Sections 3.5.3 and 3.5.5).

Remark 3.9

For Assumptions 3.4 and 3.5b to hold, an infinite-dimensional basis Φ (3.4) may be needed, yielding an infinite-dimensional Koopman lifted system (3.58), (3.70).

3.3. Finding the Basis of Koopman Observables

After having characterized the Koopman lifted form of general nonlinear systems, let us focus our attention on systems with inputs, given by Eqs. (3.16), (3.32), and (3.63) – representing, respectively, continuous-time, continuous-time input-affine, and discrete-time systems.

Now, a key challenge to address is finding suitable bases of observables Φ (3.4) for such systems, satisfying Assumptions 3.2b, 3.3, and 3.5b, respectively. If we are able to find such bases, the related Koopman lifted systems can be effectively constructed by leveraging Theorems 3.2, 3.3, and 3.5, respectively.

In this section, we propose a novel algorithmic procedure that allows to analytically derive the basis Φ in a systematic way. Such a procedure constructs Φ starting from an initial small set of hand-picked observables Φ_{in} and, from it, generates the whole basis.

In the following, we begin by considering the general CT and DT cases (3.16), (3.63). Then, we focus on the CT input-affine case (3.32), which requires a more tailored analysis due to its differing assumption (Assumption 3.3).

3.3.1. Continuous-Time and Discrete-Time Systems With Inputs

For notational simplicity, let us combine CT and DT representations as follows:

- Original systems (3.16) and (3.63) (under Assumptions 3.2a and 3.5a):

$$x^+ = f(x,u) = f_0(x) + g(x,u), \tag{3.78}$$

- Lifted dynamics (3.20) and (3.66):

$$z^+ = \phi(x)^+ = \mathcal{L}_\phi(f(x,u)), \quad \mathcal{L}_\phi(\star) = \begin{cases} \frac{\partial \phi}{\partial x} \cdot (\star) & \text{in CT,} \\ \phi(\star) & \text{in DT,} \end{cases} \quad \forall\, \phi \in \mathscr{F}, \tag{3.79}$$

- Koopman lifted systems (3.24) and (3.70):

$$z^+ = Az + \mathcal{B}(z, u), \tag{3.80}$$

- Conditions required by Assumptions 3.2b and 3.5b:

$$\mathcal{L}_\phi(f_0(x)) \in \text{span}(\Phi), \quad \forall\, \phi \in \Phi. \tag{3.81}$$

In Algorithm 3.1, we describe the procedure to analytically derive a basis Φ for system (3.78), satisfying Eq. (3.81):

Algorithm 3.1 (Koopman observables basis generation)

Input: $f_0(x)$, Φ_{in}
Output: Φ

$\Phi \leftarrow \Phi_{\text{in}}$, $i \leftarrow 1$
while $i \leq |\Phi|$ **do**
  $\alpha(x) \leftarrow \mathcal{L}_{\phi_i}(f_0(x))$
  Expand $\alpha(x)$ = linear combination of $\Phi + \alpha_{\text{res}}(x)$
  Expand $\alpha_{\text{res}}(x)$ = linear combination of nonlinear terms $\{\psi_{\alpha,i}(x)\}_{i=1}^{N_\alpha} = \Psi_\alpha$
  $\Phi \leftarrow \Phi \cup \Psi_\alpha$, $i \leftarrow i + 1$
return Φ

The basis Φ is incrementally built. Φ is initialized with an initial set of observables Φ_{in}, containing the system states $\{x_i\}_{i=1}^{n_x}$ (so to ensure the existence of the inverse map ϕ^{-1}, see Remark 3.2) and, possibly, some other functions of choice $\{\psi_{\text{in},i}(x)\}_{i=1}^{N_{\text{in}}}$. These additional functions are particularly important when extending the Koopman operator framework to NMPC problems. Specifically, as explained in Section 3.5.2, we shall include in Φ_{in} all the elementary nonlinear terms introduced by the NMPC nonlinear state constraints. This allows to obtain a basis Φ that lifts both the CT plant and the NMPC constraints, which represent the two nonlinearities of the NMPC problem.

For each observable $\phi_i(x) \in \Phi$, $i \leq |\Phi|$, the autonomous part of lifted dynamics, i.e., $\mathcal{L}_{\phi_i}(f_0(x)) = \alpha(x)$ in Eqs. (3.79) and (3.81), is computed. This term α is then expanded as a linear combination of the current observables in Φ and a nonlinear residual $\alpha_{\text{res}}(x)$. The residual α_{res} is in turn expanded as a linear combination of nonlinear terms $\{\psi_{\alpha,i}\}_{i=1}^{N_\alpha}$; such nonlinear terms are appended to Φ as new observables of the basis.

The above procedure is iterated until convergence – i.e., a finite-dimensional basis Φ is found, yielding no further residual terms – or, possibly, after a maximum number of observables has been generated (see Section 3.4).

After having found the basis Φ, we can construct the Koopman lifted system (3.80) following Theorems 3.2 and 3.5. Such a procedure is detailed in Algorithm 3.2:

Algorithm 3.2 (Koopman lifted system construction)

Input: $f_0(x)$, $g(x, u)$, Φ
Output: A, $\mathcal{B}(z, u)$ (**optional:** $B(z, u)$)

for $i = 1, \ldots, |\Phi| = N_o$ **do**
  $\alpha(x) \leftarrow \mathcal{L}_{\phi_i}(f_0(x))$
  Expand $\alpha(x) = a_i^\top \phi(x)$, $a_i \in \mathbb{R}^{N_o}$
$A \leftarrow [a_1, \ldots, a_{N_o}]^\top$

$\mathcal{B}_x(x,u) \leftarrow \begin{cases} \frac{\partial \phi}{\partial x} g(x,u) & \text{in CT,} \\ \int_0^1 \frac{\partial \phi}{\partial x}(f_0(x) + \lambda g(x,u)) d\lambda \, g(x,u) & \text{in DT} \end{cases}$
Find $\mathcal{B}(z,u)$ by enforcing $\mathcal{B}(\phi(x),u) = \mathcal{B}_x(x,u)$
if $g(x,0) = 0$ **then**
    $B(z,u) \leftarrow \int_0^1 \frac{\partial \mathcal{B}}{\partial u}(z, \lambda u) d\lambda$
    return $B(z,u)$
return A, $\mathcal{B}(z,u)$

3.3.2. Continuous-Time Input-Affine Systems

Now, let us move on to CT input-affine systems (3.32). Such systems require specific attention since the underlying Assumption 3.3 slightly differs from those in the general CT and DT cases (i.e., Assumptions 3.2b and 3.5b), as it requires that both the autonomous ($L_{f_0}\phi(x)$) and input-dependent ($L_{g_j}\phi(x)$, $j = 1, \dots, n_u$) parts of the lifted dynamics (3.33) belong to the Koopman invariant subspace spanned by Φ.

Therefore, Algorithm 3.1 is modified by computing, for each observable $\phi_i(x) \in \Phi$, $i \leq |\Phi|$, both of these parts of the lifted dynamics, i.e., $L_{f_0}\phi_i(x) = \alpha(x)$ and $L_{g_j}\phi(x) = \beta_j(x)$, $j = 1, \dots, n_u$. Then, each term α, β_j is expanded as a linear combination of the current observables in Φ and a nonlinear residual (α_{res}, $\beta_{\text{res},j}$). The residuals are in turn expanded as a linear combination of nonlinear terms (ψ_α, $\psi_{\beta,j}$); such nonlinear terms are appended to Φ as new observables of the basis.

This modified procedure is reported in Algorithm 3.3:

Algorithm 3.3 (Koopman observables basis generation – CT input-affine systems)

Input: $f_0(x)$, $(g_j(x))_{j=1}^{n_u}$, Φ_{in}
Output: Φ

$\Phi \leftarrow \Phi_{\text{in}}$, $i \leftarrow 1$
while $i \leq |\Phi|$ **do**
    $\alpha(x) \leftarrow \frac{\partial \phi_i}{\partial x} f_0(x)$
    Expand $\alpha(x)$ = linear combination of $\Phi + \alpha_{\text{res}}(x)$
    Expand $\alpha_{\text{res}}(x)$ = linear combination of nonlinear terms $\{\psi_{\alpha,i}(x)\}_{i=1}^{N_\alpha} = \Psi_\alpha$
    for $j = 1, \dots, n_u$ **do**
        $\beta_j(x) \leftarrow \frac{\partial \phi_i}{\partial x} g_j(x)$
        Expand $\beta_j(x)$ = linear combination of $\Phi + \beta_{\text{res},j}(x)$
        Expand $\beta_{\text{res},j}(x)$ = linear combination of nonlinear terms $\{\psi_{\beta,j,i}(x)\}_{i=1}^{N_{\beta,j}} = \Psi_{\beta,j}$
    $\Phi \leftarrow \Phi \cup \Psi_\alpha \cup \left(\bigcup_{j=1}^{n_u} \Psi_{\beta,j}\right)$, $i \leftarrow i + 1$
return Φ

After having found the basis Φ, we can construct the Koopman lifted system (3.35) following Theorem 3.3. Such a procedure is detailed in Algorithm 3.4:

Algorithm 3.4 (Koopman lifted system construction – CT input-affine systems)

Input: $f_0(x)$, $(g_j(x))_{j=1}^{n_u}$, Φ
Output: A, $(B_j)_{j=1}^{n_u}$

for $i = 1, \dots, |\Phi| = N_o$ **do**
    $\alpha(x) \leftarrow \frac{\partial \phi_i}{\partial x} f_0(x)$

Expand $\alpha(x) = a_i^\top \phi(x)$, $a_i \in \mathbb{R}^{N_o}$
for $j = 1, \dots, n_u$ **do**
  $\beta_j(x) \leftarrow \frac{\partial \phi_i}{\partial x} g_j(x)$
  Expand $\beta_j(x) = b_{i,j}^\top \phi(x)$, $b_{i,j} \in \mathbb{R}^{N_o}$
$A \leftarrow [a_1, \dots, a_{N_o}]^\top$, $B_j \leftarrow [b_{1,j}, \dots, b_{N_o,j}]^\top$, $j = 1, \dots, n_u$
return A, $(B_j)_{j=1}^{n_u}$

Remark 3.10

If the CT input-affine system contains an input-linear term as in Eq. (3.38), Algorithm 3.4 can be suitably modified according to the results of Corollaries 3.1 and 3.2.

We now illustrate, through some examples, the procedures for generating the basis of Koopman observables and constructing the related Koopman lifted system.

Example 3.1 Consider the following CT system with inputs:

$$\dot{x} = \begin{bmatrix} x_1 + x_2 \sin(u_1) \\ -x_2 + x_1^2 + u_1 u_2 \\ x_3 - x_1 x_2 + x_1 e^{u_2} \end{bmatrix} = \underbrace{\begin{bmatrix} x_1 \\ -x_2 + x_1^2 \\ x_3 - x_1 x_2 + x_1 \end{bmatrix}}_{f_0(x)} + \underbrace{\begin{bmatrix} x_2 \sin(u_1) \\ u_1 u_2 \\ x_1 e^{u_2} - x_1 \end{bmatrix}}_{g(x,u)}, \tag{3.82}$$

which has been suitably separated into autonomous and input-dependent parts, so that $g(x, 0) = 0$ (Remark 3.5).

We apply Algorithm 3.1 to generate a basis Φ for system (3.82) satisfying Assumption 3.2b. Being in CT, we have that, by Eq. (3.79), $\mathcal{L}_\phi(f_0(x)) = \frac{\partial \phi}{\partial x} f_0(x) = L_{f_0} \phi(x)$.

Let us consider two possible initial sets of observables, i.e.,

$$\Phi_{\text{in}}^{(1)} = \{x_1,\ x_2,\ x_3\},$$
$$\Phi_{\text{in}}^{(2)} = \{x_1,\ x_2,\ x_3,\ x_1 x_3\},$$

where $\Phi_{\text{in}}^{(1)}$ comprises only the system states, while $\Phi_{\text{in}}^{(2)}$ contains an additional observable of interest ($x_1 x_3$).

Let us focus first on $\Phi_{\text{in}}^{(1)}$. To generate the whole basis $\Phi^{(1)} = \Phi$ from $\Phi_{\text{in}}^{(1)}$, we iteratively apply Algorithm 3.1, for each observable $\phi_i \in \Phi$, $i = 1, \dots, |\Phi|$:

- At step $i = 1$, $\phi_1(x) = x_1$:

  $$\mathcal{L}_{\phi_1}(f_0(x)) = L_{f_0}\phi_1(x) = \frac{\partial \phi_1}{\partial x} f_0(x) = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ -x_2 + x_1^2 \\ x_3 - x_1 x_2 + x_1 \end{bmatrix} = x_1 = \phi_1(x).$$

  This step yields no residual, so no new observables are added to the basis.

- At step $i = 2$, $\phi_2(x) = x_2$:

  $$\mathcal{L}_{\phi_2}(f_0(x)) = \frac{\partial \phi_2}{\partial x} f_0(x) = \begin{bmatrix} 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ -x_2 + x_1^2 \\ x_3 - x_1 x_2 + x_1 \end{bmatrix} = -x_2 + x_1^2 = -\phi_2(x) + x_1^2.$$

  This step yields the residual $\alpha_{\text{res}}(x) = x_1^2$, from which we can extract $\Psi_\alpha = \{x_1^2\}$. Then, by introducing the new observable $\phi_4(x) = x_1^2$, we can rewrite $\mathcal{L}_{\phi_2}(f_0(x)) = -\phi_2(x) + \phi_4(x)$ with no residual.

- At steps $i \geq 3$:

$$\phi_3(x) = x_3, \quad \mathcal{L}_{\phi_3}(f_0(x)) = \phi_3(x) - x_1x_2 + \phi_1(x)$$
$$\Rightarrow \phi_5(x) = x_1x_2,$$

$$\phi_4(x) = x_1^2, \quad \mathcal{L}_{\phi_4}(f_0(x)) = \begin{bmatrix} 2x_1 & 0 & 0 \end{bmatrix} f_0(x) = 2\phi_4(x),$$

$$\phi_5(x) = x_1x_2, \quad \mathcal{L}_{\phi_5}(f_0(x)) = \begin{bmatrix} x_2 & x_1 & 0 \end{bmatrix} f_0(x) = x_1^3$$
$$\Rightarrow \phi_6(x) = x_1^3,$$

$$\phi_6(x) = x_1^3, \quad \mathcal{L}_{\phi_6}(f_0(x)) = \begin{bmatrix} 3x_1^2 & 0 & 0 \end{bmatrix} f_0(x) = 3\phi_6(x).$$

We observe that the algorithm has converged, since it reached the last observable of the basis with no further residual.

Then, system (3.82) admits a finite-dimensional basis of 6 observables, i.e.,

$$\Phi = \left\{ x_1,\ x_2,\ x_3,\ x_1^2,\ x_1x_2,\ x_1^3 \right\}.$$

Now that we have the basis Φ, we employ Algorithm 3.2 to construct the Koopman lifted system. First, we compute the matrix A, obtaining

$$A = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & -1 & 0 \\ 0 & 0 & 0 & 2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 3 \end{bmatrix}.$$

Second, we compute the term $\mathcal{B}_x(x,u)$, obtaining

$$\mathcal{B}_x(x,u) = \frac{\partial \phi}{\partial x} g(x,u) = \begin{bmatrix} x_2 \sin(u_1) \\ u_1u_2 \\ x_1e^{u_2} - x_1 \\ 2x_1x_2 \sin(u_1) \\ x_2^2 \sin(u_1) + x_1u_1u_2 \\ 3x_1^2x_2 \sin(u_1) \end{bmatrix}.$$

Now, we have to find $\mathcal{B}(z,u)$ such that the equality $\mathcal{B}(\phi(x),u) = \mathcal{B}_x(x,u)$ holds. By inspection of $\mathcal{B}_x$, one possible choice is given by

$$\mathcal{B}_x(x,u) = \mathcal{B}(\phi(x),u) = \begin{bmatrix} \phi_2(x) \sin(u_1) \\ u_1u_2 \\ \phi_1(x)e^{u_2} - \phi_1(x) \\ 2\phi_5(x) \sin(u_1) \\ \phi_2^2(x) \sin(u_1) + \phi_1(x)u_1u_2 \\ 3\phi_1(x)\phi_5(x) \sin(u_1) \end{bmatrix}$$
$$\Rightarrow \mathcal{B}(z,u) = \begin{bmatrix} z_2 \sin(u_1) \\ u_1u_2 \\ z_1e^{u_2} - z_1 \\ 2z_5 \sin(u_1) \\ z_2^2 \sin(u_1) + z_1u_1u_2 \\ 3z_1z_5 \sin(u_1) \end{bmatrix}.$$

Finally, since $g(x,0) = 0$ by construction, we can apply Lemma 3.1 to obtain the Koopman lifted system in LPV form. Specifically, we compute the term $B(z,u)$ as

$$B(z,u) = \int_0^1 \frac{\partial \mathcal{B}}{\partial u}(z, \lambda u) d\lambda = \begin{bmatrix} \frac{1}{u_1} z_2 \sin(u_1) & 0 \\ \frac{1}{2} u_2 & \frac{1}{2} u_1 \\ 0 & \frac{1}{u_2} z_1 (e^{u_2} - 1) \\ \frac{2}{u_1} z_5 \sin(u_1) & 0 \\ \frac{1}{u_1} z_2^2 \sin(u_1) + \frac{1}{2} z_1 u_2 & \frac{1}{2} z_1 u_1 \\ \frac{3}{u_1} z_1 z_5 \sin(u_1) & 0 \end{bmatrix}.$$

Combining the above terms, we obtain the Koopman lifted system $\dot{z} = Az + \mathcal{B}(z,u) = Az + B(z,u)u$.

Recalling now the second initial set $\Phi_{\text{in}}^{(2)}$, we just report that, also in this case, Algorithm 3.1 converges, generating a finite-dimensional basis of 9 observables, i.e.,

$$\Phi^{(2)} = \left\{x_1,\ x_2,\ x_3,\ x_1x_3,\ x_1^2,\ x_1x_2,\ x_1^2x_2,\ x_1^3,\ x_1^4\right\}.$$

Example 3.2 Consider the following CT input-affine system, representing a kinematic unicycle model:

$$\dot{x} = \begin{bmatrix} x_3\cos(x_4) \\ x_3\sin(x_4) \\ u_1 \\ u_2 \end{bmatrix} = \underbrace{\begin{bmatrix} x_3\cos(x_4) \\ x_3\sin(x_4) \\ 0 \\ 0 \end{bmatrix}}_{f_0(x)} + \underbrace{\begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix}}_{g_1(x)} u_1 + \underbrace{\begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}}_{g_2(x)} u_2. \tag{3.85}$$

We apply Algorithm 3.3 to generate a basis Φ for system (3.85) satisfying Assumption 3.3. Being in CT, we have that, by Eq. (3.79), $\mathcal{L}_\phi(f_0(x)) = \frac{\partial\phi}{\partial x} f_0(x) = L_{f_0}\phi(x)$ and $\mathcal{L}_\phi(g_j(x)) = \frac{\partial\phi}{\partial x} g_j(x) = L_{g_j}\phi(x)$, $j = 1,\dots,n_u$.

Let us consider two possible initial sets of observables, i.e.,

$$\Phi_{\text{in}}^{(1)} = \{x_1,\ x_2,\ x_3,\ x_4\},$$
$$\Phi_{\text{in}}^{(2)} = \{x_1,\ x_2,\ x_3,\ x_4,\ x_1^2,\ x_2^2\},$$

where $\Phi_{\text{in}}^{(1)}$ comprises only the system states, while $\Phi_{\text{in}}^{(2)}$ contains additional observables of interest $(x_1^2,\ x_2^2)$.

Let us focus first on $\Phi_{\text{in}}^{(1)}$. To generate the whole basis $\Phi^{(1)} = \Phi$ from $\Phi_{\text{in}}^{(1)}$, we iteratively apply Algorithm 3.3, for each observable $\phi_i \in \Phi$, $i = 1,\dots,|\Phi|$:

$\phi_1(x) = x_1, \quad \mathcal{L}_{\phi_1}(f_0(x)) = x_3\cos(x_4), \quad \mathcal{L}_{\phi_1}(g_1(x)) = 0, \quad \mathcal{L}_{\phi_1}(g_2(x)) = 0$
$\Rightarrow \phi_5(x) = x_3\cos(x_4),$

$\phi_2(x) = x_2, \quad \mathcal{L}_{\phi_2}(f_0(x)) = x_3\sin(x_4), \quad \mathcal{L}_{\phi_2}(g_1(x)) = 0, \quad \mathcal{L}_{\phi_2}(g_2(x)) = 0$
$\Rightarrow \phi_6(x) = x_3\sin(x_4),$

$\phi_3(x) = x_3, \quad \mathcal{L}_{\phi_3}(f_0(x)) = 0, \quad \mathcal{L}_{\phi_3}(g_1(x)) = 1, \quad \mathcal{L}_{\phi_3}(g_2(x)) = 0,$

$\phi_4(x) = x_4, \quad \mathcal{L}_{\phi_4}(f_0(x)) = 0, \quad \mathcal{L}_{\phi_4}(g_1(x)) = 0, \quad \mathcal{L}_{\phi_4}(g_2(x)) = 1,$

$\phi_5(x) = x_3\cos(x_4), \quad \mathcal{L}_{\phi_5}(f_0(x)) = 0, \quad \mathcal{L}_{\phi_5}(g_1(x)) = \cos(x_4), \quad \mathcal{L}_{\phi_5}(g_2(x)) = -\phi_6(x)$
$\Rightarrow \phi_7(x) = \cos(x_4),$

$\phi_6(x) = x_3\sin(x_4), \quad \mathcal{L}_{\phi_6}(f_0(x)) = 0, \quad \mathcal{L}_{\phi_6}(g_1(x)) = \sin(x_4), \quad \mathcal{L}_{\phi_6}(g_2(x)) = \phi_5(x)$
$\Rightarrow \phi_8(x) = \sin(x_4),$

$\phi_7(x) = \cos(x_4), \quad \mathcal{L}_{\phi_7}(f_0(x)) = 0, \quad \mathcal{L}_{\phi_7}(g_1(x)) = 0, \quad \mathcal{L}_{\phi_7}(g_2(x)) = -\phi_8(x),$

$$\phi_8(x) = \sin(x_4), \quad \mathcal{L}_{\phi_8}(f_0(x)) = 0, \quad \mathcal{L}_{\phi_8}(g_1(x)) = 0, \quad \mathcal{L}_{\phi_8}(g_2(x)) = \phi_7(x).$$

Then, system (3.85) admits a finite-dimensional basis of 8 observables, i.e.,

$$\Phi = \{x_1,\ x_2,\ x_3,\ x_4,\ x_3\cos(x_4),\ x_3\sin(x_4),\ \cos(x_4),\ \sin(x_4)\}.$$

Now, we employ Algorithm 3.4 to construct the Koopman lifted system, obtaining

$$\dot{z} = Az + B_0 u + B_1 z u_1 + B_2 z u_2,$$

where

$$A = \begin{bmatrix} 0&0&0&0&1&0&0&0\\ 0&0&0&0&0&1&0&0\\ 0&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&0 \end{bmatrix}, \quad B_0 = \begin{bmatrix} 0&0\\ 0&0\\ 1&0\\ 0&1\\ 0&0\\ 0&0\\ 0&0\\ 0&0 \end{bmatrix}, \quad B_1 = \begin{bmatrix} 0&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&1&0\\ 0&0&0&0&0&0&0&1\\ 0&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&0 \end{bmatrix}, \quad B_2 = \begin{bmatrix} 0&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&0\\ 0&0&0&0&0&-1&0&0\\ 0&0&0&0&1&0&0&0\\ 0&0&0&0&0&0&0&-1\\ 0&0&0&0&0&0&1&0 \end{bmatrix}.$$

We can observe that the resulting Koopman lifted system matches the structure in Eq. (3.43), since system (3.85) contains an input-linear term.

Recalling now the second initial set $\Phi_{\text{in}}^{(2)}$, we just report that, also in this case, Algorithm 3.3 converges, generating a finite-dimensional basis of 27 observables, i.e.,

$$\begin{aligned}\Phi^{(2)} = \{&x_1,\ x_2,\ x_3,\ x_4,\ x_1^2,\ x_2^2,\ x_3\cos(x_4),\ x_3\sin(x_4),\ x_1x_3\cos(x_4),\ x_2x_3\sin(x_4),\\ &\cos(x_4),\ \sin(x_4),\ x_3^2\cos^2(x_4),\ x_1x_3\sin(x_4),\ x_1\cos(x_4),\ x_3^2\sin^2(x_4),\ x_2x_3\cos(x_4),\\ &x_2\sin(x_4),\ x_3\cos^2(x_4),\ x_3^2\cos(x_4)\sin(x_4),\ x_1\sin(x_4),\ x_3\sin^2(x_4),\ x_2\cos(x_4),\\ &\cos^2(x_4),\ x_3\cos(x_4)\sin(x_4),\ \sin^2(x_4),\ \cos(x_4)\sin(x_4)\}.\end{aligned}$$

Example 3.3 Consider the following DT system with inputs:

$$x_{k+1} = \begin{bmatrix} 2x_{1,k} + u_k \\ x_{2,k} - x_{1,k}^3 + x_{1,k}x_{2,k}u_k \end{bmatrix} = \underbrace{\begin{bmatrix} 2x_{1,k} \\ x_{2,k} - x_{1,k}^3 \end{bmatrix}}_{f_0(x_k)} + \underbrace{\begin{bmatrix} u_k \\ x_{1,k}x_{2,k}u_k \end{bmatrix}}_{g(x_k,u_k)}. \tag{3.88}$$

We apply Algorithm 3.1 to generate a basis Φ for system (3.88) satisfies Assumption 3.5b. Being in DT, we have that, by Eq. (3.79), $\mathcal{L}_\phi(f_0(x)) = \phi(f_0(x))$.

Let us consider, as initial set of observables, $\Phi_{\text{in}} = \{x_1, x_2\}$, i.e., only comprising the system states.

To generate the whole basis Φ from Φ_{in}, we iteratively apply Algorithm 3.1, for each observable $\phi_i \in \Phi$, $i = 1, \ldots, |\Phi|$:

$$\phi_1(x) = x_1, \quad \mathcal{L}_{\phi_1}(f_0(x_k)) = \phi_1(f_0(x_k)) = 2x_{1,k} = 2\phi_1(x_k),$$

$$\begin{aligned}&\phi_2(x) = x_2, \quad \mathcal{L}_{\phi_2}(f_0(x_k)) = \phi_2(f_0(x_k)) = x_{2,k} - x_{1,k}^3 = \phi_2(x_k) - x_{1,k}^3\\ &\Rightarrow \phi_3(x) = x_1^3,\end{aligned}$$

$$\phi_3(x) = x_1^3, \quad \mathcal{L}_{\phi_3}(f_0(x_k)) = \phi_3(f_0(x_k)) = 8x_{1,k}^3 = 8\phi_3(x_k).$$

Then, system (3.88) admits a finite-dimensional basis of 3 observables, i.e.,

$$\Phi = \left\{x_1,\ x_2,\ x_1^3\right\}.$$

Now, we employ Algorithm 3.2 to construct the Koopman lifted system, obtaining

$$z_{k+1} = \underbrace{\begin{bmatrix} 2 & 0 & 0 \\ 0 & 1 & -1 \\ 0 & 0 & 8 \end{bmatrix}}_{A} z_k + \underbrace{\begin{bmatrix} 1 \\ z_{1,k} z_{2,k} \\ 12 z_{1,k}^2 + 6 z_{1,k} u_k + u_k^2 \end{bmatrix}}_{B(z_k, u_k)} u_k.$$

We can observe that, while the original system in Eq. (3.88) is input-affine, the resulting Koopman lifted system is, instead, nonlinear in both state and input, as highlighted in Remark 3.8.

3.4. Dimensionality Reduction of the Koopman Lifted System

As noted in Remarks 3.6 and 3.9, an infinite-dimensional basis of Koopman observables may be needed to exactly lift systems (3.16) and (3.63) – respectively, continuous-time and discrete-time systems with inputs.

In this section, we provide a general approach to reduce the basis of observables and the related Koopman lifted systems (3.24), (3.70) to a finite dimension of choice.

Let us set a-priori the maximum number of observables $\overline{N}_o < N_o$ (where N_o may be infinite) to be generated by Algorithm 3.1. By setting $\overline{N}_o$, the lifted equations associated with the lastly generated observables will contain a residual nonlinear term.

In detail, let us split the observables as follows:

$$\phi(x) = \begin{bmatrix} \phi'(x) \\ \phi''(x) \end{bmatrix},$$

where $\phi' : \mathbb{R}^{n_x} \to \mathbb{R}^{n'_z}$, $\phi'' : \mathbb{R}^{n_x} \to \mathbb{R}^{n''_z}$, and $n_z = \overline{N}_o = n'_z + n''_z$.

With the reduced number of observables, the lifted dynamics is the following:

$$\phi(x)^+ = \begin{bmatrix} \phi'(x)^+ \\ \phi''(x)^+ \end{bmatrix} = \underbrace{\begin{bmatrix} A' \\ A'' \end{bmatrix}}_{A} \phi(x) + \underbrace{\begin{bmatrix} \mathcal{B}'_x(x,u) \\ \mathcal{B}''_x(x,u) \end{bmatrix}}_{\mathcal{B}_x(x,u)} + \begin{bmatrix} \mathbf{0}_{n'_z} \\ f_{\text{res},x}(x,u) \end{bmatrix}, \tag{3.90}$$

where $f_{\text{res},x} : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \to \mathbb{R}^{n''_z}$ is the residual nonlinear term containing the dynamics of the unlifted part of the system.

Similarly to $\mathcal{B}_x$, by enforcing the equality $f_{\text{res}}(z,u) = f_{\text{res},x}(\phi(x),u)$, we can express the residual in terms of the lifted state z, allowing us to obtain the reduced Koopman lifted system, i.e.,

$$z^+ = \begin{bmatrix} z'^+ \\ z''^+ \end{bmatrix} = Az + \underbrace{\begin{bmatrix} \mathcal{B}'(z,u) \\ \mathcal{B}''(z,u) \end{bmatrix}}_{\mathcal{B}(z,u)} + \begin{bmatrix} \mathbf{0}_{n'_z} \\ f_{\text{res}}(z,u) \end{bmatrix}. \tag{3.91}$$

Remark 3.11

Let us consider:

- the solution $x(t)$ (or x_k) of the original system (3.16) (or (3.63));

- the solution $z^*(t)$ (or z_k^*) of the exactly-lifted system (3.24) (or (3.70)), with infinite-dimensional basis of observables $\phi^*(x)$;
- the solution $z(t)$ (or z_k) of the reduced system (3.91), with finite-dimensional basis $\phi(x)$.

Then, it holds by construction that

$$\begin{aligned} x(t) = \phi^{-1}(z(t)) &= \phi^{*-1}(z^*(t)), \quad \forall\, t \geq 0, \quad \text{or} \\ x_k = \phi^{-1}(z_k) \quad &= \phi^{*-1}(z_k^*), \qquad \forall\, k \geq 0, \end{aligned} \tag{3.92}$$

meaning that the three solutions are equivalent, even if ϕ has a finite dimension compared to ϕ^*.

Linear/Affine Parameter-Varying Form

The reduced Koopman lifted system (3.91) can be expressed in LPV or APV form. Its general expression is given by

$$z^+ = \begin{bmatrix} z'^+ \\ z''^+ \end{bmatrix} = \underbrace{\begin{bmatrix} \tilde{A}'(\overline{z},\overline{u}) \\ \tilde{A}''(\overline{z},\overline{u}) \end{bmatrix}}_{\tilde{A}(\overline{z},\overline{u})} z + \underbrace{\begin{bmatrix} \tilde{B}'(\overline{z},\overline{u}) \\ \tilde{B}''(\overline{z},\overline{u}) \end{bmatrix}}_{\tilde{B}(\overline{z},\overline{u})} u + \underbrace{\begin{bmatrix} \tilde{b}'(\overline{z},\overline{u}) \\ \tilde{b}''(\overline{z},\overline{u}) \end{bmatrix}}_{\tilde{b}(\overline{z},\overline{u})}, \tag{3.93}$$

and depends on whether $\mathcal{B}(z,u)$ and $f_{\text{res}}(z,u)$ can be factorized through Lemma 3.1.

Let us start from the first part of the dynamics, i.e., $z'^+ = \tilde{A}'(\overline{z},\overline{u})z + \tilde{B}'(\overline{z},\overline{u})u + \tilde{b}'(\overline{z},\overline{u})$, which does not depend on f_{res}. Two cases arise:

(a) $\mathcal{B}'(z,0) = 0$, i.e., we can factorize $\mathcal{B}'$ through Lemma 3.1 as

$$\mathcal{B}'(z,u) = B'(z,u)u,$$

(b) $\mathcal{B}'(z,0) \neq 0$, i.e., we consider the first-order Taylor expansion of $\mathcal{B}'$ around $(\overline{z},\overline{u})$,

$$\mathcal{B}'(z,u) \approx \mathcal{B}'(\overline{z},\overline{u}) + \frac{\partial \mathcal{B}'}{\partial z}(\overline{z},\overline{u})(z-\overline{z}) + \frac{\partial \mathcal{B}'}{\partial u}(\overline{z},\overline{u})(u-\overline{u}).$$

Then, for each of the above cases, we have the following expressions:

	(a)	(b)
$\tilde{A}'(\overline{z},\overline{u})$	A'	$A' + \frac{\partial \mathcal{B}'}{\partial z}(\overline{z},\overline{u})$
$\tilde{B}'(\overline{z},\overline{u})$	$B'(\overline{z},\overline{u})$	$\frac{\partial \mathcal{B}'}{\partial u}(\overline{z},\overline{u})$
$\tilde{b}'(\overline{z},\overline{u})$	$\mathbf{0}_{n'_z}$	$\mathcal{B}'(\overline{z},\overline{u}) - \frac{\partial \mathcal{B}'}{\partial z}(\overline{z},\overline{u})\overline{z} - \frac{\partial \mathcal{B}'}{\partial u}(\overline{z},\overline{u})\overline{u}$

Now, let us consider the second part of the dynamics, i.e., $z''^+ = \tilde{A}''(\overline{z},\overline{u})z + \tilde{B}''(\overline{z},\overline{u})u + \tilde{b}''(\overline{z},\overline{u})$, which depends on f_{res}. Four cases arise:

(a) $\mathcal{B}''(z,0) = 0$ and $f_{\text{res}}(z,0) = 0$, i.e., concerning f_{res}, we can factorize it through Lemma 3.1 as

$$f_{\text{res}}(z,u) = \varphi_{\text{res}}(z,u)u,$$

(b) $\mathcal{B}''(z,0) = 0$ and $f_{\text{res}}(z,0) \neq 0$, i.e., concerning f_{res}, we consider its first-order Taylor expansion around $(\overline{z},\overline{u})$, i.e.,

$$f_{\text{res}}(z,u) \approx f_{\text{res}}(\overline{z},\overline{u}) + \frac{\partial f_{\text{res}}}{\partial z}(\overline{z},\overline{u})(z-\overline{z}) + \frac{\partial f_{\text{res}}}{\partial u}(\overline{z},\overline{u})(u-\overline{u}),$$

(c) $\mathcal{B}''(z,0) \neq 0$ and $f_{\text{res}}(z,0) = 0$;

(d) $\mathcal{B}''(z,0) \neq 0$ and $f_{\text{res}}(z,0) \neq 0$.

Then, for each of the above cases, we have the following expressions:

	(a)	(b)
$\tilde{A}''(\overline{z},\overline{u})$	A''	$A'' + \frac{\partial f_{\text{res}}}{\partial z}(\overline{z},\overline{u})$
$\tilde{B}''(\overline{z},\overline{u})$	$B''(z,u) + \varphi_{\text{res}}(z,u)$	$B''(z,u) + \frac{\partial f_{\text{res}}}{\partial u}(\overline{z},\overline{u})$
$\tilde{b}''(\overline{z},\overline{u})$	$\mathbf{0}_{n_z''}$	$f_{\text{res}}(\overline{z},\overline{u}) - \frac{\partial f_{\text{res}}}{\partial z}(\overline{z},\overline{u})\overline{z} - \frac{\partial f_{\text{res}}}{\partial u}(\overline{z},\overline{u})\overline{u}$

	(c)	(d)
$\tilde{A}''(\overline{z},\overline{u})$	$A'' + \frac{\partial \mathcal{B}''}{\partial z}(\overline{z},\overline{u})$	$A'' + \frac{\partial \mathcal{B}''}{\partial z}(\overline{z},\overline{u}) + \frac{\partial f_{\text{res}}}{\partial z}(\overline{z},\overline{u})$
$\tilde{B}''(\overline{z},\overline{u})$	$\frac{\partial \mathcal{B}''}{\partial u}(\overline{z},\overline{u}) + \varphi_{\text{res}}(z,u)$	$\frac{\partial \mathcal{B}''}{\partial u}(\overline{z},\overline{u}) + \frac{\partial f_{\text{res}}}{\partial u}(\overline{z},\overline{u})$
$\tilde{b}''(\overline{z},\overline{u})$	$\mathcal{B}''(\overline{z},\overline{u}) - \frac{\partial \mathcal{B}''}{\partial z}(\overline{z},\overline{u})\overline{z} - \frac{\partial \mathcal{B}''}{\partial u}(\overline{z},\overline{u})\overline{u}$	$\mathcal{B}''(\overline{z},\overline{u}) - \frac{\partial \mathcal{B}''}{\partial z}(\overline{z},\overline{u})\overline{z} - \frac{\partial \mathcal{B}''}{\partial u}(\overline{z},\overline{u})\overline{u} +$ $f_{\text{res}}(\overline{z},\overline{u}) - \frac{\partial f_{\text{res}}}{\partial z}(\overline{z},\overline{u})\overline{z} - \frac{\partial f_{\text{res}}}{\partial u}(\overline{z},\overline{u})\overline{u}$

3.4.1. When to Truncate the Basis of Observables

After having introduced the dimensionality reduction procedure, a key question that remains to be addressed is when should we truncate the basis of observables Φ, i.e., which is the "best" dimension reduction for the Koopman lifted system.

Here, we propose a possible approach based on the empirical evaluation of the prediction accuracy of the reduced Koopman lifted system for increasing number of observables ϕ. Without loss of generality, we consider in the following the CT case only.

We start by selecting a family of input signals $\{u^p(t)\}_{p\in\mathcal{P}}$, parameterized by the parameter p which can take values in the set $\mathcal{P}$.

Taking an input signal $u(t) \in \{u^p(t)\}_{p\in\mathcal{P}}$, we use it to generate a trajectory $x(t)$ of the original system system (3.18). Hence, the family of input signals must be designed to ensure a sufficiently rich "excitation" of the system, yielding representative state trajectories for evaluating the prediction accuracy. This point is better clarified in Section 3.6, when dealing with the actual case studies employed to validate our approach, i.e., mobile robot navigation in cluttered environments (Section 3.6.1, p. 97) and autonomous parallel parking (Section 3.6.2, p. 110).

The trajectory $x(t)$ is then sampled at N_{pred} equally-spaced time instants $t_n = \frac{n}{N_{\text{pred}}}T$, extracting the states $x_n = x(t_n)$, $n = 1,\dots,N_{\text{pred}}$. Taking each lifted state $z_n = \phi(x_n)$ as an initial condition, a prediction $z^{(n)}_{\text{pred}}(\tau)$ of the reduced lifted system is computed, over the prediction time interval $\tau \in [0, T_p]$, applying the input $u(t)$.

Let us write the basis of observables as $\Phi = \Phi_{\text{in}} \cup \Psi$, where $\Phi_{\text{in}} = \{x_i\}_{i=1}^{n_x} \cup \{\psi_{\text{in},i}(x)\}_{i=1}^{N_{\text{in}}}$ is the initial set of observables, $\{\psi_{\text{in},i}(x)\}_{i=1}^{N_{\text{in}}}$ are hand-picked observables of interest, and $\Psi = \{\psi_i(x)\}_{i=1}^{N_\psi}$ are the observables generated by Algorithm 3.1. Then, let us rewrite the

lifted state as $z = [z_x^\top,\ z_{\text{in}}^\top,\ z_\psi^\top]^\top$, where z_x, z_{in}, and z_ψ are the lifted states related to the observables x, $\psi_{\text{in}}(x)$, and $\psi(x)$, respectively.

The lifted trajectory $z(t) = \phi(x(t))$ and the predictions $z_{\text{pred}}(t)$ are then expressed as

$$z(t) = \phi(x(t)) = \begin{bmatrix} z_x(t) \\ z_{\text{in}}(t) \\ z_\psi(t) \end{bmatrix}, \quad z_{\text{pred}}(t) = \begin{bmatrix} z_{x,\text{pred}}(t) \\ z_{\text{in,pred}}(t) \\ z_{\psi,\text{pred}}(t) \end{bmatrix}. \tag{3.94}$$

The true and predicted trajectories (3.94) are compared by sampling each of them with sampling time T_s. Specifically, each prediction time interval will contain $N_p + 1$ sampling points, with $N_p = \frac{T_p}{T_s}$.

For each n-th predicted trajectory, three prediction errors are considered:

1) Prediction error $\epsilon_{\text{pred},1}$ between the original trajectory $x(t)$ and the predicted lifted states $z_{x,\text{pred}}(t)$, associated with the original states x, i.e.,

$$\epsilon_{\text{pred},1}^{(n)} = \frac{1}{N_p} \sum_{i=1}^{N_p} \left\| z_{x,\text{pred}}^{(n)}(iT_s) - x(t_n + iT_s) \right\|_\infty, \tag{3.95a}$$

2) Prediction error $\epsilon_{\text{pred},2}$ between the true lifted states $z_{\text{in}}(t)$ and the predicted lifted states $z_{\text{in,pred}}(t)$, associated with the hand-picked observables of interest $\psi_{\text{in}}(x)$, i.e.,

$$\epsilon_{\text{pred},2}^{(n)} = \frac{1}{N_p} \sum_{i=1}^{N_p} \left\| z_{\text{in,pred}}^{(n)}(iT_s) - z_{\text{in}}(t_n + iT_s) \right\|_\infty, \tag{3.95b}$$

3) Prediction error $\epsilon_{\text{pred},3}$, combining the previous ones, i.e.,

$$\epsilon_{\text{pred},3}^{(n)} = \frac{1}{N_p} \sum_{i=1}^{N_p} \left\| \begin{bmatrix} z_{x,\text{pred}}^{(n)}(iT_s) \\ z_{\text{in,pred}}^{(n)}(iT_s) \end{bmatrix} - \begin{bmatrix} x(t_n+iT_s) \\ z_{\text{in}}(t_n+iT_s) \end{bmatrix} \right\|_\infty. \tag{3.95c}$$

The prediction errors (3.95a)-(3.95c) quantify the average per-sample deviation, in the ∞-norm, between true and predicted trajectory. This design choice was adopted as it yielded consistent results during validation (see Section 3.6). Nevertheless, other definitions for the prediction error are equally admissible.

It is worth noticing that, in the above prediction errors, we are not including $z_\psi(t)$. The reason is that z_ψ are the lifted states generated by Algorithm 3.1, which naturally arise from the initial set of observables Φ_{in} in order to construct the Koopman lifted system. Therefore, in most of the cases, the observables $\psi(x)$ and the corresponding lifted states z_ψ do not have a readily interpretable "physical" meaning, on the contrary of the original system states x and the hand-picked observables $\psi_{\text{in}}(x)$. Therefore, in most cases, it is more informative to evaluate the prediction error on $z_x(t)$ and $z_{\text{in}}(t)$ only.

All the above calculations are repeated for increasing number of observables $\phi(x) = [\phi_i(x)]_{i=1}^{\overline{N}_o}$, with $\overline{N}_o \geq |\Phi_{\text{in}}|$, and for multiple trajectories $x(t)$, obtained by, e.g., randomly selecting multiple values of the parameter $p \in \mathcal{P}$ of the input signal $u^p(t)$.

After having collected all the prediction error data $\epsilon_{\text{pred},1}$, $\epsilon_{\text{pred},2}$, and $\epsilon_{\text{pred},3}$ from Eqs. (3.95a)-(3.95c), we determine the "best" dimension for the reduced lifted system by evaluating how many observables are required before the average values of $\epsilon_{\text{pred},1}$, $\epsilon_{\text{pred},2}$, and $\epsilon_{\text{pred},3}$ decrease to a sufficiently low values.

3.5. Koopman NMPC: Lifting Nonlinear MPC in the Koopman Space

After having introduced the Koopman operator framework for dynamical systems, in this section we extend its application to optimization problems, with specific focus to nonlinear programs (NLPs) that arise from Nonlinear Model Predictive Control (NMPC) problems.

The key idea is to employ the Koopman operator to transform a general NMPC problem, originally formulated as a NLP in the state x, into a quadratic program (QP) in the lifted state z. The obtained QP-MPC problem will hereafter be referred to as *Koopman NMPC* (K-NMPC) [31].

Being a QP, K-NMPC is considerably more tractable than the original NLP and can be solved with superior computational performance [35].

3.5.1. Nonlinear MPC Formulation

In the following, let us consider a CT input-affine system, matching Eq. (3.32), acting as a plant to control, i.e.,

$$\dot{x} = f(x,u) = f_0(x) + \sum_{j=1}^{n_u} g_j(x)u_j, \quad x(0) = x_0. \tag{3.96}$$

Assuming an input-affine structure for the plant is not a restrictive choice, as it holds for many systems in practical applications. Moreover, this case is the most interesting one from the Koopman operator perspective, since, as highlighted in Section 3.2.2, the related lifted system is bilinear – unlike the general CT case, where the input-dependent term retains a general nonlinearity.

From the CT plant (3.96), we construct a DT input-affine prediction model, by applying a discretization method of choice (refer to Section 2.2, p. 12) with discrete time step $T_s > 0$, i.e.,

$$x_{k+1} = f_d(x_k, u_k). \tag{3.97}$$

Nonlinear MPC (NMPC) provides, at each discrete time instant $k \geq 0$, an optimal control input u_k^*, obtained by solving the following NLP optimal control problem:

$$\begin{aligned}
\min_{\hat{\boldsymbol{x}},\hat{\boldsymbol{u}}} \quad & J(\hat{\boldsymbol{x}},\hat{\boldsymbol{u}}) = \sum_{i=0}^{N_p-1} \|\hat{x}_i - x_{r,k+i}\|_Q^2 + \|\hat{u}_i\|_R^2 \\
& \qquad + \sum_{i=1}^{N_p-1} \|\hat{x}_i - \hat{x}_{i-1}\|_{Q_\Delta}^2 + \|\hat{u}_i - \hat{u}_{i-1}\|_{R_\Delta}^2 && \text{(3.98a)} \\
\text{s.t.} \quad & \hat{x}_0 = x_k, && \text{(3.98b)} \\
& \hat{x}_{i+1} = f_d(\hat{x}_i, \hat{u}_i), \quad i = 0,\dots,N_p-1, && \text{(3.98c)} \\
& \hat{u}_i \in \mathcal{U}, \quad i = 0,\dots,N_p-1, && \text{(3.98d)} \\
& \hat{x}_i \in \mathcal{X}, \quad i = 0,\dots,N_p-1, && \text{(3.98e)} \\
& h(\hat{x}_i) \leq 0, \quad i = 0,\dots,N_p-1. && \text{(3.98f)}
\end{aligned}$$

The NMPC problem (3.98) is formulated to attain tracking towards the admissible state trajectory $\boldsymbol{x}_r = (x_{r,k})_{k\geq 0}$ of the discretized system (3.97).

The cost function J in Eq. (3.98a) is composed of four terms:

- The quadratic tracking term $\|\hat{x}_i - x_{r,k+i}\|_Q^2$ for the predicted state, towards the reference state $x_{r,k+i}$ of the trajectory $\boldsymbol{x}_r$, with $Q \succeq 0$.
- The quadratic term $\|\hat{u}_i\|_R^2$, with $R \succ 0$, penalizing the input magnitude, in which we did not include the reference input $u_{r,k+i}$ of the admissible sequence $\boldsymbol{u}_r$, which generates $\boldsymbol{x}_r$ through Eq. (3.97). The inclusion of such a term is a common practice in NMPC (see, e.g., [32, 33]), in those cases where only the state trajectory $\boldsymbol{x}_r$ is readily available, preserving the strict convexity of the cost function J.
- The quadratic terms $\|\hat{x}_i - \hat{x}_{i-1}\|_{Q_\Delta}^2$ and $\|\hat{u}_i - \hat{u}_{i-1}\|_{R_\Delta}^2$, with $Q_\Delta \succeq 0$ and $R_\Delta \succeq 0$, which penalize the variation in time of the predicted states and inputs, to obtain smoother predicted trajectories [32, 33].

The prediction model (3.98c) is given by Eq. (3.97). Eqs. (3.98d) and (3.98e) report linear input and state constraints, given by the convex polytopic sets $\mathcal{U} = \{u \in \mathbb{R}^{n_u} : H_u u \leq h_u\}$ and $\mathcal{X} = \{x \in \mathbb{R}^{n_x} : H_x x \leq h_x\}$. Eq. (3.98f) reports nonlinear state constraints, given by the nonlinear function $h : \mathbb{R}^{n_x} \to \mathbb{R}^{N_h}$.

Lifting the NMPC Problem in the Koopman space

To lift the NMPC problem (3.98) in the Koopman space, three main steps are required:

1) generate with Algorithm 3.3 a basis of observables Φ (3.4) for the CT plant (3.96) and the nonlinear state constraints (3.98f), obtaining a bilinear Koopman lifted system (3.35);
2) convert the NMPC problem (3.98) from a NLP in the original state x into a QP in the lifted state z, obtaining the K-NMPC problem;
3) reduce the dimension of the lifted state z, in order to either reduce the K-NMPC complexity or render tractable the case of an infinite-dimensional Koopman lifted system (3.35).

3.5.2. Observables Generation and Lifted System

To obtain the basis Φ, we first need to specify the set of observables Φ_{in} to initialize Algorithm 3.3. Since Φ should allow to lift both the CT plant (3.96) and the NMPC nonlinear state constraints (3.98f), we begin by expanding the nonlinear function $h(x)$ in Eq. (3.98f) as follows:

$$h(x) = C'x + \sum_{i=1}^{N_\sigma} c_i'' \, \sigma_i(x) - d = C \begin{bmatrix} x \\ \sigma(x) \end{bmatrix} - d, \tag{3.99}$$

where $\sigma_i : \mathbb{R}^{n_x} \to \mathbb{R}$ are linearly-combined elementary nonlinear terms, with coefficients $c_i \in \mathbb{R}^{N_h}$, $i = 1, \ldots, N_\sigma$, $d \in \mathbb{R}^{N_h}$ is a constant affine term, and $C'x$ is a possible linear term composing h, with $C' \in \mathbb{R}^{N_h \times n_x}$. Eq. (3.99) is compactly rewritten by defining $\sigma(x) = [\sigma_i(x)]_{i=1}^{N_\sigma}$, $C'' = [c_1'', \cdots, c_{N_\sigma}''] \in \mathbb{R}^{N_h \times N_\sigma}$, and $C = [C',\ C''] \in \mathbb{R}^{N_h \times n_x + N_\sigma}$.

Then, the initial set Φ_{in} is defined as follows:

$$\Phi_{\text{in}} = \{x_i\}_{i=1}^{n_x} \cup \{\sigma_i(x)\}_{i=1}^{N_\sigma}, \quad \phi_{\text{in}}(x) = \begin{bmatrix} x \\ \sigma(x) \end{bmatrix},$$

$$|\Phi_{\text{in}}| = n_x + N_\sigma. \tag{3.100}$$

We can see that Φ_{in} comprises both the system states x (Remark 3.2) and, additionally, all the nonlinear terms introduced by the NMPC nonlinear state constraints (3.98f).

From Φ_{in}, Algorithm 3.3 generates the whole basis Φ, i.e.,

$$\begin{aligned} \Phi &= \Phi_{\text{in}} \cup \{\psi_i(x)\}_{i=1}^{N_\psi}, \quad \phi(x) = \begin{bmatrix} x \\ \sigma(x) \\ \psi(x) \end{bmatrix}, \\ |\Phi| &= N_o = n_x + N_\sigma + N_\psi = n_z, \end{aligned} \tag{3.101}$$

where $\psi(x) = [\psi_i(x)]_{i=1}^{N_\psi}$ are the newly generated observables.

Knowing the basis Φ (3.101), we can now apply Algorithm 3.4 to obtain the bilinear Koopman lifted system, matching Eq. (3.35), i.e.,

$$\dot{z} = Az + \sum_{j=1}^{n_u} B_j z u_j. \tag{3.102}$$

3.5.3. Lifted System Discretization and Linear Parameter-Varying Form

After having obtained the Koopman lifted system (3.102), the next step is to derive a linear DT version of it, to serve as prediction model for the K-NMPC problem.

To this end, we proceed in two steps. First, we resort to the LPV form of the lifted system (3.102), which is given by Eq. (3.52), with parameter $\overline{z} \in \mathcal{Z}$, i.e.,

$$\dot{z} = Az + \sum_{j=1}^{n_u} B_j \overline{z} u_j = Az + \underbrace{[B_1\overline{z}, \cdots, B_{n_u}\overline{z}]}_{\tilde{B}(\overline{z})} u = Az + \tilde{B}(\overline{z})u. \tag{3.103}$$

Second, we discretize the LPV lifted system (3.103) with discrete time step T_s. Since the system is linear and, by the receding horizon policy, the input signal is piecewise constant, i.e., $u(t) = u_k$, $\forall\, t \in [kT_s, (k+1)T_s]$, an exact discretization of system (3.103) can be obtained through the following convolution integral (Section 2.2, p. 12), which can be solved in closed form:

$$\begin{aligned} z_{k+1} &= e^{AT_s} z_k + \int_{kT_s}^{(k+1)T_s} e^{A((k+1)T_s - \tau)} \tilde{B}(\overline{z}) u(\tau) d\tau \\ &= \underbrace{e^{AT_s}}_{A_d} z_k + \underbrace{(e^{AT_s} - I)A^{-1}\tilde{B}(\overline{z})}_{B_d(\overline{z})} u_k = A_d z_k + B_d(\overline{z}) u_k. \end{aligned} \tag{3.104}$$

3.5.4. Koopman NMPC Formulation

Now, we can formulate the K-NMPC problem. Let us rewrite the lifted state as $z = [z_x^\top,\ z_\sigma^\top,\ z_\psi^\top]^\top$, where z_x, z_σ, and z_ψ are the lifted states related to the observables x, $\sigma(x)$, and $\psi(x)$, respectively. Then, the K-NMPC optimal control problem is given by the following QP, for each $k \geq 0$:

$$\min_{\hat{\boldsymbol{z}}, \hat{\boldsymbol{u}}} \quad J(\hat{\boldsymbol{z}}_x, \hat{\boldsymbol{u}}) \tag{3.105a}$$

$$\begin{aligned}
\text{s.t.} \quad & \hat{z}_0 = \phi(x_k), && \text{(3.105b)}\\
& \hat{z}_{i+1} = \bar{A}\hat{z}_i + \bar{B}_{k,i}\hat{u}_i, \quad i = 0, \dots, N_p - 1, && \text{(3.105c)}\\
& \hat{u}_i \in \mathcal{U}, \quad i = 0, \dots, N_p - 1, && \text{(3.105d)}\\
& \hat{z}_{x,i} \in \mathcal{X}, \quad i = 0, \dots, N_p - 1 && \text{(3.105e)}\\
& C \begin{bmatrix} \hat{z}_{x,i} \\ \hat{z}_{\sigma,i} \end{bmatrix} \leq d, \quad i = 0, \dots, N_p - 1. && \text{(3.105f)}
\end{aligned}$$

In the K-NMPC problem (3.105), $\hat{\boldsymbol{z}} = (\hat{z}_i)_{i=0}^{N_p}$ is the predicted trajectory of lifted states. The LPV prediction model (3.105c) is given by Eq. (3.104), where $\bar{A} = A_d$ and $\bar{B}_{k,i} = B_d(\overline{z}_{k,i})$; $\overline{z}_{k,i}$ is the model parameter, which varies at each $k \geq 0$ and along the prediction horizon $i = 0, \dots, N_p - 1$. Specifically, the sequence of parameters $\overline{\boldsymbol{z}}_k = (\overline{z}_{k,i})_{i=0}^{N_p-1}$, as reported in Section 2.3.3, is set equal to the shifted optimal trajectory computed by K-NMPC at the previous time instant $k-1$, i.e., $\hat{\boldsymbol{z}}^*_{k-1} = (\hat{z}^*_{i|k-1})_{i=0}^{N_p}$, as follows:

$$\overline{\boldsymbol{z}}_k = (\overline{z}_{k,i})_{i=0}^{N_p-1} = \begin{cases} (\phi(x_0),\ \dots,\ \phi(x_0)) & \text{if } k = 0, \\ (\phi(x_k),\ \hat{z}^*_{2|k-1},\ \dots,\ \hat{z}^*_{i+1|k-1},\ \dots,\ \hat{z}^*_{N_p|k-1}) & \text{if } k \geq 1. \end{cases} \tag{3.106}$$

The linear state constraints in the original state x (3.98e) are expressed in the lifted state z, as in Eq. (3.105e), by applying the map $z_x = x$ defined by ϕ. Equivalently, the nonlinear state constraints in x (3.98f) are expressed in z, as in Eq. (3.105f), by applying the maps $z_x = x$ and $z_\sigma = \sigma(x)$ to Eq. (3.99). Finally, in the cost function (3.105a), the original state x is replaced by z_x.

NMPC and K-NMPC Solutions

The optimal solution of the NMPC problem (3.98) can be obtained by computing the solution of the K-NMPC problem (3.105) and applying to it the mapping defined by ϕ.

It is worth noticing, however, that, having approximated the bilinearity of the Koopman lifted system (3.102) by employing its LPV version as prediction model, solving the K-NMPC problem leads to an *approximation* of the NMPC solution, i.e.,

$$\hat{x}^*_{\text{NMPC}} \approx \phi^{-1}(\hat{z}^*_{\text{K-NMPC}}), \quad \hat{u}^*_{\text{NMPC}} \approx \hat{u}^*_{\text{K-NMPC}}. \tag{3.107}$$

3.5.5. Koopman NMPC Dimensionality Reduction

When the basis of observables Φ (3.101) is infinite-dimensional or a reduction in the complexity of the K-NMPC problem (3.105) is desired, we can operate a dimensionality reduction of the basis Φ and the Koopman lifted system (3.102), as outlined in Section 3.4.

The reduced lifted system can be expressed in APV form, recalling Eq. (3.93), with parameter $(\overline{z}, \overline{u}) \in \mathcal{Z} \times \mathcal{U}$, i.e.,

$$\dot{z} = \tilde{A}(\overline{z}, \overline{u})z + \tilde{B}(\overline{z}, \overline{u})u + \tilde{b}(\overline{z}, \overline{u}). \tag{3.108}$$

An exact discretization of the APV lifted system (3.108) can be obtained by computing the closed-form solution of the following convolution integral (Section 2.2, p. 12):

$$z_{k+1} = e^{\tilde{A}(\overline{z},\overline{u})T_s} z_k + \int_{kT_s}^{(k+1)T_s} e^{\tilde{A}(\overline{z},\overline{u})((k+1)T_s - \tau)} (\tilde{B}(\overline{z}, \overline{u})u(\tau) + \tilde{b}(\overline{z}, \overline{u}))d\tau$$

$$
\begin{aligned}
&= e^{\tilde{A}(\overline{z},\overline{u})T_s} z_k + (e^{\tilde{A}(\overline{z},\overline{u})T_s} - I)\tilde{A}^{-1}(\overline{z},\overline{u})(\tilde{B}(\overline{z},\overline{u})u_k + \tilde{b}(\overline{z},\overline{u})) \\
&= \underbrace{e^{\tilde{A}(\overline{z},\overline{u})T_s}}_{A_d(\overline{z},\overline{u})} z_k + \underbrace{(e^{\tilde{A}(\overline{z},\overline{u})T_s} - I)\tilde{A}^{-1}(\overline{z},\overline{u})\tilde{B}(\overline{z},\overline{u})}_{B_d(\overline{z},\overline{u})} u_k + \underbrace{(e^{\tilde{A}(\overline{z},\overline{u})T_s} - I)\tilde{A}^{-1}(\overline{z},\overline{u})\tilde{b}(\overline{z},\overline{u})}_{b_d(\overline{z},\overline{u})} \\
&= A_d(\overline{z},\overline{u})z_k + B_d(\overline{z},\overline{u})u_k + b_d(\overline{z},\overline{u}). \qquad (3.109)
\end{aligned}
$$

Then, the reduced K-NMPC problem is formulated as follows, for each $k \geq 0$:

$$
\begin{aligned}
\min_{\hat{\boldsymbol{z}},\hat{\boldsymbol{u}}} \quad & J(\hat{\boldsymbol{z}}_x, \hat{\boldsymbol{u}}) && \text{(3.110a)} \\
\text{s.t.} \quad & \hat{z}_0 = \phi(x_k), && \text{(3.110b)} \\
& \hat{z}_{i+1} = \bar{A}_{k,i}\hat{z}_i + \bar{B}_{k,i}\hat{u}_i + \bar{b}_{k,i}, \quad i = 0, \dots, N_p - 1, && \text{(3.110c)} \\
& \hat{u}_i \in \mathcal{U}, \quad i = 0, \dots, N_p - 1, && \text{(3.110d)} \\
& \hat{z}_{x,i} \in \mathcal{X}, \quad i = 0, \dots, N_p - 1, && \text{(3.110e)} \\
& C \begin{bmatrix} \hat{z}_{x,i} \\ \hat{z}_{\sigma,i} \end{bmatrix} \leq d, \quad i = 0, \dots, N_p - 1. && \text{(3.110f)}
\end{aligned}
$$

In the reduced K-NMPC problem (3.110), the APV prediction model (3.110c) is given by Eq. (3.109), where $\bar{A}_{k,i} = A_d(\overline{z}_{k,i}, \overline{u}_{k,i})$, $\bar{B}_{k,i} = B_d(\overline{z}_{k,i}, \overline{u}_{k,i})$, and $\bar{b}_{k,i} = b_d(\overline{z}_{k,i}, \overline{u}_{k,i})$; $(\overline{z}_{k,i}, \overline{u}_{k,i})$ is the model parameter, which varies at each $k \geq 0$ and along the prediction horizon $i = 0, \dots, N_p - 1$. Specifically, the sequence of parameters $(\overline{\boldsymbol{z}}_k, \overline{\boldsymbol{u}}_k) = \big((\overline{z}_{k,i}, \overline{u}_{k,i})\big)_{i=0}^{N_p-1}$, as reported in Section 2.3.3, is set equal to the shifted optimal trajectory computed by K-NMPC at the previous time instant $k-1$, i.e., $\hat{\boldsymbol{z}}^*_{k-1} = (\hat{z}^*_{i|k-1})_{i=0}^{N_p}$ and $\hat{\boldsymbol{u}}^*_{k-1} = (\hat{u}^*_{i|k-1})_{i=0}^{N_p-1}$, as follows:

$$
\begin{aligned}
(\overline{\boldsymbol{z}}_k, \overline{\boldsymbol{u}}_k) &= \big((\overline{z}_{k,i}, \overline{u}_{k,i})\big)_{i=0}^{N_p-1} \\
&= \begin{cases}
((\phi(x_0), \mathbf{0}_{n_u}),\ \dots,\ (\phi(x_0), \mathbf{0}_{n_u})) & \text{if } k = 0, \\
\begin{array}{l}((\phi(x_k), \hat{u}^*_{1|k-1}),\ (\hat{z}^*_{2|k-1}, \hat{u}^*_{2|k-1}),\ \dots,\ (\hat{z}^*_{i+1|k-1}, \hat{u}^*_{i+1|k-1}),\ \dots, \\ \ (\hat{z}^*_{N_p-1|k-1}, \hat{u}^*_{N_p-1|k-1}),\ (\hat{z}^*_{N_p|k-1}, \mathbf{0}_{n_u}))\end{array} & \text{if } k \geq 1.
\end{cases}
\end{aligned} \qquad (3.111)
$$

3.6. Simulated and Experimental Results

To validate the K-NMPC strategy, we select two different real-world case studies, on which we carry out simulations and experimental validations:

1) mobile robot navigation in cluttered environments (Section 3.6.1);
2) autonomous parallel parking (Section 3.6.2).

3.6.1. Mobile Robot Navigation in Cluttered Environments

As a first application case study for the K-NMPC strategy, we consider the scenario of mobile robot navigation in a cluttered environment.

In this first case study, we assess the capability of K-NMPC to provide a real-time feasible and reliable control action to attain the task. The performance of K-NMPC is compared with classic NMPC, both in simulation and with a real experimental validation.

An illustrative view of the scenario is reported in Figure 3.1.

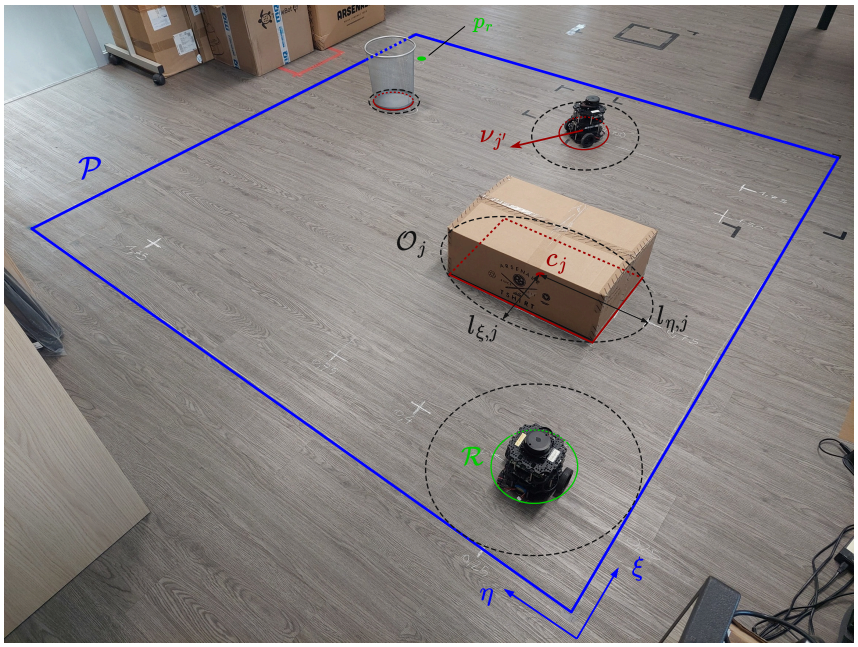


Figure 3.1. Experimental setup for mobile robot navigation in a cluttered environment. The picture shows: the region of space $\mathcal{P}$; the ego robot $\mathcal{R}$, enclosed within its safety circle; the obstacles $\mathcal{O}_j$ populating the cluttered region, with center c_j and velocity ν_j, enclosed within safety ellipsoids with semi-axes $l_{\xi,j}, l_{\eta,j}$ ($j = 1, \ldots, N_{\text{obst}}$); the target position $p_r = [\xi_r, \eta_r]^\top$.

Scenario

The mobile robot motion takes place on the (ξ, η)-plane, within a rectangular region of space $\mathcal{P} = \{p = [\xi, \eta]^\top \in \mathbb{R}^2 : \xi_{\text{lb}} \leq \xi \leq \xi_{\text{ub}},\ \eta_{\text{lb}} \leq \eta \leq \eta_{\text{ub}}\}$. The ego robot is tasked with reaching a target position $p_r = [\xi_r, \eta_r]^\top \in \mathcal{P}$, while avoiding collisions with any obstacle within $\mathcal{P}$, either stationary or moving.

Ego Mobile Robot The ego mobile robot ($\mathcal{R}$) is the plant to control, whose dynamics is governed by a CT nonlinear system

$$\dot{x}(t) = f_{\mathcal{R}}(x(t), u(t)), \tag{3.112}$$

where $x = [\xi, \eta, \psi]^\top \in \mathbb{R}^3$ is the state, comprising the robot planar position (ξ, η) and heading angle ψ; $u = [v, \omega]^\top \in \mathbb{R}^2$ is the input, comprising the robot linear velocity v and angular velocity ω.

The ego robot has intrinsic limits on the input values u, defining an admissible input set

$$\mathcal{U} = \{u \in \mathbb{R}^2 : v_{\text{lb}} \leq v \leq v_{\text{ub}},\ \omega_{\text{lb}} \leq \omega \leq \omega_{\text{ub}}\}. \tag{3.113}$$

The state-space manifold onto which the trajectories of system (3.112) evolve is given by

$$\mathcal{X} = \{x \in \mathbb{R}^3 : [\xi, \eta]^\top \in \mathcal{P},\ \psi \in \mathbb{R}\}. \tag{3.114}$$

We see that the sets $\mathcal{X}$ and $\mathcal{U}$ are convex polytopes.

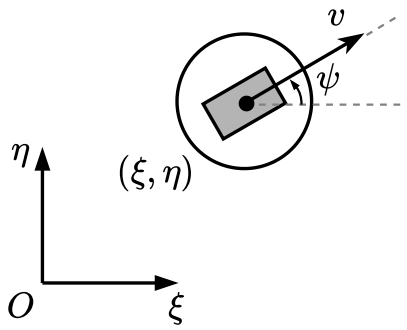


Figure 3.2. Ego robot model (kinematic unicycle).

To model the true plant behavior in Eq. (3.112), we employ the kinematic unicycle model (Figure 3.2), which is CT and input-affine, i.e.,

$$\begin{aligned}\dot{x}(t) = f(x(t), u(t)) &= \begin{bmatrix} v(t)\cos\psi(t) \\ v(t)\sin\psi(t) \\ \omega(t) \end{bmatrix} \\ &= f_0(x(t)) + g_1(x(t))\underbrace{v(t)}_{u_1(t)} + g_2(x(t))\underbrace{\omega(t)}_{u_2(t)} \\ &= 0 + \begin{bmatrix} \cos\psi(t) \\ \sin\psi(t) \\ 0 \end{bmatrix} v(t) + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \omega(t). \end{aligned} \tag{3.115}$$

The approximate model (3.115) is known to represent sufficiently well the true plant dynamics (3.112) (i.e., $f_{\mathcal{R}} \approx f$), ensuring an effective control action [46].

A DT version of model (3.115) can be obtained by applying a discretization method of choice (refer to Section 2.2, p. 12) with discrete time step $T_s > 0$, yielding

$$x_{k+1} = f_d(x_k, u_k). \tag{3.116}$$

Obstacles Several obstacles, either stationary or moving, populate the cluttered region $\mathcal{P}$. Let N_{obst} be the total number of obstacles and let $\mathcal{O}_j$ denote the j-th obstacle ($j = 1, \ldots, N_{\text{obst}}$). To uniformly handle arbitrary obstacle shapes, each j-th obstacle is enclosed within a safety ellipsoid, defined by its center $c_j(t) = [c_{\xi,j}(t), c_{\eta,j}(t)]^\top$ and semi-axes $(l_{\xi,j}, l_{\eta,j})$; let $c(t) = (c_j(t))_{j=1}^{N_{\text{obst}}}$ and $\nu(t) = \dot{c}(t)$ be the obstacle velocities.

Also, we enclose the ego robot within a circle of center $p(t) = [\xi(t), \eta(t)]^\top$ and radius r.

Assumption 3.6

At each time t, the ego robot is assumed to know only the current position $c_j(t)$ and velocity $\nu_j(t)$ of each obstacle.

Given the available information by Assumption 3.6, the ego robot relies on a prediction $\hat{c}_j(t+\tau)$ of each obstacle position at future time instants ($\tau \geq 0$). Specifically, in the prediction, a linear motion is assumed, starting from $c_j(t)$ and with constant velocity $\nu_j(t)$,

$$\hat{c}_j(t+\tau) = c_j(t) + \nu_j(t)\tau, \quad \tau \geq 0. \tag{3.117}$$

Assumption 3.7

The true motion of any obstacle is predicted exactly, i.e.,

$$c_j(t+\tau) = \hat{c}_j(t+\tau), \quad \forall\, t \geq 0,\ \tau \geq 0. \tag{3.118}$$

Hence, by Eq. (3.117), each obstacle moves linearly with constant velocity.

Finally, we represent each j-th obstacle as a time-varying nonlinear inequality constraint on the state x, defining the planar positions for which the ego robot is not colliding with the obstacle at time $t+\tau$ under the prediction (3.117). The safety ellipsoids yield a common constraint expression for any obstacle, given by

$$\begin{aligned} &1 - \frac{(\xi - c_{\xi,j}(t) - \nu_{\xi,j}(t)\tau)^2}{(l_{\xi,j} + \alpha_j r)^2} - \frac{(\eta - c_{\eta,j}(t) - \nu_{\eta,j}(t)\tau)^2}{(l_{\eta,j} + \alpha_j r)^2} \leq 0 \\ &\Rightarrow h_{j,\tau}(x, c_j(t), \nu_j(t)) \leq 0. \end{aligned} \tag{3.119}$$

where $h_\tau(x, c(t), \nu(t)) = [h_{j,\tau}(x, c_j(t), \nu_j(t))]_{j=1}^{N_{\text{obst}}}$ and the safety margin $\alpha_j \geq 1$ can be either selected at hand or set to its minimal value, which can be computed offline for each obstacle through the following optimization problem [139]:

$$\alpha_j = -\frac{1}{r} \min_{\theta \in [0, \pi/2]} l_{\xi,j}\mathrm{c}_\theta^2 + l_{\eta,j}\mathrm{s}_\theta^2 - \Big(\big(l_{\xi,j}\mathrm{c}_\theta^2 + l_{\eta,j}\mathrm{s}_\theta^2\big)^2 + 2r\big(l_{\xi,j}^2\mathrm{c}_\theta^2 + l_{\eta,j}^2\mathrm{s}_\theta^2\big)^{1/2} + r^2 \Big)^{1/2}, \tag{3.120}$$

with $\mathrm{c}_\theta = \cos\theta$ and $\mathrm{s}_\theta = \sin\theta$.

A DT version of constraints (3.119) can be obtained by setting $t = kT_s$ and $\tau = iT_s$, $i \in \mathbb{Z}_{\geq 0}$, yielding

$$h_{iT_s}(x, c(kT_s), \nu(kT_s)) = h_{d,i}(x, c_k, \nu_k). \tag{3.121}$$

NMPC Control Strategy

Here, we formulate a classic NMPC optimal control problem for the mobile robot navigation task, for each $k \geq 0$, as follows:

$$\begin{aligned} \min_{\hat{\boldsymbol{x}}, \hat{\boldsymbol{u}}} \quad & J(\hat{\boldsymbol{x}}, \hat{\boldsymbol{u}}) && \text{(3.122a)} \\ \text{s.t.} \quad & \hat{x}_0 = x_k, && \text{(3.122b)} \\ & \hat{x}_{i+1} = f_d(\hat{x}_i, \hat{u}_i), \quad i = 0, \dots, N_p - 1, && \text{(3.122c)} \\ & \hat{u}_i \in \mathcal{U}, \quad i = 0, \dots, N_p - 1, && \text{(3.122d)} \\ & \hat{x}_i \in \mathcal{X}, \quad i = 0, \dots, N_p - 1, && \text{(3.122e)} \\ & h_{d,i}(\hat{x}_i, c_k, \nu_k) \leq 0, \quad i = 0, \dots, N_p - 1. && \text{(3.122f)} \end{aligned}$$

In the NMPC problem (3.122), the cost function (3.122a) is given by Eq. (3.98a), with reference state $x_{r,k} = [\xi_r, \eta_r, 0]^\top$, $\forall\, k \geq 0$. The prediction model (3.122c) is given by the DT kinematic unicycle model (3.116). The linear constraints (3.122d) and (3.122e) are given by Eqs. (3.113) and (3.114), respectively. The nonlinear state constraints (3.122f) are given by Eq. (3.121), modeling the obstacles.

Koopman NMPC Control Strategy

Now, we begin the construction of the K-NMPC control policy, by lifting the NMPC problem (3.122) in the Koopman space, as detailed in Section 3.5.

Nonlinear State Constraints and Initial Set of Observables Recalling the nonlinear state constraint expression in Eq. (3.119) for each j-th obstacle, we can expand it following Eq. (3.99); defining $a_j = 1/(l_{\xi,j} + \alpha_j r)^2$ and $b_j = 1/(l_{\eta,j} + \alpha_j r)^2$, we obtain

$$\underbrace{\begin{bmatrix} 2b_j(c_{\xi,j}(t) + v_{\xi,j}(t)\tau) \\ 2a_j(c_{\eta,j}(t) + v_{\eta,j}(t)\tau) \\ 0 \\ -b_j \\ -a_j \end{bmatrix}^\top}_{C} \left[\begin{array}{c} \xi \\ \eta \\ \psi \\ \hline \xi^2 \\ \eta^2 \end{array}\right] - \Big(\underbrace{-a_j b_j + b_j(c_{\xi,j}(t) + v_{\xi,j}(t)\tau)^2 + a_j(c_{\eta,j}(t) + v_{\eta,j}(t)\tau)^2}_{d}\Big) \leq 0. \tag{3.123}$$

We see that the nonlinear state constraints introduce 2 nonlinear terms, ξ^2 and η^2, which are collected as $\sigma(x) = [\xi^2, \eta^2]^\top$.

Then, as discussed in Section 3.5.2, the initial set of observables Φ_{in} is given by

$$\Phi_{\text{in}} = \{\xi,\ \eta,\ \psi,\ \xi^2,\ \eta^2\}, \quad |\Phi_{\text{in}}| = 5, \tag{3.124}$$

comprising the states of system (3.115) and the nonlinear terms introduced by constraints (3.119).

Observables Generation and Lifted System From Φ_{in} (3.124), the basis of observables Φ and the related Koopman lifted system – lifting both the kinematic unicycle model (3.115) and the nonlinear state constraints (3.119) – are generated through Algorithms 3.3 and 3.4.

Algorithm 3.3 generates a finite-dimensional basis Φ of 14 observables, i.e.,

$$\begin{aligned} \Phi = \{&\xi,\ \eta,\ \psi,\ \xi^2,\ \eta^2,\ \cos\psi,\ \sin\psi,\ \xi\cos\psi,\ \eta\cos\psi, \\ &\cos^2\psi,\ \xi\sin\psi,\ \sin^2\psi,\ \eta\cos\psi,\ \cos\psi\sin\psi\}, \quad |\Phi| = 14. \end{aligned} \tag{3.125}$$

This means that, remarkably, system (3.115), with nonlinear state constraints (3.119), admits an exact finite-dimensional Koopman representation.

Then, Algorithm 3.4 generates the bilinear Koopman lifted system, with dimension $n_z = |\Phi| = 14$ and characterized as follows:

$$\dot{z} = Az + B_0 u + B_1 z u_1 + B_2 z u_2, \tag{3.126}$$

$$A = \mathbf{0}_{14\times 14}, \qquad B_0 = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 1 \\ \mathbf{0}_{11\times 2} \end{bmatrix},$$

$$B_1 = \begin{bmatrix} & \begin{smallmatrix} 1&0&0&0&0&0&0&0&0 \\ 0&1&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0 \\ 0&0&2&0&0&0&0&0&0 \\ 0&0&0&2&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&1&0&0&0&0 \\ 0&0&0&0&0&0&1&0&0 \\ 0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&1 \\ 0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&1 \\ 0&0&0&0&0&0&0&0&0 \end{smallmatrix} \\ \mathbf{0}_{14\times5} & \end{bmatrix}, \quad B_2 = \begin{bmatrix} \mathbf{0}_{5\times5} & \mathbf{0}_{5\times9} \\ & \begin{smallmatrix} 0&-1&0&0&0&0&0&0&0 \\ 1&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&-1&0&0&0 \\ 0&0&0&0&0&0&0&1&0 \\ 0&0&0&0&0&0&0&0&-2 \\ 0&0&1&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&2 \\ 0&0&0&-1&0&0&0&0&0 \\ 0&0&0&0&1&0&-1&0&0 \end{smallmatrix} \\ \mathbf{0}_{9\times5} & \end{bmatrix}. \tag{3.127}$$

The structure of the Koopman lifted system (3.126) matches Eq. (3.43), since system (3.115) contains an input-linear term.

Dimensionality Reduction Even if system (3.115), (3.119) admits an exact finite-dimensional lifted system, we may want to further reduce its dimension, so to have a less complex K-NMPC problem.

To determine the "best" dimension reduction, we follow the procedure detailed in Section 3.4.1: we consider a set of trajectories $x(t)$ of system (3.115) and we evaluate the prediction accuracy of the reduced Koopman lifted system for increasing number of observables ϕ.

Each state trajectory $x(t) = [\xi(t), \eta(t), \psi(t)]^\top$ is computed over the time interval $t \in [0, T]$, with initial condition $x(0) = [0 \text{ m}, \ 0 \text{ m}, \ 0 \text{ rad}]^\top$, applying an input signal belonging to the following family:

$$u(t) = \begin{bmatrix} v_0 \sin\left(2\pi \frac{t}{n_1 T} + \varphi_1\right) \\ \omega_0 \sin\left(2\pi \frac{t}{n_2 T} + \varphi_2\right) \end{bmatrix}, \tag{3.128}$$

with parameters v_0, ω_0, $n_1, n_2 \in \mathbb{R}_{>0}$, and $\varphi_1, \varphi_2 \in [-\pi, \pi]$ rad. The family of input signals (3.128) is designed to ensure a sufficiently rich excitation of the system, yielding representative state trajectories for evaluating the prediction accuracy. Specifically, the sinusoidal signals $v(t)$ and $\omega(t)$ produce time-varying curvatures, sign changes, and low-speed intervals. Moreover, the resulting trajectories are quasi-periodic and non-closed, allowing to expose error accumulation over longer prediction horizons.

The trajectory $x(t)$ is sampled at N_{pred} equally-spaced time instants $t_n = \frac{n}{N_{\text{pred}}} T$, extracting the states $x_n = x(t_n)$, $n = 1, \ldots, N_{\text{pred}}$. Taking each lifted state $z_n = \phi(x_n)$ as an initial condition, a prediction $z^{(n)}_{\text{pred}}(\tau)$ of the reduced lifted system is computed, over the prediction time interval $\tau \in [0, T_p]$, applying the input $u(t)$.

The lifted trajectory $z(t) = \phi(x(t))$ and the predictions $z_{\text{pred}}(t)$ are then expressed as

$$z(t) = \phi(x(t)) = \begin{bmatrix} z_x(t) \\ z_\sigma(t) \\ z_\psi(t) \end{bmatrix}, \quad z_{\text{pred}}(t) = \begin{bmatrix} z_{x,\text{pred}}(t) \\ z_{\sigma,\text{pred}}(t) \\ z_{\psi,\text{pred}}(t) \end{bmatrix}. \tag{3.129}$$

The true and predicted trajectories (3.129) are compared by sampling each of them with sampling time T_s. Specifically, each prediction time interval will contain $N_p + 1$ sampling points, with $N_p = \frac{T_p}{T_s}$.

For each n-th predicted trajectory, three prediction errors are considered:

1) Prediction error $\epsilon_{\text{pred},1}$ between the original trajectory $x(t)$ and the predicted lifted states $z_{x,\text{pred}}(t)$, associated with the original states x, i.e.,

$$\epsilon^{(n)}_{\text{pred},1} = \frac{1}{N_p}\sum_{i=1}^{N_p} \left\| z^{(n)}_{x,\text{pred}}(iT_s) - x(t_n + iT_s) \right\|_\infty ,$$

2) Prediction error $\epsilon_{\text{pred},2}$ between the true lifted states $z_\sigma(t)$ and the predicted lifted states $z_{\sigma,\text{pred}}(t)$, associated with the constraint nonlinearities $\sigma(x)$, i.e.,

$$\epsilon^{(n)}_{\text{pred},2} = \frac{1}{N_p}\sum_{i=1}^{N_p} \left\| z^{(n)}_{\sigma,\text{pred}}(iT_s) - z_\sigma(t_n + iT_s) \right\|_\infty ,$$

3) Prediction error $\epsilon_{\text{pred},3}$, combining the previous ones, i.e.,

$$\epsilon^{(n)}_{\text{pred},3} = \frac{1}{N_p}\sum_{i=1}^{N_p} \left\| \begin{bmatrix} z^{(n)}_{x,\text{pred}}(iT_s) \\ z^{(n)}_{\sigma,\text{pred}}(iT_s) \end{bmatrix} - \begin{bmatrix} x(t_n+iT_s) \\ z_\sigma(t_n+iT_s) \end{bmatrix} \right\|_\infty .$$

All the above calculations are repeated for increasing number of observables $\phi(x) = [\phi_i(x)]_{i=1}^{\overline{N}_o}$, with $\overline{N}_o = 5, \ldots, 14$.

The prediction accuracy has been evaluated on a set of 10 trajectories $x(t)$, generated by randomly selecting the parameters of the input signal $u(t)$ in Eq. (3.128) as follows:

$$\begin{aligned} &v_0 \sim U([0.1, 0.3]\ \text{m s}^{-1}), && n_1 \sim U([0.25, 1]), && \varphi_1 = -\pi/2\ \text{rad}, \\ &\omega_0 \sim U([\pi/4, \pi/2]\ \text{rad s}^{-1}), && n_2 \sim U([0.25, 1]), && \varphi_2 = 0\ \text{rad}, \end{aligned}$$

where $U(\cdot)$ is the uniform probability distribution.

Each trajectory $x(t)$ and predictions $z_{\text{pred}}(t)$ are computed with the Runge-Kutta 4(5) integration method (with tolerances $\epsilon_{\text{abs}} = \epsilon_{\text{rel}} = 1 \times 10^{-6}$).

Additional data is as follows: $T = 5$ s, $T_s = 0.05$ s, $N_{\text{pred}} = 10$, $T_p = 1$ s ($N_p = 20$).

Results are reported in Figure 3.3.

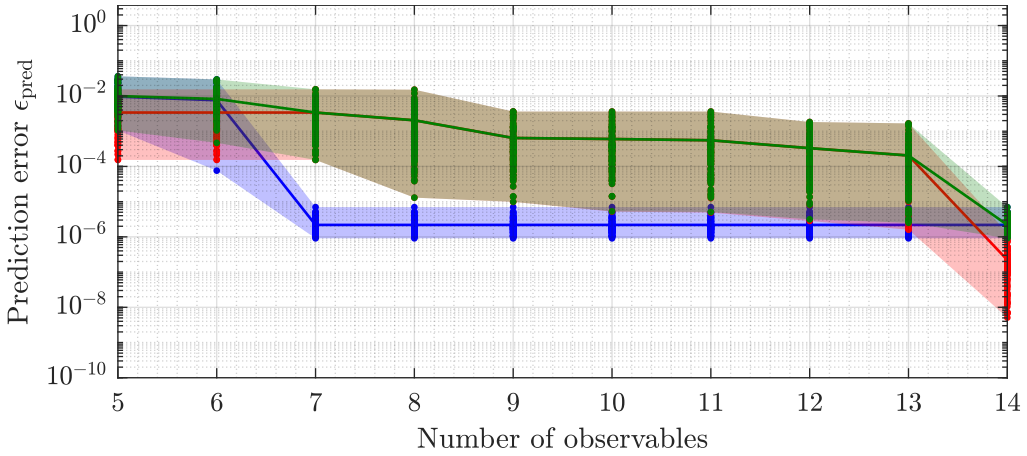


Figure 3.3. Prediction errors of the reduced Koopman lifted system for the kinematic unicycle model (3.115) with nonlinear state constraints (3.119), for increasing number of observables ϕ: single predictions ($\epsilon_{\text{pred},1}$ •, $\epsilon_{\text{pred},2}$ •, $\epsilon_{\text{pred},3}$ •); average values ($\bar{\epsilon}_{\text{pred},1}$ —, $\bar{\epsilon}_{\text{pred},2}$ —, $\bar{\epsilon}_{\text{pred},3}$ —).

First, we can observe that all prediction errors are non-increasing as the number of observables increases, which aligns with the theoretical expectation that the more

observables are generated, the closer the reduced Koopman lifted system will be to the exact one.

For what concerns the prediction accuracy of the original trajectory $x(t)$ only (i.e., disregarding the constraint nonlinearities $\sigma(x(t))$), given by $\epsilon_{\text{pred},1}$ (•, — in Fig. 3.3), a steep reduction in the prediction error occurs once the number of observables reaches 7, after which no further decrease is observed and its value settles at approximately 0 (being it close to the integration tolerances).

This result can be readily interpreted if we explicitly compute the reduced Koopman lifted system (Section 3.4); truncating the basis at 7 observables, the nonlinear residual $f_{\text{res}}(z,u)$ affects only the last 5 lifted states and is given by

$$f_{\text{res}}(z,u) = \begin{bmatrix} 0 \\ 2u_1 z_1 \cos(z_3) \\ 2u_1 z_2 \sin(z_3) \\ 0 \\ 0 \end{bmatrix}.$$

The non-null terms of f_{res} only affect the 4^{th} and 5^{th} lifted states, i.e., z_σ (those associated with the constraint nonlinearities $\sigma(x)$). Since the lifted dynamics of the states z_x does not depend on z_σ, the reduced lifted system exactly evaluates $z_x(t) = x(t)$.

Therefore, in absence of nonlinear constraints (3.119), the lifted system dimension can be reduced to $n_z = 7$.

Focusing instead on the prediction accuracy of the constraint nonlinearities $\sigma(x(t))$ only, given by $\epsilon_{\text{pred},2}$ (•, — in Fig. 3.3), the prediction error decreases more slowly: an initial reduction is observed up to 9 observables, followed by a more gradual decline until 14 observables, at which point the lifted system becomes exact.

Therefore, to ensure that the lifted system accurately captures the constraint nonlinearities – so that the closed-loop system, controlled by K-NMPC, reliably satisfies the nonlinear constraints (3.119) – its dimension can be reduced to no less than $n_z = 9$.

The prediction error $\epsilon_{\text{pred},3}$ (•, — in Fig. 3.3) sums up all the above observations, allowing us to conclude that we can confidently reduce the dimension of the Koopman lifted system to $n_z = 9$.

K-NMPC Formulation To write the K-NMPC optimal control problem for the mobile robot navigation task, the following two final steps are performed:

- As K-NMPC prediction model, we consider the DT LPV/APV form of the Koopman lifted system (3.127) (refer to Sections 3.5.3 and 3.5.5), i.e.,

$$\begin{aligned} \hat{z}_{i+1} &= A_d(\overline{z}_{k,i}, \overline{u}_{k,i})\hat{z}_i + B_d(\overline{z}_{k,i}, \overline{u}_{k,i})\hat{u}_i + b_d(\overline{z}_{k,i}, \overline{u}_{k,i}) \\ &= \bar{A}_{k,i}\hat{z}_i + \bar{B}_{k,i}\hat{u}_i + \bar{b}_{k,i}. \end{aligned}$$

- We construct the lifted constraints (3.105d) by stacking together the expressions (3.123) of all obstacles $j = 1, \dots, N_{\text{obst}}$ and considering their DT version, as in Eq. (3.121); we then obtain the constraint matrices $C_{d,i}(c_k, \nu_k)$ and $d_{d,i}(c_k, \nu_k)$.

Then, the K-NMPC problem is formulated as follows, for each $k \geq 0$:

$$\min_{\hat{\boldsymbol{z}}, \hat{\boldsymbol{u}}} \quad J(\hat{\boldsymbol{z}}_x, \hat{\boldsymbol{u}}) \tag{3.131a}$$

$$\text{s.t.} \quad \hat{x}_0 = x_k, \tag{3.131b}$$
$$\hat{z}_{i+1} = \bar{A}_{k,i}\hat{z}_i + \bar{B}_{k,i}\hat{u}_i + \bar{b}_{k,i}, \quad i = 0, \dots, N_p - 1, \tag{3.131c}$$
$$\hat{u}_i \in \mathcal{U}, \quad i = 0, \dots, N_p - 1, \tag{3.131d}$$
$$\hat{z}_{x,i} \in \mathcal{X}, \quad i = 0, \dots, N_p - 1, \tag{3.131e}$$
$$C_{d,i}(c_k, \nu_k) \begin{bmatrix} \hat{z}_{x,i} \\ \hat{z}_{\sigma,i} \end{bmatrix} \leq d_{d,i}(c_k, \nu_k), \quad i = 0, \dots, N_p - 1. \tag{3.131f}$$

Implementation Details

In the following, we report some details on the implementation of this case study, concerning simulations and the real experimental validation.

Simulations Simulations are performed in MATLAB® (ver. 2023b) on a machine powered by a 13$^{\text{th}}$ Gen Intel Core™ i7 CPU at 1.7 GHz, with 16 GB of RAM.

The K-NMPC optimal control problem (3.131) is formulated with YALMIP [96] and solved with the QP interior-point solver MOSEK (ver. 10.1) [108].

The classic NMPC optimal control problem (3.122) is formulated with CasADi [8] and solved with the NLP interior-point solver Ipopt [153].

Experimental Validation The experimental validation is performed employing MATLAB (ver. 2025a) in combination with the ROS Toolbox. This integration enables a direct communication between MATLAB and the ROS2 network, ensuring seamless interaction with the robotic platforms used for testing.

The robots employed in the experiments are *TurtleBot3 Burger* units, a differential-drive mobile robot platform commonly used for research and education [150]. Specifically, two TurtleBots are used: the first acting as ego robot, and the second acting as a moving obstacle within the cluttered region. Although each robot is equipped with a 2D LiDAR, this onboard sensor is not exploited for the purposes of this validation. The robotic system is configured to operate with *ROS2 Jazzy* and the standard navigation stack *Nav2*.

When powered on, the robots automatically connect to the same local area network (LAN). To enable communication between the robots, a common ROS domain ID is specified in the configuration of each robot; in this way, the robots become part of the same ROS2 network and are capable of exchanging information with each other.

On the MATLAB side, communication with the ROS2 network is established by setting the same ROS domain ID as an environment variable. This step, supported natively by the ROS Toolbox, allows the K-NMPC controller (3.131) (implemented with YALMIP as a MATLAB script) to act as a ROS2 node, subscribing to and publishing data on the same network as the robots. Within this framework, the control script is connected to the robots in a structured manner through the creation of ROS2 subscribers and publishers. Specifically, two subscribers are implemented to acquire odometry data from the two TurtleBots.

The odometry data, transmitted in the form of structured ROS messages, is extracted, processed in MATLAB, and passed to the K-NMPC controller. The K-NMPC optimal control problem is solved at this stage, computing the optimal control input, which is transmitted back to the ego robot through a ROS2 publisher, which publishes the commands

for the linear and angular velocities, closing the control loop.

Data Relevant data used in both simulation and experimental validation is reported in the following:

- General data: $T_s = 0.1$ s.
- Ego mobile robot: $r = 0.25$ m, $x_0 = [0 \text{ m}, \ 0 \text{ m}, \ 0 \text{ deg}]^\top$.
- MPC: $N_p = 20$, $x_r = [2 \text{ m}, \ 2 \text{ m}, \ 0 \text{ rad}]^\top$, $Q = \text{diag}([50, 50, 10^{-3}]^\top)$, $R = \text{diag}([10^{-3}, 1]^\top)$, $Q_\Delta = \text{diag}([10^{-1}, 10^{-1}, 10^{-1}]^\top)$, $R_\Delta = \text{diag}([10^{-1}, 25]^\top)$, $\mathcal{U} = \{u \in \mathbb{R}^2 : u_{\text{lb}} \leq u \leq u_{\text{ub}}\}$, $u_{\text{lb}} = [-0.2 \text{ m s}^{-1}, \ -1.5 \text{ rad s}^{-1}]^\top$, $u_{\text{ub}} = [0.2 \text{ m s}^{-1}, \ 1.5 \text{ rad s}^{-1}]^\top$, $\mathcal{X} = \{x \in \mathbb{R}^3 : x_{\text{lb}} \leq x \leq x_{\text{ub}}\}$, $x_{\text{lb}} = [-0.25 \text{ m}, \ -0.25 \text{ m}, \ -\infty]^\top$, $x_{\text{ub}} = [2.25 \text{ m}, \ 2.25 \text{ m}, \ +\infty]^\top$.
- Obstacles: 3 obstacles, 2 fixed and 1 moving.
  - Fixed obstacle 1: $c_1 = [0.75, 0.4]^\top$ m, $l_{\xi,1} = 0.4/\sqrt{2}$ m, $l_{\eta,1} = 0.6/\sqrt{2}$ m, $\alpha_1 = 1.1$.
  - Fixed obstacle 2: $c_2 = [1.5, 1.75]^\top$ m, $l_{\xi,2} = 0.15$ m, $l_{\eta,2} = 0.15$ m, $\alpha_2 = 1.1$.
  - Moving obstacle: $c_3 = [1.75, 0.75]^\top$ m at $t = 0$ s, $\nu_3 = [-0.1, 0.1]^\top$ m s^{-1}, $l_{\xi,3} = 0.3$ m, $l_{\eta,3} = 0.3$ m, $\alpha_3 = 1.1$.

Results

Figure 3.4 reports the ego robot planar motion within the cluttered environment. The ego robot closed-loop planar trajectories have been obtained by employing the K-NMPC (3.131) and classic NMPC (3.122) control strategies, both in simulation and through experimental validation.

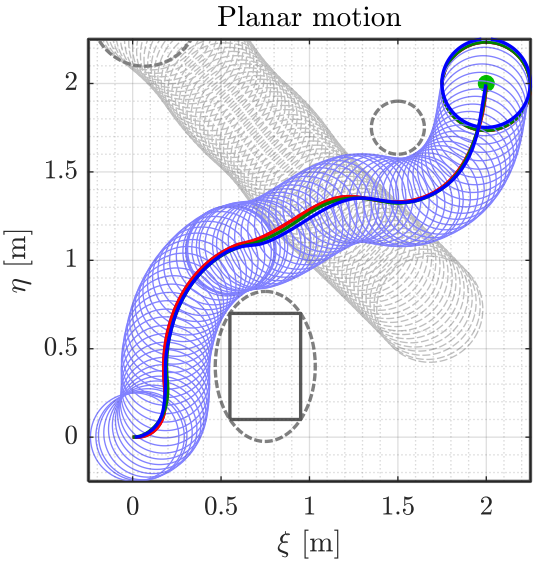


Figure 3.4. Planar motion of the ego mobile robot within the cluttered region: ego robot closed-loop trajectories (K-NMPC, experimental ——; K-NMPC, simulated, $n_z = 14$ ——; K-NMPC, simulated, reduced to $n_z = 9$ ——; NMPC ——); reference position $p_r = [\xi_r, \eta_r]^\top$ (●); obstacles within their safety ellipsoids. For the K-NMPC experimental trajectory (——), the instantaneous positions of the ego robot safety circle and the moving obstacle safety ellipsoid are also shown.

Specifically, simulated trajectories have been obtained using: K-NMPC with full-dimensional lifted state $n_z = 14$ (— in Fig. 3.4); K-NMPC with dimensionality reduction, reducing the lifted state dimension to $n_z = 9$ (— in Fig. 3.4), in accordance with the results obtained in Section 3.6.1, p. 97; classic NMPC (— in Fig. 3.4).

The experimental trajectory has been obtained using K-NMPC with full-dimensional lifted state (— in Fig. 3.4).

Figure 3.5 focuses on the experimental validation and reports some snapshots of the real ego robot motion within the experimental scenario. Snapshots have been captured every 0.5 s; the figure reports the most significant ones.

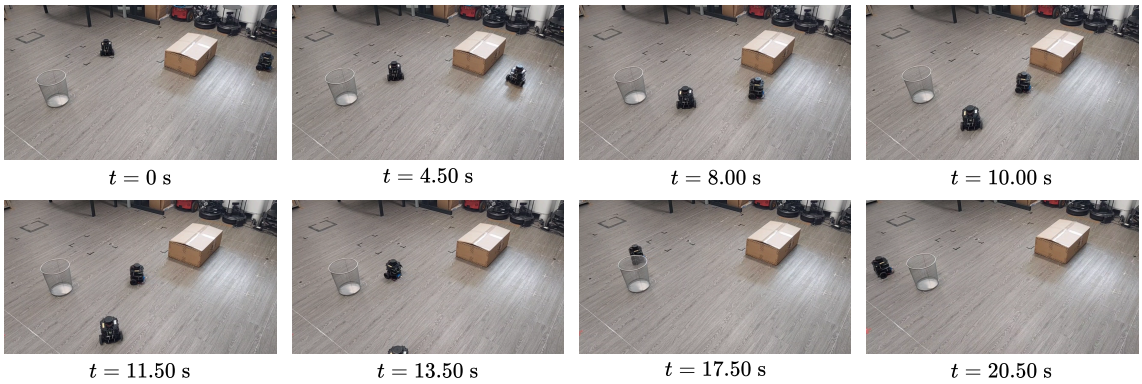


Figure 3.5. Planar motion of the ego mobile robot within the cluttered region: snapshots of the experimental validation, captured every 0.5 s.

Figure 3.6 reports the closed-loop state and input trajectories of the ego mobile robot as functions of the continuous time t. Depending on which control strategy is used, the ego robot reaches the target position at slightly different times; therefore, Figure 3.6 employs a normalized time axis to allow for a better visual comparison of the trajectories. In simulation, the control time interval of each strategy is the following: K-NMPC with full-dimensional lifted state, $t \in [0, 18.70]$ s; K-NMPC with dimensionality reduction, $t \in [0, 18.70]$ s; NMPC, $t \in [0, 18.30]$ s. In the experimental validation, the control time interval of K-NMPC is $t \in [0, 20.66]$ s.

From Figure 3.4, we see that K-NMPC manages to successfully attain the control task, leading the ego mobile robot to the target position and effectively avoiding all the obstacles within the cluttered region, both fixed and moving. This can be especially observed in the experimental validation, as shown by Figure 3.5. To better visualize the proficiency of K-NMPC in avoiding the moving obstacle, a detail of the maneuver is reported in Figure 3.7, considering only the experimental trajectory.

From Figures 3.4 and 3.6, we can also observe that the K-NMPC and NMPC trajectories are remarkably similar. This result is coherent with Eq. (3.107) and substantiates the reliability of K-NMPC (3.131), which is able to provide a control action closely matching that of the original unlifted NMPC problem (3.122). Moreover, this also shows that the K-NMPC prediction model (3.110b), based on the LPV/APV form of the lifted system and evaluated along the previous optimal predicted trajectory, provides a very good approximation of the true bilinear Koopman lifted system.

Additionally, the closed-loop trajectories given by the full-dimensional K-NMPC ($n_z = 14$) and the reduced K-NMPC ($n_z = 9$) are practically identical (indeed, the red — and

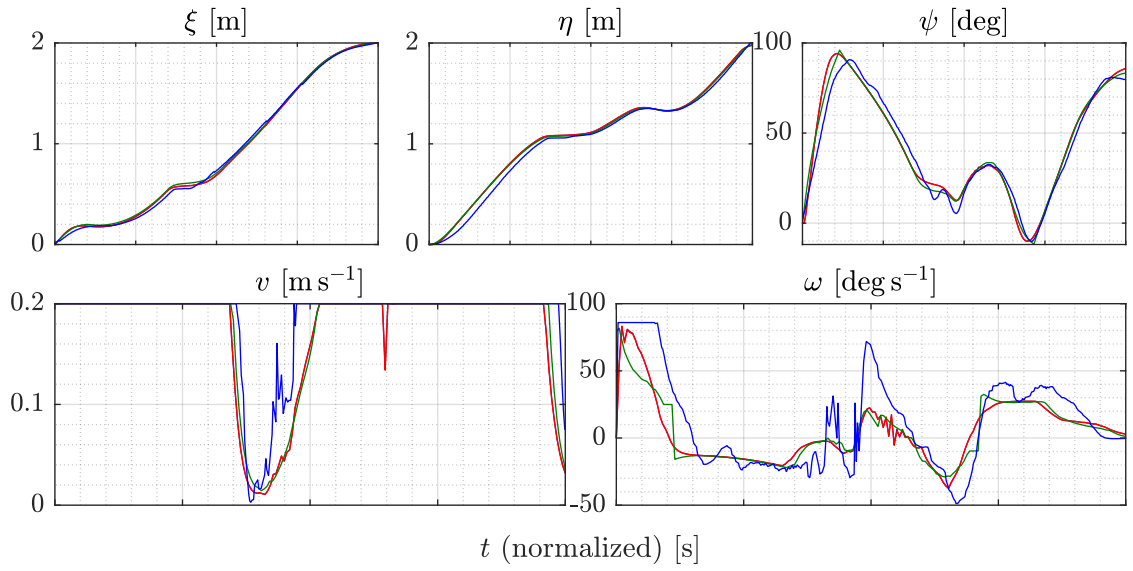


Figure 3.6. Closed-loop state and input trajectories of the ego mobile robot, moving within the cluttered region; a normalized time axis is used to better compare the trajectories, as they reach the reference in slightly different times: K-NMPC, experimental — ($t \in [0, 20.66]$ s); K-NMPC, simulated, $n_z = 14$ — ($t \in [0, 18.70]$ s); K-NMPC, simulated, reduced to $n_z = 9$ — ($t \in [0, 18.70]$ s); NMPC — ($t \in [0, 18.30]$ s).

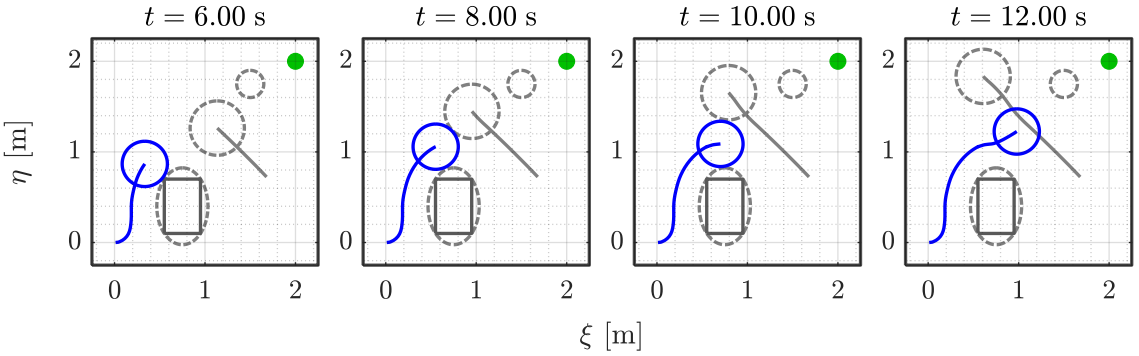


Figure 3.7. Planar motion of the ego mobile robot within the cluttered region: detail of the moving obstacle avoidance (K-NMPC, experimental —; moving obstacle, experimental —; reference position ●).

purple — plots in Figs. 3.4 and 3.6 are almost coincident). This proves the effectiveness of the dimensionality reduction procedure described in Sections 3.4.1 and 3.6.1, p. 97.

Execution Times Execution times have been collected over 20 simulations. Data is reported in Figure 3.8, as function of the continuous time t; minimum, maximum, and average execution times are summarized in Table 3.1.

We can see that K-NMPC achieves a remarkable ten-fold reduction in the execution time compared to classic NMPC. Additionally, the reduced K-NMPC manages to attain even lower execution times.

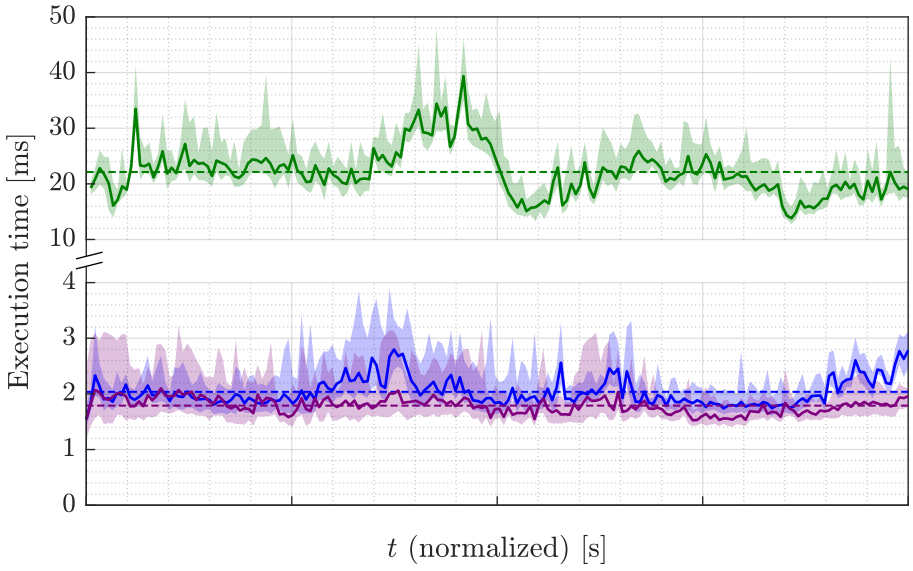


Figure 3.8. Execution times for the ego mobile robot navigation in the cluttered region: K-NMPC, experimental (envelope, average, cumulative average); K-NMPC, simulated, reduced to $n_z = 9$ (envelope, average, cumulative average); NMPC (envelope, average, cumulative average).

	Min	Average	Max
NMPC	12.72 ms	22.14 ms	48.22 ms
K-NMPC	1.39 ms	2.04 ms	3.90 ms
K-NMPC (reduced)	1.18 ms	1.79 ms	3.22 ms

Table 3.1. Execution times for the ego mobile robot navigation in the cluttered region: minimum, maximum and average values.

3.6.2. Autonomous Parallel Parking

As a second application case study for K-NMPC, we consider the scenario of autonomous parallel parking.

Differently from the many advanced driver assistance systems (ADAS) that are increasingly present in modern cars, autonomous parking represents a fully-autonomous feature. Currently, implementing autonomous functions in vehicles remains a challenge, due to safety and regulatory concerns. Nonetheless, autonomous parking is expected to be one of the first to reach full autonomy, as it takes place in structured environments and involves relatively low risks. Furthermore, the increasing number of vehicles in metropolitan areas has diminished parking availability, leading to a reduction in parking space sizes, which complicates manual parking and worsens traffic congestion [20].

Therefore, autonomous parking is a topic of interest within the domain of automotive control, as it would greatly simplify the actions required by the driver to park, reduce the maneuvering time, and improve the parked vehicle positioning.

Scenario

The ego vehicle motion takes place on the (ξ, η)-plane, within a region of space $\mathcal{P}$ modeling a horizontal road, delimited by two footpaths. Then, $\mathcal{P}$ is bounded only on the η-dimension, i.e., $\mathcal{P} = \{p = [\xi, \eta]^\top \in \mathbb{R}^2 : \eta_{\text{lb}} \leq \eta \leq \eta_{\text{ub}}\}$.

Two vehicles are parked along the lower footpath, leaving a parking space between them.

The ego vehicle, whose starting position is to the left side of the road, is tasked with performing a parallel parking between the two parked vehicles, through a two-step maneuver:

1) approaching the front parked vehicle until reaching lateral alignment;
2) switch to reverse driving and precisely position between the parked vehicles.

Each phase of the parking maneuver is formulated as a regulation control problem, with reference states $(x_r^{(n)})_{n=1}^2$ that the ego vehicle has to reach. The transition between successive references, i.e., $x_r^{(1)} \to x_r^{(2)}$, is triggered when

$$\|x_k - x_r^{(n)}\|_\infty \leq \epsilon_r, \tag{3.132}$$

being x_k the current ego vehicle state and ϵ_r a fixed tolerance.

The parking maneuver has to be perform avoiding any collision with the parked vehicles and the footpaths.

Ego Vehicle The ego vehicle ($\mathcal{V}$) is the plant to control, whose dynamics is governed by a CT nonlinear system. To model such a system, we consider a high-fidelity Dynamic Dual-Track (DDT) vehicle model (Figure 3.9a), presented in the following:

$$\dot{\xi} = v_\xi \cos\psi - v_\eta \sin\psi, \tag{3.133a}$$
$$\dot{\eta} = v_\xi \sin\psi + v_\eta \cos\psi, \tag{3.133b}$$
$$\dot{\psi} = \omega_\psi, \tag{3.133c}$$
$$\dot{v}_\xi = v_\eta \omega_\psi + a_\xi + \frac{1}{M} F_{\xi,e}, \tag{3.133d}$$
$$\dot{v}_\eta = -v_\xi \omega_\psi + \frac{1}{M}(F_{\eta,fl} + F_{\eta,fr} + F_{\eta,rl} + F_{\eta,rr} + F_{\eta,e}), \tag{3.133e}$$
$$\dot{\omega}_\psi = \frac{1}{I_z}(l_f(F_{\eta,fl} + F_{\eta,fr}) - l_r(F_{\eta,rl} + F_{\eta,rr}) + d_f(F_{\xi,fl} - F_{\xi,fr}) + d_r(F_{\xi,rl} - F_{\xi,rr})). \tag{3.133f}$$

In model (3.133), (ξ, η) is the ego vehicle planar position, ψ the heading angle, v_ξ and v_η the longitudinal and lateral velocities, respectively, ω_ψ the yaw rate, M and I_z the vehicle mass and yaw polar inertia; (l_f, l_r) are the distances between front/rear wheels and the vehicle center of gravity (CG), (d_f, d_r) are half of the front/rear track widths; $F_{\xi,\star}$ and $F_{\eta,\star}$, $\star \in \{fl, fr, rl, rr\} = \{\text{front-left}, \text{front-right}, \text{rear-left}, \text{rear-right}\}$, are the longitudinal and lateral forces exchanged between tires and road, having the following expressions:

$$F_{\xi,\star} = f_P(\beta_\star)\mu_\star \frac{F_{\zeta,\star}}{F_{\zeta,\text{nom}}} \sin\delta_\star, \qquad F_{\eta,\star} = -f_P(\beta_\star)\mu_\star \frac{F_{\zeta,\star}}{F_{\zeta,\text{nom}}} \cos\delta_\star, \tag{3.134a}$$
$$\beta_{fl} = \text{atan}\left(\frac{v_\eta + l_f\omega_\psi}{v_\xi + d_f\omega_\psi}\right) - \delta_{fl}, \quad \beta_{fr} = \text{atan}\left(\frac{v_\eta + l_f\omega_\psi}{v_\xi - d_f\omega_\psi}\right) - \delta_{fr}, \tag{3.134b}$$

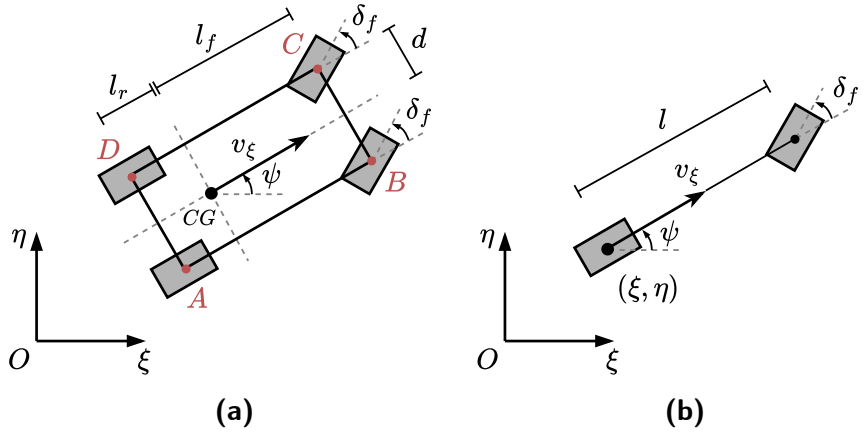


Figure 3.9. **(a)** Dynamic Dual-Track vehicle model (3.133). **(b)** Kinematic bicycle model (3.138).

$$\beta_{rl} = \operatorname{atan}\left(\frac{v_\eta - l_r\omega_\psi}{v_\xi + d_r\omega_\psi}\right) - \delta_{rl}, \qquad \beta_{rr} = \operatorname{atan}\left(\frac{v_\eta - l_r\omega_\psi}{v_\xi - d_r\omega_\psi}\right) - \delta_{rr}, \tag{3.134c}$$

where $\delta_\star$ is the wheel steering angle, $\beta_\star$ is the wheel slip angle, f_P is the Pacejka's magic formula [116], $\mu_\star$ is the wheel friction coefficient, $F_{\zeta,\star}$ is the normal force applied to the wheel, and $F_{\zeta,\text{nom}}$ is the nominal normal force applied to each wheel.

The model (3.133) considers dynamic load transfer with pitch and roll equilibrium, to effectively vary the wheel friction parameters in Eq. (3.134a); the corresponding equations for the normal forces $F_{\zeta,\star}$ are the following [97]:

$$\begin{aligned}
F_{\zeta,f} &= \frac{1}{l_f + l_r}(l_r mg - (\dot{v}_\xi - v_\eta\omega_\psi)mh + hF_{\xi,e}),\\
F_{\zeta,r} &= \frac{1}{l_f + l_r}(l_f mg + (\dot{v}_\xi - v_\eta\omega_\psi)mh - hF_{\xi,e}),\\
F_{\zeta,fl} &= F_{\zeta,f} + \frac{1}{d_f}(mh(\dot{v}_\eta + v_\xi\omega_\psi) - hF_{\eta,e}),\\
F_{\zeta,fr} &= F_{\zeta,f} + \frac{1}{d_f}(-mh(\dot{v}_\eta + v_\xi\omega_\psi) + hF_{\eta,e}),\\
F_{\zeta,rl} &= F_{\zeta,r} + \frac{1}{d_r}(mh(\dot{v}_\eta + v_\xi\omega_\psi) - hF_{\eta,e}),\\
F_{\zeta,rr} &= F_{\zeta,r} + \frac{1}{d_r}(-mh(\dot{v}_\eta + v_\xi\omega_\psi) + hF_{\eta,e}).
\end{aligned}$$

where g is the gravitational acceleration and h is the height of the vehicle CG above the axle plane.

Finally, in Eq. (3.133), $F_{\xi,e}$ and $F_{\eta,e}$ are longitudinal and lateral external forces, which are employed to introduce in the model the viscous and rolling friction forces, i.e.,

$$\begin{aligned}
F_{\xi,e} &= F_{\xi,\text{vis}} + F_{\xi,\text{rol}}, & F_{\eta,e} &= F_{\eta,\text{vis}} + F_{\eta,\text{rol}},\\
F_{\xi,\text{vis}} &= -\frac{1}{2}\rho C_\text{d} A_\text{f} v_\xi\sqrt{v_\xi^2 + v_\eta^2}, & F_{\eta,\text{vis}} &= -\frac{1}{2}\rho C_\text{d} A_\text{f} v_\eta\sqrt{v_\xi^2 + v_\eta^2},\\
F_{\xi,\text{rol}} &= \begin{cases} -C_\text{r} mg\frac{v_\xi}{\sqrt{v_\xi^2+v_\eta^2}} & \text{if } (v_\xi, v_\eta) \neq 0,\\ 0 & \text{if } (v_\xi, v_\eta) = 0, \end{cases} & F_{\xi,\text{rol}} &= \begin{cases} -C_\text{r} mg\frac{v_\eta}{\sqrt{v_\xi^2+v_\eta^2}} & \text{if } (v_\xi, v_\eta) \neq 0,\\ 0 & \text{if } (v_\xi, v_\eta) = 0, \end{cases}
\end{aligned}$$

where ρ is the air density, A_f is the vehicle frontal area, C_d and C_r are the viscous and rolling friction coefficients, respectively.

Some simplifications can be performed on model (3.133). First, only the front wheels can steer, i.e., $\delta_{fl} = \delta_{fr} = \delta_f$, $\delta_{rl} = \delta_{rr} = 0$; second, front and rear vehicle track widths are equal, i.e., $d_f = d_r = d$.

Then, the plant model (3.133) can be written in compact form as follows:

$$\dot{x} = f_{\mathcal{V}}(x, u), \tag{3.137}$$

where $x = [\xi, \eta, \psi, v_\xi, v_\eta, \omega_\psi]^\top \in \mathbb{R}^6$ is the state and $u = [a_\xi, \delta_f]^\top \in \mathbb{R}^2$ is the input.

The ego vehicle has actuation limits on the input values u, defining an admissible input set $\mathcal{U} = \{u \in \mathbb{R}^2 : a_{\xi,\mathrm{lb}} \leq a_\xi \leq a_{\xi,\mathrm{ub}},\ \delta_{f,\mathrm{lb}} \leq \delta_f \leq \delta_{f,\mathrm{ub}}\}$. The state-space manifold onto which the trajectories of system (3.137) evolve is given by $\mathcal{X} = \{x \in \mathbb{R}^6 : [\xi, \eta]^\top \in \mathcal{P},\ \psi \in \mathbb{R},\ v_{\xi,\mathrm{lb}} \leq v_\xi \leq v_{\xi,\mathrm{ub}},\ v_{\eta,\mathrm{lb}} \leq v_\xi \leq v_{\eta,\mathrm{ub}},\ \omega_{\psi,\mathrm{lb}} \leq \omega_\psi \leq \omega_{\psi,\mathrm{ub}}\}$. We see that the sets $\mathcal{X}$ and $\mathcal{U}$ are convex polytopes.

For MPC prediction, we consider a simplified model to capture the true plant behavior in Eq. (3.137). Specifically, we employ the kinematic bicycle model (Figure 3.9b), which is CT and input-affine, i.e.,

$$\begin{aligned}
\dot{x} = f(x, u) &= \begin{bmatrix} v_\xi \cos\psi \\ v_\xi \sin\psi \\ \frac{v_\xi}{l}\tan(\delta_f) \\ a_\xi \\ \omega_f \end{bmatrix} \\
&= f_0(x) + g_1(x)\underbrace{a_\xi}_{u_1} + g_2(x)\underbrace{\omega_f}_{u_2} \\
&= \begin{bmatrix} v_\xi \cos\psi \\ v_\xi \sin\psi \\ \frac{v_\xi}{l}\tan(\delta_f) \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 0 \end{bmatrix} a_\xi + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \omega_f.
\end{aligned} \tag{3.138}$$

where $x = [\xi, \eta, \psi, v_\xi, \delta_f]^\top \in \mathbb{R}^5$ is the state, $u = [a_\xi, \omega_f]^\top \in \mathbb{R}^2$ is the input, ω_f is the front wheels steering velocity, and $l = l_f + l_r$ is the vehicle wheelbase.

Since the parking maneuver is performed at low speed, the simplified model (3.138) provides a sufficiently accurate description of the vehicle motion, thus ensuring an effective control action on the plant (3.133).

The model (3.138) inherits a modified version of the admissible input set and state-space manifold of the plant (3.133), given by

$$\mathcal{U} = \{u \in \mathbb{R}^2 : a_{\xi,\mathrm{lb}} \leq a_\xi \leq a_{\xi,\mathrm{ub}},\ \omega_{f,\mathrm{lb}} \leq \omega_f \leq \omega_{f,\mathrm{ub}}\}, \tag{3.139}$$

$$\mathcal{X} = \{x \in \mathbb{R}^5 : [\xi, \eta]^\top \in \mathcal{P},\ \psi \in \mathbb{R},\ v_{\xi,\mathrm{lb}} \leq v_\xi \leq v_{\xi,\mathrm{ub}},\ \delta_{f,\mathrm{lb}} \leq \delta_f \leq \delta_{f,\mathrm{ub}}\}. \tag{3.140}$$

A DT version of model (3.138) can be obtained by applying a discretization method of choice (refer to Section 2.2, p. 12) with discrete time step $T_s > 0$, yielding

$$x_{k+1} = f_d(x_k, u_k). \tag{3.141}$$

Remark 3.12

It is worth noticing that, for the DDT model (3.133), the planar position (ξ, η) is that of the vehicle CG, i.e., $(\xi_{\text{CG}}, \eta_{\text{CG}})$; instead, for the kinematic bicycle model (3.138), the planar position (ξ, η) is that of the middle point of the rear vehicle track.

Hence, when closing the loop between controller and plant, the following transformation must be employed:

$$\begin{aligned} \xi &= \xi_{\text{CG}} - l_r \cos\psi, \\ \eta &= \eta_{\text{CG}} - l_r \sin\psi. \end{aligned} \tag{3.142}$$

Additionally, the control input ω_f in Eq. (3.138) must pass through an integrator, before being fed to the plant (3.133) as δ_f.

Obstacles In parallel parking, obstacles are represented by the parked vehicles and the road footpaths.

Nonlinear state constraints for these obstacles are formulated as suggested in [93]: we ensure that each corner of the ego vehicle (i.e., A, B, C, D in Figure 3.9a) does not collide with the parked vehicles and the footpaths.

Referring to the kinematic bicycle model (3.138) (see Remark 3.12), the planar positions of the vehicle corners are given by

$$\xi_A = \xi + l\,\cos\psi - d\,\sin\psi, \quad \eta_A = \eta + l\,\sin\psi + d\,\cos\psi, \tag{3.143a}$$

$$\xi_B = \xi + l\,\cos\psi + d\,\sin\psi, \quad \eta_B = \eta + l\,\sin\psi - d\,\cos\psi, \tag{3.143b}$$

$$\xi_C = \xi + d\,\sin\psi, \quad \eta_C = \eta - d\,\cos\psi, \tag{3.143c}$$

$$\xi_D = \xi - d\,\sin\psi, \quad \eta_D = \eta + d\,\cos\psi. \tag{3.143d}$$

Each parked vehicle (indexed by $j = 1, 2$, 1 = front vehicle, 2 = rear vehicle) is enclosed within a safety ellipsoid, defined by its center $c_j = [c_{\xi,j}, c_{\eta,j}]^\top$ and semi-axes $(l_{\xi,j}, l_{\eta,j})$.

Then, each vehicle corner in Eq. (3.143) must remain inside the road region $\mathcal{P}$, without exceeding the curbs, and must stay outside the parked vehicles safety ellipsoids. These conditions are represented by 16 nonlinear inequalities on the state x, i.e.,

$$\eta_\star(\eta, \psi) - \eta_{\text{ub}} \leq 0, \tag{3.144a}$$

$$-\eta_\star(\eta, \psi) + \eta_{\text{lb}} \leq 0, \tag{3.144b}$$

$$1 - \frac{\big(\xi_\star(\xi, \psi) - c_{\xi,j}\big)^2}{l_{\xi,j}^2} - \frac{\big(\eta_\star(\eta, \psi) - c_{\eta,j}\big)^2}{l_{\eta,j}^2} \leq 0, \tag{3.144c}$$

$$\forall \star \in \{A, B, C, D\}, \quad j = 1, 2.$$

Finally, we collect in the function $h : \mathbb{R}^{n_x} \to \mathbb{R}^{16}$ the left-hand side of the inequalities in Eq. (3.144).

NMPC Control Strategy

The classic NMPC optimal control problem for the autonomous parallel parking task is formulated as in Eq. (3.98), i.e.,

$$\min_{\hat{\boldsymbol{x}}, \hat{\boldsymbol{u}}} \quad J(\hat{\boldsymbol{x}}, \hat{\boldsymbol{u}}) \tag{3.145a}$$

$$\text{s.t.} \quad \hat{x}_0 = x_k, \tag{3.145b}$$
$$\hat{x}_{i+1} = f_d(\hat{x}_i, \hat{u}_i), \quad i = 0, \dots, N_p - 1, \tag{3.145c}$$
$$\hat{u}_i \in \mathcal{U}, \quad i = 0, \dots, N_p - 1, \tag{3.145d}$$
$$\hat{x}_i \in \mathcal{X}, \quad i = 0, \dots, N_p - 1, \tag{3.145e}$$
$$h(\hat{x}_i) \leq 0, \quad i = 0, \dots, N_p - 1. \tag{3.145f}$$

Here, the cost function (3.145a) is given by Eq. (3.98a), with reference state $x_{r,k} \in (x_r^{(n)})_{n=1}^2$, changing according to the policy (3.132). The prediction model (3.145c) is given by the DT kinematic bicycle model (3.141). The linear constraints (3.145d) and (3.145e) are given by Eqs. (3.139) and (3.140), respectively. The nonlinear state constraints (3.145f) are given by Eq. (3.144), modeling the obstacles.

Koopman NMPC Control Strategy

Now, we begin the construction of the K-NMPC control policy, by lifting the NMPC problem (3.145) in the Koopman space, as detailed in Section 3.5.

Nonlinear State Constraints and Initial Set of Observables Recalling the nonlinear state constraint expressions in Eq. (3.144) for the parked vehicles and the footpaths, we can expand them following Eq. (3.99); defining $a_j = 1/l_{\xi,j}^2$ and $b_j = 1/l_{\eta,j}^2$, we obtain

$$\begin{bmatrix} 0 & 1 & 0 & 0 & 0 & d & l \\ 0 & 1 & 0 & 0 & 0 & -d & l \\ 0 & 1 & 0 & 0 & 0 & -d & 0 \\ 0 & 1 & 0 & 0 & 0 & d & 0 \end{bmatrix} \begin{bmatrix} x \\ \hline \cos\psi \\ \sin\psi \end{bmatrix} - \begin{bmatrix} \eta_{\text{lb}} \\ \eta_{\text{lb}} \\ \eta_{\text{lb}} \\ \eta_{\text{lb}} \end{bmatrix} \leq 0, \tag{3.146a}$$

$$\begin{bmatrix} 0 & -1 & 0 & 0 & 0 & -d & -l \\ 0 & -1 & 0 & 0 & 0 & d & -l \\ 0 & -1 & 0 & 0 & 0 & d & 0 \\ 0 & -1 & 0 & 0 & 0 & -d & 0 \end{bmatrix} \begin{bmatrix} x \\ \hline \cos\psi \\ \sin\psi \end{bmatrix} - \begin{bmatrix} -\eta_{\text{ub}} \\ -\eta_{\text{ub}} \\ -\eta_{\text{ub}} \\ -\eta_{\text{ub}} \end{bmatrix} \leq 0, \tag{3.146b}$$

$$\begin{bmatrix} 2a_jc_{\xi,j} & 2a_jc_{\xi,j} & 2a_jc_{\xi,j} & 2a_jc_{\xi,j} \\ 2b_jc_{\eta,j} & 2b_jc_{\eta,j} & 2b_jc_{\eta,j} & 2b_jc_{\eta,j} \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ -a_j & -a_j & -a_j & -a_j \\ -b_j & -b_j & -b_j & -b_j \\ 2la_jc_{\xi,j}+2b_jc_{\eta,j}d & -2b_jc_{\eta,j}d+2la_jc_{\xi,j} & -2b_jc_{\eta,j}d & 2b_jc_{\eta,j}d \\ -2a_jc_{\xi,j}d+2lb_jc_{\eta,j} & 2lb_jc_{\eta,j}+2a_jc_{\xi,j}d & 2a_jc_{\xi,j}d & -2a_jc_{\xi,j}d \\ -a_jl^2-b_jd^2 & -a_jl^2-b_jd^2 & -b_jd^2 & -b_jd^2 \\ -b_jl^2-a_jd^2 & -b_jl^2-a_jd^2 & -a_jd^2 & -a_jd^2 \\ -2lb_jd+2la_jd & -2la_jd+2lb_jd & 0 & 0 \\ -2la_j & -2la_j & 0 & 0 \\ 2a_jd & -2a_jd & -2a_jd & 2a_jd \\ -2b_jd & 2b_jd & 2b_jd & -2b_jd \\ -2lb_j & -2lb_j & 0 & 0 \end{bmatrix}^\top \begin{bmatrix} x \\ \hline \xi^2 \\ \eta^2 \\ \cos\psi \\ \sin\psi \\ \cos^2\psi \\ \sin^2\psi \\ \cos\psi\sin\psi \\ \xi\cos\psi \\ \xi\sin\psi \\ \eta\cos\psi \\ \eta\sin\psi \end{bmatrix}$$
$$- \begin{bmatrix} -1+a_jc_{\xi,j}^2+b_jc_{\eta,j}^2 \\ -1+a_jc_{\xi,j}^2+b_jc_{\eta,j}^2 \\ -1+a_jc_{\xi,j}^2+b_jc_{\eta,j}^2 \\ -1+a_jc_{\xi,j}^2+b_jc_{\eta,j}^2 \end{bmatrix} \leq 0, \quad j = 1, 2. \tag{3.146c}$$

We see that, overall, the nonlinear state constraints introduce 11 nonlinear terms, which are collected as

$$\sigma(x) = [\xi^2,\ \eta^2,\ \cos\psi,\ \sin\psi,\ \cos^2\psi,\ \sin^2\psi,\ \cos\psi\sin\psi,\ \xi\cos\psi,\ \xi\sin\psi,\ \eta\cos\psi,\ \eta\sin\psi]^\top.$$

Then, as discussed in Section 3.5.2, the initial set of observables Φ_{in} is given by

$$\begin{aligned}\Phi_{\text{in}} = \{&\xi,\ \eta,\ \psi,\ v_\xi,\ \delta_f,\ \xi^2,\ \eta^2,\ \cos\psi,\ \sin\psi,\ \cos^2\psi,\ \sin^2\psi,\\ &\cos\psi\sin\psi,\ \xi\cos\psi,\ \xi\sin\psi,\ \eta\cos\psi,\ \eta\sin\psi\},\quad |\Phi_{\text{in}}| = 16,\end{aligned} \tag{3.147}$$

comprising the states of system (3.138) and the nonlinear terms introduced by constraints (3.144).

Observables Generation and Lifted System From Φ_{in} (3.147), the basis of observables Φ and the related Koopman lifted system – lifting both the kinematic bicycle model (3.138) and the nonlinear state constraints (3.144) – are generated through Algorithms 3.3 and 3.4.

In this case, Algorithm 3.3 does not reach convergence, generating a infinite-dimensional basis Φ. We report the generated basis up to the 30^{th} observable, i.e.,

$$\begin{aligned}\Phi = \Phi_{\text{in}} \cup \{&\psi\cos v_\xi,\ \psi\sin v_\xi,\ \psi\tan\theta_f,\ \xi\psi\cos v_\xi,\ \eta\psi\sin v_\xi,\ \psi\sin v_\xi\tan\theta_f,\\ &\psi\cos v_\xi\tan\theta_f,\ \psi\cos v_\xi\sin v_\xi\tan\theta_f,\ \psi\cos^2 v_\xi\tan\theta_f,\ \psi\sin^2 v_\xi\tan\theta_f,\\ &\psi\cos^2 v_\xi,\ \xi\psi\sin v_\xi\tan\theta_f,\ \psi\cos v_\xi\sin v_\xi,\ \xi\psi\cos v_\xi\tan\theta_f,\ \cdots\}.\end{aligned} \tag{3.148}$$

Therefore, system (3.138), with nonlinear state constraints (3.144), does not admit a finite-dimensional Koopman representation.

As well, Algorithm 3.4 generates an infinite-dimensional bilinear Koopman lifted system,

$$\dot{z} = Az + B_0u + B_1zu_1 + B_2zu_2. \tag{3.149}$$

The structure of the Koopman lifted system (3.149) matches Eq. (3.43), since system (3.138) contains input-linear terms.

Dimensionality Reduction Since system (3.138), (3.144) does not admit a finite-dimensional lifted system, we have to reduce it to a suitable finite dimension.

To determine the "best" dimension reduction, we follow the procedure detailed in Section 3.4.1: we consider a set of trajectories $x(t)$ of system (3.138) and we evaluate the prediction accuracy of the reduced Koopman lifted system for increasing number of observables ϕ.

Each state trajectory $x(t) = [\xi(t), \eta(t), \psi(t), v_\xi(t), \delta_f(t)]^\top$ is computed over the time interval $t \in [0, T]$, with initial condition $x(0) = [0\text{ m},\ 0\text{ m},\ 0\text{ rad},\ 0\text{ m s}^{-1},\ 0\text{ rad}]^\top$, applying an input signal belonging to the following family:

$$u(t) = \begin{bmatrix} a_{\xi,0}\sin\left(2\pi\frac{t}{n_1T} + \varphi_1\right) \\ \omega_{f,0}\sin\left(2\pi\frac{t}{n_2T} + \varphi_2\right)\end{bmatrix}, \tag{3.150}$$

with parameters $a_{\xi,0}$, $\omega_{f,0}$, $n_1, n_2 \in \mathbb{R}_{>0}$, and $\varphi_1, \varphi_2 \in [-\pi, \pi]$ rad. The family of input signals (3.150) is designed to ensure a sufficiently rich excitation of the system, yielding representative state trajectories for evaluating the prediction accuracy.

The trajectory $x(t)$ is sampled at N_{pred} equally-spaced time instants $t_n = \frac{n}{N_{\text{pred}}}T$, extracting the states $x_n = x(t_n)$, $n = 1, \dots, N_{\text{pred}}$. Taking each lifted state $z_n = \phi(x_n)$ as an initial condition, a prediction $z^{(n)}_{\text{pred}}(\tau)$ of the reduced lifted system is computed, over the prediction time interval $\tau \in [0, T_p]$, applying the input $u(t)$.

The lifted trajectory $z(t) = \phi(x(t))$ and the predictions $z_{\text{pred}}(t)$ are then expressed as

$$z(t) = \phi(x(t)) = \begin{bmatrix} z_x(t) \\ z_\sigma(t) \\ z_\psi(t) \end{bmatrix}, \quad z_{\text{pred}}(t) = \begin{bmatrix} z_{x,\text{pred}}(t) \\ z_{\sigma,\text{pred}}(t) \\ z_{\psi,\text{pred}}(t) \end{bmatrix}. \tag{3.151}$$

The true and predicted trajectories (3.151) are compared by sampling each of them with sampling time T_s. Specifically, each prediction time interval will contain $N_p + 1$ sampling points, with $N_p = \frac{T_p}{T_s}$.

For each n-th predicted trajectory, three prediction errors are considered:

1) Prediction error $\epsilon_{\text{pred},1}$ between the original trajectory $x(t)$ and the predicted lifted states $z_{x,\text{pred}}(t)$, associated with the original states x, i.e.,

$$\epsilon^{(n)}_{\text{pred},1} = \frac{1}{N_p} \sum_{i=1}^{N_p} \left\| z^{(n)}_{x,\text{pred}}(iT_s) - x(t_n + iT_s) \right\|_\infty,$$

2) Prediction error $\epsilon_{\text{pred},2}$ between the true lifted states $z_\sigma(t)$ and the predicted lifted states $z_{\sigma,\text{pred}}(t)$, associated with the constraint nonlinearities $\sigma(x)$, i.e.,

$$\epsilon^{(n)}_{\text{pred},2} = \frac{1}{N_p} \sum_{i=1}^{N_p} \left\| z^{(n)}_{\sigma,\text{pred}}(iT_s) - z_\sigma(t_n + iT_s) \right\|_\infty,$$

3) Prediction error $\epsilon_{\text{pred},3}$, combining the previous ones, i.e.,

$$\epsilon^{(n)}_{\text{pred},3} = \frac{1}{N_p} \sum_{i=1}^{N_p} \left\| \begin{bmatrix} z^{(n)}_{x,\text{pred}}(iT_s) \\ z^{(n)}_{\sigma,\text{pred}}(iT_s) \end{bmatrix} - \begin{bmatrix} x(t_n+iT_s) \\ z_\sigma(t_n+iT_s) \end{bmatrix} \right\|_\infty.$$

All the above calculations are repeated for increasing number of observables $\phi(x) = [\phi_i(x)]_{i=1}^{\overline{N}_o}$, with $\overline{N}_o = 16, \dots, 30$.

The prediction accuracy has been evaluated on a set of 10 trajectories $x(t)$, generated by randomly selecting the parameters of the input signal $u(t)$ in Eq. (3.150) as follows:

$$\begin{aligned} &a_{\xi,0} \sim U([1, 10]\ \text{m s}^{-2}), && n_1 \sim U([0.25, 1]), && \varphi_1 = -\pi/2\ \text{rad}, \\ &\omega_{f,0} \sim U([\pi/6, \pi]\ \text{rad s}^{-1}), && n_2 \sim U([0.25, 1]), && \varphi_2 = 0\ \text{rad}, \end{aligned}$$

where $U(\cdot)$ is the uniform probability distribution.

Each trajectory $x(t)$ and predictions $z_{\text{pred}}(t)$ are computed with the Runge-Kutta 4(5) integration method (with tolerances $\epsilon_{\text{abs}} = \epsilon_{\text{rel}} = 1 \times 10^{-6}$).

Additional data is as follows: $T = 5$ s, $T_s = 0.025$ s, $N_{\text{pred}} = 10$, $T_p = 0.75$ s ($N_p = 30$).

Results are reported in Figure 3.10.

First, we can observe that all prediction errors are non-increasing as the number of observables increases, which aligns with our theoretical expectations.

For what concerns the prediction accuracy of the original trajectory $x(t)$ only (i.e., disregarding the constraint nonlinearities $\sigma(x(t))$), given by $\epsilon_{\text{pred},1}$ (•, — in Fig. 3.10), a reduction in the prediction error is observed up to 19 observables; then, the value settles.

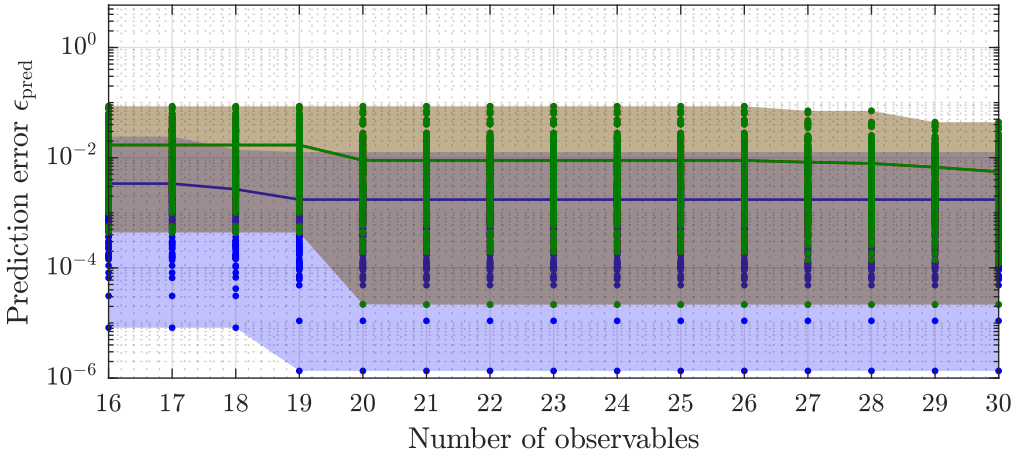


Figure 3.10. Prediction errors of the reduced Koopman lifted system for the kinematic bicycle model (3.138) with nonlinear state constraints (3.144), for increasing number of observables ϕ: single predictions ($\epsilon_{\text{pred},1}$ •, $\epsilon_{\text{pred},2}$ •, $\epsilon_{\text{pred},3}$ •); average values ($\bar{\epsilon}_{\text{pred},1}$ —, $\bar{\epsilon}_{\text{pred},2}$ —, $\bar{\epsilon}_{\text{pred},3}$ —).

Therefore, in absence of nonlinear constraints (3.144), the lifted system dimension can be reduced to $n_z = 19$.

Focusing instead on the prediction accuracy of the constraint nonlinearities $\sigma(x(t))$ only, given by $\epsilon_{\text{pred},2}$ (•, — in Fig. 3.10)[2], a steep reduction in the prediction error occurs once the number of observables reaches 20; then, a further slight reduction is observed from the 26^{th} observable.

Therefore, to ensure that the lifted system accurately captures the constraint nonlinearities – so that the closed-loop system, controlled by K-NMPC, reliably satisfies the nonlinear constraints (3.144) – its dimension can be reduced to no less than $n_z = 20$.

The prediction error $\epsilon_{\text{pred},3}$ (•, — in Fig. 3.10) sums up all the above observations, allowing us to conclude that we can confidently reduce the dimension of the Koopman lifted system to $n_z = 20$.

K-NMPC Formulation To write the K-NMPC optimal control problem for the autonomous parallel parking task, the following two final steps are performed:

- As K-NMPC prediction model, we consider the DT LPV/APV form of the reduced version of the Koopman lifted system (3.149) (refer to Sections 3.5.3 and 3.5.5), i.e.,

$$\begin{aligned}\hat{z}_{i+1} &= A_d(\overline{z}_{k,i}, \overline{u}_{k,i})\hat{z}_i + B_d(\overline{z}_{k,i}, \overline{u}_{k,i})\hat{u}_i + b_d(\overline{z}_{k,i}, \overline{u}_{k,i}) \\ &= \bar{A}_{k,i}\hat{z}_i + \bar{B}_{k,i}\hat{u}_i + \bar{b}_{k,i}.\end{aligned}$$

- We construct the lifted constraints (3.105d) by stacking together the expressions (3.146) of the parked vehicles and footpaths, and considering their DT version; we then obtain the constraint matrices C_d and d_d.

Then, the K-NMPC problem is formulated as follows, for each $k \geq 0$:

$$\min_{\hat{\boldsymbol{z}},\hat{\boldsymbol{u}}} \quad J(\hat{\boldsymbol{z}}_x, \hat{\boldsymbol{u}}) \tag{3.153a}$$

$$\text{s.t.} \quad \hat{x}_0 = x_k, \tag{3.153b}$$

[2]Note that, in Figure 3.10, the red (—) and green (—) plots are almost coincident.

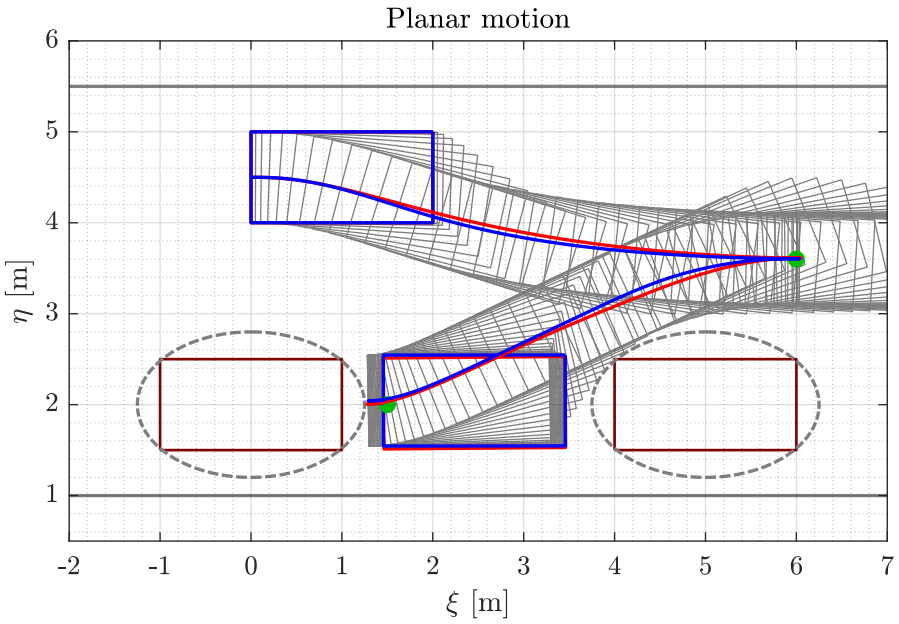


Figure 3.11. Planar motion of the ego vehicle while performing the parking maneuver: ego vehicle closed-loop trajectories (K-NMPC ——; NMPC ——); reference positions $x_r = [\xi_r, \eta_r, \psi_r]^\top$ (●); footpaths and parked vehicles within their safety ellipsoids. For the K-NMPC trajectory (——), the instantaneous positions of the ego vehicle are also shown.

$$\hat{z}_{i+1} = \bar{A}_{k,i}\hat{z}_i + \bar{B}_{k,i}\hat{u}_i + \bar{b}_{k,i}, \quad i = 0, \dots, N_p - 1, \tag{3.153c}$$

$$\hat{u}_i \in \mathcal{U}, \quad i = 0, \dots, N_p - 1, \tag{3.153d}$$

$$\hat{z}_{x,i} \in \mathcal{X}, \quad i = 0, \dots, N_p - 1, \tag{3.153e}$$

$$C_d \begin{bmatrix} \hat{z}_{x,i} \\ \hat{z}_{\sigma,i} \end{bmatrix} \leq d_d, \quad i = 0, \dots, N_p - 1. \tag{3.153f}$$

Implementation Details

In the following, we report some details on the implementation of this case study.

Simulations Simulations are performed in MATLAB® (ver. 2023b), on a machine powered by a 13th Gen Intel Core™ i7 CPU at 1.7 GHz, with 16 GB of RAM.

The K-NMPC optimal control problem (3.153) is formulated with YALMIP [96] and solved with the QP interior-point solver MOSEK (ver. 10.1) [108].

The classic NMPC optimal control problem (3.145) is formulated with CasADi [8] and solved with the NLP interior-point solver Ipopt [153].

Data Relevant data used in both simulation and experimental validation is reported in the following:

- General data: $T_s = 25$ ms.
- Ego vehicle: $x_0 = [0\text{ m},\ 4.5\text{ m},\ 0\text{ deg},\ 0\text{ m s}^{-1},\ 0\text{ deg}]^\top$.
  - Kinematic bicycle model: $l = 2$ m, $d = 0.5$ m.
  - Dynamic Dual-Track model: $m = 1500$ kg, $l_r = 0.75$ m, $l_f = 1.25$ m, $h = 0.25$ m, $I_z = 4 \times 10^3$ kg m^2.

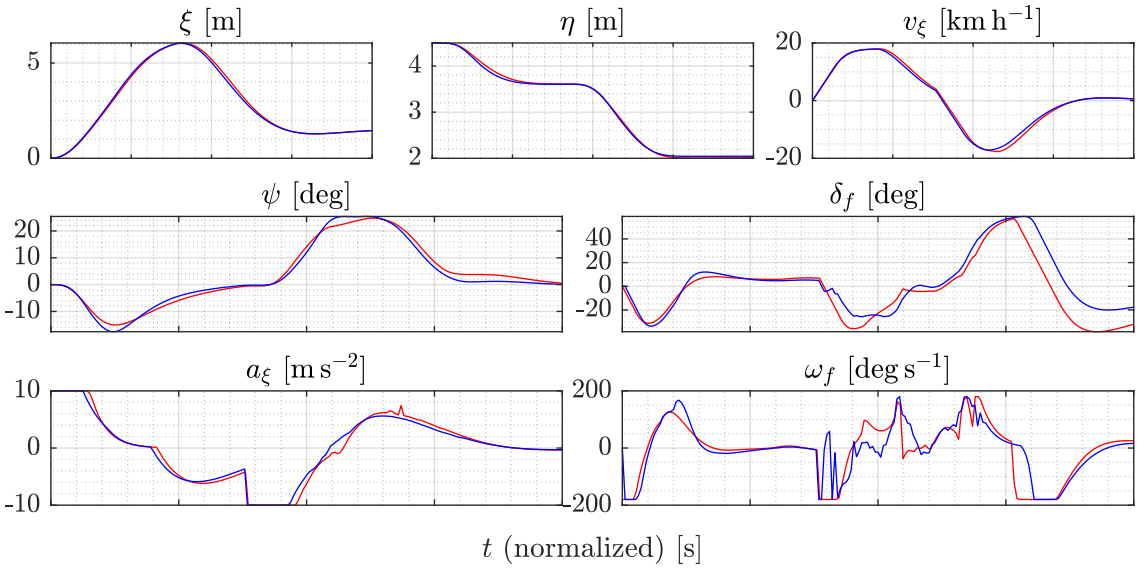


Figure 3.12. Closed-loop state and input trajectories of the ego vehicle while performing the parking maneuver; a normalized time axis is used to better compare the trajectories, as they reach the reference in slightly different times: K-NMPC — ($t \in [0, 4.75]$ s); NMPC — ($t \in [0, 4.60]$ s).

- MPC: $N_p = 30$, $x_r^{(1)} = [6\text{ m},\ 3.6\text{ m},\ 0\text{ deg},\ 0\text{ m s}^{-1},\ 0\text{ deg}]^\top$, $x_r^{(2)} = [1.5\text{ m},\ 2\text{ m},\ 0\text{ deg},\ 0\text{ m s}^{-1},\ 0\text{ deg}]^\top$, $\epsilon_r = [0.05\text{ m},\ 0.1\text{ m},\ 2\text{ deg},\ +\infty,\ +\infty]^\top$, $Q = \text{diag}([10, 50, 10, 10^{-1}, 10^{-3}]^\top)$, $R = \text{diag}([10^{-3}, 10^{-3}]^\top)$, $Q_\Delta = \text{diag}([10^{-1}, 10^{-1}, 10^{-1}, 10^{-1}, 10^{-1}]^\top)$, $R_\Delta = \text{diag}([1, 10^{-3}]^\top)$, $\mathcal{U} = \{u \in \mathbb{R}^2 : u_{\text{lb}} \leq u \leq u_{\text{ub}}\}$, $u_{\text{lb}} = [-10\text{ m s}^{-2},\ -\pi\text{ rad s}^{-1}]^\top$, $u_{\text{ub}} = [10\text{ m s}^{-2},\ \pi\text{ rad s}^{-1}]^\top$, $\mathcal{X} = \{x \in \mathbb{R}^5 : x_{\text{lb}} \leq x \leq x_{\text{ub}}\}$, $x_{\text{lb}} = [-2\text{ m},\ 0.5\text{ m},\ -\infty,\ -5\text{ m s}^{-1},\ -\pi/3\text{ rad}]^\top$, $x_{\text{ub}} = [7\text{ m},\ 6\text{ m},\ +\infty,\ 5\text{ m s}^{-1},\ \pi/3\text{ rad}]^\top$.
- Obstacles:
  - Footpaths: $\eta_{\text{lb}} = 1$ m, $\eta_{\text{ub}} = 5.5$ m.
  - Parked vehicles: $c_1 = [0, 2]^\top$ m, $l_{\xi,1} = 1.25$ m, $l_{\eta,1} = 0.8$ m, $c_2 = [5, 2]^\top$ m, $l_{\xi,1} = 1.25$ m, $l_{\eta,1} = 0.8$ m.

Results

Figure 3.11 reports the ego vehicle planar motion while performing the parking maneuver. The ego vehicle closed-loop planar trajectories have been obtained in simulation using: K-NMPC with dimension $n_z = 20$ (— in Fig. 3.11), in accordance with the results obtained in Section 3.6.2, p. 110; classic NMPC (— in Fig. 3.11).

Figure 3.12 reports the closed-loop state and input trajectories of the ego vehicle as functions of the continuous time t. Depending on which control strategy is used, the ego vehicle completes the parking maneuver at slightly different times; therefore, Figure 3.12 employs a normalized time axis to allow for a better visual comparison of the trajectories. The control time interval of each strategy is the following: K-NMPC, $t \in [0, 4.75]$ s; NMPC, $t \in [0, 4.60]$ s.

From Figure 3.11, we see that K-NMPC manages to successfully perform the whole parking maneuver, effectively avoiding collisions with the parked vehicles and footpaths.

From Figure 3.12, we can also observe that the K-NMPC and NMPC trajectories are remarkably similar. This result is coherent with Eq. (3.107) and substantiates the reliability of K-NMPC (3.131) in providing a control action closely matching that of the original unlifted NMPC problem (3.122). This also shows that the K-NMPC prediction model (3.110b), based on the LPV/APV form of the lifted system and evaluated along the previous optimal predicted trajectory, provides an accurate approximation of the true bilinear Koopman lifted system. Finally, this result further proves the reliability of the dimensionality reduction procedure described in Sections 3.4.1 and 3.6.2, p. 110.

Execution Times Execution times have been collected over 20 simulations. Data is reported in Figure 3.13, as function of the continuous time t; minimum, maximum, and average execution times are summarized in Table 3.2.

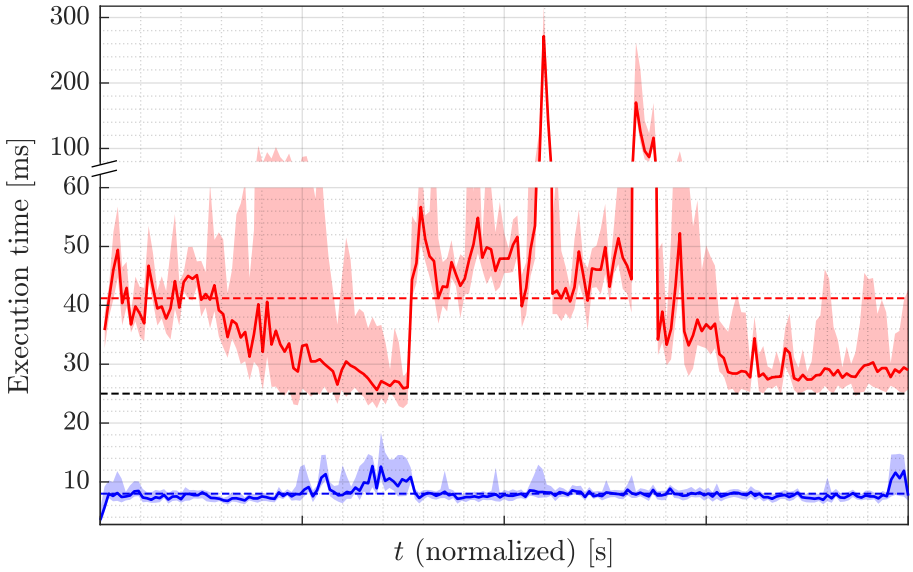


Figure 3.13. Execution times for the autonomous parallel parking: K-NMPC (envelope, average, cumulative average); NMPC (envelope, average, cumulative average); discrete time step $T_s = 25$ ms.

	Min	Average	Max
NMPC	22.50 ms	41.21 ms	317.34 ms
K-NMPC	2.77 ms	8.00 ms	18.39 ms

Table 3.2. Execution times for the autonomous parallel parking: minimum, maximum and average values.

We can see that K-NMPC achieves a remarkable ten-fold reduction in the execution time compared to classic NMPC. Additionally, the execution time of K-NMPC always stays below the discrete time step T_s, unlike NMPC which almost always exceeds it.

3.7. Chapter Summary

In this chapter, we presented a comprehensive analytical framework to efficiently solve NMPC problems in the Koopman lifted space. We formulated a systematic procedure to

iteratively derive a basis of observables for generic nonlinear systems. A general approach to reduce the dimension of the resulting lifted system was also presented, to handle the case of infinite dimensionality. This approach was employed to lift a general NMPC problem, including its constraints, obtaining an equivalent QP-MPC problem in the Koopman space (K-NMPC), which closely approximates the original NMPC solution. Our K-NMPC approach was validated on two real-world case studies, showcasing the proficiency of K-NMPC in attaining the control tasks, providing closed-loop trajectories that are very close to the NMPC ones, and outperforming NMPC execution time by over an order of magnitude.

4

Fully-Analog Circuital Implementation of MPC With Application to Buck Converters

4.1. Introduction

Owing to its remarkable versatility, Linear-Quadratic Model Predictive Control (QP-MPC) is a well-established control framework that is highly attractive for a wide range of practical applications.

Despite its desirable features, the practical implementation of QP-MPC is not straightforward. First, typical implementations of QP-MPC rely on digital hardware, which imposes a significant overhead in terms of costs and circuit area. This is caused by the need for expensive digital components, including analog-to-digital converters (ADC), digital-to-analog converters (DAC), and a computing device, making the overall system too expensive for mass production, where a limited-budget design is often a primary requirement. Second, despite the rapid advancement in the development of novel algorithms for convex optimization [35, 37], solving online a quadratic programming (QP) optimal control problem at each time step is impractical for systems exhibiting fast dynamics, for which a very small sampling time is required, or in presence of digital hardware with limited computational resources.

In this context, a popular approach, entailing considerable computational savings, consists in reformulating QP-MPC as a static state-feedback control policy, instead of a QP problem to be solved online. Such a static control policy is proved to have a closed-form explicit expression [18]. For this reason, this approach is referred to as Explicit Model Predictive Control (Ex-MPC). The Ex-MPC control policy exhibits a piecewise-affine (PWA) structure, defined over a collection of polytopic regions [18] and, thanks to its static nature, can be evaluated with negligible computation time. This obviates the need for online optimization to solve the QP-MPC problem. Since the pioneering work [17], several algorithms have been proposed to efficiently compute the Ex-MPC control policy [121, 13, 113]. However, the Ex-MPC approach still raises criticalities in budget-constrained and fast-sampled applications. On the one hand, the need for digital

hardware persists; on the other hand, in many practical scenarios, the PWA policy is defined over a large number of polytopic regions, whose real-time evaluation may remain critical [14]. These issues are present in, e.g., the work [158], which applies Ex-MPC to Buck converters.

A viable alternative is brought by analog implementations. A notable result in this direction is reported in [152], which constructs an analog circuit that solves linear and quadratic optimization problems, thus directly applicable to QP-MPC implementations. However, the applications of this method are limited by the complexity of the resulting circuit.

In this chapter, we present a novel, completely general methodology, for designing fully-analog electronic circuits implementing Ex-MPC policies. In the following, we shall refer to our methodology as *Analog Circuital Ex-MPC* [124]. Staring from a general QP-MPC problem, we begin by deriving the corresponding Ex-MPC policy. Then, we introduce a set of techniques to reduce the complexity of the obtained policy, namely reducing the number of polytopic regions and the number of linear inequalities defining them. Finally, we proceed with the actual circuital implementation of the Ex-MPC policy, using only commercially-available low-latency analog components: resistors, capacitors, comparators, operational amplifiers (OP-AMPs), logic gates, and one multiplexer (MUX).

Our Analog Circuital Ex-MPC approach is applied to the case study of DC-DC Buck converters control. Buck converters are devices used to regulate a high, possibly fluctuating, supply voltage to a lower and constant value. The applications of Buck converters range from biomedical ones [119] to renewable energy generation [154, 41], electric mobility [122], and consumer electronics. The accurate regulation of the output voltage in Buck converters is fundamental for the proper operation of the devices supplied by the converter, which otherwise may be damaged or not work properly. However, the tracking accuracy of the converter is hindered by unpredictable and sudden variations of the input supply voltage (line disturbances) and of the current drawn by the load (load disturbances) [50, 125]. To improve the tracking performance, several approaches have been considered. As to line disturbance attenuation, we mention the popular feedforwarding technique [6]. In contrast, the problem of attenuating load disturbance effects is more challenging. It has been partially addressed by methods based on the design of additional circuitry to modify either the feedback signal or the pulse width of the Pulse Width Modulated (PWM) signal, in response to load variations [89, 68, 73]. However, these solutions lack solid theoretical foundations. Other approaches aim at estimating the output current disturbance to apply a proper feedforward action. Among these, we mention the extended state observer [80], generalized proportional-integral observer [156], and the load estimator-compensator [125] approaches. These solutions are often combined with optimal control strategies, e.g., [80] employs linear quadratic Gaussian control in conjunction with the extended state observer, and [125] employs $\mathcal{H}_\infty$ optimal control in conjunction with the load-estimator-compensator.

For Buck converters, MPC is particularly compelling thanks to the possibility of directly handling the constraints imposed by the input duty-cycle, which can only range between 0 (0%) and 1 (100%). Furthermore, other predictive approaches to disturbance rejection, such as [160], showed promising results. The application of MPC to Buck converters has been explored in several contributions [100, 95, 3, 42]. These works report successful results on practical implementations, but, in all cases, the sampling frequency must be kept small

to guarantee convergence of the optimization algorithm. [42], [100], and [3] report switching frequencies limited to 10 kHz, 20 kHz, and 25 kHz, respectively. Similar considerations apply to [95], which employs a field-programmable gate array (FPGA) to implement the MPC.

To apply our Analog Circuital Ex-MPC approach to DC-DC Buck converters, we first introduce a discrete-time mathematical model of the device and its linearization. Next, to achieve effective load disturbance rejection, we employ the load estimator proposed in [125], which allows for a cheap analog implementation. Finally, we resort to our fully-analog circuital implementation of Ex-MPC. Compared to previous works on Ex-MPC for Buck converters, e.g., [158], we achieve greatly improved results in terms of Ex-MPC complexity reduction, ensuring a cheap circuit design. Also, our design strategy leads to a less expensive and faster implementation compared to the approach proposed in [152]. Furthermore, we provide formal guarantees on the local robust stability of our closed-loop control system, based on contraction theory [28].

Finally, we perform a comprehensive simulation study to verify the feasibility and effectiveness of our Analog Circuital Ex-MPC approach. Our analog implementation enables high-frequency sampling at 500 kHz, which is compatible with modern Buck design and considerably improves previous solutions [100, 95, 3, 42]. We analyze, through high-level Monte Carlo simulations, the robust control performance in presence of parametric uncertainty on the nominal value of Buck converter passive components, considering both the cases of ceramic and electrolytic capacitor. Next, we perform accurate low-level circuit simulations, employing the manufacturer models of commercially-available components. These simulations allow to assess the feasibility of the proposed approach in real-world conditions, which accounts for circuit non-idealities, such as the finite gain-bandwidth product of OP-AMPs, and the response delay of comparators, logic gates, and the MUX. Overall, numerical results demonstrate the effectiveness of the proposed analog Ex-MPC circuit in controlling the Buck converter, achieving a remarkable line and load disturbance rejection performance, that outclasses conventional methods.

4.1.1. Outline

This chapter is organized as follows. Section 4.2 introduces the Linear-Quadratic MPC problem formulation. Section 4.3 contains our main contributions: first, we derive the Ex-MPC static control policy; second, we apply a set of complexity-reduction strategies to reduce the number of Ex-MPC polytopic regions and the number of linear inequalities defining them; third, we describe the implementation of the Ex-MPC policy as a fully-analog electronic circuit. Section 4.4 introduces the application case study on the control of Buck converters. Specifically, we first derive the mathematical model of the Buck converter. Then, we design the estimators for states and disturbances, along with their circuital implementation. Finally, we tailor the Ex-MPC policy for Buck converter control and establish local robust stability guarantees, leveraging contraction theory. Section 4.5 validates our Analog Circuital Ex-MPC approach through extensive simulations, both high-level and circuit-level ones, to assess the robust performance in presence of parametric uncertainty, and the control performance in a high-fidelity circuit simulation environment.

4.1.2. Related Works

This chapter is based, in part, on the following work:

- S. Pirrera, L. Calogero, F. Gabriele, D. Regruto, A. Rizzo, and G. Setti, "A Fully Analog Implementation of Model Predictive Control with Application to Buck Converters," *arXiv:2509.05463 [eess.SY]*, Sep. 2025.

4.2. MPC Problem Formulation

Let us consider a discrete-time (DT) affine system, i.e.,

$$x_{k+1} = Ax_k + Bu_k + B_\nu \nu_k + b, \tag{4.1a}$$
$$y_k = Cx_k + Du_k + D_\nu \nu_k + d, \tag{4.1b}$$

where $x \in \mathbb{R}^{n_x}$, $u \in \mathbb{R}^{n_u}$, and $y \in \mathbb{R}^{n_y}$ are the state, input, and output vectors, respectively, while $\nu \in \mathbb{R}^{n_\nu}$ is a vector of exogenous disturbances. With respect to Eq. (2.119), we have introduced the affine output equation (4.1b).

System (4.1) is subject to the following linear constraints, at all time instants $k \geq 0$:

$$x_k \in \mathcal{X} = \{x \in \mathbb{R}^{n_x} : H_x x \leq h_x\} \subseteq \mathbb{R}^{n_x}, \tag{4.2a}$$
$$u_k \in \mathcal{U} = \{u \in \mathbb{R}^{n_u} : H_u u \leq h_u\} \subseteq \mathbb{R}^{n_u}, \tag{4.2b}$$
$$y_k \in \mathcal{Y} = \{y \in \mathbb{R}^{n_y} : H_y y \leq h_y\} \subseteq \mathbb{R}^{n_y}. \tag{4.2c}$$

For system (4.1), we formulate a Linear-Quadratic MPC (QP-MPC) problem (Section 2.3), which, at each time instant $k \geq 0$, provides the optimal control input u_k by solving the following QP optimal control problem:

$$\min_{\hat{\boldsymbol{x}},\hat{\boldsymbol{u}},\hat{\boldsymbol{y}}} \quad J(\hat{\boldsymbol{y}}, \hat{\boldsymbol{u}}) = \sum_{i=0}^{N_p-1} \left(\|\hat{y}_i - y_r\|_Q^2 + \|\hat{u}_i - u_r\|_R^2 \right) + \sum_{i=1}^{N_p-1} \|\hat{u}_i - \hat{u}_{i-1}\|_{R_\Delta}^2 \tag{4.3a}$$
$$\text{s.t.} \quad \hat{x}_0 = x_k, \tag{4.3b}$$
$$\hat{x}_{i+1} = A\hat{x}_i + B\hat{u}_i + B_\nu \nu_k + b, \quad i = 0, \ldots, N_p - 1, \tag{4.3c}$$
$$\hat{y}_i = C\hat{x}_i + D\hat{u}_i + D_\nu \nu_k + d, \quad i = 0, \ldots, N_p - 1, \tag{4.3d}$$
$$H_x \hat{x}_i \leq h_x, \quad i = 0, \ldots, N_p - 1, \tag{4.3e}$$
$$H_u \hat{u}_i \leq h_u, \quad i = 0, \ldots, N_p - 1, \tag{4.3f}$$
$$H_y \hat{y}_i \leq h_y, \quad i = 0, \ldots, N_p - 1. \tag{4.3g}$$

With respect to problem (2.121), $(\hat{y}_0, \hat{y}_1, \ldots, \hat{y}_{N_p-1}) = (\hat{y}_i)_{i=0}^{N_p-1} = \hat{\boldsymbol{y}}$ is the predicted output trajectory, whose samples act as additional decision variables of the problem.

The QP-MPC problem (4.3) is formulated to attain regulation towards the admissible output $y_r \in \mathcal{Y}$. In particular, the triple (x_r, u_r, y_r) is an admissible equilibrium of the undisturbed system (4.1), i.e., with $\nu_k = \mathbf{0}_{n_\nu}$, it holds that

$$x_r = Ax_r + Bu_r + b, \tag{4.4a}$$
$$y_r = Cx_r + Du_r + d. \tag{4.4b}$$

The cost function J in Eq. (4.3a) is composed of three terms:

- The quadratic regulation terms $\|\hat{y}_i - y_r\|_Q^2$ and $\|\hat{u}_i - u_r\|_R^2$ for the predicted output and input, respectively, towards the reference (y_r, u_r), with $Q \succeq 0$ and $R \succ 0$.
- The quadratic term $\|\hat{u}_i - \hat{u}_{i-1}\|_{R_\Delta}^2$, with $R_\Delta \succeq 0$, which penalizes the variation in time of the inputs, to obtain smoother predicted trajectories (see, e.g., [32, 33]).

By assuming that the states x_k and disturbances ν_k are available for measurement at all time instants $k \geq 0$, let us collect them as

$$p_k = \begin{bmatrix} x_k \\ \nu_k \end{bmatrix} \in \mathbb{R}^{n_p}, \quad n_p = n_x + n_\nu, \tag{4.5}$$

where p_k is the vector of variables acting as parameters of the QP-MPC problem (4.3).

Then, we can define the QP-MPC static state-feedback control policy $\pi : \mathbb{R}^{n_p} \to \mathbb{R}^{n_u}$, as in Eq. (2.122), under the one-step receding horizon policy, i.e.,

$$u_k = \hat{u}_{0|k}^* = \pi(p_k). \tag{4.6}$$

The closed-loop system (4.1), (4.6) evolves as

$$x_{k+1} = Ax_k + B\pi([x_k^\top,\ \nu_k^\top]^\top) + B_\nu \nu_k + b, \tag{4.7a}$$
$$y_k = Cx_k + D\pi([x_k^\top,\ \nu_k^\top]^\top) + D_\nu \nu_k + d, \quad k \geq 0. \tag{4.7b}$$

Following Section 2.4 and performing slight modifications, the QP-MPC problem (4.3) can be rewritten in a simpler and more compact form, comprising only $\hat{\boldsymbol{u}}$ as decision variables and p_k as a parameter.

We recall Eq. (2.163) to express the $\hat{\boldsymbol{x}}$ as a function of $\hat{\boldsymbol{u}}$ and p_k by

$$\hat{\boldsymbol{x}} = \begin{bmatrix} \Phi & \Gamma_\nu \end{bmatrix} p_k + \Gamma \hat{\boldsymbol{u}} + \gamma, \tag{4.8}$$

where Φ, Γ, and γ are given in Eq. (2.165), while

$$\Gamma_\nu = \begin{bmatrix} \mathbf{0}_{n_x \times n_x} \\ \sum_{i=1}^{N_p} \left(e_{N_p}^{(i)} \otimes \sum_{j=0}^{i-1} A^j \right) \end{bmatrix} B_\nu. \tag{4.9}$$

By Eq. (4.3d), we can also rewrite

$$\hat{\boldsymbol{y}} = \overline{C}\hat{\boldsymbol{x}} + \overline{D}\hat{\boldsymbol{u}} + \overline{D}_\nu \nu_k + \overline{d}, \tag{4.10}$$

where

$$\overline{C} = \begin{bmatrix} I_{N_p} \otimes C & \mathbf{0}_{n_y N_p \times n_x} \end{bmatrix}, \quad \overline{D} = I_{N_p} \otimes D, \tag{4.11a}$$
$$\overline{D}_\nu = I_{N_p} \otimes D_\nu, \qquad \overline{d} = \mathbf{1}_{N_p} \otimes d. \tag{4.11b}$$

Next, by defining

$$\begin{aligned} &\overline{Q} = I_{N_p} \otimes Q, \quad \overline{R} = I_{N_p} \otimes R, \quad \overline{R}_\Delta = M \otimes R_\Delta, \\ &M = \operatorname{diag}([1,\ 2 \cdot \mathbf{1}_{N_p-2}^\top,\ 1]^\top) - E_{N_p}^{(1)} - E_{N_p}^{(1)\top}, \\ &\boldsymbol{y}_r = \mathbf{1}_{N_p} \otimes y_r, \quad \boldsymbol{u}_r = \mathbf{1}_{N_p} \otimes u_r, \end{aligned} \tag{4.12}$$

we can compactly rewrite the cost function $J(\hat{\boldsymbol{y}}, \hat{\boldsymbol{u}})$ in Eq. (4.3a) as

$$J(\hat{\boldsymbol{y}}, \hat{\boldsymbol{u}}) = \|\hat{\boldsymbol{y}} - \boldsymbol{y}_r\|_{\overline{Q}}^2 + \|\hat{\boldsymbol{u}} - \boldsymbol{u}_r\|_{\overline{R}}^2 + \|\hat{\boldsymbol{u}}\|_{\overline{R}_\Delta}^2. \tag{4.13}$$

Replacing Eq. (4.10) into Eq. (4.13) yields

$$J(\hat{\boldsymbol{y}}, \hat{\boldsymbol{u}}) = J_u(\hat{\boldsymbol{u}}) = \frac{1}{2}\hat{\boldsymbol{u}}^\top H \hat{\boldsymbol{u}} + (F p_k + c)^\top \hat{\boldsymbol{u}}, \tag{4.14}$$

where

$$H = (\overline{C}\Gamma + \overline{D})^\top \overline{Q}(\overline{C}\Gamma + \overline{D}) + \overline{R} + \overline{R}_\Delta, \tag{4.15a}$$
$$F = (\overline{C}\Gamma + \overline{D})^\top \overline{Q}\begin{bmatrix}\overline{C}\Phi & \overline{C}\Gamma_\nu + \overline{D}_\nu\end{bmatrix}, \tag{4.15b}$$
$$c = (\overline{C}\Gamma + \overline{D})^\top \overline{Q}(\overline{C}\gamma + \overline{d} - \boldsymbol{y}_r) - \overline{R}\boldsymbol{u}_r. \tag{4.15c}$$

Finally, we compact the linear constraints (4.3e)-(4.3g) as

$$\overline{H}_x \hat{\boldsymbol{x}} \leq h_x, \quad \overline{H}_u \hat{\boldsymbol{u}} \leq h_u, \quad \overline{H}_y \hat{\boldsymbol{y}} \leq h_y, \tag{4.16}$$

where

$$\overline{H}_x = \begin{bmatrix} I_{N_p} \otimes H_x & \mathbf{0}\end{bmatrix}, \quad \overline{h}_x = \mathbf{1}_{N_p} \otimes h_x, \tag{4.17a}$$
$$\overline{H}_u = I_{N_p} \otimes H_u, \qquad \overline{h}_u = \mathbf{1}_{N_p} \otimes h_u, \tag{4.17b}$$
$$\overline{H}_y = I_{N_p} \otimes H_y, \qquad \overline{h}_y = \mathbf{1}_{N_p} \otimes h_y. \tag{4.17c}$$

Then, replacing Eqs. (4.8) and (4.10) into Eq. (4.16) yields

$$G\hat{\boldsymbol{u}} \leq w + K p_k, \tag{4.18}$$

where

$$G = \begin{bmatrix} \overline{H}_x\Gamma \\ \overline{H}_u \\ \overline{H}_y(\overline{C}\Gamma + \overline{D})\end{bmatrix}, \quad w = \begin{bmatrix} \overline{h}_x - \overline{H}_x\gamma \\ \overline{h}_u \\ \overline{h}_y - \overline{H}_y(\overline{C}\gamma + \overline{d})\end{bmatrix}, \quad K = \begin{bmatrix} -\overline{H}_x\Phi & -\overline{H}_x\Gamma_\nu \\ \mathbf{0} & \mathbf{0} \\ -\overline{H}_y\overline{C}\Phi & -\overline{H}_y(\overline{C}\Gamma_\nu + \overline{D}_\nu)\end{bmatrix}. \tag{4.19}$$

Overall, the original QP-MPC problem (4.3) is equivalent to the following compact quadratic program:

$$\min_{\hat{\boldsymbol{u}}} \quad \frac{1}{2}\hat{\boldsymbol{u}}^\top H \hat{\boldsymbol{u}} + (F p_k + c)^\top \hat{\boldsymbol{u}} \tag{4.20a}$$
$$\text{s.t.} \quad G\hat{\boldsymbol{u}} \leq w + K p_k. \tag{4.20b}$$

As reported in Section 2.4, the QP-MPC problem (4.3), (4.20) is strongly convex and admits a unique and global optimum $\hat{\boldsymbol{u}}^*$ for any feasible parameter p_k [23, 19, 35]. Consequently, the policy π (4.6) is well-defined and establishes a bijection between each feasible parameter $p_k \in \mathcal{P}$ and the optimal control input $u_k = \hat{u}_{0|k}^*$.

Remark 4.1

By Lemma 2.3, for the QP-MPC policy (4.6) it holds that $u_r = \pi([x_r^\top,\ \mathbf{0}_{n_\nu}^\top]^\top)$.

Consequently, by Lemma 2.3 and Eq. (4.4), the couple (x_r, y_r) is an equilibrium of the undisturbed closed-loop system (4.7), i.e.,

$$x_r = Ax_r + B\pi([x_r^\top,\ \mathbf{0}_{n_\nu}^\top]^\top) + b, \tag{4.21a}$$
$$y_r = Cx_r + D\pi([x_r^\top,\ \mathbf{0}_{n_\nu}^\top]^\top) + d. \tag{4.21b}$$

4.3. Analog Circuital Explicit MPC

In this section, we present our general design methodology to implement the QP-MPC control policy π (4.6) as a fully-analog electronic circuit. As said in Section 4.1, we refer to our methodology as *Analog Circuital Explicit MPC* [124].

First, we leverage Explicit MPC (Ex-MPC) to obtain a closed-form expression for the QP-MPC policy. Next, we apply a series of complexity-reduction techniques aimed at reducing the number of regions in the explicit formulation. Finally, we describe the implementation of Ex-MPC as an analog circuit and its component selection.

4.3.1. Explicit MPC Design

Consider the QP-MPC policy π (4.6) and introduce the following assumption on the set of feasible parameters p_k:

Assumption 4.1

The values of interest for the parameters p_k lie in a convex polytope $\mathcal{P} \subseteq \mathbb{R}^{n_p}$.

Under Assumption 4.1, as reported in Section 2.4, the QP-MPC policy $\pi : \mathcal{P} \to \mathcal{U}$ admits a closed-form expression that can be found through Explicit MPC and multi-parametric quadratic programming (mp-QP). Specifically, according to Theorem 2.10, π takes the form of a continuous piecewise-affine (PWA) function, defined over a collection of R polytopic regions $(\mathcal{R}_i)_{i=1}^R$ that constitute a polytopic partition of $\mathcal{P}$, i.e.,

$$u_k = \pi(p_k) = K_i p_k + l_i = \kappa_i(p_k) \quad \text{if } \ p_k \in \mathcal{R}_i, \tag{4.22}$$

where

$$\mathcal{R}_i = \{p \in \mathbb{R}^{n_p} : H_i p \leq h_i\}, \quad i = 1, \ldots, R, \tag{4.23}$$

and

$$\bigcup_{i=1}^{R} \mathcal{R}_i = \mathcal{P}, \quad \text{int}(\mathcal{R}_i) \cap \text{int}(\mathcal{R}_j) = \emptyset, \ \ \forall\, 1 \leq i, j \leq R,\ i \neq j. \tag{4.24}$$

We shall refer to Eq. (4.22) as Explicit MPC policy.

4.3.2. Complexity Reduction of the Explicit MPC Policy

Replacing the QP-MPC problem (4.3) with the Ex-MPC policy (4.22) offers the significant advantage of being able to compute the optimal control input using a static function, rather than solving online a QP problem. Moreover, as we shall see in Section 4.3.3, thanks to the fact that the Ex-MPC policy (4.22) has a PWA structure and the Ex-MPC regions (4.23) are polytopes, we can realize the Ex-MPC as an electronic circuit, employing only static analog components, and, thus, achieving a very fast function evaluation.

Yet, the practical applicability of this approach based on Ex-MPC is limited to problems of modest size. Specifically, as highlighted in Section 2.4.2, the number of polytopic regions R of the Ex-MPC policy π is problem-dependent. A high number of regions may easily make Ex-MPC not feasible for both digital and analog practical realization, since:

- For digital/embedded implementation, a high number of regions may exceed the storage capacity of the hardware [14]. Also, identifying the region $\mathcal{R}_i$ containing the current parameter p_k (to then apply the corresponding control action $u_k = K_i p_k + l_i$) may take more time than solving online the actual QP-MPC problem (4.3) using conventional algorithms for convex optimization [14].
- For analog implementation, a high number of regions would require a large number of components, resulting in an overly bulky and expensive circuit. We present a detailed discussion on this aspect in Section 4.3.3.

As reported in Section 2.4.2, the number of regions R is equal to the number of possible combinations of active constraints for the QP-MPC problem (4.20) [61]. In particular, two upper bounds on R can be found, given by Eqs. (2.193) and (2.194), i.e.,

$$R \leq \sum_{i=0}^{q} \binom{q}{i} = 2^q, \tag{4.25a}$$

$$R \leq \sum_{i=0}^{n_u N_p} \binom{q}{i}, \tag{4.25b}$$

where q is the total number of linear inequality constraints imposed in problem (4.20), i.e., $G \in \mathbb{R}^{q \times n_u N_p}$.

However, in most practical applications, R is much smaller because most of the constraint combinations are never active at optimality. For instance, in presence of upper-lower bound constraints that are non-coincident, at most one of the two inequalities is active, thus halving the maximum number of regions.

In the following, our goal is to further reduce the number of regions and the number of inequalities defining them. To this end, we employ in sequence the following four complexity-reduction strategies:

1) Move blocking strategy;
2) Optimal merging of regions;
3) Hyperplane separation of saturated regions;
4) Removal of trivial inequalities.

In the remainder of this section, we detail these four steps.

Move Blocking Strategy

A widely-employed approach to reduce the complexity of QP-MPC problems like (4.3) is the so-called *move blocking strategy* [30].

Such a strategy consists in linearly mapping the input sequence $\hat{\boldsymbol{u}}$ to a smaller set of decision variables $\boldsymbol{u}_c = (u_{c,j})_{j=0}^{N_c-1}$, $N_c < N_p$. The mapping is defined as follows:

$$\hat{\boldsymbol{u}} = T\boldsymbol{u}_c, \quad T \in \mathbb{R}^{N_p n_u \times N_c n_u}, \tag{4.26}$$

where the matrix T is lower-trapezoidal and can be chosen at will to "shape" the input sequence $\hat{\boldsymbol{u}}$. Two "popular" choices are:

$$T^{(1)} = \begin{bmatrix} I_{N_c-1} & \mathbf{0} \\ \mathbf{0} & \mathbf{1}_{N_p-N_c+1} \end{bmatrix} \otimes I_{n_u}, \tag{4.27a}$$

$$T^{(2)} = \begin{bmatrix} \mathbf{1}_{\ell_0} & & \mathbf{0} \\ & \ddots & \\ \mathbf{0} & & \mathbf{1}_{\ell_{N_c-1}} \end{bmatrix} \otimes I_{n_u}, \quad \ell_j = \begin{cases} \left\lfloor \frac{N_p}{N_c} \right\rfloor & \text{if } j \in \{0, \dots, N_c-2\}, \\ N_p - (N_c-1)\left\lfloor \frac{N_p}{N_c} \right\rfloor & \text{if } j = N_c-1. \end{cases} \tag{4.27b}$$

Both choices in Eq. (4.27) shape $\hat{\boldsymbol{u}}$ as a piecewise constant sequence consisting of N_c segments: with the matrix $T^{(1)}$ (4.27a), the first N_c-1 segments consist of one sample only, while the last one is longer and covers the rest of the horizon; with the matrix $T^{(2)}$ (4.27b), all segment have equal width $\lfloor \frac{N_p}{N_c} \rfloor$ (except, possibly, for the last one, if N_c is not a divisor of N_p).

Eq. (4.26) can be directly replaced into the QP-MPC problem (4.20), obtaining its reduced version with decision variables $\boldsymbol{u}_c$. Specifically, problem (4.20) becomes

$$\min_{\boldsymbol{u}_c} \quad \frac{1}{2}\boldsymbol{u}_c^\top \tilde{H}\boldsymbol{u}_c + (\tilde{F}p_k + \tilde{c})^\top \boldsymbol{u}_c \tag{4.28a}$$

$$\text{s.t.} \quad \tilde{G}\boldsymbol{u}_c \leq w + Kp_k. \tag{4.28b}$$

where

$$\tilde{H} = T^\top H T, \quad \tilde{F} = T^\top F, \quad \tilde{c} = T^\top c, \quad \tilde{G} = GT. \tag{4.29}$$

We notice that all the fundamental properties of the original QP-MPC problem (4.20), such as its strong convexity and Remark 4.1, are preserved in Eq. (4.28).

The move blocking strategy translates into a reduction of the number of regions R in the Ex-MPC policy (4.22). Indeed, the number of decision variables in the QP-MPC problem (4.20) has now reduced from $N_p n_u$ to $N_c n_u$. Thus, recalling the upper bound on R in Eq. (4.25b), we have that

$$R \leq \sum_{i=0}^{n_u N_c} \binom{q}{i} < \sum_{i=0}^{n_u N_p} \binom{q}{i}. \tag{4.30}$$

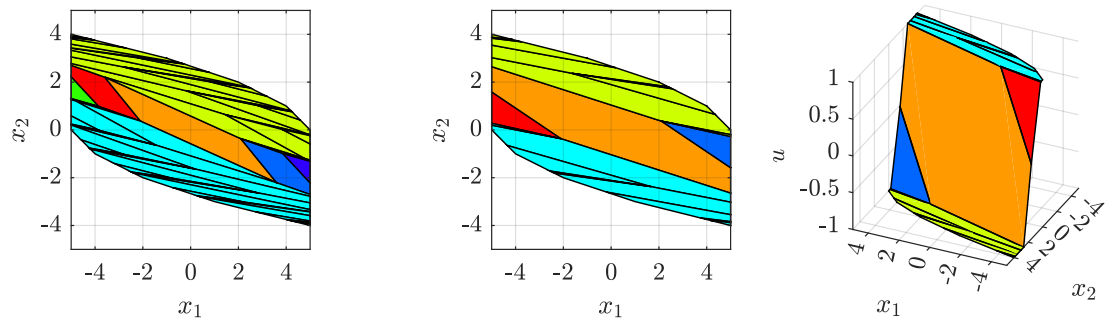
For more details on this aspect, we refer the reader to [4].

Remark 4.2

It is worth noting that, with the modification introduced by move blocking, the optimal solution of the QP-MPC problem (4.20) is not preserved when solving problem (4.28); however, we can always perform a suitable tuning of the parameter N_c to ensure a negligible decrease in the closed-loop performance.

Example 4.1 Let us recall Example 2.2. Considering its Ex-MPC policy, composed by $R = 63$ regions (Figure 4.1a) and with $N_c = 10$, we apply on it the move blocking strategy presented in this section, setting $N_c = 5$ (Figure 4.1b).

We can observe that move blocking reduces the partition to $R = 19$ regions (70% reduction). We also see that, as noted in Remark 4.2, the Ex-MPC policy changes when applying move blocking.



(a) Without move blocking ($R = 63$).

(b) With move blocking ($R = 19$).

Figure 4.1. Ex-MPC policy with the move blocking strategy, based on Example 2.2. Each color relates to a different affine function.

Optimal Merging of Regions

Typically, in Ex-MPC policies like (4.22), several regions share the same affine function; this can be noticed, e.g., in Example 2.2. Thus, a possible way to simplify the policy would consist in combining these regions, so to reduce their number while preserving convexity. This process, by which regions sharing the same affine function are merged into larger convex polytopes, is generally referred to as *region merging*.

A baseline strategy is *greedy merging*. Such an approach iteratively inspects adjacent regions (i.e., regions with a common facet), sharing the same affine function, and merges pairs of regions whenever their union is convex. Convexity of the merged regions is typically verified with linear programs (LP) [15]. The method stops when no more pairs can be merged.

While this strategy bears the hallmarks of simplicity, it only looks at adjacent pairs of regions whose immediate union is convex. Due to this "myopic" merging criterion, greedy merging is suboptimal and often stalls far from the best achievable reduction.

A more effective merging strategy was proposed in [57], exhibiting optimality guarantees and, thus, allowing to obtain the minimal number of regions. Two version of this optimal

merging strategy have been proposed:

- *Disjoint optimal merging.* The first optimal approach seeks the minimum number of disjoint merged regions whose union forms the domain of each affine function.

  The Ex-MPC partition is converted into an hyperplane arrangement, built from the facets of each polytopic region. The regions are cut by the arrangement hyperplanes into smaller polytopes, which constitute the cells of the arrangement. Each cell is encoded by its "marking", a sign pattern describing the cell position relative to every hyperplane in the arrangement. A key structural fact is that the convexity of the union of two cells can be verified purely from their markings: the union is convex if and only if the two markings differ in exactly one component. This observation allows to avoid LPs to check convexity, as done in greedy merging [15].

  The optimal merging is then performed through a branch-and-bound procedure. The algorithm recursively splits the current envelope[1] of cells associated with the same affine function, using hyperplanes from the arrangement. Once a branch terminates, the corresponding envelope is convex and contains only cells with the same law; thus, it is taken as a new merged region. This procedure is proved to be optimal, i.e., it yields the minimal number of disjoint merged regions.

- *Non-disjoint optimal merging.* The second optimal approach relaxes the disjointness requirement and allows merged regions to overlap. The problem is formulated as a Boolean logic minimization task, where each hyperplane side is a Boolean variable, each cell is a minterm, and the union of all cells sharing the same affine function becomes a Boolean function. Minimizing the number of merged regions is equivalent to minimizing the number of product terms in a sum-of-products representation of this function. Such a problem is solved using standard logic minimizers, such as ESPRESSO [24].

  Also this procedure is proved to be optimal, i.e., it yields the minimal number of non-disjoint merged regions.

  Importantly, allowing non-disjoint polytopes leads to solutions with fewer regions and fewer facets (i.e., fewer inequalities defining the region) compared to the case where we restrict ourself to disjoint polytopes [57].

For what concerns computational complexity and execution time, greedy merging is simple and fast, relying on LP-based convexity checks between neighboring regions; however, it provides a merely suboptimal solution. Optimal disjoint merging is rather slow, as its complexity scales exponentially with the the size of the arrangement. Non-disjoint merging, instead, generally scales better and is substantially faster than the disjoint version, thanks to its formulation as a logic minimization problem.

In the subsequent design, we rely on the non-disjoint optimal merging strategy, for its capability to provide the minimal number of regions and facets, while also exhibiting a very fast execution.

The non-disjoint optimal merging algorithm is implemented within the Multi-Parametric Toolbox (MPT) for MATLAB [71].

[1] Given two polytopes $\mathcal{P}_1$ and $\mathcal{P}_2$, their envelope $\text{env}(\mathcal{P}_1, \mathcal{P}_2)$ is defined as the intersection of half spaces that contain both polytopes, where the half spaces are induced by the facets of the polytopes.

Remark 4.3

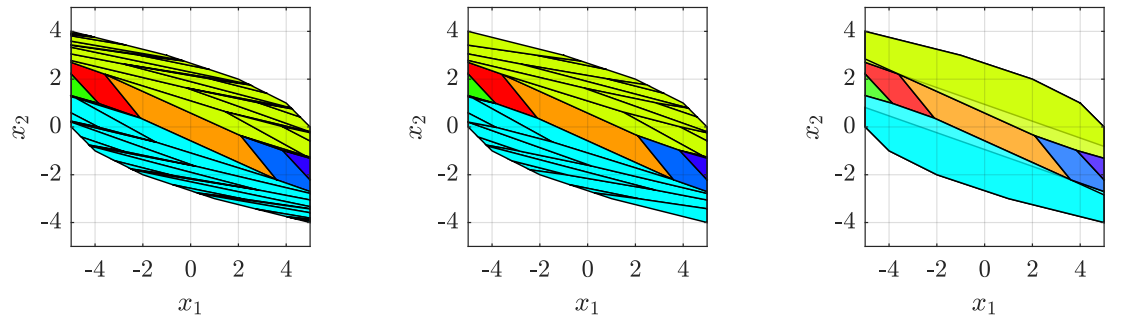
In a non-simplified Ex-MPC policy, by Theorem 2.10, all regions are disjoint, i.e., no overlaps are present between any of them. Instead, when the non-disjoint optimal merging strategy is employed, some regions, sharing the same affine function, may partially overlap.

This overlap, however, poses no issue when evaluating the simplified Ex-MPC policy, since the non-disjoint merging strategy ensures that overlaps happen only for regions sharing the same affine function. As a consequence, if the parameter p_k lies within an overlap between regions, the point location algorithm will select either one of these regions, providing in any case the correct optimal control input, as these regions share the same affine function.

Example 4.2 Let us recall Example 2.2. Considering its Ex-MPC policy, composed by $R = 63$ regions (Figure 4.2a), we apply on it the region merging strategies presented in this section, in particular greeding merging (Figure 4.2b) and optimal non-disjoint merging (Figure 4.2c).

We can observe that greedy merging reduces the partition to $R = 31$ regions (51% reduction), while optimal non-disjoint merging yields $R = 9$ regions (86% reduction).

Applying both the optimal non-disjoint merging and the move blocking strategy as in Example 4.1, we would obtain $R = 5$ regions (92% reduction).



(a) No merging ($R = 63$).

(b) Greedy merging ($R = 31$).

(c) Optimal non-disjoint merging ($R = 9$).

Figure 4.2. Ex-MPC policy with region merging, based on Example 2.2. Each color relates to a different affine function. In **(c)**, region overlaps are reported with a lighter shading.

Hyperplane Separation of Saturated Regions

In many practical applications, QP-MPC problems like (4.3) consider bound constraints on the input, i.e., $\mathcal{U} = \{x \in \mathbb{R}^{n_u} : u_{\text{lb}} \leq u \leq u_{\text{ub}}\}$ in Eq. (4.2b), or, equivalently, $H_u = [I_{n_u},\ -I_{n_u}]^\top$ and $h_u = [u_{\text{ub}}^\top,\ u_{\text{lb}}^\top]^\top$ in Eq. (4.3f). In this case, the Ex-MPC policy π (4.22) naturally leads to several regions where the control law is constantly equal to either u_{lb} or u_{ub}. Henceforth, such regions are referred to as saturated regions; the other regions, in contrast, are called unsaturated regions.

Let $\mathcal{I}_{\text{lb}}$ and $\mathcal{I}_{\text{ub}}$ denote the sets of indices of the regions saturated at u_{lb} and u_{ub},

respectively. Then, we define

$$\underline{\mathcal{S}} = \bigcup_{i \in \mathcal{I}_{\text{lb}}} \mathcal{R}_i, \quad \overline{\mathcal{S}} = \bigcup_{i \in \mathcal{I}_{\text{ub}}} \mathcal{R}_i. \tag{4.31}$$

Also, $\mathcal{I}_{\text{unsat}} = \{1, \dots, R\} \setminus (\mathcal{I}_{\text{lb}} \cup \mathcal{I}_{\text{ub}})$ is the set of indices of the unsaturated regions. Moreover, we define the union of unsaturated regions $\mathcal{R}_{\text{unsat}}$ as

$$\mathcal{R}_{\text{unsat}} = \bigcup_{i \in \mathcal{I}_{\text{unsat}}} \mathcal{R}_i. \tag{4.32}$$

By Theorem 2.10, the Ex-MPC policy is continuous, therefore it holds that $\underline{\mathcal{S}} \cap \overline{\mathcal{S}} = \emptyset$. Moreover, the polytopic regions $\mathcal{R}_i$ do not overlap, meaning that

$$\mathcal{P} = \underline{\mathcal{S}} \cup \mathcal{R}_{\text{unsat}} \cup \overline{\mathcal{S}}, \quad \text{int}(\mathcal{R}_{\text{unsat}}) \cap \text{int}(\underline{\mathcal{S}}) = \emptyset, \quad \text{int}(\mathcal{R}_{\text{unsat}}) \cap \text{int}(\overline{\mathcal{S}}) = \emptyset. \tag{4.33}$$

It is worth noticing that the sets $\underline{\mathcal{S}}$ and $\overline{\mathcal{S}}$ are, in general, non-convex and, possibly, non-connected.

The Ex-MPC policy π (4.22) can be then rewritten as follows:

$$u_k = \pi(p_k) = \begin{cases} K_i p_k + l_i & \text{if } \; p_k \in \mathcal{R}_i, \; i \in \mathcal{I}_{\text{unsat}}, \\ u_{\text{lb}} & \text{if } \; p_k \in \underline{\mathcal{S}}, \\ u_{\text{ub}} & \text{if } \; p_k \in \overline{\mathcal{S}}. \end{cases} \tag{4.34}$$

To reduce the complexity of the Ex-MPC policy π in Eq. (4.34), we can think of removing the saturated regions and replace them by means of a function $\sigma : \mathcal{P} \to \mathbb{R}$ which satisfies

$$\sigma(p) < 0, \;\; \forall\, p \in \underline{\mathcal{S}}, \quad \sigma(p) > 0, \;\; \forall\, p \in \overline{\mathcal{S}}. \tag{4.35}$$

The function σ is called *separation function*, as it actually separates the two sets $\underline{\mathcal{S}}$ and $\overline{\mathcal{S}}$, according to its sign.

Therefore, we can equivalently rewrite Eq. (4.34) as

$$u_k = \pi(p_k) = \begin{cases} K_i p_k + l_i & \text{if } \; p_k \in \mathcal{R}_i, \; i \in \mathcal{I}_{\text{unsat}}, \\ u_{\text{lb}} & \text{if } \; p_k \notin \mathcal{R}_{\text{unsat}}, \;\; \sigma(p_k) < 0, \\ u_{\text{ub}} & \text{if } \; p_k \notin \mathcal{R}_{\text{unsat}}, \;\; \sigma(p_k) > 0. \end{cases} \tag{4.36}$$

The Ex-MPC policy formulated as in Eq. (4.36) involves a reduced number of regions, thanks to the removal of saturated regions. This comes at the price of introducing the separation function σ.

To make σ require less memory/analog components to be implemented with respect to the removed saturated regions, we must ensure that σ is sufficiently "simple". Thus, in the following, we seek an affine separation function (i.e., a hyperplane), in the form

$$\sigma(p) = a^\top p + b, \quad a \in \mathbb{R}^p, \;\; b \in \mathbb{R}. \tag{4.37}$$

To this end, we formulate the following two theorems, establishing existence conditions and how to compute such a function.

Theorem 4.1 (Existence of the affine separation function)

Let $\mathcal{S}_1$ and $\mathcal{S}_2$ be the unions of two sets of polytopes, such that $\mathcal{S}_1 \cap \mathcal{S}_2 = \emptyset$. Let V_1 and V_2 be the sets containing the vertices of the polytopes forming $\mathcal{S}_1$ and $\mathcal{S}_2$, respectively. Finally, let $\sigma(x) = a^\top x + b$ be an affine function.

Then, σ separates $\mathcal{S}_1$ and $\mathcal{S}_2$, i.e.,

$$\sigma(x) < 0, \ \forall\, x \in \mathcal{S}_1, \quad \sigma(x) > 0, \ \forall\, x \in \mathcal{S}_2, \tag{4.38}$$

if and only if

$$\sigma(v) < 0, \ \forall\, v \in V_1, \quad \sigma(v) > 0, \ \forall\, v \in V_2. \tag{4.39}$$

Proof

The "$\Rightarrow$" part is immediate, since $V_1 \subset \mathcal{S}_1$ and $V_2 \subset \mathcal{S}_2$.

For the "$\Leftarrow$" part, let us drop the subscripts 1 and 2, to consider both cases in Eqs. (4.38) and (4.39) at the same time. Consider $\lambda \in \mathbb{R}^{|V|}$ such that $\lambda \geq 0$ and $\sum_{i=1}^{|V|} \lambda_i = 1$, and associate each λ_i to a vertex $v_i \in V$, $i = 1, \ldots, |V|$. Then, by Eq. (4.39), it holds that

$$\begin{aligned} &\lambda_i (a^\top v_i + b) \gtreqless 0, \quad \forall\, i \in \{1, \ldots, |V|\} \\ &a^\top \sum_{i=1}^{|V|} \lambda_i v_i + b \sum_{i=1}^{|V|} \lambda_i \gtrless 0 \\ &a^\top x + b \gtrless 0, \end{aligned} \tag{4.40}$$

where $x = \sum_{i=1}^{|V|} \lambda_i v_i \in \operatorname{conv}(V)$, by definition of convex hull. Since $\mathcal{S} \subseteq \operatorname{conv}(V)$, it holds that $x \in \mathcal{S}$, yielding Eq. (4.38).

Theorem 4.2 (Computation of the affine separation function)

Given the results of Theorem 4.1, there exists an affine function $\sigma(x) = a^\top x + b$ that separates $\mathcal{S}_1$ and $\mathcal{S}_2$ if and only if the following LP is feasible:

$$\begin{aligned} \max_{a,b,\varepsilon} \quad & \varepsilon, \quad \alpha = [a^\top, b]^\top, \\ \text{s.t.} \quad & \varepsilon \geq 0, && \text{(4.41a)} \\ & a^\top v^{(1)} + b \leq -\varepsilon, \quad \forall\, v^{(1)} \in V_1, && \text{(4.41b)} \\ & a^\top v^{(2)} + b \geq \ \ \varepsilon, \quad \forall\, v^{(2)} \in V_2, && \text{(4.41c)} \end{aligned}$$

and $\varepsilon^* > 0$. Then, the affine separator is given by $\sigma(x) = a^{*\top} x + b^*$, where $(\alpha^*, \varepsilon^*) = ([a^{*\top}, b^*]^\top, \varepsilon^*)$ is the unique and global optimum of problem (4.41).

Proof

For the "$\Rightarrow$" part, by Theorem 4.1, if $\sigma(x)$ separates $\mathcal{S}_1$ and $\mathcal{S}_2$, then Eq. (4.39) holds.

This means that there exists $\varepsilon > 0$ such that

$$\sigma(v) \leq -\varepsilon < 0, \ \ \forall\, v \in V_1, \quad \sigma(v) \geq \varepsilon > 0, \ \ \forall\, v \in V_2. \tag{4.42}$$

Eq. (4.42) matches constraints (4.41a)-(4.41c); thus, a feasible solution to problem (4.41) exists.

For the "$\Leftarrow$" part, if a feasible solution $(\alpha^*, \varepsilon^*)$ to problem (4.41) exists, then constraints (4.41b) and (4.41c) are satisfied for α^*. Moreover, if $\varepsilon^* > 0$, Eq. (4.39) also holds. Consequently, by Theorem 4.1, Eq. (4.38) is satisfied as well. Thus, there exists $\sigma(x) = a^{*\top} x + b^*$ separating $\mathcal{S}_1$ and $\mathcal{S}_2$.

By convexity, the feasible solution $(\alpha^*, \varepsilon^*)$ is also the unique and global optimum of problem (4.41) [23].

Remark 4.4

The LP feasibility problem (4.41) provides the maximal separation margin $\varepsilon^* > 0$ between the two regions $\mathcal{S}_1$ and $\mathcal{S}_2$, and the separation hyperplane σ, thereby enhancing the robustness to manufacturing tolerances of the circuit components used in the implementation of σ.

Remark 4.5

If the LP problem (4.41) does not admit a feasible solution, then there exists no affine function $\sigma(x)$ that can separate $\mathcal{S}_1$ and $\mathcal{S}_2$. In such cases, it is necessary to resort to more complex separating functions, e.g., polynomial ones [85].

Nevertheless, in the numerical results presented in Section 4.5, which refer to the application considered in this paper, as introduced in Section 4.4, we could always find an affine separator for the Ex-MPC policy (4.36).

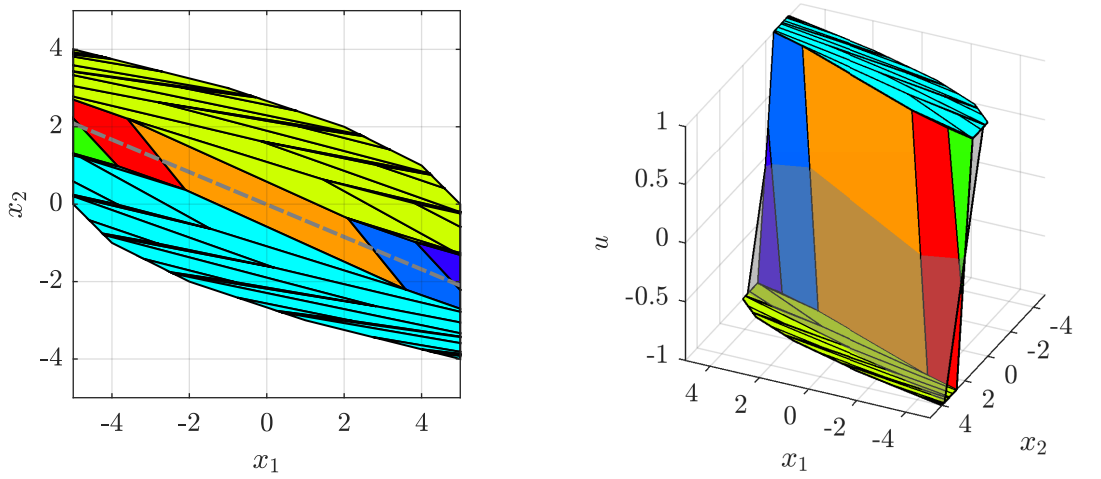
Remark 4.6

Hyperplane separation of saturated regions is most effective when a significant proportion of the regions are saturated. As observed in [61], the number of saturated regions is mainly affected by the tightness of input constraints $\mathcal{U}$ and the magnitude of the input weighting matrix R. Tighter constraints and/or a smaller $\|R\|$ result in more regions becoming saturated, thereby enhancing the method effectiveness.

Example 4.3 Let us recall Example 2.2. Considering its Ex-MPC policy, composed by $R = 63$ regions (Figure 4.3a), we apply on it hyperplane separation of saturated regions, as described in this section. To compute the affine separation function $\sigma(x) = a^\top x + b$, we leverage Theorem 4.2. Specifically, by solving the LP feasibility problem (4.41), we obtain the feasible solution $(a^*, b^*) = ([-2.4875 \times 10^{-1}, \ -5.9384 \times 10^{-1}]^\top, \ -4.9952 \times 10^{-3})$ and $\varepsilon^* = 6.5115 \times 10^{-2} > 0$. Therefore, the affine separator exists and is computed as $\sigma(x) = a^{*\top} x + b^*$ (Figure 4.3b).

To better visualize the regions separation, we compute the zero-level set of σ, i.e., $\{x \in \mathbb{R}^2 : \sigma(x) = 0\} = \{x \in \mathbb{R}^2 : x_2 = -\frac{a_1^*}{a_2^*} x_1 - \frac{b^*}{a_2^*}\}$, and report it in Figure 4.3a.

We can observe that, with hyperplane separation of saturated regions, we are left with only 5 unsaturated regions in the partition (92% reduction).



(a) Ex-MPC partition and zero-level set of the affine separation function σ (- - -).

(b) Ex-MPC policy and affine separation function σ (▭).

Figure 4.3. Ex-MPC policy with hyperplane separation of saturated regions, based on Example 2.2. Each color relates to a different affine function: regions saturated to u_{lb} ($\underline{\mathcal{S}}$, ▭), regions saturated to u_{ub} ($\overline{\mathcal{S}}$, ▭).

Removal of Trivial Inequalities

The Ex-MPC policy (4.22), by Assumption 4.1, is defined over the polytopic domain $\mathcal{P}$, which is the feasible set of parameters p_k (4.5) of the QP-MPC problem (4.3).

The set $\mathcal{P}$ is typically specified by the user, based on prior knowledge of the admissible values that p_k can take. Consequently, any parameter p_k considered in practice always belongs to $\mathcal{P}$. As a result, in the Ex-MPC partition (4.23), the inequalities associated with the facets of $\mathcal{P}$ are trivial (i.e., they are always satisfied) and can therefore be removed.

Overall, after having employed the four complexity-reduction strategies, a comprehensive description of the Ex-MPC policy is given as follows.

First, let N_κ be the number of *unique* unsaturated affine functions in the Ex-MPC policy (4.22) and collect them as $(\kappa_i(p))_{i=1}^{N_\kappa}$. The remaining $R - N_\kappa$ affine functions are either copies of the first N_κ ones or are saturated.

Next, let us consider the unsaturated Ex-MPC regions (4.23), collect them as $(\mathcal{R}_j)_{j=1}^{N_{\text{unsat}}}$, and define $\mathcal{I}_{\text{unsat}} = \{1, \dots, N_{\text{unsat}}\}$. Let $\mathcal{I}_{\kappa_i} \subseteq \mathcal{I}_{\text{unsat}}$ be the set of indeces associated with the regions sharing the same affine function κ_i, $i = 1, \dots, N_\kappa$. It clearly holds that $\bigcup_{i=1}^{N_\kappa} \mathcal{I}_{\kappa_i} = \mathcal{I}_{\text{unsat}}$. Then, each κ_i is defined over the domain $\bigcup_{j \in \mathcal{I}_{\kappa_i}} \mathcal{R}_j$. The remaining $R - N_{\text{unsat}}$ regions are saturated regions.

The Ex-MPC policy (4.22) has the following equivalent expression:

$$u_k = \pi(p_k) = \begin{cases} K_i p_k + l_i & \text{if } p_k \in \bigcup\limits_{j \in \mathcal{I}_{\kappa_i}} \mathcal{R}_j, \quad i = 1, \dots, N_\kappa, \\ u_{\text{ub}} & \text{if } p_k \notin \mathcal{R}_{\text{unsat}}, \quad \sigma(p_k) > 0, \\ u_{\text{lb}} & \text{if } p_k \notin \mathcal{R}_{\text{unsat}}, \quad \sigma(p_k) < 0. \end{cases} \tag{4.43}$$

Eq. (4.43) can be equivalently rewritten by introducing some auxiliary Boolean (logic) variables, as follows:

$$u_k = \pi(p_k) = \begin{cases} K_i p_k + l_i & \text{if } \bigoplus\limits_{j \in \mathcal{I}_{\kappa_i}} r_j, \quad i = 1, \dots, N_\kappa, \\ u_{\text{ub}} & \text{if } s_a \odot s, \\ u_{\text{lb}} & \text{if } s_a \odot \overline{s}, \end{cases} \tag{4.44}$$

where $(r_j)_{j=1}^{N_{\text{unsat}}}$, s, $s_a \in \{0, 1\}$ are Boolean variables, defined by

$$r_j = \begin{cases} 1 & \text{if } H_j p_k \leq h_j, \\ 0 & \text{otherwise,} \end{cases} \quad s = \begin{cases} 1 & \text{if } {a^*}^\top p_k + b^* > 0, \\ 0 & \text{otherwise,} \end{cases} \quad s_a = \bigodot_{j=1}^{N_{\text{unsat}}} \overline{r}_j, \tag{4.45}$$

where we recall that $\odot$, $\oplus$, and $\overline{\star}$ are the AND, OR, and NOT operations on Boolean variables, respectively.

Specifically, in Eq. (4.45), r_j is 1 when p_k belongs to the j-th unsaturated region $\mathcal{R}_j$, $j = 1, \dots, N_{\text{unsat}}$, and 0 otherwise; s is 1 when the affine separator $\sigma(p_k) > 0$ (p_k belongs to a region saturated to u_{ub}, i.e., $\overline{\mathcal{S}}$), and 0 if $\sigma(p_k) < 0$ (p_k belongs to a region saturated to u_{lb}, i.e., $\underline{\mathcal{S}}$); s_a is 0 if there is at least one active unsaturated region, and 1 otherwise.

The Ex-MPC policy formulated as in Eq. (4.44) accounts for the problem reported in Remark 4.3, i.e., by employing the non-disjoint optimal merging strategy, some unsaturated regions, sharing the same affine function, may partially overlap. Indeed, considering any subset of indices $\mathcal{J} \subseteq \mathcal{I}_{\kappa_i}$, for those $p \in \bigcap_{j \in \mathcal{J}} \mathcal{R}_j \neq \emptyset$, it happens that all variables r_j associated with these regions are 1 at the same time, i.e., $r_j = 1$ for all $j \in J$. The logic relation

$$u_k = \pi(p_k) = K_i p_k + l_i \quad \text{if } \bigoplus_{j \in \mathcal{I}_{\kappa_i}} r_j, \quad i = 1, \dots, N_\kappa \tag{4.46}$$

in Eq. (4.44) makes all variables r_j pass through an OR operator, whose output is 1 if p_k belongs to $\bigcup_{j \in \mathcal{I}_{\kappa_i}} \mathcal{R}_j$, i.e., the whole domain of κ_i.

4.3.3. Circuital Implementation

In the following, we proceed with the actual circuital implementation of Ex-MPC. Starting from the complexity-reduced Ex-MPC policy (4.44), we describe our general approach to implement it as an electronic circuit, using only commercially-available low-latency analog components. Specifically, the analog Ex-MPC circuit shall be implemented using:

- one multiplexer (MUX);
- a set of generalized adders;

- a set of comparators;
- a small logic gate network.

Figure 4.4 reports a top-view representation of the circuit that we will implement. A sample and hold (S/H) circuit is included after the MUX to ensure proper sampling and avoid inter-sample oscillations of the control input.

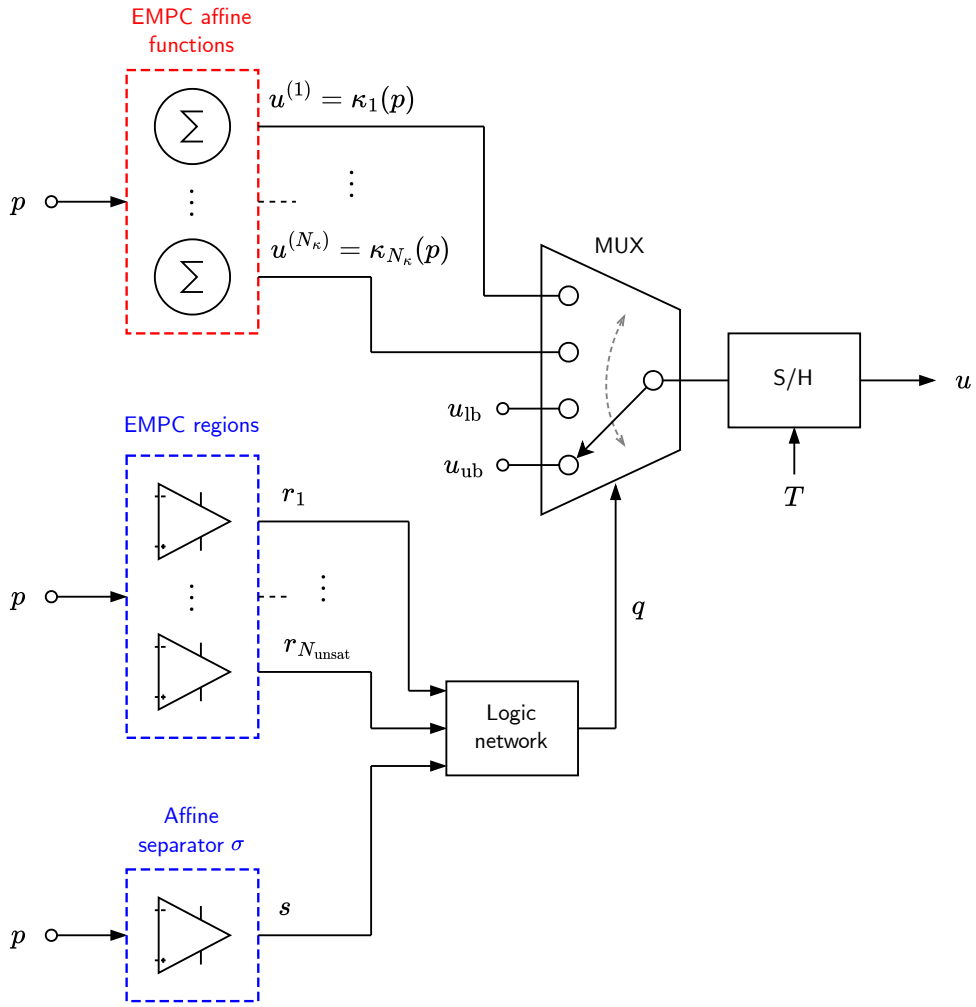


Figure 4.4. Ex-MPC analog circuit: high-level schematic.

Below, we detail every stage of the design for the four main building blocks of the Ex-MPC circuit.

Multiplexer

It is the main component in our Ex-MPC circuit design. We use an analog multiplexer to implement the by-case PWA control policy in Eq. (4.44). Specifically, we use a multiplexer with $N_\kappa + 2$ inputs: one for each unique unsaturated affine function, plus the two saturated input values u_{ub} and u_{lb}. While u_{ub} and u_{lb} are trivially imposed through a constant voltage, the inputs corresponding to the affine functions need to be computed online, given the current measured value of states and disturbances. To this end, we introduce generalized adders.

Generalized Adders

Generalized adders are a standard circuit in analog electronics [142] and their design is easily automated, as detailed in the following.

We employ generalized adders to generate the control input $u^{(i)}$ given by each i-th unique unsaturated affine function $\kappa_i(p)$ in the Ex-MPC policy (4.44), i.e.,

$$u^{(i)} = K_i p + l_i, \quad i = 1, \dots, N_\kappa, \tag{4.47}$$

where, for notational convenience, we omit the explicit time dependence on k. Each control input $u^{(i)}$ is then, in practice, a voltage value.

Without loss of generality, we consider the case $n_u = 1$ (i.e., $K_i \in \mathbb{R}^{1\times n_p}$ and $l_i \in \mathbb{R}$). In this setting, being $u^{(i)}$ a scalar, each affine function requires a single generalized adder, so N_κ in total. If $n_u > 1$, then a generalized adder is needed for each component $(u^{(i)})_j$, $j = 1, \dots, n_u$, resulting in n_u generalized adders for each affine function, so $N_\kappa n_u$ in total.

For each generalized adder, the circuit inputs are the voltage measures of the Ex-MPC parameters p (4.5) and a positive constant voltage V_0, which can be arbitrarily chosen and is used to implement the constant offset l_i; the circuit output is the voltage $u^{(i)}$.

Let us append V_0 to p, i.e., $p_{n_p+1} = V_0$. Also, let $K_{i,j} = (K_i)_j$ and append l_i to K_i, i.e., $K_{i,n_p+1} = l_i$. Then, by Eq. (4.47), we can equivalently rewrite $u^{(i)}$ as follows:

$$u^{(i)} = \sum_{j=1}^{n_p+1} K_{i,j}\, p_i \tag{4.48a}$$

$$= \sum_{k=1}^{N_i} g_{i,k}^{(+)}\, p_{i,k}^{(+)} - \sum_{k=1}^{M_i} g_{i,k}^{(-)}\, p_{i,k}^{(-)}, \tag{4.48b}$$

where $(g_{i,k}^{(+)})_{k=1}^{N_i}$ collects the positive elements of K_i, i.e., $K_{i,j} \in (g_{i,k}^{(+)})_{k=1}^{N_i}$ if $K_{i,j} \geq 0$; $(g_{i,k}^{(-)})_{k=1}^{M_i}$ collects the negative elements (with changed sign) of K_i, i.e., $-K_{i,j} \in (g_{i,k}^{(-)})_{k=1}^{M_i}$ if $K_{i,j} < 0$; $(p_{i,k}^{(+)})_{k=1}^{N_i}$ collects the inputs p associated with positive gains; $(p_{i,k}^{(-)})_{k=1}^{M_i}$ collects the inputs p associated with negative gains; N_i and M_i are the number of inputs with positive and negative gains, respectively ($N_i + M_i = n_p + 1$). In the following, for notational convenience, we omit the adder index i.

Let us define the total positive and negative gains, i.e.,

$$K^{(+)} = \sum_{k=1}^{N} g_k^{(+)}, \quad K^{(-)} = \sum_{k=1}^{M} g_k^{(-)}. \tag{4.49}$$

We identify two topologically distinct circuits for the considered design, depending on whether $K^{(+)} \geq K^{(-)} + 1$ or $K^{(+)} < K^{(-)} + 1$. Both topologies are reported in Figure 4.5.

In the first case, with $K^{(+)} \geq K^{(-)} + 1$, the circuit includes the additional resistance $R_{N+1}^{(-)}$, connected to the positive input of the OP-AMP $\mathrm{OP}^{\mathrm{add}}$, as shown in Figure 4.5 (green resistor). This resistor serves to increase the overall gain of the negative side. In this case, if we ensure that

$$G_f + \sum_{k=1}^{N+1} G_k^{(-)} = \sum_{k=1}^{M} G_j^{(+)}, \tag{4.50}$$

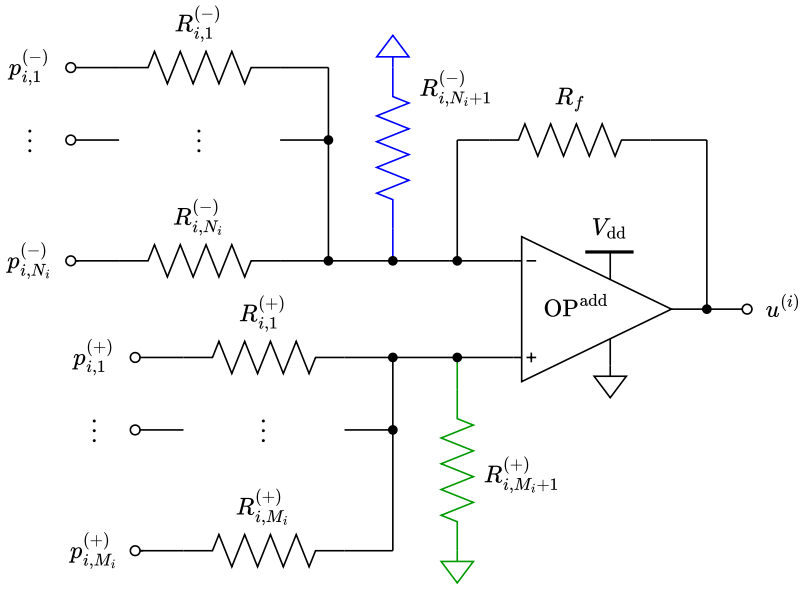


Figure 4.5. Generalized adder for the i-th affine function. The blue resistor is used if $K_i^{(+)} < K_i^{(-)} + 1$, while the green one is used when $K_i^{(+)} \geq K_i^{(-)} + 1$.

where $G_f = 1/R_f$, $G_k^{(-)} = 1/R_k^{(-)}$, and $G_k^{(+)} = 1/R_k^{(+)}$, we obtain the following:

$$g_k^{(-)} = \frac{G_k^{(-)}}{G_f}, \quad g_k^{(+)} = \frac{G_k^{(+)}}{G_f}. \tag{4.51}$$

Therefore, fixed a value for R_f, we can trivially obtain $R_k^{(-)}$ and $R_k^{(+)}$ from Eq. (4.51) and, next, $R_{N+1}^{(-)}$ is obtained using Eq. (4.50).

Similarly, in the second case, with $K^{(+)} < K^{(-)} + 1$, we increase the overall gain of the positive side by including the additional resistance $R_{M+1}^{(+)}$ to the positive input of OP^{add}, as shown in Figure 4.5 (blue resistor). In this case, if

$$G_f + \sum_{k=1}^{N} G_k^{(-)} = \sum_{k=1}^{M+1} G_k^{(+)}, \tag{4.52}$$

we obtain, as for the previous case, the design equations in Eq. (4.51), through which we can compute $R_k^{(-)}$ and $R_k^{(+)}$. Finally, $R_{M+1}^{(+)}$ is computed using Eq. (4.52).

Adders constitute the most expensive part of the device, as they require OP-AMPs for the implementation. Still, the overall number of required adders coincides with the number of unique unsaturated affine functions of the Ex-MPC policy (4.44), which can be reduced through the techniques described in Section 4.3.2, particularly with the move blocking strategy (see Example 4.1).

Comparators

We employ a set of comparators to obtain the Boolean signals $(r_j)_{j=1}^{N_{\text{unsat}}}$ and s, as defined in Eq. (4.45). For each r_j, let us define the auxiliary Boolean signal

$$r_{j,k} = \begin{cases} 1 & \text{if } \; -(H_j)_{k,\cdot}\, p + (h_j)_k \geq 0, \\ 0 & \text{otherwise,} \end{cases} \tag{4.53}$$

where $(H_j)_{k,\cdot}$ and $(h_j)_k$ are the k-th row of matrix H_j and the k-th element of vector h_j, respectively. Then, being N_j the number of inequality constraints defining the j-th region, we have that

$$r_j = \bigodot_{k=1}^{N_j} r_{j,k}, \quad j = 1, \ldots, N_{\text{unsat}}, \tag{4.54}$$

which can be easily realized through logic AND gates.

Obtaining each of the Boolean signals $(r_{j,k})_{(j,k)\in\{1,\ldots,N_{\text{unsat}}\}\times\{1,\ldots,N_j\}}$ and s requires evaluating a set of inequalities of the kind

$$[\alpha_1,\ \ldots,\ \alpha_{n_p}]\, p + \alpha_{n_p+1} V_0 \geq 0, \tag{4.55}$$

where V_0 is a positive constant voltage, which can be arbitrarily chosen (it is also employed in the generalized adders), and $(\alpha_i)_{i=1}^{n_p+1}$ are gains depending on H_i, h_i, a^*, and b^*, whose sign is not known a-priori. Let us append V_0 to p, i.e., $p_{n_p+1} = V_0$.

We realize the comparison altogether with the sums in Eq. (4.55) through the circuit topology in Figure 4.6. The selected circuit compares the voltage on the "+" terminal with that on the "−" terminal; these voltages are obtained through voltage dividers.

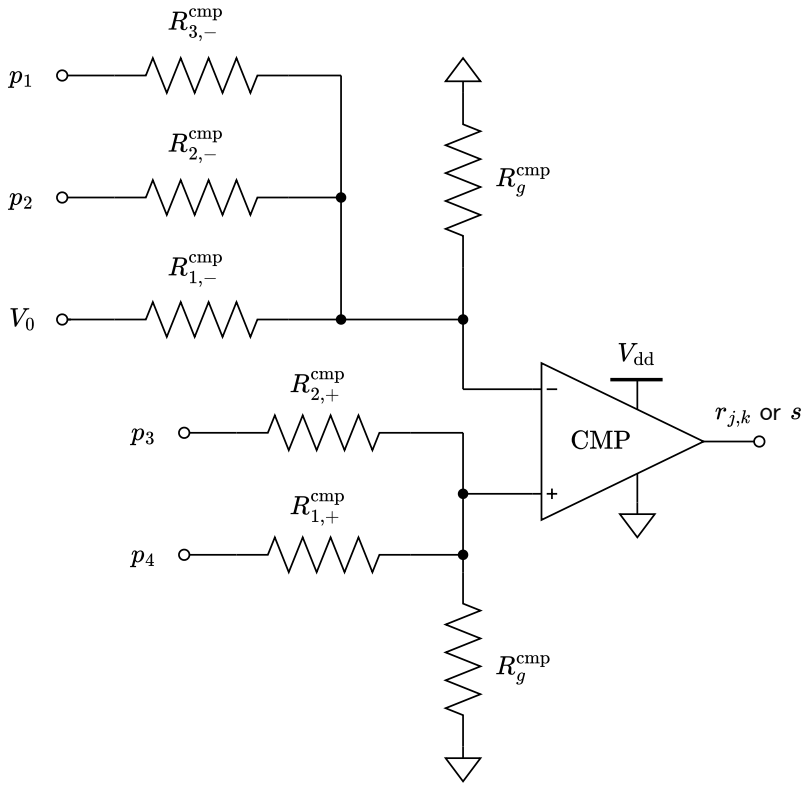


Figure 4.6. Comparator (case with $n_p = 4$, 2 positive gains, and 3 negative gains).

To this end, we first rewrite Eq. (4.55) as

$$\sum_{i\in\mathcal{I}_p} \gamma_i\, p_i \geq \sum_{i\in\mathcal{I}_m} \gamma_i\, p_i, \tag{4.56}$$

where $\mathcal{I}_p, \mathcal{I}_m \subseteq \{1, \dots, n_p + 1\}$ are the set of indices for which $\alpha_i > 0$ and $\alpha_i \leq 0$ in Eq. (4.55), respectively, and

$$\gamma_i = \frac{c\,|\alpha_i|}{\max_{i \in \{1,\dots,n_p+1\}} |\alpha_i|}, \quad c \in (0, 1) \tag{4.57}$$

are normalized and rescaled versions of the gains α_i.

In this way, the gains γ_i associated with each p_i are such that $0 < \gamma_i < 1$, for all $i = 1, \dots, n_p + 1$. This enables the use of inexpensive voltage dividers for comparisons, eliminating the need for additional OP-AMPs.

Consider the networks of $|\mathcal{I}_p| + 1$ and $|\mathcal{I}_m| + 1$ resistors on the "+" and "−" terminal of the comparator, respectively. Let R_g^{cmp} be the resistor connected to ground, R_i^{cmp} the resistor associated with the input p_i, $G_g^{\text{cmp}} = 1/R_g^{\text{cmp}}$ and $G_i^{\text{cmp}} = 1/R_i^{\text{cmp}}$ the corresponding conductances. Then, the voltages on the two terminals are

$$v_+ = \sum_{i \in \mathcal{I}_p} \frac{(G_g^{\text{cmp}} + \sum_{j \neq i} G_j^{\text{cmp}})^{-1}}{R_i^{\text{cmp}} + (G_g^{\text{cmp}} + \sum_{j \neq i} G_j^{\text{cmp}})^{-1}} p_i \tag{4.58}$$

and similarly for v_-, with the sum over $i \in \mathcal{I}_m$.

Imposing the gains match, i.e.,

$$\gamma_i = \sum_{i \in \mathcal{I}_p} \frac{(G_g^{\text{cmp}} + \sum_{j \neq i} G_j^{\text{cmp}})^{-1}}{R_i^{\text{cmp}} + (G_g^{\text{cmp}} + \sum_{j \neq i} G_j^{\text{cmp}})^{-1}}, \tag{4.59}$$

is equivalent to

$$\gamma_i \Big(G_g^{\text{cmp}} + \sum_{i \in \mathcal{I}_p} G_i^{\text{cmp}} \Big) = G_i^{\text{cmp}}, \quad i \in \mathcal{I}_p. \tag{4.60}$$

For any fixed resistor R_g^{cmp}, Eq. (4.60) is a linear system of $|\mathcal{I}_p|$ equations in $|\mathcal{I}_p|$ unknowns, which we recast in matrix form as:

$$\begin{bmatrix} \gamma_1 - 1 & \gamma_1 & \dots & \gamma_1 \\ \gamma_2 & \gamma_2 - 1 & \dots & \gamma_2 \\ \vdots & \vdots & \ddots & \vdots \\ \gamma_{|\mathcal{I}_p|} & \gamma_{|\mathcal{I}_p|} & \dots & \gamma_{|\mathcal{I}_p|} - 1 \end{bmatrix} \begin{bmatrix} G_1^{\text{cmp}} \\ G_2^{\text{cmp}} \\ \vdots \\ G_{|\mathcal{I}_p|}^{\text{cmp}} \end{bmatrix} = -G_g^{\text{cmp}} \begin{bmatrix} \gamma_1 \\ \gamma_2 \\ \vdots \\ \gamma_{|\mathcal{I}_p|} \end{bmatrix}. \tag{4.61}$$

We notice that the system matrix in Eq. (4.61) is always full-rank, thus admits a unique solution for the conductances G_i^{cmp}. Then, resistors R_i^{cmp} are easily obtained by taking the inverse. The very same procedure applies to the network at the "−" terminal.

Overall, the number of comparators is upper-bounded by $1 + \sum_{j=1}^{N_{\text{unsat}}} N_j$, which is often effectively optimized through the complexity-reduction techniques described in Section 4.3.2, particularly the removal of trivial inequalities and the non-disjoint optimal merging strategy (which reduces both the number of Ex-MPC regions and the number of facets, i.e., the linear inequalities defining the regions).

This number can be further reduced if different regions share the same inequality or use opposite ones (i.e., two or more regions have facets on the same hyperplane); in such scenarios, we may reuse comparators from other regions, eventually introducing a NOT gate.

Logic Network

A network of logic gates is needed to drive the selection signal q of the MUX, using the Boolean signals $(r_j)_{j=1}^{N_{\text{unsat}}}$ and s.

Multiplexers admit two types of encoding for the selection signal q: one-hot encoding and binary encoding.

In the case of one-hot encoding, the selection signal $q = (q_i)_{i=1}^{M}$ is composed by $M = N_\kappa + 2$ signals: each i-th signal q_i selects the unsaturated affine function κ_i, $i = 1, \dots, N_\kappa$, while $q_{N_\kappa+1}$ and $q_{N_\kappa+2}$ select the saturated input values u_{lb} and u_{ub}, respectively. According to Eq. (4.44), these selection signals can be obtained through the following logic relations:

$$q_i = \bigoplus_{j \in \mathcal{I}_{\kappa_i}} r_j, \quad i = 1, \dots, N_\kappa, \tag{4.62a}$$

$$q_{N_\kappa+1} = \Big(\bigodot_{j=1}^{N_{\text{unsat}}} \overline{r}_j \Big) \odot \overline{s}, \tag{4.62b}$$

$$q_{N_\kappa+2} = \Big(\bigodot_{j=1}^{N_{\text{unsat}}} \overline{r}_j \Big) \odot s. \tag{4.62c}$$

In the case of binary encoding, the selection signal $q = (q_i)_{i=0}^{M-1}$ is composed by $M = \lceil \log_2(N_\kappa + 2) \rceil$ signals, which select each unsaturated affine function and the two saturated values u_{lb} and u_{ub} by associating a binary number, between 0 and $N_\kappa + 1$, to each of them.

For both encodings, the logic network implementing q can be easily designed using standard logic function optimization methods, such as Karnaugh maps.

4.4. Application to DC-DC Buck Converters

In this section, we apply our Analog Circuital Ex-MPC approach to the case study of DC-DC Buck converters control.

Let us consider a Buck converter operating at a fixed switching frequency $f_{\text{sw}} = \frac{1}{T}$. With reference to the circuit diagram of the converter reported in Figure 4.7, let $i_L(t)$ denote the inductor current, $v_C(t)$ the capacitor voltage, $d(t)$ the duty cycle of the square wave voltage $v_{\text{sq}}(t)$, $i_o(t)$ the drawn output current, $V_{\text{in}}(t)$ the supply voltage, and $v_o(t)$ the output voltage.

As we shall see from the model derivation in the next section, the states $x(t)$, input $u(t)$, output $y(t)$, and external disturbances $\nu(t)$ of the Buck converter are the following:

$$x(t) = \begin{bmatrix} i_L(t) \\ v_C(t) \end{bmatrix}, \quad u(t) = d(t), \quad y(t) = v_o(t), \quad \nu(t) = \begin{bmatrix} i_o(t) \\ v_{\text{in}}(t) \end{bmatrix}, \tag{4.63}$$

where $v_{\text{in}}(t) = V_{\text{in}}(t) - \overline{V}_{\text{in}}$ and $\overline{V}_{\text{in}}$ is the nominal supply voltage.

In a DC-DC converter like the Buck, the main control task is regulating the output voltage $y(t) = v_o(t)$ towards a desired reference $y_r = \overline{V}_o$. Specifically, by Eq. (4.4), we regulate the system towards an admissible equilibrium triple $(x_r, u_r, y_r) = (\overline{x}, \overline{D}, \overline{V}_o)$.

Therefore, in order to formulate the QP-MPC problem (4.3) for controlling the Buck converter, and construct the related Ex-MPC policy (4.22), two key preliminary steps have to be performed, which will be carried out in the following sections:

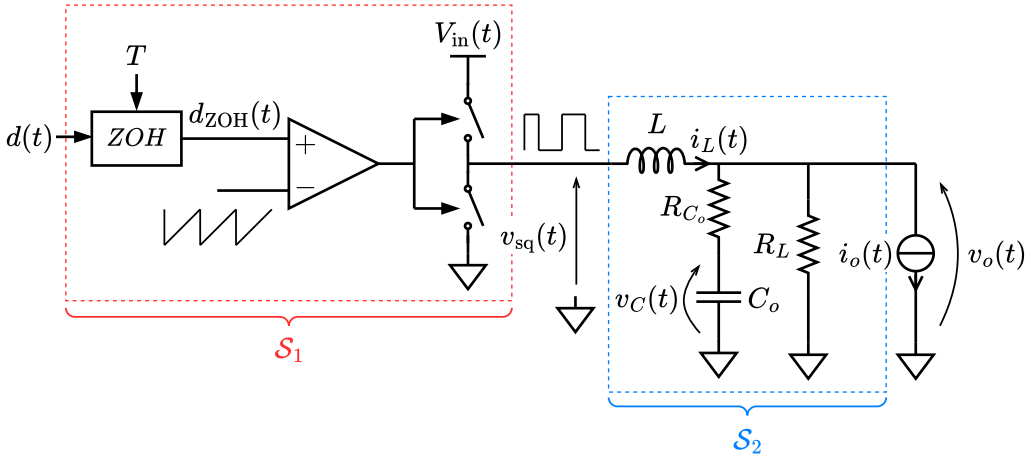


Figure 4.7. Buck converter circuit diagram.

1) As we shall see in the next section, the mathematical model of the Buck converter is continuous-time (CT) and strongly nonlinear, while, instead, the QP-MPC prediction model in Eqs. (4.1), (4.3c), and (4.3d) is DT and affine. Therefore, we need to introduce suitable transformations, namely discretization and linearization, to describe the Buck converter model into the desired form, matching Eq. (4.1).
2) The Ex-MPC policy (4.22) requires the measurement of the system states x_k and exogenous disturbances ν_k, which serve as its input parameters p_k (4.5). Consequently, both the states $x(t)$ and the external disturbances $\nu(t)$ of the Buck converter need to be either measured or estimated.

4.4.1. Buck Converter Mathematical Model

In the following, we rely on the first principles of physics to derive the mathematical model of the Buck converter.

We describe the Buck converter as the cascade of two subsystems (see Figure 4.7).

The first subsystem, $\mathcal{S}_1$, takes as inputs the duty cycle $d(t)$, in the form of a voltage signal ranging in $[0, 1]$ V, and the supply voltage $V_{\mathrm{in}}(t)$, with nominal value $\overline{V}_{\mathrm{in}}$, i.e., $V_{\mathrm{in}}(t) = \overline{V}_{\mathrm{in}} + v_{\mathrm{in}}(t)$. The output of $\mathcal{S}_1$ is the voltage $v_{\mathrm{sq}}(t)$. This subsystem consists of three parts: a zero-order hold (ZOH), a comparator, and two switches.

The ZOH samples $d(t)$ at each time instant $t = kT$, $k \in \mathbb{Z}_{\geq 0}$, and holds it constant at the value $d_k = d(kT)$ over the k-th switching period $I_k = [kT, (k+1)T)$. The ZOH output signal is given by

$$d_{\mathrm{ZOH}}(t) = d_k, \quad \forall\, t \in I_k. \tag{4.64}$$

Then, $d_{\mathrm{ZOH}}(t)$ is compared to a sawtooth wave, with fixed frequency f_{sw} and ranging in $[0, 1]$ V. This operation is performed by means of a comparator. When the sawtooth value is below $d_{\mathrm{ZOH}}(t)$, the comparator outputs a voltage V_H; conversely, when the sawtooth value is larger than $d_{\mathrm{ZOH}}(t)$, the comparator outputs a voltage V_L. The comparator output signal drives the two switches. One switch is connected to $V_{\mathrm{in}}(t)$ and is called the high side; the other is connected to the ground and called the low side. The switches are driven in anti-synchronized mode, i.e., when the comparator outputs V_H, the high side is open

and the low side is closed; while if the comparator outputs V_L, the high side is closed and the low side is open. As a result, the output of subsystem $\mathcal{S}_1$ is a square wave $v_{\text{sq}}(t)$ with duty cycle $d(t)$ and amplitude $V_{\text{in}}(t)$.

Accordingly, $\mathcal{S}_1$ is a nonlinear block that we describe according to the following definition:

$$\mathcal{S}_1(d(t), V_{\text{in}}(t)) = \begin{cases} V_{\text{in}}(t), & t \in [kT, (k+d_k)T), \\ 0, & t \in [(k+d_k)T, (k+1)T). \end{cases} \tag{4.65}$$

The subsystem $\mathcal{S}_2$ is dynamical and linear time-invariant (LTI). Its inputs are the square wave voltage $v_{\text{sq}}(t)$ from $\mathcal{S}_1$ and the drawn output current $i_o(t)$. The output of $\mathcal{S}_2$ is the controlled output voltage $v_o(t)$. Following the standard approach in circuit theory, we define the vector of states as the collection of inductor currents and capacitor voltages, i.e.,

$$x(t) = \begin{bmatrix} i_L(t) \\ v_C(t) \end{bmatrix}. \tag{4.66}$$

With this choice, we obtain the following state-space model for $\mathcal{S}_2$:

$$\mathcal{S}_2 : \begin{cases} \dot{x}(t) = A_c x(t) + B_{c,1} i_o(t) + B_{c,2} v_{\text{sq}}(t), \\ v_o(t) = C_c x(t) + D_{c,1} i_o(t), \end{cases} \tag{4.67}$$

where

$$A_c = \begin{bmatrix} -\frac{R_L \| R_{C_o}}{L} & -\frac{R_L}{L(R_{C_o}+R_L)} \\ \frac{R_L}{C_o(R_{C_o}+R_L)} & -\frac{1}{C_o(R_{C_o}+R_L)} \end{bmatrix}, \tag{4.68a}$$

$$B_{c,1} = \begin{bmatrix} \frac{R_L \| R_{C_o}}{L} \\ -\frac{R_L}{C_o(R_{C_o}+R_L)} \end{bmatrix}, \quad B_{c,2} = \begin{bmatrix} \frac{1}{L} \\ 0 \end{bmatrix}, \tag{4.68b}$$

$$C_c = \begin{bmatrix} R_L \parallel R_{C_o} & \frac{R_L}{R_L+R_{C_o}} \end{bmatrix}, \tag{4.68c}$$

$$D_{c,1} = -R_L \parallel R_{C_o}. \tag{4.68d}$$

We refer the reader to [50] for details.

Overall, the model $\mathcal{S}$ of the entire plant is a nonlinear system described by the interconnection of $\mathcal{S}_1$ and $\mathcal{S}_2$. Specifically,

$$v_o(t) = \mathcal{S}(d(t), i_o(t), V_{\text{in}}(t)) \tag{4.69a}$$

$$= \mathcal{S}_2(i_o(t), \mathcal{S}_1(d(t), V_{\text{in}}(t))). \tag{4.69b}$$

Notice that $v_o(t)$ depends on $d(t)$, which we use as the control input (i.e., $u(t) = d(t)$), but also on $i_o(t)$ and $v_{\text{in}}(t) = V_{\text{in}}(t) - \overline{V}_{\text{in}}$. The latter two quantities are external signals that we cannot act upon; therefore, we shall consider them as disturbances to be compensated, i.e., $\nu_1(t) = i_o(t)$ and $\nu_2(t) = V_{\text{in}}(t) - \overline{V}_{\text{in}} = v_{\text{in}}(t)$. Moreover, we refer to $y(t) = v_o(t)$ as the controlled output.

As previously noted, we need to describe the CT nonlinear Buck converter model (4.69) as a DT affine system, matching Eq. (4.1), so to obtain the QP-MPC prediction model (4.3c), (4.3d). To this end, we perform two consecutive steps: discretization and linearization.

Model Discretization and Linearization

We select the switching period T as the discrete time step for discretization. Henceforth, any quantity sampled at the switching time instants $t = kT$ will be denoted using the subscript k, i.e., $\star_k = \star(kT)$.

We address discretization leveraging the exact discretization approach (see Section 2.2, p. 12), i.e., by integrating Eq. (4.67) over the k-th switching period I_k. Thanks to the linearity of Eq. (4.67), we can evaluate the model prediction x_{k+1} in closed form using the following convolution integral:

$$x_{k+1} = e^{A_c T} x_k + \int_{kT}^{(k+1)T} e^{A_c((k+1)T-\tau)}(B_{c,1} i_o(t) + B_{c,2} v_{\text{sq}}(t))d\tau. \tag{4.70}$$

By linearity, Eq. (4.70) is equivalent to

$$\begin{aligned} x_{k+1} = e^{A_c T} x_k & \\ &+ e^{A_c(k+1)T} \int_{kT}^{(k+1)T} e^{-A_c\tau} B_{c,1} i_o(t) d\tau \\ &+ e^{A_c(k+1)T} \int_{kT}^{(k+1)T} e^{-A_c\tau} B_{c,2} v_{\text{sq}}(t) d\tau. \end{aligned} \tag{4.71}$$

The integrals in Eq. (4.71) are solved in closed form. To this end, we introduce the following assumption:

Assumption 4.2

The input voltage $V_{\text{in}}(t)$ and the output current disturbance $i_o(t)$ remain constant over the switching periods, i.e., $V_{\text{in}}(t) = V_{\text{in},k}$ and $i_o(t) = i_{o,k}$ for all $t \in I_k$ and $k \geq 0$.

Remark 4.7

Note that Assumption 4.2 is respected with a good approximation as long as:

1) the input voltage $V_{\text{in}}(t)$ slowly varies (i.e., its bandwidth is much lower than f_{sw});
2) the drawn output current $i_o(t)$ is constant almost everywhere (i.e., it is well modeled by a piecewise-constant signal, since it only changes due to sudden load connections or removals).

Under Assumption 4.2 and using $v_{\text{sq}}(t) = \mathcal{S}_1(d(t), V_{\text{in}}(t))$ as defined in Eq. (4.65), Eq. (4.71) simplifies to

$$\begin{aligned} x_{k+1} = e^{A_c T} x_k & \\ &+ e^{A_c(k+1)T} \int_{kT}^{(k+1)T} e^{-A_c\tau} B_{c,1} i_{o,k} d\tau \\ &+ e^{A_c(k+1)T} \int_{kT}^{(k+d_k)T} e^{-A_c\tau} B_{c,2} V_{\text{in},k} d\tau. \end{aligned} \tag{4.72}$$

Solving the integrals in Eq. (4.72) leads to the following DT model:

$$\begin{aligned} x_{k+1} &= f(x_k, d_k, i_{o,k}, V_{\text{in},k}) \\ &= e^{A_c T} x_k + (e^{A_c T} - I) A_c^{-1} B_{c,1} i_{o,k} + e^{A_c T}(I - e^{-A_c d_k T}) A_c^{-1} B_{c,2} V_{\text{in},k}. \end{aligned} \tag{4.73}$$

Proposition 4.1

Let $i_{o,k} = i_o$ and $V_{\text{in},k} = V_{\text{in}} = \overline{V}_{\text{in}} + v_{\text{in}}$, where i_o and v_{in} are constant values. The system (4.73) admits a unique equilibrium point, satisfying the output regulation task towards the desired constant reference $\overline{V}_o$, i.e., there exist unique $\overline{x} \in \mathbb{R}^{n_x}$ and $\overline{d} \in \mathbb{R}$ such that

$$\overline{x} = f(\overline{x}, \overline{d}, i_o, V_{\text{in}}), \tag{4.74a}$$

$$\overline{V}_o = C\overline{x}. \tag{4.74b}$$

Proof

Solving Eq. (4.74a) for $\overline{x}$ and replacing it into Eq. (4.74b) yields

$$\overline{V}_o = C(I - A)^{-1}(A - A^{1-\overline{d}})A_c^{-1}B_{c,2}V_{\text{in}} - CA_c^{-1}B_{c,1}i_o, \tag{4.75}$$

with $A = e^{A_c T}$. Existence is guaranteed by noting that the scalar function of $\overline{d}$ on the right-hand side of Eq. (4.75), defined over the interval $0 \leq \overline{d} \leq 1$, admits as image the interval $-CA_c^{-1}B_{c,1}i_o \leq \overline{V}_o \leq -CA_c^{-1}B_{c,1}i_o + CA_c^{-1}B_{c,2}V_{\text{in}}$. Uniqueness is a direct consequence of the fact that the function $\overline{V}_o(\overline{d})$ is monotonically strictly increasing.

Remark 4.8

The value of $\overline{d}$ can be computed using, e.g., the Newton's method to solve the nonlinear equation (4.75), or by taking a linear approximation of the exponential function. In the undisturbed case, i.e., $i_{o,k} = 0$ and $V_{\text{in},k} = \overline{V}_{\text{in}}$, the latter approach leads to

$$\overline{D} = \frac{\overline{V}_o}{\overline{V}_{\text{in}}}, \tag{4.76}$$

which is a good approximation in all our numerical tests and matches the classical assessment of steady-state duty cycle [50].

Since the model in Eq. (4.73) is nonlinear, we perform linearization on it. Specifically, the DT model (4.73) is nonlinear in d_k and $V_{\text{in},k}$ due to the presence of the term $e^{-A_c d_k T}V_{\text{in},k}$. To proceed with the required linearization, we introduce the following assumption:

Assumption 4.3

The quantities d_k and $V_{\text{in},k}$ are well described by

$$d_k = \overline{D} + \delta_k, \quad V_{\text{in},k} = \overline{V}_{\text{in}} + v_{\text{in},k}, \tag{4.77}$$

where δ_k and $v_{\text{in},k}$ are small variations around the nominal values $\overline{D}$ and $\overline{V}_{\text{in}}$, respectively.

Assumption 4.3 is a standard assumption in set-point regulation. Moreover, as we will demonstrate in Section 4.4.4, we are able to guarantee the local robust stability of the closed-loop system when the plant is described by the non-approximated nonlinear model in Eq. (4.73). Therefore, Assumption 4.3 is only used to construct the DT affine prediction model for the QP-MPC problem (4.3), but is not critical for stability or performance.

Let us consider the nonlinearity of model (4.73), i.e.,

$$g(\delta_k, v_{\text{in},k}) = e^{A_c T}(I - e^{-A_c(\overline{D}+\delta_k)T})A_c^{-1}B_{c,2}(\overline{V}_{\text{in}} + v_{\text{in},k}). \tag{4.78}$$

Under Assumption 4.3, we can linearize Eq. (4.78) around the point $(\delta_k, v_{\text{in},k}) = (0, 0)$ as follows:

$$\begin{aligned} g(\delta_k, v_{\text{in},k}) &\approx g(0,0) + \frac{\partial g}{\partial \delta_k}(0,0)\,\delta_k + \frac{\partial g}{\partial v_{\text{in},k}}(0,0)\,v_{\text{in},k} \\ &= e^{A_c T}(I - e^{-A_c \overline{D} T})A_c^{-1}B_{c,2}\overline{V}_{\text{in}} \\ &\quad + e^{A_c(1-\overline{D})T}TB_{c,2}\overline{V}_{\text{in}}\,\delta_k \\ &\quad + e^{A_c T}(I - e^{-A_c \overline{D} T})A_c^{-1}B_{c,2}\,v_{\text{in},k}. \end{aligned} \tag{4.79}$$

Finally, using Eq. (4.79) to replace $g(\delta, v_{\text{in},k})$ in Eq. (4.73), we obtain a linearization that matches the structure of Eq. (4.1), i.e.,

$$x_{k+1} = Ax_k + Bd_k + B_{\nu_1} i_{o,k} + B_{\nu_2} v_{\text{in},k} + b, \tag{4.80a}$$
$$v_{o,k} = Cx_k + Dd_k + D_{\nu_1} i_{o,k} + D_{\nu_2} v_{\text{in},k} + d, \tag{4.80b}$$

where

$$A = e^{A_c T}, \qquad C = C_c, \tag{4.81a}$$
$$B = e^{A_c(1-\overline{D})T}TB_{c,2}\overline{V}_{\text{in}}, \qquad D = \mathbf{0}_{n_y \times n_u}, \tag{4.81b}$$
$$B_{\nu_1} = (e^{A_c T} - I)A_c^{-1}B_{c,1}, \qquad D_{\nu_1} = D_{c,1}, \tag{4.81c}$$
$$B_{\nu_2} = e^{A_c T}(I - e^{-A_c \overline{D} T})A_c^{-1}B_{c,2}, \qquad D_{\nu_2} = \mathbf{0}_{n_y \times n_\nu}, \tag{4.81d}$$
$$b = e^{A_c(1-\overline{D})T}\left[(e^{A_c \overline{D} T} - I)A_c^{-1} - T\overline{D}\,I\right]B_{c,2}\overline{V}_{\text{in}}, \qquad d = \mathbf{0}_{n_y}. \tag{4.81e}$$

Remark 4.9

We highlight that, since the linearization is performed around the nominal values $\overline{D}$ and $\overline{V}_{\text{in}}$, the equilibrium point $(\overline{x}, \overline{D}, \overline{V}_o)$ of the nonlinear system (4.73) is also an equilibrium of the linearized system. Consequently, the equilibrium condition in Eq. (4.4) is met.

4.4.2. Output Current Disturbance and State Estimation

As previously noted, the Ex-MPC policy (4.22) requires the measurement of the system states x_k and exogenous disturbances ν_k, which serve as its input parameters p_k (4.5). Consequently, both the states $x(t)$ and the external disturbances $\nu(t)$ of the Buck converter need to be either measured or estimated. In this section, we propose a simple and cheap design of devices estimating $\nu(t)$ and $x(t)$.

Disturbances Estimation

Let us recall the external disturbances $\nu(t)$ of the Buck converter, i.e.:

- $\nu_1(t) = i_o(t)$. The output current drawn from the load is not directly available for measurement; therefore, it must be estimated.

- $\nu_2(t) = v_{\text{in}}(t)$. Since $V_{\text{in}}(t)$ is available for measurement, $v_{\text{in}}(t)$ is trivially obtained as

$$v_{\text{in}}(t) = V_{\text{in}}(t) - \overline{V}_{\text{in}} \tag{4.82}$$

  and does not need to be estimated.

In the remainder of this section, we describe a method for estimating $\hat{i}_o(t)$ of $i_o(t)$. Specifically, we resort to the linear estimator proposed in [125], based on algebraic design.

Let us consider the transfer function description of the linear Buck subsystem $\mathcal{S}_2$, augmented with $i_L(t)$ as an additional output (in the Laplace domain), i.e.,

$$\begin{bmatrix} v_o(s) \\ i_L(s) \end{bmatrix} = \begin{bmatrix} P_{11}(s) & P_{12}(s) \\ P_{21}(s) & P_{22}(s) \end{bmatrix} \begin{bmatrix} v_{\text{sq}}(s) \\ i_o(s) \end{bmatrix}. \tag{4.83}$$

The problem of estimating the input i_o, given the measurements of v_o and i_L, is a linear algebraic problem. Indeed, we can easily see that

$$i_o(s) = \frac{1}{P_{22}(s)}(i_L(s) - P_{21}(s)v_{\text{sq}}(s)), \tag{4.84a}$$

$$v_{\text{sq}}(s) = \frac{1}{P_{11}(s)}(v_o(s) - P_{12}(s)i_o). \tag{4.84b}$$

Replacing Eq. (4.84b) into Eq. (4.84a) and performing suitable algebraic manipulations, we obtain

$$i_o(s) = \begin{bmatrix} -E_1(s) & 1 \end{bmatrix} \begin{bmatrix} v_o(s) \\ i_L(s) \end{bmatrix}, \tag{4.85}$$

where

$$E_1(s) = \frac{P_{21}(s)}{P_{11}(s)} = \frac{C_o(R_L + R_{C_o})s + 1}{R_L(1 + CR_{C_o}s)}. \tag{4.86}$$

We refer the reader to [125] for the complete proof and a detailed discussion.

Overall, the drawn output current estimate $\hat{i}_o(t)$ is obtained by

$$\hat{i}_o(s) = \mathcal{E}(s) \begin{bmatrix} v_o(s) \\ i_{L,\text{sc}}(s) \end{bmatrix}, \quad \mathcal{E}(s) = \begin{bmatrix} -E_1(s) & \frac{1}{g_{i_L}} \end{bmatrix}, \tag{4.87}$$

where $i_{L,\text{sc}}(t) = g_{i_L} i_L(t)$ is the available measurement of the inductor current, which is scaled by $g_{i_L} > 0$, that is the corresponding sensor gain.

Remark 4.10

The design of the estimator is entirely conducted in CT and relies on signals involved in the linear subsystem only. Consequently, the problem is solved exactly by the CT LTI filter $\mathcal{E}(s)$.

In ideal conditions, $\hat{i}_o(t) = i_o(t)$ for all $t \in \mathbb{R}_{\geq 0}$, but the presence of noise and modeling uncertainties in the plant description may hinder the quality of the estimate.

Concerning noise, an appropriate printed circuit board (PCB) design easily allows us to have negligible parasitic effects perturbing the measured signals before being processed by $\mathcal{E}$.

As to modeling uncertainties, we notice that only R_L, C_o, and R_{C_o} are involved in the design, thus uncertainties on any other component are irrelevant. We study the impact of the uncertainties on R_L, C_o, and R_{C_o} through the following sensitivity analysis:

$$S_{C_o}^{E_1}(s) = \frac{\partial E_1}{\partial C_o} = \frac{1}{(C_o R_{C_o})^2} \frac{s}{(s + \frac{1}{C_o R_{C_o}})^2}, \tag{4.88a}$$

$$S_{R_{C_o}}^{E_1}(s) = \frac{\partial E_1}{\partial R_{C_o}} = -\frac{1}{R_{C_o}^2} \frac{s^2}{(s + \frac{1}{C_o R_{C_o}})^2}, \tag{4.88b}$$

$$S_{R_L}^{E_1}(s) = \frac{\partial E_1}{\partial R_L} = -\frac{1}{R_L^2}. \tag{4.88c}$$

As $s \to 0$, both $|S_{C_o}^{E_1}| \to 0$ and $|S_{R_{C_o}}^{E_1}| \to 0$, indicating that uncertainty on C_o and R_{C_o} poorly influences the estimate $\hat{i}_o$ at low frequencies. Instead, $|S_{R_L}^{E_1}|$ is non-zero, indicating that the uncertainty of R_L leads to some steady-state estimation error $|\hat{i}_o(t) - i_o(t)|$. We proceed by studying this error. Let $\hat{C}_o$, $\hat{R}_{C_o}$, and $\hat{R}_L$ denote the nominal values of the components used to design $\mathcal{E}$, and $\hat{E}_1$ the transfer function E_1, as defined in Eq. (4.86) but evaluated using $\hat{C}_o$, $\hat{R}_{C_o}$, and $\hat{R}_L$. At steady state, v_o is constant, since $v_o(s) = \overline{V}_o/s$. By the final value theorem, we obtain

$$\begin{aligned} \lim_{t\to\infty} \left|\hat{i}_o(t) - i_o(t)\right| &= \lim_{s\to 0} \left| s \left(\hat{i}_o(s) - i_o(s)\right)\right| \\ &= \lim_{s\to 0} \left| s \left(\hat{E}_1(s) - E_1(s)\right) \frac{\overline{V}_o}{s} \right| = \overline{V}_o \left| \frac{1}{\hat{R}_L} - \frac{1}{R_L} \right|. \end{aligned} \tag{4.89}$$

Such an error is negligible when $R_L, \hat{R}_L \to \infty$ or $\hat{R}_L \to R_L$. To counteract the effect of the uncertainty on R_L, we propose a robust design of $\hat{E}_1$, optimally selecting $\hat{R}_L$ to minimize the expected value of the error. Formally, let us assume that R_L is a random variable with uniform distribution in the interval $[R_{L,\min}, R_{L,\max}]$. Then, we aim to compute

$$\hat{R}_L^* = \underset{\hat{R}_L \in \mathbb{R}}{\arg\min}\, \mathbb{E}\left[\left| \frac{1}{\hat{R}_L} - \frac{1}{R_L} \right| \right], \quad R_L \sim U([R_{L,\min}, R_{L,\max}]). \tag{4.90}$$

Proposition 4.2 (Optimal estimator $\hat{R}_L^*$)

The solution to problem (4.90) is given by the mean resistance

$$\hat{R}_L^* = \frac{1}{2}(R_{L,\min} + R_{L,\max}). \tag{4.91}$$

Proof

Consider the change of coordinates $a = 1/\hat{R}_L$ and $B = 1/R_L$. The probability density function of B is given by the following change of variables formula [36, Section 2.3]:

$$f_B(b) = \begin{cases} \frac{1}{R_{L,\max} - R_{L,\min}} b^{-2} & \text{if } \frac{1}{R_{L,\min}} \le b \le \frac{1}{R_{L,\max}}, \\ 0 & \text{otherwise.} \end{cases} \tag{4.92}$$

After the change of variables, the problem takes the form

$$\hat{a}^* = \arg\min_{a\in\mathbb{R}} \mathbb{E}\left[\,|a-B|\,\right]. \tag{4.93}$$

It is a standard result that the optimum of such a problem corresponds to the median of the distribution of B [90, Chapter 1], i.e., a^* satisfies

$$\int_{-\infty}^{a^*} f_B(b)db = \frac{1}{2}. \tag{4.94}$$

Computing the integral, we get

$$\frac{1}{2} = \frac{1}{R_{L,\max}-R_{L,\min}}\left[-\frac{1}{b}\right]_{R_{L,\max}^{-1}}^{a^*} = \frac{R_{L,\max}-a^*}{R_{L,\max}-R_{L,\min}}. \tag{4.95}$$

Solving for a^* and taking its inverse yields the desired result.

The estimator $\mathcal{E}(s)$, defined in Eq. (4.87), takes only the measurements of i_L and v_o as inputs. Moreover, $E_1(s)$ is a first-order filter and i_L is not filtered. These features allow for a cheap circuital implementation of $\mathcal{E}(s)$. In the remainder of this section, we detail such an implementation, which differs from the one considered in [125].

We start from the definition of $E_1(s)$ in Eq. (4.86), which can be equivalently rewritten as follows:

$$E_1(s) = \frac{1}{R_L}\frac{1+\frac{s}{(C_o(R_L+R_{C_o}))^{-1}}}{1+\frac{s}{(C_oR_{C_o})^{-1}}} = K_{\mathcal{E}}\frac{1+\frac{s}{z_{\mathcal{E}}}}{1+\frac{s}{p_{\mathcal{E}}}}, \tag{4.96}$$

where

$$K_{\mathcal{E}} = \frac{1}{R_L}, \quad z_{\mathcal{E}} = \frac{1}{C_o(R_L+R_{C_o})}, \quad p_{\mathcal{E}} = \frac{1}{C_oR_{C_o}}. \tag{4.97}$$

Since $R_{C_o}+R_L \gg R_{C_o}$, we have that $z_{\mathcal{E}} < p_{\mathcal{E}}$ and $E_1(s)$ is a high-pass filter. Consequently, we can resort to the circuital implementation of $\mathcal{E}(s)$ shown in Figure 4.8. To avoid that the output of the OP-AMP $\mathrm{OP}^{\mathcal{E}}$ exceeds the saturation limits, we scale the gain of $\mathcal{E}$ by a factor $g_{i_o} \in (0,1)$, obtaining a scaled version of the current estimate $\hat{i}_{o,\mathrm{sc}} = g_{i_o}\hat{i}_o$, i.e.,

$$\hat{i}_{o,\mathrm{sc}}(s) = g_{i_o}\mathcal{E}(s)\begin{bmatrix} v_o(s) \\ i_{L,\mathrm{sc}}(s)\end{bmatrix} = \begin{bmatrix} -g_{i_o}E_1(s) & \frac{g_{i_o}}{g_{i_L}}\end{bmatrix}\begin{bmatrix} v_o(s) \\ i_{L,\mathrm{sc}}(s)\end{bmatrix}. \tag{4.98}$$

The transfer functions of the circuit in Figure 4.8, from inputs v_o and $i_{L,\mathrm{sc}}$ to output $\hat{i}_{o,\mathrm{sc}}$, are given by

$$\frac{\hat{i}_{o,\mathrm{sc}}(s)}{v_o(s)} = -\frac{R_5^{\mathcal{E}}}{R_4^{\mathcal{E}}}\frac{1+sR_4^{\mathcal{E}}C_1^{\mathcal{E}}}{1+sR_5^{\mathcal{E}}C_2^{\mathcal{E}}}, \tag{4.99a}$$

$$\frac{\hat{i}_{o,\mathrm{sc}}(s)}{i_{L,\mathrm{sc}}(s)} = \frac{R_1^{\mathcal{E}}}{R_1^{\mathcal{E}}+R_2^{\mathcal{E}}}\frac{1+R_5^{\mathcal{E}}(R_3^{\mathcal{E}} \parallel R_4^{\mathcal{E}})^{-1}+sR_5(C_1+C_2)}{1+sR_5C_2}. \tag{4.99b}$$

By imposing the conditions

$$g_{i_o}K_{\mathcal{E}} = \frac{R_5^{\mathcal{E}}}{R_4^{\mathcal{E}}}, \quad p_{\mathcal{E}} = \frac{1}{R_5^{\mathcal{E}}C_2^{\mathcal{E}}}, \quad z_{\mathcal{E}} = \frac{1}{R_4^{\mathcal{E}}C_1^{\mathcal{E}}}, \tag{4.100}$$

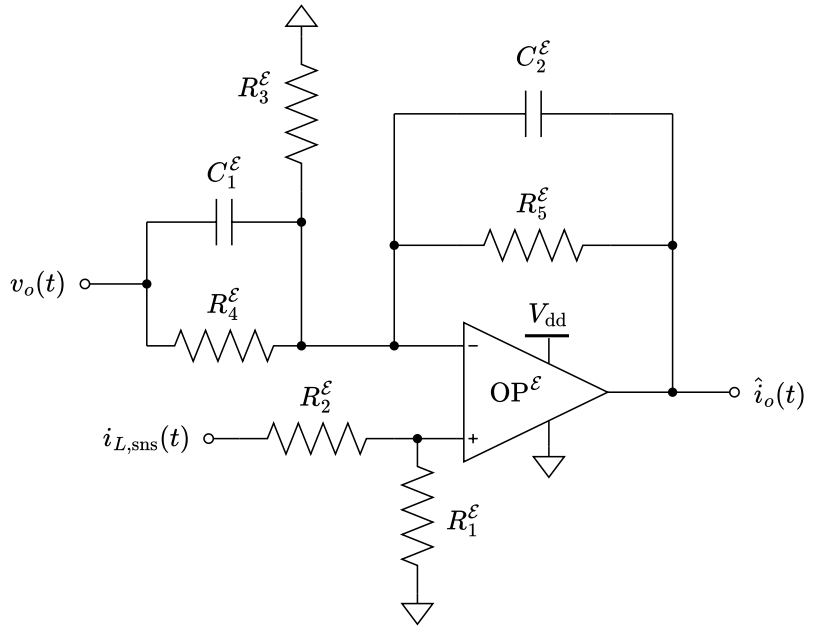


Figure 4.8. Estimator for $i_o(t)$.

we implement the $E_1(s)$ transfer function. Then, we select $R_1^{\mathcal{E}}$, $R_2^{\mathcal{E}}$, and $R_3^{\mathcal{E}}$ such that

$$C_1^{\mathcal{E}}(R_3^{\mathcal{E}} \parallel R_4^{\mathcal{E}}) = C_2^{\mathcal{E}} R_5^{\mathcal{E}}, \quad \frac{R_1^{\mathcal{E}}}{R_1^{\mathcal{E}} + R_2^{\mathcal{E}}} = \frac{g_{i_o}}{g_{i_L}}, \tag{4.101}$$

ensuring that Eq. (4.99b) equals $\frac{g_{i_o}}{g_{i_L}}$.

State Estimation

Let us now focus on the states $x(t)$ of the Buck converter, i.e., $x(t) = [i_L(t),\ v_C(t)]^\top$. The inductor current $i_L(t)$ can easily be sensed (see, e.g., [125]), but the capacitor voltage $v_C(t)$ is not accessible, due to the presence of the capacitor equivalent series resistance (ESR), i.e., R_{C_o}. Therefore, the estimation of $v_C(t)$ is needed. This section presents a simple algebraic estimator design for $v_C(t)$.

Starting from the Kirchhoff's current law on the output node of the Buck converter, we have that

$$i_C = i_o - i_L - \frac{v_o}{R_L} = \frac{v_o - v_C}{R_{C_o}}. \tag{4.102}$$

Then, given the measurements of i_L and v_o, and the estimate $\hat{i}_o$ of i_o, we define the estimator

$$\hat{v}_C = R_{C_o}(\hat{i}_o - i_L) + \left(\frac{R_{C_o}}{R_L} + 1\right) v_o. \tag{4.103}$$

We highlight that, with this procedure, $\hat{v}_C(t)$ is obtained using only weighted sums. Consequently, we argue that, compared to the conventional strategy of employing an observer, the proposed approach streamlines the design and allows for a low-cost analog implementation.

Remark 4.11

In many practical applications, $R_{C_o} \ll R_L$. Under this assumption, Eq. (4.103) can be simplified to the straightforward relation $\hat{v}_C = v_o$, meaning that the controlled output voltage can be directly used as a reliable estimate of the state $v_C(t)$. This choice simplifies the circuit design of the system and reduces its costs both in economic terms and board area.

4.4.3. Explicit MPC Design

We can now proceed with constructing the Ex-MPC policy (4.22) to control the Buck converter.

First, following Section 4.2, we formulate the QP-MPC problem as follows:

$$\min_{\hat{\boldsymbol{x}},\hat{\boldsymbol{u}},\hat{\boldsymbol{y}}} \quad J(\hat{\boldsymbol{y}},\hat{\boldsymbol{u}}) = \sum_{i=0}^{N_p-1} \left(\|\hat{v}_{o,i} - \overline{V}_o\|_Q^2 + \|\hat{d}_i - \overline{D}\|_R^2 \right) + \sum_{i=1}^{N_p-1} \|\hat{d}_i - \hat{d}_{i-1}\|_{R_\Delta}^2 \tag{4.104a}$$

$$\text{s.t.} \quad \hat{x}_0 = x_k = \begin{bmatrix} i_{L,k} \\ \hat{v}_{C,k} \end{bmatrix}, \tag{4.104b}$$

$$\hat{x}_{i+1} = A\hat{x}_i + B\hat{d}_i + B_{\nu_1}\hat{i}_{o,k} + B_{\nu_2}(V_{\text{in},k} - \overline{V}_{\text{in}}) + b, \quad i = 0, \dots, N_p - 1, \tag{4.104c}$$

$$\hat{v}_{o,i} = C\hat{x}_i + D_{\nu_1}\hat{i}_{o,k}, \quad i = 0, \dots, N_p - 1, \tag{4.104d}$$

$$0 \le \hat{d}_i \le 1, \quad i = 0, \dots, N_p - 1. \tag{4.104e}$$

Problem (4.104) is a re-adaptation of the more general version (4.3), in which we have considered the Buck converter states $x = [i_L,\ \hat{v}_C]$, input $u = d$, output $y = v_o$, and disturbances $\nu = [\hat{i}_o,\ v_{\text{in}}]$, as in Eq. (4.63). Particularly, as explained in Section 4.4.2, the quantities $v_{C,k}$ and $i_{o,k}$ are estimated, i.e.,

$$v_{C,k} = \hat{v}_{C,k}, \quad i_{o,k} = \hat{i}_{o,k} = \frac{1}{g_{i_o}}\hat{i}_{o,\text{sc},k}, \tag{4.105}$$

where $g_{i_o} \in (0, 1)$ the scaling factor of the i_o estimate. Moreover, $v_{\text{in},k} = V_{\text{in},k} - \overline{V}_{\text{in}}$, where $V_{\text{in},k}$ is available for measurement, and $i_{L,k} = \frac{1}{g_{i_L}} i_{L,\text{sc},k}$, where $g_{i_L} > 0$ is the gain of the sensor used to measure i_L.

The cost function J in Eq. (4.104a) contains, as reference for regulation, the reference output voltage $y_r = \overline{V}_o$ and duty cycle $u_r = \overline{D} = \frac{\overline{V}_o}{\overline{V}_{\text{in}}}$, as defined in Proposition 4.1 and Remarks 4.8, 4.9.

The prediction model in Eqs. (4.104c), (4.104d) is given by the DT affine Buck converter model (4.80).

As for the linear constraints (4.104e), we only impose upper-lower bound constraints on the input, i.e., the duty cycle d, since, by definition, it can only range between 0 (0%) and 1 (100%). This yields the constraint matrices $H_u = [1,\ -1]^\top$ and $h_u = [1,\ 0]^\top$ in Eq. (4.3f).

Then, the QP-MPC problem (4.104) has the following parameters:

$$\begin{aligned} p_k &= [i_{L,k},\ \hat{v}_{C,k},\ \hat{i}_{o,k},\ V_{\text{in},k}]^\top \\ &= \left[\tfrac{1}{g_{i_L}} i_{L,\text{sc},k},\ \hat{v}_{C,k},\ \tfrac{1}{g_{i_o}}\hat{i}_{o,\text{sc},k},\ V_{\text{in},k}\right]^\top. \end{aligned} \tag{4.106}$$

Finally, following Section 4.3, we can construct the Ex-MPC policy associated with problem (4.104), i.e.,

$$\begin{aligned} u_k = d_k = \pi(p_k) &= \pi([i_{L,k},\ \hat{v}_{C,k},\ \hat{i}_{o,k},\ V_{\text{in},k}]^\top) \\ &= \pi\left(\left[\tfrac{1}{g_{i_L}} i_{L,\text{sc},k},\ \hat{v}_{C,k},\ \tfrac{1}{g_{i_o}}\hat{i}_{o,\text{sc},k},\ V_{\text{in},k}\right]^\top\right), \end{aligned} \tag{4.107}$$

and implement it as a fully-analog electronic circuit. We will analyze this design step in more detail in Section 4.5.

4.4.4. Stability Analysis

In this section, we establish local robust stability guarantees for the closed-loop system, i.e., the DT nonlinear Buck converter model (4.73), controlled by the Ex-MPC policy (4.107), in presence of constant load and line disturbances.

Proposition 4.3 (Local robust stability)

Consider the DT nonlinear Buck converter model in Eq. (4.73), i.e.,

$$\begin{aligned} x_{k+1} &= f(x_k, d_k, i_{o,k}, V_{\text{in},k}) \\ &= \underbrace{e^{A_c T}}_{A} x_k + (e^{A_c T} - I) A_c^{-1} B_{c,1} i_{o,k} + e^{A_c T}(I - e^{-A_c d_k T}) A_c^{-1} B_{c,2} V_{\text{in},k}. \end{aligned} \tag{4.108}$$

Assume that $V_{\text{in},k} = V_{\text{in}} = \overline{V}_{\text{in}} + v_{\text{in}}$ and $i_{o,k} = i_o$, where v_{in} and i_o are constant values.

Also, consider the Ex-MPC policy (4.107). Assume that the states are available (v_C does not need estimation), so that

$$d_k = \pi([x_k^\top,\ i_o,\ V_{\text{in}}]^\top) \tag{4.109a}$$

$$= \begin{cases} [K_{i,1},\ K_{i,2}]x_k + K_{i,3} i_o + K_{i,4} V_{\text{in}} + l_i & \text{if } x_k \in \mathcal{R}_i,\ i \in \mathcal{I}_{\text{unsat}}, \\ 0 & \text{if } x_k \in \underline{\mathcal{S}}, \\ 1 & \text{if } x_k \in \overline{\mathcal{S}}. \end{cases} \tag{4.109b}$$

Then, the closed-loop system (4.108), (4.109) admits an equilibrium state $\overline{x}$.

Now, let $\zeta(i_o, v_{\text{in}}) = e^{A_c T(\overline{D} - \overline{d}(i_o, v_{\text{in}}))}$, where $\overline{D}$ is given by Eq. (4.75) in the undisturbed case (i.e., $V_{\text{in},k} = \overline{V}_{\text{in}}$ and $i_{o,k} = 0$), and $\overline{d}(i_o, v_{\text{in}}) = \pi(\overline{p})$, with $\overline{p} = [\overline{x}^\top,\ i_o,\ V_{\text{in}}]^\top$, according to Eq. (4.109).

Let $\mathcal{R}_r$ be the unsaturated region containing $\overline{p}$. Assume that there exists a matrix $P \in \mathbb{R}^{2\times 2}$ such that $\|A + BK_r\|_P \leq 1 - \epsilon$, where $0 < \epsilon < 1$ and $K_r = [K_{r,1}, K_{r,2}]$.

If the disturbances i_o and v_{in} are such that

$$\|\zeta(i_o, v_{\text{in}}) - I\|_P + \frac{v_{\text{in}}}{\overline{V}_{\text{in}}} \|\zeta(i_o, v_{\text{in}})\|_P < \frac{\epsilon}{\|BK_r\|_P}, \tag{4.110}$$

then the equilibrium state $\overline{x}$ is locally stable.

Proof

First, we prove the existence of the equilibrium state $\overline{x}$ satisfying the closed-loop equilibrium condition

$$(I - A)\overline{x} = (A - I)A_c^{-1}B_{c,1}i_o + (A - A^{1-\overline{d}})A_c^{-1}B_{c,2}V_{\text{in}}, \tag{4.111a}$$
$$\overline{d} = \pi([\overline{x}^\top, i_o, V_{\text{in}}]^\top). \tag{4.111b}$$

Let $g(\overline{x}, i_o, V_{\text{in}}) = (A - A^{1-\pi([\overline{x}^\top, i_o, V_{\text{in}}]^\top)})A_c^{-1}B_{c,2}V_{\text{in}}$. We observe that, for all $[\overline{x}^\top, i_o, V_{\text{in}}]^\top \in \mathcal{P}$, which is a convex and compact set by Assumption 4.1, it holds that $0 \leq \overline{d} \leq 1$, by construction of the Ex-MPC policy π (4.109); therefore, the function g is continuous and bounded over $\mathcal{P}$. Also, the matrix $I - A$ is invertible. Then, we can leverage Brouwer's fixed-point theorem [49], which states that, under the above conditions, Eq. (4.111) admits at least one solution $\overline{x}$. Note that, the value of $\overline{x}$ depends only on i_o and v_{in}.

Next, we prove the local stability of $\overline{x}$, robustly with respect to the disturbances i_o and v_{in}. First, let us assume that the region $\mathcal{R}_r$ contains a neighborhood of $\overline{p} = [\overline{x}^\top, i_o, V_{\text{in}}]^\top$. Then, in such a neighborhood, the closed-loop system (4.108), (4.109) evolves under the following equation:

$$\begin{aligned} x^+ = F(x) = e^{A_cT}x + (e^{A_cT} - I)A_c^{-1}B_{c,1}i_o \\ + e^{A_cT}(I - e^{-A_c(K_rx + K_{r,3}i_o + K_{r,4}V_{\text{in}} + l_r)T})A_c^{-1}B_{c,2}V_{\text{in}}. \end{aligned} \tag{4.112}$$

We want so show that Eq. (4.112) locally defines a contraction in the P-norm, which is equivalent to the following condition [28]:

$$\left\| \frac{\partial F}{\partial x}(\overline{x}) \right\|_P < 1. \tag{4.113}$$

For system (4.112), we have that

$$\frac{\partial F}{\partial x}(\overline{x}) = e^{A_cT} + e^{A_c(1 - K_r\overline{x} - K_{r,3}i_o - K_{r,4}V_{\text{in}} - l_r)T}TB_{c,2}V_{\text{in}}K_r. \tag{4.114}$$

Recalling from Eq. (4.81) that $B = e^{A_c(1-\overline{D})T}TB_{c,2}\overline{V}_{\text{in}}$ and letting $\overline{d}(i_o, v_{\text{in}}) = \pi(\overline{p})$, we can rewrite

$$\frac{\partial F}{\partial x}(\overline{x}) = A + \Delta(i_o, v_{\text{in}})BK_r, \quad \Delta(i_o, v_{\text{in}}) = \underbrace{e^{A_c(\overline{D} - \overline{d}(i_o, v_{\text{in}}))T}}_{\zeta(i_o, v_{\text{in}})} \left(1 + \frac{v_{\text{in}}}{\overline{V}_{\text{in}}}\right). \tag{4.115}$$

A sufficient condition to satisfy Eq. (4.113) is that

$$\begin{aligned} \|A + \Delta BK_r\|_P &\leq \|A + BK_r\|_P + \|\Delta - I\|_P \|BK_r\|_P \\ &= \|A + BK_r\|_P + \left\| \zeta \left(1 + \frac{v_{\text{in}}}{\overline{V}_{\text{in}}}\right) - I \right\|_P \|BK_r\|_P \\ &\leq \|A + BK_r\|_P + \left(\|\zeta - I\|_P + \frac{v_{\text{in}}}{\overline{V}_{\text{in}}} \|\zeta\|_P \right) \|BK_r\|_P < 1. \end{aligned} \tag{4.116}$$

Since, by assumption, $\|A + BK_r\|_P \leq 1 - \epsilon$, condition (4.116) becomes

$$\|\zeta - I\|_P + \frac{v_{\text{in}}}{\overline{V}_{\text{in}}} \|\zeta\|_P < \frac{\epsilon}{\|BK_r\|_P}, \tag{4.117}$$

matching Eq. (4.110).

It is worth noting that, if the region $\mathcal{R}_r$ also contains $\overline{p}' = [\overline{x}'^\top, 0, \overline{V}_{\text{in}}]^\top$, where $\overline{x}'$ is the undisturbed closed-loop equilibrium state (with related input $\overline{D}$), then the local feedback matrix K_r coincides with the LQR matrix, arising from the infinite-horizon unconstrained version of the QP-MPC problem (4.3), under the assumption that the prediction horizon is sufficiently large and no move blocking is considered (refer to Example 2.2 for more details). In these ideal conditions, the matrix $A + BK_r$ is guaranteed to be Schur stable; thus, there surely exist a matrix P and a scalar $\epsilon > 0$ such that $\|A + BK_r\|_P = 1 - \epsilon$. In our setting, the adopted move blocking strategy induces only mild perturbations compared to the original problem (4.3), so the resulting closed-loop matrix is expected to remain stabilizing.

Nevertheless, the Schur stability property of $A + BK_r$ can be verified a posteriori and serves only as a preliminary condition, as the local stability guarantees in Proposition 4.3 provide a stronger result in terms of robustness. Indeed, beyond asymptotic stability, our analysis quantifies the admissible disturbance set, defined by Eq. (4.110), for which the nonlinear closed-loop system (4.108), (4.109) remains stable.

It is possible to obtain a polytopic approximation of such a disturbance set. First, we recall that, for any induced matrix norm, the following inequalities hold for the matrix exponential:

$$\|e^M\|_P \leq e^{\|M\|_P}, \quad \|e^M - I\|_P \leq e^{\|M\|_P} - 1. \tag{4.118}$$

Defining $\Delta d = \overline{d} - \overline{D}$ and letting $M = -A_c T \Delta d$, by Eq. (4.118) it holds that $\|\zeta\|_P \leq e^{\alpha|\Delta d|}$ and $\|\zeta - I\|_P \leq e^{\alpha|\Delta d|} - 1$, where $\alpha = \|A_c\|_P T$. Then, Eq. (4.110) can be upper-bounded as follows:

$$\|\zeta - I\|_P + \frac{v_{\text{in}}}{\overline{V}_{\text{in}}}\|\zeta\|_P \leq \left(1 - \frac{v_{\text{in}}}{\overline{V}_{\text{in}}}\right) e^{\alpha|\Delta d|} < 1 + \beta, \tag{4.119}$$

where $\beta = \frac{\epsilon}{\|BK_r\|_P}$.

Now, we want to express Δd as a function of the disturbances i_o and v_{in} only. To do so, let us recall the closed-loop equilibrium condition within region $\mathcal{R}_r$, i.e., by Eq. (4.112),

$$\begin{aligned} H(x, i, V) = {} & e^{A_c T} x + (e^{A_c T} - I) A_c^{-1} B_{c,1} i \\ & + e^{A_c T}(I - e^{-A_c(K_r x + K_{r,3} i + K_{r,4} V + l_r)T}) A_c^{-1} B_{c,2} V - x = 0. \end{aligned} \tag{4.120}$$

Under the assumption that the region $\mathcal{R}_r$ contains the undisturbed parameter vector $\overline{p}' = [\overline{x}'^\top, 0, \overline{V}_{\text{in}}]^\top$, we have that

$$H(\overline{x}', 0, \overline{V}_{\text{in}}) = 0, \quad \overline{D} = K_r \overline{x}' + K_{r,4} \overline{V}_{\text{in}} + l_r. \tag{4.121}$$

Assuming also that $\overline{p} = [\overline{x}^\top, i_o, V_{\text{in}}]^\top$ and $\overline{p}'$ are sufficiently close, let

$$\Delta p = \overline{p} - \overline{p}' = \begin{bmatrix} \Delta x \\ \Delta i \\ \Delta V \end{bmatrix} = \begin{bmatrix} \overline{x} - \overline{x}' \\ i_o \\ v_{\text{in}} \end{bmatrix}. \tag{4.122}$$

Then, we can approximate Eq. (4.120) by computing its first order Taylor expansion with respect to $\overline{p}'$, obtaining

$$0 = H(\overline{x}, i_o, V_{\text{in}}) \approx \underbrace{H(\overline{x}', 0, \overline{V}_{\text{in}})}_{=0} + \frac{\partial H}{\partial x}(\overline{x}', 0, \overline{V}_{\text{in}})\Delta x + \frac{\partial H}{\partial i}(\overline{x}', 0, \overline{V}_{\text{in}})\underbrace{\Delta i}_{i_o}$$

$$+ \frac{\partial H}{\partial V}(\overline{x}', 0, \overline{V}_{\text{in}}) \underbrace{\Delta V}_{v_{\text{in}}}, \tag{4.123}$$

where

$$\frac{\partial H}{\partial x}(\overline{x}', 0, \overline{V}_{\text{in}}) = H_x = I - (A + BK_r), \tag{4.124a}$$
$$\frac{\partial H}{\partial i}(\overline{x}', 0, \overline{V}_{\text{in}}) = H_i = (I - e^{A_c T})A_c^{-1}B_{c,1} - BK_{r,3}, \tag{4.124b}$$
$$\frac{\partial H}{\partial V}(\overline{x}', 0, \overline{V}_{\text{in}}) = H_V = e^{A_c T}(e^{-A_c \overline{D} T} - I)A_c^{-1}B_{c,2} - BK_{r,4}. \tag{4.124c}$$

Therefore, we can express

$$\Delta x = -H_x^{-1}(H_i i_o + H_V v_{\text{in}}) \tag{4.125}$$

and it also holds that

$$\|\Delta x\|_P \leq \|H_x^{-1}\|_P(\|H_i\|_P |i_o| + \|H_V\|_P |v_{\text{in}}|). \tag{4.126}$$

Now, recalling the expression of Δd, i.e.,

$$\Delta d = \overline{d} - \overline{D} = K_r \Delta x + K_{r,3} i_o + K_{r,4} v_{\text{in}}, \tag{4.127}$$

we have that

$$\begin{aligned} |\Delta d| &\leq \|K_r\|_P \|\Delta x\|_P + |K_{r,3}||i_o| + |K_{r,4}||v_{\text{in}}| \\ &\leq \|K_r\|_P \left(\|H_x^{-1}\|_P(\|H_i\|_P |i_o| + \|H_V\|_P |v_{\text{in}}|)\right) + |K_{r,3}||i_o| + |K_{r,4}||v_{\text{in}}| \\ &= c_i |i_o| + c_v |v_{\text{in}}|, \end{aligned} \tag{4.128}$$

where

$$c_i = \|K_r\|_P \|H_x^{-1}\|_P \|H_i\|_P + |K_{r,3}|, \tag{4.129a}$$
$$c_v = \|K_r\|_P \|H_x^{-1}\|_P \|H_V\|_P + |K_{r,4}|. \tag{4.129b}$$

Finally, we recall Eq. (4.119) which, under the previous assumptions, can be further approximated as

$$\begin{aligned} \left(1 - \frac{v_{\text{in}}}{\overline{V}_{\text{in}}}\right) e^{\alpha |\Delta d|} \approx \left(1 - \frac{v_{\text{in}}}{\overline{V}_{\text{in}}}\right)(1 + \alpha |\Delta d|) &\approx 1 + \alpha |\Delta d| - \frac{v_{\text{in}}}{\overline{V}_{\text{in}}} \\ &\leq 1 + \alpha(c_i |i_o| + c_v |v_{\text{in}}|) + \frac{1}{\overline{V}_{\text{in}}} |v_{\text{in}}| < 1 + \beta, \end{aligned} \tag{4.130}$$

yielding

$$\alpha c_i |i_o| + \left(\alpha c_v + \frac{1}{\overline{V}_{\text{in}}}\right) |v_{\text{in}}| < \beta. \tag{4.131}$$

Eq. (4.131) defines a conservative polytopic approximation of the true admissible disturbance set, given by Eq. (4.110).

4.5. Simulations and Results

In this section, we validate our Analog Circuital Ex-MPC approach, applied to the case study of Buck converters control, through extensive simulations.

Specifically, we consider two simulation scenarios:

1) Through high-level Monte Carlo simulations, we verify the robust control performance in presence of parametric uncertainty on the nominal value of Buck converter passive components.
2) Through high-fidelity circuit-level simulations, we assess the control performance under realistic operating conditions, including the most relevant circuit non-idealities.

Together with the simulations, we provide a detailed description of the Ex-MPC policy design, its complexity reduction, and the circuital implementation.

4.5.1. Implementation Details

Simulations

Simulations are run on a machine powered by a 13$^{\text{th}}$ Gen Intel Core™ i7 CPU at 1.7 GHz, with 16 GB of RAM.

High-level simulations are performed in MATLAB® and Simulink® (ver. 2023b). Circuit-level simulations are performed in LTSpice®, a high-fidelity circuit simulator software [7].

The QP-MPC optimal control problem (4.104) is formulated with YALMIP [96]. The related Ex-MPC policy (4.107) is computed with the Multi-Parametric Toolbox (MPT) for MATLAB [71]. The non-disjoint optimal merging of regions (Section 4.3.2, p. 126) is carried out using its implementation within the MPT toolbox.

Concerning the closed-loop system in simulations:

- For high-level simulations, the CT nonlinear Buck converter model in Eqs. (4.65), (4.67), and (4.69) is employed, acting as the plant to control. This model includes all relevant nonlinearities, including the trailing-edge PWM and the discontinuous conduction mode (DCM) behavior.

  The plant is directly controlled by the Ex-MPC policy (4.107), i.e., not implemented as a circuit. Thus, in high-level simulations, we neglect the effect of circuit non-idealities. These will be considered instead in circuit-level simulations.

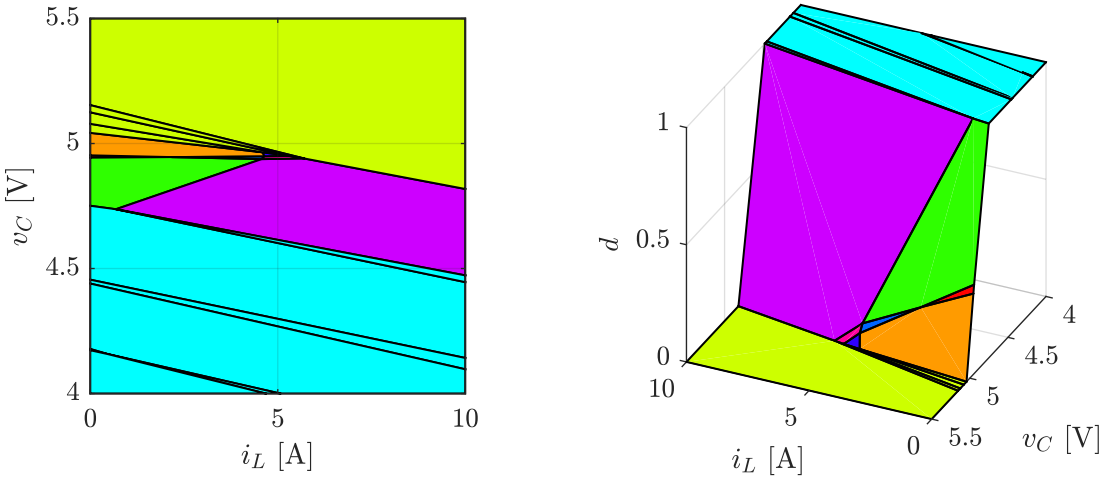
- For circuit-level simulations, the Buck converter is considered in its circuit form.

  The plant is controlled by the analog Ex-MPC circuit, described in Section 4.3.3. Further details on the Ex-MPC circuit implementation, especially concerning components selection, is reported in Section 4.5.3.

Data

Relevant data shared by all simulations is the following:

- General data: $T = 2\,\mu\text{s}$, $f_{\text{sw}} = \frac{1}{T} = 500\,\text{kHz}$.
- Buck converter: $\overline{V}_{\text{in}} = 50$ V (nominal), $V_{\text{in}} \in [25,\ 75]$ V (range); $\overline{V}_o = 5$ V; $I_{o,\max} = 15$ A; $R_L = 3.681\ \Omega$ (nominal), $R_L \in [0.333,\ 7.029]\ \Omega$ (range); $C_o = 250\ \mu$F (nominal), $C_o \in 250\ \mu\text{F} \pm 10\% = [225,\ 275]$ mF (uncertainty); $L = 8.2\ \mu$H (nominal), $L \in 8.2\ \mu\text{H} \pm 20\% = [6.56,\ 9.84]\ \mu$H (uncertainty).



(a) Ex-MPC partition. **(b)** Ex-MPC policy.

Figure 4.9. Ex-MPC policy (4.107) for the Buck converter.

  - Ceramic capacitor: $R_{C_o} = 5$ mΩ (nominal), $R_{C_o} \in 5$ mΩ $\pm$ 50% $=$ $[2.5,\ 7.5]$ mΩ (uncertainty).
  - Electrolytic capacitor: $R_{C_o} = 50$ mΩ (nominal), $R_{C_o} \in 50$ mΩ $\pm$ 50% $=$ $[25,\ 75]$ mΩ (uncertainty).
- Estimators: $g_{i_L} = 0.2$ V A^{-1}, $g_{i_o} = 0.1$.
- MPC: $N_p = 5$, $N_c = 2$, $Q = 10^2$, $R = 10^{-2}$, $R_\Delta = 1$.
- Ex-MPC: $\mathcal{P} = \{p = (i_L, \hat{v}_C, \hat{i}_o, V_{\text{in}}) \in \mathbb{R}^4 : i_L \in [0,\ 80]$ A, $\hat{v}_C \in [0,\ 20]$ V, $\hat{i}_o \in [-5,\ 20]$ A, $V_{\text{in}} = [\overline{V}_{\text{in}} - 35$ V, $\overline{V}_{\text{in}} + 35$ V$]\}$.

The range of possible values for the load resistance R_L is evaluated as follows. First, we consider the maximum output current bearable by the circuit $I_{o,\max}$, and we estimate its value as $I_{o,\max} = 15$ A.

Then, the range of values $[R_{L,\min}, R_{L,\max}]$ can be computed as follows [50]:

$$R_{L,\min} = \frac{\overline{V}_o}{I_{o,\max}} \simeq 0.333\ \Omega, \quad R_{L,\max} = \frac{2L_{\min} f_{\text{sw}}}{1 - \frac{\overline{V}_o}{V_{\text{in,max}}}} \simeq 7.029\ \Omega. \tag{4.132}$$

Finally, according to Proposition 4.2, we choose the nominal value for R_L as its optimal estimate, i.e., $R_L = \hat{R}_L^* = \frac{1}{2}(R_{L,\min} + R_{L,\max}) \simeq 3.681\ \Omega$.

Concerning the capacitor ESR R_{C_o}, we highlight that a very large uncertainty is considered.

4.5.2. Explicit MPC Design and Complexity Reduction

We formulate the QP-MPC problem (4.104) and the related Ex-MPC policy π (4.107) as reported in Section 4.4.3. In the design, we employ the nominal values reported in the data on p. 154. Here, let us consider the case with the ceramic capacitor (i.e., $R_{C_o} = 5$ mΩ).

The resulting Ex-MPC policy is, at the moment, composed of $R = 19$ regions (7 unsaturated, 12 saturated), with 9 unique affine functions (7 unsaturated, 1 saturated to u_{lb}, 1 saturated to u_{ub}).

In the following, we will represent the Ex-MPC policy in two dimensions, considering only the states i_L and $\hat{v}_C$ as parameters, and setting $\hat{i}_o = 0$ A and $V_{\text{in}} = \overline{V}_{\text{in}} = 50$ V. Moreover, to obtain clearer plots, we represent the Ex-MPC policy over the reduced parameter set $\{(i_L, \hat{v}_C) \in \mathbb{R}^2 : i_L \in [0,\ 10]$ A, $\hat{v}_C \in [4,\ 5.5]$ V$\}$.

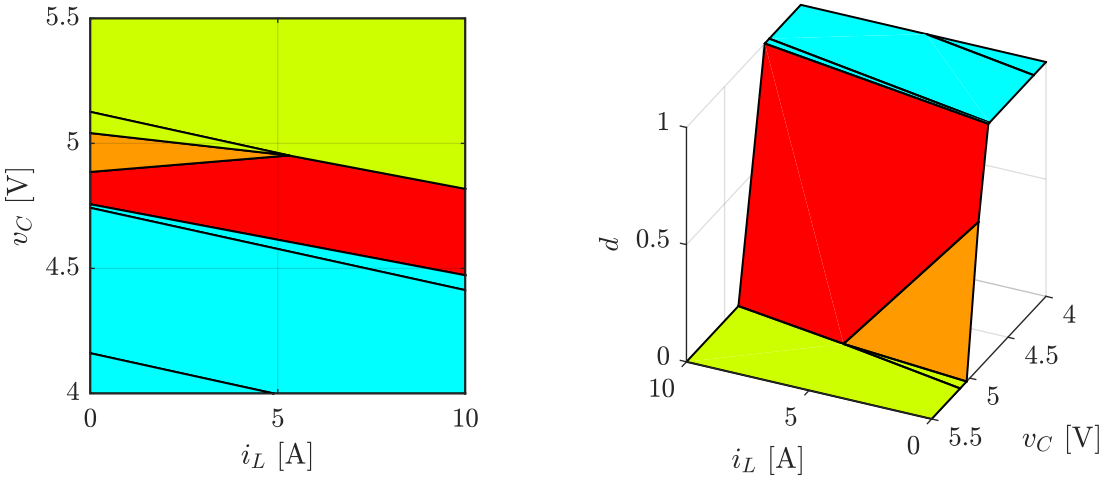
The Ex-MPC policy π (4.107) is reported in Figure 4.9.

We reduce the complexity of π by applying, in sequence, all the four techniques reported in Section 4.3.2.

Move Blocking Strategy

We introduce move blocking in the QP-MPC problem (4.104), with $N_c = 2$ (lower than $N_p = 5$) and selecting the matrix T to match the type in Eq. (4.27a).

The new simplified Ex-MPC policy is reported in Figure 4.10: it is composed of $R = 7$ regions (2 unsaturated, 5 saturated), with 4 unique affine functions (2 unsaturated, 1 saturated to u_{lb}, 1 saturated to u_{ub}). Therefore, with the move blocking strategy, complexity has been reduced by 63%.



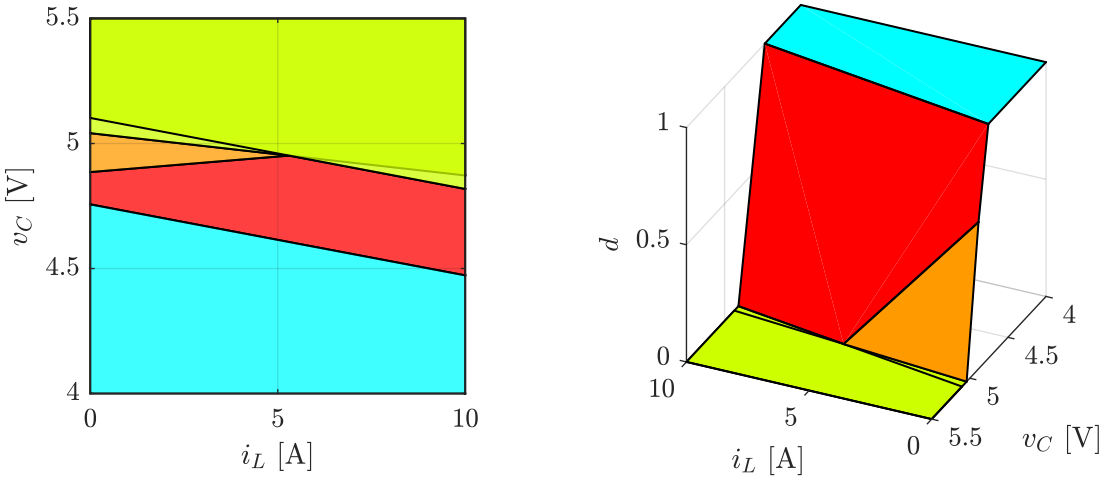
(a) Simplified Ex-MPC partition. **(b)** Simplified Ex-MPC policy.

Figure 4.10. Simplified Ex-MPC policy for the Buck converter, obtained by applying the move blocking strategy. Each color relates to a different affine function.

Non-Disjoint Optimal Merging of Regions

After move blocking, we apply the non-disjoint optimal merging of regions.

The new simplified Ex-MPC policy is reported in Figure 4.11: it is composed of $R = 5$ regions (2 unsaturated, 3 saturated), with 4 unique affine functions (2 unsaturated, 1 saturated to u_{lb}, 1 saturated to u_{ub}). From Figure 4.11a, we can see that a single overlap occurs between the couple of regions saturated to $u_{\text{ub}} = d_{\text{ub}} = 0$ ($\underline{\mathcal{S}}$, ▭). Therefore, with non-disjoint optimal merging, complexity has been further reduced by 29% (overall, 74%).



(a) Simplified Ex-MPC partition.

(b) Simplified Ex-MPC policy.

Figure 4.11. Simplified Ex-MPC policy for the Buck converter, obtained by applying move blocking and non-disjoint optimal merging of regions. Each color relates to a different affine function. In **(a)**, region overlaps are reported with a lighter shading.

Hyperplane Separation of Saturated Regions

After move blocking and non-disjoint optimal merging, we apply hyperplane separation of saturated regions.

To compute the affine separation function $\sigma(p) = a^\top p + b$, with $p = [i_L,\ \hat{v}_C,\ \hat{i}_o,\ V_{\text{in}}]^\top$, we leverage Theorem 4.2. Specifically, by solving the LP feasibility problem (4.41), we obtain the feasible solution

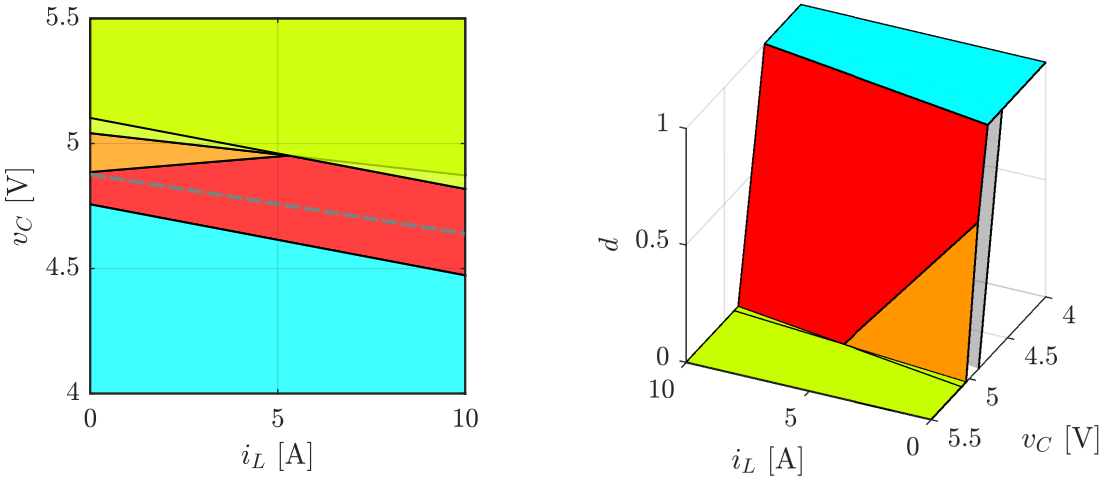
$$\begin{aligned} a^* &= [-0.010059,\ -0.39410,\ 0.010033,\ -0.00050301]^\top, \\ b^* &= 1.9191, \\ \varepsilon^* &= 1.3042 \times 10^{-2} > 0. \end{aligned} \tag{4.133}$$

Therefore, the affine separator exists and is computed as $\sigma(p) = a^{*\top} p + b^*$ (Figure 4.12b).

To better visualize (in two dimensions) the regions separation, we compute the zero-level set of σ, i.e., $\{(i_L, \hat{v}_C) \in \mathbb{R}^2 : \sigma(i_L, \hat{v}_C, 0, \overline{V}_{\text{in}}) = 0\} = \{(i_L, \hat{v}_C) \in \mathbb{R}^2 : \hat{v}_C = -\frac{a_1^*}{a_2^*} i_L - \frac{b^*}{a_2^*}\}$, and report it in Figure 4.12a.

After this simplification step, we are left with only $R = 2$ unsaturated regions, with 2 unique unsaturated affine functions. Therefore, with hyperplane separation of saturated regions, complexity has been further reduced by 60% (overall, 89%).

After this three-fold complexity reduction (i.e., move blocking strategy, non-disjoint optimal merging of regions, and hyperplane separation of saturated regions), the simplified



(a) Simplified Ex-MPC partition and zero-level set of the affine separation function σ (---).

(b) Simplified Ex-MPC policy and affine separation function σ (▭).

Figure 4.12. Simplified Ex-MPC policy (4.134a) for the Buck converter, obtained by applying move blocking, non-disjoint optimal merging, and hyperplane separation of saturated regions. Each color relates to a different affine function: regions saturated to u_{lb} ($\underline{\mathcal{S}}$, ▭), regions saturated to u_{ub} ($\overline{\mathcal{S}}$, ▭). In **(a)**, region overlaps are reported with a lighter shading.

Ex-MPC policy in Figure 4.12 has the following expression:

$$d_k = \begin{cases} \begin{bmatrix} -0.082188 \\ -2.8934 \\ 0.082451 \\ -0.0044955 \end{bmatrix}^\top p_k + 14.988 & \text{if } \begin{bmatrix} -0.013029 & 1.0464 & 0.013172 & -0.0022499 \\ 0.0625 & 0 & 0 & 0 \\ 0 & 0 & 0.25 & 0 \\ 0 & 0 & 0 & 0.058824 \\ 0 & 0 & 0 & -0.33333 \\ 0 & 0 & -1 & 0 \\ -1 & 0 & 0 & 0 \\ -0.029379 & -1.0343 & 0.029473 & -0.0016069 \\ 0.027419 & 0.96527 & -0.027506 & 0.0014997 \end{bmatrix} p_k + \begin{bmatrix} -5 \\ -5 \\ -5 \\ -5 \\ 5 \\ -5 \\ 0 \\ 5 \\ -5 \end{bmatrix} \leq 0, \\ \begin{bmatrix} -0.068097 \\ -4.0252 \\ 0.068205 \\ -0.0020622 \end{bmatrix}^\top p_k + 20.395 & \text{if } \begin{bmatrix} 0 & 0 & 0.25 & 0 \\ 0 & 0 & 0 & 0.058824 \\ 0 & 0 & 0 & -0.33333 \\ 0 & 0 & -1 & 0 \\ -1 & 0 & 0 & 0 \\ -0.017555 & -1.0377 & 0.017583 & -0.00053163 \\ 0.013029 & -1.0464 & -0.013172 & 0.0022499 \\ 0.016694 & 0.98678 & -0.016721 & 0.00050556 \end{bmatrix} p_k + \begin{bmatrix} -5 \\ -5 \\ 5 \\ -5 \\ 0 \\ 5 \\ 5 \\ -5 \end{bmatrix} \leq 0, \\ 0 & \text{if } p_k \notin \mathcal{R}_{\text{unsat}},\ \sigma(p_k) < 0, \\ 1 & \text{if } p_k \notin \mathcal{R}_{\text{unsat}},\ \sigma(p_k) > 0. \end{cases} \tag{4.134a}$$

where $p_k = [i_{L,k},\ \hat{v}_{C,k},\ \hat{i}_{o,k},\ V_{\text{in},k}]^\top$ and

$$\sigma(p_k) = \begin{bmatrix} -0.010059 & -0.39410 & 0.010033 & -0.00050301 \end{bmatrix} p_k + 1.9191. \tag{4.134b}$$

Removal of Trivial Inequalities

Observing Eq. (4.134a) and Figure 4.12a, we see that the 2 unsaturated regions have 1 common facet and are delimited by 6 hyperplanes, 2 of which are given by the set $\mathcal{P}$.

As a consequence, the unsaturated regions of the simplified Ex-MPC policy (4.134a) can be defined by 4 non-trivial inequalities only, of which one is shared.

Applying the same procedure, we can obtain the following results for the case with the electrolytic capacitor (i.e., $R_{C_o} = 50$ mΩ):

- Without complexity reduction, the Ex-MPC policy consists of $R = 21$ regions (6 unsaturated, 15 saturated) with 8 unique affine functions (6 unsaturated, 1 saturated to u_{lb}, 1 saturated to u_{ub}).
- Applying the move blocking strategy, the simplified Ex-MPC policy consists of $R = 7$ regions (2 unsaturated, 5 saturated) with 4 unique affine functions (2 unsaturated, 1 saturated to u_{lb}, 1 saturated to u_{ub}), achieving 67% reduction.
- Applying also the non-disjoint optimal merging of regions, we obtain $R = 5$ regions (2 unsaturated, 3 saturated) with 4 unique affine functions (2 unsaturated, 1 saturated to u_{lb}, 1 saturated to u_{ub}), achieving 76% reduction.
- Applying also the hyperplane separation of saturated regions, we are left with only $R = 2$ unsaturated regions, with 2 unique unsaturated affine functions, achieving 90% reduction.

4.5.3. Circuital Implementation

The analog circuit implementing the Ex-MPC policy (4.134a) is realized according to Section 4.3.3. The estimator $\mathcal{E}$ is designed according to Section 4.4.2.

In the design of both circuits, we employ the nominal values reported in the data on p. 154. For passive circuit components (i.e., resistors and capacitors), we use values taken from the standard E-series.

The overall circuit schematic is reported in Figure 4.13.

The power stage of the Buck converter is implemented through the synchronous Buck controller `LTC7060`. The latter drives the high-side and low-side power MOSFETs of the half-bridge, which are the `BSC050N04LS` and `BSC016N04LS`, respectively, depending on the state of the logic signal provided on the PWM input pin.

The Dickson's charge pump, including C_{pump}-D_{pump}, serves to drive the gate of the n-channel MOSFET `BSC050N04LS`, and it takes as input the voltage $V_{\text{cc}} = 12$ V.

Both the load current estimator (Figure 4.8) and the generalized adders, implementing the Ex-MPC affine functions (Figure 4.5), employ the `AD8031` OP-AMP. The `AD8031` is a rail-to-rail OP-AMP that operates with a single-supply voltage and is characterized by a gain-bandwidth product of 80 MHz.

The single-supply comparator used to implement the Ex-MPC regions and the affine separator (Figure 4.6) is the `LT1721`.

The inductor current sensing is performed through a 10 mΩ shunt resistance R_{sense}, together with the current-sense amplifier `AD8410A`. The latter offers a 2.2 MHz bandwidth (four times higher compared to f_{sw}), and it is configured so that its voltage gain is $A_{\text{sense}} = 20$. Therefore, the overall gain is $g_{i_L} = R_{\text{sense}} A_{\text{sense}} = 0.2\ \text{V A}^{-1}$.

The OP-AMPs, the current-sense amplifier, and the comparators are supplied with a voltage $V_{\text{dd}} = 5$ V. Also, within the generalized adders and comparators, we set the constant voltage $V_0 = V_{\text{dd}} = 5$ V (refer to Section 4.3.3).

Finally, the 4-channel analog MUX is the `ADG5404`, operating with the V_{cc} supply voltage.

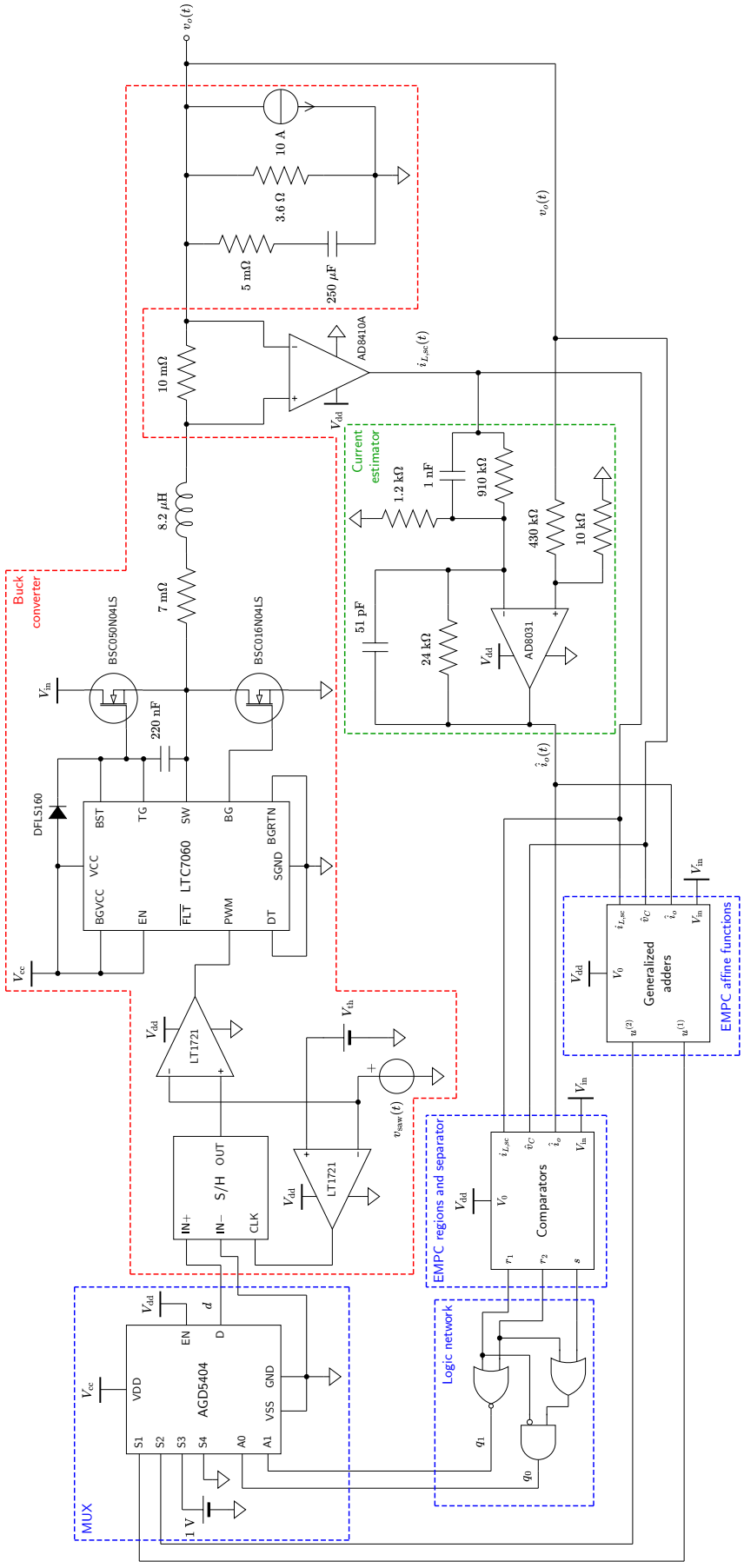


Figure 4.13. Ex-MPC analog circuit controlling the Buck converter: complete schematic.

Effect of Complexity Reduction on Circuital Implementation

After the four-fold complexity reduction carried out in Section 4.5.3, we can comment on how many components are required to implement the Ex-MPC policy (4.134a) as an analog circuit. With reference to Section 4.3.3, we can say the following:

- A four-channel MUX is required, in order to select the control input from each of the 2 unsaturated affine functions, and the two saturated input values, $d_{\text{lb}} = 0$ and $d_{\text{lb}} = 1$.
- To implement the Ex-MPC affine functions by means of generalized adders (Section 4.3.3, p. 135), 2 generalized adders are needed, 1 for each unique affine function.
- To implement the Ex-MPC regions and the affine separator by means of comparators (Section 4.3.3, p. 136), a total of 5 comparators are needed, 1 for each non-trivial inequality and 1 for the affine separator.

Therefore, our analog Ex-MPC circuit can be implemented with only 5 comparators, 2 OP-AMPs, and a four-channel MUX.

Logic Gate Network

In order to implement the network of logic gates in Section 4.3.3, p. 139, whose role is to generate the selection signal q for the MUX, we construct a truth table (Table 4.1), relating the signals $(r_j)_{j=1}^{N_{\text{unsat}}}$, s, and q. Recall that, in the simplified Ex-MPC policy (4.134a), we have only 2 unsaturated regions, with 2 unique unsaturated affine functions, thus $N_{\text{unsat}} = N_\kappa = 2$. Also, in our four-channel MUX, the selection signal is binary encoded, so $q = (q_i)_{i=0}^{1}$ with $\lceil \log_2(N_\kappa + 2) \rceil = \lceil \log_2(4) \rceil = 2$.

s	r_1	r_2	q_1	q_0
0	0	0	1	0
0	0	1	0	1
0	1	0	0	0
0	1	1	-	-
1	0	0	1	1
1	0	1	0	1
1	1	0	0	0
1	1	1	-	-

Table 4.1. Truth table for the MUX selection signal. "-" stands for "don't care".

The resulting logic expression that can be inferred from Table 4.1 is the following:

$$q_0 = \overline{r}_1 \odot (s \oplus r_2), \quad q_1 = \overline{r_1 \oplus r_2}. \tag{4.135}$$

Eq. (4.135) represents the digital logic network to be implemented, and requires only 5 logic gates: 1 AND, 2 OR, and 2 NOT.

Comparison With the Literature

Compared to the analog circuit for solving QPs proposed in [152], our Ex-MPC circuit uses OP-AMPs operating entirely in the linear region, thereby avoiding slew-rate limitations

and resulting in a lower latency. This behavior is confirmed by the simulation results in Section 4.5.5: using the reported commercial components, our Ex-MPC circuit achieves a total latency of ~ 1 µs, representing a substantial improvement (in relative terms) compared to the 6 µs reported in [152]. Furthermore, in our design fewer OP-AMPs are needed: we only require 1 OP-AMP per region, versus the 2 OP-AMPs per inequality in [152].

4.5.4. Robust Performance Assessment

In this section, we assess the robust control performance of the Ex-MPC policy (4.107) in presence of uncertainties in the Buck converter model parameters.

To this end, we carry out an extensive Monte Carlo simulation campaign. To perform these simulations, we have chosen the MATLAB-Simulink environment for its higher computational efficiency, compared to other more accurate circuit simulators.

Within the Buck converter model (4.69), we introduce parametric uncertainty on the nominal value of its passive components, i.e., R_L, L, C_o, and R_{C_o}, according to the data on p. 154. In the Monte Carlo simulations, these uncertain parameters are treated as random variables, with uniform probability distribution within their respective uncertainty intervals, i.e.,

$$\begin{aligned} R_L &\sim U([R_{L,\min},\ R_{L,\max}]), \qquad L \sim U([L_{\min},\ L_{\max}]), \\ C_o &\sim U([C_{o,\min},\ C_{o,\min}]), \qquad R_{C_o} \sim U([R_{C_o,\min},\ R_{C_o,\max}]). \end{aligned} \tag{4.136}$$

For R_{C_o}, we consider both the cases of ceramic and electrolytic capacitor.

A total of 500 random runs are performed.

We study the response of the closed-loop system, i.e., the controlled output voltage $v_o(t)$, in presence of both load and line variations, i.e., injecting the disturbances $i_o(t)$ and $v_{\text{in}}(t)$, respectively.

Figure 4.14 reports the closed-loop system response $v_o(t)$ in presence of a step load disturbance $i_o(t)$, with amplitude equal to $I_{o,\max} - \frac{\overline{V}_o}{R_L}$; such an amplitude is chosen so that the total output current jumps exactly to the worst case value $I_{o,\max}$.

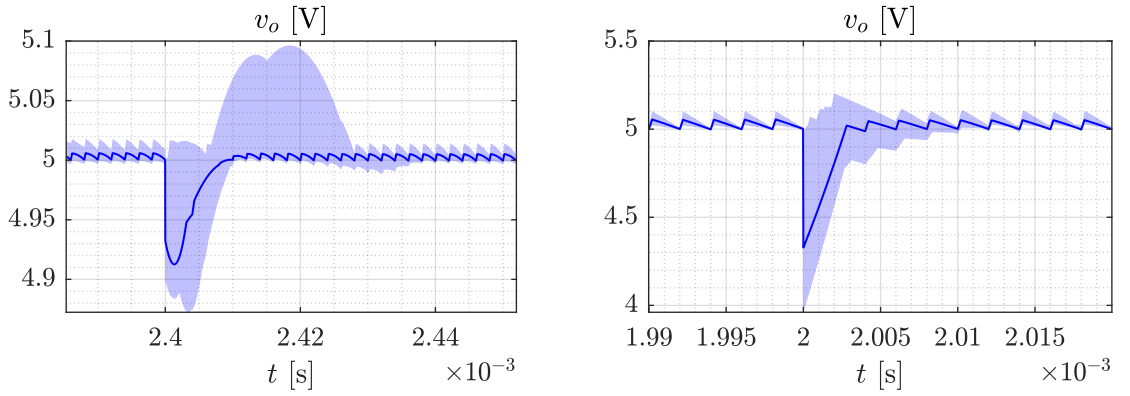
Figure 4.14a reports the ceramic capacitor case, in which the load disturbance is injected at $t = 2.4 \times 10^{-3}$ s; Figure 4.14b reports the electrolytic capacitor case, in which the load disturbance is injected at $t = 2 \times 10^{-3}$ s.

Figure 4.15 reports the closed-loop system response $v_o(t)$ in presence of a step line disturbance $v_{\text{in}}(t)$, with amplitude equal to 10 V; such an amplitude is significantly high compared to the nominal line voltage $\overline{V}_{\text{in}} = 50$ V.

Figure 4.15a reports the ceramic capacitor case, in which the load disturbance is injected at $t = 2.8 \times 10^{-3}$ s; Figure 4.15b reports the electrolytic capacitor case, in which the load disturbance is injected at $t = 2.4 \times 10^{-3}$ s.

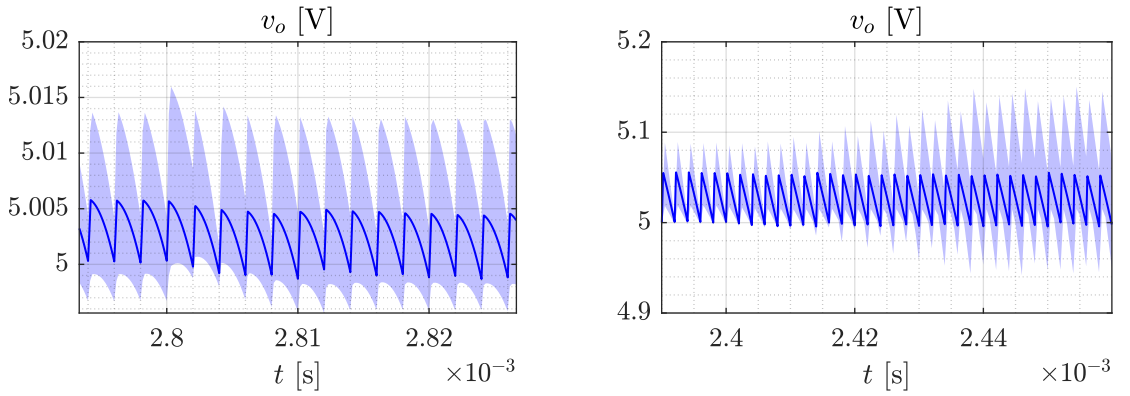
Overall, we observe that the Ex-MPC policy (4.107), controlling the Buck converter model (4.69) with parametric uncertainty, achieves consistently good performance across the entire range of admissible component values, demonstrating the robustness of the adopted methodology. In general, we observe that the parametric uncertainty leads to a slightly erroneous steady state value, i.e., the reference output $\overline{V}_o = 5$ V is not exactly tracked. Still, the error is always lower compared to the ripple voltage, which is acceptable.

Concerning disturbance rejection, a very small settling time is achieved in all simulations.



(a) Ceramic capacitor. **(b)** Electrolytic capacitor.

Figure 4.14. Ex-MPC robust performance in presence of parametric uncertainty: closed-loop system response $v_o(t)$ (nominal ——, uncertainty ▬) to a step load disturbance $i_o(t)$.



(a) Ceramic capacitor. **(b)** Electrolytic capacitor.

Figure 4.15. Ex-MPC robust performance in presence of parametric uncertainty: closed-loop system response $v_o(t)$ (nominal ——, uncertainty ▬) to a step line disturbance $v_{\text{in}}(t)$.

Such a settling time is comparable to just a few cycles of the PWM modulation frequency: on avarage, three cycles are sufficient to restore steady-state cyclostationary operation.

In the case of load disturbance (Figure 4.14), the undershoot/overshoot strongly depends on the disturbance amplitude and the capacitor ESR value R_{C_o}, being significantly larger for the electrolytic capacitor case. Specifically, we see that, when considering the electrolytic capacitor (i.e., higher R_{C_o}), the load disturbance undershoot becomes more pronounced, but the settling time improves.

For what concerns line disturbance (Figure 4.15), we observe, as expected, a small steady-state tracking error, resulting from the linearized model nonlinearity involving V_{in}. However, this effect is always comparable to, or less than, the voltage ripple.

4.5.5. Circuital Simulations

Now, we assess the control performance of the analog Ex-MPC circuit, controlling the Buck converter, by means of circuit-level simulations.

These simulations, carried out in LTSpice, are extremely accurate and allow to demon-

strate the practical feasibility of the proposed approach in real-world conditions. Through these simulations, we can also investigate the impact of circuit non-idealities on control performance.

In order to conduct the simulations, we select commercially-available components to implement the circuit (refer to Section 4.5.3 for more details). For each component, the manufacturer model has been imported into the LTSpice scheme. This enables accurate transistor-level simulation results, which include the non-idealities of the selected components. Among these, we have the finite gain-bandwidth product, the offset currents and voltages, and the slew-rate limitations for OP-AMPs, and the response delay for comparators, MUX, and logic gates.

We conduct a total of 9 simulations, considering 3 values for the resistive load R_L and 3 values for the supply voltage V_{in}. Specifically, $R_L \in \{1\ \Omega,\ 3\ \Omega,\ 5\ \Omega\}$ and $V_{\text{in}} \in \{40\ \text{V},\ 50\ \text{V},\ 60\ \text{V}\}$. The different values for the supply voltage allow to model a constant line disturbance $v_{\text{in}} \in \{-10\ \text{V},\ 0\ \text{V},\ 10\ \text{V}\}$. Instead, concerning load disturbance, a drawn output current pulse $i_o(t)$ is considered, with amplitude 10 A and duration 0.2 ms.

Figures 4.16a-4.16c report the closed-loop system response, i.e., the controlled output voltage $v_o(t)$, in presence of the load and line disturbances described above, within the considered high-fidelity simulation environment.

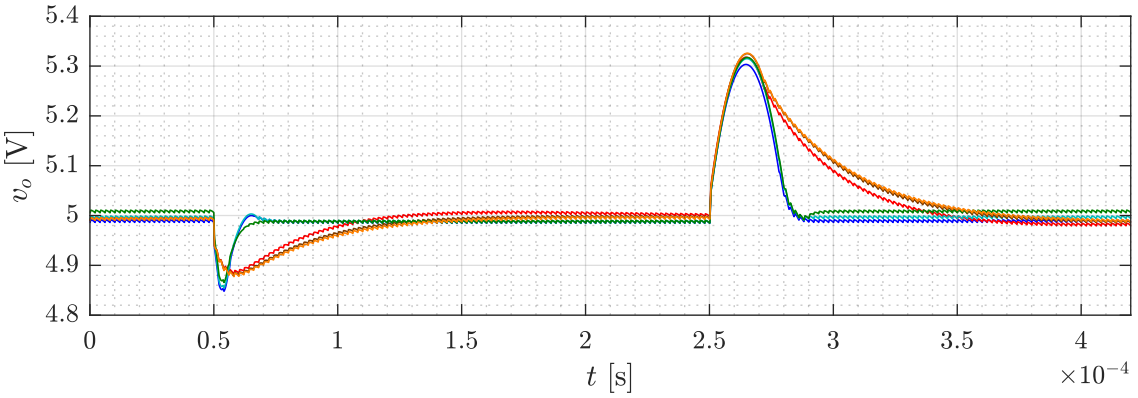
For each simulation, the response is compared to that achieved using standard voltage mode control (VMC). The closed-loop system response with VMC is also obtained through LTSpice simulations. For details on design and implementation of the VMC controller, we refer the reader to [125].

Results indicate that, with the selected components, the impact of circuit non-idealities is negligible, as the circuit-level system responses are mostly overlapping with those obtained through the high-level simulations in Section 4.5.4.

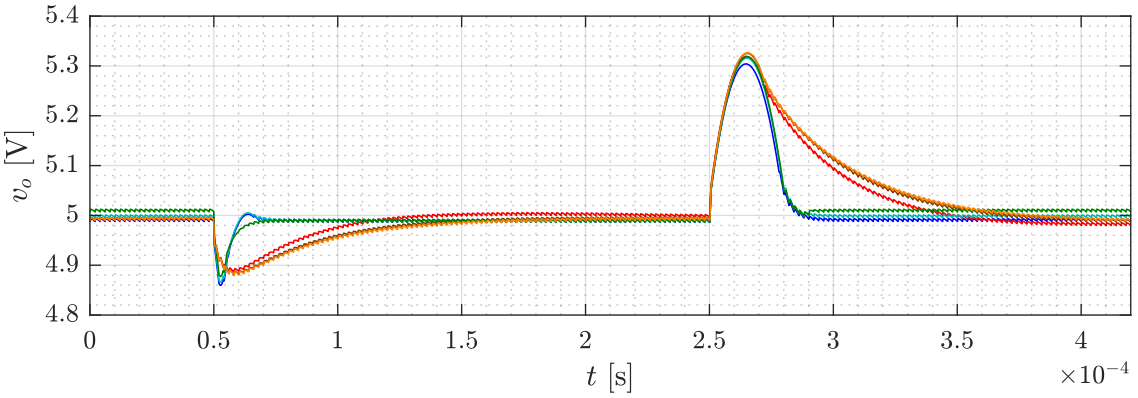
The Ex-MPC circuit exhibits a total propagation delay of 1 µs, which is shorter than the switching period $T = 2$ µs ($f_{\text{sw}} = 500$ kHz). Since T also serves as the discrete time step in the QP-MPC problem (4.104), underlying the Ex-MPC policy (4.107), this confirms the feasibility of the Ex-MPC circuit for high-frequency operation. This result represents a considerable improvement with respect to state-of-the-art analog approaches [152] and existing digital MPC implementations for Buck converters [42, 100, 95, 3].

In terms of load disturbance rejection, the analog Ex-MPC circuit significantly outperforms standard VMC. At $t = 0.5 \times 10^{-4}$ s (i_o rising edge), the undershoot is comparable between Ex-MPC and VMC; instead, the settling time is 2.5 µs for Ex-MPC, significantly lower than the 10 µs of VMC. Similarly, during the recovery phase at $t = 2.5 \times 10^{-4}$ s (i_o falling edge), the settling time is 50 µs for Ex-MPC, while VMC largely exceeds 250 µs. In this latter case, the faster response is accompanied by a larger overshoot, i.e., 7.2% for Ex-MPC and 5.6% for VMC, on average. These results are consistent across the various supply voltages and load configurations.

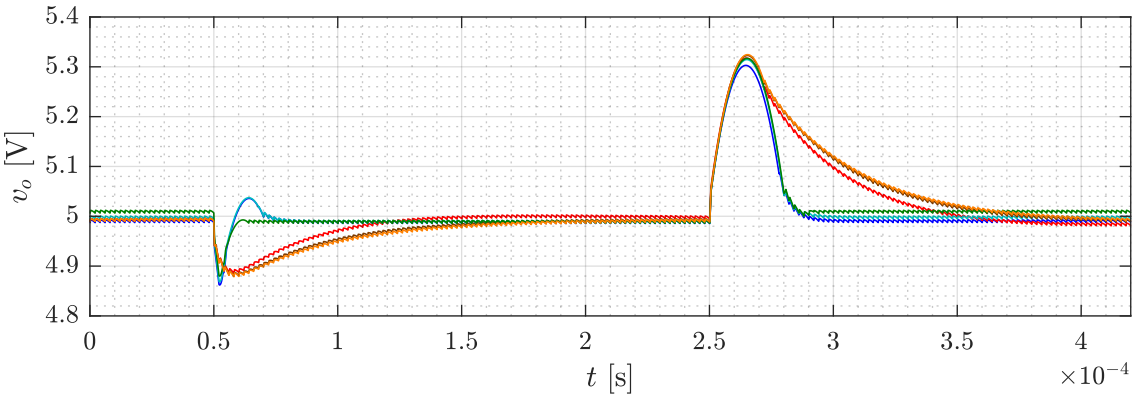
In all simulations, both Ex-MPC and VMC exhibit a small steady-state tracking error, which, in either case, is at maximum around 10 mV. For Ex-MPC, this error arises due to the joint effect of the uncertain load value R_L, the model linearization error, and the approximation introduced by rounding passive components (which define the Ex-MPC policy coefficients) to the nearest standard E-series value. Conversely, VMC benefits from an inherent integral action that theoretically ensures zero steady-state error. The small



(a) Supply voltage $V_{\text{in}} = 40$ V ($v_{\text{in}} = -10$ V).



(b) Supply voltage $V_{\text{in}} = 50$ V ($v_{\text{in}} = 0$ V).



(c) Supply voltage $V_{\text{in}} = 60$ V ($v_{\text{in}} = 10$ V).

Figure 4.16. Ex-MPC control performance, through high-fidelity circuit-level simulations, and comparison with voltage mode control (VMC): closed-loop system response $v_o(t)$ for different resistive loads $R_L \in \{1\ \Omega,\ 3\ \Omega,\ 5\ \Omega\}$, in presence of a constant line disturbance $v_{\text{in}} \in \{-10\text{ V},\ 0\text{ V},\ 10\text{ V}\}$ and a pulse load disturbance $i_o(t)$, with amplitude 10 A, starting at $t = 0.5 \times 10^{-4}$ s and ending at $t = 2.5 \times 10^{-4}$ s (Ex-MPC, $R_L = 1\ \Omega$ ——; Ex-MPC, $R_L = 3\ \Omega$ ——; Ex-MPC, $R_L = 5\ \Omega$ ——; VMC, $R_L = 1\ \Omega$ ——; VMC, $R_L = 3\ \Omega$ ——; VMC, $R_L = 5\ \Omega$ ——).

error observed in simulations can be attributed to the non-ideality of the OP-AMP used in VMC implementation.

4.6. Chapter Summary

In this chapter, we presented a general methodology for designing fully-analog electronic circuits implementing Explicit Model Predictive Control (Ex-MPC) policies. Our approach leverages the piecewise-affine (PWA) structure of Ex-MPC policies to map the control law directly into an analog architecture. Furthermore, we adopt a series of complexity-reduction strategies to minimize the number of Ex-MPC regions and the number of linear inequalities defining them, thus ensuring a cheaper and faster analog implementation.

Our Analog Circuital Ex-MPC approach was applied to the control of Buck converters, achieving effective rejection of line and load disturbances. Also, we established theoretical guarantees on the local robust stability of the closed-loop system, controlled by the analog Ex-MPC circuit, leveraging contraction theory.

Our approach was validated through high-level and circuit-level simulations, to assess robust performance, in presence of parametric uncertainty, and control performance in a real-world conditions, including circuit non-idealities. Simulations confirmed the proficiency of the analog Ex-MPC circuit in delivering the control action, while outperforming conventional strategies in terms of disturbance rejection.

II

Conflicting Objectives in Economic MPC Applications

5

Economic Nonlinear MPC for Conflicting Control Objectives With Constructive Stability Guarantees

5.1. Introduction

In the last two decades, Economic Nonlinear Model Predictive Control (E-NMPC) has emerged as a noteworthy variant of classic Nonlinear MPC (NMPC) for systems where economic performance is of primary concern [135, 48, 9, 134]. Unlike conventional NMPC, which mainly considers regulation or tracking objectives, E-NMPC directly employs an economic criterion as stage cost of the optimal control problem. In this way, E-NMPC is capable of steering the system under control towards its economically optimal equilibrium state, while also ensuring a profitable economic performance during the transient [134]. This capability makes E-NMPC a promising strategy for applications where the key focus is minimizing operational costs or resource consumption.

Alongside its considerable promise, E-NMPC presents two main limitations. First, the optimal control problem only accounts for the economic objective, thus not allowing to include any additional conflicting control task, such as regulation or tracking towards a desired reference. Second, employing an economic criterion as the E-NMPC stage cost makes the latter, in general, non-minimal at the optimal economic equilibrium [48, 134]. As a consequence, the optimal E-NMPC cost value cannot serve anymore as a Lyapunov function, rendering stability guarantees more challenging to ensure than in standard NMPC [104, 48]. Dissipativity-based arguments have been explored to provide stability results [9, 134], yet these approaches require a case-by-case analysis and usually apply only to restricted classes of systems [134].

In this chapter, we address these limitations – namely, the inability to handle conflicting objectives and the non-triviality of stability guarantees – by presenting a novel E-NMPC control problem formulation [34]. Our E-NMPC approach jointly accounts for both economic and tracking objectives, regulating the plant towards the optimal trade-off equilibrium between these two conflicting tasks, thus mediating economic and tracking performance within the same E-NMPC control policy. In addition, we introduce a general constructive

procedure for the design of suitable stabilizing terms for E-NMPC, ensuring its closed-loop stability with minimal impact on the economic performance.

A particularly relevant field of application for our E-NMPC approach is represented by electric vehicles (EVs) energy management. In recent years, the escalating environmental impact of global warming and pollution has become a growing concern for both the public and the scientific community worldwide. In response, the automotive industry has accelerated its transition towards more sustainable alternatives to internal combustion engines. EVs have emerged as a leading solution, offering an effective way to reduce emissions and mitigate the environmental damage caused by conventional transportation methods [33].

However, being EVs supplied with energy that is still largely generated through conventional means, eliminating emissions is not enough to minimize their environmental impact: an important factor lies in optimizing the energy consumption during vehicle operation. From a control perspective, energy optimization can be integrated with autonomous vehicle functionalities. A notable example is represented by Adaptive Cruise Control (ACC), a driver-assistance system designed to control the vehicle speed, while ensuring safe driving conditions with respect to the preceding vehicle [151, 165]. ACC can either maintain a velocity selected by the driver or dynamically adapt it according to the leading vehicle behavior.

Conventional ACC control strategies typically only account for the tracking task, disregarding economic performance. Examples include the constant time gap (CTG) policy [132, 155, 136], whose goal is to maintain a prescribed time-based headway gap between the ego and leading vehicle; such a gap equals the time needed by the ego vehicle to reach the leading one at its current speed. CTG, however, exhibits several drawbacks, among which non-optimality and inability to handle state and input constraints. Other widely-adopted ACC strategies include fuzzy logic control (FLC) [118, 12, 106] and reinforcement learning (RL)-based control [25, 1]. FLC employs a rule-based decision-making approach, bearing the hallmarks of interpretability and fast execution, but lacking prediction and adaptation capability. RL-based control, on the other hand, leverages machine learning to deliver suitable control actions by learning from real driving data. RL, however, requires high amounts of data and extensive training in diverse scenarios, while its lack of interpretability poses a challenge for ensuring safety.

The ACC case study is employed to validate our E-NMPC strategy, considering two different scenarios: velocity regulation only and with the inclusion of a prescribed intervehicular distance, given by the Constant Time Gap (CTG) policy [136]. These tracking objectives are considered in combination with the economic task of minimizing the vehicle energy consumption, which in the case of EVs specifically translates into the minimization of battery usage.

The effectiveness of the ACC based on E-NMPC is thoroughly assessed in simulation, encompassing E-NMPC control performance, trade-off between conflicting objectives, stability, and economic profit compared to standard NMPC. Results demonstrate the proficiency of E-NMPC in delivering the ACC control action, for both tracking and economic tasks. Also, E-NMPC attains a higher economic profit compared to NMPC, while ensuring its closed-loop stability.

5.1.1. Outline

This chapter is organized as follows. Section 5.2 introduces the problem statement. Section 5.3 presents the E-NMPC approach for conflicting control objectives with constructive stability guarantees. Section 5.4 describes the application to Adaptive Cruise Control. Section 5.5 reports simulation results.

5.1.2. Related Works

This chapter is based, in part, on the following works:

- L. Calogero, M. Pagone, C. Novara, and A. Rizzo, "Economic Nonlinear MPC for Conflicting Control Objectives: The Case of Adaptive Cruise Control," in *Proceedings of the IEEE Conference on Decision and Control*, 2025.
- L. Calogero, M. Pagone, L. Zino, C. Novara, and A. Rizzo, "Economic Nonlinear MPC for Conflicting Control Objectives with Constructive Stability Guarantees: A Two-Fold Application Case Study," in *Proceedings of the "Automatica.it" Conference*, 2025.

5.2. Problem Statement

Let us consider a discrete-time (DT) nonlinear dynamical system, i.e.,

$$x_{k+1} = f(x_k, u_k), \quad k \in \mathbb{Z}_{\geq 0}, \tag{5.1}$$

subject to the linear constraints

$$x_k \in \mathcal{X} = \{x \in \mathbb{R}^{n_x} : H_x x \leq h_x\} \subseteq \mathbb{R}^{n_x}, \tag{5.2a}$$
$$u_k \in \mathcal{U} = \{u \in \mathbb{R}^{n_u} : H_u u \leq h_u\} \subseteq \mathbb{R}^{n_u}, \tag{5.2b}$$

at all time instants $k \geq 0$. Let $\mathcal{Z} = \mathcal{X} \times \mathcal{U}$.

We consider the following regularity assumptions for system (5.1):

Assumption 5.1

System (5.1) is controllable [48], the state x_k is available at each time instant $k \geq 0$, and f is C^1-smooth on $\mathcal{Z}$.

Assumption 5.2

System (5.1) admits a manifold $\mathcal{Z}_s$ of admissible equilibrium points (x_s, u_s), i.e.,

$$\mathcal{Z}_s = \{(x_s, u_s) \in \mathcal{Z} : x_s = f(x_s, u_s)\}. \tag{5.3}$$

In the following, we employ the notations $z = [x^\top, u^\top]^\top$ and $z = (x, u)$ interchangeably, to denote a couple of state and input vectors.

Let us now consider a desired reference equilibrium point $(x_r, u_r) = z_r \in \mathcal{Z}_s$, toward which we want to stabilize system (5.1). However, we would also like that, in the transient phase of regulation, the system operates with a *profitable economic performance*. Such two control objectives, i.e., regulation and economic profit, are conflicting with each other.

To quantify these two control tasks, let us define a "performance-metric" stage cost function ℓ as follows:

$$\ell(x,u) = \ell_r(x,u) + \ell_e(x,u), \tag{5.4a}$$

$$\ell_r(x,u) = \|x - x_r\|_Q^2 + \|u - u_r\|_R^2. \tag{5.4b}$$

Here, the stage cost function ℓ encodes the two conflicting control objectives mentioned above: the term ℓ_r is the classic quadratic stage cost, quantifying the regulation control task towards the reference $z_r \in \mathcal{Z}_s$ (Remark 2.5); the term ℓ_e, instead, represents an economic criterion – which, typically, is non-quadratic and non-convex – to be optimized for the system to operate with a profitable economic performance.

Therefore, ℓ is an *economic* stage cost function [134].

Assumption 5.3

The economic criterion ℓ_e (and, thus, the economic stage cost ℓ) is locally Lipschitz continuous on $\mathcal{Z}$.

If we minimize ℓ_r and ℓ_e individually over the equilibrium manifold $\mathcal{Z}_s$, we obtain

$$\min_{(x,u)\in\mathcal{Z}_s} \ell_r(x,u) = 0, \qquad \arg\min_{(x,u)\in\mathcal{Z}_s} \ell_r(x,u) = (x_r,u_r) = z_r, \tag{5.5a}$$

$$\min_{(x,u)\in\mathcal{Z}_s} \ell_e(x,u) = \ell_e(x_e,u_e), \qquad \arg\min_{(x,u)\in\mathcal{Z}_s} \ell_e(x,u) = (x_e,u_e) = z_e. \tag{5.5b}$$

In general, being ℓ_e an economic criterion, there exist some transient points $(x,u) \in \mathcal{Z} \setminus \mathcal{Z}_s$ such that $\ell_e(x,u) < \ell_e(x_e,u_e)$, i.e., ℓ_e is not minimal in z_e [134].

Now, minimizing the composite stage cost ℓ over $\mathcal{Z}_s$, we obtain

$$\min_{(x,u)\in\mathcal{Z}_s} \ell(x,u) = \ell(x_s,u_s), \qquad \arg\min_{(x,u)\in\mathcal{Z}_s} \ell(x,u) = (x_s,u_s) = z_s. \tag{5.6}$$

Here, z_s is a *trade-off equilibrium* between the two conflicting objectives, for which it holds that $\ell_r(x_s,u_s) > \ell_r(x_r,u_r)$ and $\ell_e(x_s,u_s) > \ell_e(x_e,u_e)$. The priority of one objective over the other can be set by properly tuning the weighting parameters of ℓ_r and ℓ_e. Also for ℓ, in general, it holds

$$\ell(x,u) < \ell(x_s,u_s) \;\text{ for some }\; (x,u) \in \mathcal{Z} \setminus \mathcal{Z}_s. \tag{5.7}$$

Given the above setting, our goal is to design an MPC-based optimal control strategy to regulate system (5.1) towards the trade-off equilibrium z_s, while ensuring that:

- in the transient phase of regulation, the optimal trade-off between the two conflicting objectives (i.e., tracking and economic performance) is achieved;
- the stability of the closed-loop system is guaranteed.

Being ℓ an economic stage cost function, we leverage Economic Nonlinear MPC (E-NMPC), as detailed in the next section.

5.3. Economic Model Predictive Control for Conflicting Control Objectives

Economic Nonlinear MPC (E-NMPC) is a noteworthy variant of classic Nonlinear MPC (NMPC), whose purpose is to steer the system under control towards an economically

optimal equilibrium state – for a given economic criterion – while also ensuring a profitable economic performance during the transient phase of regulation. This is achieved by directly employing the economic criterion as stage cost function of the E-NMPC optimal control problem [134].

5.3.1. Economic NMPC Formulation

Economic NMPC provides, at each discrete time instant $k \geq 0$, an optimal control input u_k^*, obtained by solving the following NLP optimal control problem:

$$\min_{\hat{\boldsymbol{x}},\hat{\boldsymbol{u}}} \quad J(\hat{\boldsymbol{x}},\hat{\boldsymbol{u}}) = \sum_{i=0}^{N_p-1} \varphi(\hat{x}_i,\hat{u}_i) + V_f(\hat{x}_{N_p}) \tag{5.8a}$$

$$\text{s.t.} \quad \hat{x}_0 = x_k, \tag{5.8b}$$

$$\hat{x}_{i+1} = f(\hat{x}_i,\hat{u}_i), \quad i = 0,\dots,N_p-1, \tag{5.8c}$$

$$(\hat{x}_i,\hat{u}_i) \in \mathcal{Z}, \quad i = 0,\dots,N_p-1, \tag{5.8d}$$

$$\hat{x}_{N_p} \in \mathcal{X}_f. \tag{5.8e}$$

In the E-NMPC economic cost function (5.8a), φ is the economic stage cost and V_f is a terminal cost. While V_f is present in general E-NMPC formulations (see, e.g., [52]), for our needs we set $V_f(\hat{x}_{N_p}) = 0$. The nonlinear prediction model (5.8b) is given by system (5.1). Eq. (5.8c) reports linear input and state constraints, given by the convex polytopic sets $\mathcal{X}$ and $\mathcal{U}$, according to Eq. (5.2). Eq. (5.8d) enforces a terminal state constraint, defined by the terminal set $\mathcal{X}_f \subseteq \mathbb{R}^{n_x}$.

System (5.1) is controlled via the one-step receding horizon policy, i.e., at each k, only the first optimal predicted input $\hat{u}_{0|k}^*$ is applied to the system.

Under the one-step receding horizon policy (see Section 2.2), the E-NMPC problem (5.8) can be represented by a static state-feedback control policy π, as follows:

$$u_k = \hat{u}_{0|k}^* = \pi(x_k), \tag{5.9}$$

and the closed-loop system (5.1), (5.9) evolves as

$$x_{k+1} = f(x_k,\pi(x_k)), \quad k \geq 0. \tag{5.10}$$

Now, recalling Eq. (5.4), we may think in principle to employ as stage cost φ in Eq. (5.8a) each of the terms $\ell_\star$, $\star \in \{r,e\}$, individually. Then, assuming that the E-NMPC policy (5.9) ensures closed-loop stability, system (5.1) is regulated towards the equilibrium $z_\star$ (see Eq. (5.5)), attaining the corresponding control task with the optimal transient behavior. However, being the two tasks conflicting, $\ell_r \neq \ell_e$ and $z_r \neq z_e$. Therefore, we employ the composite function ℓ in Eq. (5.4) as stage cost φ in Eq. (5.8a), i.e.,

$$\varphi(x,u) = \ell(x,u) = \|x - x_r\|_Q^2 + \|u - u_r\|_R^2 + \ell_e(x,u). \tag{5.11}$$

By employing the above stage cost, system (5.1) is going to be regulated towards the trade-off equilibrium z_s (see Eq. (5.6)), achieving, during the transient, the optimal trade-off between the two conflicting control tasks.

Now, a further challenge is posed by guaranteeing the closed-loop stability of system (5.1), under the control of the E-NMPC policy (5.8), (5.9) with stage cost (5.11).

To this end, in the next section we propose a constructive procedure to design stabilizing terms for the E-NMPC strategy (5.8). Moreover, we will analyze the trade-off between economic profit and stabilization introduced by such terms.

5.3.2. Constructive Stability Guarantees

In the Economic MPC framework, ensuring closed-loop stability is not as straightforward as in the classic MPC setting (Section 2.2.1). This arises because the economic stage cost ℓ (5.4) is not minimal at the equilibrium z_s, as highlighted by Eq. (5.7). This condition on the stage cost is expressed by Assumption 2.10, concerning the closed-loop stability of classic NMPC.

This aspect does not allow to employ the optimal cost value $J^*(x)$ of the E-NMPC problem (5.8) as a Lyapunov function (Theorem 2.2), since J^* may not be monotonically decreasing along the trajectories of the closed-loop system (5.10), even if the latter is stable.

In the following, our goal is to introduce suitable ingredients into the E-NMPC problem (5.8) to enforce closed-loop stability. These stabilizing terms should also be tunable, enabling to set a trade-off between stabilization and the economic profit given by the original non-stabilized problem [34].

We start by stating the following theorem, which establishes the structure of the stabilized E-NMPC problem, by introducing suitable additional terms:

Theorem 5.1 (E-NMPC constructive closed-loop stability)

Consider the E-NMPC optimal control problem in Eq. (5.8) and let Assumptions 5.1-5.3 hold.

a) Let the cost function in Eq. (5.8a) be

$$J(\hat{\boldsymbol{x}}, \hat{\boldsymbol{u}}) = \sum_{i=0}^{N_p-1} \overline{\ell}(\hat{x}_i, \hat{u}_i), \tag{5.12}$$

where $\overline{\ell}$ is an augmented stage cost, defined as

$$\overline{\ell}(x,u) = \ell(x,u) + \alpha(x,u), \quad \alpha(x,u) = a\,\kappa(x - x_s, u - u_s), \tag{5.13}$$

where ℓ is given by Eq. (5.4), $\kappa : \mathcal{Z} \to \mathbb{R}_{\geq 0}$ is a positive-definite function, and $a \in \mathbb{R}_{>0}$.

b) Let the terminal set in Eq. (5.8d) be

$$\mathcal{X}_f = \{x_s\}. \tag{5.14}$$

If the weighting parameter a satisfies

$$a > \frac{\ell(x_s, u_s) - \ell(x,u)}{\kappa(x - x_s, u - u_s)}, \quad \forall\, (x,u) \in \mathcal{Z} \setminus \{(x_s, u_s)\}, \tag{5.15}$$

then x_s is an asymptotically stable equilibrium of the closed-loop system (5.10), with region of attraction $\mathcal{X}_{N_p}$, being $\mathcal{X}_{N_p}$ the set of feasible states (see Definition 2.4).

Proof

By Eq. (5.15), it holds that, $\forall\,(x,u) \in \mathcal{Z} \setminus \{(x_s,u_s)\}$,

$$\ell(x_s,u_s) - \ell(x,u) < \alpha(x,u), \tag{5.16a}$$

$$\overline{\ell}(x,u) = \ell(x,u) + \alpha(x,u) > \ell(x_s,u_s). \tag{5.16b}$$

Also, $\overline{\ell}(x_s,u_s) = \ell(x_s,u_s)$. Therefore, $\min_{(x,u)\in\mathcal{Z}} \overline{\ell}(x,u) = \ell(x_s,u_s)$ and (x_s,u_s) is the unique minimizer of $\overline{\ell}$.

Let us consider the optimal cost value $J^*(x)$ of the E-NMPC problem (5.8), (5.12)-(5.14), as a function of the initial state $x \in \mathcal{X}_{N_p}$. By Eq. (5.16b), it holds that

$$\min_{x\in\mathcal{X}_{N_p}} J^*(x) = N_p\ell(x_s,u_s) = J^*(x_s), \tag{5.17}$$

with feasible state and input sequences

$$(\hat{x}_i,\hat{u}_i) = (x_s,u_s), \;\; i = 0,\ldots,N_p-1, \quad \hat{x}_{N_p} = x_s. \tag{5.18}$$

Then, it holds that $x_s \in \mathcal{X}_{N_p}$ and $u_s = \pi(x_s)$ (5.9).

Now, let us consider the candidate Lyapunov function $V(x) = J^*(x) - J^*(x_s)$, which is positive definite in $\mathcal{X}_{N_p}$ wrt x_s, since $V(x_s) = 0$ and, by Eq. (5.17), $V(x) > 0$, $\forall\, x \in \mathcal{X}_{N_p} \setminus \{x_s\}$.

By letting the terminal set in Eq. (5.8d) be $\mathcal{X}_f = \{x_s\}$ as in Eq. (5.14), we can leverage the stability arguments employed in the proof of Corollary 2.2, for which it holds that

$$V(x^+) \leq V(x) + \ell(x_s,u_s) - \overline{\ell}(x,\pi(x)), \tag{5.19}$$

where $x^+ = f(x,\pi(x))$ is the closed-loop successor state, under the E-NMPC control policy (5.9).

By Eqs. (5.13) and (5.16a), it holds that

$$\begin{aligned} V(x^+) - V(x) &\leq \ell(x_s,u_s) - \ell(x,\pi(x)) - \alpha(x,\pi(x)) \\ &< \alpha(x,\pi(x)) - \alpha(x,\pi(x)) = 0, \end{aligned} \tag{5.20}$$

$\forall\, x \in \mathcal{X}_{N_p} \setminus \{x_s\}$, while the equality holds for $x = x_s$, since $\pi(x_s) = u_s$. Therefore, $V(x^+) - V(x)$ is negative definite in $\mathcal{X}_{N_p}$ wrt x_s.

We then conclude that x_s is an asymptotically stable equilibrium of the closed-loop system (5.10), with region of attraction $\mathcal{X}_{N_p}$.

After having formulated the stabilized E-NMPC problem (5.8), (5.12)-(5.14) through Theorem 5.1, we would like to define a *constructive procedure* to design the additional stabilizing terms, i.e., the weighting parameter a and the positive-definite function κ in Eq. (5.13).

Let us start from the weighting parameter a. In order to guarantee the stability of the closed-loop system (5.10), the value of a must be sufficiently high to satisfy Eq. (5.15).

However, the higher is the value selected for a, the lower will be the economic profit

achieved during the transient phase of regulation. This occurs since the function α in Eq. (5.13) increasingly favors a more stable transient response, at the expense of a higher economic profitability that would otherwise be achieved by the original non-stabilized E-NMPC problem (5.8).

Therefore, in order to attain stabilization with the minimal impact on economic performance, we select an optimal value for a, leveraging the following result:

Result 5.1

The optimal value of the weighting parameter a that minimizes the influence of α on the original stage cost ℓ is given by

$$a^* = \sup_{(x,u)\in\mathcal{Z}\setminus\{(x_s,u_s)\}} \frac{\ell(x_s,u_s)-\ell(x,u)}{\kappa(x-x_s,u-u_s)} + \epsilon, \tag{5.21}$$

where $\epsilon > 0$ is arbitrarily small.

Result 5.1 can be readily interpreted by noticing that a^* is the minimal value satisfying Eq. (5.15).

After having selected the value of a, the next step is to define the shape of the function κ. A key design requirement is that κ must ensure the finiteness of the optimal value a^*. This condition is crucial because, if, for a given κ, Eq. (5.21) admits only the solution $a^* = +\infty$, then the E-NMPC problem cannot be stabilized with that choice of κ.

To this end, we state the following proposition, which establishes a general condition on κ that guarantees the finiteness of a^*:

Proposition 5.1

If the original stage cost ℓ and the function κ are related as follows:

$$\ell(x,u) = \ell(x_s,u_s) + O\big(\kappa(x-x_s,u-u_s)\big) \ \text{ as } \ (x,u)\to(x_s,u_s), \tag{5.22}$$

where $O(\cdot)$ is Landau's big O [87], then a^* in Eq. (5.21) is finite.

Proof

Recalling Eq. (5.21), let

$$\beta(x,u) = \frac{\ell(x_s,u_s)-\ell(x,u)}{\kappa(x-x_s,u-u_s)}.$$

By the extreme value theorem, any continuous function defined on a compact set is bounded and attains its bounds [137]. By Assumption 5.3, β is continuous on $\mathcal{Z}\setminus\{z_s\}$. While the set $\mathcal{Z}$ is compact by Eq 5.2, the punctured set $\mathcal{Z}\setminus\{z_s\}$ is not. Then, let us define

$$\mathcal{Z}' = \mathcal{Z}\setminus \mathrm{int}(\mathcal{B}_\delta(z_s)), \tag{5.23}$$

where $\mathcal{B}_\delta(z_s) = \{z\in\mathcal{Z} : \|z-z_s\|_2 \le \delta\}$.

The set $\mathcal{Z}'$ (5.23) is compact for any $\delta > 0$; then, $\beta(z)$ is bounded on $\mathcal{Z}'$ and $a_1^* = \sup_{z\in\mathcal{Z}'}\beta(z)$ is finite.

Letting $\delta \to 0^+$, we have that

$$a_2^* = \sup_{z \in \mathcal{B}_\delta(z_s) \setminus \{z_s\}} \beta(z) = \limsup_{z \to z_s} \beta(z). \tag{5.24}$$

Eq. (5.22) can be equivalently rewritten as follows [87]:

$$\limsup_{z \to z_s} \frac{|\ell(z) - \ell(z_s)|}{|\kappa(z - z_s)|} = \limsup_{z \to z_s} |\beta(z)| < \infty, \tag{5.25}$$

which means that a_2^* is finite.

Then, we can compute $a^* = \sup_{z \in \mathcal{Z} \setminus \{z_s\}} \beta(z) + \epsilon = \max(a_1^*, a_2^*) + \epsilon$, which is finite.

With Proposition 5.1, we have characterized the shape of κ, ensuring that the optimal value a^* is finite. However, the criterion given by Eq. (5.22) for selecting κ is closely tied to the original stage cost ℓ, making the design of κ problem-specific.

To derive a more general form of κ, that guarantees the finiteness of a^* independently of the original stage cost ℓ, we can leverage the local Lipschitz continuity of ℓ, as ensured by Assumption 5.3, and state the following corollary:

Corollary 5.1

Under Assumption 5.3, if the function κ is constructed as follows:

$$\kappa(x - x_s, u - u_s) = \|x - x_s\|_1 + \|u - u_s\|_1 = \|z - z_s\|_1, \tag{5.26}$$

then a^* in Eq. (5.21) is finite.

Proof

Eq. (5.22) can be equivalently rewritten as follows [87]:

$$\begin{aligned} &\exists\, c > 0,\ \delta > 0 : \|z - z_s\|_2 < \delta \Rightarrow \\ &\Rightarrow |\ell(z) - \ell(z_s)| \leq c\, |\kappa(z - z_s)|, \\ &\quad |\ell(z) - \ell(z_s)| \leq c\, \|z - z_s\|_1. \end{aligned} \tag{5.27}$$

Condition (5.27) is the 1-norm local Lipschitz continuity of ℓ in $z_s \in \mathcal{Z}$, which is ensured by Assumption 5.3.

In summary, in Theorem 5.1 we have designed an additional cost term $\alpha(x, u) = a\, \kappa(x - x_s, u - u_s)$ that, if added to the original stage cost ℓ (5.4) as in Eq. (5.13), guarantees the stability of the closed-loop system (5.10).

The optimal value a^* for the weighting parameter a is given by Eq. (5.21), which is ensured to be finite under the hypotheses of Proposition 5.1.

If the original stage cost ℓ satisfies Assumption 5.3 (i.e., ℓ is locally Lipschitz continuous), Corollary 5.1 provides a suitable and general choice for the function κ.

Remark 5.1

The inclusion of α promotes a stable regulation towards z_s, but comes at the price of reducing the economic profit during the transient. While choosing $a = a^*$ ensures stability with minimal reduction in the economic profit, selecting lower values $0 \leq a < a^*$ further favors the economic performance, at the price of (possibly) not guaranteeing closed-loop stability.

Remark 5.2

While the above stability guarantees have been proven for regulation towards a constant trade-off equilibrium z_s, they can be extended to tracking of slowly-varying piecewise-constant sequences $(z_{s,k})_{k\geq 0}$ (refer to Section 2.2.1, p. 30).

Remark 5.3

In practice, since the functions ℓ and κ are known, the optimal value a^* can be computed by solving Eq. (5.21) as a maximization problem over the set $\mathcal{Z}$.

Solving such a problem online may be quite computationally expensive, especially in the case of tracking piecewise-constant sequences $(z_{s,k})_{k\geq 0}$, where the value of a^* needs to be recomputed for each new equilibrium point. This issue, however, can be easily solved by either of these two measures:

- if the sequence $(z_{s,k})_{k\geq 0}$ is known in advance, we can precompute offline the values of a^*_k for each trade-off equilibrium $z_{s,k}$, $k \geq 0$;
- if $(z_{s,k})_{k\geq 0}$ is not known in advance, we can precompute offline a^* over a sufficiently dense set of trade-off equilibria $z_s \in \mathcal{Z}_s$, interpolate these precomputed values, and provide them online to E-NMPC by means of a suitable scheduling policy.

Comparison With the Literature

The idea of enforcing closed-loop stability by augmenting the original stage cost ℓ with an additional term, that is positive definite wrt the trade-off equilibrium z_s, was also proposed in [134]. Specifically, in [134, Theorem 4], the authors first design the stabilizing cost term as $\alpha(x,u) = \overline{h}(\|(x,u) - (x_s,u_s)\|)$, leveraging strict dissipativity arguments. Then, in [134, Section VI.B], they assess stabilization by setting α as a quadratic function, i.e., $\alpha(x,u) = \|x - x_s\|^2_Q + \|u - u_s\|^2_R$. However, the proposed construction of $\overline{h}$ is non-trivial and strongly depends on the system dynamics (5.1) and the original stage cost ℓ. This may exacerbate the nonlinearity of the augmented stage cost $\overline{\ell}$, thereby worsening the computational burden of the stabilized E-NMPC problem. Moreover, simply taking α as a quadratic cost term may not always ensure stabilization, as shown in Proposition 5.1 and Corollary 5.1.

With our approach, we address these limitations by constructing a stabilizing cost term $\alpha(x,u) = a\,\kappa(x - x_s, u - u_s)$, which, leveraging Corollary 5.1, is independent of the system dynamics and the original stage cost. Stabilization is guaranteed by properly choosing the weighting parameter a, whose optimal value a^* is ensured to be finite, under the hypotheses

of Proposition 5.1, and can be computed offline through the maximization problem in Eq. (5.21).

5.4. Application to Adaptive Cruise Control in Electric Vehicles

The proposed E-NMPC strategy for conflicting control objectives is applied to the case study of Adaptive Cruise Control (ACC) in electric vehicles (EVs). As mentioned in Section 5.1, the ACC is devoted to controlling the vehicle velocity, either to maintain a driver-defined value or to dynamically adjust it according to the leading vehicle behavior. This latter function relies on onboard sensors (e.g., radar and LiDAR) to measure the relative distance and velocity between the ego vehicle and the leading vehicle.

In addition to velocity regulation, the ACC case study is well-suited to include additional conflicting objectives. Among these, we are especially concerned with minimizing the vehicle energy consumption. For EVs, this specifically involves battery consumption.

In the following, we begin by deriving models for the vehicle longitudinal dynamics and the electric battery, which serve as the basis for formulating the two conflicting control objectives: attainment of the reference velocity (tracking) and battery saving (economic), i.e., minimizing battery consumption and maximizing battery recharge. We then tailor the E-NMPC strategy to the ACC case study, addressing these conflicting control tasks.

5.4.1. Longitudinal Vehicle Model

The longitudinal vehicle dynamics can be represented by the following continuous-time (CT) dynamical system:

$$\dot{x}(t) = f_c(x(t), u(t)), \quad t \in \mathbb{R}_{\geq 0}, \tag{5.28}$$

where $x = v$ and $u = \tau$ are the state and input, coinciding with the vehicle longitudinal velocity v and the electric motor (EM) torque τ, respectively.

The system dynamics (5.28) is governed by the following equation:

$$\begin{aligned} \dot{v} = a &= \frac{1}{M}(F - F_{\text{vis}}(v) - F_{\text{roll}}(v)) \\ &= \frac{1}{M}\left(\frac{1}{r_{\text{w}}}\tau - \frac{1}{2}\rho A_{\text{f}} C_{\text{d}} v|v| - C_{\text{r}} M g \,\text{sign}(v)\right), \end{aligned} \tag{5.29}$$

where M is the vehicle mass, g is the gravitational acceleration, r_{w} is the wheel radius, ρ is the air density, A_{f} is the vehicle frontal area, C_{d} and C_{r} are the viscous and rolling friction coefficients, respectively. Let us define $\beta_{\text{d}} = \frac{1}{2}\rho A_{\text{f}} C_{\text{d}}$ and $\beta_{\text{r}} = C_{\text{r}} M g$.

From the above model, the total power P_{tot} requested by the vehicle is given by $P_{\text{tot}} = F_{\text{tot}} v$, where $F_{\text{tot}} = F_{\text{vis}} + F_{\text{roll}} + Ma$ is the total thrust force acting on the vehicle. Such a force and power are provided by the EM, meaning that $P_{\text{tot}} = P$ and $F_{\text{tot}} = F = \frac{1}{r_{\text{w}}}\tau$. Therefore, it holds that

$$P_{\text{tot}} = P = \frac{1}{r_{\text{w}}}\tau v. \tag{5.30}$$

It is worth noticing that system (5.28), (5.29) has an equilibrium manifold $\mathcal{Z}_s$ given by

$$\mathcal{Z}_s = \Big\{(v,\tau) : \tau = r_\mathrm{w}\beta_\mathrm{d} v|v| + r_\mathrm{w}\beta_\mathrm{r}\,\mathrm{sign}(v)\Big\}. \tag{5.31}$$

Notably, $(0,0) \in \mathcal{Z}_s$. Henceforth, let $z = [v,\tau]^\top = (v,\tau)$.

To match the DT formulation in Eq. (5.1), the CT system (5.28) is converted to DT by employing a discretization method of choice (refer to Section 2.2, p. 12) with discrete time step $T_s > 0$, yielding

$$x_{k+1} = f(x_k, u_k), \tag{5.32}$$

where $x_k = x(kT_s)$ and $u_k = u(kT_s)$.

By Remark 2.2, the equilibrium manifold $\mathcal{Z}_s$ (5.31) is the same for both systems (5.28) and (5.32).

5.4.2. Battery Model

Together with the longitudinal dynamics, we also take into account the dynamics of the electric battery. We consider the following battery model [33]:

$$\dot{\zeta} = f_\mathrm{b}(\zeta, P_\mathrm{b}) = -\frac{\eta_{\mathrm{b},1}(P_\mathrm{b})}{Q_\mathrm{nom}}\,\frac{1}{2R_\mathrm{b}^\mathrm{o}(\zeta)}\Big(V_\mathrm{b}^\mathrm{oc}(\zeta) - \sqrt{V_\mathrm{b}^\mathrm{oc}(\zeta)^2 - 4R_\mathrm{b}^\mathrm{o}(\zeta)\,\eta_{\mathrm{b},2}(P_\mathrm{b})\,P_\mathrm{b}}\Big). \tag{5.33}$$

In the battery model (5.33), the state and input are $x = SOC = \zeta$ and $u = P_\mathrm{b}$, respectively, where ζ is the battery state of charge (SOC) and P_b is the power requested to the battery. The SOC is defined as the ratio between the battery charge Q_b and the nominal battery capacity Q_nom, i.e., $SOC = \zeta = \frac{Q_\mathrm{b}}{Q_\mathrm{nom}} \in [0,1]$.

For a detailed derivation of Eq. (5.33), we refer the reader to Section 6.5.1, p. 207.

To a first approximation, the dependence of f_b on ζ can be removed by replacing it with a suitable constant value $\tilde{\zeta}$ [115], yielding

$$\dot{\zeta} \approx f_\mathrm{b}(\tilde{\zeta}, P_\mathrm{b}) = \tilde{f}_\mathrm{b}(P_\mathrm{b}). \tag{5.34}$$

The total power P_tot, provided by the EM, is entirely requested to the battery, meaning that $P_\mathrm{tot} = P_\mathrm{b}$. From this, we can relate $\dot{\zeta}$ to the vehicle state and input as follows:

$$\dot{\zeta} = \tilde{f}_\mathrm{b}(P_\mathrm{b}) = \tilde{f}_\mathrm{b}(P_\mathrm{tot}) = \tilde{f}_\mathrm{b}\left(\frac{1}{r_\mathrm{w}}\tau v\right) = \ell_\mathrm{b}(v,\tau). \tag{5.35}$$

5.4.3. Economic NMPC Formulation

The E-NMPC problem (5.8) for the ACC case study is designed according to its stabilized version (5.8), (5.12)-(5.14), given by Theorem 5.1.

Let us start from the E-NMPC cost function in Eqs. (5.8a) and (5.12). The regulation term ℓ_r matches Eq. (5.4b), i.e.,

$$\ell_r(v,\tau) = \|v - v_r\|_Q^2 + \|\tau - \tau_r\|_R^2. \tag{5.36}$$

where v_r is the ACC reference velocity (either driver-defined or dependent on the leading vehicle) and τ_r is the corresponding reference torque such that $(v_r,\tau_r) = z_r \in \mathcal{Z}_s$.

The economic criterion ℓ_e encodes the objective of battery saving. It accounts for the SOC variation over time $\dot{\zeta}$ and is designed to capture the following two objectives: minimize $\dot{\zeta}$ when $P > 0$ (battery discharge) and maximize $\dot{\zeta}$ when $P < 0$ (battery recharge). The expression of $\dot{\zeta}$ as a function of (v, τ) is given in Eq. (5.35). Then, we construct ℓ_e as follows:

$$\ell_e(v, \tau) = Q_e \operatorname{sign}(\tau v) |\ell_{\mathrm{b}}(v, \tau)|. \tag{5.37}$$

Eq. (5.37) comprises the absolute value of $\dot{\zeta}$, which is going to be minimized or maximized according to the sign of P, since, by Eq. (5.30), $\operatorname{sign}(\tau v) = \operatorname{sign}(P)$; $Q_e \in \mathbb{R}_{>0}$ is the weighting parameter of ℓ_e.

The stabilizing term α is designed according to Section 5.3.2. Specifically, leveraging Corollary 5.1, we define

$$\kappa(v - v_s, \tau - \tau_s) = |v - v_s| + |\tau - \tau_s| \tag{5.38}$$

as in Eq. (5.26), where $z_s = (v_s, \tau_s)$ is the trade-off equilibrium given by Eq. (5.6). Then,

$$\alpha(v, \tau) = a\big(|v - v_s| + |\tau - \tau_s|\big). \tag{5.39}$$

where, to guarantee the closed-loop stability, we must ensure that $a \geq a^*$, where a^* is given by Eq. (5.21).

Finally, the stabilizing stage cost is build according to Eq. (5.13), as follows:

$$\bar{\ell}(v, \tau) = \ell_r(v, \tau) + \ell_e(v, \tau) + \alpha(v, \tau). \tag{5.40}$$

As a last step, the terminal set $\mathcal{X}_f$ in Eq. (5.8d) is set equal to the singleton

$$\mathcal{X}_f = \{v_s\}, \tag{5.41}$$

according to Eq. (5.14).

5.5. Simulations and Results

The E-NMPC strategy for conflicting control objectives is validated in simulation on the case study of Adaptive Cruise Control (ACC), described in Section 5.4. For ACC, two different scenarios are considered:

1) Velocity regulation (Section 5.5.2);
2) Velocity regulation with prescribed intervehicular distance, given by the Constant Time Gap (CTG) policy [136] (Section 5.5.3).

5.5.1. Implementation Details

Simulations

Simulations are performed in MATLAB® (ver. 2023b), on a machine powered by a 13$^{\text{th}}$ Gen Intel Core™ i7 CPU at 1.7 GHz, with 16 GB of RAM.

The stabilized E-NMPC optimal control problem (5.8), (5.12)-(5.14) is formulated with CasADi [8] and solved with the NLP interior-point solver Ipopt [153].

The optimal weight a^* for the stabilizing term α in Eq. (5.12) is computed by solving the maximization problem (5.21) (Remark 5.3) with a global metaheuristic solver based on the Genetic Algorithm (GA) [33] (for more details, we refer the reader to Section 6.4.3, p. 205).

Concerning numerical implementation, a smooth approximation of the 1-norm has been employed [58], i.e.,

$$\|x\|_1 \approx \sum_{i=1}^{n_x} \sqrt{x_i^2 + \delta^2} - n_x\delta, \quad \delta \gtrsim 0. \tag{5.42}$$

Directly using the non-differentiable 1-norm in optimization may cause solution instability, as the solution may rapidly switch between competing minimizers near non-differentiable points. Such a behavior may hinder convergence and degrade solution quality. The smooth approximation (5.42) mitigates this problem, ensuring that the function κ (5.26) given by Corollary 5.1, the economic term ℓ_e (5.37), and the augmented stage cost $\overline{\ell}$ (5.40) are C^1-smooth on $\mathcal{Z}$.

Data

Relevant data shared by all simulations is reported in the following:

- Longitudinal vehicle model: $M = 1.2 \times 10^3$ kg, $r_\mathrm{w} = 0.3$ m, $\beta_\mathrm{d} = 0.4043$ kg m^{-1}, $\beta_\mathrm{r} = 117.72$ N.

5.5.2. ACC for Velocity Regulation

Let $v_{r,k}$ be the ACC reference velocity (either driver-defined or equal to the leading vehicle velocity) and related torque $\tau_{r,k}$, such that $(v_{r,k}, \tau_{r,k}) = z_{r,k} \in \mathcal{Z}_s$, for each $k \geq 0$.

For $v_{r,k}$, two velocity profiles are considered:

1) a step transition between two constant velocity values (Figure 5.1a);
2) the Worldwide Harmonized Light Vehicles Test Cycle–Class 3b (WLTC-3b) [145] (Figure 5.1b).

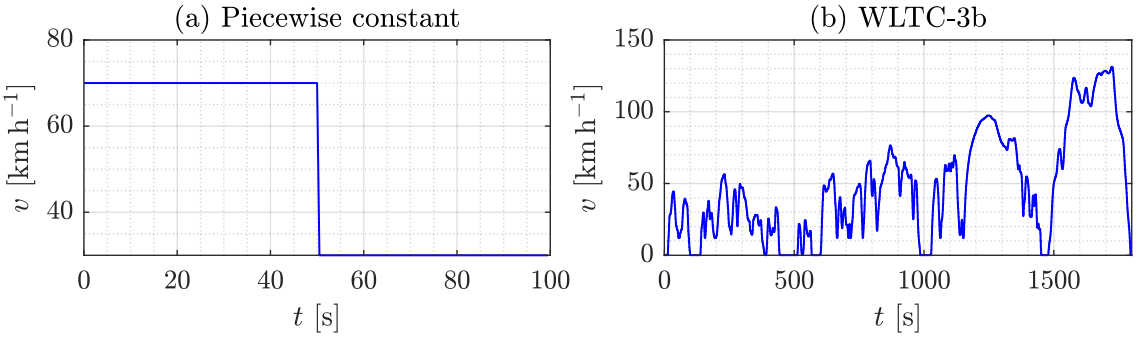


Figure 5.1. ACC velocity profiles: **(a)** Piecewise constant. **(b)** Worldwide Harmonized Light Vehicles Test Cycle–Class 3b (WLTC-3b).

By Remark 5.2, both velocity profiles are shaped as piecewise-constant signals.

With reference to Section 5.4.3, to assess control performance, stability, and trade-off between conflicting objectives, we compare the following three stage costs[1]:

$$\overline{\ell}_1(v,\tau) = \ell_r(v,\tau), \tag{5.43a}$$

$$\overline{\ell}_2(v,\tau) = \ell_r(v,\tau) + \ell_e(v,\tau), \tag{5.43b}$$

$$\overline{\ell}_3(v,\tau) = \ell_r(v,\tau) + \ell_e(v,\tau) + \alpha(v,\tau), \tag{5.43c}$$

where $\overline{\ell}_1$ comprises the velocity regulation term (5.36) only (hence, it is a standard tracking NMPC)[2], $\overline{\ell}_2$ adds the battery saving economic term (5.37) (E-NMPC), and $\overline{\ell}_3$ adds the stabilizing term (5.39) (stabilized E-NMPC).

Data

Relevant data for the current scenario is reported in the following:

- General data:
  - Velocity profile 1 (Figure 5.1a): $T_s = 0.5$ s.
  - Velocity profile 2 (Figure 5.1b): $T_s = 0.25$ s.
- MPC: $N_p = 10$, $R = 10$, $\mathcal{X} = \{x \in \mathbb{R}^{n_x} : x_{\text{lb}} \leq x \leq x_{\text{ub}}\}$, $x_{\text{lb}} = -50$ km h^{-1}, $x_{\text{ub}} = 150$ km h^{-1}, $\mathcal{U} = \{u \in \mathbb{R}^{n_u} : u_{\text{lb}} \leq u \leq u_{\text{ub}}\}$, $u_{\text{lb}} = -500$ N m, $u_{\text{ub}} = 10^3$ N m.
  - Velocity profile 1: $Q = 10$, $Q_e = 10$.
  - Velocity profile 2: $Q = 100$, $Q_e = 100$.

Trade-Off Equilibrium and Stage Cost Minimizer

Let us consider the velocity profile 1 (Figure 5.1a). It consists of two constant velocity values $v_r = (70, 30)$ km h^{-1}; the related torques are, by Eq. (5.31), $\tau_r = (81.17, 43.74)$ N m.

For each value of $z_r = (v_r, \tau_r)$, the corresponding trade-off equilibria $z_s = (v_s, \tau_s)$ of $\overline{\ell}_2$ can be computed through Eq. (5.6), obtaining

$$v_s = (54.99, 27.40) \text{ km h}^{-1}, \quad \tau_s = (63.61, 42.34) \text{ N m}.$$

It is also worth computing, for each z_r, the minimizers z^* of $\overline{\ell}_2$, i.e.,

$$z^* = (v^*, \tau^*) = \arg\min_{z \in \mathcal{Z}} \ \overline{\ell}_2(z), \tag{5.44}$$

obtaining

$$v^* = (150, 117.03) \text{ km h}^{-1}, \quad \tau^* = (-500, -500) \text{ N m}.$$

As expected from Eq. (5.7), $z^* \neq z_s$; specifically, the values of z^* correspond to transient points for which $P < 0$ (i.e., the battery is recharging), meaning that the economic term ℓ_e has a dominant influence over the tracking term ℓ_r.

Now, for each z_r, we compute the optimal weighting parameters a^* (5.21) for the stabilizing term α in $\overline{\ell}_3$, obtaining

$$a^* = (7.43, 3.20),$$

[1] Since the quantities in Eqs. (5.29) and (5.33) differ in magnitude by some orders, suitable rescaling factors have been included in the stage costs (5.43).

[2] For the stage cost $\overline{\ell}_1$ (5.43a), we set $\mathcal{X}_f = \{x_r\}$ in Eq. (5.8d).

which are finite values, in accordance with the results of Corollary 5.1.

Now, computing the minimizers z^* of $\overline{\ell}_3$, i.e.,

$$z^* = (v^*, \tau^*) = \arg\min_{z \in \mathcal{Z}} \ \overline{\ell}_3(z), \tag{5.45}$$

we obtain

$$v^* = (54.99, 27.40)\ \mathrm{km\,h^{-1}}, \quad \tau^* = (63.38, 42.34)\ \mathrm{N\,m}$$

which are equal to z_s, in accordance with the results of Theorem 5.1.

All the above observations hold equivalently also for the velocity profile 2 (Figure 5.1b).

Control Performance and Stage Costs Comparison

Simulation results for both velocity profiles are reported in Figures 5.2 and 5.3, respectively.

Focusing on velocity profile 1, we observe that regulation is achieved by both NMPC (towards z_r) and E-NMPC (towards z_s). With the addition of the stabilizing term, the tracking behavior of E-NMPC is slightly favored during transients (see time intervals $[0, 30]$ s and $[50, 70]$ s) but, importantly, the closed-loop trajectories reach z_s in finite time and evolve under the optimal input τ_s for all subsequent time instants (see the detail in Figure 5.2c).

Economic profit is quantified by the SOC evolution in Figure 5.2d. As expected, E-NMPC attains around 33 % less battery consumption, in relative terms, compared to NMPC. Stabilized E-NMPC consumes slightly more battery than E-NMPC but, interestingly, achieves better battery recharging ($[50, 60]$ s), thanks to its enhanced tracking behavior when the requested power is negative.

Considering now velocity profile 2, tracking is attained by both NMPC (towards $z_{r,k}$) and E-NMPC (towards $z_{s,k}$), with a slight error due to the velocity profile not varying too slowly. From Figure 5.3b, we observe that stabilized E-NMPC better tracks z_s compared to E-NMPC. Finally, from Figure 5.3c, we see that, as expected, E-NMPC attains a higher economic profit than NMPC.

Trade-Off Between Stabilization and Economic Profit

Now, we evaluate the trade-off between stabilization and economic profit. Recalling Remark 5.1, such a trade-off is tuned by the value of the parameter a, which weights the stabilizing term α (5.39).

To this end, we consider a constant velocity $v_r = 70\ \mathrm{km\,h^{-1}}$ (i.e., the first segment of velocity profile 1) and we let a vary in the interval $[0, 10\,a^*]$, where a^* is given by Eq. (5.21).

Results are reported in Figure 5.4.

We can see that, as expected, higher values of a lead to a slight decrease in the economic profit (Figure 5.4b). Also, for $a \geq a^*$, the closed-loop trajectories reach v_s in finite time.

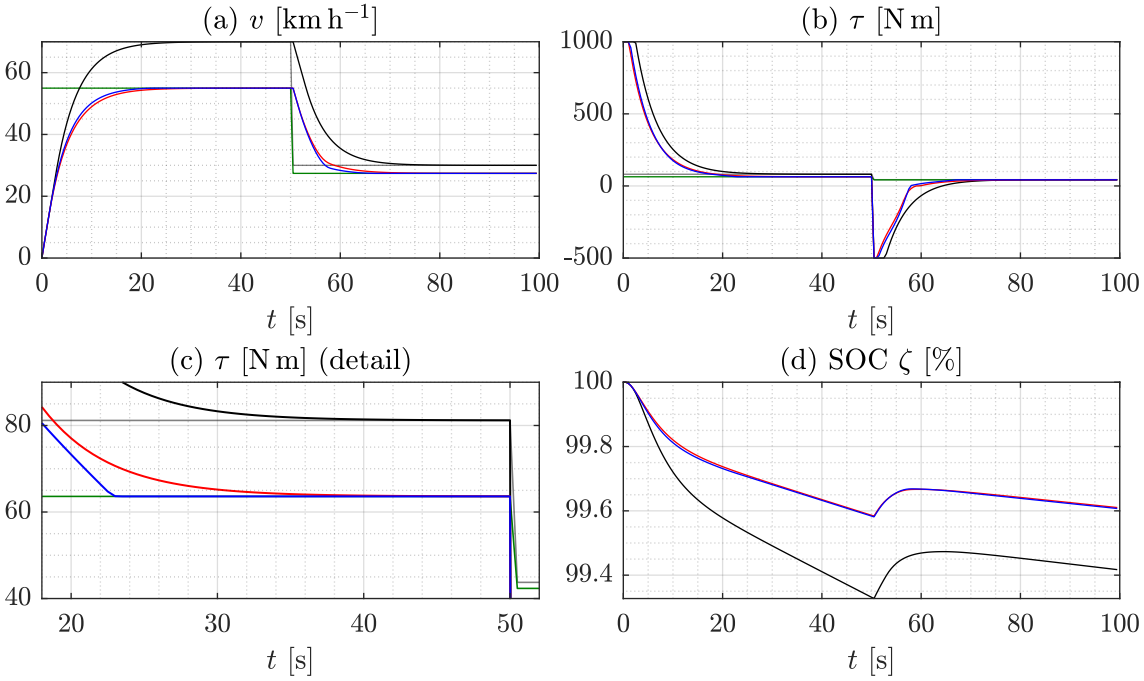


Figure 5.2. ACC closed-loop trajectories with velocity profile 1 (Figure 5.1a): **(a)** Ego vehicle velocity v; **(b)** EM torque τ; **(c)** Detail of the EM torque τ in the time interval $[18, 52]$ s; **(d)** Battery state of charge ζ (economic criterion). NMPC ($\bar{\ell}_1$) ——; E-NMPC ($\bar{\ell}_2$) ——; Stabilized E-NMPC ($\bar{\ell}_3$) ——; reference equilibria $z_r = (v_r, \tau_r)$ ——; trade-off equilibria $z_s = (v_s, \tau_s)$ ——.

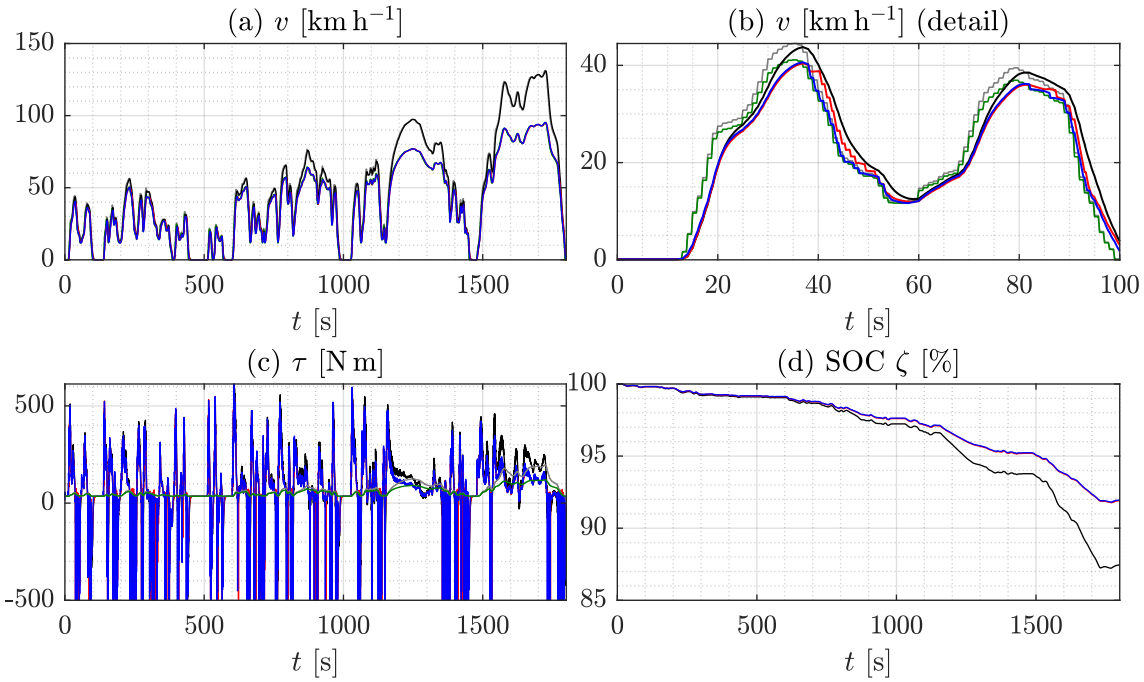


Figure 5.3. ACC closed-loop trajectories with velocity profile 2, (WLTC-3b driving cycle, Figure 5.1b): **(a)** Ego vehicle velocity v; **(b)** Detail of the ego vehicle velocity v in the time interval $[0, 100]$ s; **(c)** EM torque τ; **(d)** Battery state of charge ζ (economic criterion). NMPC ($\bar{\ell}_1$) ——; E-NMPC ($\bar{\ell}_2$) ——; Stabilized E-NMPC ($\bar{\ell}_3$) ——; reference equilibria $z_r = (v_r, \tau_r)$ ——; trade-off equilibria $z_s = (v_s, \tau_s)$ ——.

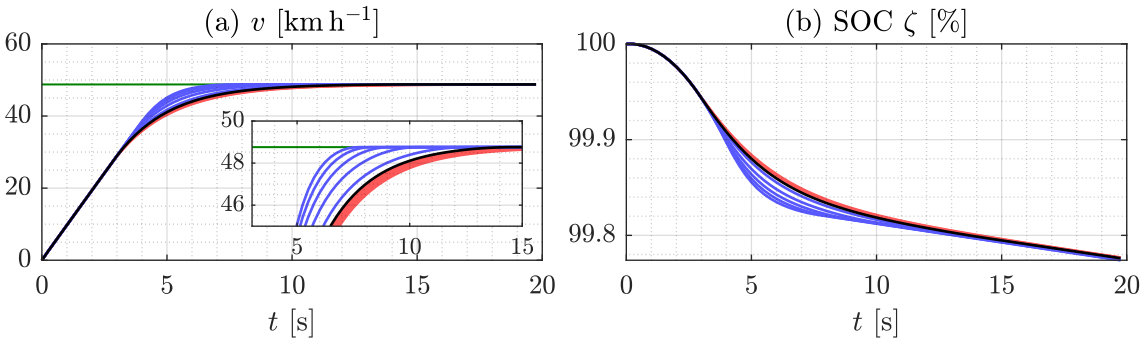


Figure 5.4. E-NMPC trade-off between stabilization and economic profit: **(a)** Ego vehicle velocity v; **(b)** Battery state of charge ζ. $a = a^*$ (—); $a \in [0, a^*)$ (—); $a \in (a^*, 10\,a^*]$ (—); trade-off equilibria v_s (—).

Remark 5.4

It is important to remark that, in this first ACC scenario, E-NMPC attains regulation towards the trade-off velocity v_s, which is typically lower than the original reference velocity v_r, by the presence of the economic cost term $\ell_e(v, \tau)$ (5.37).

However, by tuning the weight Q_e, associated with the economic term ℓ_e (5.37), the trade-off velocity v_s will get closer to v_r, until reaching equality when the velocity regulation term ℓ_r (5.36) dominates over ℓ_e, i.e., $(v_r, \tau_r) = \arg\min_{(v,\tau)\in\mathcal{Z}_s} \ell_r(v, \tau) + \ell_e(v, \tau)$ by Eq. (5.6).

Alternatively, as we will see in the next ACC scenario, by adding a further regulation task on the intervehicular distance between ego and leading vehicle, the equilibrium manifold $\mathcal{Z}_s$ reduces to a singleton, yielding in any case $v_r = v_s$.

5.5.3. ACC for Velocity Regulation With Constant Time Gap

Now, we consider the scenario where the ACC is required not only to regulate the ego vehicle velocity, but also to maintain a prescribed intervehicular distance from the leading vehicle.

Let (p, v) and (p_l, v_l) denote the longitudinal position and velocity of the ego and leading vehicle, respectively; let $d = p_l - p$ denote the intervehicular distance. We require the ACC to regulate d towards a prescribed distance d_r and v towards the leading vehicle velocity $v_l = v_r$. To this end, we augment system (5.28), (5.29) with the new state d, i.e.,

$$\dot{d} = v_l - v = v_r - v, \tag{5.46a}$$

$$\dot{v} = \frac{1}{M}\left(\frac{1}{r_\mathrm{w}}\tau - \beta_\mathrm{d} v|v| - \beta_\mathrm{r}\,\mathrm{sign}(v)\right). \tag{5.46b}$$

By Eq. (5.46a), v_r acts as an exogenous input of the augmented system (5.46), that is measured at each time instant k. By Remark 5.2, we consider v_r as a piecewise constant signal.

Henceforth, let $z = [d, v, \tau]^\top = (d, v, \tau)$. The equilibrium manifold $\mathcal{Z}_s$ of the augmented system (5.46) is given by

$$\mathcal{Z}_s = \{(d, v, \tau) : d \in \mathbb{R},\ v = v_r,\ \tau = r_\mathrm{w}\beta_\mathrm{d} v_r|v_r| + r_\mathrm{w}\beta_\mathrm{r}\,\mathrm{sign}(v_r) = \tau_r\}. \tag{5.47}$$

Since (v_r, τ_r) is the only equilibrium couple, it follows that the reference and trade-off equilibrium coincide, i.e., $z_r = z_s = (d_r, v_r, \tau_r)$.

Constant Time Gap Policy

The Constant Time Gap (CTG) policy [136] provides the value for the reference intervehicular distance d_r. Specifically,

$$d_r = d_o + T_\mathrm{g} v_r, \tag{5.48}$$

where d_o is the standstill distance and T_g is the time headway.

Data

Relevant data for the current scenario is reported in the following:

- General data: $T_s = 0.5$ s.
- MPC: $N_p = 10$, $Q = 1$, $R = 10$, $Q_e = 50$, $x_\mathrm{lb} = [0\ \mathrm{m}, -50\ \mathrm{km\,h^{-1}}]^\top$, $x_\mathrm{ub} = [+\infty, 150\ \mathrm{km\,h^{-1}}]^\top$, $u_\mathrm{lb} = -10^3$ N m, $u_\mathrm{ub} = 10^3$ N m.
- CTG policy: $d_o = 1$ m, $T_\mathrm{g} = 2$ s.

Control Performance

Let us consider a piecewise constant velocity profile with two segments $v_r = (50, 10)$ km h^{-1} (Figure 5.5); the corresponding reference torques are $\tau_r = (58.71, 36.25)$ N m.

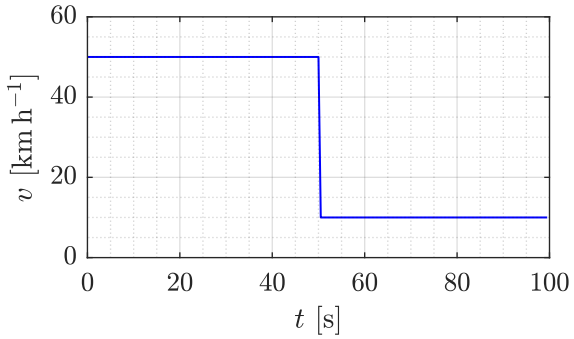


Figure 5.5. ACC with CTG policy velocity profile.

Through the CTG policy (5.48), we obtain $d_r = (28.78, 6.56)$ m.
Also, the optimal stabilizing weights are then equal to

$$a^* = (64.76, 54.90). \tag{5.49}$$

Simulation results are reported in Figure 5.6, comparing NMPC (5.43a) and stabilized E-NMPC (5.43c).

Regulation is achieved by both strategies towards the common reference/trade-off equilibia $z_r = z_s$. Despite being the latter equilibria coincident, E-NMPC is able to still deliver a higher economic performance compared to NMPC (i.e., 7.9 % less battery consumption, in relative terms).

Moreover, thanks to the stabilizing term (5.39), E-NMPC closed-loop trajectories manage to reach z_s in advance compared to NMPC, as noticeable from the detail in Figure 5.6a.

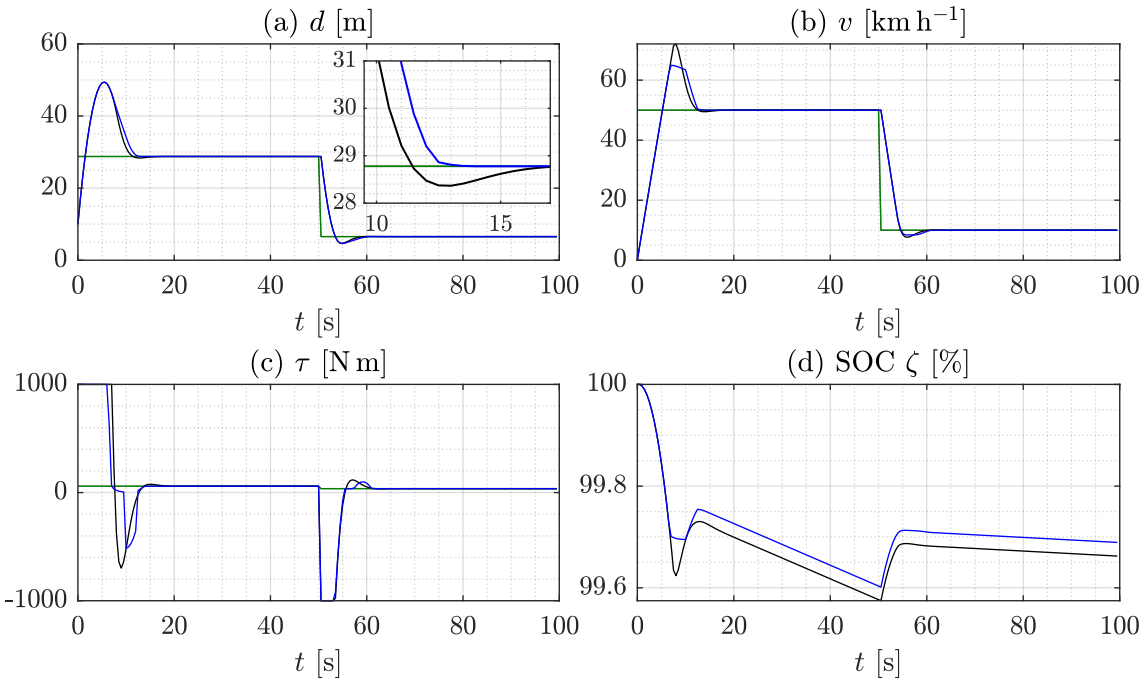


Figure 5.6. ACC with CTG policy closed-loop trajectories: **(a)** Intervehicular distance d; **(b)** Ego vehicle velocity v; **(c)** EM torque τ; **(d)** Battery state of charge ζ (economic criterion). NMPC ($\bar{\ell}_1$) ——; Stabilized E-NMPC ($\bar{\ell}_3$) ——; reference/trade-off equilibria $z_r = z_s = (d_r, v_r, \tau_r)$ ——.

5.6. Chapter Summary

In this chapter, we presented a novel formulation of Economic Nonlinear MPC (E-NMPC) that integrates economic criteria with additional conflicting control objectives, such as tracking. The proposed E-NMPC approach regulates the plant towards the optimal trade-off equilibrium state, mediating the given conflicting objectives. Additionally, we proposed a general constructive procedure to design stabilizing terms for E-NMPC, ensuring its closed-loop stability with minimal impact on the economic performance.

We applied our methodology to the case study of autonomous electric vehicles (EVs), with specific interest to Adaptive Cruise Control (ACC). The E-NMPC strategy is tasked to attain the velocity regulation task while optimizing an economic criterion that consists in minimizing the vehicle energy consumption. The ACC based on E-NMPC has been validated with an extensive simulation campaign, demonstrating its proficient control action and effectiveness in mediating the conflicting tasks, while ensuring closed-loop stability.

6

Neural Adaptive MPC With Online Metaheuristic Tuning for Power Management in Fuel Cell Hybrid Electric Vehicles

6.1. Introduction

Hbrid electric vehicles (HEVs) have established as a new paradigm in transportation, driven by the rapid shift of automotive industry towards sustainable alternatives to internal combustion engines. In recent years, fuel cell hybrid electric vehicles (FCHEVs), which integrate hydrogen fuel cells (FCs), electric batteries, and other supplementary energy sources (e.g., supercapacitors [76]), have gained prominence, thanks to their enhanced autonomy and capability to withstand high power demands [144, 76, 130]. As a response to this growing interest in hybrid transportation, novel and efficient strategies for energy management are becoming essential to enable such a new transportation paradigm.

To fulfill this requirement, we borrow from the established framework of optimal control to address the challenges associated with power management. In this context, we introduce a novel control strategy, called *Neural Adaptive Model Predictive Control* (NA-MPC) [33].

The NA-MPC strategy is formulated starting from a general Linear-Quadratic MPC (QP-MPC) optimal control problem, which typically represents an approximated Nonlinear MPC (NMPC) problem, with nonlinear prediction model and linear constraints (Section 2.3.3). NA-MPC augments such a QP-MPC problem with three key additional features.

First, to achieve multiple, concurrent, and potentially conflicting control objectives at once, the QP-MPC weights, i.e., the elements of the weighting matrices within the QP-MPC cost function, are adapted in real time by means of an online metaheuristic tuning strategy, which minimizes a set of performance-metric functions, each one encoding a different control objective.

Second, to ensure real-time feasibility and low computational demand, the QP-MPC control policy is replaced by a neural emulation of it, yielding a neural MPC controller.

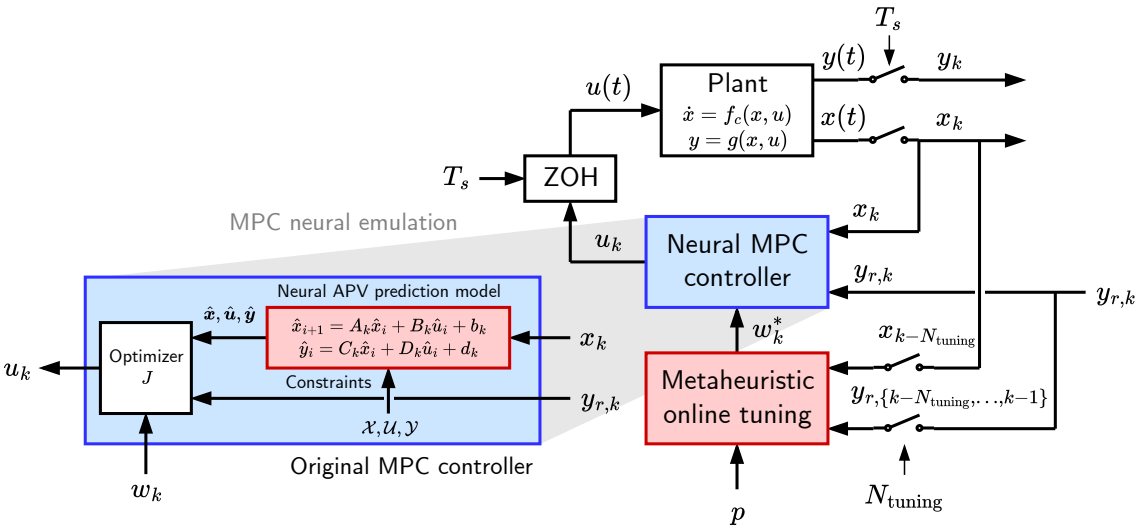


Figure 6.1. Neural Adaptive MPC (NA-MPC) control system scheme.

Third, for the sake of applicability in those cases where an accurate white-box, first-principles model of the plant to control is not readily available, a neural black-box model of the plant is considered, employing it to derive the prediction model for the QP-MPC problem in affine parameter-varying (APV) form. Such a neural plant model is identified from noise-corrupted input-output measurements taken from the plant itself.

A depiction of the NA-MPC control system scheme is reported in Figure 6.1.

In this chapter, we tailor the NA-MPC framework for the task of power management, consisting of finding the optimal power allocation among multiple energy sources driving a shared actuator. While our immediate focus is on FCHEVs (Figure 6.2), the proposed approach can be extended to a broad range of power management scenarios, such as alternative hybrid powertrain configurations, microgrids with energy storage systems, HVAC systems, and others. Moreover, the general formulation of NA-MPC makes it potentially applicable to a variety of control applications beyond power management.

In the context of FCHEV power management, the QP-MPC problem is tasked to find the optimal allocation between battery power and FC power, while predicting the consumption of supplies over time, i.e., the battery state of charge (SOC) and the hydrogen mass. As said, the QP-MPC prediction model is obtained from a neural black-box model of the FCHEV power sources and distribution system. Such a model is identified from noise-corrupted input-output measurements, collected from a high-fidelity white-box model of the FCHEV power system.

As mentioned above, NA-MPC is capable to optimally achieve multiple, conflicting control objectives at once, by means of an online metaheuristic tuning strategy, adapting the QP-MPC weights in real time. For the specific case of FCHEV power management, these control objectives are accurate tracking of the power requested by the vehicle, minimization of supplies consumption, and distinct consumption priority for each supply. In more detail, at fixed time intervals, the tuning strategy simulates back the closed-loop QP-MPC control action over the previous time interval for a given set of QP-MPC weights. These weight are then updated through a metaheuristic optimization policy, which finds the optimal weights minimizing the set of performance-metric functions, each associated to a different

control objective. The optimal weights are then employed during the next time interval, at the end of which a new tuning is initiated.

Concerning instead the neural emulation of the QP-MPC control policy, the neural MPC controller employed by NA-MPC is realized with a feedforward neural network (FNN), which takes as inputs all the parameters of the original QP-MPC problem (including the QP-MPC weights) and provides as output the optimal control input. In this regard, we provide theoretical guarantees on the smoothness of the QP-MPC policy, so that the universal approximation theorem of FNNs [72] holds for the QP-MPC neural emulation. Unlike solving the original QP-MPC problem, evaluating its neural approximation requires near-zero computation time, ensuring the real-time feasibility of the NA-MPC strategy, even in case of limited computational resources. Other studies have addressed MPC emulation using neural networks, see, e.g., [43, 84, 40]. However, most of the proposed approaches consider only the system states as inputs of the neural network, disregarding the MPC weights, which remain static parameters within the optimal control problem. Moreover, they do not provide formal guarantees on the neural network ability to accurately emulate the MPC policy.

Thanks to its capabilities, our NA-MPC strategy represents an advancement in the state of the art on power management strategies and, particularly, within the domain of HEVs. At present, several investigations have delved into HEV energy management, delineating two main approaches: rule-based control and optimization-based control [76]. For such two approaches, the most prominent strategies are Fuzzy-Logic Control (FLC) [69, 70, 51] and Equivalent Consumption Minimization Strategy (ECMS) [114, 163, 162], respectively. FLC classifies power and supply values into qualitative ranges; then, power allocation is performed by means of logic implications on such qualitative values. ECMS, instead, seeks the optimal power values by minimizing, at each time instant, the instantaneous supply rates. FLC bears the hallmarks of simplicity, interpretability, and fast execution, while ECMS has the advantage of providing a locally optimal power allocation. Nevertheless, both FLC and ECMS have significant shortcomings with respect to our NA-MPC strategy: FLC and ECMS do not perform any prediction, thus they cannot handle constraints on the supply values; due to their formulation, they are inherently unable to perform an effective trade-off between tracking accuracy and supply saving; also, so far, only few and non-optimal adaptive strategies have been developed for FLC and ECMS.

Within the optimization-based approaches for power management, also MPC has been investigated by several works. Most of these studies, however, focus on conventional HEVs, thus not including FCs as a power source [21, 39]. While some works have proposed MPC strategies that augment the power management task with additional features (e.g., vehicle speed prediction [131], vehicle mass online estimation [159], etc.), all of them mainly focus only on requested power tracking and charge sustainment, not including additional control objectives. Also, in most cases, the MPC problem formulation is merely an extension of the ECMS policy over a certain time horizon. Moreover, little focus is given to the MPC computational demand to assess real-time feasibility [123].

The effectiveness of our NA-MPC strategy is validated through an extensive simulation campaign. First, we assess the capability of NA-MPC to effectively achieve conflicting control objectives concurrently. Second, the NA-MPC online metaheuristic tuning strategy is compared with an "ideal" tuning policy that assumes to know the whole power request

in advance. Third, NA-MPC is compared with the most prominent state-of-the-art power management strategies, i.e., FLC [70, 51] and ECMS with online adaptation [114]. Simulations demonstrate the capability of NA-MPC in successfully attaining all the given control objectives. The NA-MPC online tuning strategy proves its effectiveness by achieving very similar results with respect to the ideal tuning policy. Also, NA-MPC outperforms both FLC and ECMS, consistently achieving the given control task with better performance.

6.1.1. Outline

This chapter is organized as follows. Section 6.2 introduces the MPC problem formulation. Section 6.3 details the first step to develop the NA-MPC strategy. Specifically, we construct the neural black-box model of the plant to control, and present the QP-MPC neural emulation using feedforward neural networks, supported by universal approximation guarantees. Section 6.4 presents the online metaheuristic tuning strategy, to adapt in real time the weights of the neural MPC controller. Section 6.5 describes the tailoring of the NA-MPC strategy for the case study of power management in FCHEVs. In particular, we outline the construction of a high-fidelity white-box model of the FCHEV power sources and distribution system. Section 6.6 validates the NA-MPC strategy through extensive simulations and comparisons with state-of-the-art techniques.

6.1.2. Related Works

This chapter is based, in part, on the following works:

- L. Calogero, M. Pagone, F. Cianflone, E. Gandino, C. Karam, and A. Rizzo, "Neural Adaptive MPC With Online Metaheuristic Tuning for Power Management in Fuel Cell Hybrid Electric Vehicles," *IEEE Transactions on Automation Science and Engineering*, vol. 22, pp. 11540–11553, Jan. 2025.
- L. Calogero, M. Pagone, F. Cianflone, E. Gandino, C. Karam, and A. Rizzo, "Optimal Tuning and Neural Emulation of MPC for Power Management in Fuel Cell Hybrid Electric Vehicles," in *Proceedings of the "Automatica.it" Conference*, 2023.

6.2. MPC Problem Formulation

Let us consider a continuous-time (CT) nonlinear dynamical system, i.e.,

$$\dot{x}(t) = f_c(x(t), u(t)), \quad x(0) = x_0, \tag{6.1a}$$

$$y(t) = g(x(t), u(t)), \quad t \in \mathbb{R}_{\geq 0}, \tag{6.1b}$$

subject to the linear constraints

$$x_k \in \mathcal{X} = \{x \in \mathbb{R}^{n_x} : H_x x \leq h_x\} \subseteq \mathbb{R}^{n_x}, \tag{6.2a}$$

$$u_k \in \mathcal{U} = \{u \in \mathbb{R}^{n_u} : H_u u \leq h_u\} \subseteq \mathbb{R}^{n_u}, \tag{6.2b}$$

$$y_k \in \mathcal{Y} = \{y \in \mathbb{R}^{n_y} : H_y y \leq h_y\} \subseteq \mathbb{R}^{n_y}, \tag{6.2c}$$

at all time instants $k \geq 0$. With respect to Eq. (2.11), we have introduced the nonlinear output equation (6.1b).

We consider the following regularity assumptions for system (6.1):

Assumption 6.1

a) The state dynamics function $f_c(\cdot, u)$ is Lipschitz continuous on $\mathcal{X}$ and $f_c(x, \cdot)$ is continuous on $\mathcal{U}$.

b) The output function g is continuous on $\mathcal{X} \times \mathcal{U}$.

Assumption 6.2

The input signal $\boldsymbol{u} = (u(t))_{t\in[0,+\infty)}$ belongs to the space of piecewise continuous signals $\mathscr{U}$

Assumption 6.3

The state $x(t)$ of system (6.1) is available at each time instant $t \geq 0$.

Remark 6.1

By Assumptions 6.1a and 6.2, the Picard-Lindelöf theorem guarantees the global existence of a unique solution $(x(t), y(t))$ of system (6.1) in the time interval $[0, +\infty)$, starting from (x_0, y_0), where $y_0 = g(x_0, u(0))$ [77].

Remark 6.2

By Assumption 6.3, without loss of generality, we can set $(y)_{\{1,\dots,n_x\}} = x$, i.e., the states x are included as the first n_x outputs of system (6.1).

Then, by Eq. (6.2), $\text{Proj}_x\mathcal{Y} = \mathcal{X}$. Hence, the state constraints defined by $\mathcal{X}$ are redundant, since now they are already included in $\mathcal{Y}$.

Now, let system (6.1) act as a CT plant, controlled by Nonlinear MPC (NMPC).

First, we construct a discrete-time (DT) prediction model of plant (6.1), by applying a discretization method of choice (refer to Section 2.2, p. 12), with discrete time step $T_s > 0$, obtaining

$$x_{k+1} = f(x_k, u_k), \tag{6.3a}$$
$$y_k = g(x_k, u_k), \quad k \in \mathbb{Z}_{\geq 0}, \tag{6.3b}$$

where $x_k = x(kT_s)$, $u_k = u(kT_s)$, and $y_k = y(kT_s)$.

Nonlinear MPC provides, at each discrete time instant $k \geq 0$, an optimal control input u_k^*, obtained by solving the following NLP optimal control problem:

$$\min_{\hat{\boldsymbol{x}},\hat{\boldsymbol{u}},\hat{\boldsymbol{y}}} \quad J(\hat{\boldsymbol{y}}, \hat{\boldsymbol{u}}) = \sum_{i=0}^{N_p-1} \|\hat{y}_i - y_{r,k}\|^2_{Q_k} + \|\hat{u}_i\|^2_{R_k} + \sum_{i=1}^{N_p-1} \|\hat{y}_i - \hat{y}_{i-1}\|^2_{Q_{\Delta,k}} + \|\hat{u}_i - \hat{u}_{i-1}\|^2_{R_{\Delta,k}} \tag{6.4a}$$

$$\text{s.t.} \quad \hat{x}_0 = x_k, \tag{6.4b}$$
$$\hat{x}_{i+1} = f(\hat{x}_i, \hat{u}_i), \quad i = 0, \dots, N_p - 1, \tag{6.4c}$$

$$\hat{y}_i = g(\hat{x}_i, \hat{u}_i), \quad i = 0, \dots, N_p - 1, \tag{6.4d}$$
$$H_u \hat{u}_i \leq h_u, \quad i = 0, \dots, N_p - 1, \tag{6.4e}$$
$$H_y \hat{y}_i \leq h_y, \quad i = 0, \dots, N_p - 1. \tag{6.4f}$$

The NMPC problem (6.4) is formulated to attain regulation towards the admissible output $y_{r,k}$. It is worth noticing, however, that such a reference output changes at each $k \geq 0$, but we have knowledge of its value only at the current time instant k. As a consequence, we consider it as a constant reference along the prediction horizon.

The cost function J in Eq. (6.4a), similarly to that in Eq. (3.98a), is composed of four terms:

- The quadratic regulation term $\|\hat{y}_i - y_{r,k}\|^2_{Q_k}$ for the predicted output, towards the reference output $y_{r,k}$, with $Q_k \succeq 0$.
- The quadratic term $\|\hat{u}_i\|^2_{R_k}$, with $R_k \succ 0$, penalizing the input magnitude [32].
- The quadratic terms $\|\hat{y}_i - \hat{y}_{i-1}\|^2_{Q_{\Delta,k}}$ and $\|\hat{u}_i - \hat{u}_{i-1}\|^2_{R_{\Delta,k}}$, with $Q_{\Delta,k} \succeq 0$ and $R_{\Delta,k} \succeq 0$, which penalize the variation in time of the predicted outputs and inputs, to obtain smoother predicted trajectories [32].

The weighting matrices Q_k, R_k, $Q_{\Delta,k}$, and $R_{\Delta,k}$ in Eq. (6.4a) are diagonal and may change at each time instant $k \geq 0$.

The prediction model (6.4c), (6.4d) is given by Eqs. (6.3a) and (6.3b), respectively. Eqs. (6.4e) and (6.4f) report linear input and output constraints, given by Eq. (6.2) and following Remark 6.2.

The diagonal elements of the weighting matrices in Eq. (6.4a) are called *MPC weights* and we collect them in a vector w_k, as follows:

$$\begin{aligned} w_k &= \left[w_{Q,k}^\top,\ w_{R,k}^\top,\ w_{Q_\Delta,k}^\top,\ w_{R_\Delta,k}^\top\right]^\top \in \mathbb{R}^{n_w}, \\ w_{Q,k} &= \mathrm{diag}(Q_k), \quad w_{R,k} = \mathrm{diag}(R_k), \\ w_{Q_\Delta,k} &= \mathrm{diag}(Q_{\Delta,k}), \quad w_{R_\Delta,k} = \mathrm{diag}(R_{\Delta,k}), \end{aligned} \tag{6.5}$$

where $n_w = 2(n_y + n_u)$.

6.2.1. Approximation of the NMPC Problem

The NMPC problem (6.4) involves a nonlinear prediction model (6.4c), (6.4d) and linear constraints (6.4e), (6.4f).

Therefore, similarly to what we have done in Section 2.3.3, we would like to resort to Linear-Quadratic MPC (QP-MPC), rather than NMPC, for controlling system (6.1) with improved computational performance. Consequently, we aim to approximate the NMPC problem (6.4) and solve it by means of quadratic programming (QP).

To achieve this approximation, we only need to address the nonlinearity of the prediction model (6.4c), (6.4d). To this end, we transform the nonlinear prediction model into an affine parameter-varying (APV) system.

However, differently from the approach used in Section 2.3.3, in the NMPC problem (6.4) the reference output $y_{r,k}$ is known only at the current time instant k. Consequently, evaluating the APV prediction model along the shifted optimal trajectory obtained at the previous time instant $k-1$ may fail to effectively compensate for the modeling error

introduced by linearization, since the current reference $y_{r,k}$ may differ significantly from the previous one $y_{r,k-1}$. Therefore, in this case, it is more convenient to construct the APV model so to depend only on the current system state x_k, acting as the only parameter, i.e.,

$$\begin{aligned}\hat{x}_{i+1} = f(\hat{x}_i, \hat{u}_i) &\approx A(x_k)\hat{x}_i + B(x_k)\hat{u}_i + b(x_k), \\ &= A_k\hat{x}_i + B_k\hat{u}_i + b_k, \qquad (6.6a)\\ \hat{y}_i = g(\hat{x}_i, \hat{u}_i) &\approx C(x_k)\hat{x}_i + D(x_k)\hat{u}_i + d(x_k), \\ &= C_k\hat{x}_i + D_k\hat{u}_i + d_k. \qquad (6.6b)\end{aligned}$$

To derive the APV model in Eq. (6.6), we propose a procedure based on least-squares (LS) fitting.

In the following, let us define $\mathcal{Z} = \mathcal{X} \times \mathcal{U}$ and employ the notations $z = [x^\top, u^\top]^\top$ and $z = (x, u)$ interchangeably, to denote a couple of state and input vectors.

For any value of the parameter x_k, we want to ensure that the corresponding affine model (6.6a), (6.6b) is the most accurate approximation, in the LS sense, of the true nonlinear model (6.4c), (6.4d) over the set $\hat{\mathcal{Z}}_k = \hat{\mathcal{X}}_k \times \mathcal{U}$, defined as

$$\begin{aligned}&\hat{\mathcal{X}}_k = \{x \in \mathbb{R}^{n_x} : x_k + N_p\,\Delta x_{\min} \le x \le x_k + N_p\,\Delta x_{\max}\}, \\ &\Delta x_{\min,j} = \min_{(x,u)\in\mathcal{Z}} f_j(x,u) - x_j, \quad j = 1, \dots, n_x, \\ &\Delta x_{\max,j} = \max_{(x,u)\in\mathcal{Z}} f_j(x,u) - x_j, \quad j = 1, \dots, n_x, \qquad (6.7)\end{aligned}$$

where $\Delta x_{\min} = [\Delta x_{\min,j}]_{j=1}^{n_x}$, $\Delta x_{\max} = [\Delta x_{\max,j}]_{j=1}^{n_x}$, $[x_j]_{j=1}^{n_x} = x$, and $[f_j(x,u)]_{j=1}^{n_x} = f(x,u)$ in Eq. (6.3a).

The set $\hat{\mathcal{X}}_k$ is a hyperrectangular over-approximation of the set of reachable states within the MPC prediction horizon, starting from the current state x_k. To obtain the most accurate affine model over $\hat{\mathcal{Z}}_k$, we compute the parameter-varying terms A_k, B_k, b_k, C_k, D_k, and d_k in Eq. (6.6) through LS fitting, using the following fitting data:

$$\begin{aligned}&z_i \in \hat{\mathcal{Z}}_k, \quad i = 1, \dots, N_{\text{data}}, \qquad (6.8a)\\ &\phi_z = [z_1,\ \dots,\ z_{N_{\text{data}}}] \in \mathbb{R}^{n_x+n_u\times N_{\text{data}}}, \qquad (6.8b)\\ &\phi_x = [x_1^+,\ \dots,\ x_{N_{\text{data}}}^+] = [f(z_1),\ \dots,\ f(z_{N_{\text{data}}})] \in \mathbb{R}^{n_x\times N_{\text{data}}}, \qquad (6.8c)\\ &\phi_y = [y_1,\ \dots,\ y_{N_{\text{data}}}] = [g(z_1),\ \dots,\ g(z_{N_{\text{data}}})] \in \mathbb{R}^{n_y\times N_{\text{data}}}, \qquad (6.8d)\end{aligned}$$

where N_{data} is the number of fitting data points. Defining $\Phi = [\phi_z^\top, \mathbf{1}_{N_{\text{data}}}]$, by LS fitting we have that

$$[A_k,\ B_k,\ b_k] = \phi_x{\Phi^\dagger}^\top, \quad [C_k,\ D_k,\ d_k] = \phi_y{\Phi^\dagger}^\top, \qquad (6.9)$$

where the superscript † is the pseudoinverse matrix operator.

It is worth noting that the fitting data employed in Eqs. (6.8) and (6.9) is not obtained via direct measurement on the plant (6.1); instead, it is "virtually" collected from the DT prediction model (6.3), as in Eqs. (6.8c) and (6.8d).

Remark 6.3

By the LS fitting in Eqs. (6.8) and (6.9), the parameters $A_k = A(x_k)$, $B_k = B(x_k)$, $b_k = b(x_k)$, $C_k = C(x_k)$, $D_k = D(x_k)$, and $d_k = d(x_k)$ in Eq. (6.6) are all continuous functions of the current state x_k.

By replacing the nonlinear prediction model in Eqs. (6.4c), (6.4d) with the APV model in Eqs. (6.6a), (6.6b), we obtain the following QP-MPC problem, approximating the original NMPC problem (6.4), for each $k \geq 0$:

$$\min_{\hat{\boldsymbol{x}},\hat{\boldsymbol{u}},\hat{\boldsymbol{y}}} \quad J(\hat{\boldsymbol{y}},\hat{\boldsymbol{u}}) \tag{6.10a}$$

$$\text{s.t.} \quad \hat{x}_0 = x_k, \tag{6.10b}$$

$$\hat{x}_{i+1} = A_k\hat{x}_i + B_k\hat{u}_i + b_k, \quad i = 0,\dots,N_p-1, \tag{6.10c}$$

$$\hat{y}_i = C_k\hat{x}_i + D_k\hat{u}_i + d_k, \quad i = 0,\dots,N_p-1, \tag{6.10d}$$

$$H_u\hat{u}_i \leq h_u, \quad i = 0,\dots,N_p-1, \tag{6.10e}$$

$$H_y\hat{y}_i \leq h_y, \quad i = 0,\dots,N_p-1. \tag{6.10f}$$

6.3. Neural Adaptive Model Predictive Control

In the following, starting from the MPC formulation in Section 6.2, we present our novel *Neural Adaptive MPC* (NA-MPC) control strategy [33].

As stated in Section 6.1, the construction of NA-MPC requires three main steps:

1) Under the assumption that an accurate white-box model of the CT plant (6.1) is not available, construct a black-box model of the plant by means of feedforward neural networks (FNNs). Then, employ such a black-box model to derive the APV prediction model of the QP-MPC problem (6.10).
2) Represent the QP-MPC problem (6.10) as a static control policy $u_k = \pi(\Pi_k)$, where Π_k are the parameters of the QP-MPC problem (6.10), and employ a suitable feedforward neural network to accurately emulate it.
3) Formulate an MPC tuning algorithm that, given a set of control specifications, is able to obtain optimal parameters Π_k^* in real time, adapting the QP-MPC policy π online.

6.3.1. Feedforward Neural Networks

Several results presented in the following rely on the property of universal approximation of feedforward neural networks.

Definition 6.1 (Feedforward neural network)

A *feedforward neural network* (FNN) with L fully-connected layers is a static function $\mathcal{N} : \mathbb{R}^{n_{\text{in}}} \to \mathbb{R}^{n_{\text{out}}}$ with the following structure:

$$z_{\text{out}} = \mathcal{N}(z_{\text{in}}) = (\phi_L \circ \cdots \circ \phi_1)(z_{\text{in}}), \tag{6.11a}$$

$$z_{l+1} = \phi_l(z_l) = \sigma_l(W_l z_l + b_l), \quad l = 1,\dots,L. \tag{6.11b}$$

The collection of network inputs z_{in} is called *input layer*. The layers $l = 1, \ldots, L-1$ are called *hidden layers*. The layer $l = L$ is called *output layer*.

Each l-th layer ϕ_l is characterized by:

- the learnable parameters, i.e., the weights $W_l \in \mathbb{R}^{N_l \times N_{l-1}}$ and biases $b_l \in \mathbb{R}^{N_l}$;
- the number of neurons N_l ($N_0 = n_{\text{in}}$, $N_L = n_{\text{out}}$);
- the activation function $\sigma_l : \mathbb{R} \to \mathbb{R}$, which is applied element-wise in Eq. (6.11b); the output layer has a linear activation function, i.e., $\sigma_L(z) = z$.

The approximation capability of FNNs is formally proven by the well-known *universal approximation theorem* [72, 92].

Definition 6.2 (Dense set)

Let $\mathcal{Z} \subset \mathbb{R}^n$ be a compact set and consider the $L^p(\mathcal{Z})$ space of functions $f : \mathcal{Z} \to \mathbb{R}$, equipped with the p-norm, i.e.,

$$\|f\|_p = \begin{cases} (\int_{\mathcal{Z}} |f(z)|^p dz)^{1/p} & \text{if } 1 \le p < \infty, \\ \operatorname{ess\,sup}_{z \in \mathcal{Z}} |f(z)| & \text{if } p = \infty. \end{cases} \tag{6.12}$$

Consider two sets of functions $\mathscr{A}, \mathscr{B} \subseteq L^p(\mathcal{Z})$. We say that $\mathscr{A}$ is *dense* in $\mathscr{B}$ (with respect to the p-norm) if, for every $g \in \mathscr{B}$ and every $\varepsilon > 0$, there exists $f \in \mathscr{A}$ such that

$$\|f - g\|_p < \varepsilon. \tag{6.13}$$

Theorem 6.1 (FNNs universal approximation [72, 92])

Let us consider a feedforward neural network $\mathcal{N} : \mathbb{R}^{n_{\text{in}}} \to \mathbb{R}^{n_{\text{out}}}$ with $L = 2$ (single hidden layer) and $n_{\text{out}} = 1$ (single output), i.e.,

$$z_{\text{out}} = \mathcal{N}_\theta(z_{\text{in}}) = W_2 \sigma(W_1 z_{\text{in}} + b_1) + b_2, \tag{6.14}$$

where $\theta = (W_1, b_1, N, \sigma, W_2, b_2) \in \Theta$ collects the network parameters $W_1 \in \mathbb{R}^{N \times n_{\text{in}}}$, $b_1 \in \mathbb{R}^N$, $N \in \mathbb{Z}_{\ge 0}$, $\sigma : \mathbb{R} \to \mathbb{R}$, $W_2 \in \mathbb{R}^{1 \times N}$, $b_2 \in \mathbb{R}$. Let $\mathscr{N} = \{\mathcal{N}_\theta : \theta \in \Theta\}$ denote the family of FNNs defined in Eq. (6.14).

Assume that the activation function σ in Eq. (6.14) is locally bounded, piecewise continuous, and non-polynomial.

Let $\mathcal{Z} \subseteq \mathbb{R}^{n_{\text{in}}}$ be a compact set and let $C(\mathcal{Z})$ be the space of continuous functions $f : \mathcal{Z} \to \mathbb{R}$.

Then, the following claims are true and equivalent:

i) The set $\mathscr{N}$ is dense in $C(\mathcal{Z})$ with respect to the ∞-norm.

ii) For every continuous function $f \in C(\mathcal{Z})$ and every $\varepsilon > 0$, there exists a FNN $\mathcal{N} \in \mathscr{N}$ such that

$$\|\mathcal{N} - f\|_\infty = \sup_{z \in \mathcal{Z}} |\mathcal{N}(z) - f(z)| < \varepsilon. \tag{6.15}$$

In practical terms, Theorem 6.1 states that, for any compact set $\mathcal{Z} \subseteq \mathbb{R}^{n_{\text{in}}}$, FNNs can approximate any continuous function f over $\mathcal{Z}$ with arbitrary accuracy, provided that [72, 92]:

- the activation function σ is locally bounded, piecewise continuous and non-polynomial;
- the number of hidden neurons N is sufficiently high;
- the datasets $\mathcal{D}_{\text{in}} \subset \mathcal{Z}$ and $\mathcal{D}_{\text{out}} \subset f(\mathcal{Z}) \subseteq \mathbb{R}$ used to train the FNN are sufficiently rich.

Remark 6.4

Theorem 6.1 can be easily extended to the case of multilayer and multioutput FNNs, i.e., $L \geq 3$ (2 or more hidden layers) and $n_{\text{out}} \geq 2$ [92].

For these networks, the results of Theorem 6.1 equivalently hold, considering the space $C(\mathcal{Z})$ of continuous functions $f : \mathcal{Z} \to \mathbb{R}^{n_{\text{out}}}$.

6.3.2. Neural Black-Box Model of the Continuous-Time Plant

MPC requires a sufficiently accurate prediction model of the plant to deliver an effective control action. However, real-world scenarios often present a lack of relevant physical information on the plant, posing a significant challenge for conventional white or gray-box modeling.

Therefore, in the following, we assume that, for the CT plant (6.1), an accurate white-box model (i.e., derived by means of the first principles of physics) is not available. Additionally, we assume that, from the plant (6.1), we can only collect measurements of the input $(\tilde{u}_i)_{i=1}^{N_{\text{meas}}}$ and output $(\tilde{y}_i)_{i=1}^{N_{\text{meas}}}$, both corrupted by measurement errors n_u and n_y, respectively, as follows:

$$\dot{x}_i = f_c(x_i, u_i), \quad y_i = g(x_i, u_i), \quad i = 1, \dots, N_{\text{meas}}, \tag{6.16a}$$

$$\tilde{u}_i = u_i \odot n_u, \quad n_u \overset{\text{i.i.d.}}{\sim} N(1, \sigma_u^2), \quad i = 1, \dots, N_{\text{meas}}, \tag{6.16b}$$

$$\tilde{y}_i = y_i \odot n_y, \quad n_y \overset{\text{i.i.d.}}{\sim} N(1, \sigma_y^2), \quad i = 1, \dots, N_{\text{meas}}, \tag{6.16c}$$

where N_{meas} is the number of collected measurements and $\odot$ is the element-wise product operator. In Eq. (6.16), n_u and n_y represent element-wise multiplicative errors, modeled as vectors of independent and identically distributed (i.i.d.) random variables, following a normal distribution N with mean 1 and variance $\sigma_u = \overline{n}_u/3$ and $\sigma_y = \overline{n}_y/3$, where $\overline{n}_u$ and $\overline{n}_y$ are the "3-sigma" values of the two distributions. Note that, any other modeling choice for the measurement errors is equally admissible.

By Assumption 6.3, we can measure the error-corrupted states $(\tilde{x}_i)_{i=1}^{N_{\text{meas}}}$ and, from them, we can also estimate the error-corrupted values $(\dot{\tilde{x}}_i)_{i=1}^{N_{\text{meas}}}$, by approximating the time derivative with finite differences.

Finally, we construct the input and output datasets

$$\mathcal{D}_{\text{in}} = ((\tilde{x}_i, \tilde{u}_i))_{i=1}^{N_{\text{meas}}}, \quad \mathcal{D}_{\text{out}} = ((\dot{\tilde{x}}_i, \tilde{y}_i))_{i=1}^{N_{\text{meas}}}. \tag{6.17}$$

We employ the collected data $\mathcal{D}_{\text{in}}$ and $\mathcal{D}_{\text{out}}$ to identify a neural black-box model of the CT plant (6.1). Such a model is composed by two FNNs, $\mathcal{N}_{f_c}$ and $\mathcal{N}_g$, to approximate the

state dynamics function f_c and the output function g, respectively, i.e.,[1]

$$\dot{x}(t) = \mathcal{N}_{f_c}(x(t), u(t)), \tag{6.18a}$$
$$y(t) = \mathcal{N}_g(x(t), u(t)). \tag{6.18b}$$

By Assumption 6.1, f_c and g are continuous on the compact set $\mathcal{Z} = \mathcal{X} \times \mathcal{U}$. Thus, Theorem 6.1 ensures that f_c and g can be approximated over $\mathcal{Z}$ with arbitrary accuracy by the FNNs $\mathcal{N}_{f_c}$ and $\mathcal{N}_g$, respectively. Moreover, if the activation functions σ_l of $\mathcal{N}_{f_c}$ and $\mathcal{N}_g$ are Lipschitz continuous, then both networks are Lipschitz continuous as well [60]. Therefore, the neural plant model (6.18) satisfies Assumptions 6.1 and 6.3.

The neural plant model (6.18) is then discretized, by applying a discretization method of choice (refer to Section 2.2, p. 12) with discrete time step T_s, obtaining

$$x_{k+1} = \mathcal{N}_f(x_k, u_k), \tag{6.19a}$$
$$y_k = \mathcal{N}_g(x_k, u_k). \tag{6.19b}$$

The discretized model (6.19) is then converted in APV form, as described in Section 6.2.1, so to be used as prediction model of the QP-MPC problem (6.10).

6.3.3. Neural Emulation of the MPC Control Policy

The QP-MPC problem (6.10), employing the APV form of the DT neural plant model (6.19) as its prediction model, can be equivalently rewritten in a compact QP form, as follows (refer to Section 2.4 for more details):

$$\min_{\hat{\boldsymbol{v}}} \quad \frac{1}{2}\hat{\boldsymbol{v}}^\top H_k \hat{\boldsymbol{v}} + c_k^\top \hat{\boldsymbol{v}} \tag{6.20a}$$
$$\text{s.t.} \quad M_k \hat{\boldsymbol{v}} = p_k, \tag{6.20b}$$
$$N\hat{\boldsymbol{v}} \leq q, \tag{6.20c}$$

where $\hat{\boldsymbol{v}}$ collects the decision variables $\hat{\boldsymbol{x}}$, $\hat{\boldsymbol{u}}$, and $\hat{\boldsymbol{y}}$ in Eq. (6.10), $H_k \succ 0$, and M_k has full row rank [22, 35]. Each time-varying term of problem (6.20) is explicitly dependent on the time-varying parameters of the QP-MPC problem (6.10), i.e.,

$$H_k = H(Q_k, Q_{\Delta,k}, R_k, R_{\Delta,k}), \tag{6.21a}$$
$$c_k = c(y_{r,k}, Q_k), \tag{6.21b}$$
$$M_k = M(A_k, B_k, C_k, D_k), \tag{6.21c}$$
$$p_k = p(x_k, b_k, d_k). \tag{6.21d}$$

In Eq. (6.21), H, c, M, and p are all continuous functions [22].

Let Π_k denote the tuple of time-varying parameters of the QP-MPC problem (6.10) at the current time instant k, i.e.,

$$\Pi_k = (x_k, y_{r,k}, Q_k, R_k, Q_{\Delta,k}, R_{\Delta,k}, A_k, B_k, b_k, C_k, D_k, d_k). \tag{6.22}$$

[1] Hereafter, for notational convenience, we split up FNN inputs into multiple arguments; e.g., in Eq. (6.18), the notations $\mathcal{N}(x, u)$ and $\mathcal{N}(z)$, with $z = [x^\top, u^\top]^\top$, are equivalent.

In problem (6.20), the cost and the inequality constraints are convex functions; thus, the QP-MPC problem (6.10) and its compact form (6.20) admit the same, unique global optimum $\hat{\boldsymbol{v}}^*$ [23]. Then, the QP-MPC problem (6.10) can be represented by a static control policy π, associating each feasible value of Π_k with the related unique solution $u_k = \hat{u}^*_{0|k}$ (according to the receding horizon policy, see Section 2.2), as follows:

$$u_k = \pi(\Pi_k). \tag{6.23}$$

Since Π_k includes both the MPC weighting matrices Q_k, R_k, $Q_{\Delta,k}$, $R_{\Delta,k}$ and the APV prediction model terms A_k, B_k, b_k, C_k, D_k, d_k, the computation of the static policy (6.23) cannot be tackled with Explicit MPC (see Section 2.4).

Therefore, in the following, we rely on the universal approximation capability of FNNs to obtain an accurate estimate of the policy (6.23).

To properly approximate the QP-MPC policy (6.23) with a FNN, we have to verify first that such a policy is continuous. Several works have investigated the continuity of QPs, both in terms of parameter perturbation [45] and upper-lower semicontinuity [146]. For our purposes, we employ the following result:

Theorem 6.2 (Continuity of quadratic programs [45])

Let the following QP problem be given:

$$\min_x \quad \frac{1}{2}x^\top Hx + c^\top x \tag{6.24a}$$
$$\text{s.t.} \quad Mx = p, \tag{6.24b}$$
$$Nx \leq q, \tag{6.24c}$$

where $H \succ 0$ and M has full row rank. Let $\Pi = (H, c, M, p, N, q)$ denote the tuple of parameters of the QP problem (6.24) and consider the function $x^* = \pi(\Pi)$, which provides the unique optimum of problem (6.24) as a function of the parameters Π.

Then, there exist $\alpha > 0$ and $\delta > 0$ such that, for any $\varepsilon \leq \delta$, if $\Pi' = (H', c', M', p', N', q')$ satisfies

$$\max\{\|H - H'\|, \|c - c'\|, \|M - M'\|, \|p - p'\|, \|N - N'\|, \|q - q'\|\} \leq \delta, \tag{6.25}$$

then it holds

$$\|x^* - x^{*\prime}\| = \|\pi(\Pi) - \pi(\Pi')\| \leq \alpha\varepsilon. \tag{6.26}$$

Therefore, Theorem 6.2, leveraging Weierstrass' definition of continuity, proves that the optimum x^* of the QP problem (6.24) is a continuous function of the QP parameters Π.

Since problem (6.20) matches the structure in Eq. (6.24) and the functions in Eq. (6.21) are all continuous, then, by Theorem 6.2, the QP-MPC policy π (6.23) is continuous. Therefore, by Theorem 6.1, it can be effectively approximated by a FNN $\mathcal{N}_\pi$, i.e.,

$$u_k = \mathcal{N}_\pi(\Pi_k). \tag{6.27}$$

The FNN (6.27) constitutes the *neural MPC controller*.

Remark 6.5

While continuity results can be established for QP-MPC policies (6.23), the same cannot be done for NMPC policies, which are, in general, discontinuous. Therefore, FNNs cannot approximate NMPC with arbitrary accuracy.

The number of inputs Π_k of the neural MPC controller (6.27) can be reduced by recalling Remark 6.3, for which we can keep only the parameter x_k in place of A_k, B_k, b_k, C_k, D_k, d_k, while still retaining the continuity of π, as required by Theorem 6.1. Moreover, from Eq. (6.5), the parameters Q_k, R_k, $Q_{\Delta,k}$, and $R_{\Delta,k}$ can be replaced by their diagonal elements, i.e., the MPC weights w_k in Eq. (6.5).

Then, the final structure of the neural MPC controller is

$$u_k = \mathcal{N}_\pi(x_k, y_{r,k}, w_k). \tag{6.28}$$

The neural MPC controller (6.27) is trained by constructing input and output datasets $\mathcal{D}_{\text{in}}$ and $\mathcal{D}_{\text{out}}$: $\mathcal{D}_{\text{in}}$ is obtained by selecting a set of feasible values of Π; $\mathcal{D}_{\text{out}}$ is obtained by solving the QP-MPC problem (6.10) for each parameter value in $\mathcal{D}_{\text{in}}$.

Remark 6.6

Emulating the QP-MPC policy (6.23) using a FNN (6.27) offers a significant advantage: unlike solving online the original QP-MPC problem (6.10) by means of conventional QP solvers, evaluating the policy (6.27) requires near-zero computation time, since FNNs are static functions.

Also, unlike Explicit MPC, the policy (6.27) does not require to solve a point location problem prior to its evaluation, which constitutes the major bottleneck of Ex-MPC (see Section 2.4.2 for more details).

These are key features for enabling the use of an online optimal tuning strategy for the neural MPC controller, as explained in the next section.

6.4. Online Metaheuristic Strategy for Adaptive MPC Tuning

Online tuning has the purpose of adapting in real time the MPC weights w_k (6.5), allowing to maintain consistent control performance over time and to perform optimal trade-offs between conflicting and concurrent control objectives.

Online tuning is especially useful when the reference output trajectory $(y_{r,k})_{k\geq 0}$ is available only up to the current time instant k and is not known a-priori for all $k \geq 0$, thus not allowing to precompute offline the optimal weights; this is the case of our NMPC problem (6.4).

To this end, we propose an online tuning strategy which, at the current time k (or $t = kT_s$), employs past information on the closed-loop system evolution, collected over the time interval $[(k - N_{\text{tuning}})T_s, (k-1)T_s]$, with $N_{\text{tuning}} \in \mathbb{Z}_{\geq 1}$, to obtain some optimal weights w_k^* to be used over the next time interval $[kT_s, (k + N_{\text{tuning}} - 1)T_s]$.

6.4.1. Performance-Metric Functions

Let us assume that we want to mediate N_{obj} concurrent control objectives, which may, in general, be conflicting with each other. For each i-th control objective, $i = 1, \ldots, N_{\text{obj}}$,

we define a *performance-metric index* p_i, which serves as a quantitative measure of the extent to which the corresponding objective is satisfied: the smaller p_i, the better the i-th objective is achieved.

The performance-metric indices are evaluated as functions of the N-step output trajectory $\boldsymbol{y} = (y_i)_{i=1}^{N}$. Accordingly, we define the *performance-metric functions* $p_i(\boldsymbol{y})$, where $p_i : \mathbb{R}^{n_y N} \to \mathbb{R}$.

6.4.2. Optimal Tuning Problem

The tuning strategy, which seeks the optimal weights w_k^* at time k, is formulated as the following optimization problem:

$$\min_{w} \quad J_{\text{tuning},k}(w) \tag{6.29a}$$

$$\text{s.t.} \quad w \in \mathcal{W} = \{w \in \mathbb{R}^{n_x} : w_{\text{lb}} \leq w \leq w_{\text{ub}}\}. \tag{6.29b}$$

The tuning cost function $J_{\text{tuning},k}$ in Eq. (6.29a) is not given as a closed-form expression; instead, the cost value $J_{\text{tuning},k}(w)$, for a given weight $w \in \mathbb{R}^{n_w}$ and time instant $k \geq 0$, is evaluated through the following procedure:

1) Simulate, over the (past) N_{tuning} discrete time instants $i = k - N_{\text{tuning}}, \ldots, k-1$, the closed-loop system composed by the DT neural plant model (6.19) and the neural MPC controller (6.28), i.e.,

$$\hat{x}_{i+1} = \mathcal{N}_f(\hat{x}_i,\, \mathcal{N}_\pi(\hat{x}_i, y_{r,i}, w)), \quad \hat{x}_{k-N_{\text{tuning}}} = x_{k-N_{\text{tuning}}}, \tag{6.30a}$$

$$\hat{y}_i = \mathcal{N}_g(\hat{x}_i,\, \mathcal{N}_\pi(\hat{x}_i, y_{r,i}, w)), \tag{6.30b}$$

   obtaining the simulated closed-loop output trajectory $\hat{\boldsymbol{y}} = (\hat{y}_i)_{i=k-N_{\text{tuning}}}^{k-1}$.

2) For each i-th control objective, evaluate the related performance-metric functions through the simulated trajectory $\hat{\boldsymbol{y}}$, i.e., $p_i(\hat{\boldsymbol{y}})$, $i = 1, \ldots, N_{\text{obj}}$.

3) Compute the cost value as

$$J_{\text{tuning},k}(w) = \sum_{j=1}^{N_{\text{obj}}} \alpha_i \, p_i(\hat{\boldsymbol{y}}), \tag{6.31}$$

   where $\alpha_i \in \mathbb{R}_{>0}$ are scalars for setting the priority of each metric over the others.

Solving the tuning problem (6.29) provides the optimal weights w_k^*, which, according to the above procedure, yield the MPC control action that optimally trades off the concurrent control objectives over the previous N_{tuning} discrete time instants.

We employ these optimal weights w_k^* in the neural MPC controller (6.28) to control plant (6.1) over the next time interval $[kT_s, (k + N_{\text{tuning}} - 1)T_s]$. Hence, successive tunings are then performed every N_{tuning} time instants, i.e., the optimal weights w_k^* are kept constant over the time interval $[kT_s, (k + N_{\text{tuning}} - 1)T_s]$ and the tuning problem (6.29) is solved only at time instants $k = nN_{\text{tuning}}$, $n \in \mathbb{Z}_{\geq 1}$. Therefore, the weights w_0, employed during the first time interval $[0, (N_{\text{tuning}} - 1)T_s]$, have to be initialized beforehand.

As stated in Remark 6.6, the closed-loop simulation in Eq. (6.30) can be performed in real time since the control policy is evaluated through the neural MPC controller $\mathcal{N}_\pi$ (6.28), which, being a FNN, exhibits near-zero computation time.

The union of the neural MPC controller (6.28) with the online tuning strategy (6.29) constitutes the complete *NA-MPC controller*.

6.4.3. Tuning Problem Solution via Metaheuristic Optimization

The tuning problem (6.29) is not a standard optimization problem, since the cost function $J_{\text{tuning},k}(w)$ has to be evaluated through the procedure described in the previous section. As a consequence, it cannot be effectively solved with conventional optimization algorithms. For this reason, we leverage *metaheuristic optimization*.

Metaheuristic Optimization

Let us consider the following NLP optimization problem:

$$\min_{x} \quad J(x) \tag{6.32a}$$

$$\text{s.t.} \quad x \in \mathcal{X} = \{x \in \mathbb{R}^n : x_{\text{lb}} \leq x \leq x_{\text{ub}}\}. \tag{6.32b}$$

Conventional NLP solvers may not always be a feasible choice for solving problems like (6.32). Such solution methods are mainly deterministic and gradient-based (e.g., sequential quadratic programming, interior-point, active-set, etc.) and, thus, rely on the smoothness and differentiability of the cost function J to provide a reliable optimal solution.

As a consequence, when the cost function J exhibits a highly irregular shape (e.g., non-convex, discontinuous, etc.), conventional methods are prone to converging only to local minima, thus providing a sub-optimal and unreliable solution (see, e.g., Eq. (5.21) and Remark 5.3).

Even more critically, when the cost function J is not given in closed form, but instead has to be evaluated through algorithmic or simulation-based procedures that map the decision variables x to the related cost value $J(x)$ (like in our tuning problem (6.29)), conventional solvers may not be applicable at all, as the computation of gradients may not be possible.

In these situations, it is worth resorting to *metaheuristic optimization algorithms*. These kind of solution methods employ stochastic, population-based search strategies that balance global exploration and local exploitation, rendering them capable of escaping from local minima and performing a more robust search of the solution space [56]. Moreover, metaheuristic strategies naturally enforce bound constraints like (6.32b).

Among the various metaheuristic optimization techniques available [56, 107], we adopt the following two, both of which are widely recognized and extensively applied:

1) *Particle Swarm Optimization* (PSO), belonging to the family of swarm intelligence techniques.
2) *Genetic Algorithm* (GA), belonging to the family of evolutionary optimization techniques.

Particle Swarm Optimization Particle Swarm Optimization (PSO) is the leading metaheuristic algorithm among the family of swarm intelligence techniques [56].

In PSO, a *swarm of particles* $\mathcal{S} = (x_i)_{i=1}^{N_{\text{part}}}$, each one representing a candidate solution x_i of the optimization problem (6.32), explores the search domain $\mathcal{X}$ (6.32b) by progressively updating their positions over a number of iterations $j = 0, \ldots, N_{\text{PSO}}$.

At the first iteration $j = 0$, the particle swarm $\mathcal{S}^{(0)} = (x_i^{(0)})_{i=1}^{N_{\text{part}}}$ is randomly initialized

over the set $\mathcal{X}$, e.g.,

$$x_i^{(0)} \sim U([x_{\text{lb}}, x_{\text{ub}}]), \quad i = 1, \dots, N_{\text{part}}, \tag{6.33}$$

where U is the uniform probability distribution.

Each particle is associated with a velocity vector $v_i \in \mathbb{R}^n$, $i = 1, \dots, N_{\text{part}}$, which, at iteration $j = 0$, is also randomly initialized over $\mathcal{X}$.

At the j-th iteration, $j = 0, \dots, N_{\text{PSO}} - 1$, the particles positions $\mathcal{S}^{(j)} = (x_i^{(j)})_{i=1}^{N_{\text{part}}}$ and velocities $(v_i^{(j)})_{i=1}^{N_{\text{part}}}$ are updated as follows:

$$v_i^{(j+1)} = m v_i^{(j)} + c_{\text{i}} r_{\text{i}} \left(p_i^{(j)} - x_i^{(j)} \right) + c_{\text{s}} r_{\text{s}} \left(g^{(j)} - x_i^{(j)} \right), \tag{6.34a}$$

$$x_i^{(j+1)} = x_i^{(j)} + v_i^{(j+1)}, \quad i = 1, \dots, N_{\text{part}}. \tag{6.34b}$$

In Eq. (6.34a), $p_i^{(j)}$ is the personal best position of particle i up to iteration j, i.e.,

$$p_i^{(j)} = \underset{x \in \left\{ x_i^{(0)},\, x_i^{(1)},\, \dots,\, x_i^{(j)} \right\}}{\arg\min} J(x), \tag{6.35}$$

$g^{(j)}$ is the global best position of the swarm up to iteration j, i.e.,

$$g^{(j)} = \underset{x \in \mathcal{S}^{(0)} \cup \mathcal{S}^{(1)} \cup \dots \cup \mathcal{S}^{(j)}}{\arg\min} J(x), \tag{6.36}$$

$m \in \mathbb{R}_{>0}$ is the particle inertia; $c_{\text{i}}, c_{\text{s}} \in \mathbb{R}_{>0}$ are the individual and social attraction coefficients, respectively; $r_{\text{i}}, r_{\text{s}} \sim U([0, 1])$ are random numbers with uniform distribution.

In order to ensure that the updated positions $x_i^{(j+1)}$ stay within the set $\mathcal{X}$, two possible measures are typically adopted:

- *"Reflecting" bounds*, whereby, when a particle exceeds a bound, the related position component is reflected back into $\mathcal{X}$ with respect to the violated bound, and the velocity component is reversed, i.e., if the particle $x_i^{(j+1)}$ exceeds the k-th bound, $k = 1, \dots, n$, then the k-th component of the updated position $\tilde{x}_i^{(j+1)}$ and velocity $\tilde{v}_i^{(j+1)}$ are

$$(\tilde{x}_i^{(j+1)})_k = \begin{cases} 2(x_{\text{lb}})_k - (x_i^{(j+1)})_k & \text{if } (x_i^{(j+1)})_k < (x_{\text{lb}})_k, \\ 2(x_{\text{ub}})_k - (x_i^{(j+1)})_k & \text{if } (x_i^{(j+1)})_k > (x_{\text{ub}})_k, \end{cases} \tag{6.37a}$$

$$(\tilde{v}_i^{(j+1)})_k = -(v_i^{(j+1)})_k. \tag{6.37b}$$

- *"Absorbing" bounds*, whereby, when a particle exceeds a bound, the related position component is set equal to the violated bound, and the velocity component is set to zero, i.e., if the particle $x_i^{(j+1)}$ exceeds the k-th bound, then

$$(\tilde{x}_i^{(j+1)})_k = \begin{cases} (x_{\text{lb}})_k & \text{if } (x_i^{(j+1)})_k < (x_{\text{lb}})_k, \\ (x_{\text{ub}})_k & \text{if } (x_i^{(j+1)})_k > (x_{\text{ub}})_k, \end{cases} \tag{6.38a}$$

$$(\tilde{v}_i^{(j+1)})_k = \mathbf{0}_n. \tag{6.38b}$$

The optimal solution to problem (6.32) is selected as the global best position at the last iteration $j = N_{\text{PSO}}$, i.e.,

$$x^* = g^{(N_{\text{PSO}})}. \tag{6.39}$$

Concerning the tuning problem (6.29), the optimal solution w_k^*, for $k = nN_{\text{tuning}}$, $n \geq 1$, is computed as follows:

$$w_k^* = \frac{1}{2}\left(g^{(N_{\text{PSO}})} + w_{k-N_{\text{tuning}}}^*\right). \tag{6.40}$$

The policy (6.40) has the purpose of smoothing the evolution of the optimal MPC weights over time.

Remark 6.7

Concerning parameter selection for PSO, [107] suggests optimal choices for m, c_{i}, and c_{s} that, in general, allow to achieve the best balance between exploration and exploitation. Such values are reported in the data on p. 214.

Genetic Algorithm Genetic Algorithm (GA) is a widely employed metaheuristic optimization method inspired by the process of natural selection.

It works by evolving a *population* of candidate solutions $\mathcal{P} = (x_i)_{i=1}^{N_{\text{ind}}}$ (i.e., the individuals) over a number of iterations $j = 0, \ldots, N_{\text{GA}}$ (i.e., generations) [83, 107].

At the first generation $j = 0$, the initial population $\mathcal{P}^{(0)}$ is randomly initialized over the set $\mathcal{X}$ of problem (6.32), e.g.,

$$x_i^{(0)} \sim U([x_{\text{lb}}, x_{\text{ub}}]), \quad i = 1, \ldots, N_{\text{ind}}. \tag{6.41}$$

At the j-th generation, $j = 0, \ldots, N_{\text{GA}} - 1$, for each individual $x_i^{(j)}$, it is evaluated the related cost value $J_i^{(j)} = J(x_i^{(j)})$, which allows to quantify the individual "fitness" $\frac{1}{J_i^{(j)}}$.

Then, a *selection* process is performed: at each generation j, individuals are selected to create the new population of the next generation $j + 1$. Selection is based on the fitness values, where individuals with better fitness (i.e., lower $J_i^{(j)}$) have a higher probability of being selected. Among the many possible selection methods (i.e., roulette wheel, tournament, rank-based, etc. [107]), we adopt the stochastic uniform selection [83], i.e., each individual has a probability of being selected proportional to its fitness, i.e.,

$$\mathbb{P}[x_i^{(j)} \text{ is selected}] = \frac{\frac{1}{J_i^{(j)}}}{\sum_{i=1}^{N_{\text{ind}}} \frac{1}{J_i^{(j)}}}. \tag{6.42}$$

Selection produces the sub-population $\mathcal{P}'^{(j)}$, which undergoes *crossover*, i.e., pairs of individuals (i.e., the parents) exchange information to create offspring. For each couple of parents $x_a^{(j)}$ and $x_b^{(j)}$, $a, b \in \{1, \ldots, N_{\text{ind}}\}$, $a \neq b$, crossover consists in randomly selecting a binary mask $m \in \{0, 1\}^n$ and generating the offspring x' by extracting components from either $x_a^{(j)}$ or $x_b^{(j)}$ according to the mask, i.e.,

$$(x')_k = \begin{cases} (x_a^{(j)})_k & \text{if } (m)_k = 0, \\ (x_b^{(j)})_k & \text{if } (m)_k = 1, \end{cases} \quad k = 1, \ldots, n. \tag{6.43}$$

Offspring are added to $\mathcal{P}'^{(j)}$, obtaining the new population $\mathcal{P}^{(j+1)}$, corresponding to the next generation $j + 1$.

The number of generated offspring is typically selected to keep constant the number of individuals N_{ind} over the generations.

Finally, a *mutation* is applied to each individual of $\mathcal{P}^{(j+1)}$, as follows [83]:

$$\tilde{x}_i^{(j+1)} = x_i^{(j+1)} + \varepsilon, \quad \varepsilon \sim N(0, \sigma_j^2), \tag{6.44}$$

where ε is a random number with normal distribution N; the variance σ_j tunes the mutation magnitude and is adapted at each generation, as follows [83]:

$$\begin{aligned} \sigma_0 &= c_{\text{s}}(x_{\text{ub}} - x_{\text{lb}}), \\ \sigma_j &= \sigma_{j-1}\Big(1 - c_{\text{r}}\frac{j}{N_{\text{GA}}}\Big), \quad j = 1, \dots, N_{\text{GA}}, \end{aligned} \tag{6.45}$$

where c_{s} and c_{r} are tuning parameters for the scale factor and the reduction rate of σ_j, respectively.

To ensure that mutated individuals stay within the set $\mathcal{X}$, typically a saturation within the bounds x_{lb} and x_{ub} is applied after the mutation (6.44).

The optimal solution to problem (6.32) is selected as the individual with highest fitness at the last generation $j = N_{\text{GA}}$, i.e.,

$$x^* = \underset{x \in \mathcal{P}^{(N_{\text{GA}})}}{\arg\min}\ J(x). \tag{6.46}$$

6.5. Application to Power Management in Fuel Cell Hybrid Electric Vehicles

The NA-MPC strategy is applied to the case study of power management in fuel cell hybrid electric vehicles (FCHEVs).

In the following, we outline the construction of a white-box, first-principles model of the power sources and distribution system in a FCHEV, which constitute the FCHEV *power system*. Such a model will be employed, in simulation (Section 6.6), as the CT plant to control and as a "virtual" source of input-output measurements, as detailed in Section 6.3.2.

After that, we tailor the NA-MPC strategy, developed in Section 6.3, to achieve the main control tasks of FCHEV power management.

6.5.1. Modeling the Power System of a FCHEV

A simplified scheme of the FCHEV power system is reported in Figure 6.2:

- The battery and the fuel cell (FC) are the two hybrid sources of electrical power for the FCHEV propulsion. The two sources deliver the battery power P_{b} and FC power P_{fc}, respectively, to the DC bus. In the power system model, we consider P_{b} and P_{fc} as powers *requested* to each source, i.e., they act as inputs of the model,

$$u(t) = \begin{bmatrix} u_1(t) \\ u_2(t) \end{bmatrix} = \begin{bmatrix} P_{\text{b}}(t) \\ P_{\text{fc}}(t) \end{bmatrix}. \tag{6.47}$$

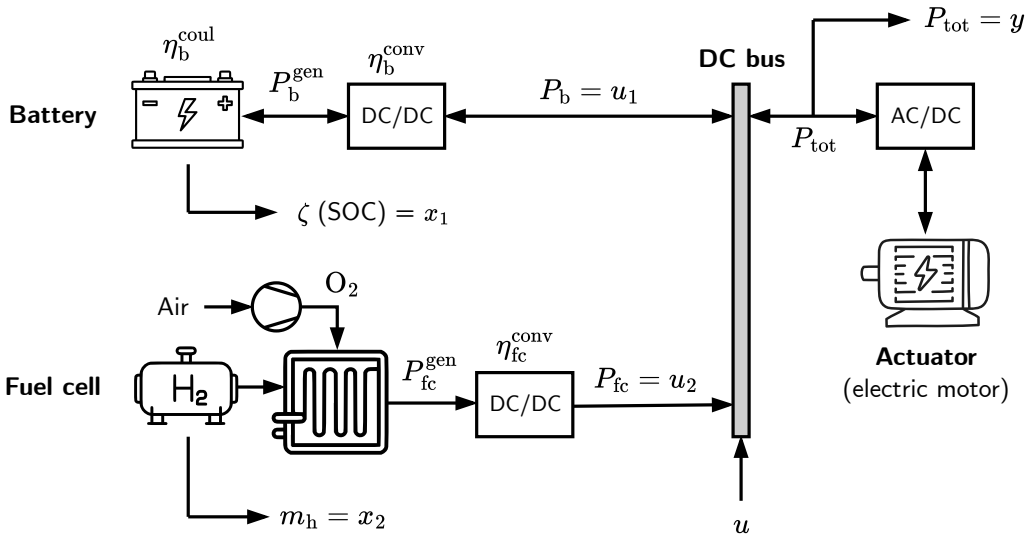


Figure 6.2. FCHEV power sources and distribution system.

- The two sources, battery and FC, are associated with their current supply value, i.e., the battery state of charge ζ (SOC) and the hydrogen (H_2) mass m_{h}, respectively. According to the requested powers $P_{\mathrm{b}}(t)$ and $P_{\mathrm{fc}}(t)$, the two supplies $\zeta(t)$ and $m_{\mathrm{h}}(t)$ change over time. Thus, they are the states of the power system model, i.e.,

$$x(t) = \begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix} = \begin{bmatrix} \zeta(t) \\ m_{\mathrm{h}}(t) \end{bmatrix}. \tag{6.48}$$

- The DC bus acts as a central electrical distribution system, allowing for bidirectional power transfer, i.e., the DC bus is capable of recharging the battery (i.e., $P_{\mathrm{b}} < 0$) during regenerative braking or by delivering the FC power to the battery.
- The total power P_{tot} is then delivered by the DC bus to the electric motor, i.e., the actuator, usually through an AC/DC interface. The total delivered power P_{tot} is a model output, i.e.,

$$y(t) = P_{\mathrm{tot}}(t). \tag{6.49}$$

- The NA-MPC controller will act on the DC bus, defining the power allocation between battery and FC.

Therefore, in order to construct the model for the FCHEV power system, we have to first derive three sub-models, for the battery, the FC, and the DC bus, respectively.

Battery Model

In the battery model, the state and input are $x = \zeta$ and $u = P_{\mathrm{b}}$, respectively, where ζ is the battery state of charge (SOC) and P_{b} is the power requested to the battery. The SOC is defined as the ratio between the battery charge Q_{b} and the nominal battery capacity Q_{nom}, i.e.,

$$\zeta = \frac{Q_{\mathrm{b}}}{Q_{\mathrm{nom}}} \in [0, 1]. \tag{6.50}$$

Differentiating both sides, we can relate the SOC rate $\dot{\zeta}$ with the battery current I_b as follows:

$$\dot{\zeta} = -\frac{\eta_{\mathrm{b},1}(P_\mathrm{b})\, I_\mathrm{b}}{Q_\mathrm{nom}}, \quad \eta_{\mathrm{b},1}(P_\mathrm{b}) = \begin{cases} \frac{1}{\eta_\mathrm{b}^\mathrm{coul}} & \text{if } P_\mathrm{b} > 0 \quad \text{(discharge)}, \\ \eta_\mathrm{b}^\mathrm{coul} & \text{if } P_\mathrm{b} < 0 \quad \text{(charge)}, \end{cases} \tag{6.51}$$

where $\eta_\mathrm{b}^\mathrm{coul} \in (0,1)$ is the Coulombic efficiency, quantifying the fraction of current that is lost during battery charge and discharge [115].

The battery is typically made of N_b battery cells connected in series. Each cell is modeled as an ideal voltage source $V_\mathrm{b}^{\mathrm{oc,s}}$ with a series output resistance $R_\mathrm{b}^{\mathrm{o,s}}$; the latter quantities can be combined as $V_\mathrm{b}^\mathrm{oc} = N_\mathrm{b} V_\mathrm{b}^{\mathrm{oc,s}}$ and $R_\mathrm{b}^\mathrm{o} = N_\mathrm{b} R_\mathrm{b}^{\mathrm{o,s}}$, respectively.

In real batteries, V_b^oc and R_b^o are functions of the battery SOC. Such a dependence on ζ can be derived from experimental data (taken from, e.g., [115]). The functions $V_\mathrm{b}^\mathrm{oc}(\zeta)$ and $R_\mathrm{b}^\mathrm{o}(\zeta)$ are then constructed from this data by means of piecewise polynomial fitting.

The power generated by the battery ($P_\mathrm{b}^\mathrm{gen}$) is then equal to

$$P_\mathrm{b}^\mathrm{gen} = V_\mathrm{b}^\mathrm{oc} I_\mathrm{b} - R_\mathrm{b}^\mathrm{o} I_\mathrm{b}^2. \tag{6.52}$$

To compute the actual delivered power P_b, we have to account for the losses due to the power converters interfacing the battery with the DC bus. Therefore, we set

$$P_\mathrm{b}^\mathrm{gen} = \eta_{\mathrm{b},2}(P_\mathrm{b}) P_\mathrm{b}, \quad \eta_{\mathrm{b},2}(P_\mathrm{b}) = \begin{cases} \frac{1}{\eta_\mathrm{b}^\mathrm{conv}} & \text{if } P_\mathrm{b} > 0, \\ \eta_\mathrm{b}^\mathrm{conv} & \text{if } P_\mathrm{b} < 0, \end{cases} \tag{6.53}$$

where $\eta_\mathrm{b}^\mathrm{conv} \in (0,1)$ is the efficiency of the battery power converters. The efficiency $\eta_{\mathrm{b},2}$ accounts for the bidirectional flow of the battery power during charge and discharge.

Solving Eq. (6.52) for I_b and adding Eq. (6.53) yields

$$I_\mathrm{b} = \frac{1}{2R_\mathrm{b}^\mathrm{o}} \Big(V_\mathrm{b}^\mathrm{oc} - \sqrt{V_\mathrm{b}^{\mathrm{oc}2} - 4R_\mathrm{b}^\mathrm{o}\, \eta_{\mathrm{b},2}(P_\mathrm{b})\, P_\mathrm{b}} \Big). \tag{6.54}$$

The battery model is then the union of Eqs. (6.51) and (6.54), i.e.,

$$\dot{\zeta} = f_{\mathrm{b},c}(\zeta, P_\mathrm{b}) = -\frac{\eta_{\mathrm{b},1}(P_\mathrm{b})}{Q_\mathrm{nom}} \frac{1}{2R_\mathrm{b}^\mathrm{o}(\zeta)} \Big(V_\mathrm{b}^\mathrm{oc}(\zeta) - \sqrt{V_\mathrm{b}^\mathrm{oc}(\zeta)^2 - 4R_\mathrm{b}^\mathrm{o}(\zeta)\, \eta_{\mathrm{b},2}(P_\mathrm{b})\, P_\mathrm{b}} \Big). \tag{6.55}$$

The battery model (6.55) forces an upper bound on P_b, i.e.,

$$P_\mathrm{b} \le \eta_\mathrm{b}^\mathrm{conv} \frac{V_\mathrm{b}^\mathrm{oc}(\zeta)^2}{4R_\mathrm{b}^\mathrm{o}(\zeta)} = P_\mathrm{b,ub}^\mathrm{model}(\zeta). \tag{6.56}$$

Real battery specifications also provide a range of admissible values for P_b and ζ, given by the bounds $P_\mathrm{b,lb}^\mathrm{spec}$, $P_\mathrm{b,ub}^\mathrm{spec}$ and ζ_lb, ζ_ub, respectively. Typically,

$$P_\mathrm{b,ub}^\mathrm{spec} < \min_{\zeta \in [\zeta_\mathrm{lb}, \zeta_\mathrm{ub}]} P_\mathrm{b,ub}^\mathrm{model}(\zeta),$$

so that the upper bound on P_b is independent of ζ and is feasible for any $\zeta \in [\zeta_\mathrm{lb}, \zeta_\mathrm{ub}]$. Therefore, ζ and P_b are bounded as follows:

$$\zeta \in [\zeta_\mathrm{lb}, \zeta_\mathrm{ub}], \quad P_\mathrm{b} \in [P_\mathrm{b,lb}^\mathrm{spec}, P_\mathrm{b,ub}^\mathrm{spec}] = [P_\mathrm{b,lb}, P_\mathrm{b,ub}]. \tag{6.57}$$

Fuel Cell Model

In the fuel cell model, the state and input are $x = m_{\mathrm{h}}$ and $u = P_{\mathrm{fc}}$, respectively, where m_{h} is the hydrogen mass and P_{fc} is the power requested to the FC. In the model, proton-exchange membrane (PEM) fuel cell behaviour is assumed [67].

The fuel cell is made of N_{fc} smaller cells connected in series as a stack. The electrical behavior of each smaller cell is defined by its characteristic curve, relating the cell voltage $V_{\mathrm{fc}}^{\mathrm{s}}$ to the cell current density $i_{\mathrm{fc}}^{\mathrm{s}}$ [67, 164], defined as the cell current per active area, i.e., $i_{\mathrm{fc}}^{\mathrm{s}} = \frac{I_{\mathrm{fc}}^{\mathrm{s}}}{A_{\mathrm{fc}}^{\mathrm{s}}}$.

Within a wide range of current densities, such a curve can be conveniently fitted by a linear function [67], i.e.,

$$V_{\mathrm{fc}}^{\mathrm{s}} \approx V_{\mathrm{fc}}^{\mathrm{oc,s}} - r_{\mathrm{fc}}^{\mathrm{o,s}} i_{\mathrm{fc}}^{\mathrm{s}} = V_{\mathrm{fc}}^{\mathrm{oc,s}} - R_{\mathrm{fc}}^{\mathrm{o,s}} I_{\mathrm{fc}}^{\mathrm{s}}. \tag{6.58}$$

The fuel cell stack voltage and current are then equal to $V_{\mathrm{fc}} = N_{\mathrm{fc}} V_{\mathrm{fc}}^{\mathrm{s}}$ and $I_{\mathrm{fc}} = I_{\mathrm{fc}}^{\mathrm{s}}$, respectively. We also define $V_{\mathrm{fc}}^{\mathrm{oc}} = N_{\mathrm{fc}} V_{\mathrm{fc}}^{\mathrm{oc,s}}$ and $R_{\mathrm{fc}}^{\mathrm{o}} = N_{\mathrm{fc}} R_{\mathrm{fc}}^{\mathrm{o,s}}$.

The power generated by the fuel cell ($P_{\mathrm{fc}}^{\mathrm{gen}}$) is equal to the ideal power $P_{\mathrm{fc}}^{\mathrm{id}} = V_{\mathrm{fc}} I_{\mathrm{fc}}$, reduced by a quantity $P_{\mathrm{fc}}^{\mathrm{aux}}$, which is the power needed to keep the fuel cell on [67]. It can be shown that $P_{\mathrm{fc}}^{\mathrm{aux}}$ depends quadratically on I_{fc} [67] and its expression can be further simplified with a linear relation [66], i.e., $P_{\mathrm{fc}}^{\mathrm{aux}} \approx P_{\mathrm{fc}}^{\mathrm{aux}\prime} + V_{\mathrm{fc}}^{\mathrm{aux}} I_{\mathrm{fc}}$. Therefore, we have that

$$P_{\mathrm{fc}}^{\mathrm{gen}} = (V_{\mathrm{fc}} - V_{\mathrm{fc}}^{\mathrm{aux}}) I_{\mathrm{fc}} - P_{\mathrm{fc}}^{\mathrm{aux}\prime}. \tag{6.59}$$

As done for the battery, to compute the actual delivered power P_{fc}, we have to account for the losses due to the power converters interfacing the fuel cell with the DC bus. Thus,

$$P_{\mathrm{fc}}^{\mathrm{gen}} = \frac{1}{\eta_{\mathrm{fc}}^{\mathrm{conv}}} P_{\mathrm{fc}}, \tag{6.60}$$

where $\eta_{\mathrm{fc}}^{\mathrm{conv}} \in (0, 1)$ is the efficiency of the fuel cell power converters. Combining Eqs. (6.58)-(6.60) and solving for I_{fc} yields

$$I_{\mathrm{fc}} = \frac{1}{2R_{\mathrm{fc}}^{\mathrm{o}}} \Big(V_{\mathrm{fc}}^{\mathrm{oc}} - V_{\mathrm{fc}}^{\mathrm{aux}} - \sqrt{(V_{\mathrm{fc}}^{\mathrm{oc}} - V_{\mathrm{fc}}^{\mathrm{aux}})^2 - 4R_{\mathrm{fc}}^{\mathrm{o}} (\tfrac{1}{\eta_{\mathrm{fc}}^{\mathrm{conv}}} P_{\mathrm{fc}} + P_{\mathrm{fc}}^{\mathrm{aux}\prime})} \Big). \tag{6.61}$$

Experimental data on both the fuel cell characteristic curve and auxiliary power are taken from [67].

Finally, we relate the hydrogen mass rate $\dot{m}_{\mathrm{h}}$ with the fuel cell current I_{fc} recalling the oxidation semi-reaction taking place at the fuel cell anode, i.e., $\mathrm{H_2} \rightarrow 2\mathrm{H^+} + 2\mathrm{e^-}$. The number of hydrogen molecules that dissociate every second equals $\frac{I_{\mathrm{fc}}}{n_{\mathrm{e}} q}$, where $n_{\mathrm{e}} = 2$ (number of electrons released) and q is the elementary charge. The hydrogen rate $\dot{m}_{\mathrm{h}}$ of the fuel cell stack is then equal to

$$\dot{m}_{\mathrm{h}} = -N_{\mathrm{fc}} \frac{I_{\mathrm{fc}}\, M_{\mathrm{h}}}{n_{\mathrm{e}}\, q\, N_{\mathrm{A}}}, \tag{6.62}$$

being M_{h} the $\mathrm{H_2}$ molar mass and N_{A} the Avogadro's number.

The fuel cell model is then the union of Eqs. (6.61) and (6.62), i.e.,

$$\dot{m}_{\mathrm{h}} = f_{\mathrm{fc},c}(P_{\mathrm{fc}}) = \; - \frac{N_{\mathrm{fc}}\, M_{\mathrm{h}}}{n_{\mathrm{e}}\, q\, N_{\mathrm{A}}} \frac{1}{2R_{\mathrm{fc}}^{\mathrm{o}}} \Big(V_{\mathrm{fc}}^{\mathrm{oc}} - V_{\mathrm{fc}}^{\mathrm{aux}}$$

$$- \sqrt{(V_{\mathrm{fc}}^{\mathrm{oc}} - V_{\mathrm{fc}}^{\mathrm{aux}})^2 - 4R_{\mathrm{fc}}^{\mathrm{o}}(\tfrac{1}{\eta_{\mathrm{fc}}^{\mathrm{conv}}} P_{\mathrm{fc}} + P_{\mathrm{fc}}^{\mathrm{aux}\prime})} \Big). \tag{6.63}$$

The FC model (6.63) forces an upper bound on P_{fc}, i.e.,

$$P_{\mathrm{fc}} \leq \eta_{\mathrm{fc}}^{\mathrm{conv}} \left(\frac{(V_{\mathrm{fc}}^{\mathrm{oc}} - V_{\mathrm{fc}}^{\mathrm{aux}})^2}{4R_{\mathrm{fc}}^{\mathrm{o}}} - P_{\mathrm{fc}}^{\mathrm{aux}\prime} \right) = P_{\mathrm{fc,ub}}^{\mathrm{model}}. \tag{6.64}$$

Real fuel cell specifications also provide an upper bound on P_{fc}, denoted by $P_{\mathrm{fc,ub}}^{\mathrm{spec}}$, satisfying $P_{\mathrm{fc,ub}}^{\mathrm{spec}} < P_{\mathrm{fc,ub}}^{\mathrm{model}}$. The hydrogen mass is bounded by $m_{\mathrm{h,lb}}$ and $m_{\mathrm{h,ub}}$, where $m_{\mathrm{h,ub}}$ is the hydrogen tank capacity. Thus, m_{h} and P_{fc} are bounded as follows:

$$m_{\mathrm{h}} \in [m_{\mathrm{h,lb}}, m_{\mathrm{h,ub}}], \quad P_{\mathrm{fc}} \in [0, P_{\mathrm{fc,ub}}^{\mathrm{spec}}] = [P_{\mathrm{fc,lb}}, P_{\mathrm{fc,ub}}]. \tag{6.65}$$

DC Bus Model

In the DC bus model, we have to relate the requested powers P_{b} and P_{fc} with the total delivered power P_{tot}. This can be done by means of the following static equation:

$$P_{\mathrm{tot}} = g_{P_{\mathrm{tot}}}(\zeta, m_{\mathrm{h}}, P_{\mathrm{b}}, P_{\mathrm{fc}}) = \sigma(\zeta, P_{\mathrm{b}}) + \sigma(m_{\mathrm{h}}, P_{\mathrm{fc}}), \tag{6.66}$$

where the function $\sigma(x, u)$, considering $x \in \{\zeta, m_{\mathrm{h}}\}$ and $u \in \{P_{\mathrm{b}}, P_{\mathrm{fc}}\}$, is given by

$$\sigma(x,u) = \begin{cases} 0 & \text{if } \; x < x_{\mathrm{lb}}, \; u > 0 \\ & \text{or } \; x > x_{\mathrm{ub}}, \; u < 0, \\ u & \text{otherwise.} \end{cases} \tag{6.67}$$

Eq. (6.67) states that a source cannot deliver power if its supply is depleted and the requested power is positive (i.e., $x < x_{\mathrm{lb}}$, $u > 0$), or the supply is full and the requested power is negative (i.e., $x > x_{\mathrm{ub}}$, $u < 0$).

FCHEV Power System Model

The FCHEV power system model is the union of the battery model in Eq. (6.55), the FC model in Eq. (6.63), and the DC bus model in Eq. (6.66), as follows:

$$\dot{x} = f_c(x,u) = \begin{bmatrix} f_{\mathrm{b},c}(\zeta, P_{\mathrm{b}}) \\ f_{\mathrm{fc},c}(P_{\mathrm{fc}}) \end{bmatrix}, \tag{6.68a}$$

$$y = \begin{bmatrix} x \\ P_{\mathrm{tot}} \end{bmatrix} = g(x,u) = \begin{bmatrix} \zeta \\ m_{\mathrm{h}} \\ g_{P_{\mathrm{tot}}}(\zeta, m_{\mathrm{h}}, P_{\mathrm{b}}, P_{\mathrm{fc}}) \end{bmatrix}. \tag{6.68b}$$

The model (6.68) is subject to linear constraints, defined by the sets

$$\mathcal{X} = \left\{ (\zeta, m_{\mathrm{h}}) \in \mathbb{R}^2 : \begin{bmatrix} \zeta_{\mathrm{lb}} \\ m_{\mathrm{h,lb}} \end{bmatrix} \leq \begin{bmatrix} \zeta \\ m_{\mathrm{h}} \end{bmatrix} \leq \begin{bmatrix} \zeta_{\mathrm{ub}} \\ m_{\mathrm{h,ub}} \end{bmatrix} \right\}, \tag{6.69a}$$

$$\mathcal{U} = \left\{ (P_{\mathrm{b}}, P_{\mathrm{fc}}) \in \mathbb{R}^2 : \begin{bmatrix} P_{\mathrm{b,lb}} \\ P_{\mathrm{fc,lb}} \end{bmatrix} \leq \begin{bmatrix} P_{\mathrm{b}} \\ P_{\mathrm{fc}} \end{bmatrix} \leq \begin{bmatrix} P_{\mathrm{b,ub}} \\ P_{\mathrm{fc,ub}} \end{bmatrix} \right\}, \tag{6.69b}$$

$$\mathcal{Y} = \{ P_{\mathrm{tot}} \in \mathbb{R} : P_{\mathrm{b,lb}} + P_{\mathrm{fc,lb}} \leq P_{\mathrm{tot}} \leq P_{\mathrm{b,ub}} + P_{\mathrm{fc,ub}} \}, \tag{6.69c}$$

given by Eqs. (6.57), (6.65), and matching Eq. (6.2).

Finally, we note that the FCHEV power system model (6.68) satisfies Assumptions 6.1 and 6.3.

Remark 6.8

For each source $x \in \{\zeta, m_\mathrm{h}\}$ and $u \in \{P_\mathrm{b}, P_\mathrm{fc}\}$, if $u = 0$ (i.e., null requested power), then $\dot{x} = 0$, $\forall\, x \in [x_\mathrm{lb}, x_\mathrm{ub}]$, i.e., the supply level does not change.

Remark 6.9

By Remark 6.8 and Eq. (6.67), for any requested power $u(t) \in \mathcal{U}$, the evolution over time of the supplies is such that $x(t) \in \mathcal{X}$, $\forall\, t \geq 0$.

Then, under the constraint $x \in \mathcal{X}$, the DC bus model in Eq. (6.66) simplifies as

$$P_\mathrm{tot} = P_\mathrm{b} + P_\mathrm{fc}. \tag{6.70}$$

6.5.2. Longitudinal Vehicle Model for Requested Power Calculation

The FCHEV power system model (6.68) provides as output the total delivered power P_tot, as function of the requested powers P_b and P_fc.

In power management applications, the total delivered power $P_\mathrm{tot}(t)$ is typically required to track a given reference power, corresponding to the vehicle requested power $P_\mathrm{req}(t)$. Such a requested power is, in general, computed from a *driving cycle*, i.e., the sequence of vehicle velocity values $v(t)$ over time. To compute the requested power $P_\mathrm{req}(t)$ from the driving cycle $v(t)$, we need a longitudinal model of the vehicle dynamics, like the following (which is a modified version of the model in Eq. (5.28)):

$$P_\mathrm{req}(t) = F_\mathrm{tot}(t)\, v(t), \tag{6.71a}$$

$$\begin{aligned} F_\mathrm{tot}(t) &= F_\mathrm{vis}(t) + F_\mathrm{roll}(t) + F_\mathrm{slope}(t) + Ma(t) \\ &= \frac{1}{2}\rho A_\mathrm{f} C_\mathrm{d} v(t)|v(t)| + C_\mathrm{r} Mg\cos(\theta(t))\,\mathrm{sign}(v(t)) + Mg\sin(\theta(t)) + Ma(t), \end{aligned} \tag{6.71b}$$

where M is the vehicle mass, g the gravitational acceleration, $v(t)$ the vehicle velocity from the given driving cycle, $a(t) = \dot{v}(t)$ the vehicle acceleration, ρ the air density, A_f the vehicle frontal area, C_d the viscous friction coefficient, C_r the rolling friction coefficient, and $\theta(t)$ the road slope, expressed as a function of time[2].

The model (6.71) can be sampled with discrete time step T_s, allowing to employ a DT driving cycle $v_k = v(kT_s)$, $k \geq 0$.

6.5.3. NA-MPC Strategy For FCHEVs Power Management

In this section, we tailor the NA-MPC strategy for the case study of FCHEV power management. Specifically, NA-MPC should be able to address the following three main control tasks:

1) accurate requested power tracking;
2) combined supplies saving;
3) selective supplies saving (i.e., either battery or FC).

[2]In the case θ is given as a function of the vehicle longitudinal position x, i.e., $\theta(x)$, then it is sufficient to compute $x(t) = \int_0^t v(\tau)d\tau$ from the driving cycle, and replace $\theta(x(t))$, obtaining θ as a function of time.

Neural Plant Model

Data for training the CT neural black-box plant model (6.18) is "virtually" collected from the white-box model of the FCHEV power system (6.68), as described in Section 6.3.2.

In this sense, the high-fidelity model (6.68) has the purpose of emulating a real-world FCHEV, onto which real measurement data would be collected.

Neural MPC Controller

From Section 6.3.3, the neural MPC controller $\mathcal{N}_\pi$ (6.28) for FCHEV power management has the following inputs:

1) the current state $x_k = [\zeta_k,\ m_{\mathrm{h},k}]^\top$;
2) the reference output $y_{r,k} = [\zeta_r,\ m_{\mathrm{h},r},\ P_{\mathrm{tot},r,k}]^\top$;
3) the MPC weights w_k, given by

$$\begin{aligned} &Q_k = \mathrm{diag}([w_{\zeta,k},\ w_{m_\mathrm{h},k},\ w_{P_\mathrm{tot},k}]^\top), \quad Q_{\Delta,k} = \mathrm{diag}([w_{\Delta,\zeta,k},\ w_{\Delta,m_\mathrm{h},k},\ 0]^\top), \\ &R = \mathrm{diag}([w_{P_\mathrm{b}},\ w_{P_\mathrm{fc}}]^\top), \quad R_\Delta = \mathrm{diag}([w_{\Delta,P_\mathrm{b}},\ w_{\Delta,P_\mathrm{fc}}]^\top). \end{aligned} \tag{6.72}$$

In Eq. (6.72), the weights that do not effectively contribute in achieving the control tasks are set to constant values.

Concerning the reference output $y_{r,k}$:

- To encode the supplies saving task, the references for the supplies, i.e., ζ_r and $m_{\mathrm{h},r}$, are set to their respective upper bounds,

$$\zeta_r = \zeta_\mathrm{ub}, \quad m_{\mathrm{h},r} = m_{\mathrm{h,ub}}.$$

- To encode the power tracking task, the reference for the total delivered power, i.e., $P_{\mathrm{tot},r,k}$, is set to the current requested power,

$$P_{\mathrm{tot},r,k} = P_{\mathrm{req},k}.$$

The neural MPC controller has then 8 inputs, i.e.,

$$u_k = \begin{bmatrix} P_{\mathrm{b},k} \\ P_{\mathrm{fc},k} \end{bmatrix} = \mathcal{N}_\pi(\Pi_k) = \mathcal{N}_\pi(\zeta_k, m_{\mathrm{h},k}, P_{\mathrm{req},k}, w_k). \tag{6.73}$$

To ensure a precise input constraint satisfaction, the output of the neural MPC controller (6.73) is saturated within the bounds u_lb and u_ub in Eqs. (6.57) and (6.65), i.e.,

$$u_k^* = \mathrm{sat}_{[u_\mathrm{lb},u_\mathrm{ub}]}(\mathcal{N}_\pi(\Pi_k)). \tag{6.74}$$

Performance-Metric Functions for Online Tuning

As detailed in Section 6.4, we define $N_\mathrm{obj} = 3$ performance-metric functions $p_i(\hat{\boldsymbol{y}})$, $i = 1, \ldots, N_\mathrm{obj}$, to quantify the satisfaction of the three power management control objectives:

1) Requested power tracking:

$$p_1(\hat{\boldsymbol{y}}) = \begin{cases} m\left(1 - \frac{\hat{e}_k}{e_\mathrm{ub}}\right)^2 & \text{if } \hat{e}_k \le e_\mathrm{ub}, \\ \frac{2M}{1+\exp(-K(\hat{e}_k - e_\mathrm{ub}))} - M & \text{if } \hat{e}_k > e_\mathrm{ub}, \end{cases} \tag{6.75a}$$

$$\hat{e}_k = \frac{\max_{i\in\{k-N_{\text{tuning}},\dots,k-1\}} |\hat{P}_{\text{tot},i} - P_{\text{req},i}|}{\max(|P_{\text{req,lb}}|, |P_{\text{req,ub}}|)}, \tag{6.75b}$$

where $P_{\text{req,ub}} = P_{\text{b,ub}} + P_{\text{fc,ub}}$ and $P_{\text{req,lb}} = P_{\text{b,lb}} + P_{\text{fc,ub}}$.

2) Battery saving:

$$p_2(\hat{\boldsymbol{y}}) = \begin{cases} M & \text{if } \hat{\zeta}_{k-1} < \zeta_{\text{lb}}, \\ M\left(\frac{\hat{\zeta}_{k-1}-\zeta_{\text{ub}}}{\zeta_{\text{lb}}-\zeta_{\text{ub}}}\right)^2 & \text{if } \hat{\zeta}_{k-1} \in [\zeta_{\text{lb}}, \zeta_{\text{ub}}], \\ 0 & \text{if } \hat{\zeta}_{k-1} > \zeta_{\text{ub}}. \end{cases} \tag{6.75c}$$

3) Fuel cell saving:

$$p_3(\hat{\boldsymbol{y}}) = \begin{cases} M & \text{if } \hat{m}_{\text{h},k-1} < m_{\text{h,lb}}, \\ M\left(\frac{\hat{m}_{\text{h},k-1}-m_{\text{h,ub}}}{m_{\text{h,lb}}-m_{\text{h,ub}}}\right)^2 & \text{if } \hat{m}_{\text{h},k-1} \in [m_{\text{h,lb}}, m_{\text{h,ub}}], \\ 0 & \text{if } \hat{m}_{\text{h},k-1} > m_{\text{h,ub}}. \end{cases} \tag{6.75d}$$

In Eq. (6.75), m, M, K, and e_{ub} are design parameters; $\hat{\boldsymbol{y}} = (\hat{y}_i)_{i=k-N_{\text{tuning}}}^{k-1}$, with $\hat{y}_i = [\hat{\zeta}_i,\ \hat{m}_{\text{h},i},\ \hat{P}_{\text{tot},i}]^\top$, is the simulated closed-loop output trajectory, computed according to Eq. (6.30).

Concerning the performance-metric functions:

- p_1 attains its minimal value when the normalized power tracking error (6.75b) is close to e_{ub} from below (thus, e_{ub} acts a "soft" upper bound on the maximum tracking error);
- p_2 and p_3 penalize the reduction of supplies with respect to their upper bounds.

Therefore, p_1 defines the trade-off between power tracking accuracy and supplies saving; p_2 and p_3 set the consumption priority between battery and fuel cell.

Characterization of Closed-Loop Trajectories and Stability

In this case study, we focus on the problem of power management, which is fundamentally distinct from conventional regulation or tracking.

Given a reference power $P_{\text{tot},r}(t)$ (typically corresponding to the vehicle power request $P_{\text{req}}(t)$), our objective is to determine the optimal allocation of the delivered powers $(u_i)_{i=1}^{n_u}$, ensuring that $P_{\text{tot}}(t) = \sum_{i=1}^{n_u} u_i(t) \to P_{\text{tot},r}(t)$. Therefore, this task is not formulated as a state regulation or tracking problem, since our aim is not to steer the closed-loop state trajectories $x(t)$ (i.e., the supplies) towards an equilibrium state or trajectory of the system; rather, the supplies $x(t)$ are permitted to evolve organically under the influence of the delivered powers $u_i(t)$, calculated through the optimal allocation.

In this regard, while the supplies $x(t)$ are not requested to track a given trajectory (or regulate towards an equilibrium), their time evolution contributes in the evaluation of the optimal power allocation, as detailed in Section 6.5.3, p. 212.

For the above reasons, classic stability arguments are not frequently employed within the scope of power management.

In terms of characterization of the closed-loop trajectories, the most relevant property that we can highlight for controlled power systems is that, in general, as reported in Remark 6.9, the closed-loop supply trajectories $x(t)$ are ensured to remain always bounded within the set $\mathcal{X}$ for any control input $u(t) \in \mathcal{U}$.

6.6. Simulations and Results

In this section, we validate the NA-MPC strategy through simulations. The white-box model of the FCHEV power system (6.68) is employed as the CT plant. The plant is controlled by the neural MPC controller (6.73), which is adapted in real time by the online metaheuristic tuning strategy described in Section 6.4.

The full control system scheme is depicted in Figure 6.1.

Simulations encompass the following aspects:

- capability of NA-MPC to deliver the requested control tasks;
- efficacy of the online tuning strategy;
- comparison of NA-MPC with state-of-the-art techniques for FCHEV power management;
- NA-MPC execution time and real-time feasibility.

6.6.1. Implementation Details

Simulations

Simulations are performed in MATLAB® (ver. 2023b), on a machine powered by a 13$^{\text{th}}$ Gen Intel Core™ i7 CPU at 1.7 GHz, with 16 GB of RAM.

The NA-MPC strategy is implemented in MATLAB and Python™ 3. The QP-MPC optimal control problem (6.10) is formulated with YALMIP [96] and solved with the QP interior-point solver MOSEK (ver. 10.1) [108]. The FNNs for the neural plant model (6.18) and the neural MPC controller (6.73) are constructed and trained using PyTorch.

Data

Relevant data shared by all simulations is reported in the following:

- General data: $T_s = 1$ s, simulation time $T = 1000$ s, $N = \lfloor \frac{T}{T_s} \rfloor = 1000$.
- White-box model of the CT plant (FCHEV power system, Eq. (6.68)): $x_0 = [\zeta_0, m_{\text{h},0}]^\top = [0.6,\ 900\ \text{g}]^\top$ or $[0.5,\ 700\ \text{g}]^\top$, $N_\text{b} = 200$, $Q_\text{nom} = 60$ A h, $\eta_\text{b}^\text{coul} = 0.95$, $\eta_\text{b}^\text{conv} = 0.97$, $P_\text{b,lb} = -290$ kW, $P_\text{b,ub} = 290$ kW, $\zeta_\text{lb} = 0.2$, $\zeta_\text{ub} = 0.8$, $N_\text{fc} = 500$, $V_\text{fc}^\text{oc,s} = 0.812$ V [67], $R_\text{fc}^\text{o,s} = 1.21$ mΩ [67], $P_\text{fc}^{\text{aux}\prime} = 100$ W [67], $V_\text{fc}^\text{aux} = 50$ mV [67], $\eta_\text{fc}^\text{conv} = 0.97$, $P_\text{fc,ub} = 45$ kW, $m_\text{h,lb} = 100$ g, $m_\text{h,ub} = 1000$ g.
- Longitudinal vehicle model (Eq. (6.71)): $m = 5 \times 10^3$ kg, $A_\text{f} = 8$ m^2, $C_\text{d} = 5 \times 10^{-1}$, $C_\text{r} = 10^{-2}$, $\theta \in [-3,\ 9]$ deg.
- Driving cycles:
  - California Unified Cycle (UC/LA-29, Figure 6.3a);
  - City Suburban Heavy Vehicle Cycle (CSHVC, Figure 6.3b);
  - EPA Federal Test Procedure (FTP-72/UDDS/LA-4, Figure 6.3c).
- Neural black-box model of the CT plant (Eq. (6.18)): $n_\text{in} = 4$; $n_\text{out} = 2$; number of layers $L = 3$; neurons for each layer $(N_l)_{l=1}^{L+1} = (4, 10, 10, 2)$; $N_\text{data} = 50 \times 10^3$; $N_\text{train} = 47.5 \times 10^3$; $N_\text{val} = 2.5 \times 10^3$; $N_\text{train}^\text{batch} = 47.5 \times 10^2$; $N_\text{val}^\text{batch} = 2.5 \times 10^2$; errors on collected data $\overline{n}_u = 5\%$, $\overline{n}_y = 5\%$ (see Section 6.3.2).

- Activation functions: rectified linear unit (ReLU(z)) for hidden layers; linear function (z) for the output layer.

- MPC: $N_p = 10$.
- Neural MPC controller (Eq. (6.73)): $n_{\text{in}} = 8$, $n_{\text{out}} = 2$, $L = 4$; $(N_l)_{l=1}^{L+1} = (8, 100, 100, 100, 2)$, $N_{\text{data}} = 500 \times 10^3$, $N_{\text{train}} = 475 \times 10^3$, $N_{\text{val}} = 25 \times 10^3$, $N_{\text{train}}^{\text{batch}} = 47.5 \times 10^2$, $N_{\text{val}}^{\text{batch}} = 2.5 \times 10^2$.
  - Activation functions: rectified linear unit (ReLU(z)) for hidden layers; linear function (z) for the output layer.
- Performance-metric functions (Eq. (6.75)): $m = 2.5$, $M = 10$, $K = 2$.
- Online metaheuristic tuning: $w_{\text{lb}} = [1, 1, 1, 1, 1]^\top$, $w_{\text{ub}} = [10, 10, 10, 100, 100]^\top$.
  - Particle Swarm Optimization (PSO): m linearly decreasing from 0.9 to 0.2, $N_{\text{part}} = 10$, $N_{\text{PSO}} = 10$, $c_{\text{i}} = 2$, $c_{\text{s}} = 2$; absorbing bounds (see Eq. (6.38)).
  - Genetic Algorithm (GA): $N_{\text{ind}} = 10$, $N_{\text{GA}} = 10$, $c_{\text{s}} = 1$, $c_{\text{r}} = 1$.

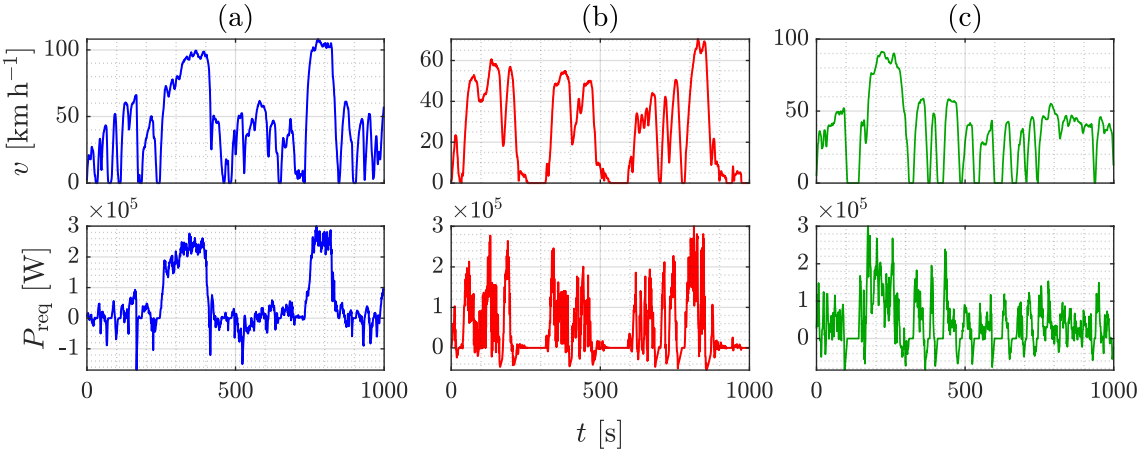


Figure 6.3. Driving cycles (v) and corresponding requested power (P_{req}). **(a)** California Unified Cycle (UC/LA-29, ——). **(b)** City Suburban Heavy Vehicle Cycle (CSHVC, ——). **(c)** EPA Federal Test Procedure (FTP-72/UDDS/LA-4, ——).

Note that all driving cycles in Figure 6.3 are standard, based on real-world vehicle operation, and widely adopted for validation and testing in automotive scenarios [110].

6.6.2. NA-MPC FNNs Training and Validation

In this section, we report training and validation results for the two FNNs employed by the NA-MPC strategy: the neural plant model ($\mathcal{N}_{f_c}, \mathcal{N}_g$) in Eq. (6.18) and the neural MPC controller $\mathcal{N}_\pi$ in Eq. (6.73).

Neural Plant Model

Recalling that the CT plant is represented by the white-box model of the FCHEV power system in Eq. (6.68), the output function g is known exactly. Thus, we only need the FNN $\mathcal{N}_{f_c}$ for the state dynamics function f_c.

The input and output datasets $\mathcal{D}_{\text{in}}$ and $\mathcal{D}_{\text{out}}$ are constructed as described in Section 6.3.2. Specifically, we ensure that $\mathcal{D}_{\text{in}}$ uniformly spans the set $\mathcal{Z} = \mathcal{X} \times \mathcal{U}$, defined by the linear

state and input constraints in Eqs. (6.57) and (6.65). A total of N_{data} data points are collected, of which N_{train} are used for training and N_{val} for validation. These data points are then subdivided into batches, with $N_{\text{train}}^{\text{batch}}$ and $N_{\text{val}}^{\text{batch}}$ data points each, respectively.

The FNN is trained with the Adam optimization algorithm [78], employing the following loss function:

$$L = \frac{1}{N_{\text{batch}}} \sum_{i=1}^{N_{\text{batch}}} \ell(z_i, \hat{z}_i) + \alpha \|p\|_2^2. \tag{6.76}$$

In the loss function (6.76), z and $\hat{z}$ are the target and predicted outputs of the FNN, respectively, ℓ is the L_2 loss, i.e., $\ell(z, \hat{z}) = (z - \hat{z})^2$, and p are the learnable parameters of the FNN, on which L_2 regularization is applied.

Neural MPC Controller

The input and output datasets are constructed as described in Section 6.3.3. Specifically, we ensure that $\mathcal{D}_{\text{in}}$ uniformly spans the feasible set of parameters Π in Eq. (6.73). The datasets are split for training/validation and are subdivided into batches.

The FNN is trained with the Adam optimization algorithm, employing the following loss function:

$$\begin{aligned} L = \frac{1}{N_{\text{batch}}} \sum_{i=1}^{N_{\text{batch}}} \Big(& \lambda(\ell(P_{\text{b},i}, \hat{P}_{\text{b},i}) + \ell(P_{\text{fc},i}, \hat{P}_{\text{fc},i})) \\ & + (1-\lambda)\, \ell(P_{\text{b},i} + P_{\text{fc},i}, \hat{P}_{\text{b},i} + \hat{P}_{\text{fc},i}) \Big) + \alpha \|p\|_2^2, \end{aligned} \tag{6.77}$$

where $\lambda \in (0, 1)$ and ℓ is the L_1 loss, i.e., $\ell(z, \hat{z}) = |z - \hat{z}|$. The loss function (6.77) penalizes both the error on each output of the neural MPC controller (i.e., P_{b} and P_{fc}) and the error on the sum of the two outputs (which corresponds to the total delivered power $P_{\text{tot}} = P_{\text{b}} + P_{\text{fc}}$).

Training and Validation Results

For both FNNs $\mathcal{N}_{f_c}$ and $\mathcal{N}_{\pi}$, input and output data points are normalized, so to have zero mean and variance equal to 1. Also, early stopping regularization is employed to prevent overfitting; cross-validation is not employed.

Figures 6.4a and 6.4b report the training and validation losses of $\mathcal{N}_{f_c}$ and $\mathcal{N}_{\pi}$, respectively, over the training epochs. In both cases, we observe a good convergence behavior and no overfitting.

To further assess the FNNs emulation capability, we evaluate, for both $\mathcal{N}_{f_c}$ and $\mathcal{N}_{\pi}$, and for each output neuron, the normalized errors

$$(e_{f_c}(z))_i = \frac{|(\mathcal{N}_{f_c}(z) - f_c(z))_i|}{\max_{z \in \mathcal{D}_{\text{in}}} |(f_c(z))_i|}, \quad i = 1, \dots, n_{\text{out}}, \tag{6.78a}$$

$$(e_{\pi}(z))_i = \frac{|(\mathcal{N}_{\pi}(z) - \pi(z))_i|}{\max_{z \in \mathcal{D}_{\text{in}}} |(\pi(z))_i|}, \quad i = 1, \dots, n_{\text{out}}, \tag{6.78b}$$

for all input data points $z \in \mathcal{D}_{\text{in}}$.

Results are summarized in Table 6.1, showing satisfactory approximation errors for our purposes.

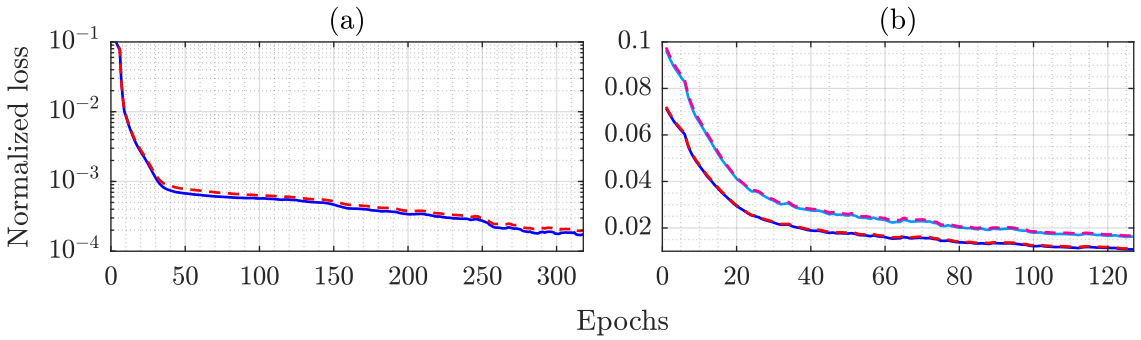


Figure 6.4. NA-MPC FNNs training and validation losses over the training epochs. **(a)** Neural plant model ($\mathcal{N}_{f_c}$): training (——); validation (- - -). **(b)** Neural MPC controller ($\mathcal{N}_\pi$), reporting the losses on each FNN output and on the sum of the two outputs: training ($P_\mathrm{b}, P_\mathrm{fc}$ ——; P_tot ——); validation ($P_\mathrm{b}, P_\mathrm{fc}$ - - -; P_tot - - -).

FNN	Output quantity	Normalized error		
		Min	Average	Max
Neural plant model ($\mathcal{N}_{f_c}$)	$\dot{\zeta}$	0.13%	1.28%	6.04%
	$\dot{m}_\mathrm{h}$	0.07%	0.84%	2.97%
Neural MPC controller ($\mathcal{N}_\pi$)	P_b	0.00%	0.38%	8.07%
	P_fc	0.00%	0.16%	8.60%
	P_tot	0.00%	0.30%	5.87%

Table 6.1. NA-MPC FNNs approximation errors.

6.6.3. NA-MPC Performance

The NA-MPC strategy with online metaheuristic tuning is tested on four different power management control tasks. Such tasks are reported in the following, together with the related parameters for the performance-metric functions in Eqs. (6.31) and (6.75):

1) Requested power tracking ($\alpha_1 = 10$, $\alpha_2 = \alpha_3 = 1$, $e_\mathrm{ub} = 1\%$);
2) Requested power tracking and combined supplies saving ($\alpha_1 = 10$, $\alpha_2 = \alpha_3 = 5$, $e_\mathrm{ub} = 10\%$);
3) Requested power tracking and battery saving ($\alpha_1 = \alpha_2 = 10$, $\alpha_3 = 1$, $e_\mathrm{ub} = 10\%$);
4) Requested power tracking and fuel cell saving ($\alpha_1 = \alpha_3 = 10$, $\alpha_2 = 1$, $e_\mathrm{ub} = 10\%$).

To verify the effectiveness of the online tuning strategy, we compare it with an "ideal" tuning policy. Such an ideal tuning policy is formulated with the same optimization problem (6.29) and performance-metric functions (6.75), but, at each tuning instant k, the prior simulation (6.30) is performed over the whole time interval $[kT_s, (N-1)T_s]$, assuming to know in advance the future power request $P_{\mathrm{req},\{k+1,\ldots,N-1\}}$. Thus, such a tuning strategy provides ideal results, but is not realizable in practice, since it does not respect causality.

The initial weights w_0 for the neural MPC controller are set to $w_0 = \frac{w_\mathrm{lb}+w_\mathrm{ub}}{2}$ (refer to the data reported on p. 214).

Figures 6.5-6.7 report the simulation results of NA-MPC with online tuning for control task 1, employing each driving cycle in Figure 6.3. Simulation results for all control tasks and all driving cycles, including both online tuning and ideal non-causal tuning, are

summarized in Tables 6.2 and 6.3.

Task	Max tracking error	Average tracking error	Battery consumption	Fuel cell consumption	Cumulative consumption
NA-MPC (online tuning)					
1	1.28%	0.63%	72.43%	41.44%	56.93%
2	16.72%	5.00%	48.54%	43.48%	46.01%
3	13.06%	4.10%	31.71%	79.39%	55.55%
4	15.44%	3.33%	83.53%	11.61%	47.57%
NA-MPC (ideal non-causal tuning)					
1	1.18%	0.47%	71.54%	42.07%	56.81%
2	10.54%	3.08%	58.81%	39.77%	49.29%
3	10.85%	3.02%	31.74%	89.45%	60.59%
4	10.28%	2.03%	92.35%	9.57%	50.96%

Table 6.2. Control performance: NA-MPC online tuning and ideal non-causal tuning (driving cycle (a)).

Task	Max tracking error	Average tracking error	Battery consumption	Fuel cell consumption	Cumulative consumption
Driving cycle (b)					
1	1.89%	0.55%	76.57%	36.75%	56.66%
2	15.39%	3.60%	53.31%	42.14%	47.73%
3	17.18%	3.80%	31.59%	71.91%	51.75%
4	16.94%	3.70%	70.60%	18.71%	44.66%
Driving cycle (c)					
1	1.70%	0.62%	53.53%	30.96%	42.25%
2	15.30%	4.72%	25.98%	37.88%	31.93%
3	15.81%	4.72%	−0.58%	79.71%	39.56%
4	14.33%	4.37%	45.10%	12.12%	28.61%

Table 6.3. Control performance of NA-MPC with online tuning (driving cycles (b) and (c)).

We notice that NA-MPC with online tuning consistently attains all the given tasks: the power tracking error stays close from below to the soft upper bound e_{ub}, which matches the intended behavior enforced by the function p_1 in Eqs. (6.75a) and (6.75b); also, the trade-off between tracking accuracy and supplies saving is effectively attained, both for combined and selective supplies saving.

Ideal non-causal tuning, as expected, yields more precise and consistent results, owing to its knowledge of the future power request. Nonetheless, online tuning exhibits remarkably similar performance to ideal non-causal tuning, proving the effectiveness of online causal tuning in delivering equally accurate control performance.

6.6.4. Comparison of NA-MPC with State-of-the-Art Techniques

We now compare NA-MPC with online tuning against the two most prominent state-of-the-art techniques for HEV power management: Equivalent Consumption Minimization Strategy (ECMS) with online adaptation [114] and Fuzzy-Logic Control (FLC) [70, 51].

Adaptive ECMS (A-ECMS) seeks the optimal powers P_{b} and P_{fc} minimizing, at each

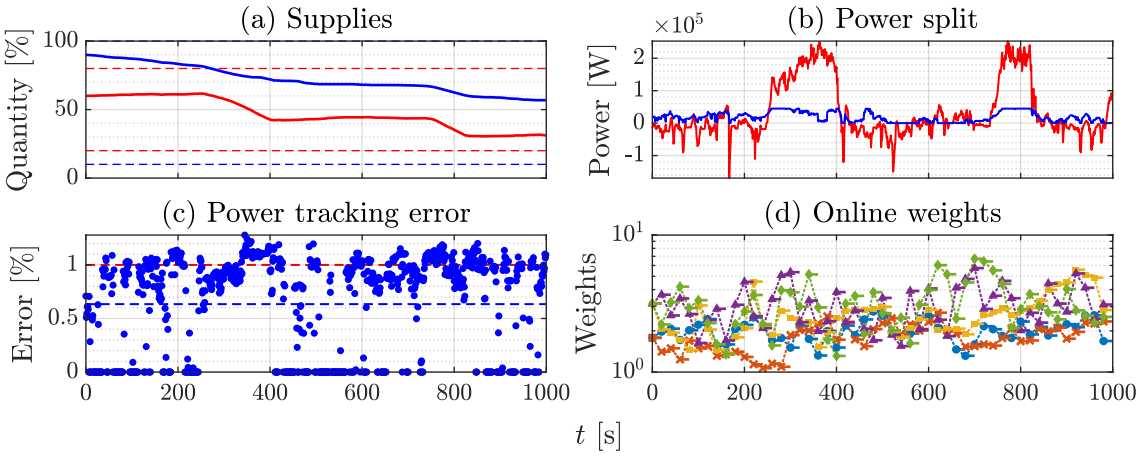


Figure 6.5. Control task 1, driving cycle (a).

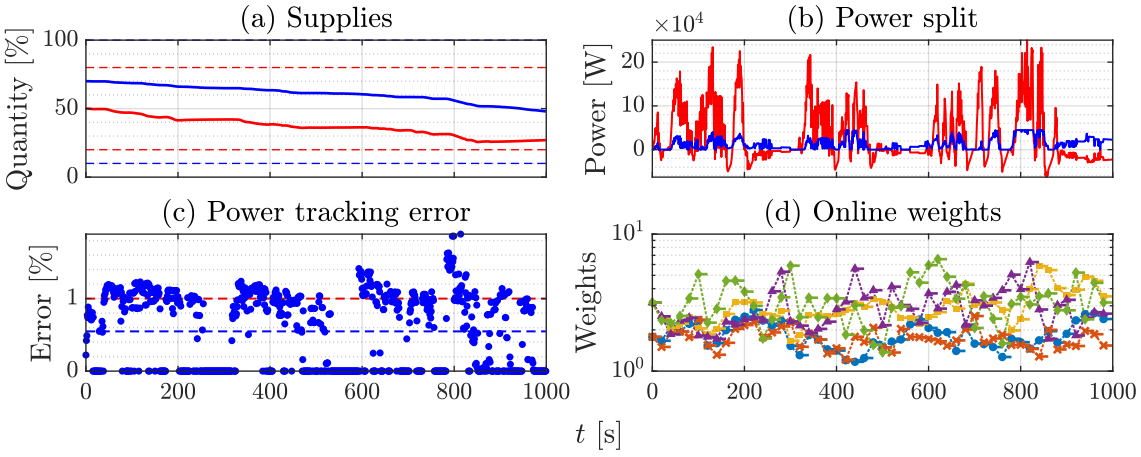


Figure 6.6. Control task 1, driving cycle (b).

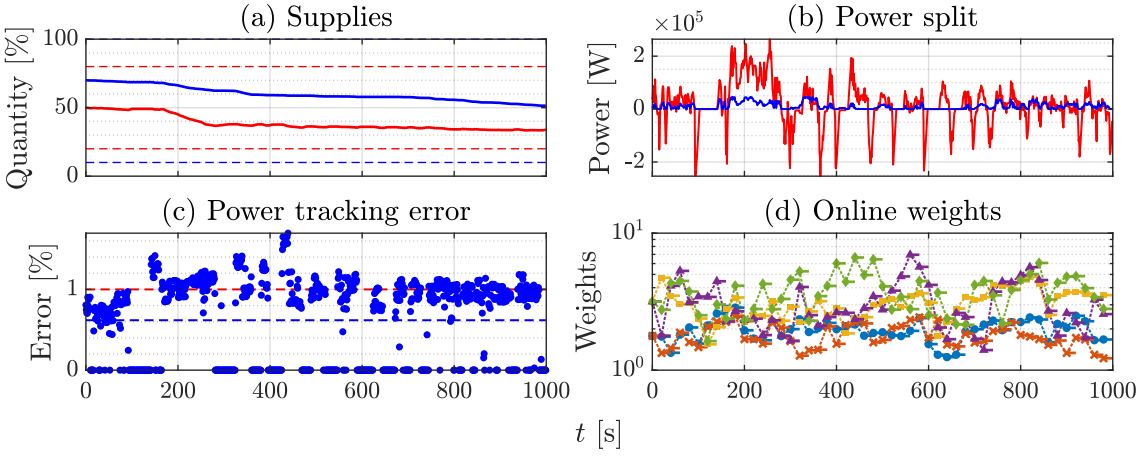


Figure 6.7. Control task 1, driving cycle (c).

Figures 6.5-6.7. Simulation of NA-MPC with online tuning. **(a)** Supplies: SOC (——), H_2 mass (——), and related upper-lower bounds (dashed lines). **(b)** Power allocation: P_b (——); P_{fc} (——). **(c)** Normalized power tracking error $e_k = |P_{tot,k} - P_{req,k}| / \max_{i\in\{0,\dots,N-1\}} |P_{req,i}|$: e (•) and average value (- - -); e_{ub} (- - -). **(d)** Optimal MPC weights obtained from the online tuning: w_ζ (•); w_{m_h} (×); $w_{P_{tot}}$ (■); $w_{\Delta,\zeta}$ (▲); w_{Δ,m_h} (♦).

time instant k, the instantaneous supply rates [114], i.e.,

$$\begin{aligned} \min_{P_\mathrm{b},P_\mathrm{fc}} \quad & \dot{\zeta}(\zeta', P_\mathrm{b}) + s \cdot \dot{m}_\mathrm{h}(P_\mathrm{fc}) \\ \text{s.t.} \quad & P_\mathrm{b} + P_\mathrm{fc} = P_{\mathrm{req},k}, \\ & P_\mathrm{b} \in [P_{\mathrm{b,lb}}, P_{\mathrm{b,ub}}], \quad P_\mathrm{fc} \in [P_{\mathrm{fc,lb}}, P_{\mathrm{fc,ub}}], \end{aligned} \tag{6.79}$$

where $\zeta' = \frac{\zeta_\mathrm{lb}+\zeta_\mathrm{ub}}{2}$ makes the cost function depend only on P_b and P_fc [114]. The functions $\dot{\zeta}(\zeta, P_\mathrm{b})$ and $\dot{m}_\mathrm{h}(P_\mathrm{fc})$ are taken from the CT neural plant model (6.18). Adaption is performed every N_tuning time steps and, at each $k = nN_\mathrm{tuning}$, $n \in \mathbb{N}_{\geq 1}$, the parameter s is tuned through the following policy [114]:

$$s_n = \frac{1}{2}(s_{n-1} + s_{n-2}) + K\frac{\zeta_k - \zeta'}{\zeta_\mathrm{ub} - \zeta_\mathrm{lb}}. \tag{6.80}$$

Fuzzy-Logic Control (FLC), at each time instant k, takes as inputs the current SOC ζ_k and requested power $P_{\mathrm{req},k}$, and returns as output the FC power $P_{\mathrm{fc},k}$; the battery power is then computed through the equality $P_{\mathrm{b},k} = P_{\mathrm{req},k} - P_{\mathrm{fc},k}$. Following Mamdani's FLC paradigm [51], inputs and outputs are mapped to linguistic variables by trapezoidal membership functions, according to their value (i.e., "negative", "zero", "very low", "low", "medium", "high") [70, 51]. The output linguistic value is obtained from the inputs by means of logic implications. Finally, the output numerical value is computed from the linguistic one via centroid defuzzification.

As also stated in Section 6.1, due to their formulations [114, 51], both A-ECMS and FLC are inherently unable to perform an effective trade-off between tracking accuracy and supplies saving (due to the enforced equality $P_\mathrm{b} + P_\mathrm{fc} = P_\mathrm{req}$); therefore, the two techniques are compared with NA-MPC only for task 1 (i.e., requested power tracking). Also, both A-ECMS and FLC are unable to handle constraints on the supply values; thus, when one of ζ or m_h goes outside the bounds $[\zeta_\mathrm{lb}, \zeta_\mathrm{ub}]$ or $[m_{\mathrm{h,lb}}, m_{\mathrm{h,ub}}]$, the corresponding power P_b or P_fc goes to 0, according to Eqs. (6.66) and (6.67).

A-ECMS is initialized with two sets of values:

1) $s_0 = s_{-1} = 1$, $K = 1.5$;
2) $s_0 = s_{-1} = 2$, $K = 3$.

Also FLC is set up with two sets of membership functions and logic rules, where the first set is taken from [51] (adapting the ranges of inputs and output).

For both A-ECMS and FLC, these two setups are denoted with the superscripts (1) and (2), respectively.

Simulation results for control task 1 and each driving cycle in Figure 6.3 are reported in Figures 6.8-6.10, and are summarized in Table 6.4.

We notice that, for driving cycle (a), A-ECMS$^{(1)}$ and FLC$^{(1)}$ do not track effectively the requested power, since they fail to meet the SOC lower bound at time $t = 825$ s. Similarly, for driving cycles (b) and (c), A-ECMS$^{(2)}$ and FLC$^{(1)}$ violate the supplies constraints as well. The alternative setups (i.e., A-ECMS$^{(2)}$, FLC$^{(2)}$ for driving cycle (a), and A-ECMS$^{(1)}$, FLC$^{(2)}$ for driving cycles (b) and (c)) attain tracking, but they are unable to perform a suitable trade-off between tracking accuracy and supplies saving.

In contrast, NA-MPC manages to save $\sim$ 8-9% more supplies, with a rather minimal increase in the power tracking error.

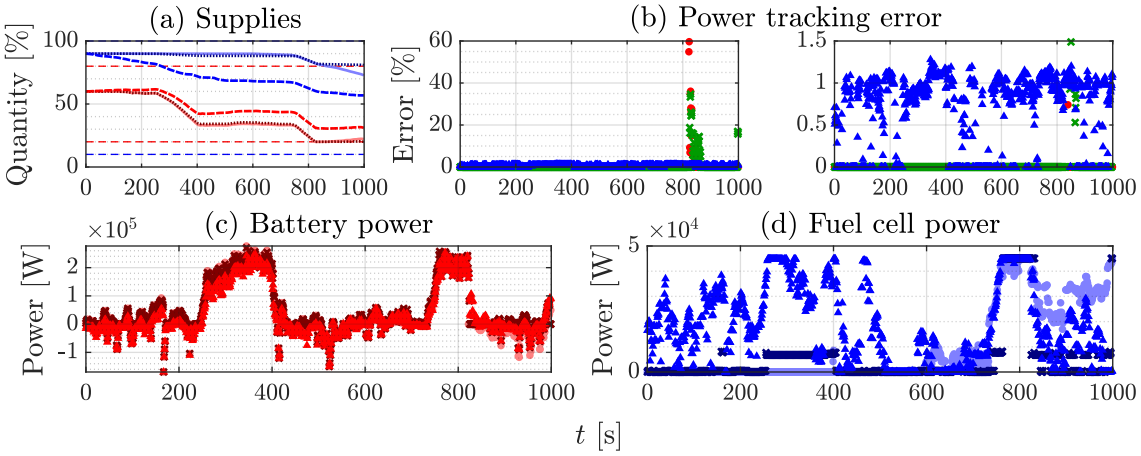


Figure 6.8. Control task 1, driving cycle (a), comparing NA-MPC with A-ECMS$^{(1)}$ and FLC$^{(1)}$.

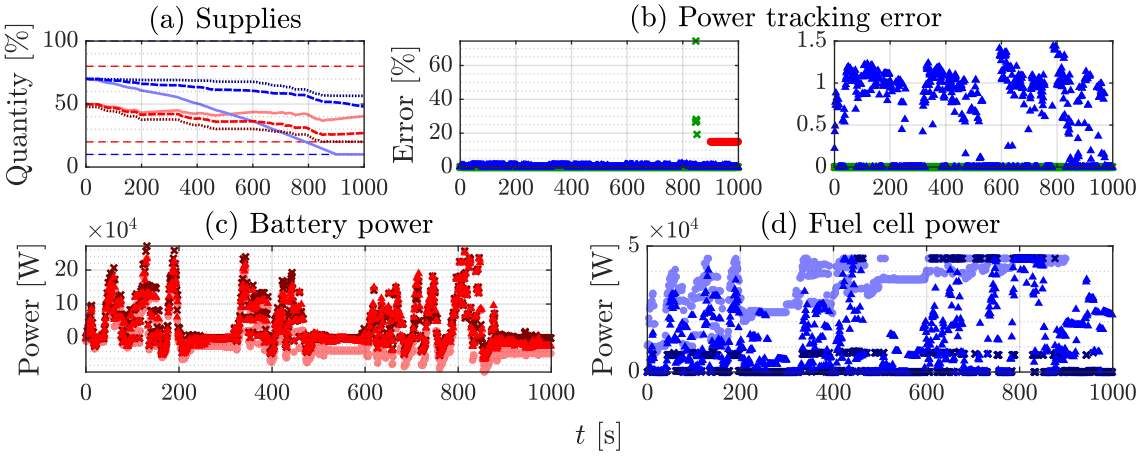


Figure 6.9. Control task 1, driving cycle (b), comparing NA-MPC with A-ECMS$^{(2)}$ and FLC$^{(1)}$.

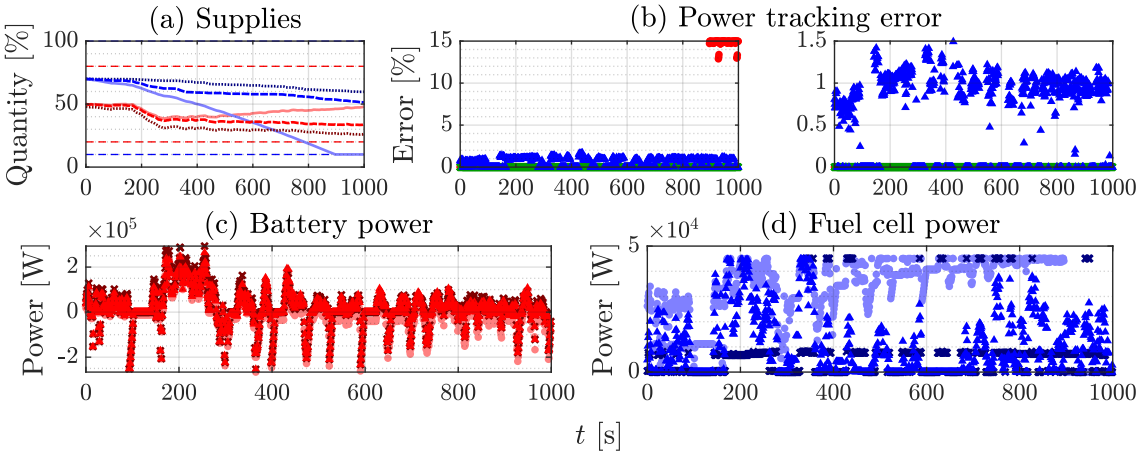


Figure 6.10. Control task 1, driving cycle (c), comparing NA-MPC with A-ECMS$^{(2)}$ and FLC$^{(1)}$.

Figures 6.8-6.10. Comparison of NA-MPC with A-ECMS and FLC. **(a)** Supplies: SOC (NA-MPC ----; A-ECMS ——; FLC ·····), H_2 mass (NA-MPC ----; A-ECMS ——; FLC ·····), and related upper-lower bounds (dashed lines). **(b)** Normalized power tracking error $e_k = |P_{\text{tot},k} - P_{\text{req},k}| / \max_{i\in\{0,\dots,N-1\}} |P_{\text{req},i}|$, with detail of the error interval $[0, 1.5]\%$ (NA-MPC ▲; A-ECMS ●; FLC ×). **(c)** Battery power P_{b} (NA-MPC ▲; A-ECMS ●; FLC ×). **(d)** Fuel cell power P_{fc} (NA-MPC ▲; A-ECMS ●; FLC ×).

	Max. tracking error	Average tracking error	Battery consumption	Fuel cell consumption	Cumulative consumption
			Driving cycle (a)		
NA-MPC	1.28%	0.63%	72.43%	41.44%	56.93%
A-ECMS$^{(1)}$	59.67%	0.33%	94.70%	21.60%	58.15%
A-ECMS$^{(2)}$	0.00%	0.00%	73.57%	51.10%	62.33%
FLC$^{(1)}$	34.69%	0.50%	99.93%	11.60%	55.76%
FLC$^{(2)}$	0.00%	0.00%	61.70%	65.16%	63.43%
			Driving cycle (b)		
NA-MPC	1.89%	0.55%	76.57%	36.75%	56.66%
A-ECMS$^{(1)}$	0.00%	0.00%	71.75%	54.87%	63.31%
A-ECMS$^{(2)}$	14.98%	1.50%	31.96%	99.91%	65.93%
FLC$^{(1)}$	74.44%	0.18%	99.67%	22.45%	61.06%
FLC$^{(2)}$	0.00%	0.00%	79.74%	46.71%	63.23%
			Driving cycle (c)		
NA-MPC	1.70%	0.62%	53.53%	30.96%	42.25%
A-ECMS$^{(1)}$	0.00%	0.00%	45.31%	55.42%	50.37%
A-ECMS$^{(2)}$	14.98%	1.54%	6.31%	99.91%	53.11%
FLC$^{(1)}$	0.00%	0.00%	77.56%	17.11%	47.33%
FLC$^{(2)}$	0.00%	0.00%	58.70%	36.94%	47.82%

Table 6.4. Control performance: NA-MPC and state-of-the-art techniques (control task 1).

These results also highlight that, unlike NA-MPC, A-ECMS and FLC are highly sensitive to the choice of initial values, membership functions, and logic rules.

6.6.5. Online Metaheuristic Tuning Performance

PSO and GA Comparison

We now compare the performance of PSO and GA in adapting the MPC weights, by serving as metaheuristic optimization algorithms to solve online the tuning problem (6.29).

The comparison considers NA-MPC with online tuning for control task 1 and driving cycle (a).

Results are reported in Figure 6.11 and Table 6.5. In particular, the online weights obtained with GA are reported in Figure 6.12.

	Max tracking error	Average tracking error	Battery consumption	Fuel cell consumption	Cumulative consumption
PSO	1.28%	0.63%	72.43%	41.44%	56.93%
GA	1.51%	0.64%	71.11%	42.39%	56.75%

Table 6.5. Comparison of PSO and GA for online metaheuristic tuning (control task 1, driving cycle (a)).

We notice that PSO and GA achieve almost equivalent results for both power tracking and supplies consumption.

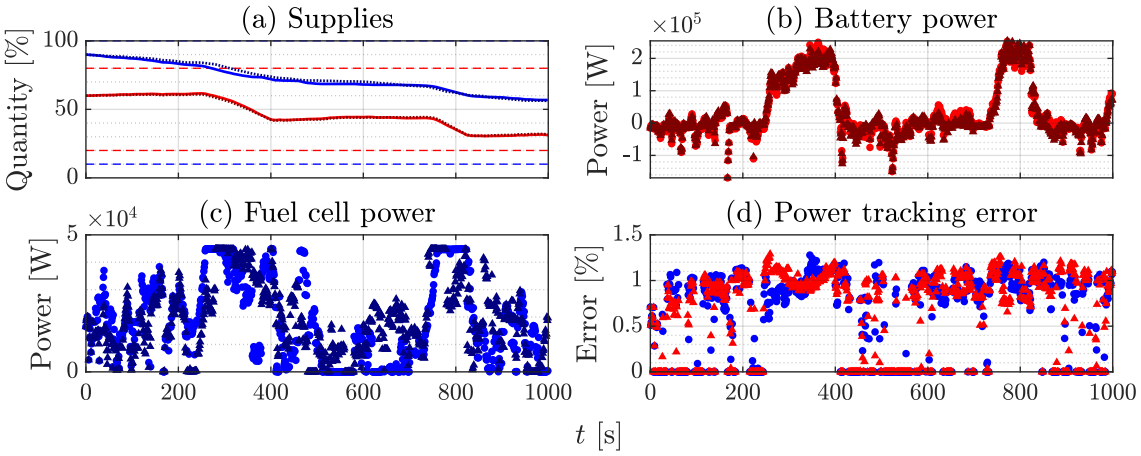


Figure 6.11. Comparison of PSO and GA for online metaheuristic tuning (control task 1, driving cycle (a)). **(a)** Supplies: SOC (PSO ——; GA ·····), H_2 mass (PSO ——; GA ·····), and related upper-lower bounds (dashed lines). **(b)** Battery power P_b: PSO (•); GA (▲). **(c)** Fuel cell power P_{fc}: PSO (•); GA (▲). **(d)** Normalized power tracking error $e_k = |P_{tot,k} - P_{req,k}| / \max_{i\in\{0,\dots,N-1\}} |P_{req,i}|$: PSO (•); GA (▲).

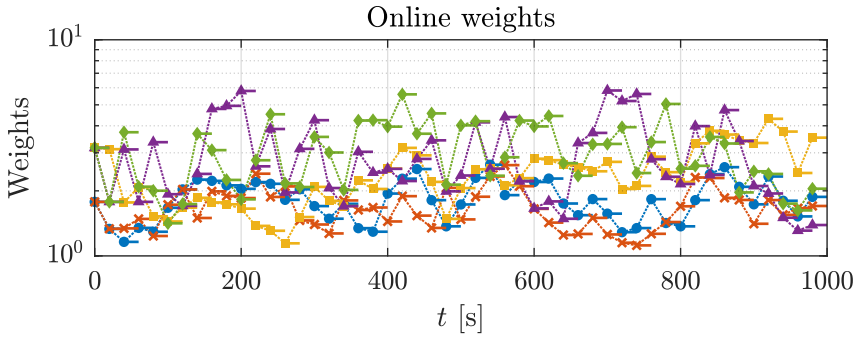


Figure 6.12. Optimal MPC weights obtained from the online tuning with GA (control task 1, driving cycle (a)): w_ζ (•); w_{m_h} (×); $w_{P_{tot}}$ (■); $w_{\Delta,\zeta}$ (▲); w_{Δ,m_h} (♦).

Convergence of PSO

To assess whether PSO is effectively converging towards the optimal MPC weights at each tuning instant, we simulate NA-MPC with online tuning (for control task 1 and driving cycle (a)), collecting, for each tuning instant, the PSO particles positions at every PSO iteration.

Figure 6.13 reports the particles positions at the first and last iteration of PSO. We clearly notice that, at each tuning instant, all PSO particles effectively converge, gathering in a neighborhood of the MPC weight value that will be selected as optimal. This also proves that the PSO parameters, selected according to [107] (see Remark 6.7 and the data on p. 214), provide a good balance between exploration and exploitation.

Figure 6.13 also reports the actual weights w_k^* that are provided online to the neural MPC controller (dashed lines), obtained through the smoothing policy (6.40).

6.6.6. Execution Time

Considering all simulations, computing the optimal control input through the neural MPC controller $\mathcal{N}_\pi$ (6.73) requires, at each time instant, within [0.0126, 0.4292] ms, with average

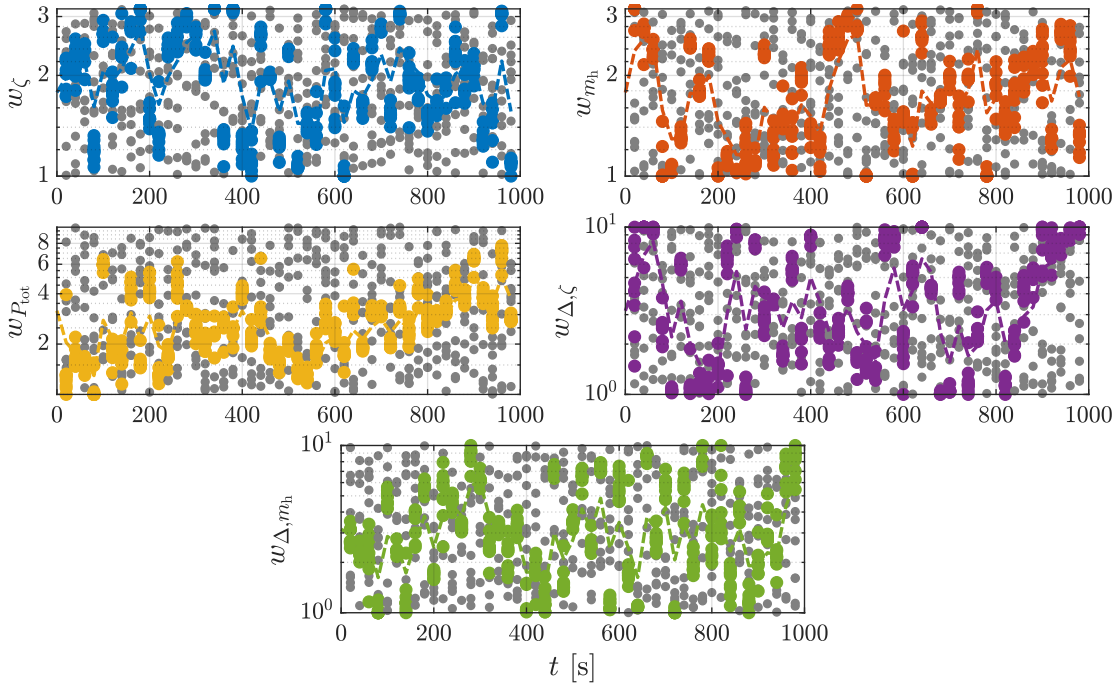


Figure 6.13. Convergence of PSO particles for NA-MPC with online tuning (control task 1, driving cycle (a)). For each tuning time instant and for each MPC weight are reported: particles initial positions (grey marks •); particles final positions (colored marks); online weights provided to the neural MPC controller (dashed lines).

value 0.0248 ms.

For what concerns online tuning, the optimal MPC weights are computed, at each tuning instant, in:

- [108.34, 198.80] ms, with average value 119.36 ms, using PSO;
- [108.34, 198.80] ms, with average value 119.36 ms, using GA.

We see that, for both PSO and GA, execution times are very similar.

Overall, execution times are within the discrete time step $T_s = 1$ s, meaning that NA-MPC is real-time feasible.

Note that, if the neural MPC controller and the online tuning policy are implemented to run in parallel, the discrete time step T_s could be reduced up to the maximum evaluation time of the neural MPC controller (i.e., ~ 0.01 ms). This would also allow to consider a finer sampling of the requested power $P_{\text{req},k}$.

6.7. Chapter Summary

In this chapter, we presented the Neural Adaptive Model Predictive Control (NA-MPC) strategy, applied to the case study of power management in fuel cell hybrid electric vehicles (FCHEVs).

The NA-MPC strategy optimally allocates the powers generated by the FCHEV energy sources, i.e, the electric battery and the hydrogen fuel cell (FC). NA-MPC employs a neural black-box model of the FCHEV power sources and distribution system, to enable applicability when an accurate white-box model of the plant is not readily available. NA-

MPC is capable of adapting its control action in real-time to achieve multiple, conflicting control objectives at once, thanks to an online metaheuristic tuning policy for the MPC weights. Furthermore, real-time feasibility of NA-MPC is ensured by emulating the MPC control policy by means of feedforward neural networks (FNNs). Guarantees on the smoothness of the MPC policy have been provided, ensuring that the universal approximation theorem of FNNs holds for the MPC neural emulation.

The effectiveness of the NA-MPC strategy was demonstrated through an extensive simulation campaign, showcasing the effectiveness in handling multiple control objectives concurrently and the excellent performance of the adaptation policy. Also, NA-MPC outperformed the most prominent power management strategies for hybrid electric vehicles, namely, fuzzy-logic control (FLC) and equivalent consumption minimization strategy (ECMS).

7

Conclusions

In this thesis, we presented a range of novel contributions in the domain of Model Predictive Control (MPC), with the aim of advancing it along two main directions: the development of fast and real-time feasible MPC frameworks, capable of operating under strict timing constraints and limited computational resources, and the extension of Nonlinear Economic MPC (E-NMPC) to handle multiple, conflicting, and time-varying control objectives, while ensuring consistent control performance and systematic closed-loop stability guarantees.

The first contribution of this thesis addressed the computational limitations inherent to Nonlinear MPC (NMPC) by formulating a general analytical framework for fast solving NMPC problems in the Koopman lifted space. The proposed Koopman NMPC (K-NMPC) approach is grounded in the Koopman operator theory, of which we considered a tailored extension to optimal control problems. We introduced a systematic analytical procedure to derive a suitable basis of Koopman observables, capable of lifting both the nonlinear prediction model and the nonlinear constraints of NMPC. Moreover, a general methodology to handle infinite-dimensional Koopman systems was developed, allowing an arbitrary dimensionality reduction while preserving a suitable approximation accuracy. In this way, we are able to transform NMPC problem into equivalent quadratic programs (QPs) in the Koopman lifted space, which can be solved with superior computational performance. Our K-NMPC framework was validated, both in simulation and through experimental validations, on real-world case studies, namely mobile robot navigation in cluttered environments and autonomous parallel parking, which showcased the proficiency of K-NMPC in attaining the given control tasks, providing closed-loop trajectories that are very close to the NMPC ones, and outperforming NMPC execution time by over an order of magnitude.

The second contribution explored an alternative pathway to achieve fast MPC operation through direct hardware implementation, introducing a general methodology for designing fully-analog electronic circuits that realize Linear-Quadratic MPC (QP-MPC) policies in their explicit form. Starting from a general QP-MPC problem, its corresponding Explicit MPC (Ex-MPC) policy, characterized by a piecewise-affine control law over a partition of polytopic regions, was derived and then simplified through tailored complexity-reduction techniques. The resulting reduced Ex-MPC was then implemented as an analog

circuit, composed solely of commercially-available low-latency components, including resistors, capacitors, comparators, operational amplifiers, logic gates, and multiplexers. This approach, referred to as Analog Circuital Ex-MPC, eliminates the need for online computation on digital hardware, and ensures a remarkably fast evaluation of the MPC policy, with computation times comparable with the analog circuit propagation delay of few microseconds. Our methodology was validated through the control of DC-DC Buck converters, a class of systems with inherently fast dynamics and high switching frequencies. Comprehensive simulations, including high-level Monte Carlo analyses under parametric uncertainty and low-level circuit simulations accounting for non-ideal component behavior, demonstrated the feasibility, robustness, and solid control performance of the proposed design. Furthermore, the analog Ex-MPC circuit outperformed conventional control strategies in terms of line and load disturbance rejection.

The third contribution focused on extending the classical framework of Economic NMPC to address scenarios involving conflicting control objectives and to systematically guarantee closed-loop stability. A novel E-NMPC formulation was proposed, which integrates both economic and tracking tasks within the same optimal control problem, enabling regulation towards an optimal trade-off equilibrium. To overcome the difficulty of ensuring closed-loop stability in E-NMPC, arising from the non-minimality of the multi-objective stage cost at the trade-off equilibrium, a general constructive procedure was developed for designing suitable stabilizing terms, which guarantee asymptotic stability with minimal impact on the economic performance. The proposed E-NMPC strategy was validated on the case study of Adaptive Cruise Control (ACC) for electric vehicles, a domain where energy optimization and driving-related tracking objectives naturally conflict. Simulations demonstrated that our E-NMPC strategy effectively achieved the optimal trade-off between both objectives, attaining the tracking task while delivering superior economic performance compared to conventional NMPC, together with guaranteeing closed-loop stability.

The fourth and final contribution presented a new advanced MPC framework, named Neural Adaptive Model Predictive Control (NA-MPC), designed to address time-varying and conflicting control objectives while ensuring real-time feasibility. NA-MPC extends the standard formulation of QP-MPC by introducing three key components: an online metaheuristic tuning strategy that dynamically adapts the MPC weights to achieve multiple concurrent objectives; a neural emulation of the QP-MPC control policy, providing an equivalent neural controller that exhibits negligible computation time; and a neural black-box model of the plant to control, to enable applicability when an accurate white-box model of the plant is not readily available. The NA-MPC framework was applied to the case study of power management in fuel cell hybrid electric vehicles (FCHEVs). In this application, the control task consists in optimally allocating the power generated by the electric battery and the hydrogen fuel cell, balancing an accurate power delivery to the vehicle with the minimization of supplies consumption. Extensive simulations and comparative analyses against state-of-the-art power management strategies, namely Fuzzy Logic Control (FLC) and Equivalent Consumption Minimization Strategy (ECMS), confirmed the superiority of NA-MPC in terms of performance, adaptability, and computational efficiency.

Overall, this thesis successfully achieved its objectives, by proposing two novel MPC frameworks, the Koopman Nonlinear MPC and the Analog Circuital Explicit MPC, which ensure fast and real-time feasible solutions for Nonlinear MPC and Linear-Quadratic MPC

problems, respectively. Moreover, the novel Economic NMPC formulation and the Neural Adaptive MPC strategy have shown that MPC can be endowed with the flexibility to handle conflicting and time-varying control objectives, while systematically guaranteeing closed-loop stability and ensuring high computational efficiency.

7.1. Future Research

While the results achieved in this thesis are satisfactory, its theoretical and practical contributions still present some limitations. Accordingly, a first natural direction for future research is to investigate and address these issues, with the aim of strengthening the proposed approaches and obtaining even better results:

- Concerning the Koopman NMPC strategy, a key open point consists in evaluating how the approximations performed on the Koopman lifted system affect the K-NMPC solution quality and its closed-loop stability. Specifically, this includes analyzing the impact of considering an affine parameter-varying (APV) form of the lifted system, whose purpose is to linearize the system bilinearity, as well as the effects of dimensionality reduction, both from truncating the basis of observables and from approximating the nonlinear residual term in APV form as well.

  A second limitation lies in the procedure for generating the basis of observables, which, currently, performs a static generation, starting from the original system states and the additional hand-picked functions of interest. We would like to extend it so that the most "relevant" observables, i.e., those contributing most to the prediction accuracy of the reduced lifted system, are generated in advance.

- Regarding the Analog Circuital Explicit MPC approach, we aim to extend the current stability analysis, which at present yields only local results. Our goal is to systematically verify the global asymptotic stability of the closed-loop system, together with analyzing its robustness to bounded disturbances.

  Another limitation that we would like to address concerns the hyperplane separation of Ex-MPC regions, which, in some cases, may not guarantee the existence of the desired affine separation function. When this occurs, it is necessary to resort to more complex separating functions, like polynomial ones [85]. A key challenge then is to design an efficient analog implementation of these more general separating functions, in order to maintain the simplicity and low-cost advantages of our Analog Circuital Ex-MPC framework.

- For what concerns Economic NMPC for conflicting control objectives, a main limitation of the proposed approach is that it employs a singleton terminal set $\mathcal{X}_f$ for ensuring the closed-loop stability. As highlighted in Remark 2.7, such a choice bears the hallmarks of simplicity but reduces the size of the feasible states set $\mathcal{X}_{N_p}$, which coincides with the region of attraction of the trade-off equilibrium state x_s. Therefore, we would like to improve the E-NMPC optimal control problem by considering more general terminal ingredients and, possibly, introducing artificial references [52]; both of these approaches shall allow to enlarge the region of attraction.

- Finally, with regard to the Neural Adaptive MPC strategy, a relevant direction for future research is to explore alternative neural network architectures – rather than

restricting to feedforward neural networks (FNNs) only – for both MPC emulation and plant identification, such as recurrent neural networks (RNNs) [117] and long short-term memory (LSTM) networks [147].

Another limitation of the NA-MPC strategy is that the emulated MPC controller does not inherently guarantee closed-loop stability. While this is generally not problematic for power management applications, as discussed in Section 6.5.3, p. 213, it becomes critical when state regulation or tracking constitute the primary control objective. Accordingly, we intend to further investigate this aspect, also drawing on relevant results available in the literature [120, 141].

Beyond addressing current limitations, the promising results presented in this thesis pave the way for several other avenues of future research. In the following, we propose a range of possible research directions that are worth further investigating.

Lifting Economic NMPC in the Koopman Space An interesting direction for future research consists in extending the Koopman NMPC framework to Economic NMPC problems. The K-NMPC approach, in its current formulation, allows to lift NMPC problems that feature a nonlinear prediction model, nonlinear state constraints, and a quadratic stage cost function. This limits its applicability to NMPC for regulation and tracking tasks only. Economic NMPC, in contrast, considers a nonlinear and generally non-quadratic stage cost, representing an economic criterion. Therefore, we could think of extending the K-NMPC lifting strategy to handle nonlinear stage cost functions, arising in E-NMPC problems.

This extension could be achieved by including the elementary nonlinear terms composing the stage cost within the initial set of observables, analogously to the procedure already employed for lifting nonlinear constraints. Then, the basis of observables, generated through the analytical procedure proposed in this thesis, would now lift also the nonlinear stage cost, together with the nonlinear prediction model and nonlinear state constraints of the E-NMPC problem. The resulting optimal control problem in the Koopman lifted space would then take the form of a linear program (LP), closely approximating the original E-NMPC solution and significantly reducing its computational complexity.

This extended K-NMPC framework, that we can call *Koopman Economic NMPC* (KE-NMPC), would represent a significant step forward in rendering E-NMPC problems computationally tractable and feasible for real-time applications.

Dual-Layer MPC for Energy-Efficient Vehicle Operation This thesis presented two novel MPC frameworks capable of providing an economically profitable control action while also attaining other adversarial tasks, namely Economic NMPC for conflicting control objectives and Neural Adaptive MPC. A promising avenue of future research would consist in integrating these two frameworks together, obtaining a unified dual-layer control architecture for advanced power management in hybrid vehicles (HVs).

In such a scheme, the outer control layer is governed by E-NMPC, which accounts for the vehicle dynamics and, thus, can be tasked with motion-level control objectives, such as cruise control, autonomous parking, lane keeping, and other driving assistance tasks. Concurrently, E-NMPC is capable of attaining conflicting economic objectives, among

which minimizing the overall power demand required to execute the motion-level task. Importantly, this outer control layer is ensured to be closed-loop stable, thanks to its general constructive stability guarantees.

The inner layer, instead, is entrusted to the NA-MPC framework, which receives from the outer layer the optimized power demand and determines, in real time, the optimal allocation of this power among the available onboard energy sources. Thanks to its online tuning capability, NA-MPC can dynamically adapt to time-varying driving conditions and upper-level control requirements, ensuring consistent performance and optimal utilization of the supplies.

The synergy between E-NMPC and NA-MPC in such a hierarchical structure would offer a powerful framework endowing HVs with autonomous driving functions and enabling a two-fold optimization of energy consumption during their operation.

Robust Stability Analysis of Koopman NMPC Another promising research direction concerns the analysis of the robust stability of Koopman NMPC in presence of modeling errors, arising from the approximations introduced by the Koopman lifting process. In its current formulation, K-NMPC introduces two sources of approximation: first, the bilinear terms of the lifted prediction model are linearized to obtain an affine time-varying representation; second, in the case of an infinite-dimensional basis of observables, the latter is truncated to a finite dimension of choice. These approximations inevitably introduce modeling errors, which correspond to removing a residual nonlinear dynamics from the lifted prediction model. However, thanks to the fully-analytical nature of the proposed K-NMPC framework, the resulting model mismatch can be expressed in closed form, thus allowing a systematic study of its effect on closed-loop stability.

A possible development in this direction would consist in leveraging finite-gain stability (FGS) arguments. The FGS framework provides a flexible and practically meaningful notion of robustness, quantifying the sensitivity of the closed-loop system to uncertainties through an input-state gain relation. Within this framework, we can characterize the impact of the Koopman modeling errors as a structured uncertainty and establish sufficient conditions under which the K-NMPC in closed-loop exhibits finite-gain stability with respect to this uncertainty. Importantly, the FGS framework paves to way for developing a systematic approach to determine suitable NMPC configurations, in terms of, e.g., parameter selection, such as the weighing matrices, that guarantee FGS, providing at the same time consistent control performances.

Solving Nonlinear MPC with Quantum Computing In the 1980s, Richard Feynman argued that *quantum computing* (QC) would become an effective tool to simulate large-scale physical systems, overcoming the inherent limitations of classical computers [53]. This claim relies on the observation that the intrinsic structure of quantum processors makes them well suited for specific computational tasks, which can be executed exponentially faster on a quantum processor rather than on a classical one [11].

QC leverages the principles of quantum mechanics, among which *superposition* and *entanglement*, to execute computational tasks with an exponential increase in performance compared to classical computers [143]. Unlike conventional digital processors, which rely on bits, taking values of either 0 or 1, quantum computers employ *qubits*, which can exist

in a superposition of both states simultaneously. This distinctive property allows quantum processors to explore a vast number of possible solutions in parallel, thereby offering a dramatic increase in computational performance.

Several paradigms and hardware architectures have been proposed to realize QC. The most two relevant ones are *gate-based quantum computing* [86] and *adiabatic quantum computing* (AQC) [2]. In gate-based QC (also known as circuit-based or digital QC), the computation is performed through a sequence of unitary operations (quantum gates) applied to a register of qubits, whose final state is measured at the end of the process.

Conversely, AQC relies on the *adiabatic theorem* of quantum mechanics, which states that a quantum system remains in its instantaneous ground state if the Hamiltonian governing it is varied sufficiently slowly. This framework is particularly well suited for optimization problems. The idea is to encode the desired optimal solution in the ground state of a target Hamiltonian, while the system is prepared in the ground state of an initial Hamiltonian that is easy to construct. The system is then evolved under a time-varying Hamiltonian that smoothly interpolates between the initial and the target one. If this evolution is slow enough, the system tracks the ground state throughout the process and ends in the ground state of the target Hamiltonian, which encodes the optimal solution.

A closely related approach is *quantum annealing* (QA) [10, 161]. QA can be viewed as a practical relaxation of AQC. Instead of strictly enforcing fully adiabatic evolution, QA employs a heuristic quantum process to drive the system towards low-energy states. Specifically, QA exploits quantum effects such as *tunneling* to explore the target Hamiltonian, in search for its ground state (or an approximation of it), encoding the optimal solution.

QA effectively enables the solution of Ising-type problems, which are a class of NP-hard problems [105]. In practice, several combinatorial optimization problems can be reformulated either in Ising form, using spin variables in the $\{-1, 1\}$ basis, or as quadratic unconstrained binary optimization (QUBO) problems, in the $\{0, 1\}$ basis [161]. These two formulations are equivalent and, thus, represent a suitable format for quantum annealers.

Recently, QA has gained increasing attention thanks to several industrial efforts that have developed small and intermediate-scale quantum processors for programmable use via cloud-based access [161]. As a consequence, several research efforts have employed QA to tackle real-world optimization problems, ranging within the domains of optimization [161], scheduling [99], machine learning [109], and others.

In this context, QA may represent a promising direction also within the field of optimal control. In the recent literature, only few works have tried to employ QA for solving optimal control problems [75, 112]. To date, a comprehensive framework allowing to solve nonlinear optimal control problems via QA is still lacking.

The link between NMPC and quantum computing could be established through the Koopman NMPC (K-NMPC) framework developed in this thesis. The K-NMPC approach lifts NMPC problems in the Koopman space, obtaining an equivalent lifted QP-MPC form, which closely approximates the original NMPC solution. This QP formulation constitutes a crucial intermediate step, as it can be conveniently approximated into a QUBO problem [75], matching the format required by quantum annealers. Hence, K-NMPC represents a promising approach to bridge NMPC and quantum computing, possibly enabling a dramatic speed-up in the solution of nonlinear optimal control problems, which would not be reachable by traditional computational approaches.

References

[1] M. Acquarone, F. Miretti, D. Misul, and L. Sassara, "Cooperative Adaptive Cruise Control Based on Reinforcement Learning for Heavy-Duty BEVs," *IEEE Access*, vol. 11, pp. 127 145–127 156, Nov. 2023.

[2] T. Albash and D. A. Lidar, "Adiabatic quantum computation," *Reviews of Modern Physics*, vol. 90, no. 1, 2018.

[3] M. E. Albira and M. A. Zohdy, "Adaptive Model Predictive Control for DC-DC Power Converters With Parameters' Uncertainties," *IEEE Access*, vol. 9, pp. 135 121–135 131, Sep. 2021.

[4] A. Alessio and A. Bemporad, "A Survey on Explicit Model Predictive Control," in *Nonlinear Model Predictive Control: Towards New Challenging Applications*, ser. Lecture Notes in Control and Information Sciences, L. Magni, D. M. Raimondo, and F. Allgöwer, Eds. Berlin, Heidelberg, Germany: Springer, 2009, pp. 345–369.

[5] D. J. Alford-Lago, C. W. Curtis, A. T. Ihler, and O. Issan, "Deep learning enhanced dynamic mode decomposition," *Chaos*, vol. 32, no. 3, 2022.

[6] M. Amer, A. Abuelnasr, M. Ali, A. Hassan, A. Trigui, A. Ragab, M. Sawan, and Y. Savaria, "Enhanced Dynamic Regulation in Buck Converters: Integrating Input-Voltage Feedforward With Voltage-Mode Feedback," *IEEE Access*, vol. 12, pp. 7310–7328, Jan. 2024.

[7] Analog Devices, "LTSpice," analog.com/ltspice.

[8] J. A. E. Andersson, J. Gillis, G. Horn, J. B. Rawlings, and M. Diehl, "CasADi: a software framework for nonlinear optimization and optimal control," *Mathematical Programming Computation*, vol. 11, pp. 1–36, Mar. 2019.

[9] D. Angeli, R. Amrit, and J. B. Rawlings, "On Average Performance and Stability of Economic Model Predictive Control," *IEEE Transactions on Automatic Control*, vol. 57, no. 7, pp. 1615–1626, 2012.

[10] B. Apolloni, N. Cesa-Bianchi, and D. De Falco, "A Numerical Implementation of "Quantum Annealing"," in *Proceedings of the International Conference on Stochastic Processes, Physics and Geometry*, 1990, pp. 97–111.

[11] F. Arute, K. Arya, R. Babbush, D. Bacon, J. C. Bardin, R. Barends *et al.*, "Quantum supremacy using a programmable superconducting processor," *Nature*, vol. 574, no. 7779, pp. 505–510, 2019.

[12] H. Asere, C. Lei, and R. Jia, "Cruise Control Design Using Fuzzy Logic Controller," in *Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics*, 2015, pp. 2210–2215.

[13] A. Bemporad, "A Multiparametric Quadratic Programming Algorithm With Polyhedral Computations Based on Nonnegative Least Squares," *IEEE Transactions on Automatic Control*, vol. 60, no. 11, pp. 2892–2903, 2015.

[14] ——, "Explicit Model Predictive Control," in *Encyclopedia of Systems and Control*, 2nd ed., J. Baillieul and T. Samad, Eds. Cham, Switzerland: Springer, 2021, pp. 744–751.

[15] A. Bemporad, K. Fukuda, and F. D. Torrisi, "Convexity recognition of the union of polyhedra," *Computational Geometry*, vol. 18, no. 3, pp. 141–154, 2001.

[16] A. Bemporad and M. Morari, "Control of systems integrating logic, dynamics, and constraints," *Automatica*, vol. 35, no. 3, pp. 407–427, 1999.

[17] A. Bemporad, M. Morari, V. Dua, and E. N. Pistikopoulos, "The explicit solution of model predictive control via multiparametric quadratic programming," in *Proceedings of the American Control Conference*, 2000, pp. 872–876.

[18] ——, "The explicit linear quadratic regulator for constrained systems," *Automatica*, vol. 38, no. 1, pp. 3–20, 2002.

[19] D. P. Bertsekas, *Nonlinear Programming*, 3rd ed. Nashua, NH, USA: Athena Scientific, 2016.

[20] M. Boggio, C. Novara, and M. Taragna, "Trajectory planning and control for autonomous vehicles: a "fast" data-aided NMPC approach," *European Journal of Control*, vol. 74, Nov. 2023.

[21] H. Borhan, A. Vahidi, A. M. Phillips, M. L. Kuang, I. V. Kolmanovsky, and S. Di Cairano, "MPC-Based Energy Management of a Power-Split Hybrid Electric Vehicle," *IEEE Transactions on Control Systems Technology*, vol. 20, no. 3, pp. 593–603, 2011.

[22] F. Borrelli, A. Bemporad, and M. Morari, *Predictive Control for Linear and Hybrid Systems*. Cambridge, UK: Cambridge University Press, 2017.

[23] S. P. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge, UK: Cambridge University Press, 2004.

[24] R. K. Brayton, G. D. Hachtel, C. McMullen, and A. Sangiovanni-Vincentelli, *Logic Minimization Algorithms for VLSI Synthesis*. New York, NY, USA: Springer, 1984.

[25] M. Brosowsky, F. Keck, J. Ketterer, S. Isele, D. Slieter, and M. Zöllner, "Safe Deep Reinforcement Learning for Adaptive Cruise Control by Imposing State-Specific Safe Sets," in *Proceedings of the IEEE Intelligent Vehicles Symposium*, 2021, pp. 488–495.

[26] S. L. Brunton, B. W. Brunton, J. L. Proctor, and J. N. Kutz, "Koopman Invariant Subspaces and Finite Linear Representations of Nonlinear Dynamical Systems for Control," *PLoS One*, vol. 11, no. 2, 2016.

[27] S. L. Brunton, M. Budišić, E. Kaiser, and J. N. Kutz, "Modern Koopman Theory for Dynamical Systems," *SIAM Review*, vol. 64, no. 2, pp. 229–340, 2022.

[28] F. Bullo, *Contraction Theory for Dynamical Systems*, 1st ed. Kindle Direct Publishing, 2024. [Online]. Available: fbullo.github.io/ctds

[29] J. C. Butcher, "On Runge-Kutta processes of high order," *Journal of the Australian Mathematical Society*, vol. 4, no. 2, pp. 179–194, 1964.

[30] R. Cagienard, P. Grieder, E. C. Kerrigan, and M. Morari, "Move blocking strategies in receding horizon control," *Journal of Process Control*, vol. 17, no. 6, pp. 563–570, 2007.

[31] L. Calogero, M. Boggio, C. Novara, and A. Rizzo, "A General Analytical Framework for Fast Solving Nonlinear MPC Problems in the Linear Koopman Space," in *Proceedings of the European Control Conference*, 2025.

[32] L. Calogero, M. Mammarella, and F. Dabbene, "Learning Model Predictive Control for Quadrotors Minimum-Time Flight in Autonomous Racing Scenarios," *IFAC-PapersOnLine*, vol. 56, no. 2, pp. 1063–1068, 2023.

[33] L. Calogero, M. Pagone, F. Cianflone, E. Gandino, C. Karam, and A. Rizzo, "Neural Adaptive MPC with Online Metaheuristic Tuning for Power Management in Fuel Cell Hybrid Electric Vehicles," *IEEE Transactions on Automation Science and Engineering*, vol. 22, pp. 11 540–11 553, Jan. 2025.

[34] L. Calogero, M. Pagone, C. Novara, and A. Rizzo, "Economic Nonlinear MPC for Conflicting Control Objectives: The Case of Adaptive Cruise Control," in *Proceedings of the IEEE Conference on Decision and Control*, 2025.

[35] L. Calogero, M. Pagone, and A. Rizzo, "Enhanced Quadratic Programming via Pseudo-Transient Continuation: An Application to Model Predictive Control," *IEEE Control Systems Letters*, vol. 8, pp. 1661–1666, June 2024.

[36] G. Casella and R. L. Berger, *Statistical Inference*, 2nd ed. Pacific Grove, CA, USA: Duxbury Press, 2001.

[37] V. Cerone, S. M. Fosson, S. Pirrera, and D. Regruto, "A new framework for constrained optimization via feedback control of Lagrange multipliers," *IEEE Transactions on Automatic Control*, 2025.

[38] H. Chen and F. Allgöwer, "Nonlinear Model Predictive Control Schemes with Guaranteed Stability," in *Nonlinear Model Based Process Control*, R. Berber and C. Kravaris, Eds. Dordrecht, Netherlands: Springer, 1998, pp. 465–494.

[39] J. Chen, A. Behal, Z. Li, and C. Li, "Active Battery Cell Balancing by Real-Time Model Predictive Control for Extending Electric Vehicle Driving Range," *IEEE Transactions on Automation Science and Engineering*, vol. 21, no. 3, pp. 4003–4015, 2023.

[40] S. Chen, K. Saulnier, N. Atanasov, D. D. Lee, V. Kumar, G. J. Pappas, and M. Morari, "Approximating Explicit Model Predictive Control Using Constrained Neural Networks," in *Proceedings of the American control conference*, 2018, pp. 1520–1527.

[41] W. Chen, A. Q. Huang, C. Li, G. Wang, and W. Gu, "Analysis and Comparison of Medium Voltage High Power DC/DC Converters for Offshore Wind Energy Systems," *IEEE Transactions on Power Electronics*, vol. 28, no. 4, pp. 2014–2023, 2013.

[42] G. Cimini, R. Felicetti, F. Ferracuti, L. Cavanini, and A. Monteriù, "Adaptive Reference Governor for DC–DC Converters Based on Model Predictive Control," *IEEE Transactions on Control Systems Technology*, 2025.

[43] L. H. Csekő, M. Kvasnica, and B. Lantos, "Explicit MPC-Based RBF Neural Network Controller Design With Discrete-Time Actual Kalman Filter for Semiactive Suspension," *IEEE Transactions on Control Systems Technology*, vol. 23, no. 5, pp. 1736–1753, 2015.

[44] C. R. Cutler and B. L. Ramaker, "Dynamic matrix control–A computer control algorithm," in *Proceedings of the Joint Automatic Control Conference*, 1980, pp. 13–15.

[45] J. W. Daniel, "Stability of the Solution of Definite Quadratic Programs," *Mathematical Programming*, vol. 5, no. 1, pp. 41–53, 1973.

[46] A. De Luca, G. Oriolo, and M. Vendittelli, "Control of Wheeled Mobile Robots: An Experimental Overview," in *Articulated and Mobile Robotics for Services and Technology*, ser. Lecture Notes in Control and Information Sciences, S. Nicosia, B. Siciliano, A. Bicchi, and P. Valigi, Eds. Berlin, Heidelberg, Germany: Springer, Jan. 2002, pp. 181–226.

[47] G. De Nicolao, L. Magni, and R. Scattolini, "Stabilizing receding-horizon control of nonlinear time-varying systems," *IEEE Transactions on Automatic Control*, vol. 43, no. 7, pp. 1030–1036, 1998.

[48] M. Diehl, R. Amrit, and J. B. Rawlings, "A Lyapunov Function for Economic Optimizing Model Predictive Control," *IEEE Transactions on Automatic Control*, vol. 56, no. 3, pp. 703–707, 2011.

[49] G. Dinca and J. Mawhin, *Brouwer Degree: The Core of Nonlinear Analysis*. Cham, Switzerland: Birkhäuser, 2021.

[50] R. W. Erickson and D. Maksimović, *Fundamentals of Power Electronics*, 3rd ed. Cham, Switzerland: Springer, 2020.

[51] M. Essoufi, B. Hajji, and A. Rabhi, "Fuzzy Logic Based Energy Management Strategy for Fuel Cell Hybrid Electric Vehicle," in *Proceedings of the International Conference on Electrical and Information Technologies*, 2020.

[52] A. Ferramosca, D. Limon, and E. Camacho, "Economic MPC for a Changing Economic Criterion for Linear Systems," *IEEE Transactions on Automatic Control*, vol. 59, no. 10, pp. 2657–2667, 2014.

[53] R. P. Feynman, "Simulating Physics with Computers," *Int. J. Theor. Phys.*, vol. 21, no. 6/7, pp. 467–488, 1982.

[54] C. Folkestad and J. W. Burdick, "Koopman NMPC: Koopman-based Learning and Nonlinear Model Predictive Control of Control-affine Systems," in *Proceedings of the IEEE International Conference on Robotics and Automation*, 2021, pp. 7350–7356.

[55] C. E. Garcia, D. M. Prett, and M. Morari, "Model predictive control: Theory and practice–A survey," *Automatica*, vol. 25, no. 3, pp. 335–348, 1989.

[56] M. Gendreau and J.-Y. Potvin, Eds., *Handbook of Metaheuristics*. Cham, Switzerland: Springer, 2019.

[57] T. Geyer, F. D. Torrisi, and M. Morari, "Optimal complexity reduction of polyhedral piecewise affine systems," *Automatica*, vol. 44, no. 7, pp. 1728–1740, 2008.

[58] K. Gokcesu and H. Gokcesu, "Generalized Huber Loss for Robust Learning and its Efficient Minimization for a Robust Statistics," *arXiv:2108.12627 [stat.ML]*, Aug. 2021.

[59] D. Goswami and D. A. Paley, "Bilinearization, Reachability, and Optimal Control of Control-Affine Nonlinear Systems: A Koopman Spectral Approach," *IEEE Transactions on Automatic Control*, vol. 67, no. 6, pp. 2715–2728, 2021.

[60] H. Gouk, E. Frank, B. Pfahringer, and M. J. Cree, "Regularisation of Neural Networks by Enforcing Lipschitz Continuity," *Machine Learning*, vol. 110, pp. 393–416, Feb. 2021.

[61] P. Grieder and M. Morari, "Complexity Reduction of Receding Horizon Control," in *Proceedings of the IEEE Conference on Decision and Control*, 2003, pp. 3179–3190.

[62] G. Grimm, M. J. Messina, S. E. Tuna, and A. R. Teel, "Model predictive control: for want of a local control Lyapunov function, all is not lost," *IEEE Transactions on Automatic Control*, vol. 50, no. 5, pp. 546–558, 2005.

[63] S. Gros, M. Zanon, R. Quirynen, A. Bemporad, and M. Diehl, "From linear to nonlinear MPC: bridging the gap via the real-time iteration," *International Journal of Control*, vol. 93, no. 1, pp. 62–80, 2020.

[64] L. Grüne and J. Pannek, *Nonlinear Model Predictive Control: Theory and Algorithms*, 2nd ed. Cham, Switzerland: Springer, 2017.

[65] L. Grüne, J. Pannek, M. Seehafer, and K. Worthmann, "Analysis of Unconstrained Nonlinear MPC Schemes with Time Varying Control Horizon," *SIAM Journal on Control and Optimization*, vol. 48, no. 8, pp. 4938–4962, 2010.

[66] L. Guzzella and A. Amstutz, "CAE Tools for Quasi-Static Modeling and Optimization of Hybrid Powertrains," *IEEE Transactions on Vehicular Technology*, vol. 48, no. 6, pp. 1762–1769, 1999.

[67] L. Guzzella and A. Sciarretta, *Vehicle Propulsion Systems: Introduction to Modeling and Optimization*. Berlin, Heidelberg, Germany: Springer, 2012.

[68] B. He, H. Li, M. Chen, Y. Li, Y. Liu, and Y. Zhao, "An Input-Output Regulated Adaptive Ramp for Fast Load Transition of PWM Buck Convertor," in *Proceedings of the IEEE International Symposium on Circuits and Systems*, 2022, pp. 3566–3570.

[69] H. He, R. Xiong, K. Zhao, and Z. Liu, "Energy Management Strategy Research on a Hybrid Power System by Hardware-In-Loop Experiments," *Applied Energy*, vol. 112, pp. 1311–1317, Dec. 2013.

[70] H. Hemi, J. Ghouili, and A. Cheriti, "A Real Time Fuzzy Logic Power Management Strategy for a Fuel Cell Vehicle," *Energy Conversion and Management*, vol. 80, pp. 63–70, Apr. 2014.

[71] M. Herceg, M. Kvasnica, C. N. Jones, and M. Morari, "Multi-Parametric Toolbox 3.0," in *Proceedings of the European Control Conference*, 2013, pp. 502–510.

[72] K. Hornik, M. Stinchcombe, and H. White, "Multilayer Feedforward Networks Are Universal Approximators," *Neural Networks*, vol. 2, no. 5, pp. 359–366, 1989.

[73] Y.-C. Hsu, C.-Y. Ting, L.-S. Hsu, J.-Y. Lin, and C. C.-P. Chen, "A Transient Enhancement DC–DC Buck Converter With Dual Operating Modes Control Technique," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 66, no. 8, pp. 1376–1380, 2019.

[74] L. C. Iacob, R. Tóth, and M. Schoukens, "Koopman form of nonlinear systems with inputs," *Automatica*, vol. 162, Apr. 2024.

[75] D. Inoue and H. Yoshida, "Model Predictive Control for Finite Input Systems using the D-Wave Quantum Annealer," *Scientific Reports*, vol. 10, Jan. 2020.

[76] V. K. Kasimalla and V. Velisala, "A Review on Energy Allocation of Fuel Cell/Battery/Ultracapacitor for Hybrid Electric Vehicles," *International Journal of Energy Research*, vol. 42, no. 14, pp. 4263–4283, 2018.

[77] H. K. Khalil, *Nonlinear Systems*. Upper Saddle River, NJ, USA: Prentice Hall, 2002.

[78] D. P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," *arXiv:1412.6980 [cs.LG]*, Jan. 2017.

[79] M. Korda and I. Mezić, "Linear predictors for nonlinear dynamical systems: Koopman operator meets model predictive control," *Automatica*, vol. 93, pp. 149–160, 2018.

[80] A. Korompili, O. Ekin, M. Stevic, V. Hagenmeyer, and A. Monti, "Linear Active Disturbance Rejection Control-Based Voltage Controller for Buck and Boost DC/DC Converters in DC Distribution Grids," *IEEE Access*, vol. 13, pp. 19 085–19 109, Jan. 2025.

[81] B. Kouvaritakis and M. Cannon, *Model Predictive Control: Classical, Robust and Stochastic*. Cham, Switzerland: Springer, 2016.

[82] K. Kowalski and W.-H. Steeb, *Nonlinear Dynamical Systems And Carleman Linearization*. Singapore: World Scientific, 1991.

[83] O. Kramer, *Genetic Algorithm Essentials*. Cham, Switzerland: Springer, 2017.

[84] S. S. P. Kumar, A. Tulsyan, B. Gopaluni, and P. Loewen, "A Deep Learning Architecture for Predictive Control," *IFAC-PapersOnLine*, vol. 51, no. 18, pp. 512–517, 2018.

[85] M. Kvasnica, J. Hledík, I. Rauová, and M. Fikar, "Complexity reduction of explicit model predictive control via separation," *Automatica*, vol. 49, no. 6, pp. 1776–1781, 2013.

[86] S. Kwon, A. Tomonaga, G. Lakshmi Bhai, S. J. Devitt, and J.-S. Tsai, "Gate-based superconducting quantum computing," *Journal of Applied Physics*, vol. 129, no. 4, 2021.

[87] E. Landau, *Handbuch der Lehre von der Verteilung der Primzahlen*. Berlin, Germany: Leipzig B.G. Teubner, 1909.

[88] E. B. Lee and L. Markus, *Foundations of Optimal Control Theory*. New York, NY, USA: Wiley, 1967.

[89] Y.-H. Lee, S.-C. Huang, S.-W. Wang, and K.-H. Chen, "Fast Transient (FT) Technique With Adaptive Phase Margin (APM) for Current Mode DC-DC Buck Converters," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 20, no. 10, pp. 1781–1793, 2012.

[90] E. L. Lehmann and G. Casella, *Theory of Point Estimation*, 2nd ed. New York, NY, USA: Springer, 1998.

[91] A. M. Lepschy, G. A. Mian, and U. Viaro, "Feedback control in ancient water and mechanical clocks," *IEEE Transactions on Education*, vol. 35, no. 1, pp. 3–10, 2002.

[92] M. Leshno, V. Y. Lin, A. Pinkus, and S. Schocken, "Multilayer Feedforward Networks with a Nonpolynomial Activation Function Can Approximate Any Function," *Neural Networks*, vol. 6, no. 6, pp. 861–867, 1993.

[93] B. Li, K. Wang, and Z. Shao, "Time-Optimal Maneuver Planning in Automatic Parallel Parking Using a Simultaneous Dynamic Optimization Approach," *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 11, pp. 3263–3274, 2016.

[94] H. Li and Y. Shi, "Event-triggered robust model predictive control of continuous-time nonlinear systems," *Automatica*, vol. 50, no. 5, pp. 1507–1513, 2014.

[95] Z. Liu, L. Xie, A. Bemporad, and S. Lu, "Fast Linear Parameter Varying Model Predictive Control of Buck DC-DC Converters Based on FPGA," *IEEE Access*, vol. 6, pp. 52 434–52 446, Sep. 2018.

[96] J. Löfberg, "YALMIP: A toolbox for modeling and optimization in MATLAB," in *Proceedings of the IEEE International Conference on Robotics and Automation*, 2004, pp. 284–289.

[97] K. Lundahl, K. Berntorp, B. Olofsson, J. Åslund, and L. Nielsen, "Studying the Influence of Roll and Pitch Dynamics in Optimal Road-Vehicle Maneuvers," in *Proceedings of the International Symposium on Dynamics of Vehicles on Roads and Tracks*, 2013.

[98] B. Lusch, J. N. Kutz, and S. L. Brunton, "Deep learning for universal linear embeddings of nonlinear dynamics," *Nature Communications*, vol. 9, no. 1, 2018.

[99] V. Marchioli, M. Boggio, D. Volpe, L. Massotti, and C. Novara, "Quantum Optimization for Closed-Loop Scheduling of Earth Observation Satellite Formation," *SN Computer Science*, vol. 6, Aug. 2025.

[100] S. Mariethoz, S. Almer, M. Baja, A. G. Beccuti, D. Patino, A. Wernrud, J. Buisson, H. Cormerais, T. Geyer, H. Fujioka, U. T. Jonsson, C.-Y. Kao, M. Morari, G. Papafotiou, A. Rantzer, and P. Riedinger, "Comparison of Hybrid Control Techniques for Buck and Boost DC-DC Converters," *IEEE Transactions on Control Systems Technology*, vol. 18, no. 5, pp. 1126–1145, 2010.

[101] A. Mauroy, I. Mezić, and Y. Susuki, *The Koopman Operator in Systems and Control: Concepts, Methodologies, and Applications*. Cham, Switzerland: Springer, 2020.

[102] D. Q. Mayne and P. Falugi, "Generalized Stabilizing Conditions for Model Predictive Control," *Journal of Optimization Theory and Applications*, vol. 169, no. 3, pp. 719–734, 2016.

[103] D. Q. Mayne and H. Michalska, "Receding horizon control of nonlinear systems," *IEEE Transactions on Automatic Control*, vol. 35, no. 7, pp. 814–824, 1990.

[104] D. Q. Mayne, J. B. Rawlings, C. V. Rao, and P. O. M. Scokaert, "Constrained model predictive control: Stability and optimality," *Automatica*, vol. 36, no. 6, pp. 789–814, 2000.

[105] B. M. McCoy and T. T. Wu, *The Two-Dimensional Ising Model*. Mineola, NY, USA: Dover Publications, 2014.

[106] Z. Mehraban, A. Y. Zadeh, H. Khayyam, R. Mallipeddi, and A. Jamali, "Fuzzy adaptive cruise control with model predictive control responding to dynamic traffic conditions for automated driving," *Engineering Applications of Artificial Intelligence*, vol. 136, Oct. 2024.

[107] S. Mirjalili, J. Song Dong, and A. Lewis, Eds., *Nature-Inspired Optimizers: Theories, Literature Reviews and Applications*. Cham, Switzerland: Springer, 2020.

[108] MOSEK ApS, *The MOSEK Optimization Toolbox for MATLAB. Version 10.1*, 2024. [Online]. Available: docs.mosek.com/latest/toolbox/index.html

[109] A. Mott, J. Job, J.-R. Vlimant, D. Lidar, and M. Spiropulu, "Solving a Higgs optimization problem with quantum annealing for machine learning," *Nature*, vol. 550, no. 7676, pp. 375–379, 2017.

[110] NREL DriveCAT: Drive Cycle Analysis Tool. National Renewable Energy Laboratory. [Online]. Available: nrel.gov/transportation/drive-cycle-tool

[111] J. E. Normey-Rico and E. F. Camacho, *Control of Dead-time Processes*. London, UK: Springer, 2007.

[112] C. Novara, M. Boggio, and D. Volpe, "A quantum optimization approach to Nonlinear Model Predictive Control," in *Proceedings of the European Control Conference*, 2025, pp. 2810–2815.

[113] R. Oberdieck, N. A. Diangelakis, and E. N. Pistikopoulos, "Explicit model predictive control: A connected-graph approach," *Automatica*, vol. 76, pp. 103–112, Feb. 2017.

[114] S. Onori, L. Serrao, and G. Rizzoni, "Adaptive Equivalent Consumption Minimization Strategy for Hybrid Electric Vehicles," in *Proceedings of the ASME Dynamic Systems and Control Conference*, 2010, pp. 499–505.

[115] ——, *Hybrid Electric Vehicles: Energy Management Strategies*. London, UK: Springer, 2016.

[116] H. B. Pacejka and E. Bakker, "The Magic Formula Tyre Model," *Vehicle System Dynamics*, vol. 21, no. 1, pp. 1–18, 1992.

[117] Y. Pan and J. Wang, "Model Predictive Control of Unknown Nonlinear Dynamical Systems Based on Recurrent Neural Networks," *IEEE Transactions on Industrial Electronics*, vol. 59, no. 8, pp. 3089–3101, 2011.

[118] W. Pananurak, S. Thanok, and M. Parnichkun, "Adaptive cruise control for an intelligent vehicle," in *Proceedings of the IEEE International Conference on Robotics and Biomimetics*, 2008, pp. 1794–1799.

[119] W. Park, G. Namgoong, E. Choi, and F. Bien, "An On-/Off-Time Sensing-Based Load-Adaptive Mode Control of Triple Mode Buck Converter for Implantable Medical Devices," *IEEE Transactions on Biomedical Circuits and Systems*, vol. 17, no. 3, pp. 585–597, 2023.

[120] K. Patan, "Neural Network-Based Model Predictive Control: Fault Tolerance and Stability," *IEEE Transactions on Control Systems Technology*, vol. 23, no. 3, pp. 1147–1155, 2014.

[121] P. Patrinos and H. Sarimveis, "A new algorithm for solving convex parametric quadratic programs based on graphical derivatives of solution mappings," *Automatica*, vol. 46, no. 9, pp. 1405–1418, 2010.

[122] M. Pavlovský, G. Guidi, and A. Kawamura, "Assessment of coupled and independent phase designs of interleaved multiphase buck/boost DC-DC converter for EV power train," *IEEE Transactions on Power Electronics*, vol. 29, no. 6, pp. 2693–2704, 2013.

[123] D. F. Pereira, F. da Costa Lopes, and E. H. Watanabe, "Nonlinear Model Predictive Control for the Energy Management of Fuel Cell Hybrid Electric Vehicles in Real Time," *IEEE Transactions on Industrial Electronics*, vol. 68, no. 4, pp. 3213–3223, 2020.

[124] S. Pirrera, L. Calogero, F. Gabriele, D. Regruto, A. Rizzo, and G. Setti, "A Fully Analog Implementation of Model Predictive Control with Application to Buck Converters," *arXiv:2509.05463 [eess.SY]*, Sep. 2025.

[125] S. Pirrera, F. Gabriele, D. Lena, F. Pareschi, D. Regruto, and G. Setti, "Robust Load Disturbance Rejection in PWM DC-DC Buck Converters," *arXiv:2509.02102 [eess.SY]*, Sep. 2025.

[126] J. L. Proctor, S. L. Brunton, and J. N. Kutz, "Generalizing Koopman Theory to Allow for Inputs and Control," *SIAM Journal on Applied Dynamical Systems*, vol. 17, no. 1, pp. 909–930, 2018.

[127] A. I. Propoi, "Use of linear programming methods for synthesizing sampled-data automatic systems," *Automation and Remote Control*, vol. 24, no. 7, pp. 837–844, 1963.

[128] S. J. Qin and T. A. Badgwell, "An Overview of Nonlinear Model Predictive Control Applications," in *Nonlinear Model Predictive Control*, F. Allgöwer and A. Zheng, Eds. Basel, Switzerland: Springer, 2000, pp. 369–392.

[129] ——, "A survey of industrial model predictive control technology," *Control Engineering Practice*, vol. 11, no. 7, pp. 733–764, 2003.

[130] Y. Qiu, T. Zeng, C. Zhang, G. Wang, Y. Wang, Z. Hu, M. Yan, and Z. Wei, "Progress and Challenges in Multi-Stack Fuel Cell System for High Power Applications: Architecture and Energy Management," *Green Energy and Intelligent Transportation*, vol. 2, no. 2, 2023.

[131] S. Quan, Y.-X. Wang, X. Xiao, H. He, and F. Sun, "Real-Time Energy Management for Fuel Cell Electric Vehicle Using Speed Prediction-Based Model Predictive Control Considering Performance Degradation," *Applied Energy*, vol. 304, Dec. 2021.

[132] R. Rajamani and C. Zhu, "Semi-autonomous adaptive cruise control systems," *IEEE Transactions on Vehicular Technology*, vol. 51, no. 5, pp. 1186–1192, 2002.

[133] J. B. Rawlings, D. Q. Mayne, and M. Diehl, *Model Predictive Control: Theory, Computation, and Design*, 2nd ed. Madison, Wisconsin, USA: Nob Hill Publishing, 2017.

[134] J. B. Rawlings, D. Angeli, and C. N. Bates, "Fundamentals of Economic Model Predictive Control," in *Proceedings of the IEEE Conference on Decision and Control*, 2012, pp. 3851–3861.

[135] J. B. Rawlings, D. Bonne, J. B. Jorgensen, A. N. Venkat, and S. B. Jorgensen, "Unreachable Setpoints in Model Predictive Control," *IEEE Transactions on Automatic Control*, vol. 53, no. 9, pp. 2209–2215, 2008.

[136] H. Rezaee, T. Parisini, and M. M. Polycarpou, "Leaderless Cooperative Adaptive Cruise Control Based on the Constant Time-Gap Spacing Policy," *IEEE Transactions on Automatic Control*, vol. 69, no. 1, pp. 659–666, 2023.

[137] W. Rudin, *Principles of Mathematical Analysis*. New York, USA: McGraw-Hill, 1976.

[138] R. Scattolini, "Architectures for distributed and hierarchical Model Predictive Control–A review," *Journal of Process Control*, vol. 19, no. 5, pp. 723–731, 2009.

[139] R. Schneider, *Convex Bodies: The Brunn–Minkowski Theory*. Cambridge, UK: Cambridge University Press, 2014.

[140] J. C. Schulze and A. Mitsos, "Data-Driven Nonlinear Model Reduction Using Koopman Theory: Integrated Control Form and NMPC Case Study," *IEEE Control Systems Letters*, vol. 6, pp. 2978–2983, 2022.

[141] R. Schwan, C. N. Jones, and D. Kuhn, "Stability Verification of Neural Network Controllers Using Mixed-Integer Programming," *IEEE Transactions on Automatic Control*, vol. 68, no. 12, pp. 7514–7529, 2023.

[142] A. S. Sedra and K. C. Smith, *Microelectronic Circuits*, 7th ed. New York, NY, USA: Oxford University Press, 2016.

[143] A. Steane, "Quantum computing," *Reports on Progress in Physics*, vol. 61, no. 2, p. 117, 1998.

[144] N. Sulaiman, M. A. Hannan, A. Mohamed, E. H. Majlan, and W. R. W. Daud, "A Review on Energy Management System for Fuel Cell Hybrid Electric Vehicle: Issues and Challenges," *Renewable and Sustainable Energy Reviews*, vol. 52, pp. 802–814, Dec. 2015.

[145] JRCSTU WLTP Repository. Sustainable Transport Unit, Institute of Energy and Transport, Joint Research Center, European Commission. [Online]. Available: github.com/JRCSTU/wltp/tree/master/wltp/cycles

[146] N. N. Tam, "On Continuity Properties of the Solution Map in Quadratic Programming," *Acta Mathematica Vietnamica*, vol. 24, no. 1, pp. 47–61, 1999.

[147] E. Terzi, F. Bonassi, M. Farina, and R. Scattolini, "Learning model predictive control with long short-term memory networks," *International Journal of Robust and Nonlinear Control*, vol. 31, no. 18, pp. 8877–8896, 2021.

[148] J. Testud, J. Richalet, A. Rault, and J. Papon, "Model predictive heuristic control: Applications to industial processes," *Automatica*, vol. 14, no. 5, pp. 413–428, 1978.

[149] P. Tøndel, T. A. Johansen, and A. Bemporad, "An algorithm for multi-parametric quadratic programming and explicit MPC solutions," *Automatica*, vol. 39, no. 3, pp. 489–497, 2003.

[150] TurtleBot, *TurtleBot3 Manual*. [Online]. Available: emanual.robotis.com/docs/en/platform/turtlebot3/overview

[151] S. Vasebi, Y. M. Hayeri, and A. M. Saghiri, "A Literature Review of Energy Optimal Adaptive Cruise Control Algorithms," *IEEE Access*, vol. 11, pp. 13 636–13 646, Jan. 2023.

[152] S. Vichik and F. Borrelli, "Solving linear and quadratic programs with an analog circuit," *Computers and Chemical Engineering*, vol. 70, pp. 160–171, Nov. 2014.

[153] A. Wächter and L. T. Biegler, "On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming," *Mathematical Programming*, vol. 106, pp. 25–57, Mar. 2006.

[154] G. R. Walker and P. C. Sernia, "Cascaded DC-DC converter connection of photovoltaic modules," *IEEE Transactions on Power Electronics*, vol. 19, no. 4, pp. 1130–1139, 2004.

[155] J. Wang and R. Rajamani, "Should adaptive cruise-control systems be designed to maintain a constant time gap between vehicles?" *IEEE Transactions on Vehicular Technology*, vol. 53, no. 5, pp. 1480–1490, 2004.

[156] Z. Wang, T. Guo, X. Wang, and S. Li, "GPI observer-based composite current-constrained control approach for DC-DC buck converters," in *Proceedings of the Jiangsu Annual Conference on Automation*, 2018, pp. 581–586.

[157] M. O. Williams, I. G. Kevrekidis, and C. W. Rowley, "A Data-Driven Approximation of the Koopman Operator: Extending Dynamic Mode Decomposition," *Journal of Nonlinear Science*, vol. 25, pp. 1307–1346, 2015.

[158] V. L. Wisniewski, E. T. Maddalena, and R. B. Godoy, "Explicit MPC with Region Elimination: A Case Study on a Buck Converter," in *Proceedings of the Brazilian Power Electronics Conference*, 2021.

[159] W. Xin, E. Xu, W. Zheng, H. Feng, and J. Qin, "Optimal Energy Management of Fuel Cell Hybrid Electric Vehicle Based on Model Predictive Control and On-Line Mass Estimation," *Energy Reports*, vol. 8, pp. 4964–4974, Nov. 2022.

[160] W. Xiong, Z. Wang, B. Zhang, and S. Li, "Robust Voltage Regulation for DC–DC Converters via a Predictive GPIO-Based Control Approach," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 69, no. 12, pp. 4864–4868, 2022.

[161] S. Yarkoni, E. Raponi, T. Bäck, and S. Schmitt, "Quantum annealing for industry applications: Introduction and review," *Reports on Progress in Physics*, vol. 85, no. 10, 2022.

[162] T. Zeng, C. Zhang, Y. Zhang, C. Deng, D. Hao, Z. Zhu, H. Ran, and D. Cao, "Optimization-Oriented Adaptive Equivalent Consumption Minimization Strategy Based on Short-Term Demand Power Prediction for Fuel Cell Hybrid Vehicle," *Energy*, vol. 227, July 2021.

[163] W. Zhang, J. Li, L. Xu, and M. Ouyang, "Optimization for a Fuel Cell/Battery/Capacity Tram with Equivalent Consumption Minimization Strategy," *Energy Conversion and Management*, vol. 134, pp. 59–69, Feb. 2017.

[164] Y. Zhang, C. Huang, H. Huang, and J. Wu, "Multiple Learning Neural Network Algorithm for Parameter Estimation of Proton Exchange Membrane Fuel Cell Models," *Green Energy and Intelligent Transportation*, vol. 2, no. 1, 2023.

[165] J. Zhao, Z. Wang, Y. Lv, J. Na, C. Liu, and Z. Zhao, "Data-Driven Learning for H_∞ Control of Adaptive Cruise Control Systems," *IEEE Transactions on Vehicular Technology*, vol. 73, no. 12, pp. 18 348–18 362, 2024.