POLITECNICO DI TORINO
Repository ISTITUZIONALE

A New Framework for Constrained Optimization via Feedback Control of Lagrange Multipliers

Original
A New Framework for Constrained Optimization via Feedback Control of Lagrange Multipliers / Cerone, V., Fosson, S., Pirrera, S., Regruto, D.. - In: IEEE TRANSACTIONS ON AUTOMATIC CONTROL. - ISSN 0018-9286. - 70:11(2025), pp. 7141-7156. [10.1109/TAC.2025.3568651]

Availability:
This version is available at: 11583/3005455 since: 2025-11-29T10:21:36Z

Publisher:
IEEE-INST ELECTRICAL ELECTRONICS ENGINEERS

Published
DOI:10.1109/TAC.2025.3568651

Terms of use:

This article is made available under terms and conditions as specified in the corresponding bibliographic description in the repository

Publisher copyright
IEEE postprint/Author's Accepted Manuscript

©2025 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collecting works, for resale or lists, or reuse of any copyrighted component of this work in other works.

(Article begins on next page)

20 November 2026

A new framework for constrained optimization via feedback control of Lagrange multipliers

V. Cerone, S. M. Fosson, S. Pirrera, D. Regruto

***Abstract*—The continuous-time analysis of iterative algorithms for optimization has a long-standing history. This work introduces a novel framework for equality-constrained optimization based on control theory. The central concept is to design a feedback control system in which the Lagrange multipliers serve as the control inputs while the output represents the constraints. This system converges to a stationary point of the constrained optimization problem through suitable regulation. Concerning the Lagrange multipliers, we explore two control laws: proportional-integral control and feedback linearization. These choices lead to a variety of different methods. We rigorously develop the related algorithms, analyze their convergence theoretically, and present several numerical experiments that demonstrate their effectiveness compared to the state-of-the-art approaches.**

***Index Terms*—Constrained optimization, continuous-time dynamical systems, feedback control, Lagrange multipliers, proportional-integral control, feedback linearization.**

I. Introduction

First-order iterative algorithms are prevalent in convex and non-convex optimization and machine learning to handle large-scale datasets and leverage parallel processing architectures. Iterative algorithms for optimization are discrete-time (DT) dynamical systems that update the estimate of the optimization variables at each iteration. Their continuous-time (CT) counterparts, obtained by considering infinitesimal step sizes, are described by differential equations whose analysis can provide a deeper theoretical understanding, particularly concerning stability and convergence rate.

A paradigmatic example is the gradient flow, defined by the equation $\dot{x} = -\nabla f(x)$, where $f : \mathbb{R}^n \to \mathbb{R}$ is a differentiable, unconstrained cost function that we aim to minimize. Gradient flow is the CT version of gradient descent and proximal minimization algorithms achieved through forward and backward Euler discretization, respectively; see, e.g., [1, Sec. 4.1.1]. The study of gradient flow is particularly relevant in deep learning, where gradient descent methods have seen practical success in training despite lacking a solid theoretical understanding; see, e.g., [2], [3] and references therein. Additionally, works [4], [5] propose CT analysis of Nesterov accelerations for gradient descent.

In this work, we consider the more challenging problem of constrained optimization. A significant amount of research focuses on CT methods based on Lagrange multipliers, particularly primal-dual methods. For a general overview, we refer the reader to [6, Chapter 15]. The main CT approach to constrained optimization is the primal-dual gradient dynamics (PDGD), which was introduced in [7], [8].

In [9], the authors examine the exponential stability of PDGD when minimizing strongly convex, smooth cost functions with linear equality constraints. This analysis is extended to non-smooth composite optimization in [10], [11] by using a proximal augmented Lagrangian, and to non-convex stochastic optimization in [12]. Regarding convex composite optimization with linear equality constraints, the work [13] illustrates a CT model for the alternating direction method of multipliers (ADMM [14]). Additionally, prior research proposes modifying the gradient flow to account for equality constraints. The key idea in this approach is to build the descent direction $\dot{x}$ as a combination of a projected gradient and a Gauss-Newton direction, which drives the solution toward the feasible set; see, e.g., [15], [16], [17]. Finally, works [16], [17], [9] also consider inequality constraints by extending first-order conditions to Karush-Kuhn-Tucker (KKT) conditions [6].

This brief review emphasizes that much of the literature focuses on the CT analysis of existing algorithms rather than the development of new CT algorithms for optimization. An exception is the recent work [18], which develops CT dynamical systems that solve constrained optimization problems. The authors synthesize a feedback controller, exploiting tools from the theory of control barrier functions (CBF), that guarantees forward invariance and stability of the feasible set. These properties enforce safety when the feasible set represents the safe operation of a plant.

This work introduces a new CT framework for convex and non-convex constrained optimization, focusing on equality constraints. The proposed framework adopts an original feedback control approach: we start from the first-order necessary conditions for minima to build a CT dynamical system where the vector of Lagrange multipliers represents the control input. The output of this system corresponds to the constraints, which we regulate accordingly. Several control laws can be employed for the Lagrange multipliers to achieve the desired regulation, leading to a family of control-based first-order methods.

This work has two main contributions. The first one is developing and analyzing a control-theoretic framework that enables the synthesis of new first-order optimization algorithms using feedback control design techniques. We call this framework controlled multipliers optimization (CMO).

The second contribution is the specialization of CMO to specific control laws. We focus mainly on proportional-integral control and feedback linearization methods, denoted as PI-CMO and FL-CMO, respectively. We develop the corresponding algorithms, we study the conditions for their convergence

The authors are with the Dipartimento di Automatica e Informatica, Politecnico di Torino, corso Duca degli Abruzzi 24, 10129 Torino, Italy; e-mail: vito.cerone@polito.it, sophie.fosson@polito.it, simone.pirrera@polito.it, diego.regruto@polito.it.

in both convex and non-convex settings and the convergence speed. Through this analysis, we highlight the advantages of the proposed algorithms compared to state-of-the-art CT techniques for constrained optimization, with particular attention to references [9], [18]. Finally, we present several numerical experiments demonstrating the effectiveness of PI-CMO and FL-CMO, even beyond their theoretically established convergence conditions. We pay special attention to comparing our approaches to state-of-the-art optimization algorithms regarding solution accuracy and numerical complexity.

We organize the paper as follows. In Section II, we formulate the problem, review the theory of Lagrange multipliers, and describe the proposed control-theoretic framework. The two succeeding sections specialize the framework into two distinct control strategies. Specifically, Section III develops the PI control method, demonstrates its convergence, and analyzes the convergence rate for strongly convex problems with linear constraints. Section IV introduces the feedback linearization method and proves its convergence for both strongly convex and non-convex problems. Section V presents several numerical experiments that illustrate the practical effectiveness of the proposed methods in various applications. Finally, Section VI concludes the paper.

II. Problem statement and proposed framework

We consider the constrained optimization problem

$$\begin{aligned} \min_{x\in\mathbb{R}^n} \ & f(x) \\ \text{s.t.} \ & \\ & h(x) = 0 \end{aligned} \tag{1}$$

where $f : \mathbb{R}^n \mapsto \mathbb{R}$ and $h : \mathbb{R}^n \mapsto \mathbb{R}^m$ are differentiable, possibly non-convex functions.

The Lagrangian of problem (1) is

$$\mathcal{L}(x, \lambda) = f(x) + \lambda^\top h(x) \tag{2}$$

where $\lambda \in \mathbb{R}^m$ is the vector of Lagrange multipliers. We report the following well-known theorem for self-consistency; see, e.g., [6, Sec 11.3] for a complete overview.

Theorem 1 (First-order necessary conditions)**.** *Let $x^\star \in \mathbb{R}^n$ be a local minimum of f such that $h(x^\star) = 0$. Assume that $x^\star$ is regular, i.e., $\nabla h_1(x^\star), \dots, \nabla h_m(x^\star)$ are linearly independent. Then, there exists a unique $\lambda^\star \in \mathbb{R}^m$ such that $(x^\star, \lambda^\star)$ is a saddle point of $\mathcal{L}(x, \lambda)$, i.e,*

$$\nabla f(x^\star) + J_h(x^\star)^\top \lambda^\star = 0 \tag{3}$$

where $J_h(x) \in \mathbb{R}^{m,n}$ is the Jacobian matrix of h evaluated in x.

In the rest of the paper, we call stationary point any couple $(x^\star, \lambda^\star)$, with $x^\star \in \mathbb{R}^n$ and $\lambda^\star \in \mathbb{R}^m$, which satisfies (3) and the constraints, i.e.,

$$\begin{aligned} &\nabla f(x^\star) + J_h(x^\star)^\top \lambda^\star = 0 \\ &h(x^\star) = 0. \end{aligned} \tag{4}$$

In this work, we focus on first-order algorithms, i.e., on methods that achieve stationary points. In the non-convex case, finding a stationary point does not guarantee local optimality; however, the literature is rich in first-order methods in large-scale optimization and machine learning thanks to their low complexity when compared to, e.g., second-order methods, which require computing and storing Hessian matrices as well; see, e.g., [19], [6], [20].

A. Proposed framework: feedback control of Lagrange multipliers

In this subsection, we illustrate the proposed framework to find a stationary point of problem (1) by building a suitable CT dynamical system with controlled Lagrange multipliers.

Let us define the dynamical system $\mathcal{P}$ with state $x(t) \in \mathbb{R}^n$, input $\lambda(t) \in \mathbb{R}^m$ and output $y(t) \in \mathbb{R}^m$, described by the equations

$$\mathcal{P} : \begin{cases} \dot{x}(t) = -\nabla f(x(t)) - J_h(x(t))^\top \lambda(t) \\ y(t) = h(x(t)) \end{cases} \tag{5}$$

The following result holds.

Lemma 1. *An equilibrium point $(x^\star, \lambda^\star)$ of $\mathcal{P}$ is a stationary point of problem* (1) *if and only if $h(x^\star) = 0$.*

Proof. By definition, an equilibrium point $(x^\star, \lambda^\star)$ of $\mathcal{P}$ satisfies equation (3). If $h(x^\star) = 0$, then (4) holds and $(x^\star, \lambda^\star)$ is a stationary point to problem (1). Conversely, any stationary point satisfies (4), thus (3), then it corresponds to an equilibrium point $(x^\star, \lambda^\star)$ of $\mathcal{P}$ with $h(x^\star) = 0$. □

Lemma 1 suggests we can compute a stationary point of problem (1) by designing a suitable input $\lambda(t)$ that drives $\mathcal{P}$ to converge to an equilibrium point and regulates the output to zero. A standard way to approach this regulation problem is to design a suitable feedback controller $\mathcal{K}$. In Fig. 1, we depict a general scheme; $y(t)$ is the feedback signal to the input of $\mathcal{K}$, possibly together with the state of $\mathcal{P}$. We underline that $\mathcal{P}$ is a representation of an optimization algorithm; therefore, the state is known as well as the output, which is different from physical systems where the observation of the state may be critical.

The goal of this work is to tackle the following problem.

Problem 1. *Design a feedback controller $\mathcal{K}$ for $\mathcal{P}$ such that*

$$\begin{aligned} &\lim_{t\to\infty} x(t) = x^\star, \\ &\lim_{t\to\infty} y(t) = 0. \end{aligned} \tag{6}$$

In this work, we consider two possible design techniques for $\mathcal{K}$ to solve Problem 1: PI control and feedback linearization.

B. Related work

To our knowledge, controlling the Lagrange multipliers in constrained optimization is novel. In the literature, the use of control methods to analyze or develop optimization algorithms is only partly explored.

Regarding first-order unconstrained optimization, the works [21] and [22] present a control theory interpretation of known algorithms such as gradient descent, heavy-ball, and Nesterov's accelerated methods. In particular, they show that these

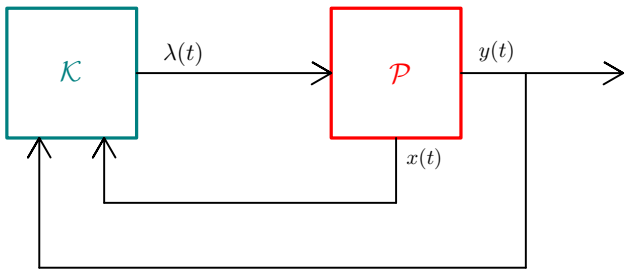


Fig. 1. Structure of the proposed feedback control approach. The state and the output of $\mathcal{P}$ defined in (5) are fed back to the controller $\mathcal{K}$, whose output is the vector of the Lagrange multipliers.

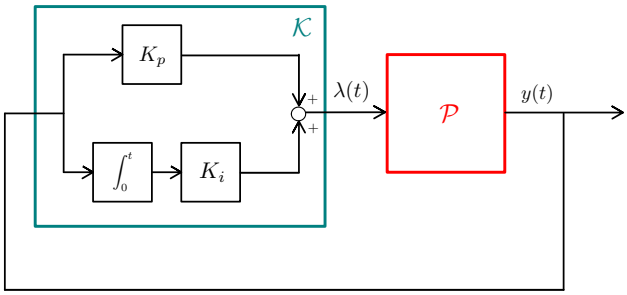


Fig. 2. First method: PI-CMO. We feedback the output $y(t)$ to the controller $\mathcal{K}$, which applies proportional an integral actions.

algorithms correspond to DT feedback systems, where the current gradient is the control input. By using integral quadratic constraints, they analyze their convergence. Unlike our work, this framework does not envisage constrained optimization.

As mentioned in the introduction, the work [18] proposes a CT algorithm based on CBF that solves equality and inequality constrained optimization. The proposed method can also be interpreted as a continuous approximation of a projected gradient flow [23], [24], which is used to analyze the stability.

A different research line studies the solution of equations via control. In the DT framework, [25] and [26] solve linear algebraic equations using iterative learning control and observer-based controller design, respectively. Instead, in [27], the authors build a CT-controlled system whose output is regulated to solve $g(x) = 0$, where $g : \mathbb{R}^n \mapsto \mathbb{R}^n$ is a vector function. By choosing appropriate control Lyapunov functions, they retrace standard iterative methods, such as Newton-Raphson and conjugate gradient methods, and develop new variants. Unlike our work, this framework considers equality-constrained problems with no cost function to minimize. However, one may argue that finding a stationary point corresponds to solving the first-order equations $(\nabla f(x) + \sum_{i=i}^{m} \lambda_i \nabla h_i(x))^\top$ together with $h(x) = 0$ in the variables x and λ, i.e., a system of dimension $(n + m) \times (n + m)$. In other terms, one can apply the methods proposed in [27] to find the zeros of the vector function $g(x, \lambda) = \left[(\nabla f(x) + \sum_{i=i}^{m} \lambda_i \nabla h_i(x))^\top, h(x)^\top\right]^\top$. However, as summarized in [27, Table 1], this approach gives rise to second-order methods and continuous Newton algorithms, which require the inversion of the Jacobian of $g(x, \lambda)$. Therefore, the numerical complexity is prohibitive for large-scale problems.

Beyond the control-theoretic approach, PDGD [9] is a valuable CT approach to solve problem (1). In the next section, we compare our approach to PDGD.

III. First method: PI-CMO

A possible strategy to design $\mathcal{K}$ for Problem 1 is to apply a PI action on $y(t) = h(x(t))$. The PI control is widespread in industrial and engineering applications, thanks to its effectiveness in regulating many processes by tuning two parameters.

In our framework, the PI control law is as follows:

$$\lambda(t) = K_p y(t) + K_i \int_0^t y(\tau)\, \mathrm{d}\tau \tag{7}$$

where $K_p \in \mathbb{R}$ and $K_i \in \mathbb{R}$ are the coefficients of the proportional and integral terms, respectively. We depict the corresponding feedback scheme in Fig. 2.

As a consequence, $\mathcal{K}$ is a dynamical system described by the differential equation

$$\begin{aligned} \dot{\lambda}(t) &= K_p \frac{\mathrm{d}}{\mathrm{d}t} y(t) + K_i y(t) \\ &= K_p J_h(x(t)) \dot{x}(t) + K_i y(t). \end{aligned} \tag{8}$$

In the following, we drop the variable t in long formulas, namely $x = x(t)$, $\dot{x} = \dot{x}(t)$, $\lambda = \lambda(t)$ and $\dot{\lambda} = \dot{\lambda}(t)$. Given the definition of $\dot{x}$ in (5), the closed-loop dynamics is

$$\begin{aligned} \dot{x} &= -\nabla f(x) - J_h(x)^\top \lambda \\ \dot{\lambda} &= -K_p J_h(x) \left[\nabla f(x) + J_h(x)^\top \lambda\right] + K_i h(x). \end{aligned} \tag{9}$$

or, equivalently,

$$\begin{aligned} \dot{x} &= -\nabla_x \mathcal{L}(x, \lambda) \\ \dot{\lambda} &= -K_p J_h(x) \nabla_x \mathcal{L}(x, \lambda) + K_i \nabla_\lambda \mathcal{L}(x, \lambda). \end{aligned} \tag{10}$$

In the following, we refer to the dynamics (9) as to PI-CMO dynamics. For any equilibrium point $(x^\star, \lambda^\star)$, from (9) we get the condition $h(x^\star) = 0$. Therefore, according to Lemma 1, each equilibrium point corresponds to a stationary point in the case of PI control.

Remark 1. *In equation* (7)*, we consider scalar K_p and K_i to reduce the number of design parameters and simplify the convergence analysis reported in section III-A. Extension to the most general case where K_p and K_i are matrix gains in $\mathbb{R}^{m,m}$ is straightforward.*

Remark 2. *It is worth noticing that by considering a purely integral control, i.e., $K_p = 0$, we obtain the dynamics*

$$\begin{aligned} \dot{x} &= -\nabla_x \mathcal{L}(x, \lambda) \\ \dot{\lambda} &= K_i \nabla_\lambda \mathcal{L}(x, \lambda), \end{aligned} \tag{11}$$

which corresponds to PDGD defined in [9, Eq. (2a)-(2b)], where the symbol η is used instead of K_i. In other words, we can interpret PDGD as applying an integral control to the Lagrange multipliers, which recast PDGD into the proposed control-theoretic framework. In contrast, the proposed PI-CMO extends PDGD to a novel family of CT optimization algorithms.

A. Global exponential convergence of PI-CMO for strongly convex problems

This section proves that the PI-CMO dynamics converges globally and exponentially in a convex setting with affine constraints, using an appropriate Lyapunov function. Additionally, we analyze the convergence rate. We adopt the same assumptions as those in [9] to enable a thorough comparison with PDGD.

Assumption 1. *f is strongly convex and twice differentiable.*

Assumption 2. *h is affine, i.e., $h(x) = Cx + d$, $C \in \mathbb{R}^{m,n}$, $d \in \mathbb{R}^m$. Moreover, C is full rank and there exist $0 < \alpha_1 \leq \alpha_2$ such that*

$$\alpha_1 I \preceq CC^\top \preceq \alpha_2 I. \tag{12}$$

The assumption that C is full rank guarantees that $Cx+d = 0$ has solutions and that there are no linearly dependent constraints. Additionally, under this assumption, any point $x \in \mathbb{R}^n$ is regular. Therefore, all assumptions required by Theorem 1 are satisfied. We define

$$z(t) := (x(t)^\top, \lambda(t)^\top)^\top \tag{13}$$

and

$$z^\star := (x^{\star\top}, \lambda^{\star\top})^\top \tag{14}$$

is the equilibrium point of (9), which corresponds to a saddle point of $\mathcal{L}(x, \lambda)$. Assumptions 1 and 2 guarantee the existence and the uniqueness of $z^\star$.

According to [9, Lemma 1], there exists a symmetric $B(x) \in \mathbb{R}^{n,n}$ satisfying $\beta_1 I \preceq B(x) \preceq \beta_2 I$ for some $0 < \beta_1 < \beta_2$ such that

$$\nabla f(x) - \nabla f(x^\star) = B(x)(x - x^\star). \tag{15}$$

The constants β_1 and β_2 satisfy

$$\beta_1 I \leq \nabla^2 f(x^\star + t(x - x^\star)) \leq \beta_2 I, \quad \forall t \in [0, 1]. \tag{16}$$

where $\nabla^2 f(x)$ is the Hessian matrix of $f(x)$. Equation (16) relates the constants β_1 and β_2 with the second-order information of $f(x)$. If $f(x)$ is quadratic and strongly convex, namely $f(x) = \frac{1}{2}x^\top W x$ with $W \succ 0$, then β_1 and β_2 are the minimum and maximum eigenvalues of W, respectively. If $f(x)$ is not a quadratic function, we can evaluate β_1 and β_2 as outlined in [28], which presents a method for estimating tight bounds on the eigenvalues of $\nabla^2 f(x)$.

In the following, we write $B = B(x)$ for brevity.

Theorem 2 (Global exponential convergence of PI-CMO)**.** *Let assumptions 1 and 2 hold. Given $K_i > 0$ and $K_p > 0$, if*

$$K_p < \frac{2K_i}{\beta_2}, \tag{17}$$

then there exist real positive constants c_1 and c_2 such that

$$\|x(t) - x^\star\|_2 \leq c_1 e^{-\frac{1}{2}\mu t}, \quad \|\lambda(t) - \lambda^\star\|_2 \leq c_2 e^{-\frac{1}{2}\mu t} \tag{18}$$

where

$$\mu = \min\left\{K_p\alpha_1, 2\beta_1 - \frac{K_p}{K_i}\beta_1\beta_2\right\} > 0. \tag{19}$$

Proof. First of all, we define the candidate Lyapunov function

$$V\big(z(t)\big) = \big(z(t) - z^\star\big)^\top P\big(z(t) - z^\star\big) \tag{20}$$

where

$$P := \begin{pmatrix} K_i I_n & 0 \\ 0 & I_m \end{pmatrix} \in \mathbb{R}^{m+n,m+n}. \tag{21}$$

If we prove that

$$\dot{V}\big(z(t)\big) \leq -\mu V\big(z(t)\big) \tag{22}$$

for some $\mu > 0$, then the theorem statement holds. Therefore, in the following, we focus on conditions that guarantee (22).

We start with some preliminary computations. Since $\nabla_x \mathcal{L}(x^\star, \lambda^\star) = 0$, $\nabla_\lambda \mathcal{L}(x^\star, \lambda^\star) = 0$ and $J_h(x) = C$, we have

$$\begin{aligned}\nabla_x \mathcal{L}(x, \lambda) &= \nabla_x \mathcal{L}(x, \lambda) - \nabla_x \mathcal{L}(x^\star, \lambda^\star) \\ &= \nabla f(x) - \nabla f(x^\star) + J_h(x)^\top \lambda - J_h(x^\star)^\top \lambda^\star \\ &= \nabla f(x) - \nabla f(x^\star) + C^\top(\lambda - \lambda^\star)\end{aligned} \tag{23}$$

and

$$\begin{aligned}\nabla_\lambda \mathcal{L}(x, \lambda) &= \nabla_\lambda \mathcal{L}(x, \lambda) - \nabla_\lambda \mathcal{L}(x^\star, \lambda^\star) \\ &= Cx + d - (Cx^\star + d) = C(x - x^\star).\end{aligned} \tag{24}$$

By using (10), (15), (23) and (24), we obtain

$$\begin{aligned}\dot{z}(t) &= \begin{pmatrix} \dot{x}(t) \\ \dot{\lambda}(t) \end{pmatrix} \\ &= \begin{pmatrix} -B(x - x^\star) - C^\top(\lambda - \lambda^\star) \\ K_i C(x - x^\star) - K_p C[B(x - x^\star) + C^\top(\lambda - \lambda^\star)] \end{pmatrix} \\ &= \begin{pmatrix} -B & -C^\top \\ K_i C - K_p CB & -K_p CC^\top \end{pmatrix} \big(z(t) - z^\star\big).\end{aligned} \tag{25}$$

Let us define

$$G := \begin{pmatrix} -B & -C^\top \\ K_i C - K_p CB & -K_p CC^\top \end{pmatrix} \tag{26}$$

so that

$$\dot{z}(t) = G\big(z(t) - z^\star\big).$$

Then,

$$\begin{aligned}\dot{V}\big(z(t)\big) &= \dot{z}(t)^\top P\big(z(t) - z^\star\big) + \big(z(t) - z^\star\big)^\top P\dot{z}(t) \\ &= \big(z(t) - z^\star\big)^\top \big(G^\top P + PG\big)\big(z(t) - z^\star\big).\end{aligned} \tag{27}$$

Therefore, a sufficient condition for $\dot{V}\big(z(t)\big) \leq -\mu V\big(z(t)\big)$, see (22), is

$$-G^\top P - PG - \mu P \succeq 0. \tag{28}$$

As a result, our next goal is to establish sufficient conditions for (28). We compute

$$PG = \begin{pmatrix} -K_i B & -K_i C^\top \\ K_i C - K_p CB & -K_p CC^\top \end{pmatrix} \tag{29}$$

while $G^\top P = (PG)^\top$. Hence,

$$\begin{aligned}&- G^\top P - PG - \mu P \\ &= \begin{pmatrix} 2K_i B - K_i \mu I & K_p BC^\top \\ K_p CB & 2K_p CC^\top - \mu I \end{pmatrix} \\ &\succeq \begin{pmatrix} 2K_i B - K_i \mu I & K_p BC^\top \\ K_p CB & K_p CC^\top \end{pmatrix}\end{aligned} \tag{30}$$

where the last step derives from $2K_pCC^\top - \mu I \succeq CC^\top$ which holds for $\mu \leq K_p\alpha_1$. Since $CC^\top \succ 0$ is invertible from (12) in Assumption 2, we can apply the Schur complement argument to conclude that the matrix

$$\begin{pmatrix} 2K_iB - K_i\mu I & K_pBC^\top \\ K_pCB & K_pCC^\top \end{pmatrix}$$

is positive semidefinite if and only if

$$2K_iB - K_i\mu I - K_pBC^\top \frac{1}{K_p}(CC^\top)^{-1}K_pCB \succeq 0. \quad (31)$$

Moreover, since $CC^\top$ is invertible, then $C^\top(CC^\top)^{-1}C \preceq I$. Therefore, a sufficient condition for (31) is

$$(2K_iI - K_pB)B \succeq K_i\mu I. \quad (32)$$

Under assumption (17), (32) holds if

$$(2K_i - K_p\beta_2)\beta_1 \geq K_i\mu \quad (33)$$

which is equivalent to

$$\mu \leq 2\beta_1 - \frac{K_p}{K_i}\beta_1\beta_2. \quad (34)$$

This completes the proof. □

Remark 3. *According to* (20)-(21)*, the considered Lyapunov function is*

$$V(x,\lambda) = K_i\|x - x^\star\|_2^2 + \|\lambda - \lambda^\star\|_2^2.$$

We highlight the role of the parameter K_i, which tunes the weight assigned to the terms in x relative to the terms in λ. According to Theorem 2, a larger K_i increases the bound on μ, meaning that the weight given to the terms in x must be sufficiently large compared to the terms in λ. We remark that in the convergence proof of PDGD in [9], the Lyapunov function is defined on a non-diagonal matrix P, which complicates the interpretation of the parameters.

According to [9], PDGD is globally exponentially convergent with rate $\frac{1}{2}\mu_{PDGD}$

$$\mu_{PDGD} = \min\left\{\frac{\eta\alpha_1}{4\beta_2}, \frac{\alpha_1\beta_1}{4\alpha_2}\right\} \quad (35)$$

Since η is a design parameter, we can set $\eta \geq \frac{\beta_1\beta_2}{\alpha_2}$ it so that μ_{PDGD} saturates to $\frac{\alpha_1\beta_1}{4\alpha_2}$. The following result holds.

Corollary 1. *For $\epsilon \in \left(0, 1 - \frac{\alpha_1}{8\alpha_2}\right)$, let*

$$K_p = \epsilon\frac{2K_i}{\beta_2} \quad \textit{and} \quad K_i > \frac{1}{\epsilon}\frac{\beta_1\beta_2}{8\alpha_2}. \quad (36)$$

Then

$$\mu > \mu_{PDGD} \quad (37)$$

i.e., PI-CMO enjoys a faster convergence rate with respect to PDGD [9].

Proof. By replacing $K_p = \epsilon\frac{2K_i}{\beta_2}$ in (19), we obtain

$$\mu = \min\left\{2\epsilon K_i\frac{\alpha_1}{\beta_2}, 2\beta_1(1-\epsilon)\right\}.$$

Now, $\epsilon \in \left(0, 1 - \frac{\alpha_1}{8\alpha_2}\right)$ implies $2\beta_1(1-\epsilon) > \frac{\alpha_1\beta_1}{4\alpha_2}$, while $K_i > \frac{1}{\epsilon}\frac{\beta_1\beta_2}{8\alpha_2}$ implies $2\epsilon K_i\frac{\alpha_1}{\beta_2} > \frac{\alpha_1\beta_1}{4\alpha_2}$. This proves the statement because $\mu_{PDGD} \leq \frac{\alpha_1\beta_1}{4\alpha_2}$. □

In Sec. V-A, we validate this result about the enhanced convergence speed through numerical simulations.

B. Illustrative example

To complete the analysis, we present an example that illustrates how the tuning of K_i and K_p may affect the convergence rate. We also compare the results to the case $K_p = 0$, representing PDGD. Let us consider the univariate optimization problem

$$\min_{x\in\mathbb{R}} \frac{1}{2}wx^2 \quad \text{s.t.} \quad x = 0 \quad (38)$$

where $w > 0$. The PI-CMO dynamics for this problem is

$$\begin{aligned} \dot{x} &= -wx - \lambda \\ \dot{\lambda} &= (K_i - K_pw)x - K_p\lambda. \end{aligned} \quad (39)$$

This is a second-order CT linear time-invariant system

$$\begin{pmatrix} \dot{x} \\ \dot{\lambda} \end{pmatrix} = \mathrm{A}\begin{pmatrix} x \\ \lambda \end{pmatrix} \quad (40)$$

with

$$\mathrm{A} := \begin{pmatrix} -w & -1 \\ K_i - K_pw & -K_p \end{pmatrix}. \quad (41)$$

The eigenvalues of A are

$$\frac{-(K_p + w) \pm \sqrt{(K_p+w)^2 - 4K_i}}{2}. \quad (42)$$

For any $K_i > 0$, the eigenvalues are either real and negative or complex with negative real parts. In particular, if $K_i \geq (K_p + w)^2/4$, K_i contributes only to the imaginary part, therefore it does not impact on the convergence rate.

According to [9], although $\eta > 0$ can be arbitrarily large for PDGD, increasing η beyond a certain threshold does not lead to a faster decaying rate. In our example, if we choose $K_p = 0$, we obtain PDGD with $\eta = K_i$ and, from (42), we see that if $K_i \geq w^2/4$, then K_i has no impact on the real parts of the eigenvalues. This explains the observation in [9].

On the other hand, in PI-CMO, we can tune $K_p > 0$ to enhance the convergence rate, provided that the conditions of Theorem 2 are satisfied, which is a benefit with respect to PDGD.

C. PI-CMO in non-convex quadratic optimization with linear constraints

To conclude this section, we analyze some properties of PI-CMO for quadratic optimization with linear constraints. In particular, we prove that optimization problems exist with non-convex cost functions in which PI-CMO converges to a stationary point while PDGD is divergent.

We consider

$$\begin{aligned} &\min_{x\in\mathbb{R}^n} \frac{1}{2}x^\top Wx \\ &\text{s.t.} \\ &Cx + d = 0 \end{aligned} \quad (43)$$

where $W \in \mathbb{R}^{n,n}$ is symmetric and $C \in \mathbb{R}^{m,n}$ and $d \in \mathbb{R}^m$, with invertible $CC^\top$.

Since the cost function is quadratic and the constraints are linear, PI-CMO corresponds to the linear time-invariant system

$$\begin{pmatrix} \dot{x} \\ \dot{\lambda} \end{pmatrix} = \mathrm{A} \begin{pmatrix} x \\ \lambda \end{pmatrix} + \begin{pmatrix} 0 \\ K_i d \end{pmatrix} \tag{44}$$

with

$$\mathrm{A} := \begin{pmatrix} -W & -C^\top \\ K_i C - K_p C W & -K_p C C^\top \end{pmatrix} \tag{45}$$

Therefore, if A is Hurwitz, then the system is asymptotically stable, and by construction, the output $y(t) = Cx(t) + d$ is regulated to zero. Thus, for PI-CMO W does not need to be positive definite, i.e., the system does not need to be strongly convex. As an example, let us consider $W = \mathrm{diag}(1, -1)$ and $C = (0, 2)$. Since W is indefinite, the quadratic cost function is not convex, while the constrained problem has a unique minimum. The eigenvalues of the corresponding dynamic matrix are -1 and $\frac{1-4K_p \pm \sqrt{(1-4K_p)^2 - 16K_i}}{2}$. If $K_p = 0$, all the eigenvalues have a positive real part for all K_i. In other terms, PDGD is always unstable. Instead, for $K_p > \frac{1}{4}$, all the eigenvalues have negative real part for all $K_i > 0$.

In conclusion, there exist problems with non-convex functions where PI-CMO converges to the minimum as long as we provide a suitable tuning of K_p. In contrast, PDGD is divergent for any hyperparameter choice. This observation encourages future study of PI-CMO in non-convex optimization.

IV. Second method: FL-CMO

In this section, we resort to feedback linearization as detailed, e.g., in [29], to design the controller $\mathcal{K}$ introduced in Section II. Moreover, we study the conditions under which the controlled dynamics is stable and the algorithm converges to the desired solution.

We organize the section as follows. Sec. IV-A summarizes the key concepts and results necessary for the development of the proposed algorithm and its stability analysis. In particular, we overview the non-interacting control problem and its solution. In Sec. IV-B, we apply the non-interacting control framework to Lagrange multipliers, which leads to the definition of the FL-CMO algorithm. The last subsections are devoted to the convergence analysis.

A. Feedback linearization basics

In this subsection, we review some basic concepts of the feedback linearization theory to design the control of λ and to analyze the convergence of the resulting dynamical system.

Definition 1. (Lie derivative, [29, Sec. 1.2], [30, Sec. 13.2]). *Let $F : \mathbb{R}^n \mapsto \mathbb{R}^n$ be a vector field and $H_i : \mathbb{R}^n \mapsto \mathbb{R}$. The Lie derivative of H_i along F is*

$$L_F H_i(x) = \nabla H_i(x)^\top F(x) \in \mathbb{R}. \tag{46}$$

By defining $L_F^0 H_i(x) = H_i(x)$, for $k = 1, 2, \ldots$, we have

$$L_F^k H_i(x) = (\nabla L_F^{k-1} H_i(x))^\top F(x). \tag{47}$$

As illustrated, e.g., in [29], [30], feedback linearization can be applied to input-affine nonlinear dynamical systems of the form

$$\begin{aligned} \dot{x} &= F(x) + G(x)u \\ y &= H(x) \end{aligned} \tag{48}$$

where $x(t) \in \mathbb{R}^n$, $u(t) \in \mathbb{R}^m$, and $y(t) \in \mathbb{R}^m$, $F : \mathbb{R}^n \mapsto \mathbb{R}^n$, $G : \mathbb{R}^n \mapsto \mathbb{R}^{n,m}$, $H : \mathbb{R}^n \mapsto \mathbb{R}^m$. For a dynamical system of this kind, we recall the definition of the relative degree.

Let $G_j(x)$ be the j-th column of $G(x)$.

Definition 2. (Relative degree, [29, Sec. 5.1], [30, Sec. 13.2]). *System* (48) *has a vector relative degree $r = (r_1, \ldots, r_m)^\top$ at $\bar{x} \in \mathbb{R}^n$ if*

(a) for each $1 \le i, j \le m$, for each $k \in \mathbb{N}$ such that $k < r_i - 1$, and for all x in a neighbourhood of $\bar{x}$

$$L_{G_j} L_F^k H_i(x) = 0 \tag{49}$$

and

(b) the matrix $\left[L_{G_j} L_F^{r_i - 1} H_i(x)\right]_{1 \le i,j \le m} \in \mathbb{R}^{m,m}$ is non-singular at $x = \bar{x}$, i.e.,

$$\textit{rank}\left(\left[L_{G_j} L_F^{r_i - 1} H_i(\bar{x})\right]_{1 \le i,j \le m}\right) = m. \tag{50}$$

For a single-input, single-output system, if the relative degree is r, the output y and its derivatives up to $(r-1)$-th order do not depend on the input, while the r-th order derivative does. For linear systems, the concept is equivalent to the relative degree of a transfer function, i.e., the difference between the degree of the denominator and the degree of the numerator.

Now, we recall the concept of zero dynamics.

Definition 3. (Zero dynamics, [31], [29, Sec. 4.3]). *Consider system* (48). *Let $(\xi^\top, \eta^\top)^\top = \Phi(x)$ be a smooth change of coordinates such that $\xi = (y_1, \dot{y}_1, \ldots y_1^{(r_1-1)}, \ldots, y_m, \ldots, y_m^{(r_m-1)}) \in \mathbb{R}^r$ and $\eta \in \mathbb{R}^{n-r}$. Using this transformation, we obtain the normal form representation of* (48)*, i.e.,*

$$\begin{aligned} \dot{\xi} &= \phi(\xi, \eta) + \gamma(\xi, \eta)u \\ \dot{\eta} &= q(\gamma, \eta), \end{aligned} \tag{51}$$

where $\phi : \mathbb{R}^r \times \mathbb{R}^{n-r} \to \mathbb{R}^r$, $\gamma : \mathbb{R}^r \times \mathbb{R}^{n-r} \to \mathbb{R}^{r,m}$, and $q : \mathbb{R}^r \times \mathbb{R}^{n-r} \to \mathbb{R}^{n-r}$. The zero dynamics of (48) *is defined as the dynamics of the system $\dot{\eta} = q(0, \eta)$.*

The zero dynamics of a system $\mathcal{S}$ describes the internal behavior of $\mathcal{S}$ when the output is constrained to be identically zero through a suitable choice of input and initial conditions. We refer the reader to [32] for further insights.

Finally, we state the non-interacting control problem and present its solution.

Definition 4. (Non-interacting control problem, [29, Sec. 5.3]) *Let us consider system* (48). *We say that a controller of the form*

$$u = \alpha(x) + \beta(x)v \tag{52}$$

with $\alpha(x) \in \mathbb{R}^m$, $\beta(x) \in \mathbb{R}^{m,m}$ and $v \in \mathbb{R}^m$ solves the non-interacting control problem if in the closed-loop system

$$\begin{cases} \dot{x} = F(x) + G(x)\alpha(x) + G(x)\beta(x)v \\ y = H(x) \end{cases} \tag{53}$$

each input v_i affects only output H_i, for each $i = 1, \ldots, m$.

The solution to the non-interacting control problem is established in [29]; for completeness, we report this result in the following theorem.

Theorem 3. (Non-interacting control solution, [29, Sec. 5.3])
(a) The non-interacting control problem admits a solution if and only if system (48) *has some vector relative degree r. In that case, the solution is given by*

$$\alpha(x) = -A^{-1}(x)b(x), \qquad \beta(x) = A^{-1}(x) \tag{54}$$

where

$$\begin{aligned} A(x) &= \left[L_{G_j} L_F^{r_i - 1} H_i(x)\right]_{1 \le i,j \le m} \in \mathbb{R}^{m,m} \\ b(x) &= \left[L_F^{r_i} H_i(x)\right]_{i=1,\ldots,m} \in \mathbb{R}^m. \end{aligned} \tag{55}$$

Moreover, for each $i = 1, 2, \ldots, m$, the input-output behavior between $v_i(t)$ and $y_i(t)$ is linear and described in the $s-$domain by the transfer function

$$\frac{1}{s^{r_i}}. \tag{56}$$

(b) If $v(t)$ stabilizes the system described by (56) *and the zero dynamics of* (48) *is asymptotically stable, then the feedback control system is asymptotically stable.*

Part (a) of Theorem 3 provides an input-output global linearization of system (48), which is decoupled in each component $i = 1, \ldots, m$. From part (b) of Theorem 3, the stability of the controlled system also depends on the zero dynamics, which accounts for the non-observable part of the system.

B. Non-interacting control of Lagrange multipliers

In this section, we apply the theory reported in Sec. IV-A to design a feedback linearized control for the plant $\mathcal{P}$ as defined in (5).

The key point is that $\mathcal{P}$ has the input-affine structure of (48), with

$$F(x) = -\nabla f(x), \; G(x) = -J_h(x)^\top, \; H(x) = h(x). \tag{57}$$

Therefore, we can apply Theorem 3 to obtain a decoupled controller if $\mathcal{P}$ has some vector relative degree. Let us analyze this point using the following assumption.

Assumption 3. *Every $x \in \mathbb{R}^n$ is a regular point for problem* (1)*, i.e.,*

$$\textit{rank}\,(J_h(x)) = m, \qquad \forall x \in \mathbb{R}^n. \tag{58}$$

In other terms, the Jacobian of the constraints is full row rank; this implies $m \le n$, i.e., there are at least as many optimization variables as constraints. This assumption is quite standard in constrained optimization; see, e.g., [33], [16], [18]. In practice, it is met in many applications, including a broad class of system identification problems, such as in the example reported in Sec. V-C, optimal control problems [34] and distributed optimization over networks [35].

In the case of affine constraints $h(x) = Cx + d = 0$, Assumption 3 guarantees that at least one feasible solution exists. Conversely, in the case of non-convex constraints, Assumption 3 implies that the feasible set $\Omega = \{x \in \mathbb{R}^n : h(x) = 0\}$ is an $(n-m)$-dimensional smooth manifold, i.e., locally similar to a $\mathbb{R}^{n-m}$ at all points. Consequently, Ω is not empty, and we can define a dynamical system whose trajectories evolve onto it, i.e., the zero dynamics. We finally remark that Assumption 3 implies the regularity condition of Theorem 1.

Lemma 2. *Under Assumption 3, the system $\mathcal{P}$ in* (5) *has a vector relative degree $r = (1, 1, \ldots, 1)^\top \in \mathbb{R}^m$.*

Proof. Let us consider (48) with (57). It is sufficient to check (50) with $r_i = 1$ for each $i = 1, \ldots, m$ to prove the thesis. From (57), $H_i(x) = h_i(x)$ and $G_j(x) = -\nabla h_j(x)$. Let us set $r_i = 1$ in (50). Then,

$$L_{G_j} L_F^0 H_i(x) = L_{G_j} H_i(x) = (\nabla h_i(x))^\top (-\nabla h_j(x)). \tag{59}$$

Thus,

$$\begin{aligned} &\text{rank}\left(\left[L_{G_j} L_F^0 H_i(x)\right]_{1 \le i,j \le m}\right) = \\ &= \text{rank}\left(\left[(\nabla h_i(x))^\top (-\nabla h_j(x))\right]_{ij}\right) \\ &= \text{rank}\left(-J_h(x) J_h(x)^\top\right) = m. \end{aligned} \tag{60}$$

Hence, (50) is satisfied for any x. □

According to Lemma 2, part (a) of Theorem 3 holds for system (53) with (57). Therefore, we resort to the non-interacting control solution given in Theorem 3. Specifically, we define the static feedback control law

$$\lambda(t) = A(x)^{-1}(-b(x) + v(t)) \tag{61}$$

where $A(x)$ and $b(x)$ are given by (55) with $r_i = 1$, i.e.,

$$\begin{aligned} A(x) &= \left[L_{G_j} H_i(x)\right]_{1 \le i,j \le m} = -J_h(x) J_h(x)^\top \in \mathbb{R}^{m,m} \\ b(x) &= \left[L_F H_i(x)\right]_{i=1,\ldots,m} = -J_h(x) \nabla f(x) \in \mathbb{R}^m. \end{aligned} \tag{62}$$

By applying the control law (61)-(62), the relationship between the new input $v(t) \in \mathbb{R}^m$ and the output $y(t) = h(x(t)) \in \mathbb{R}^m$ is

$$y_i(t) = \int_0^t v_i(\mu)\, \mathrm{d}\mu, \quad i = 1, \ldots, m, \tag{63}$$

which is linear and decoupled in each component i.

Next, we have to design $v(t)$ to regulate $y(t)$ to zero. A possible solution is to design the controller $\mathcal{G}$ in Fig. 3

$$v(t) = \mathcal{G}(y(t)) \tag{64}$$

such that the closed-loop dynamics is asymptotically stable and $y(t) \to 0$.

In the following, we refer to the closed-loop system dynamics defined by equations (5), (61) and (64), i.e.,

$$\dot{x}(t) = -\nabla f(x) + J_h^\top (J_h J_h^T)^{-1} (J_h \nabla f(x) + \mathcal{G}(y)), \tag{65}$$

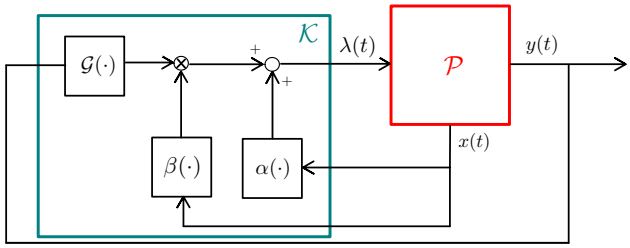


Fig. 3. Second method: feedback linearization. We feedback the output $y(t)$ to the controller $\mathcal{K}$ to compute $v(t)$ according to (64), and the state $x(t)$ to compute $u(t)$ according to (52).

as to FL-CMO dynamics.

Given the single integral structure of (63), the simplest way to design $\mathcal{G}$ is to consider m static linear feedback controllers

$$v_i(t) = -K_i y_i(t) \tag{66}$$

with $K_i > 0$ for $i = 1, \dots, m$. In fact, this leads to

$$y_i(t) = \alpha_i e^{-K_i t}, \quad i = 1, \dots, m \tag{67}$$

where α_i is a constant that depends on the initial conditions. Several more possibilities are available for the design of $\mathcal{G}$, each leading to a different CT optimization algorithm in the FL-CMO family characterized by different properties. In Sec. V-E, we present a numerical example comparing different choices for $\mathcal{G}$ and draw additional considerations on this aspect.

Remark 4. *FL-CMO requires the inversion of $A(x)$, which is a computational bottleneck for large m. To address this point, we notice that $-A(x) = J_h(x)J_h(x)^\top$ is positive definite by construction. Consequently, we can efficiently address the inversion through Cholesky factorization. Alternatively, we can compute the QR factorization $J_h(x)^\top = QR$. This choice yields the Cholesky factorization of $-A$ while avoiding the computation of the product $J_h(x)J_h(x)^\top$; in fact, $-A = R^\top Q^\top QR = R^\top R$. The sparsity level of $J_h(x)$ determines which of the two methods is more effective.*

Remark 5. *The FL-CMO dynamics* (65) *limited to the case of static $\mathcal{G}$ defined in* (66) *substantially corresponds to the systems studied in [15], [36]. The recent work [18] retrieves those systems and reframes them in the context of CBF-based feedback control, with an extension to the case of inequality constraints.*

Interestingly, the derivation of the algorithm in [18] is entirely different from the one presented in this work. The intuition behind [18] is to add a drift to the standard gradient, aiming to keep the feasibility of the state via a control action. In particular, CBF theory is used to synthesize the controller. In the case of equality constraints, the admissible control set is a singleton that corresponds to (66)*; see [18, Remark 4.2] for details.*

The proposed FL-CMO extends the CBF approach to a family of controllers $\mathcal{G}$ beyond (66)*. Although a comprehensive study of the properties of different $\mathcal{G}$'s is beyond the purpose of this work, in Sec. V-E, we propose a numerical example that compares static and dynamic $\mathcal{G}$'s in a robust optimization problem to highlight the benefits of dynamic controllers.*

From the development of FL-CMO, we can observe that its performance depends on two factors. On the one hand, the rate of convergence to the feasible set Ω is completely determined by the user when designing the external controller $\mathcal{G}$. In other terms, the convergence to Ω is arbitrarily fast. On the other hand, the zero dynamics determines the convergence speed to the optimal solution. Since the convergence to Ω is arbitrarily fast, the overall time required for convergence is determined in practice by the uncontrollable zero dynamics. Nevertheless, numerical experiments demonstrate that if the convergence rate to Ω is taken too fast, the differential equations defining the closed-loop system become stiff, rendering the numerical integration more challenging.

In Fig. 3, we summarize the feedback control scheme obtained via feedback linearization.

C. Local convergence of FL-CMO

In this section, we analyze the convergence of FL-CMO for possibly non-convex problems.

First, we notice that the FL-CMO dynamics converges to the feasible set Ω by construction. To prove the pointwise convergence, let us introduce the second-order sufficient conditions.

Definition 5. (Second-order sufficient conditions). *For problem* (1)*, let $H_{xx}\mathcal{L}(x, \lambda)$ be the Hessian matrix of $\mathcal{L}(x, \lambda)$ with respect to x. We say that the second-order sufficient conditions hold at $(x^\star, \lambda^\star)$ if*

$$v^\top H_{xx}\mathcal{L}(x^\star, \lambda^\star)v > 0 \tag{68}$$

for any $v \in \mathbb{R}^n$ such that $J_h(x^\star)v = 0$.

If the second-order sufficient conditions hold, $x^\star$ is a strict local minimum of (1). We prove the following result of local convergence.

Theorem 4 (Local convergence of FL-CMO to local minima)**.** *Let the second-order sufficient conditions hold at $(x^\star, \lambda^\star)$. Then the local minimum $x^\star$ is a locally asymptotically stable equilibrium point of the FL-CMO dynamics.*

Proof. Let us consider part (b) of Theorem 3. First, we notice that $v(t)$ stabilizes (56) by construction and regulates the output to zero, see (64).

Therefore, it is sufficient to prove that the zero dynamics of (48) is asymptotically stable to obtain the asymptotic stability of the closed-loop dynamics (53)-(57)-(64). We prove this fact in a neighbourhood of an equilibrium point $(x^\star, \lambda^\star)$ of $\mathcal{P}$, see II-A.

To analyze the zero dynamics, we begin by defining the mapping $\Phi : \mathbb{R}^n \mapsto \mathbb{R}^n$ as

$$\Phi(x) = \begin{pmatrix} h(x) \\ J_h^\perp(x^\star)(x - x^\star) \end{pmatrix}, \tag{69}$$

where we define $J_h^\perp(x) \in \mathbb{R}^{n-m,n}$ as follows: its rows are an orthonormal basis for the null space of the rows of J_h. As a

consequence, $J_h^\perp(x)J_h^\top(x)=0$ for all $x\in\mathbb{R}^n$. The Jacobian matrix of $\Phi(x)$ is

$$J_\Phi(x)=\begin{pmatrix}J_h(x)\\ J_h^\perp(x^\star)\end{pmatrix}. \tag{70}$$

Since $J_h(x)$ has rank m by Assumption 3 and $J_h^\perp(x)$ has orthogonal rows by definition, then rank $(J_\Phi(x))=n$. Therefore, Φ is invertible and provides a suitable change of coordinates in the state space of $\mathcal{P}$. In particular, we can express the transformed state $z=\Phi(x)$ as

$$z=\begin{pmatrix}\xi\\ \eta\end{pmatrix} \tag{71}$$

where $\xi=y\in\mathbb{R}^m$ corresponds to the output and $\eta\in\mathbb{R}^{n-m}$ represents the state of the zero dynamics of the system. We refer the reader to [29, Sec 5.1].

Now, by exploiting the change of variables via Φ, we write the system normal form and analyze the zero dynamics.

From (5), we have

$$\dot{x}=g(x,\lambda):=-\nabla f(x)-J_h(x)^\top\lambda.$$

Then, the normal form is

$$\dot{z}=\frac{\mathrm{d}}{\mathrm{d}t}\Phi(x)=J_\Phi(x)g(x,\lambda)=J_\Phi(\Phi^{-1}(z))g(\Phi^{-1}(z),\lambda). \tag{72}$$

We notice that $\Phi(x^\star)=0$. Then, let us represent (72) through its Taylor expansion around $(\Phi(x^\star),\lambda^\star)=(0,\lambda^\star)$, i.e.,

$$\dot{z}=\mathcal{Q}z+\mathcal{R}(\lambda-\lambda^\star)+o(z)+o(\lambda-\lambda^\star). \tag{73}$$

Specifically,

$$\begin{aligned}\mathcal{Q}&=\frac{\partial}{\partial z}\left[J_\Phi(\Phi^{-1}(z))g(\Phi^{-1}(z),\lambda^\star)\right]_{|z=0}\\ &=\frac{\partial}{\partial x}\left[J_\Phi(x)g(x,\lambda^\star)\right]_{|x=x^\star}J_{\Phi^{-1}}(0)\end{aligned} \tag{74}$$

Since $g(x^\star,\lambda^\star)=0$,

$$\begin{aligned}&\frac{\partial}{\partial x}\left[J_\Phi(x)g(x,\lambda^\star)\right]_{|x=x^\star}=\\ &=\frac{\partial}{\partial x}\left[J_\Phi(x)\right]_{|x=x^\star}g(x^\star,\lambda^\star)+J_\Phi(x^\star)\frac{\partial}{\partial x}\left[g(x,\lambda^\star)\right]_{|x=x^\star}\\ &=J_\Phi(x^\star)\frac{\partial}{\partial x}\left[g(x,\lambda^\star)\right]_{|x=x^\star}.\end{aligned} \tag{75}$$

Thus,

$$\mathcal{Q}=-J_\Phi(x^\star)H_{xx}\mathcal{L}(x^\star,\lambda^\star)J_{\Phi^{-1}}(0), \tag{76}$$

where

$$H_{xx}\mathcal{L}(x^\star,\lambda^\star)=-\frac{\partial}{\partial x}\left[g(x,\lambda^\star)\right]_{|x=x^\star}. \tag{77}$$

Moreover, by the inverse function theorem

$$J_{\Phi^{-1}}(0)=J_\Phi^{-1}(x^\star)=\begin{pmatrix}J_h^\dagger(x^\star), & J_h^{\perp\top}(x^\star)\end{pmatrix} \tag{78}$$

where $J_h^\dagger(x^\star)=J_h^\top(x^\star)[J_h(x^\star)J_h^\top(x^\star)]^{-1}$. In conclusion,

$$\mathcal{Q}=-J_\Phi(x^\star)H_{xx}\mathcal{L}(x^\star,\lambda^\star)\begin{pmatrix}J_h^\dagger(x^\star), & J_h^{\perp\top}(x^\star)\end{pmatrix}. \tag{79}$$

On the other hand, given $\frac{\partial}{\partial\lambda}g(x^\star,\lambda)_{|\lambda=\lambda^\star}=-J_h^\top(x^\star)$, we have

$$\begin{aligned}\mathcal{R}&=\frac{\partial}{\partial\lambda}\left(J_\Phi(x^\star)g(x^\star,\lambda)\right)|_{\lambda=\lambda^\star}\\ &=-J_\Phi(x^\star)J_h^\top(x^\star)=-\begin{pmatrix}J_h(x^\star)J_h^\top(x^\star)\\ J_h^\perp(x^\star)J_h^\top(x^\star)\end{pmatrix}=\\ &=-\begin{pmatrix}J_h(x^\star)J_h^\top(x^\star)\\ 0\end{pmatrix}.\end{aligned} \tag{80}$$

Next, we obtain the zero dynamics by setting $\xi=0$ and considering the last $n-m$ equations of (72):

$$\begin{aligned}\dot{\eta}&=-J_h^\perp(x^\star)H_{xx}\mathcal{L}(x^\star,\lambda^\star)\begin{pmatrix}J_h^\dagger(x^\star), & J_h^{\perp\top}(x^\star)\end{pmatrix}\begin{pmatrix}0\\ \eta\end{pmatrix}+o(z)\\ &=-J_h^\perp(x^\star)H_{xx}\mathcal{L}(x^\star,\lambda^\star)J_h^{\perp\top}(x^\star)\eta+o(z).\end{aligned} \tag{81}$$

We notice that the zero dynamics does not depend on λ.

Finally, by neglecting the high-order terms $o(z)$, the linearization of the zero dynamics is

$$\dot{\eta}=-J_h^\perp(x^\star)H_{xx}\mathcal{L}(x^\star,\lambda^\star)J_h^{\perp\top}(x^\star)\eta. \tag{82}$$

The original nonlinear zero-dynamics is locally asymptotically stable at $\eta=0$ if (82) is asymptotically stable, i.e., if the symmetric matrix $J_h^\perp(x^\star)H_{xx}\mathcal{L}(x^\star,\lambda^\star)J_h^{\perp\top}(x^\star)$ is positive definite, which holds if the second-order sufficient conditions reported in Definition 5 are satisfied. In fact, let $v=J_h^{\perp,\top}(x^\star)w$ for any non-null $w\in\mathbb{R}^{n-m}$; since $J_h(x^\star)J_h^{\perp,\top}(x^\star)=0$ by definition, $J_h(x^\star)v=0$. Then, $v^\top H_{xx}\mathcal{L}(x^\star,\lambda^\star)v=w^\top J_h^\perp(x^\star)H_{xx}\mathcal{L}(x^\star,\lambda^\star)J_h^{\perp,\top}(x^\star)w>0$. □

Remark 6. *In our setting, the linearized zero dynamics in* (82) *corresponds to the zero dynamics of the linearization of $\mathcal{P}$, as we prove in the following. We notice that the commutativity of the operations of linear approximation and computation of the zero dynamics always holds for single-input, single-output systems; see, e.g., [29, Remark 4.3.2]. However, the commutativity is not guaranteed in general for multiple-input, multiple-output systems. For this reason, it is worth remarking that the class of multiple-input, multiple-output systems considered in this work, characterized by $r=(1,1,\dots,1)\in\mathbb{R}^m$ and $G=-J_H^\top$ according to* (57)*, enjoys the commutativity.*

By Taylor expansion, the linearization of $\mathcal{P}$ in a neighbourhood of $(x^\star,\lambda^\star)$ is

$$\begin{cases}\dot{x}=-H_{xx}\mathcal{L}(x^\star,\lambda^\star)(x-x^\star)-J_h^\top(x^\star)(\lambda-\lambda^\star)\\ y=J_h(x^\star)(x-x^\star)\end{cases} \tag{83}$$

Let us consider the mapping $\Phi:\mathbb{R}^n\mapsto\mathbb{R}^n$ as

$$\begin{pmatrix}\xi\\ \eta\end{pmatrix}=\Phi(x)=\begin{pmatrix}J_h(x^\star)\\ J_h^\perp(x^\star)\end{pmatrix}(x-x^\star) \tag{84}$$

We notice that $\xi=y$. In normal form,

$$\begin{pmatrix}\dot{\xi}\\ \dot{\eta}\end{pmatrix}=\begin{pmatrix}J_h(x^\star)\\ J_h^\perp(x^\star)\end{pmatrix}\dot{x}. \tag{85}$$

Let us focus on η, which represents the state variable of the zero dynamics. We have

$$\begin{aligned}\dot{\eta}&=J_h^\perp(x^\star)\dot{x}\\ &=-J_h^\perp(x^\star)H_{xx}\mathcal{L}(x^\star,\lambda^\star)(x-x^\star).\end{aligned} \tag{86}$$

As expected, $\dot{\eta}$ does not depend on λ since $J_h^{\perp}(x^\star)J_h(x^\star)=0$. By inversion,

$$x = \begin{pmatrix} J_h^{\dagger}(x^\star), & J_h^{\perp,\top}(x^\star) \end{pmatrix} \begin{pmatrix} \xi \\ \eta \end{pmatrix} + x^\star \tag{87}$$

To study the zero dynamics, we set $\xi = y = 0$. Thus,

$$x - x^\star = \begin{pmatrix} J_h^{\dagger}(x^\star), & J_h^{\perp,\top}(x^\star) \end{pmatrix} \begin{pmatrix} 0 \\ \eta \end{pmatrix} = J_h^{\perp,\top}(x^\star)\eta \tag{88}$$

and

$$\dot{\eta} = -J_h^{\perp}(x^\star)H_{xx}\mathcal{L}(x^\star,\lambda^\star)J_h^{\perp,\top}(x^\star)\eta \tag{89}$$

which is equal to (82)*.*

D. Global exponential convergence of FL-CMO for strongly convex problems

In this section, we study the global convergence of FL-CMO dynamics, defined by (65), for strongly convex problems. More precisely, we consider the same setting in which we analyse the convergence of PI-CMO and described by assumptions 1 and 2, namely $f(x)$ in (1) is strongly convex and twice differentiable; $h(x) = Cx+d$ where $C \in \mathbb{R}^{m,n}$ is full rank and $\alpha_1 I_m \preceq CC^\top \preceq \alpha_2 I_m$. Under these assumptions, $J_h(x) = C$, and the control law for (5) defined by (61), (62) is:

$$\lambda(t) = -(CC^\top)^{-1}(C\nabla f(x(t)) + v(t)) \tag{90}$$

where $v(t) = \mathcal{G}(y(t))$, see (64), and $y(t) = Cx(t) + d$.

Moreover, according to [9, Lemma 1], there exists a symmetric $B(x) \in \mathbb{R}^{n,n}$ satisfying $\beta_1 I \preceq B(x) \preceq \beta_2 I$ for some $0 < \beta_1 < \beta_2$ such that (15) holds. As in Sec. III, for brevity we use the notation $B = B(x)$.

Theorem 5 (Global exponential convergence of FL-CMO)**.** *Let assumptions 1 and 2 hold. Let us choose $\mathcal{G}$ in* (64) *such that $\bar{y} = 0$ is a globally exponentially stable equilibrium for $\dot{y} = \mathcal{G}(y)$, with convergence rate $\mu_g \geq \beta_1$, i.e., there exists a constant $c_g \in \mathbb{R}_+$ such that*

$$\|y(t)\|_2 \leq c_g e^{-\mu_g t}. \tag{91}$$

Then, the global optimum $x^\star$ of problem (1) *is a globally exponentially stable equilibrium for FL-CMO with rate β_1, i.e., there exists a constant $c \in \mathbb{R}_+$ such that*

$$\|x - x^\star\|_2 \leq c e^{-\beta_1 t}. \tag{92}$$

Proof. Under assumptions 1-2, the global change of coordinates defined by (71) and (69) is

$$z = \Phi(x) = \begin{pmatrix} \zeta \\ \eta \end{pmatrix} = \begin{pmatrix} Cx + d \\ C^{\perp}(x - x^\star) \end{pmatrix} \tag{93}$$

where the rows of $C^{\perp} \in \mathbb{R}^{m-n,n}$ are an orthonormal basis for the null space of the rows of C.

Since (93) is an affine transformation,

$$x - x^\star = \begin{pmatrix} C \\ C^{\perp} \end{pmatrix}^{-1} (z - z^\star) \tag{94}$$

where $z^\star = \Phi(x^\star)$.

Given the spectral norm $\sigma := \left\| \begin{pmatrix} C \\ C^{\perp} \end{pmatrix}^{-1} \right\|_2$ and by applying the triangle inequality we obtain

$$\|x - x^\star\|_2 \leq \sigma \|z - z^\star\|_2 \leq \sigma \left(\left\| (\zeta - \zeta^\star) \right\|_2 + \left\| (\eta - \eta^\star) \right\|_2 \right). \tag{95}$$

Then, we study the convergence of $\|x - x^\star\|_2$ based on the convergence of $\left\| (\zeta - \zeta^\star) \right\|_2$ and $\left\| (\eta - \eta^\star) \right\|_2$.

Since $\zeta = y$ and $\zeta^\star = 0$, the global exponential convergence of ζ is given by (91).

As to η, by using $\nabla_x \mathcal{L}(x^\star, \lambda(x^\star)) = 0$, the zero dynamics is

$$\begin{aligned} \dot{\eta} &= C^{\perp}\dot{x} = C^{\perp}\left[\nabla_x \mathcal{L}(x,\lambda(x)) - \nabla_x \mathcal{L}(x^\star,\lambda(x^\star))\right] = \\ &= -C^{\perp}\left[I + C^\top (CC^\top)^{-1}C\right)\left(\nabla f(x) - \nabla f(x^\star)\right)\right] \\ &= -C^{\perp}\left(\nabla f(x) - \nabla f(x^\star)\right) \end{aligned} \tag{96}$$

where the last equality follows from $\mathcal{G}(0) = 0$ and $C^{\perp}C^\top = 0$.

According to [9, Lemma 1] on strongly convex functions, see (15), we can write $\dot{\eta} = C^{\perp}B[x - x^\star]$. Moreover,

$$x - x^\star = C^\top (CC^\top)^{-1}(\zeta - \zeta^\star) + C^{\perp\top}(\eta - \eta^\star) = C^{\perp\top}\eta$$

where we use $\zeta = \zeta^\star = 0$ by definition of zero dynamics and $\eta^\star = 0$. In conclusion,

$$\dot{\eta} = -C^{\perp}BC^{\perp\top}\eta. \tag{97}$$

From [9, Lemma 1], $B - \beta_1 I \succeq 0$. Moreover, by applying [37, Observation 7.1.8], we have

$$C^{\perp}(B - \beta_1 I)C^{\perp\top} = C^{\perp}BC^{\perp\top} - \beta_1 I \succeq 0. \tag{98}$$

Thus, the matrix $C^{\perp}BC^{\perp\top}$ is Hurwitz and

$$\|\eta\| \leq c_\eta e^{-\beta_1 t}. \tag{99}$$

By merging (95), (91), and (99), we get

$$\|x - x^\star\|_2 \leq \sigma \left(c_g e^{-\mu_g t} + c_\eta e^{-\beta_1 t} \right) \leq \sigma(c_g + c_\eta) e^{-\beta_1 t} \tag{100}$$

which proves (92) with $c = \sigma(c_g + c_\eta)$. □

Remark 7. *Theorem 5 shows that FL-CMO enjoys a better convergence rate than PI-CMO in strongly convex problems. In fact, the rate β_1 of FL-CMO is always larger than $\frac{1}{2}\mu < \beta_1$ evaluated in Theorem 2 for PI-CMO.*

Remark 8. *The convergence rate of FL-CMO uniquely depends on β_1, i.e., on $f(x)$, while it is independent from the constraints $h(x)$. This is not the case for PI-CMO and PDGD which depend on constants α_1 and α_2, that are related to the constraints.*

V. Numerical results

In this section, we illustrate five numerical examples that validate the theoretical convergence results and prove the effectiveness of the proposed approaches PI-CMO and FL-CMO compared to state-of-the-art algorithms.

In Sec. V-A, we test PI-CMO and FL-CMO in a convex problem and analyze their convergence speed. Then, in Sec. V-B, we investigate the effectiveness of PI-CMO in a non-convex problem in which FL-CMO is not feasible.

In the succeeding two examples, we test FL-CMO in non-convex problems, namely a gray-box system identification problem in Sec. V-C and a large-scale real-world chemical problem in Sec. V-D.

Finally, in Sec.V-E, we illustrate the behavior of different instances of FL-CMO based on different choices of $\mathcal{G}$.

A. PI-CMO and FL-CMO vs PDGD in convex optimization

In the first example, we resort to the quadratic optimization problem with linear constraints proposed in [9, Section IV.A]. Specifically, we consider

$$\begin{aligned} \min_{x\in\mathbb{R}^n} \ & \frac{1}{2}x^\top W x \\ \text{s.t.} \ & Cx + d = 0 \end{aligned} \tag{101}$$

where $W = 10I + W_0 W_0^\top \in \mathbb{R}^{n,n}$ is positive definite; $W_0 \in \mathbb{R}^{n,n}$, $C \in \mathbb{R}^{m,n}$ and $d \in \mathbb{R}^m$ have independent and normally distributed components.

To solve this problem, we implement and compare PI-CMO, FL-CMO and PDGD [9]. Since the cost function is quadratic and the constraints are linear, the PI-CMO dynamics is linear time-invariant, as illustrated in Sec. III-C.

Corollary 1 shows that the convergence rate bound of PI-CMO compares favorably with the one of PDGD for strongly convex problems with linear constraints. Moreover, as noticed in Remark 7, the convergence rate bound of FL-CMO enhances that of PI-CMO. In this section, we support these theoretical results with a numerical analysis of the convergence speed.

We consider $n = 50$ variables and $m \in [2, 26]$ constraints. For PDGD, we set $\eta = \frac{\beta_1\beta_2}{\alpha_2}$ to obtain the best convergence rate bound; see (35). For PI-CMO, we set K_i and K_p to meet the conditions of Corollary 1, specifically $K_i = \frac{50}{\epsilon}\frac{\beta_1\beta_2}{8\alpha_2}$, $K_p = \epsilon\frac{2K_i}{\beta_2}$ where $\epsilon = \frac{4}{5}\left(1 - \frac{\alpha_1}{8\alpha_2}\right)$. For FL-CMO, we design a static $\mathcal{G}$ with convergence rate $\mu_g = 5\beta_1$.

For all the algorithms, we run the Euler discretized versions with the discretization step size that guarantees stability, see [9, Section III.C] for details. We perform 400 random runs. Fig. 4 shows the average number of iterations required to converge to the desired minimum against m. For all the considered values of m, PI-CMO requires fewer iterations on average than PDGD, and FL-CMO is even faster than PI-CMO, as expected from the theoretical results on the convergence rate presented in Sec. III-A and IV-D.

Remark 9. *FL-CMO enjoys a better convergence rate than PI-CMO. Nevertheless, FL-CMO presents two main computational drawbacks that make PI-CMO more effective in some contexts. The first drawback is that FL-CMO requires the inversion of the $m \times m$ matrix $J_h(x)J_h(x)^\top$ in (61) at each iterative step, which generates a computational bottleneck. If the constraints are affine, $J_h(x)J_h(x)^\top = CC^\top$ is constant, which basically solves the issue: the inversion is computed once. In the other cases, we can exploit the structure of $A(x)$ to perform a suitable factorization; see Remark 4. Another possible workaround is to consider the approximate computation of the inverse; see Sec. V-E. When the inversion is not feasible, PI-CMO is a good alternative to FL-CMO. The second drawback is that implementing FL-CMO requires the invertibility of $J_h(x)J_h(x)^\top$. Therefore, this method cannot be implemented if $m > n$, which is not required for implementing PI-CMO, as we show in the next numerical example.*

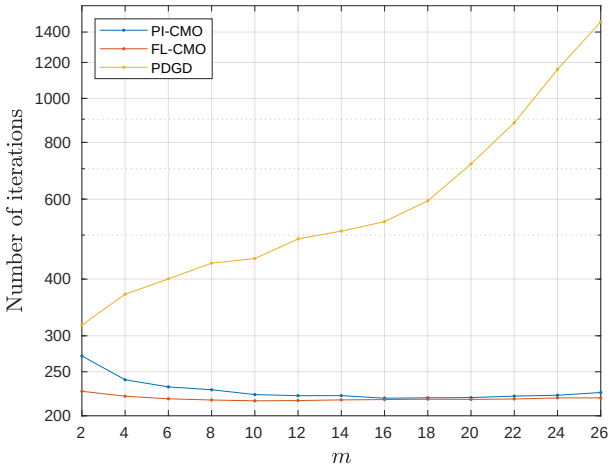


Fig. 4. Number of iterations to converge for PI-CMO, FL-CMO, and PDGD in a convex quadratic example, with m constraints. The results are averaged over 400 random runs.

B. Shidoku puzzle

Shidoku is a 4x4 version of the popular 9x9 Sudoku puzzle. Given an initial scheme, as reported in Fig. 5, the aim is to fill the empty cells with integers $x_{i,j} \in \{1, 2, 3, 4\}$ such that each row, each column, and each 2x2 corner block contains the integers $1, 2, 3, 4$. We can formulate the game in terms

	1		4
2			3

Fig. 5. Shidoku puzzle to be solved.

of the solution of polynomial equations on the values $x_{i,j}$ of each cell (i, j), $i, j = 1, \ldots, 4$. First of all, $x_{i,j} \in \{1, 2, 3, 4\}$ is guaranteed by $\prod_{h=1}^{4}(x_{i,j} - h) = 0$. Then, we obtain no repetition in groups of 4 cells by imposing the product equal to 24 and the sum equal to 10. In summary, given the corner blocks

$$\begin{aligned} B_1 &= \{(1,1),(1,2),(2,1),(2,2)\} \\ B_2 &= \{(1,3),(1,4),(2,3),(2,4)\} \\ B_3 &= \{(3,1),(3,2),(4,1),(4,2)\} \\ B_4 &= \{(3,3),(3,4),(4,3),(4,4)\}. \end{aligned}$$

0.8	1	0.0	4
0.6	0.3	0.3	1.7
2	1.3	0.6	3
0.2	0.6	0.6	0.8

1.5	1	3.1	4
3.0	4.1	1.3	1.5
2	1.0	4.0	3
3.1	4.3	1.4	1.3

1.7	1	3.1	4
4.1	3.3	1.0	1.7
2	1.8	3.1	3
2.4	3.9	2.5	1.1

2.0	1	2.9	4
4.8	2.5	1.7	1.2
2	3.9	0.9	3
1.0	2.4	5.3	1.9

3.1	1	2.0	4
3.6	2.1	3.2	1.0
2	3.4	1.0	3
1.0	3.3	3.7	2.1

2.9	1	2.0	4
4.1	1.9	2.9	1.0
2	4.1	1.0	3
1.0	2.9	4.1	2.0

Fig. 6. PI-CMO solves Shidoku puzzle. Evolution of the solution of the optimization variables at six equispaced sampling instants between initialization and convergence step 91027, from top left to bottom right.

we have

$$\begin{aligned}
&\text{Columns: for } j = 1, \dots, 4,\\
&\quad \sum_{i=0}^{4} x_{ij} = 10, \quad \prod_{i=0}^{4} x_{ij} = 24.\\
&\text{Rows: for } i = 1, \dots, 4,\\
&\quad \sum_{j=0}^{4} x_{ij} = 10, \quad \prod_{j=0}^{4} x_{ij} = 24.\\
&\text{Blocks: for } k = 1, \dots, 4,\\
&\quad \sum_{(i,j)\in B_k} x_{ij} = 10, \quad \prod_{(i,j)\in B_k} x_{ij} = 24.\\
&\text{Restriction to integers: for } i, j = 1, \dots, 4,\\
&\quad \prod_{h=1}^{4} (x_{ij} - h) = 0.\\
&\text{Initial conditions as in Fig. 5:}\\
&\quad x_{1,2} = 1 \quad x_{1,4} = 4 \quad x_{3,1} = 2 \quad x_{3,4} = 3.
\end{aligned} \tag{102}$$

Equations (102) represent non-convex polynomial constraints. We can solve the corresponding optimization problem through the proposed CMO framework if we associate them with any constant cost function. In particular, we use PI-CMO to test its convergence and effectiveness in non-convex problems. We cannot apply FL-CMO since we have $m = 40$ equations in $n = 12$ variables, and $m > n$ is not consistent with Assumption 3.

As to PI-CMO, we set $K_i = 1, K_p = 0.1$ and we generate random initial conditions according to $x_{i,j}(0) = |\xi_{i,j}|, \xi_{i,j} \sim \mathcal{N}(0,1)$, for each $i, j = 1, \dots, 4$ and $\lambda_k(0) \sim \mathcal{N}(0,1), k = 1, \dots, m$, where $\mathcal{N}(0, \sigma)$ denotes the normal distribution with zero mean and variance σ^2. We integrate the ordinary differential equations that describe the closed-loop dynamics thanks to the MATLAB ode45 solver in the time interval $[0, 100]$ seconds, corresponding to approximately 151700 iterations. We perform 20 runs with different random initial conditions. PI-CMO is always convergent in the given time interval.

3	1	2	4
4	2	3	1
2	4	1	3
1	3	4	2

Fig. 7. PI-CMO solves Shidoku puzzle: final correct solution.

We show an instance in Fig. 6, where we depict the evolution of the optimization variables for six equispaced sampling instants between the random initialization and the convergence to the correct solution at iteration 91027, shown in Fig. 7.

In conclusion, this test shows that PI-CMO is convergent even in a non-convex problem that does not satisfy assumptions 1 and 2. For further investigation, we compare PI-CMO to two state-of-the-art approaches for non-convex constrained optimization, interior-point method (IPM) and sequential quadratic programming, through the fmincon function in MATLAB. We perform 20 runs with random initial conditions for the two of them. We observe that IPM fails in all the runs due to numerical issues. More precisely, the linear system of KKT conditions to solve at each iteration is poorly conditioned, which affects the solution; see, e.g., [33, Chapter 19] for details. On the other hand, sequential quadratic programming converges to an infeasible point in all the runs.

C. Gray-box non-linear system identification

In this example, we consider the identification of the discrete-time nonlinear system described by the following regressor form

$$\begin{aligned}
y(k) =& \theta_1 e^{-(y(k-1))^2} + \theta_2 (u(k-1))^2 + \\
& \theta_3 u(k-2) y(k-1) + \theta_4 (u(k-2))^{\theta_5}
\end{aligned} \tag{103}$$

using $N = 400$ measurements of the input sequence $u(k)$ and noisy measurements of the output sequence $\tilde{y}(k) = y(k) + \zeta(k)$, where $\zeta(k)$ represents an unknown measurement noise.

Since the system is not affine in the parameters θ, we cannot employ standard solutions based on least-squares regression. We look for the parameter vector $\theta^\star \in \mathbb{R}^5$ that minimizes the ℓ_2 norm of $\zeta = [\zeta(1), \dots, \zeta(N)]^\top$, i.e., the energy of the noise sequence $\zeta(k)$. Introducing additional optimization variables $y \in \mathbb{R}^N$ representing the noise-free output samples, the considered identification problem is equivalently rewritten as the following non-convex constrained problem:

$$\begin{aligned}
[\theta^\star, y^\star] = & \underset{\theta \in \mathbb{R}^5, y \in \mathbb{R}^N}{\arg\min} \sum_{k=1}^{N} (y_k - \tilde{y}(k))^2 \\
& \text{subject to:} \\
& - y_k + \theta_1 e^{-y_{k-1}^2} + \theta_2 (u(k-1))^2 + \\
& \theta_3 u(k-2) y_{k-1} + \theta_4 (u(k-2))^{\theta_5} = 0 \\
& \quad k = 3, \dots, N.
\end{aligned} \tag{104}$$

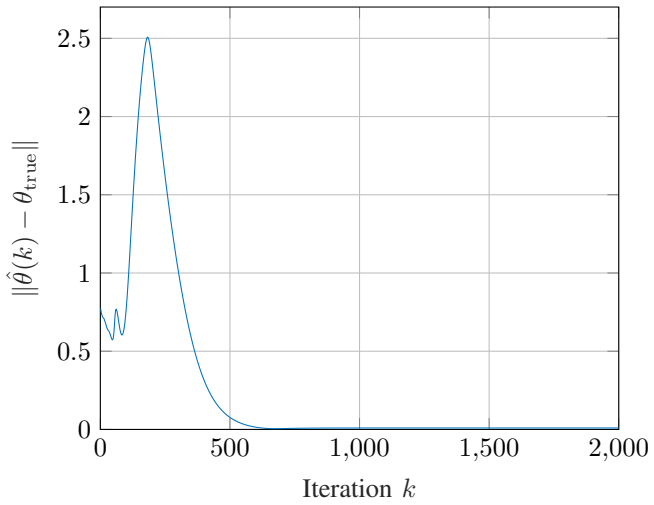


Fig. 8. Gray-box non-linear system identification: evolution of the estimation error. We denote by $\hat{\theta}(k)$ the estimates obtained with the feedback linearization-based algorithm at iteration k, and by θ_{true} the true parameter vector.

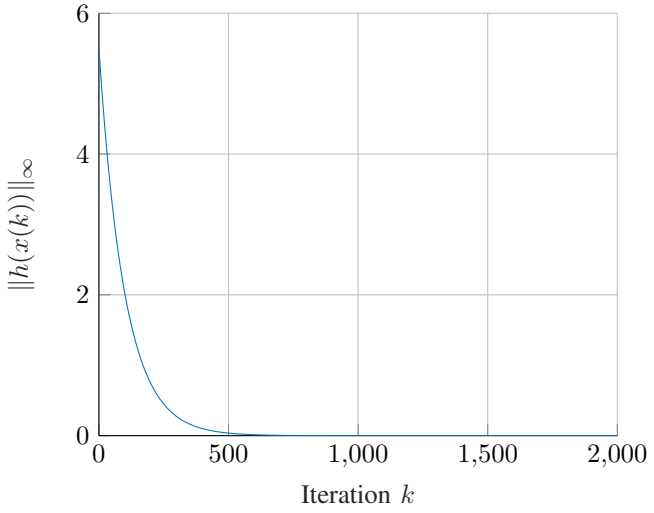


Fig. 9. Gray-box non-linear system identification: evolution of the ℓ_∞-norm of the constraints function h. In the figure, $x(k)$ refers to the estimate, at iteration k, of y and θ in (104).

We solve problem (104) using the feedback linearization method developed in Section IV. We define the controller $\mathcal{G}$ in (64) according to (66) , with $K_i = 1$ for each $i = 1, \ldots, m$.

We integrate the FL-CMO ordinary differential equations in the time interval $t = [0, 20]$ seconds using Euler discretization with step size 10^{-2} seconds. We set normally distributed initial conditions.

Fig. 8 shows the ℓ_2-norm of the estimation error as a function of the algorithm iteration. Fig. 9 shows the evolution of the ℓ_∞-norm of the constraints: as expected, it converges to zero.

For comparison, we solve problem (104) through IPM implemented with the fmincon MATLAB function and initialized with the same initial conditions used for the feedback linearization method. We use MATLAB R2021b on a processor i7-10700, 2.90GHz with 32 GB of DDR4 RAM. Table I

TABLE I
GRAY-BOX NON-LINEAR SYSTEM IDENTIFICATION: COMPARISON OF PARAMETER ESTIMATES FOR THE PROPOSED FEEDBACK LINEARIZATION APPROACH AND IPM, DENOTED BY $\hat{\theta}$ AND θ_{IPM}, RESPECTIVELY.

θ_{true}	$\hat{\theta}$	θ_{IPM}
0.5	0.496	0.489
-0.3	-0.300	-0.300
-0.7	-0.699	-0.695
-0.35	-0.345	-0.338
0.8	0.815	0.804

compares the estimated parameters. We observe that both feedback linearization and IPM provide accurate estimates, the respective errors being $\|\theta_{\text{true}} - \hat{\theta}\|_2 = 0.0163$ and $\|\theta_{\text{true}} - \theta_{\text{IPM}}\|_2 = 0.0175$.

Let us analyze the computational complexity. The time required by FL-CMO is 9.8 seconds, while IPM requires 64.4 seconds. Moreover, the proposed feedback linearization method is less memory expensive compared to IPM, as expected when we compare first-order and second-order methods because it requires storing only the Jacobian of the constraints and not the Hessian matrix. More precisely, given $m \leq n$, the FL-CMO requires $n + nm + \frac{1}{2}m^2 + m$ floating point numbers, while IPM requires $n + n^2 + nm + 3m$ floating point numbers. In both cases, we assess this number by assuming to store only the triangular part of the symmetric matrices. Concerning IPM, the leading term n^2 is due to the sum of $m + 1$ Hessian matrices, one for $f(x)$ and m for $h_i(x)$, of dimension $n \times n$. Thus, we need $\frac{1}{2}n^2$ variables to compute the current Hessian, and a further $\frac{1}{2}n^2$ ones are used to store the partial sum.

In conclusion, there is a gain of $2m + n^2 - \frac{1}{2}m^2$ floating-point numbers using FL-CMO instead of IPM.

Moreover, each iteration in feedback linearization requires $O(m^3)$ floating-point operations (FLOPs) to invert the matrix $J_h J_h^\top \in^{m \times m}$. In contrast, each iteration of IPM requires $O((n + m)^3)$ FLOPs to solve a linear system of dimension $n + m$.

We notice that similar considerations apply to sequential quadratic programming, which needs approximately the same memory and computations as IPM; see, e.g., [33, Chapter 18].

D. Industrial chemical process problem

We consider a problem that arises in the context of industrial chemical processes: the propane, isobutane, and n-butane nonsharp separation presented in [38]. It is about a three-component feed mixture required to separate products into two three-component products. This problem is included in the benchmark suite of real-world, non-convex problems proposed in [39] to test optimization algorithms. The mathematical formulation of the problem, reported in [39, Sec. 2.1.5], is as follows. Given $x = (x_1, \ldots, x_{48})$, minimize $f(x)$ defined as

$$f(x) = c_{11} + (c_{21} + c_{31}x_{24} + c_{41}x_{28} + c_{51}x_{33} + c_{61}x_{34})\, x_5 + c_{12} + (c_{22} + c_{32}x_{26} + c_{42}x_{31} + c_{52}x_{38} + c_{62}x_{39})\, x_{13}$$

where $c_{11} = 0.23947$, $c_{12} = 0.75835$, $c_{21} = -0.0139904$, $c_{22} = -0.0661588$, $c_{31} = 0.0093514$, $c_{32} = 0.0338147$, $c_{41} = 0.0077308$, $c_{42} = 0.0373349$, $c_{51} = -0.0005719$,

$c_{52} = 0.0016371$, $c_{61} = 0.0042656$, $c_{62} = 0.0288996$, subject to

$$
\begin{aligned}
&x_4 + x_3 + x_2 + x_1 = 300, && x_6 - x_8 - x_7 = 0,\\
&x_9 - x_{12} - x_{10} - x_{11} = 0, && x_{14} - x_{17} - x_{15} - x_{16} = 0,\\
&x_{18} - x_{20} - x_{19} = 0, && x_6 x_{21} - x_{24} x_{25} = 0,\\
&x_{14} x_{22} - x_{26} x_{27} = 0, && x_9 x_{23} - x_{28} x_{29} = 0,\\
&x_{18} x_{30} - x_{31} x_{32} = 0, && x_{25} - x_5 x_{33} = 0,\\
&x_{35} - x_5 x_{36} = 0, && x_{37} - x_{13} x_{38} = 0,\\
&x_{27} - x_{13} x_{39} = 0, && x_{32} - x_{13} x_{40} = 0,\\
&x_{25} - x_6 x_{21} - x_9 x_{41} = 0, && x_{29} - x_6 x_{42} - x_9 x_{23} = 0,\\
&x_{35} - x_6 x_{43} - x_9 x_{44} = 0, && x_{37} - x_{14} x_{45} - x_{18} x_{46} = 0,\\
&x_{27} - x_{14} x_{22} - x_{18} x_{47} = 0, && x_{32} - x_{14} x_{48} - x_{18} x_{30} = 0,\\
&0.33 x_1 + x_{15} x_{45} - x_{25} = 0, && 0.33 x_1 + x_{15} x_{22} - x_{29} = 0,\\
&0.33 x_1 + x_{15} x_{48} - x_{35} = 0, && 0.33 x_2 + x_{10} x_{41} - x_{37} = 0,\\
&0.33 x_2 + x_{10} x_{23} - x_{27} = 0, && 0.33 x_2 + x_{10} x_{44} - x_{32} = 0,\\
&x_{33} + x_{34} + x_{36} = 1, && x_{21} + x_{42} + x_{43} = 1,\\
&x_{41} + x_{23} + x_{44} = 1, && x_{38} + x_{39} + x_{40} = 1,\\
&x_{45} + x_{22} + x_{48} = 1, && x_{46} + x_{47} + x_{30} = 1,\\
&x_{43} = 0, && x_{46} = 0,
\end{aligned}
$$

$$
\begin{aligned}
&0.33 x_3 + x_7 x_{21} + x_{11} x_{41} + x_{16} x_{45} + x_{19} x_{46} = 30,\\
&0.33 x_3 + x_7 x_{42} + x_{11} x_{23} + x_{16} x_{22} + x_{19} x_{47} = 50,\\
&0.33 x_3 + x_7 x_{43} + x_{11} x_{44} + x_{16} x_{48} + x_{19} x_{30} = 30,
\end{aligned}
$$

with bounds

$$
\begin{aligned}
&0 \le x_1, \dots, x_{20} \le 150,\\
&0 \le x_{25}, x_{27}, x_{32}, x_{35}, x_{37}, x_{29} \le 30,\\
&0 \le x_{21}, x_{22}, x_{23}, x_{30}, x_{33}, x_{34}, x_{36}, x_{38}, x_{39} \le 1,\\
&0 \le x_{40}, x_{42}, x_{43}, x_{44}, x_{45}, x_{46}, x_{47}, x_{48} \le 1,\\
&0.85 \le x_{24}, x_{26}, x_{28}, x_{31} \le 1.
\end{aligned}
$$

The problem consists of $n = 48$ optimization variables, a bilinear cost function, and $m = 38$ linear and bilinear equality constraints. The overall problem is non-convex. Moreover, there are 47 constrained variables in bounded intervals, which give rise to 94 inequality constraints. To deal with the inequality constraints, we reformulate them by using squared-slack variables, as discussed, e.g., in [19, Sec. 3.3.2]. The basic idea is that we can rewrite any inequality $g(x) \le 0$ as $g(x) + z^2 = 0$, where $z \in \mathbb{R}$ is a slack variable.

For this problem, the benchmark objective value is $f(x) = 2.1158$, as reported in [39, Table 3].

We perform the optimization through FL-CMO, implemented in MATLAB R2021b, on a processor i7-10700, 2.90GHz with 32 GB of DDR4 RAM. As in the previous example, we design the $\mathcal{G}$ controller in Eq. (64) by pole placement. We place the closed-loop pole at -10. We integrate the closed-loop differential equations in the time interval $[0, 3]$ seconds using Euler discretization with step size $5 \cdot 10^{-5}$ seconds.

We randomly set the initial conditions with uniform distribution in $[0, 50]$. We run the algorithm 50 times with different realizations of the initial conditions. In all the runs, the algorithm converges to feasible solutions within a tolerance of 10^{-7} for each constraint. The achieved objective values are distributed according to the histogram in Fig. 10; the mean value is 2.332, and the standard deviation is 0.1852. Moreover, the best achieved objective value is 2.0333, which improves the benchmark 2.1158. In Fig. 10, we also report the distribution of the time required to converge; the mean value is 7.22 seconds, and the standard deviation is 0.33 seconds.

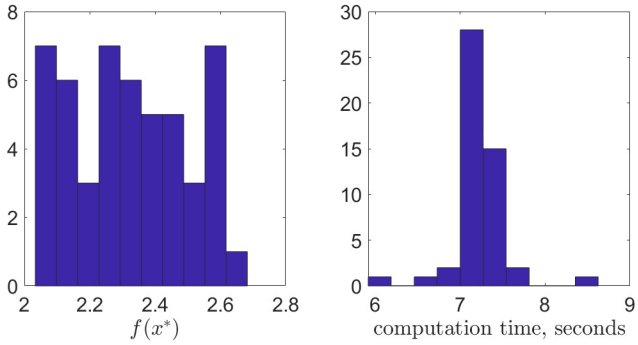


Fig. 10. Industrial chemical process problem: distribution of the achieved objective values (left) and distribution of the required time to converge (right)

We finally remark that IPM is an alternative solver for this problem, but it suffers the increase of required memory to store the Hessian matrix; see the computational complexity analysis in Section V-C.

In conclusion, this experiment proves that the proposed feedback linearization approach is valuable in large-scale, non-convex optimization problems arising from real-world applications. Moreover, using squared-slack variables is a feasible approach to deal with inequality constraints in this example, even if it increases the number of optimization variables and introduces additional non-convex constraints.

E. Robust optimization in the presence of inexact data

As noticed in Sec. IV, FL-CMO requires the inversion of $A(x) \in \mathbb{R}^{m \times m}$, which is a crucial point from different perspectives. On the one hand, the inverse of a poorly conditioned matrix is sensitive to small perturbations in the data. On the other hand, inversion is computationally intense, and one can use approximate solutions to reduce the burden. We deal with an inexact matrix $A(x)^{-1}$ in both cases.

In the following example, we demonstrate how the flexibility in the design of the controller $\mathcal{G}$ for FL-CMO leads to algorithms characterized by different robustness properties in the presence of such inaccurate data. Moreover, we compare PI-CMO and FL-CMO equipped with a PI control law for $v(t)$.

We consider the quadratic problem

$$
\min_{x \in \mathbb{R}^3} \frac{1}{2} x^\top \begin{pmatrix} 1 & 0 & 1 \\ 0 & 4 & -2 \\ 1 & -2 & 8 \end{pmatrix} x + \begin{pmatrix} -1 \\ 2 \\ -1 \end{pmatrix}^\top x \tag{105a}
$$

$$
\text{s.t.} \quad x_1 + 1 = 0, \tag{105b}
$$

$$
3x_1 + 2x_2 - 4x_3 = 0. \tag{105c}
$$

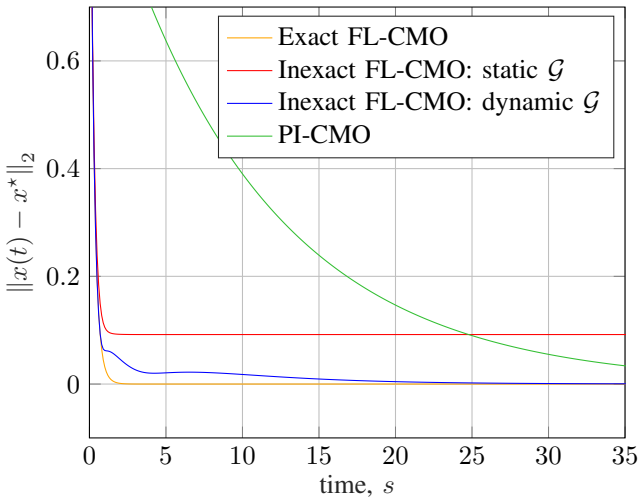


Fig. 11. Perturbed quadratic problem: optimization variables' trajectories $x(t)$ for the different algorithms. Exact FL-CMO refers to the unperturbed case.

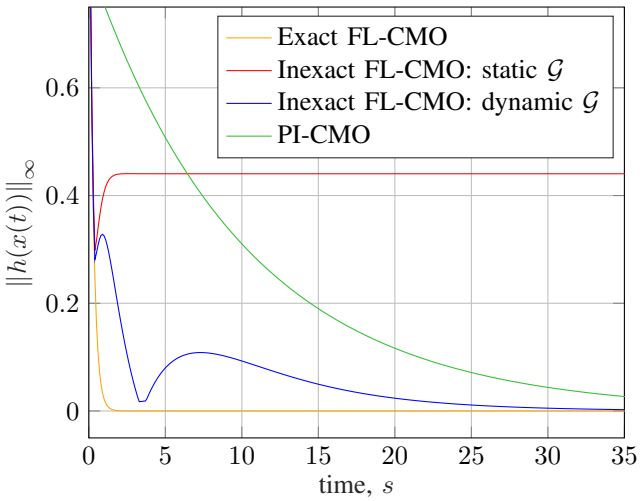


Fig. 12. Perturbed quadratic problem: optimization variables' constraints value $h(x(t))$ for the different algorithms. Exact FL-CMO refers to the unperturbed case.

Then $A(x)^{-1} = -\left(CC^\top\right)^{-1} = \begin{pmatrix} -1.45 & 0.15 \\ 0.15 & -0.05 \end{pmatrix}$. Based on previous considerations, we assume to know only a perturbed version $A(x)^{-1} \odot (\mathbf{1}_2 + P)$ where $\mathbf{1}_2$ is the 2×2 matrix of all ones, $\odot$ is the Hadamard product, and $P \in \mathbb{R}^{2,2}$ is such that each of its entries P_{ij} is bounded by $|P_{ij}| \leq 0.25$, for $1 \leq i, j \leq 2$.

For FL-CMO, we consider the static controller $v(t) = \mathcal{G}(y(t)) = Ky(t)$ with $K = -4$, which retraces the algorithm proposed in [18], and the dynamic PI controller

$$v(t) = \mathcal{G}(y(t)) = k_p y(t) + k_i \int_0^t y(\tau)\mathrm{d}\tau. \tag{106}$$

where $k_i = -1$ and $k_p = -4$. Additionally, we implement FL-CMO with exact $A(x)^{-1}$ as a benchmark. Finally, we also consider PI-CMO with the same coefficients k_i and k_p for further comparison.

We perform 40 simulations of the four dynamics, where for each run, we start from different initial conditions and sample different P_{ij} from the uniform distribution in $[-0.25, 0.25]$. We show the simulation results in terms of the average distance from the optimum $\|x(t) - x^\star\|_2$ and the maximum constraint violation $\|h(x(t))\|_\infty$ in Fig. 11 and Fig. 12, respectively. As expected, the exact FL-CMO and PI-CMO converge to the correct solution. Interestingly, we can see that the pole placement controller makes FL-CMO converge quickly to a wrong solution under the effect of perturbation. On the other hand, by introducing the integral term in $\mathcal{G}$, the inexact FL-CMO converges to the correct solution more quickly than PI-CMO.

In conclusion, this example suggests that FL-CMO with a dynamic controller is more robust to data perturbation and inaccuracies. Future work will analyze the robustness properties of FL-CMO in more general settings.

VI. Conclusion

This paper presents a control-theoretic approach to develop new algorithms for convex and non-convex optimization with equality constraints. Based on the first-order necessary conditions, we show that the considered class of optimization problems is equivalent to a class of stabilization and output regulation problems. In this context, the Lagrange multipliers of the optimization problem serve as the control inputs for the dynamical system being regulated. We explore two methods for designing this control: Proportional-Integral (PI) control and feedback linearization. Our analysis provides a theoretical framework for assessing the convergence properties of both methods. We rigorously prove that the PI control method converges for strongly convex problems and offer a detailed evaluation of its convergence rate. Additionally, we establish the local convergence of the feedback linearization method for non-convex problems, as well as its global convergence for strongly convex problems. Furthermore, we conduct experimental tests of both methods, validating our theoretical findings and highlighting the practical effectiveness of our proposed framework, especially in comparison to state-of-the-art optimization algorithms.

Future work will focus on developing methods that handle inequality constraints more effectively than the approach of using squared-slack variables and non-differentiable cost functions.

References

[1] N. Parikh and S. Boyd, "Proximal algorithms," *Foundations and Trends in Optimization*, vol. 1, no. 3, pp. 127–239, 2014.
[2] A. Saxe, J. McClelland, and S. Ganguli, "Exact solutions to the nonlinear dynamics of learning in deep linear neural networks," in *Proc. Inter. Conf. Learn. Rep. (ICLR)*, 2014.
[3] O. Elkabetz and N. Cohen, "Continuous vs. discrete optimization of deep neural networks," in *Proc. Adv. Neural Infor. Process. Syst. (NeurIPS)*, vol. 34, 2021, pp. 4947–4960.
[4] W. Su, S. Boyd, and E. J. Candès, "A differential equation for modeling Nesterov's accelerated gradient method: Theory and insights," *J. Mach. Learn. Res.*, vol. 17, no. 153, pp. 1–43, 2016.
[5] M. Muehlebach and M. Jordan, "A dynamical systems perspective on nesterov acceleration," in *Proc. Int. Conf. Mach. Learn. (ICML)*, 2019, pp. 4656–4662.
[6] D. G. Luenberger and Y. Ye, *Linear and Nonlinear Programming*, 4th ed. Springer Publishing Company, Inc., 2015.

[7] T. Kose, "Solutions of saddle value problems by differential equations," *Econometrica*, vol. 24, no. 1, pp. 59–70, 1956.

[8] K. J. Arrow, L. Hurwicz, and H. B. Chenery, *Studies in linear and non-linear programming*. Stanford University Press, 1958.

[9] G. Qu and N. Li, "On the exponential stability of primal-dual gradient dynamics," *IEEE Control Syst. Lett.*, vol. 3, no. 1, pp. 43–48, 2019.

[10] D. Ding and M. R. Jovanović, "Global exponential stability of primal-dual gradient flow dynamics based on the proximal augmented lagrangian," in *Proc. Amer. Control Conf. (ACC)*, 2019, pp. 3414–3419.

[11] N. K. Dhingra, S. Z. Khong, and M. R. Jovanović, "The proximal augmented lagrangian method for nonsmooth composite optimization," *IEEE Trans. Autom. Control*, vol. 64, no. 7, pp. 2861–2868, 2019.

[12] Z. Chen, X. Li, L. Yang, J. Haupt, and T. Zhao, "On constrained nonconvex stochastic optimization: A case study for generalized eigenvalue decomposition," in *Proc. Int. Conf. Arti. Intell. Stat. (AISTATS)*, vol. 89, 2019, pp. 916–925.

[13] G. França, D. P. Robinson, and R. Vidal, "ADMM and accelerated ADMM as continuous dynamical systems," in *Proc. Int. Conf. Mach. Learn. (ICML)*, 2018, pp. 1559–1567.

[14] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Found. Trends Mach. Learn.*, vol. 3, no. 1, pp. 1 – 122, 2010.

[15] H. Yamashita, "A differential equation approach to nonlinear programming," *Math. Program.*, vol. 18, no. 1, pp. 155–168, 1980.

[16] J. Schropp and I. Singer, "A dynamical systems approach to constrained minimization," *Numer. Funct. Anal. Optim.*, vol. 21, pp. 537–551, 2000.

[17] L. Zhou, Y. Wu, L. Zhang, and G. Zhang, "Convergence analysis of a differential equation approach for solving nonlinear programming problems," *Appl. Math. Comput.*, vol. 184, no. 2, pp. 789–797, 2007.

[18] A. Allibhoy and J. Cortés, "Control-barrier-function-based design of gradient flows for constrained nonlinear programming," *IEEE Trans. Autom. Control*, vol. 69, no. 6, pp. 3499–3514, 2024.

[19] D. Bertsekas, *Nonlinear Programming*, 2nd ed. Belmont, Massachusetts: Athena Scientific, 1999.

[20] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016.

[21] L. Lessard, B. Recht, and A. Packard, "Analysis and design of optimization algorithms via integral quadratic constraints," *SIAM J. Optim.*, vol. 26, no. 1, pp. 57–95, 2016.

[22] B. Hu and L. Lessard, "Control interpretations for first-order optimization methods," in *Amer. Control Conf. (ACC)*, 2017, pp. 3114–3119.

[23] Y. S. Xia and J. Wang, "On the stability of globally projected dynamical systems," *J. Optim. Theory Appl.*, vol. 106, no. 1, pp. 129–150, 2000.

[24] J. Cortés, "Discontinuous dynamical systems," *IEEE Control Sys. Mag.*, vol. 28, no. 3, pp. 36–73, 2008.

[25] D. Meng and Y. Wu, "Control design for iterative methods in solving linear algebraic equations," *IEEE Trans. Autom. Control*, vol. 67, no. 10, pp. 5039–5054, 2022.

[26] D. Meng, "Feedback of control on mathematics: Bettering iterative methods by observer system design," *IEEE Trans. Autom. Control*, vol. 68, no. 4, pp. 2498–2505, 2023.

[27] A. Bhaya and E. Kaszkurewicz, "A control-theoretic approach to the design of zero finding numerical methods," *IEEE Trans. Autom. Control*, vol. 52, no. 6, pp. 1014–1026, 2007.

[28] M. Mönnigmann, "Efficient calculation of bounds on spectra of Hessian matrices," *SIAM J. Sci. Comput.*, vol. 30, no. 5, pp. 2340–2357, 2008.

[29] A. Isidori, *Nonlinear Control Systems*. Springer London, 1995.

[30] H. K. Khalil, *Nonlinear Systems*, 3rd ed., ser. Pearson Education. Prentice Hall, 2002.

[31] C. I. Byrnes and A. Isidori, "A frequency domain philosophy for nonlinear systems, with applications to stabilization and to adaptive control," in *IEEE Conf. Decis. Control (CDC)*, 1984, pp. 1569–1573.

[32] A. Isidori, "The zero dynamics of a nonlinear system: From the origin to the latest progresses of a long successful story," *Europ. J. Control*, vol. 19, no. 5, pp. 369–378, 2013.

[33] J. Nocedal and S. J. Wright, *Numerical optimization*. Springer, 2006.

[34] P. Dorato, V. Cerone, and C. Abdallah, *Linear-Quadratic Control: An Introduction*. USA: Simon & Schuster, Inc., 1994.

[35] M. Hong, M. Razaviyayn, and J. Lee, "Gradient primal-dual algorithm converges to second-order stationary solution for nonconvex distributed optimization over networks," in *Proc. Inter. Conf. Machine Learn. (ICML)*, vol. 80, 2018, pp. 2009–2018.

[36] K. Tanabe, "A geometric method in nonlinear programming," *J. Optim. Theory Appl.*, vol. 30, no. 2, pp. 181–210, 1980.

[37] R. A. Horn and C. R. Johnson, *Matrix Analysis*, 2nd ed. Cambridge University Press, 2013.

[38] A. Aggarwal and C. Floudas, "Synthesis of general distillation sequences-nonsharp separations," *Comput. Chemical Eng.*, vol. 14, no. 6, pp. 631–653, 1990.

[39] A. Kumar, G. Wu, M. Z. Ali, R. Mallipeddi, P. N. Suganthan, and S. Das, "A test-suite of non-convex constrained optimization problems from the real-world and some baseline results," *Swarm Evolut. Comput.*, vol. 56, p. 100693, 2020.



Vito Cerone received the Laurea degree in Electronic Engineering in 1984 and the PhD in System Engineering in 1989 from the Politecnico di Torino, Italy. From 1989 to 1990, he has been a Research Fellow at the School of Electronic and Electrical Engineering, University of Birmingham (UK). From 1989 to 1998, he has been an Assistant Professor with the Department of Control and Computer Engineering at the Politecnico di Torino, where he has been an Associate Professor from 1998 to 2021. Since 2021 he has been a Full Professor at the Politecnico di Torino. He is a co-author of Linear Quadratic Control: An Introduction. His main research interests include system identification, parameter estimation, optimization, and control, with applications to automotive systems.



Sophie M. Fosson (Member, IEEE) received the M.Sc. degree in applied mathematics from the Politecnico di Torino, Italy, in 2005 and the Ph.D. degree in mathematics for the industrial technologies from Scuola Normale Superiore di Pisa, Italy, in 2011. From 2012 to 2016, she was a Postdoctoral Associate with the Department of Electronics and Telecommunications, Politecnico di Torino. She visited the Centre Tecnològic de Telecomunicacions de Catalunya, Spain, in 2013, 2014 and 2016. She was researcher at Istituto Superiore Mario Boella, Turin, Italy, in 2017. She is currently an Associate Professor with the Department of Control and Computer Engineering, Politecnico di Torino. She is Associate Editor for the IEEE Control Systems Letters. Her main research interests include sparse optimization, machine learning, system identification, control and cyber-physical systems.



Simone Pirrera received his bachelor's degree in electronic engineering in 2019 and the master's degree in mechatronic engineering in 2021, both from Politecnico di Torino. Currently, he is a Ph.D. student in the System Identification and Control (SIC) group of the Department of Control and Computer Engineering (DAUIN) in Politecnico di Torino. His main interests are related to system identification, robust and data-driven control, mathematical optimization, and machine learning.



Diego Regruto received the Laurea degree in Electronic Engineering and the Ph.D. degree in System Engineering from the Politecnico di Torino, Torino, Italy, in 2000 and 2004, respectively. He is an Associate Professor of Control Theory with the Dipartimento di Automatica e Informatica, Politecnico di Torino.

Prof. Regruto chaired the IEEE CSS Technical Committee on System Identification and Adaptive Control (TC-SIAC) from January 2013 to December 2015. He has served as Associate Editor of the IEEE Transactions on Automatic Control (from January 2016 to December 2022) and as Associate Editor of the IEEE Conference Editorial Board (from June 2013 to December 2020). Prof. Regruto is a Senior Member of IEEE. His main research interests are in the fields of system identification, optimization and data-driven robust control methods, with application to automotive and biomedical engineering problems.