Embedded model control of networked control systems: An experimental robotic application

Contents lists available at ScienceDirect

Mechatronics

journal homepage: www.elsevier.com/locate/mechatronics

Embedded model control of networked control systems: An experimental robotic application[☆]

Luca Nanu [a,*], Luigi Colangelo [c], Carlo Novara [c], Carlos Perez Montenegro [b]

[a] Department of Mechanical and Aerospace Engineering, Polytechnic of Turin, Corso Duca degli Abruzzi, 24, Turin, 10129, Italy
[b] With Polytechnic of Turin, Corso Duca degli Abruzzi, 24, Turin, 10129, Italy
[c] Department of Electronics and Telecommunications Engineering, Polytechnic of Turin, Corso Duca degli Abruzzi, 24, Turin, 10129, Italy

ARTICLE INFO

Keywords:
Networked control systems
Model-based control
Robot control
Active disturbance control

ABSTRACT

In Networked Control System (NCS), the absence of physical communication links in the loop leads to relevant issues, such as measurement delays and asynchronous execution of the control commands. In general, these issues may significantly compromise the performance of the NCS, possibly causing unstable behaviours. This paper presents an original approach to the design of a complete digital control unit for a system characterized by a varying sampling time and asynchronous command execution. The approach is based on the Embedded Model Control (EMC) methodology, whose key feature is the estimation of the disturbances, errors and nonlinearities affecting the plant to control and their online cancellation. In this way, measurement delays and execution asynchronicity are treated as errors and rejected up to a given frequency by the EMC unit. The effectiveness of the proposed approach is demonstrated in a real-world case-study, where the NCS consists of a differential-drive mobile robot (the plant) and a control unit, and the two subsystems communicate through the web without physical connection links. After a preliminary verification using a high-fidelity numerical simulator, the designed controller is validated in several experimental tests, carried out on a real-time embedded system incorporated in the robotic platform.

1. Introduction

In recent years, the increase in complexity of control systems and control units design has introduced as by-products several conceptual and design innovations. Sensors, whose information does not necessarily has a physical connection to the plant to be controlled, and control algorithms, whose execution can be planned to be performed remotely, are examples of such trend. In the majority of these cases, a reliable communication system, connecting the plant with the sensors and the actuators, is required, thus paving the way to Networked Control System (NCS) architectures. [1–4] review the main characteristics and applications of NCS architectures, as well as the most used control algorithms. The fact that NCS have a communication network between the control unit and the process to control adds additional undesired phenomena, such as delays and/or loss of information (package dropouts). When the delays and package dropouts are not taken into account from the early design phase, they can reduce the control capabilities in NCS scenarios.

These problems have been studied and treated considering different control strategies. In [5], a predictive control method for NCS is presented, where the network links are subject to random packet dropouts. Differently [6] tackles the problem of a delay-dependent asymptotic stability analysis, for neural networks with time-varying delays. From the design perspective, [7] proposes an observer for the state and disturbance estimation for electro-hydraulic-actuator systems. The work [8] investigates a time-varying observer, for a linear continuous-time plant with asynchronous sampled measurements, within an hybrid systems framework. In [9], an approach to the asynchronous sensing problem, based on event-triggered sub-count estimation, is presented. In [10], the NCS control problem is tackled with a fuzzy-model-based nonlinear system, showing a review of the recent advances. In [11], Sliding Mode Control methods for NCS are reviewed, showing the performances with time-delay, packet losses and different communication protocols. However, even these state-of-the-art approaches may show limitations due to their difficulty in dealing with large disturbances, significant model uncertainties, and not well-known nonlinearities (real NCS architectures are typically affected by these kinds of issues), [12–14].

[☆] This paper was recommended for publication by Associate Editor Qining Wang.

[*] Corresponding author.

E-mail addresses: luca.nanu@polito.it (L. Nanu), luigi.colangelo@polito.it (L. Colangelo), carlo.novara@polito.it (C. Novara), carlos.perez@polito.it (C. Perez Montenegro).

https://doi.org/10.1016/j.mechatronics.2024.103160
Received 10 March 2023; Received in revised form 6 January 2024; Accepted 14 February 2024
Available online 2 March 2024

Nomenclature

Embedded Model Control (EMC)

NCS	Networked Control System
$\lambda_K, \lambda_N, \lambda_R$	Discrete-time eigenvalues (control law, noise estimator, reference dynamics)
$\bar{\mathbf{e}}$	Tracking error
$\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{G}$	State, control, output and disturbance matrices
$\mathbf{e}_m$	model output error
$\mathbf{K}_c$	Feedback gains
$\mathbf{K}_r, \mathbf{N}_r$	Reference generator gains
$\mathbf{L}$	Closed-loop prediction matrix
$\mathbf{M}, \mathbf{Q}$	Rejector matrices
$\mathbf{u}$	Control/command input (Control law)
$\mathbf{u}_d$	Disturbance rejection term of $\mathbf{u}$
$\mathbf{u}_{trk}$	State feedback of $\mathbf{u}$
μ_K, μ_N, μ_R	Continuous-time eigenvalues (control law, noise estimator, reference dynamics)
$\bar{\mathbf{r}}$	Reference trajectory
$\bar{\mathbf{u}}$	Feed-forward component of $\mathbf{u}$
$\bar{\mathbf{x}}$	Reference state
T_s	Sampling time
$\bar{\mathbf{w}}$	Noise vector
$\mathbf{u}$	Control input
$\mathbf{x}$	State
$\mathbf{x}_c$	Controllable (canonical) state
$\mathbf{x}_d$	Rejector state
$\mathbf{y}$	Measured output
$\mathbf{y}_m$	Estimated output from the internal model

DC motor

β_{tot}	Motor friction
ω	Motor angular speed (after gearbox reduction N)
ω_m	Motor angular speed
τ_a	Motor electrical time constant
τ_m	Motor mechanical time constant
e	Back-electromotive force
i_a	Armature current
J_{tot}	Motor inertia
k_t	Motor torque constant
k_v	Motor back electromotive force constant
k'_v	Modified motor back electromotive force constant
L_a	Motor armature inductance
N	Motor gearbox reduction
R_a	Motor armature resistance
T_m	Motor torque
T_r	Resistive torque (motor)
V_a	Motor armature voltage
PWM	Pulse-Width Modulation

PID and LQR controllers

$K_P, K_I, K_D, N_{\text{PID}}$	PID controller gains
K_{LQR}	LQR controller gains

The aforementioned limitations are overcome by means of a design approach based on the disturbance rejection principle. Generally speaking, disturbance rejection is a significant research area in control theory and practice. The disturbance rejection, still a crucial problem within the automatic control research, [15], is often regarded as a way to enhance the applicability of the design solution and its potential impact on engineering practice, [16–18]. Within this domain, several research contributions have been developed: for example, [19] proposes a three disturbance observer and boundary controllers for a flexible Timoshenko manipulator control, [20] shows an adaptive disturbance observer for an active suspension system, instead [21] reports an high order sliding-mode observer for external torque estimation of robot manipulators. Among different disturbance rejection methods, several model-based techniques can be found in the literature, such as the Active Disturbance Rejection Control (ADRC), [15,22–24], Adaptive Robust Control (ARC), [25,26], and EMC, [16,27–30]. These methods are based on the estimation and online cancellation of the disturbances/uncertainties affecting the plant to control.

Here is a concise comparison between EMC and ADRC, taken from [31]: (i) In general, the ADRC does not consider any limitation to the control bandwidth, hence the separation between the control law and the uncertainty estimation for the ADRC is more difficult, whereas the EMC is compelled to find out an optimal BW in the presence of uncertainty. Furthermore, the EMC extends the ADRC concepts with the following points, (ii) The 'unknown' disturbance dynamics may be any and is driven by an unpredictable noise to be real-time estimated; (iii) The noise input layout may establish that not all the state variables are directly affected by uncertainty; (iv) the noise input is estimated by a dynamic feedback driven by the model output error. Such guidelines, from (i) to (iv), facilitate separation between uncertainty estimation and control design. The main difference between ADRC and EMC is that the former assumes that model output errors can be treated like input disturbances, whereas EMC shows that high frequency neglected dynamics cannot be treated as such. This result is further explained in Section 3.1, with a stability analysis study.

Looking at the ARC method, it employs a different method compared to the EMC, since it is based on an adaptive law to estimate the unknown disturbances. Instead, an observer to estimate the unknown disturbances is the core of the EMC approach and, as discussed above in the comparison with the ADRC method, it is particularly suitable to estimate and reject the effects of delays, asynchronous commands and other disturbances. A second difference is that the EMC is a model-based design method, which employs a simplified version of the plant to be controlled, the internal model, embedded in the EMC unit. The model neglected disturbances are estimated by the disturbance observer, and rejected by a suitable control law. This allows to facilitate the construction of the model for the plant to be controlled, and to avoid the plant dynamics difficult to be modelled.

In any case, the fact that EMC is effective in solving the networked system control problem does not exclude that other approaches may work as well. Both ARC and ADRC have the potential to correctly work when applied to a networked system. However, this investigation would require an extensive presentation/discussion, which would be out of the scope of the paper. In Section 5 of our paper, we decided to compare the EMC controller with controllers obtained by means of standard techniques, such as PID and LQR controllers.

The main contributions of this paper are two: (i) the standard formulation of EMC is modified to allow the control of asynchronous time systems, such as the NCS: an original approach is developed for the design of a complete asynchronous control unit; (ii) the present study expands the seminal ideas of [30], introducing experimental tests of the designed EMC design framework with asynchronous timing control. In terms of methodology, the differences between the EMC version proposed in the present paper and [30] are: (i) the study of several models for the disturbance estimator (the choices comprise static or dynamic observers), finally choosing the best in terms of control performances; (ii) a simplified version of the plant is controlled, the internal model, embedded in the EMC unit. Moreover, in terms of experimental results, the main improvement of this work compared to

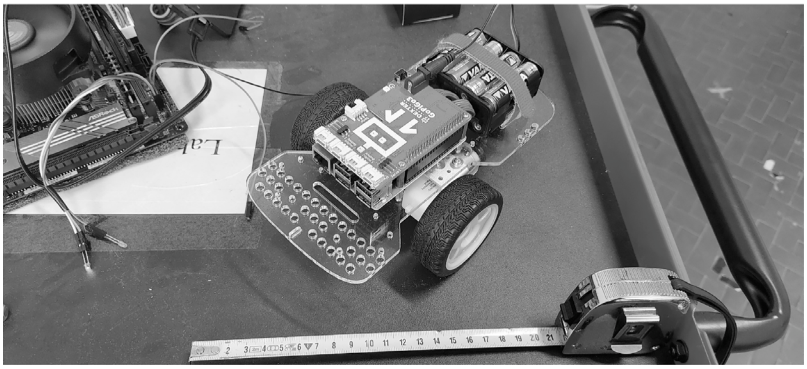


Fig. 1. Differential-drive robot experimental setup.

its predecessor, is that the EMC approach was tested not only with numerical simulations, but with several experimental tests, too. The experiments of the EMC validate its effectiveness with a real test case, a differential-drive mobile robot in an NCS scenario, [16,27].

Specifically, as basis of the experimental setup, the designed digital control unit, converted into C++ code via a robust automatic code generation from MathWorks Simulink, [32], was executed in a Raspberry Pi board able to command the mobile robot remotely. These tests show that the proposed disturbance rejection-based approach can be effective in allowing precise reference tracking under persisting external disturbances, coming both from uncertain parameters/dynamics and asynchronous sampling time in NCS scenario.

As a result, differently from other design methods, the complex asynchronous control design problem can be cast into a simpler problem: the active estimation and rejection of the generalized disturbance dynamics. In fact, following the EMC disturbance rejection design principles, the controller was designed based on an input–output controllable model and it is fixed, whereas any difference between the controllable model and the reality is interpreted as a disturbance, to be real-time estimated and rejected, [27]. This implies forcing the plant physics to behave like the controllable model. Consequently, the knowledge of the plant dynamics required for the control unit design results to be reduced to a dramatic extent, thus an accurate model plant is not strictly required. This substantially increases the practical applicability of the designed solution, [15,33]. In addition, the disturbance rejection control strategy is able to reduce the control effort to a minimum. This is an advantage compared to other state-feedback controls, which may require a large command activity, [33].

Among the disturbance rejection control methods, the EMC guarantees a systematic approach in the field of the asynchronous timing systems. This means that the proposed method is specifically designed for dealing with the disturbances generated by a system with a variable sampling time, such as the NCS scenario. On the contrary, other methodologies results to be adapted only to the asynchronous system, such as the PID control method. To understand the aforementioned EMC advantages, a comparison between the EMC and the PID control methods is presented in Section 5, related to the experimental tests of the paper. As a further result, stability and robustness properties of the EMC method for the specific test case are guaranteed, [33].

This paper is organized as follows. Section 2 reviews the distinctive features of the adopted mobile robot experimental test-bench and the NCS scenario setup exploited for the robot tests. An overview of the EMC design principles is presented in Section 3, and the asynchronous EMC equations are derived for the specific mobile robot setup in Section 4. In Section 5, the results of the designed asynchronous controller experimental tests are reported and discussed. Finally, conclusions are drawn in Section 6.

2. Experimental test-bench setup

2.1. Differential-drive mobile robot

The platform selected to test the disturbance-rejection-based asynchronous controller presented in this study is a differential-drive mobile robot, in which a real-time embedded system is incorporated.

The adopted mobile robot (which model is called GoPiGo3, manufactured by Dexter Industries company, [34]) is characterized by two wheels (left and right) independently actuated by two DC motors, Fig. 1, whose command voltage input is set via a Pulse-Width Modulation (PWM) square wave with a duty cycle from 0 to the maximum voltage value. At the DC motor tail ends, the platform includes magnetic encoders, directly connected with the robot wheels rotating shaft, to estimate the wheels angular position. A gearbox reduces the DC motor speed by a factor of $N = 120$ to establish a suitable load speed level, at the two robot wheels. Finally, the power supply is guaranteed by on-board batteries. In addition, two main boards are used to control the robot: (i) an integrated hardware board, and (ii) a Raspberry Pi 3 Model B board. The integrated hardware board, directly supplied by the manufacturer, provides the firmware, the communication interfaces, an H-bridge to change direction and sign of the two motors output speed, and a PWM generator, required to adjust the input voltage amplitude and consequently the desired output speed of the wheels. Conversely, the Raspberry Pi 3 Model B board is leveraged to control all the motor functionalities as well as to implement the NCS scenario. In Table A.4 of Appendix A, the main features of the robotic platform are listed.

2.2. NCS experimental setup

To be fully representative of a Networked Control System scenario, the experimental tests presented in this study are characterized by a software emulation of asynchronous and variable sampling times, via a random number generator directly implemented in the Raspberry Pi control board. More in detail, uniform random numbers in a predefined range are defined via the Raspberry Pi control board, to be then leveraged by real hardware timers to obtain the required discrete sampling time.

As a matter of fact, such a design choice is devised so to have a representative yet conservative test environment. Indeed, the range of the random number generator is chosen to be conservative in the delays produced, in order to further increase the fidelity of the experimental test-bench. The resulting test setup, in which the time-stamp is received with a certain variable delay, allowed to account for the control inputs to be sent remotely.

Most notably, the established NCS software emulation environment has allowed the authors to arbitrarily shape a wide range of possible disturbance scenarios via a remote control, without the need of setting up remote physical connection between the robot and remote control devices. Inter alia, the setup also has allowed to impose critical conditions, like very high sampling time variabilities, to experimentally validate at larger extent the EMC asynchronous control unit.

3. Asynchronous EMC design principles

This section generally details the building blocks and the governing equations of the EMC asynchronous control unit. Thereafter, in Section 4, the asynchronous EMC will be specifically designed to control the mobile robot in an NCS environment where asynchronous sampling times affect the whole system.

The typical EMC-controlled system (cf. Fig. 2) is mainly composed by a digital control unit coupled with the system to be controlled or its fine model (potentially, its digital twin). Inside the digital control unit, the system to be controlled is modelled by means of a controllable dynamics, in a block called internal model. In addition to the internal model, the digital control unit typically includes: (i) a state predictor, (ii) a reference dynamics, providing both the reference command, and (iii) the control law, including a disturbance-rejector term, [15]. In particular, the state predictor is based on an internal model of the system to be controlled, completed by an output-to-state feedback (the noise estimator, in Fig. 2).

The EMC internal model is written in discrete-time form and includes, in addition to the controllable dynamics, a disturbance model (called disturbance dynamics, see Fig. 2), whose structure is designed to fit the class of control problems to be addressed, [27]. This disturbance model consists in a dynamic model of the unknown disturbances and the parameter uncertainty, potentially affecting the plant, not included into the controllable input–output model. As a result, the disturbance model plays the role of the disturbances rejector, [15]. The EMC state equations are as follows [27]:

$$\begin{aligned}\mathbf{x}(k+1) &= \begin{bmatrix}\mathbf{x}_c\\ \mathbf{x}_d\end{bmatrix}(k+1) = \underbrace{\begin{bmatrix}\mathbf{A}_c & \mathbf{H}_c\\ \mathbf{0}_{n_{x_d}\times n_{x_c}} & \mathbf{A}_d\end{bmatrix}}_{\mathbf{A}}\mathbf{x}(k) + \\ &+ \underbrace{\begin{bmatrix}\mathbf{B}_c\\ \mathbf{B}_d\end{bmatrix}}_{\mathbf{B}}\mathbf{u}(k) + \underbrace{\begin{bmatrix}\mathbf{G}_c\\ \mathbf{G}_d\end{bmatrix}}_{\mathbf{G}}\overline{\mathbf{w}}(k), \\ \mathbf{y}_m(k) &= \underbrace{\begin{bmatrix}\mathbf{C}_c & \mathbf{C}_d\end{bmatrix}}_{\mathbf{C}}\mathbf{x}(k),\end{aligned} \tag{1}$$

where $k \in \mathbb{Z}$ is the discrete-time index, $\mathbf{x}_c$ (with dimension n_{x_c}) and $\mathbf{x}_d$ (with dimension n_{x_d}) refer to the controllable (or canonical) and the rejector states, respectively. The matrices $\mathbf{A}, \mathbf{B}, \mathbf{C}$ are related to the state $\mathbf{x}$, control input $\mathbf{u}$ and estimated output from the internal model $\mathbf{y}_m$, respectively. The matrix $\mathbf{G}$ is present, for the estimation of the noise vector $\overline{\mathbf{w}}$.

The noise vector $\overline{\mathbf{w}}$ includes bounded nonlinearities, unmodelled dynamics, external disturbances, sensor noises and discretization errors. In addition, the control disturbances due to delays and package dropouts of an NCS scenario are included. As in [33,35], we assume that $\overline{\mathbf{w}}$ belongs to an uncertainty set $\mathcal{W}$ of bounded, zero-mean signals respecting $|\overline{\mathbf{w}}| \leq \mathbf{w}_{max}$.

In particular, the matrix $\mathbf{A}$ is made by the sub-matrix $\mathbf{A}_c$ related to $\mathbf{x}_c$, $\mathbf{A}_d$ related to $\mathbf{x}_d$, and the $\mathbf{H}_c$ matrix, which describes the interactions of $\mathbf{x}_d$ on $\mathbf{x}_c$. Similarly, there are the sub-matrices $\mathbf{B}_c, \mathbf{B}_d$ for $\mathbf{B}$ matrix, $\mathbf{G}_c, \mathbf{G}_d$ for $\mathbf{G}$ matrix, and $\mathbf{C}_c, \mathbf{C}_d$ for $\mathbf{C}$ matrix.

Given the controllable and disturbance models, the noise estimator allows us to close the output-to-state feedback loop, thus building a digital state predictor. The state predictor provides the control law with a reliable estimate of: (i) the controllable states $\mathbf{x}_c$, to be driven according to the reference dynamics, and (ii) the disturbances affecting the plant $\mathbf{x}_d$, to be rejected. Hence, the noise estimator, with the aim of recovering the disturbance states from the output measurements, is crucial for active disturbance rejection control, [15,27]. The noise estimator is fed with the model output error (cf. $\mathbf{e}_m$ in Fig. 2), namely the difference of plant measurement $\mathbf{y}$ and the estimated output from internal model $\mathbf{y}_m$ (the estimated state $\mathbf{x}_c$ in this case coincides with the output $\mathbf{y}_m$). As a result, it holds that

$$\begin{aligned}\overline{\mathbf{w}}(k) &= \mathbf{L}\mathbf{e}_m(k), \\ \mathbf{e}_m(k) &= \mathbf{y}(k) - \mathbf{y}_m(k),\end{aligned} \tag{2}$$

where $\mathbf{L} = \left[l_1, \ldots, l_i\right]^T$ is the closed-loop predictor matrix, with $i = n_w$ as the number of $\overline{\mathbf{w}}$ noise vector components. The close-loop predictor gains $\left[l_1, \ldots, l_i\right]$ need to be designed and tuned to guarantee closed-loop stability, cf. Section 4.3.

To conclude, the EMC control law can be expressed as

$$\begin{aligned}\mathbf{u}(k) &= \overline{\mathbf{u}}(k) + \mathbf{K}_C\overline{\mathbf{e}}(k) - \mathbf{M}\mathbf{x}_d(k), \\ \overline{\mathbf{e}}(k) &= \left(\overline{\mathbf{x}}(k) - \mathbf{Q}\mathbf{x}_d(k)\right) - \mathbf{x}_c(k).\end{aligned} \tag{3}$$

Eq. (3) is made up by three terms (cf. Fig. 2): (i) the feed-forward component $\overline{\mathbf{u}}$ obtained as the control input to generate a desired reference trajectory, (ii) the state feedback $\mathbf{u}_{trk} = \mathbf{K}_C\overline{\mathbf{e}}$, being $\overline{\mathbf{e}}$ the tracking error, and (iii) the disturbance rejection term $\mathbf{u}_d = \mathbf{M}\mathbf{x}_d$. $\overline{\mathbf{x}}$ is the desired reference trajectory.

The EMC formulation in (1) is similar to a GESO (Generalized Extended State Observer) formulation, since the mismatched disturbances of the system are inserted as additional states, and then estimated with an observer-state controller with the same form of (2). However, the difference between the EMC formulation and GESO resides on the application of the aforementioned controller not directly to the plant, but to a simplified model of the plant. Thanks to the application of suitably designed control gains $\mathbf{M}, \mathbf{Q}$, defined in (3), the mismatched disturbance can be compensated by the control action.

In Fig. 2, the basic EMC scheme is presented, highlighting the sub-blocks explained above, the state predictor with the internal model inside, the control law and the reference dynamics.

3.1. Stability analysis

In this subsection, we recap the robust stability result proven in [33], regarding a closed-loop system consisting of a nonlinear plant connected in feedback with an EMC controller. The recap is adapted to the framework developed in the present paper.

Consider a continuous-time nonlinear system in the following form:

$$\begin{aligned}\dot{\mathbf{x}}(t) &= \mathcal{A}\mathbf{x}(t) + \mathcal{G}\left(\mathcal{B}\mathbf{u}(t) + h(\mathbf{x}, t) + \mathbf{w}_x(t)\right) \\ \mathbf{y}(t) &= \mathcal{C}\mathbf{x}(t) + \mathbf{w}_y(t)\end{aligned} \tag{4}$$

where $\mathbf{x}$ is the state, $\mathbf{u}$ is the input, $\mathbf{y}$ is the output; $\mathbf{w}_x$ is the process disturbance and $\mathbf{w}_y$ is the measurement error; $\mathcal{A}$, $\mathcal{B}$, $\mathcal{G}$ and $\mathcal{C}$ are real matrices of compatible dimensions; the function $h(\cdot)$ contains all the system nonlinearities.

Suppose that the plant (4) is subject to the control law (3). That is, $\mathbf{u}(t)$ is converted in discrete-time domain $\mathbf{u}(k)$ by considering $t \in [k, k+1]T_s$ and $k = 0, 1, 2, \ldots$, where T_s is the adopted sampling time and $\mathbf{u}(k)$ is given by (3). This control law is based on the EMC model (1), which is constructed on the basis of the plant Eqs. (4). In particular, the Eqs. (4) are first discretized using a suitable sampling time T_s. Then, the matrices in (1) are directly obtained from the matrices in (4). The noise $\overline{\mathbf{w}}(k)$ in (1) captures the effects of disturbances and nonlinearities appearing in (4).

Note that the control law (3) is robust with respect to the nonlinearity $h(\cdot)$, in the sense that it does not use any detailed information about it. To guarantee closed-loop stability, $h(\cdot)$ is only needed to be sector-bounded. That is, each component h_j of h is assumed to satisfy the following inequalities:

$$-p_{jl} \leq h_j(\mathbf{x}, t)/x_l \leq p_{jl}, \ \forall j, l, \ \forall t \in [0, \infty) \tag{5}$$

where $p_{jl} < \infty$ and x_l are the components of $\mathbf{x}$.

When the plant (4) is subject to the control law (3), we obtain a closed-loop system with three inputs: the disturbances $\mathbf{w}_x$ and $\mathbf{w}_y$, and the reference $\overline{\mathbf{x}}$. Such a closed-loop system is named S_{CL}. Let $H(s)$ be

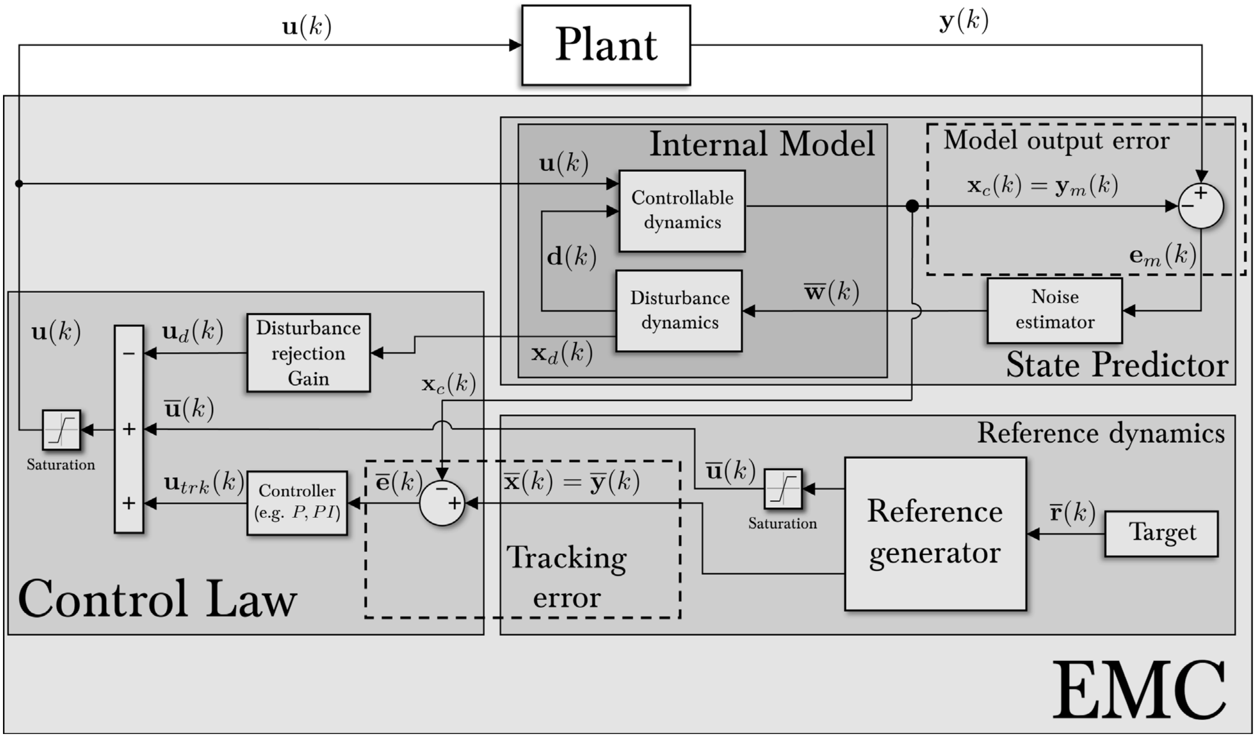


Fig. 2. EMC basic scheme.

the transfer function matrix of $\mathcal{S}_{CL}$ from $\mathbf{d}_{tot} = h + \mathbf{w}_x$ to $\mathbf{x}$. Assume that

$$\max_j \left(H_{j\infty,max}, p_{j,max}\right) < 1 \tag{6}$$

where $H_{j\infty,max} = \max_i \|H_{ij}\|_\infty$, H_{ij} are the components of H, $\|\cdot\|_\infty$ is the H-infinity norm, and $p_{j,max} > \max_l p_{jl}$.

According to the stability result given in [33], suitably adapted to the framework of the present paper, we have the following statement: Under assumptions (5) and (6), the closed-loop system $\mathcal{S}_{CL}$ is $\mathcal{L}^2$ *Finite-Gain Asymptotically Stable*. That is, the following properties hold:

(i) Finite constants λ_{wx}, λ_{wy}, λ_x and η exist, such that the state signal is bounded as

$$\|x\|_2 \le \lambda_{wx} \|\mathbf{w}_x\|_2 + \lambda_{wy} \left\|\mathbf{w}_y\right\|_2 + \lambda_x \|\overline{\mathbf{x}}\|_2 + \eta, \tag{7}$$

where $\|\cdot\|_2$ is the $\mathcal{L}^2$ signal norm.

(ii) Under null inputs $\mathbf{w}_x$, $\mathbf{w}_y$ and $\overline{\mathbf{x}}$,

$$\lim_{t\to\infty} \mathbf{x}(t) = 0, \tag{8}$$

for any bounded initial condition $\mathbf{x}(0)$.

These properties are robust, since they hold for all possible nonlinearities h which are sector-bounded according to (5) and (6). Inequality (7) implies that, if the disturbances $\mathbf{w}_x$ and $\mathbf{w}_y$, and the reference $\overline{\mathbf{x}}$ are bounded, the state signal is bounded as well. In the case of the mobile robot application considered in this paper, we have several disturbances and sources of uncertainty. All of them are captured by the noise vector $\overline{\mathbf{w}}$ in (1), which includes both bounded disturbances (like sensor noises, external disturbances, delays, package dropouts) and sector-bounded uncertainties (like linear unmodelled dynamics and discretization errors).

4. Asynchronous EMC application to differential-drive mobile robot control

This section is intended to derive the asynchronous control unit block specifically designed for the differential-drive mobile robot. The section is organized as follows: first an accurate model of the plant, i.e the fine model, is presented in Section 4.1: this is the digital twin of the platform. The fine model is needed both to test the designed algorithms, using a high-fidelity numerical simulator, and as a starting point to the internal model derivation. Secondly, in Section 4.2, a simplified model of the plant, namely the Internal Model, is derived from the fine one. The internal model plays a pivotal role in the EMC design, being embedded in the control unit as the basis to build the state and disturbance predictor (cf. Section 4.3 and Appendix B). Hence, the internal model needs to result as simple as suitable to be implemented in the remote control board of the mobile robot. Finally, the overall asynchronous digital control unit is completed via an appropriate control law (Section 4.5), leveraging the state and disturbance predictor output, jointly with a reference trajectory dynamics (Section 4.4).

4.1. Differential-drive mobile robot fine model

The fine model of a typical DC motor can be built by defining 2 state variables: (i) the armature current i_a and the motor angular speed ω_m, without considering a gearbox reduction N. These two state variables refer to the electrical and mechanical parts of the DC motor, respectively. The DC motor model equations, in the continuous time domain, are as follows:

$$\begin{cases} \dot{i}_a(t) = -\frac{R_a}{L_a} i_a(t) + \frac{1}{L_a}\left[V_a(t) - e(t)\right], \\ \dot{\omega}_m(t) = -\frac{\beta_{tot}}{J_{tot}}\omega_m(t) + \frac{1}{J_{tot}}\left[T_m(t) - T_r(t)\right], \end{cases}$$
$$i_a(0) = i_0, \quad \omega_m(0) = \omega_{m0}, \tag{9}$$
$$e(t) = k_v \omega_m(t),$$
$$T_m(t) = k_t i_a(t),$$

where T_m is the motor torque, depending on i_a, T_r is the resistive torque, due to possible external disturbances, while e is the back-electromotive force, depending on the motor output speed ω_m. The motor ω_m is then reduced through the gearbox to obtain suitable angular speeds at the wheels level: $\omega = \omega_m / N$, being N the reduction ratio. Further, in (9),

β_{tot} and J_{tot} are the robot total friction and inertia respectively, R_a and L_a the armature resistance and inductance respectively, k_v and k_t the back electromotive force and torque constants, respectively. The armature input voltage of the motor is V_a.

It is worth noticing that, given the mobile robot experimental test-bench, the input armature voltage V_a and the output reduced angular speed ω are the only variables which can be directly measured. On the other side, all the model parameters are obtained through dedicated identification analyses (cf. Appendix A).

4.2. Mobile robot asynchronous EMC: Internal model

For the sake of simplicity, the following equations will refer to the DC right motor only, being the left one similar with proper model parameters set as per Table A.5.

The internal model of the differential-drive mobile robot is intended to be directly implemented on the remote control board, a Raspberry Pi 3 board, and it is the streamlined discrete-time conversion of the continuous-time fine model detailed in (9), obtained through two design choices. Firstly, since the armature current i_a, and consequently T_m, cannot be directly measured, it is deemed suitable to treat the electrical part of DC motor as a neglected dynamics, for the purpose of the digital control unit design. This implies that, being not explicitly included within the internal model, its effects were treated as unknown disturbances to be estimated by the disturbance predictor and actively rejected. Similarly, being the exact platform values of β_{tot} and J_{tot}, and a potential external resistive torque T_r, difficult to assess, they are as well not explicitly included in the internal model.

These model simplifications do not affect the control performances of the EMC unit. Indeed, as explained in [33], the model uncertainties are directly included in the noise estimator state observer, which can be rejected with an appropriate disturbance rejection control law gain. This is true even with a large difference between the model and the reality, since the motor model has significant unmodelled dynamics related to the friction and the inertia. As a result, even a less accurate model can be exploited maintaining the desired control performances.

Following the assumptions above, to derive the internal model, (9) is first translated in the frequency domain, then written in function of τ_a, τ_m time constants (for their definition refer to (A.1) in Appendix A), and k_v:

$$\frac{\omega(s)}{V_a(s)} = \frac{1/k_v}{1 + s\tau_m + s^2\tau_m\tau_a}. \tag{10}$$

The analysis and simulation of (10) allowed us to make two crucial considerations to derive the internal model of the plant. First of all, the parameters in Table A.5 imply that $\tau_a \approx 0.01\,\text{s}$ is substantially lower compared to $\tau_m \approx 0.06\,\text{s}$. As a result, only τ_m and k_v parameters affect substantially the system behaviour. This result validates the hypotheses of neglecting the electrical dynamics of the electric motor.

Secondly, k_v is directly related to the motor angular speed at the shaft ω_m, which cannot be directly measured in the adopted platform. Hence, to simplify the model derivation, a motor without the gearbox reduction but with an output speed reduced by the factor N is considered. Consequently, the speed at the wheels, $\omega = \omega_m/N$, is assumed as model state, and the relative model parameter becomes $k_v' = k_v N$.

To sum up, the mobile robot DC motor model (9) in time domain shrinks to

$$\begin{aligned}
&\dot{\omega}(t) = -\frac{1}{\tau_m}\omega(t) + \frac{1}{\tau_m k_v'}V_a(t), \qquad \omega(0) = \omega_0,\\
&y_m(t) = \omega(t),\\
&e(t) = k_v'\omega(t).
\end{aligned} \tag{11}$$

Eq. (11) provides the final input–output controllable model, on which the digital controller design is based. Finally, to complete the internal model, the plant input–output description in (11) is paired with the general disturbance estimation dynamics: the basis for the EMC rejector (cf. Fig. 2). To build the EMC rejector, (11) was augmented by a second-order stochastic disturbance dynamics [31,36], to account for model parametric uncertainties, neglected dynamics, and other noise components, i.e.

$$\begin{aligned}
&\begin{cases}
\dot{\omega}(t) = -\frac{1}{\tau_m}\omega(t) + \frac{1}{\tau_m k_v'}V_a(t) + \overline{w}_1(t) + x_{d1}(t),\\
\dot{x}_{d1}(t) = x_{d1}(t) + x_{d2}(t) + \overline{w}_2(t),\\
\dot{x}_{d2}(t) = x_{d2}(t) + \overline{w}_3(t),
\end{cases}\\
&y_m(t) = \omega(t),\\
&\omega(0) = \omega_0, \quad x_{d1}(0) = x_{d10}, \quad x_{d2}(0) = x_{d20}.
\end{aligned} \tag{12}$$

The noise estimates $\overline{w}_j$ are selected to capture the DC motor dynamics, which is usually of second order: this is the reason why the disturbance equations with states x_{d1}, x_{d2} are of second order.

In (12), x_{d1} and x_{d2} represent the two disturbance state variables making up the disturbance predictor. In the noise estimator model selection, static and dynamic feedback disturbance estimators are studied. At the end, a static disturbance model is preferred rather than a dynamic one, since it has lower complexity with comparable estimation performances. Among the static feedback disturbance models, in the experimental results first or second order estimators are investigated.[1]

As shown in Fig. 2, according to the EMC typical architecture, the disturbance dynamics is driven by a noise vector $\overline{\mathbf{w}} = [\overline{w}_1, \overline{w}_2, \overline{w}_3]^T$, whose components are function of the model output error $\mathbf{e}_m = \mathbf{y} - \mathbf{y}_m$ (measurement minus estimate), [31]. As a result, the full internal model of the differential-drive mobile robot, discretized using the Forward-Euler method for the implementation within the control unit, is given by

$$\begin{aligned}
&\begin{cases}
\omega(k+1) = \left(-\frac{1}{\tau_m}T_s + 1\right)\omega(k) + \frac{T_s}{\tau_m k_v'}V_a(k) + d(k),\\
x_{d1}(k+1) = (T_s + 1)x_{d1}(k) + T_s\left[x_{d2}(k) + \overline{w}_2(t)\right],\\
x_{d2}(k+1) = (T_s + 1)x_{d2}(k) + T_s\overline{w}_3(k),
\end{cases}\\
&y_m(k) = \omega(k),\\
&d(k) = T_s\left[\overline{w}_1(k) + x_{d1}(k)\right],\\
&\omega(0) = \omega_0, \quad x_{d1}(0) = x_{d10}, \quad x_{d2}(0) = x_{d20},
\end{aligned} \tag{13}$$

where the sampling time T_s is introduced. The sampling time T_s is variable within a certain range, leading to an asynchronous control system.

It is worth notice that the Forward-Euler method resulted to be sufficient for the discretization of the system (13), because the sampling time T_s is sufficiently small to guarantee low discretization errors.

Consequently, by shaping (13) as per (1), the state matrices of the EMC equations hold

$$\begin{aligned}
&\mathbf{A} = \begin{bmatrix} \mathbf{A}_c & \mathbf{H}_c \\ \mathbf{0}_{n_{x_d}\times n_{x_c}} & \mathbf{A}_d \end{bmatrix} = \left[\begin{array}{c|cc} -\frac{T_s}{\tau_m}+1 & T_s & 0 \\ \hline 0 & T_s + 1 & T_s \\ 0 & 0 & T_s + 1 \end{array}\right],\\
&\mathbf{B} = \begin{bmatrix} \mathbf{B}_c \\ \mathbf{B}_d \end{bmatrix} = \left[\begin{array}{c} \frac{T_s}{\tau_m k_v'} \\ \hline 0 \\ 0 \end{array}\right],\\
&\mathbf{G} = \begin{bmatrix} \mathbf{G}_c \\ \mathbf{G}_d \end{bmatrix} = \left[\begin{array}{ccc} T_s & 0 & 0 \\ \hline 0 & T_s & 0 \\ 0 & 0 & T_s \end{array}\right],\\
&\mathbf{C} = \begin{bmatrix} \mathbf{C}_c & \mathbf{C}_d \end{bmatrix} = \left[\begin{array}{c|cc} 1 & 0 & 0 \end{array}\right],\\
&\mathbf{F} = \begin{bmatrix} \mathbf{F}_c & \mathbf{F}_d \end{bmatrix} = \mathbf{C},\\
&\mathbf{x}_c(k) = \mathbf{y}_m(k) = \omega(k), \quad \mathbf{x}_d(k) = \begin{bmatrix} x_{d1} & x_{d2} \end{bmatrix}^T(k),\\
&\overline{\mathbf{w}}(k) = \begin{bmatrix} \overline{w}_1 & \overline{w}_2 & \overline{w}_3 \end{bmatrix}^T(k),
\end{aligned} \tag{14}$$

[1] From now on, if not explicitly specified, the second order static disturbance model is considered in the experimental tests. However, in some of the tests, cfr. Section 5.5, the first order disturbance model is preferred. In these cases, to build a first order model noise estimator, the equations above are easily modified, as explained in Appendix B.2.

where the subscripts c and d refer to the matrices affecting the controllable and the disturbance states, respectively.

4.3. Mobile robot asynchronous EMC: Noise estimator

The model and the disturbance state predictor, directly coded into the mobile robot control unit, play a key role in the EMC asynchronous design. The discrete-time input–output model, augmented via the disturbance state dynamics and driven by the noise vector $\overline{\mathbf{w}}$, cf. (14), represents the basis to build the state predictor when coupled with a suitable output-to-state feedback suitably designed to close the loop, see Fig. 2. To this aim, a noise estimator block was designed with the two goals of achieving the closed-loop stabilization of the predictor, as well as to ensure a suitable disturbance estimation capability. As a matter of fact, the noise estimator was designed as a properly-dimensioned static feedback (cf. Appendix B)

$$\overline{\mathbf{w}}(k) = \begin{bmatrix}\overline{w}_1\\ \overline{w}_2\\ \overline{w}_3\end{bmatrix}(k) = \mathbf{L}\mathbf{e}_m(k) = \begin{bmatrix} l_1\\ l_2\\ l_3\end{bmatrix}\mathbf{e}_m(k), \qquad (15)$$
$$\mathbf{e}_m(k) = \mathbf{y}(k) - \mathbf{y}_m(k).$$

In (15), $\mathbf{y}_m$ is the estimated output of the internal model, $\mathbf{y}$ is the measured output of the plant, while $\mathbf{L}$ collects the closed-loop predictor gains $[l_1, l_2, l_3]^T$. As a result, the closed-loop predictor model can be determined by designing and tuning the predictor gains trading-off between the closed-loop stability and the estimation performance.

In this study, given the NCS scenario and the asynchronous control unit, the discrete sampling time T_s is variable at every step. As a matter of fact, the noise estimator gains depend on the asynchronous T_s and they are preliminary tuned by fixing the closed-loop complementary eigenvalues of the predictor state matrix (cf. Appendix B), via a pole placement, and then refined in simulations and experimental tests. Finally, being executed with asynchronous sampling times, Table 1 lists the continuous-time noise estimator eigenvalues μ_N, which are then converted to discrete-time domain eigenvalues λ_N, at every control time step. The relation between continuous and discrete sampling time eigenvalues, for the noise estimator block, is the sampling transformation mapping between the $\mathcal{L}$-transform and $\mathcal{Z}$-transform, i.e. $\lambda_N = e^{\mu_N T_s}$.

4.4. Mobile robot asynchronous EMC: Reference dynamics

The reference generator provides the reference trajectory with: (i) the nominal command $\overline{\mathbf{u}}$, and (ii) the reference canonical output and state $\overline{\mathbf{y}} = \overline{\mathbf{x}} = \overline{\omega}$ (where $\overline{\omega}$ is the reference output speed) to compute the tracking error $\overline{\mathbf{e}}$, as depicted in Fig. 2.

In line with the EMC framework, in this study, a model-based design of the reference dynamics is proposed, based on the same input–output controllable dynamics in (11). Such a design choice is aimed at reducing significantly the feedback effort, since the reference state trajectories are naturally compliant with the plant dynamics. Specifically, a static-state feedback was leveraged to obtain a reference trajectory $\overline{\mathbf{r}}(k)$ imposing as nominal control input $\overline{\mathbf{u}}(k) = -\mathbf{K}_R\overline{\mathbf{x}}(k) + \mathbf{N}_R\overline{\mathbf{r}}(k)$, where $\mathbf{K}_R = k_R$ and $\mathbf{N}_R = n_R$ are the reference generator gains, to be designed and tuned using pole-placement techniques. Consequently, the discrete-time reference dynamics state equations are

$$\overline{\mathbf{x}}(k+1) = \mathbf{A}_R\overline{\mathbf{x}}(k) + \mathbf{B}_R\overline{\mathbf{r}}(k),$$
$$\overline{\mathbf{y}}(k) = \mathbf{C}_R\overline{\mathbf{x}}(k), \qquad (16)$$
$$\mathbf{A}_R = \mathbf{A}_c - \mathbf{B}_c\mathbf{K}_R, \quad \mathbf{B}_R = \mathbf{B}_c\mathbf{N}_R, \quad \mathbf{C}_R = \mathbf{C}_c.$$

By applying pole placement to (16), given the asynchronous setup, the closed-loop gain $\mathbf{K}_R$ holds

$$\mathbf{K}_R = \frac{\mathbf{A}_c - a_R}{\mathbf{B}_c}, \qquad (17)$$

where a_R, i.e. the coefficient of the characteristic polynomial defined by a user-defined eigenvalue, changes according to the sampling time T_s. It is worth noticing that, from a frequency perspective, the eigenvalues are placed as close to zero as possible (in continuous-time domain), in order to generate a quite slow reference dynamics. Such a design choice aims at increasing the overall capability of the internal model system to track the reference $\overline{\mathbf{r}}$.

Instead, the gain matrix $\mathbf{N}_R$ is determined by imposing the DC-gain $\mathbf{K}_{DC}$ of the overall system transfer function $\mathbf{W}(s) = \overline{\mathbf{y}}(z)/\overline{\mathbf{r}}(z)$ equal to 1, i.e., by rearranging (16):

$$\mathbf{N}_R = \left[\mathbf{C}_R\left(\mathbb{I} - \mathbf{A}_R\right)^{-1}\mathbf{B}_c\right]^{-1}. \qquad (18)$$

Table 1 reports the continuous-time reference dynamic eigenvalues μ_R. As a matter of fact, since the control time step is asynchronous, the eigenvalues μ_R changes in line with asynchronous sampling times. Hence, they need to be converted into discrete-time domain eigenvalues λ_R. The reference dynamics continuous and discrete sampling time eigenvalues are related by the sampling transformation mapping between the $\mathcal{L}$-transform and $\mathcal{Z}$-transform, i.e. $\lambda_R = e^{\mu_R T_s}$.

As a final note, also an explicit saturation constraint on the nominal control input $\overline{\mathbf{u}}$ is introduced. Such a constraint avoids the commanding of unattainable reference trajectories, imposing the maximum operative voltage range of the DC motor (cf. $V_a \approx \pm 11.5$ V, in Table A.4).

4.5. Mobile robot asynchronous EMC: Control law

The typical control law design leveraged within the EMC control unit is expressed in (3), where the rejector matrices $\mathbf{M}, \mathbf{Q}$ and the feedback gains in $\mathbf{K}_C$ must be designed. In the case-study here presented, $\mathbf{M} \in \mathbb{R}^{1\times 2}$, determined from the Sylvester–Francis equation, [27,37], allows to consider a disturbance dynamics of any order (second, in this case) and holds $\mathbf{M} = \left[\tau_m k_v' \;\; 0\right]$. The matrix $\mathbf{Q}$ results with null components, hence the tracking error in (3) reduces to $\overline{\mathbf{e}} = \overline{\mathbf{x}} - \mathbf{x}_c$.

Conversely, $\mathbf{K}_C$ is designed by means of pole placement, aiming at reaching asymptotically stable closed-loop eigenvalues. To achieve a zero tracking error, a discrete proportional–integral (PI) controller design is selected. To this aim, the $\mathbf{K}_C = [k_p, k_i]$ coefficients are determined by considering the closed-loop system dynamics, including the system controllable dynamics and the PI controller, i.e.,

$$\begin{bmatrix}\mathbf{x}_1\\ \mathbf{x}_2\end{bmatrix}(k+1) = \underbrace{\begin{bmatrix}-\mathbf{A}_c - k_p\mathbf{B}_c & k_i\mathbf{B}_c\\ -1 & 1\end{bmatrix}}_{\mathbf{A}_{ctrl}}\begin{bmatrix}\mathbf{x}_1\\ \mathbf{x}_2\end{bmatrix}(k) + \qquad (19)$$
$$+ \underbrace{\begin{bmatrix}k_p\mathbf{B}_c\\ 1\end{bmatrix}}_{\mathbf{B}_{ctrl}}\overline{\mathbf{e}}(k),$$

where $\mathbf{x}_1 = \mathbf{x}_c$, $\mathbf{x}_2$ is the new state introduced by the integral action of the controller, which leads to the augmented matrices $\mathbf{A}_{ctrl}, \mathbf{B}_{ctrl}$. It is worth notice that $\mathbf{A}_c, \mathbf{B}_c$ depend on the sampling time T_s.

To conclude, by applying pole placement to (19), the closed-loop feedback gains k_p and k_i are computed so to ensure asymptotic stability and hold:

$$k_p = \frac{a_{c1} + \mathbf{A}_c + 1}{\mathbf{B}_c}, \quad k_i = \frac{\mathbf{B}_c k_p - \mathbf{A}_c + a_{c2}}{\mathbf{B}_c}, \qquad (20)$$

where a_{c1} and a_{c2} are the coefficients of the second-order characteristic polynomial defined by the assigned eigenvalues set. As a matter of fact, k_p and k_i, likewise the noise estimator gains, cf. (15), change according to the considered discrete sampling time of T_s (cf. Section 2.2). To sum up, Table 1 reports the continuous-time control block eigenvalues μ_K, to be then converted to discrete-time eigenvalues λ_K. The λ_K eigenvalues change at each control time-step in line with the asynchronous execution and sampling time. The sampling transformation between the $\mathcal{L}$-transform and $\mathcal{Z}$-transform maps the control law continuous and discrete-time eigenvalues, i.e. $\lambda_K = e^{\mu_K T_s}$.

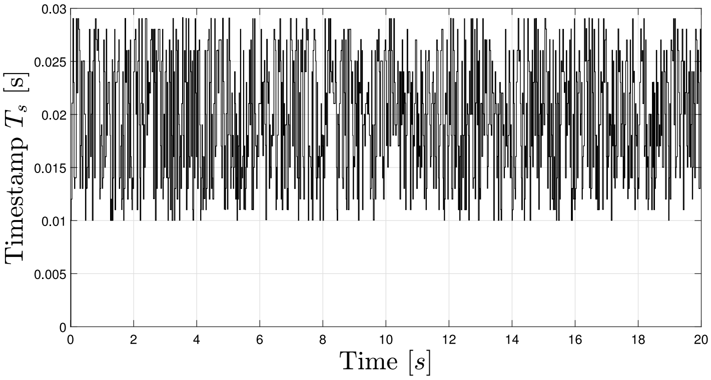


Fig. 3. Asynchronous Timestamp with range 0.01 to 0.03 s used for EMC experimental results.

Table 1
Differential-drive mobile robot control: parameters for selected experimental test.

Parameter	Value
Experimental test time step T_s [s]	[0.01, 0.03]
Timestamp mean [s]	0.0195
NCS random generator seed	Time since 01/01/1970
Reference dynamics eigenvalues μ_R	[−2.5647]
Feedback eigenvalues μ_K	[−2.5647, −2.5647]
Predictor eigenvalues μ_N	[−14.3842, −14.3842, −14.3842]

Finally, the overall command input **u** is endowed with a variable saturation constraint. Such a constraint is shaped so that, when battery-powered, the mobile robot is allowed to leverage at full extent the voltage that the batteries can supply, while continuously discharging.

5. Experimental results

The test-campaign devised for the designed asynchronous control unit focused on two sets of experimental results: (i) a high-fidelity simulations campaign, in Matlab/Simulink, [32], and (ii) experimental trials, in a range of different test manoeuvres and configurations, encompassing the selected robotic platform and a real-time embedded system. After the simulation campaign, the designed discrete-time control unit was made hardware-tests-ready by: (i) translating the Matlab/Simulink code into C++ language, using the Simulink Embedded Coder toolbox, (ii) encapsulating the control unit code within the overall software to control the motors, including suitable I/O and communications interfaces (I/F), (iii) testing the I/F and the functionalities of the embedded control unit, via a set of interface and functional tests. More specifically, the overall on-board software of the robotic platform was embedded on a Raspberry Pi board, adopting a Raspbian operating system.

For the sake of brevity, this section reviews the experimental trials of the designed EMC asynchronous control unit, applied to a differential-drive mobile robot in an NCS scenario. In particular, after the experimental setup description in Section 5.1, in Section 5.2 the EMC disturbance rejection capabilities are proven with a specific test. Furthermore, in Section 5.3 the EMC is studied in the two experimental cases of disturbance rejection presence and absence. Finally, Section 5.5 describes the results of a comparison between EMC and PID controllers in the NCS scenario.

5.1. Robotic platform tests in an NCS scenario — Setup

For conciseness, the presented results will only focus on the right DC motor of the mobile robotic platform (in loaded condition, when robot moves on the ground), since the left one shows a very similar behaviour.

After an extensive simulation campaign, the EMC asynchronous control unit was tested with real hardware, a differential-drive mobile robotic platform. The tests were conducted in a representative networked controlled system (NCS) environment characterized by variable sampling times (cf. Section 2.2). The resulting parameters of the asynchronous control unit are listed in Table 1, as well as the continuous-time eigenvalues μ_R, μ_K, μ_N respectively for the reference dynamics, control law and state predictor blocks. The continuous-time eigenvalues shown in Table Table 1 are obtained by trial and error procedure.

The closed-loop stability of the system in case of asynchronous sampling time, for T_s varying from 0.01 s to 0.03 s, is studied in Fig. 15. The stability analysis is analogous for different range sampling times. In this figure the discrete-time domain eigenvalues are presented in the polar plot, for reference dynamics $\lambda_R = e^{\mu_R T_s}$, control law $\lambda_K = e^{\mu_K T_s}$ and predictor $\lambda_N = e^{\mu_N T_s}$. It can be seen that, even with asynchronous sampling time, the values are always inside the unitary circle, thus guaranteeing asymptotic stability. In addition, a theoretical analysis of the closed-loop stability is detailed in [33].

The continuous variation of the sampling times is one of the distinctive features of the experimental campaign implemented for this study. The sampling time varies in a desired interval, which was designed in order to have the fastest possible control, in line with the practical limits of the hardware timers running on the adopted Raspberry Pi control board. Unless specified, the asynchronous time range for the tests was defined from 0.01 to 0.03 s (cf. Table 1 and Fig. 3), where the lowest bound is near to the operational limit of the hardware timers, while the maximum value lets the system be effectively affected by the disturbances due to the asynchronous sampling times.

5.2. Test 1 — EMC disturbance rejection effectiveness in an NCS scenario

Figures from 6 to 8 depict a test aiming at practically showing the effectiveness of the designed EMC asynchronous control unit in estimating and rejecting disturbances and uncertainties, potentially affecting the plant dynamics. As it can be seen in Fig. 6, a known external disturbance step (−1 rad/s, corresponding to the robot climbing a ramp with a slope of ≈ 17 %) was induced to the output speed, starting at 5 s: in this way the output is in a known steady-state value ($\mathbf{y}_{ss}$ = 6 rad/s), while the effect of the induced disturbance step can be clearly identified. In the figure the measured output **y**, the estimated output $\mathbf{y}_m$ and the reference output $\bar{\mathbf{y}}$ are compared. As a result, the effect of the disturbance can be clearly spotted on **y** and $\mathbf{y}_m$, in the time period from 5 to 6 s, showing the capability of the designed state predictor to properly estimate the external disturbances potentially affecting the plant dynamics.

Remark: The step input disturbance of this test was only considered to understand the abilities to reject the unknown disturbances by the EMC, every noise signal can be considered and, for simplicity, a

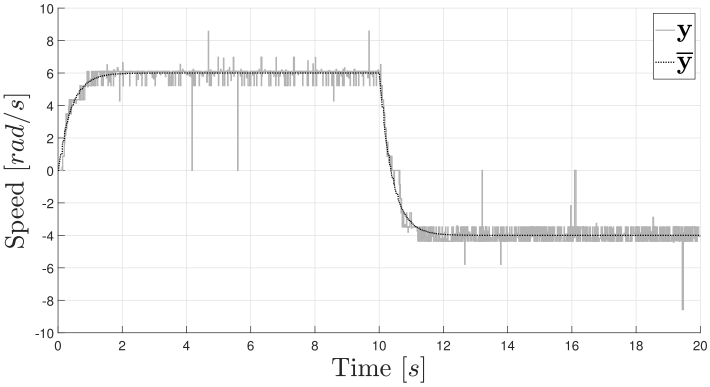


Fig. 4. Test with delays and package dropouts — Measured output $\mathbf{y}$ and reference $\overline{\mathbf{y}}$ speed.

step signal was considered. In the next tests presented in this study, the disturbances were more complex, comprising not only the DC motor unmodelled dynamics (matched and mismatched), but also the disturbance coming from the asynchronous timing control of the plant.

In Fig. 7 (top), the control input $-\mathbf{u}_d = -\mathbf{M}\mathbf{x}_d$ (cf. Eq. (3)) resulting from the robot climbing the ramp is shown. Interestingly, after the initial transient peak at the disturbance onset, starting at $5\,\mathrm{s}$ the command value assumes an increased value of about $1\,\mathrm{rad\,s^{-1}}$, with respect to the previous steady-state condition. As a matter of fact, such a value corresponds to the required voltage input to obtain the requested reference speed, with a unitary amplification factor (transducer). Indeed, the $\mathbf{u}_d$ value cancels out the effect of the step disturbance on the output $\mathbf{y}$. Finally, the peak at the onset of the disturbance action (cf. Fig. 7, from 5 to $\approx 5.5\,\mathrm{s}$) depicts the transition phase, where the EMC asynchronous disturbance estimator is estimating the dynamics of the plant system, and is strictly related to the predictor tuning. Fig. 7 (bottom) also depicts the feedback activity (or tracking error), $\mathbf{u}_{trk}$. In this plot we observe how, after the initial transient, $\mathbf{u}_{trk}$ has an almost zero value (zero-mean). This implies that the disturbance estimator, once initialized, allows the feedback portion of the controller to discharge, [33]. These results highlight as well the disturbance rejector block capability to drive the robotic platform with a low feedback effort, while ensuring a suitable performance: a very distinctive feature of a disturbance-rejection-based control law, here proven to be valid also within an NCS scenario, where the asynchronous timing causes the variation of the output speed.

Furthermore, Fig. 8 depicts the model output error $\mathbf{e}_m = \mathbf{y} - \mathbf{y}_m$ (measurement minus estimate), characterizing the disturbance rejection experimental test. The model output error, a key signal in the EMC design, is shown to be in the range of the maximum encoder angular speed resolution error, $e_{enc} = r_e/T_{\mathrm{s,min}} = 0.8726\ \mathrm{rad/s}$. In particular, $r_e = 360/\mathrm{PPR} = 0.5\ \mathrm{deg}$ is the minimum encoder resolution, $PPR = 720$ the Pulses per Revolution taken from Table A.4, and $T_{\mathrm{s,min}} = 0.01\,\mathrm{s}$ is the sampling time lower bound. This implies that, although including a streamlined model of the plant to be controlled (i.e., the internal model, cf. Section 4.2), the EMC predictor satisfactorily recovers the differences in terms of neglected dynamics and disturbances with respect to the complete DC motor plant (i.e., the fine plant model, cf. Section 4.1). Moreover, this also confirms the viability, also in a NCS experimental scenario affected by continuously variable sampling times, of the static design and gains tuning of the noise estimator (cf. Section 4.3).

5.3. Test 2 — EMC controller in case of disturbance rejection term absence, NCS scenario

From Figs. 9 to 11, it is presented a $20\,\mathrm{s}$ test of the mobile robotic platform, with and without considering the active disturbance rejection part of the control law, $\mathbf{u}_d$ (cf. Section 4.5). Specifically, Fig. 9 shows the DC motor output speed trajectories, with an indication of the measured output $\mathbf{y}$, in comparison with the reference one $\overline{\mathbf{y}}$.

In the case of no disturbance rejection, cf. Fig. 9(a), the disturbance rejection term is $\mathbf{u}_d = 0\,\mathrm{V}$, and only the tracking $\mathbf{u}_{trk}$ and reference $\overline{\mathbf{u}}$ portions of the control law are considered. High peaks can be seen in the transient phases, before reaching the steady-state reference values. Instead, in the disturbance rejection case, cf. Fig. 9(b), the real-time estimation provided by the EMC predictor suitably tracks the measured signal. This implies that the designed model-based asynchronous state predictor, embedded in the control unit, effectively describes the plant dynamics up to a meaningful frequency bandwidth, although the constraints and the harsh environmental conditions imposed by the NCS scenario. It is worthwhile to notice how the disturbance-based command departs from the saturation condition earlier. In turn, this implies faster response and convergence times, thus a lower tracking error in case of tricky manoeuvres, along the manoeuvre time-interval. Such a result is confirmed by the comparison of Fig. 10(a) and 10(b), where the tracking error $\overline{\mathbf{e}} = \overline{\mathbf{y}} - \mathbf{y}_m = \overline{\mathbf{x}} - \mathbf{x}_c$ is depicted in the two considered cases for the control law (without and with the disturbance rejection). It can be noticed how the disturbance-rejection-based control law behaves substantially better in terms of tracking performance, during the action of the unexpected disturbance as well as in the transient phase.

From these experimental results, it can be thus inferred the compelling advantage of the proposed disturbance-rejection-based approach, in a scenario characterized by high uncertainty and variability, like NCS systems.

A further result of the designed EMC asynchronous control unit concerns the feedback activity. Fig. 11(a) and 11(b) describe the feedback activity, during the experimental trial, for the two control laws, without and with the disturbance rejection, respectively. Fig. 11(b) highlights an unbiased and very faint feedback command and shows that, when the disturbance rejection term is available, the activity of the feedback is significantly reduced. Specifically, in Fig. 11(b), maximum peaks do not exceed $|1.5|\mathrm{V}$ as absolute value, in contrast with much higher values, nearly $|4|\mathrm{V}$ as absolute value, highlighted in Fig. 11(a). This is a typical and remarkable characteristic of an EMC control law, due to the active disturbance rejection enabled by the disturbance estimation dynamics (cf. Fig. 7). Indeed, when an active disturbance rejection is available, the feedback control portion $\mathbf{u}_{trk}$ only needs to manage the residual error after the intervention of $\mathbf{u}_d$. This minimizes the feedback activity, thus improving the overall control robustness, [33].

5.4. Test 3 — EMC controller with time delay and package dropouts

In Figs. 4 and 5 a test is shown, where the timestamp was not only affected by sampling time delays in the range $[1-50]\,\mathrm{ms}$, but also with package dropouts. The package dropouts are obtained by randomly forcing the timestamp to be very low, near $0\,\mathrm{s}$. The variable timestamp is presented in Fig. 5. Instead, in Fig. 4 the measured output speed $\mathbf{y}$

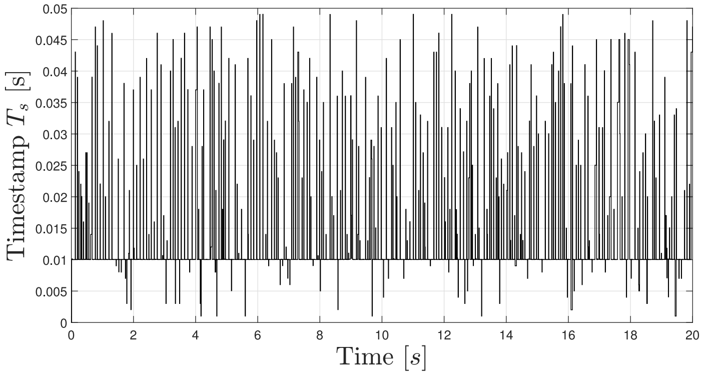


Fig. 5. Test with delays and package dropouts — Variable timestamp.

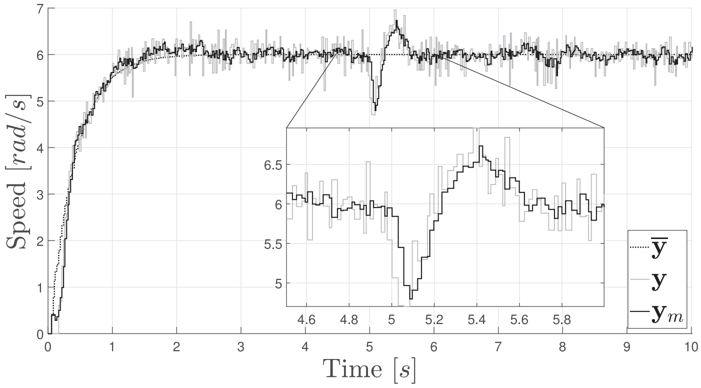


Fig. 6. Step external disturbance of $-1\,\mathrm{rad\,s^{-1}}$ applied on $\mathbf{y}$ — Measured output $\mathbf{y}$, reference $\overline{\mathbf{y}}$, and estimated $\mathbf{y}_m$ speed values.

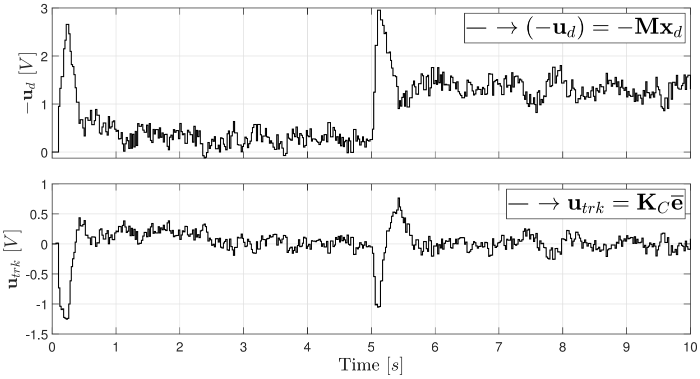


Fig. 7. Step external disturbance of $-1\,\mathrm{rad\,s^{-1}}$ applied on $\mathbf{y}$ — Negative control input $-\mathbf{u}_d$ (top) and feedback activity $\mathbf{u}_{trk}$ (bottom).

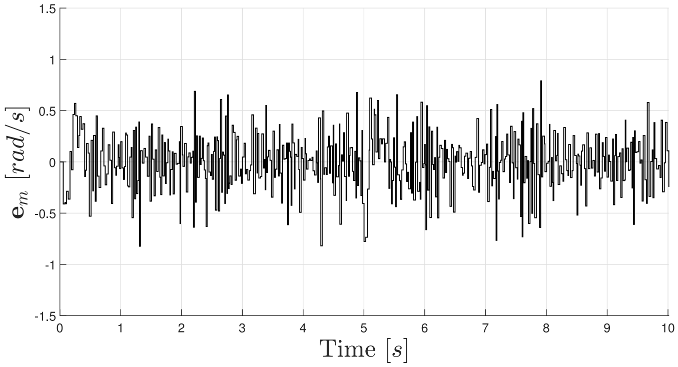


Fig. 8. Step external disturbance of $-1\,\mathrm{rad\,s^{-1}}$ applied on $\mathbf{y}$ — Model output error $\mathbf{e}_m = \mathbf{y} - \mathbf{y}_m$ for DC motor speed.

and the reference speed $\overline{\mathbf{y}}$ to be tracked are compared. Apart from a low number of spikes in the measurements, $\mathbf{y}$ is able to track with a low tracking error the reference $\overline{\mathbf{y}}$, thus proving the EMC effectiveness even in presence of package dropouts.

5.5. Test 4 — EMC controller compared to PID and LQR controllers in an NCS scenario

Finally, a comparison between an EMC, PID and LQR controllers is presented, with focus on the output motor speed. To understand the limits of the two controllers, a critical asynchronous scenario is considered, with sampling time T_s spanning in the range $[1-70]\,\mathrm{ms}$, for the control models. The timestamp range and the controller parameters are summarized in Table 2.

In this test case study, the EMC control unit parameters were obtained with a tuning procedure taking into account the simplified model dynamics of the plant. The noise estimator is selected of the first order, thus (15) is easily modified as in Appendix B.2. In this case, the continuous-time control law μ_K and the noise estimator μ_N eigenvalues are modified to fit the reduced order noise estimator, and their values are reported in Table 2.

In addition, the noise estimator was selected of the first-order since, with high sampling time ranges, the tracking error capabilities were better than the second order disturbance model, cfr. Appendix B.2. Furthermore, the tuned PID parameters were computed through the closed-loop Ziegler–Nichols method. Finally, the LQR parameters are obtained by minimizing the infinite horizon cost function

$$\begin{aligned} J(\mathbf{x}(k),\mathbf{u}(k)) &= \sum_{k=0}^{\infty} \mathbf{x}^T(k+1)\mathbf{Q}_{LQR}\mathbf{x}(k+1) + \mathbf{u}^T(k+1)\mathbf{R}_{LQR}\mathbf{u}(k+1) \\ \mathbf{u}^{\star}(k) &= \arg \min_{u(k),k\in[0,\infty)} J(\mathbf{x}(k),\mathbf{u}(k)) = -K_{LQR}\mathbf{x}(k) \end{aligned} \tag{21}$$

The matrices $\mathbf{Q}_{LQR} \geq 0$ and $\mathbf{R}_{LQR} > 0$ are the design parameters chosen according to the desired performance tradeoff. The optimal input value $\mathbf{u}^{\star}(k)$ of (21) can be found by solving the discrete-time Algebraic Riccati Equation. An additional state is added, the integral of the output speed error, for the LQR integral action. The final gain vector K_{LQR} is found with a static-state feedback control law, and its value is resumed in Table 2.

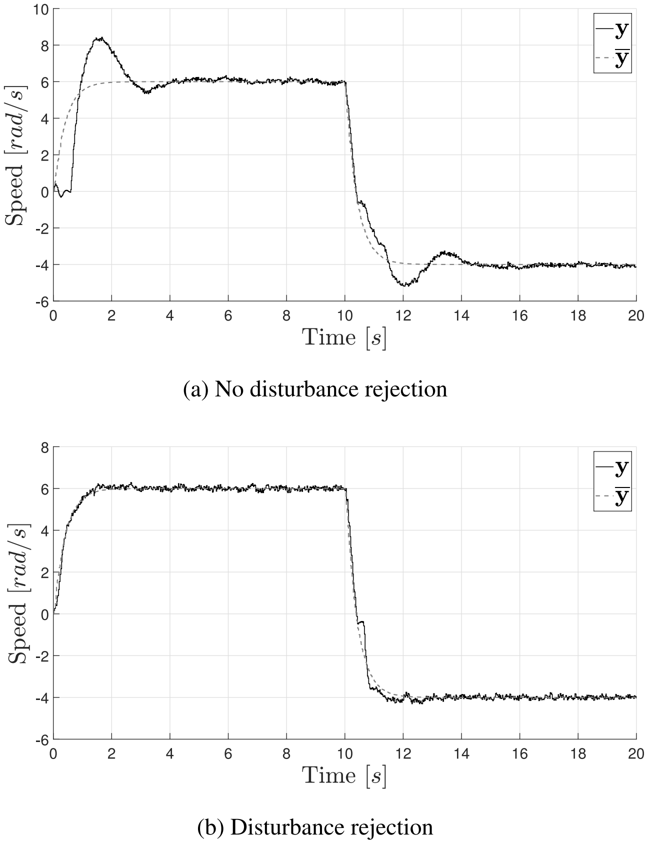
In the comparison test of Fig. 12, PID control is characterized by oscillations of $\mathbf{y}$ around the reference $\overline{\mathbf{y}}$. Such a behaviour can be linked to the presence of such a high variation of the time-step. The LQR shows better output control performances than the PID controller, however even in this case small oscillations appears on $\mathbf{y}$ output compared to the reference $\overline{\mathbf{y}}$. Conversely, the EMC measured output $\mathbf{y}$ appears not to be affected by such oscillations, although facing the same experimental dynamics of the PID and LQR control units. In fact, the EMC control unit benefits from the presence of the disturbance-rejection-based controller, that is able to counteract the disturbances introduced by the asynchronous sampling time shaping the experimental scenario. Such a compensation allows the feedback $\mathbf{u}_{trk}$ to be substantially reduced in magnitude, thus allowing for an enhanced tracking performance, cf. Fig. 13.

To conclude, Fig. 14 compares the tracking errors achieved experimentally by the EMC and PID control units. To understand the controllers tracking capabilities, the Root Mean Square Error (RMSE) parameter is computed.[2] For the EMC this value is $\text{RMSE}_{EMC} = 0.1477\,\text{rad}\,\text{s}^{-1}$, instead for the PID it results to be $\text{RMSE}_{PID} = 0.9185\,\text{rad}\,\text{s}^{-1}$, and for the LQR is $\text{RMSE}_{LQR} = 0.3223\,\text{rad}\,\text{s}^{-1}$: hence the EMC has significantly higher tracking performances than PID and LQR.

To better understand the EMC disturbance rejection capabilities in NCS scenarios, in Table 3 the tracking error RMSE for the three controllers are reported. Several experiments were accounted with different sampling time ranges, from $[1-30]\,\text{ms}$ to a critical condition of $[1-100]\,\text{ms}$. The results show that for all asynchronous time ranges, the RMSE_{EMC} is lower that RMSE_{PID} and RMSE_{LQR}, thus confirming the disturbance rejection EMC performances with asynchronous T_s.

6. Conclusion

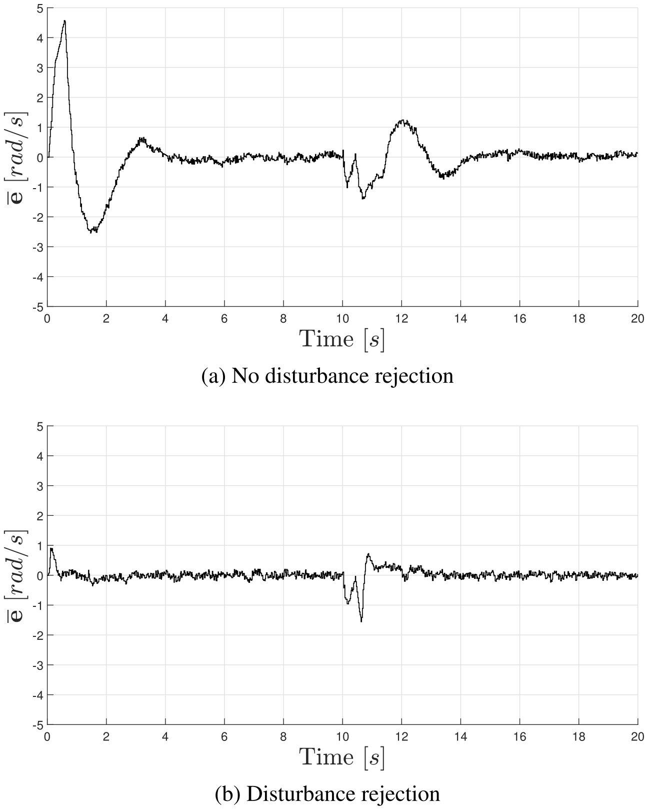
In this study, the Embedded Model Control (EMC) methodology is applied to design a complete digital control unit for a differential-drive mobile robot controlled in a network system (NCS), thus operating in a scenario characterized by a varying sampling time and asynchronous command execution. EMC is a model-based control technique leveraging an internal model of the plant, consisting of a simplified



(a) No disturbance rejection

(b) Disturbance rejection

Fig. 9. DC motor measured $\mathbf{y}$ and reference $\overline{\mathbf{y}}$ output speed — EMC comparison with and without disturbance rejection.



(a) No disturbance rejection

(b) Disturbance rejection

Fig. 10. DC motor output speed tracking error $\overline{\mathbf{e}} = \overline{\mathbf{y}} - \mathbf{y}_m = \overline{\mathbf{x}} - \mathbf{x}_c$ - EMC comparison with and without disturbance rejection.

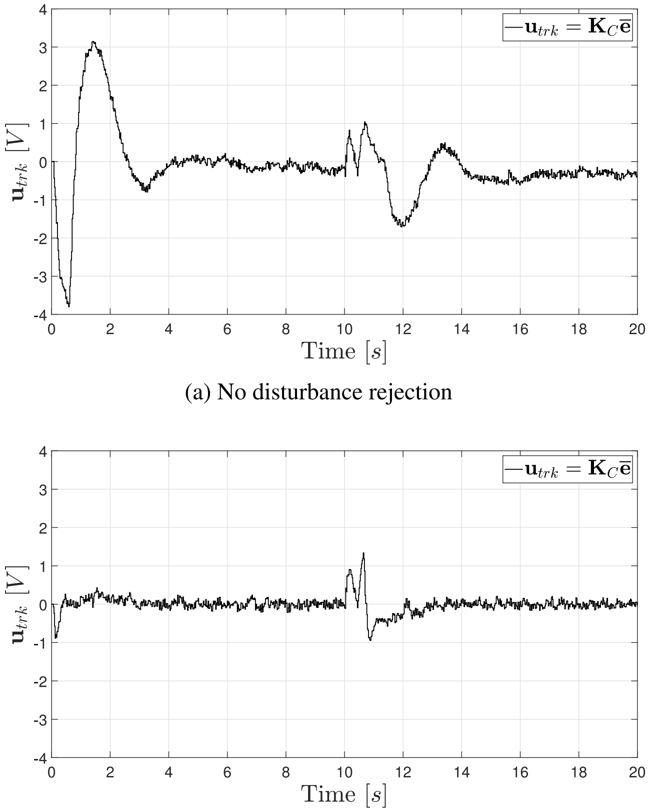
[2] RMSE $= \sqrt{1/(N-N_0)\sum_{t=N_0+1}^{N}\mathbf{y}(t)^2}$ where N is the total number of samples, N_0 is the number of samples accounting for incorrect initial measurements, which can be discarded.

Table 2
EMC, PID and LQR controller experimental test parameters.

Parameter	Value
Experimental test time step T_s [s]	[0.01, 0.07]
EMC ref. dynamics eigenvalues μ_R	[−2.5647]
EMC feedback eigenvalues μ_K	[−11.1572 − 11.1572]
EMC predictor eigenvalues μ_N	[−11.1572 − 11.1572]
PID proportional gain K_P	1.8
PID integral gain K_I	24
PID derivative gain K_D	0.0338
PID derivative closed pole gain N_{PID}	10
LQR gains K_{LQR}	[−1.3788, −0.4988]

Table 3
EMC, LQR and PID tracking error RMSE.

		Sampling time range [ms]			
		[1 – 30]	[1 – 50]	[1 – 70]	[1 – 100]
Controller	EMC	0.0769	0.1161	0.1477	0.2009
	LQR	0.2590	0.2308	0.3223	0.3887
	PID	0.2476	0.3092	0.9185	3.8655



(a) No disturbance rejection

(b) Disturbance rejection

Fig. 11. DC motor control law feedback activity $\mathbf{u}_{trk}$ - EMC comparison with and without disturbance rejection.

controllable model plus a disturbance rejector. The asynchronous EMC rejector allows us to extend the validity of the simplified controllable model, in case of discrepancies between the model and the real plant. As a result, a straightforward model-based control law is pursued in this study, although measurements and control inputs are executed in an NCS environment, characterized by a varying sampling time and asynchronous command execution. Finally, a versatile NCS experimental setup is implemented to validate the asynchronous control unit design through a wide range of experimental tests, including a benchmark study with alternative control schemes. The obtained results indicate that the designed asynchronous disturbance-rejection-based control unit performs properly, even in presence of disturbances due to NCS environments characterized by highly variable sampling times. Moreover, the experimental results highlight the practical advantages of the proposed asynchronous EMC architecture in reducing, to a great extent, the complexity of the NCS control problem, thus enhancing the applicability of the design solution and the potential impacts on engineering practice.

The EMC approach with fixed sampling time was already studied and successfully applied to several applications such as aerial systems (quadrotors, satellites) and industrial systems (electrovalves). As extension, after the foundation work laid out in this paper, the next step of this study will apply the asynchronous EMC approach to the aforementioned applications, by adding a communication network in the control loop design.

CRediT authorship contribution statement

Luca Nanu: Data curation, Software, Supervision, Validation, Visualization, Writing – original draft, Writing – review & editing. **Luigi Colangelo:** Investigation, Supervision, Validation, Visualization, Writing – original draft, Writing – review & editing. **Carlo Novara:** Conceptualization, Data curation, Formal analysis, Investigation, Methodology, Supervision, Writing – original draft, Writing – review & editing. **Carlos Perez Montenegro:** Conceptualization, Methodology, Writing – original draft, Writing – review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

No data was used for the research described in the article.

Research funding

This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

Appendix A. Experimental test-bench setup parameters

The platform selected to test the disturbance-rejection-based asynchronous controller presented in this study is a differential-drive mobile robot, in which a real-time embedded system was incorporated, cf. Fig. 1. Table A.4 lists the main features of the selected robotic platform, while Table A.5 shows the identified parameters of the DC motor model, which were included in the motor internal model (cf. (9)). The motor parameters were identified via the least-squares estimation method, whose initial estimate was then refined with the Simulink Optimization Toolbox from MathWorks, through a nonlinear least-squares method.

Among the various platform parameters, the mechanical τ_m and armature (electrical) τ_a time constants are detailed below. These values, providing the time characteristics of the plant output transfer functions, hold

$$\tau_m = \frac{J_{tot} R_a}{k_v k_t}, \quad \tau_a = \frac{L_a}{R_a}. \tag{A.1}$$

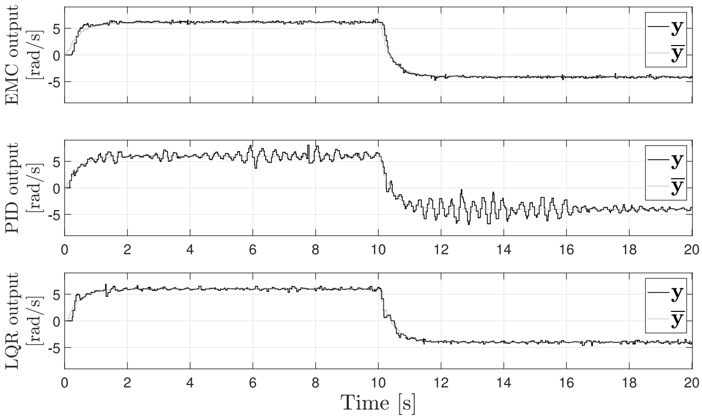


Fig. 12. EMC, PID and LQR output speed comparison — Experimental test with timestamp $T_s = [10 - 70]$ ms.

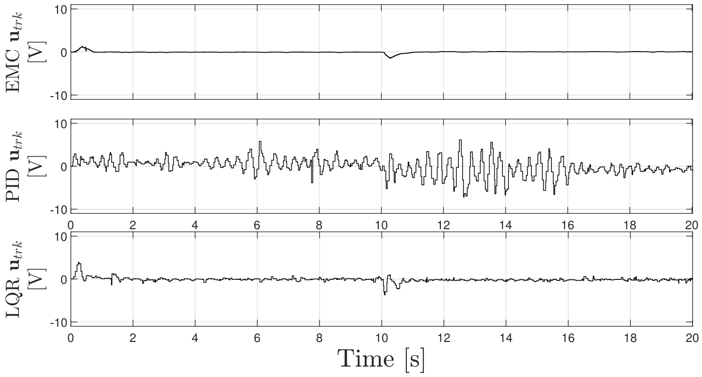


Fig. 13. EMC, PID and LQR comparison: tracking control input $\mathbf{u}_{trk}$ — Experimental test with timestamp $T_s = [10 - 70]$ ms.

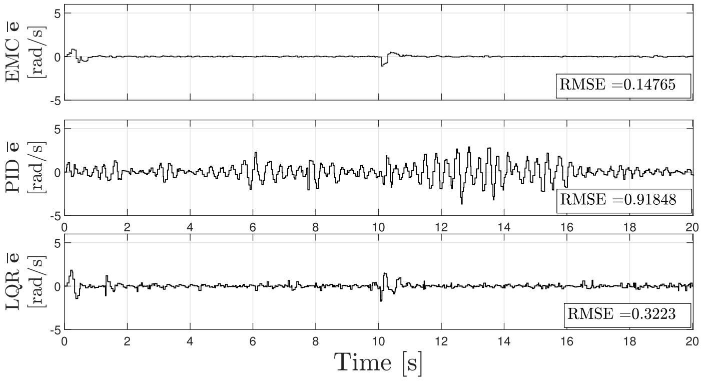


Fig. 14. EMC, PID and LQR comparison: tracking error $\overline{\mathbf{e}}$ with RMSE — Experimental test with timestamp $T_s = [10 - 70]$ ms.

Table A.4
Differential-drive mobile robot parameters.

Robot parameter	
Wheel inter-axis W [m]	0.0585
Robot length l [m]	0.2200
Wheel radius ρ [m]	0.0325
Nominal weight [kg]	0.6160
Gearbox speed reduction N [−]	120
Encoder Pulses per Revolution PPR [−]	720
PWM period DC motors [s]	0.02
Power supply range [V]	8 – 11.5 ca.
DC motor operative voltage range V_a [V]	≈ ±11.5

Table A.5
Differential-drive mobile robot model: estimated model parameters.

Parameter	Left motor	Right motor
β_{tot} [Nm(rad/s)$^{-1}$]	2.1032×10^{-20}	4.8623×10^{-14}
J_{tot} [kgm^2]	3.6898×10^{-7}	4.2876×10^{-7}
L_a [H]	0.24336	0.24735
R_a [Ω]	19.154	17.781
k_v [V s]	0.010179	0.011553
k_t [NmA^{-1}]	0.010179	0.011553
τ_m [s]	0.0682	0.0571
τ_a [s]	0.0127	0.0139

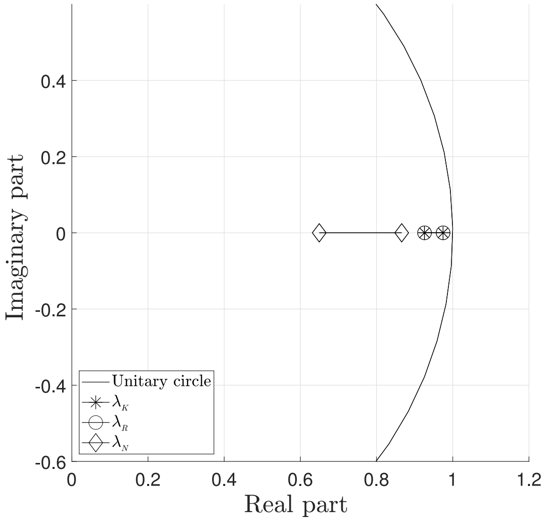


Fig. 15. Polar plot of the stability conditions of the control system, for T_s = [0.01, 0.03] *s*.

Appendix B. EMC noise estimator: Static design and gains tuning

B.1. Second order model noise estimator

The characteristics of the mobile robot model envisaged in this study, described by (1), when parametrized according to (14), allow a static feedback noise estimator to be sufficient to retrieve the noise $\overline{\mathbf{w}}$ in (15), while ensuring closed-loop stability of the state predictor. Indeed, the dimension of the state vector, characterizing the system in this study, is $\dim \mathbf{x} = n_x = 3$, while the one of the disturbance vector is $\dim \overline{\mathbf{w}} = n_w = 3$. Therefore, being $n_x = n_w$, a static gain matrix $\mathbf{L}$ can be deemed as sufficient, as proven in [27].

The suitability of the adopted noise estimator architecture can be further proven by pursuing an output feedback pole placement procedure. Specifically, a necessary (but not sufficient) condition for a static feedback structure, in a stable closed-loop predictor, is $n_w \times n_y \geq n_x$, [35]. Being in this case $n_y = \dim \mathbf{y} = 1$ the dimension of the known output (i.e., the DC motor output speed), the former relation results to be satisfied.

As a result, given the static noise estimator structure in (15), from (1), parametrized as per (14), $\overline{\mathbf{w}}$ was derived by determining the components l_i of $\mathbf{L}$. Those are in turn obtained by equalizing: (i) the coefficients $a_{cl,i}$, $i = 1, \ldots, 3$, of the characteristic polynomial of the closed loop matrix $\mathbf{A}_{CL} = \mathbf{A} - \mathbf{GLC}$, computed with $\det[\mathbf{A}_{CL} - \lambda_N \mathbb{I}]$, and (ii) the coefficients $a_{n,i}$, $i = 1, \ldots, 3$, of a characteristic polynomial with discrete-time eigenvalues $[p_{n1}, p_{n2}, p_{n3}]$, arbitrarily designed so to guarantee asymptotic internal stability ($Re(\lambda_N) < 1$). Hence, if the characteristic and desired polynomials respectively hold:

$$\begin{aligned}&\lambda_N^3 + a_{cl,1}\lambda_N^2 + a_{cl,2}\lambda_N + a_{cl,3} =\\&= (\lambda_N - p_{n1})(\lambda_N - p_{n2})(\lambda_N - p_{n3}) =\\&= \det\left[\mathbf{A_{CL}} - \lambda_N \mathbb{I}\right].\end{aligned} \tag{B.1}$$

The coefficients l_i of matrix $\mathbf{L}$, with $i = 1, \ldots, 3$, depend on the discrete sampling time T_s, which can be considered as variable at every step. As a result, the DC motor noise estimator gain parameters hold

$$\begin{aligned}l_1 &= \frac{\left(2T_s + a_{n,1} + 3\right)\tau_m - T_s}{\tau_m T_s},\\l_2 &= \frac{2(T_s+1)a_{n,1} + a_{n,2} + 6T_s + 3T_s^2 + 3}{T_s^2},\\l_3 &= \frac{\left[1 + 2T_s + T_s^2\right] a_{n,1}}{T_s^3} +\\&+ \frac{\left(1 + T_s\right) + a_{n,2} + a_{n,3} + 3T_s + 3T_s^2 + T_s^3 + 1}{T_s^3}\end{aligned} \tag{B.2}$$

As a matter of fact, the noise estimator block has, at every time-step, the information about the asynchronous timing acting on the plant system, thanks to the dependence of the l_i coefficients from T_s. This, in turn, allows the noise estimator to inject this information into the estimated noise signal $\overline{\mathbf{w}}$.

B.2. First order model noise estimator

The noise estimator can be simplified from a second order to a first order model, if in some experimental tests the tracking and model output error performances start to degrade, when high sampling time ranges are considered.

The EMC internal model in (13) can be simplified as

$$\begin{aligned}&\begin{cases}\omega(k+1) = \left(-\frac{1}{\tau_m}T_s + 1\right)\omega(k) + \frac{T_s}{\tau_m k_v'}V_a(k) + d(k),\\x_d(k+1) = (T_s+1)x_d(k) + T_s\overline{w}_2(t)\end{cases}\\&y_m(k) = \omega(k),\\&d(k) = T_s\left[\overline{w}_1(k) + x_d(k)\right],\\&\omega(0) = \omega_0, \quad x_d(0) = x_{d0},\end{aligned} \tag{B.3}$$

The noise estimator equation in (15) can be reduced as

$$\overline{\mathbf{w}}(k) = \begin{bmatrix}\overline{w}_1\\\overline{w}_2\end{bmatrix}(k) = \mathbf{L}\mathbf{e}_m(k) = \begin{bmatrix}l_1\\l_2\end{bmatrix}\mathbf{e}_m(k), \tag{B.4}$$

and the coefficients l_i of matrix $\mathbf{L}$, for $i = 1, 2$ can be then computed, solving an eigenvalue placement problem and equalizing: (i) the coefficients $a_{cl,i}$ of the characteristic polynomial $\lambda_N^2 + a_{cl,1}\lambda_N + a_{cl,2} = \det[\mathbf{A}_{CL} - \lambda_N\mathbb{I}]$ of the controlled closed loop system; (ii) the coefficients $a_{n,i}$ of a desired characteristic polynomial $(\lambda_N - p_{n1})(\lambda_N - p_{n2}) = \lambda_N^2 + a_{n,1}\lambda_N + a_{n,2}$

$$\begin{aligned}l_1 &= \frac{\left(2 + T_s + a_{n,2}\right)\tau_m - T_s}{T_s\tau_m}\\l_2 &= \frac{2T_s + T_s^2 + \left(1 + T_s\right)a_{n,2} + a_{n,3} + 1}{T_s^2}\end{aligned} \tag{B.5}$$

References

[1] Hespanha JP, Naghshtabrizi P, Xu Y. A survey of recent results in networked control systems. Proc IEEE 2007;95(1):138–62. http://dx.doi.org/10.1109/JPROC.2006.887288, URL http://ieeexplore.ieee.org/document/4118465/.

[2] Ge X, Yang F, Han Q-L. Distributed networked control systems: A brief overview. Inform Sci 2017;380:117–31. http://dx.doi.org/10.1016/j.ins.2015.07.047, URL https://www.sciencedirect.com/science/article/pii/S0020025515005551.

[3] Zhang X-M, Han Q-L, Yu X. Survey on recent advances in networked control systems. IEEE Trans Ind Inf 2016;12(5):1740–52. http://dx.doi.org/10.1109/TII.2015.2506545, URL http://ieeexplore.ieee.org/document/7349192/.

[4] Zhang X-M, Han Q-L, Ge X, Ding D, Ding L, Yue D, et al. Networked control systems: A survey of trends and techniques. IEEE/CAA J Autom Sin 2020;7(1):1–17. http://dx.doi.org/10.1109/JAS.2019.1911651, URL https://ieeexplore.ieee.org/abstract/document/8766208.

[5] Zhen S, Hou Z, Yin C. A novel data-driven predictive control for networked control systems with random packet dropouts. In: 2017 6th data driven control and learning systems. DDCLS, IEEE; 2017, p. 335–40. http://dx.doi.org/10.1109/DDCLS.2017.8068093, URL http://ieeexplore.ieee.org/document/8068093/.

[6] Xie X, Ren Z. Improved delay-dependent stability analysis for neural networks with time-varying delays. ISA Trans 2014;53(4):1000–5. http://dx.doi.org/10.1016/j.isatra.2014.05.010, URL http://linkinghub.elsevier.com/retrieve/pii/S0019057814000998.

[7] Ali SA, Langlois N. Sampled data's observer design with time varying correction gain for electro hydraulic actuator systems. In: 2017 American control conference. ACC, IEEE; 2017, p. 3276–81. http://dx.doi.org/10.23919/ACC.2017.7963452, URL http://ieeexplore.ieee.org/document/7963452/.

[8] Sferlazza A, Tarbouriech S, Zaccarian L. Time-varying sampled-data observer with asynchronous measurements. IEEE Trans Automat Control 2019;64(2):869–76. http://dx.doi.org/10.1109/TAC.2018.2839974.

[9] Chang Y-C, Chen C-W, Tsao T-C. Real-time sub-count estimation with state continuity for asynchronous and quantized sensing. IEEE/ASME Trans Mechatronics 2016;21(3):1457–66. http://dx.doi.org/10.1109/TMECH.2015.2506647, URL http://ieeexplore.ieee.org/document/7349227/.

[10] Qiu J, Gao H, Ding SX. Recent advances on fuzzy-model-based nonlinear networked control systems: A survey. IEEE Trans Ind Electron 2016;63(2):1207–17. http://dx.doi.org/10.1109/TIE.2015.2504351.

[11] Hu J, Zhang H, Liu H, Yu X. A survey on sliding mode control for networked control systems. Internat J Systems Sci 2021;52(6):1129–47. http://dx.doi.org/10.1080/00207721.2021.1885082.

[12] Yuqing L, Huajing F. Control methodologies of large delays in networked control systems. In: International conference on control and automation. 2005, p. 1225–30. http://dx.doi.org/10.1109/ICCA.2005.1528308, URL https://ieeexplore.ieee.org/abstract/document/1528308.

[13] Mahmoud MS, Hamdan MM. Fundamental issues in networked control systems. IEEE/CAA J Autom Sin 2018;5(5):902–22. http://dx.doi.org/10.1109/JAS.2018.7511162, URL https://ieeexplore.ieee.org/abstract/document/8405348.

[14] Gautam M, Pati A, Mishra S, Appasani B, Kabalcı E, Bizon N, et al. A comprehensive review of the evolution of networked control system technology and its future potentials. Sustainability 2021;13:2962. http://dx.doi.org/10.3390/su13052962, URL https://www.mdpi.com/2071-1050/13/5/2962.

[15] Gao Z. On the centrality of disturbance rejection in automatic control. ISA Trans 2014;53(4):850–7. http://dx.doi.org/10.1016/j.isatra.2013.09.012, URL http://www.sciencedirect.com/science/article/pii/S0019057813001559, Disturbance Estimation and Mitigation.

[16] Lotufo MA, Colangelo L, Perez-Montenegro C, Canuto E, Novara C. UAV quadrotor attitude control: An ADRC-EMC combined approach. Control Eng Pract 2019;84:13–22. http://dx.doi.org/10.1016/j.conengprac.2018.11.002, URL https://linkinghub.elsevier.com/retrieve/pii/S0967066118305148.

[17] Han J. From PID to active disturbance rejection control. IEEE Trans Ind Electron 2009;56(3):900–6. http://dx.doi.org/10.1109/TIE.2008.2011621.

[18] Lv J, Wang Y, Tang C, Wang S, Xu W, Wang R, et al. Disturbance rejection control for underwater free-floating manipulation. IEEE/ASME Trans Mechatronics 2021. http://dx.doi.org/10.1109/TMECH.2021.3129836.

[19] Zhao Z, Liu Z. Finite-time convergence disturbance rejection control for a flexible timoshenko manipulator. IEEE/CAA J Autom Sin 2021;8(1):157–68. http://dx.doi.org/10.1109/JAS.2020.1003378.

[20] Wang G, Chadli M, Basin M. Practical terminal sliding mode control of nonlinear uncertain active suspension systems with adaptive disturbance observer. IEEE/ASME Trans Mechatronics 2021;26:789–97. http://dx.doi.org/10.1109/TMECH.2020.3000122.

[21] Kommuri S, Han S, Lee S. External torque estimation using higher order sliding-mode observer for robot manipulators. IEEE/ASME Trans Mechatronics 2022;27:513–23. http://dx.doi.org/10.1109/TMECH.2021.3067443.

[22] Gao Z. Active disturbance rejection control: A paradigm shift in feedback control system design. In: 2006 American control conference. 2006, p. 7. http://dx.doi.org/10.1109/ACC.2006.1656579.

[23] Fareh R, Khadraoui S, Abdallah MY, Baziyad M, Bettayeb M. Active disturbance rejection control for robotic systems: A review. Mechatronics 2021;80:102671. http://dx.doi.org/10.1016/j.mechatronics.2021.102671, URL https://www.sciencedirect.com/science/article/pii/S0957415821001392.

[24] He D, Li Y, Meng X, Si Q. Anti-slip control for unmanned underwater tracked bulldozer based on active disturbance rejection control. Mechatronics 2022;84:102803. http://dx.doi.org/10.1016/j.mechatronics.2022.102803, URL https://www.sciencedirect.com/science/article/pii/S0957415822000472.

[25] Liao J, Chen Z, Yao B. Model-based coordinated control of four-wheel independently driven skid steer mobile robot with wheel–ground interaction and wheel dynamics. IEEE Trans Ind Inf 2019;15(3):1742–52. http://dx.doi.org/10.1109/TII.2018.2869573.

[26] Yao B, Tomizuka M. Adaptive robust control of SISO nonlinear systems in a semi-strict feedback form. Automatica 1997;33(5):893–900. http://dx.doi.org/10.1016/S0005-1098(96)00222-1, URL https://www.sciencedirect.com/science/article/pii/S0005109896002221.

[27] Canuto E. Embedded model control: Outline of the theory. ISA Trans 2007;46(3):363–77. http://dx.doi.org/10.1016/j.isatra.2007.01.006, URL http://linkinghub.elsevier.com/retrieve/pii/S0019057807000481.

[28] Acuña-Bravo W, Molano-Jiménez AG, Canuto E. Embedded model control for underactuated systems: An application to furuta pendulum. Control Eng Pract 2021;113:104854. http://dx.doi.org/10.1016/j.conengprac.2021.104854, URL https://www.sciencedirect.com/science/article/pii/S0967066121001313.

[29] Acuña-Bravo W, Canuto E, Agostani M, Bonadei M. Proportional electro-hydraulic valves: An embedded model control solution. Control Eng Pract 2017;62:22–35. http://dx.doi.org/10.1016/j.conengprac.2017.01.013, URL https://www.sciencedirect.com/science/article/pii/S0967066117300138.

[30] Perez-Montenegro C, Colangelo L, Pardo J, Rizzo A, Novara C. Asynchronous multi-rate sampled-data control: an embedded model control perspective. In: 2019 IEEE 58th conference on decision and control. CDC, IEEE; 2019, p. 2628–33. http://dx.doi.org/10.1109/CDC40024.2019.9030102, URL https://ieeexplore.ieee.org/document/9030102/.

[31] Canuto E, Montenegro CP, Colangelo L, Lotufo M. Active disturbance rejection control and embedded model control: A case study comparison. In: Proceedings of the 33rd Chinese control conference. IEEE; 2014, p. 3697–702. http://dx.doi.org/10.1109/ChiCC.2014.6895554, URL http://ieeexplore.ieee.org/document/6895554/.

[32] MATLAB. 9.11.0.2022996 (R2021b) Update 4. Natick, Massachusetts: The MathWorks Inc.; 2021.

[33] Novara C, Canuto E, Carlucci D. Control of systems with sector-bounded nonlinearities: robust stability and command effort minimization by disturbance rejection. Control Theory Technol 2016;14(3):209–23. http://dx.doi.org/10.1007/s11768-016-6017-6, URL https://link.springer.com/article/10.1007/s11768-016-6017-6.

[34] Dexter industries company website. 2022, https://www.dexterindustries.com/. (Last Access 19 November 2022).

[35] Canuto E. On dynamic uncertainty estimators. In: 2015 American control conference. ACC, IEEE; 2015, p. 3968–73. http://dx.doi.org/10.1109/ACC.2015.7171949, URL http://ieeexplore.ieee.org/document/7171949/.

[36] Canuto E, Montenegro CP, Colangelo L, Lotufo M. Embedded model control: Design separation under uncertainty. In: Proceedings of the 33rd Chinese control conference. IEEE; 2014, p. 3637–43. http://dx.doi.org/10.1109/ChiCC.2014.6895544, URL http://ieeexplore.ieee.org/document/6895544/.

[37] Acuña-Bravo W, Molano-Jiménez AG, Canuto E. Embedded model control, performance limits: A case study. DYNA 2017;84(201):267. http://dx.doi.org/10.15446/dyna.v84n201.59165, URL http://revistas.unal.edu.co/index.php/dyna/article/view/59165.

Luca Nanu received a Master degree in Mechatronic Engineering in 2020. After the Master Degree he worked for Stellantis company, as automotive technical engineer for hybrid/electric vehicles. He is now working at Politecnico di Torino, as a Ph.D. fellow in Aerospace Engineering. His current research interests include simulation and modelling of mechatronic systems, system identification, automation and control, electrical machines, flight dynamics, aircraft design and urban air mobility.

Luigi Colangelo received the master's degree in aerospace engineering from the Politecnico di Torino, Turin, Italy, and Politecnico di Milano, Milan, Italy, in 2013, and the ESA NPI-Ph.D. degree in control and computer engineering from the Politecnico di Torino, in partnership with the European Space Agency and Thales Alenia Space, Turin, in 2018. In 2018, he joined the Department of Electronics and Telecommunications, Politecnico di Torino, as a Research Affiliate. His current research interests include space guidance, navigation, and control, autonomous vehicles, and control theory.

Carlo Novara received the Laurea degree in Physics from Università di Torino (Italy) in 1996 and the Ph.D. degree in Computer and System Engineering from Politecnico di Torino (Italy) in 2002. He is currently a Professor at Politecnico di Torino (Italy). His research interests include nonlinear and LPV system identification, filtering/estimation, time series prediction, embedded model control, nonlinear control, predictive control, data-driven methods, set membership methods, sparse methods, and automotive, aerospace, biomedical and energy applications.

Carlos Norberto Perez Montenegro received the degree in Electronic Engineering from the Pontificia Universidad Javeriana (Bogota, Colombia), in 2007, and the Ph.D. in Computer And Control Engineering from the Control and Computer Engineering Department of the Politecnico di Torino (Turin, Italy), in 2014. He has worked in a wide field of applications as guidance navigation and control (GNC) applied to planetary landing, unmanned aerial systems (UAS), automotive and home appliance.