IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier 10.1109/ACCESS.2017.DOI

Execution Time of optimal controls in Hard Real Time, a Minimal Execution Time solution for nonlinear SDRE.

A. BONCI[1], (Member, IEEE), S. LONGHI[1] (Senior Member, IEEE), G. NABISSI[1] and G. A. SCALA[1]

[1] Departement of Information Engineering, Università Politecnica delle Marche, Ancona, 60131 Italy. (e-mail: a.bonci@univpm.it, sauro.longhi@univpm.it, g.nabissi@univpm.it, g.a.scala@pm.univpm.it)

Corresponding author: Andrea Bonci (e-mail: a.bonci@univpm.it).

This work was supported in part by the EU H2020 ENCORE Project "Energy aware BIM Cloud Platform in a COst-effective Building Renovation Context", European Union's Horizon 2020 research and innovation programme under Grant Agreement No 820434, in part by the REACT Project "Methods and innovative instruments for REACTive Product Design and Manufacturing", Project from the Italian Ministry of University and Research (MIUR)-in National Operative Plan (PON) for Research and Innovation 2014-2020, Project No. $ARS01-01031$, Grant Agreement D.D. July 13 2018, n. 1825 - Code $ARS01-01031$, and in part by the HD3Flab Project "Human Digital Flexible Factory of the Future Laboratory" EU ERDF (European Regional Development Fund), Regional Operative Plan (POR) MARCHE Region FESR (Fondo Europeo di Sviluppo Regionale) 2014/2020, AXIS 1, Specific Objective 2, ACTION 2.1.

ABSTRACT Many engineering fields, such as automotive, aerospace, and the emerging challenges towards industry 4.0, have to deal with Real-Time (RT) or Hard Real Time (HRT) systems, where temporal constraints must be fulfilled, to avoid critical behaviours or unacceptable system failures. For this reason, estimation of code's Worst-Case Execution Time (WCET) has received lots attention because in RT systems a fundamental requirement is to guarantee at least a temporal upper bound of the code execution for avoiding any drawbacks. However, until now there is no approved method to compute extremely tight WCET. Nowadays, indeed, HRT requirements are solved via hardware, using multi-cores embedded boards that allow the computation of the deterministic Execution Time (ET). The availability of these embedded architectures has encouraged the designers to look towards more computationally demanding optimal control techniques for RT scenarios, and to compare and analyze performances also evaluating a tight WCET. However, this area still lacks deep investigations. This paper has the intent of analysing results regarding the choice between three of the most established optimal controls (LQR, MPC, SDRE), providing the first link between WCET analysis and control algorithms performances. Moreover, this work shows how it is also possible to obtain a minimal ET solution for the nonlinear SDRE controller. The results might be useful for future implementations and for coping with Industry 4.0 emerging challenges. Furthermore, this approach can be useful in control system engineering field, especially in the design stage for RT or HRT systems, where temporal bounds have to be fulfilled jointly with all the other application's specifications.

INDEX TERMS Real time systems, Optimal control, Control systems, Worst Case Execution Time, Digital Implementation, SDRE.

I. INTRODUCTION

The embedded hardware systems used for control applications are becoming more and more performing to meet the growing needs, in terms of computational performances, required by the increasing complexity of control tasks, for which they are appointed in real-world applications. Many engineering fields, such as automotive, aerospace, and in recent years also the emerging challenges towards industry 4.0, have been already using Real-Time (RT) or Hard Real-Time (HRT) systems to deal with event-triggered and time-triggered tasks [1]. This is because of the increasing number of sensors and even more sophisticated control algorithms, require that the system cannot lose deadlines, especially if they involve safety concerns. The management and implementation of these algorithms are distributed among different tasks running on the embedded system while RT or HRT temporal constraints must be fulfilled as well, to avoid critical behaviours or unacceptable system failure. Therefore, these

HRT systems must be designed according to the resource adequacy policy, providing sufficient computing resources to handle both the specified worst-case load and fault scenario. However, in order to achieve even more demanding objectives to control systems, the linear controllers have to be overcome because in most of the control applications the PID is still the most used controller. For example, in an industrial environment, linear control algorithms are preferred to facilitate the PLC to save resources since it has to manage different architectures and respects different RT constraints. Whereas in the Automotive field, RT or HRT systems are used to guarantee security and performance to the vehicles and they often use nonlinear controllers due to the need to handle complex dynamics models [2]. For example advanced models of two-wheeled vehicles, such as those proposed in [2] for an all-wheel-drive vehicle, or in [3] for single-tracked vehicle, require at least seven coupled nonlinear equations of motion. Nevertheless, to achieve even more sophisticated Industry 4.0 objectives and to control new forthcoming systems, not in trivial fashion, the widely used PID linear controller has to be overcome. To satisfy the temporal constraints and to guarantee correctly working systems, research on timing analysis of RT systems started many years ago and has been focused on the Response Time Analysis (RTA) [4], [5], which refers to the time that a message requires to be sent and received. Indeed, many works have been devoted to the development of methods based on deterministic RTA, for an estimation of the code ET. Even though many factors have been considered, it is still difficult to analyze a real system using those methods. One of the main reason is the unpredictable randomness of the Central Processing Unit (CPU) scheduling which cannot be accurately modelled. Because of these deficiencies, usually, some degrees of pessimism are added to the model as a price for indeterminism, and the Worst-Case Execution Time (WCET) is evaluated. The WCET has received lots of attention because, in RT or HRT critical systems, it is a fundamental requirement to at least guarantee a temporal upper bound for the code execution and avoid any drawbacks. However, until now, there is no approved method to compute extremely tight WCET, as shown in [6]. Nowadays, HRT requirements can also be solved via hardware, using multi-cores embedded boards that allow the computation of the deterministic Execution Time (ET). In these boards, the running task is considered on a single thread, there is no CPU scheduling and each routine has a dedicated core. There are different companies specialized in HRT embedded boards, amongst them, a few years ago the benefits of XMOS technology have been pointed out in comparing with other boards [7]. This allows us to calculate with deterministic certainty, the Best Case Scenario (BCS) and Worst Case Scenario (WCS) with native tools by the workbench. The availability of this embedded architecture has encouraged the designers to look towards more computationally demanding optimal control techniques for RT scenarios and to compare and analyse performances also evaluating a tight WCET. Indeed, this area still lacks in-depth investigations while plenty of papers have been produced in the literature addressing the implementation of different control laws on single-core embedded architectures [8]. Therefore, the aim of this paper has been primarily to compare optimal control techniques through performance and ET analysis. This was possible by exploiting the HRT board characteristics, and it has been proposed a method to evaluate in a deterministic way the Execution Time required by the controller's implementations. Furthermore, once evaluated both performance and ET of three most used optimal controls, in the light of this novel, proposed analysis, it has been possible to improve the execution time of the State Dependent Riccati Equation (SDRE) nonlinear controller. The improvements are related to implementation issues and it has been proposed a method to achieve a minimal ET solution, useful for control regulation problems, which decrease the computational effort required by the SDRE nonlinear control algorithm. Amongst various control techniques, the optimal controls ranging from linear to advanced linear up to nonlinear, have been considered. In embedded system energy saving is crucial due to limited energy availability and optimal controls allow to accomplish control tasks also taking into account energy minimization. Moreover, optimal controls can handle systems which have strong dynamics couplings, even with significant non-linearities where the need to manoeuvre the systems in a non-trivial fashion to achieve good performances is required. They can also be used to open-loop trajectory optimization, e.g. in robotics, where they are extremely useful for robot's motion-planning [9]. This paper focuses mainly on discrete-time implementations [10], being those of greater interest for practical applications. Discrete-time version of Linear Quadratic Regulator (LQR) [11], Model Predictive Control (MPC) technique [12] and nonlinear technique based on the State Dependent Riccati Equation (SDRE) [13] have been considered and analyzed with an HRT board. The control laws have been tested first in a simulation environment and then applied to an experimental set-up using the HRT board based on XMOS technology. To this purpose, a motorized four-wheeled mobile robot has been built, it is equipped with an inverted pendulum mounted on top of and joined by means of a rotational joint. A control law has to keep the pendulum in balance while having the cart tracking the desired position in space. The system's sensing is accomplished providing data by an optical encoder measuring the angle of the pendulum, and by a distance sonar sensor tracking the cart position w.r.t. a fixed obstacle.

The remainder of this paper is organized as follows: in Section II an overview of related work is presented, Section III introduces the linear and nonlinear discrete models of the cart-pendulum system in state-space form; Section IV recall the theory behind the optimal control laws used to stabilize the system and provides some simulation results; in Section V a new minimal ET solution for nonlinear SDRE controller is presented while Section VI describes the hardware platform and the sensors used in the real control application. Section VII describes the differences regarding the implementation between classic SDRE and the minimal ET solution proposed. The Simulations and Experimental results are shown in Section VIII are shown, and Section IX concludes the paper.

II. RELATED WORK

In this subjects, most relevant works are presented considering the issues discussed above. Engineers of computer science have investigated a lot of methods to achieve a deterministic estimation of the WCET. The determination of upper bounds on ET is a necessary step in Real-time Computing (RTC) which refers to systems subject to RT constraints. Different work [14], [15], [16], [17], highlight the importance of the predictability's timing behaviour for RT embedded systems. Some work propose static and probabilistic analysis of the WCET as shown in [18], [19] and [20], while the probability response time distribution of periodic tasks on a uniprocessor system was developed in [21]. The approaches can be either pragmatic with a simulation [17] or based on the statistic methods [14]. Another possibility is to use estimation algorithms, to obtain a prediction of the future execution timing [15]. However, most existing analysis methods are based on deterministic RTA, and there is still no approved method to compute extremely tight WCET [14] [15]. Several tools and algorithms are available on the market that allow calculating the ET of the tasks with some degree of pessimism. Nevertheless, these tools can ensure reliability in HRT system only considering a higher limit that exceeds in every task the real WCET [22], [23]. In the field of embedded system, these problems are overcome using micro-controllers developed to HRT problems. Amongst various producers, the XMOS family has been chosen because deemed suitable for the novel analysis carried out in this paper and introduced in the section above. Indeed, until now at best knowledge of the authors, there are no works focused on the WCET of the control algorithms used for RT or HRT applications which aims to improve the controllers' feasibility. In literature, progress based on sensor selection in control design receives a substantial interest in the last few years. For example, in [24] a Linear-Quadratic-Gaussian (LQG) control is applied to a Maglev suspension and it has been pointed out the significance of achieving also, for RT scenario, even more performing solutions to combine multi-objective optimization through an adequate sensor choice. Regarding the control techniques, different works deal with optimal control laws and suggest a discrete implementation on micro-controllers [10]. Some works are focused on the evaluation of the performances of these optimal controllers, mainly in simulation and sometimes also experimentally. In [25] the study was performed on a simulated model of an inverted pendulum, and it has been shown that the LQR algorithm works better for stabilization problems and disturbance rejection, while the MPC controller is more suitable for the trajectory tracking task. In [26], two optimal control techniques such as LQR and SDRE, have been applied to a double inverted pendulum on a cart and these were investigated and compared. In particular, simulations reveal the superior performance of the SDRE over the LQR under strong nonlinear conditions, and some improvements that could be provided by the Neural Networks, which compensates model mismatching in the case of LQR. Differently in [27] was tested a DC motor speed controller. The simulations show that the control efforts are lower in PID and LQR than in MPC, but the MPC outperforms for reference tracking and constraints handling. In [28], assuming the same linear control parameters, the effectiveness of the SDRE control over the LQR control is demonstrated despite a more complex design. Besides, in [29] traditional optimal strategies such as LQR and SDRE are investigated when applied to the control of spacecraft formation flying. In this case, the accuracy and the cost of maintaining the requested orbital configuration are evaluated, and the analysis shows as the SDRE allows to better take into account the real dynamics at an increasing level of approximation. However, despite the above-mentioned analyzes and comparisons performed on these techniques, a timing analysis seems not has been deepened. In this regard, this paper aims to propose a novel approach to evaluate control techniques by a thorough analysis of both the computational performances and the WCET of the optimal control algorithms taking into account the already well-known validated implementations. The analysis carried out on an RT experimental set-up (a cart-pendulum mobile robot depicted in Fig.1) has the intent of showing results regarding the choice between three of the most established optimal controls, providing the first link between WCET analysis and the control algorithms performances. Moreover, a novel minimal ET solution for the nonlinear SDRE controller is proposed. These results might be useful also for future implementations to solve Industry 4.0 challenges, where Multi-Agents Systems have to collaborate and exchange messages without loose deadlines in RT. Indeed, the tendency to bring the calculation of CPS towards the edge computing, and therefore on low-cost and lower performance embedded devices, as described in [30] [31], requires the use of less computationally demanding control algorithms. Otherwise, once designed an SDRE nonlinear controller based on the system model, it is possible to develop more easily a digital twin to monitoring parameters and system's conditions in broader functioning range and to apply a predictive maintenance procedure, as proposed in [32]. However, the proposed analysis could be useful in control systems engineering especially during the control's design stage for RT or HRT systems, where temporal bounds have to be fulfilled jointly with all the other application's specifications.

III. CART-PENDULUM MODELS

In this section, the model of the cart-pendulum robotic system depicted in Figure 1 has been derived. Firstly, the nonlinear model is presented, and it is shown how to obtain both the nonlinear State-Dependent Coefficient (SDC) Factorization, and the discrete-time model in state-space form, which have been used in control implementation. Then, the model

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.3019776, IEEE Access

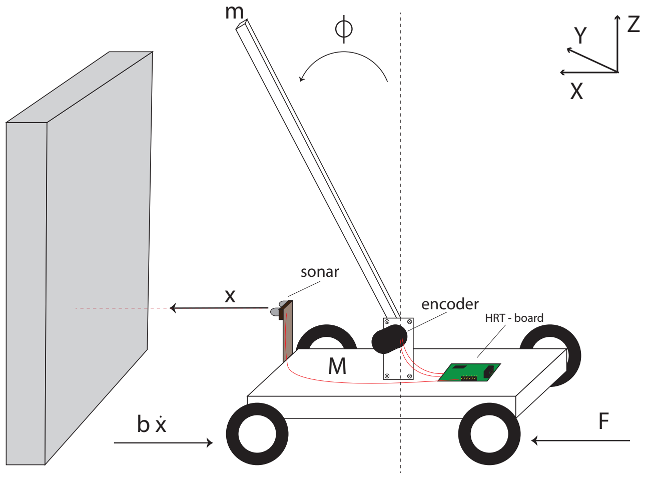

FIGURE 1: Cart pendulum system.

derived has been trivially linearized for developing the LQR and MPC linear control techniques as well. The different control laws will be recalled in the next sections.

To obtain the above mentioned discrete-time models, the nonlinear continuous-time dynamic equations for the cart-pendulum system in Figure 1 has been developed first. Then, the system has been rewritten in its state-space linear-like form for subsequent nonlinear control implementation.

A. CART-PENDULUM NONLINEAR DYNAMIC EQUATIONS

The nonlinear equations for the cart-pendulum system in Figure 1 are known and can be summarized in a compact form as follows:

$$\begin{aligned} &(m+M)\ddot{x} + (b+\gamma)\dot{x} + m\,L\ddot{\phi}\cos\phi - \\ &\quad m\,L\dot{\phi}^2\sin\phi = \beta\,E \\ &-m\,L\cos\phi\ddot{x} - (J_p + m\,L^2)\ddot{\phi} + m\,g\,L\sin\phi = 0 \end{aligned} \tag{1}$$

Meaning and description of the variables and physical parameters in Eq.(1) are listed in Table 1. The system's control input is the motor voltage, here defined as E, it actuates the electric motors connected with the wheels, their dynamics generates external forces and moments, here defined as F (see Fig. 1). The modelling details of motors-cart interaction are reported in Appendix B.

B. SDC FACTORIZATION OF NONLINEAR CART-PENDULUM MODEL

The system (1) can be written in linear-like form obtaining the SDC matrix. The system's state is defined as $\boldsymbol{x}_s = [x, \dot{x}, \phi, \dot{\phi}]^T$, the continuous-time system's SDC matrix is $A_{c_{nl}}(\boldsymbol{x}_s)$, the system is control-affine, its input matrix is $B_{c_{nl}}$, the output matrix is C_{nl} while $u = E$ is the control input. Therefore the linear-like form of the system can be written as:

TABLE 1: Physical parameters of the cart-pendulum system

Parameter	Value	Description
M[kg]	2.683	Mass of the cart
m[kg]	0.062	Mass of the pendulum
L[m]	0.303	Half the length of the pendulum
b[N/m sec]	0.15	Coefficient of friction of the cart
J_p[kg m^2]	0.0019874	Inertia moment of the pendulum with respect to the CoM (centre of mass)
g[m/ sec^2]	9.8065	Gravitational acceleration
F[N]		Force applied to the cart
x[m]		Longitudinal displacement of the cart
ϕ[rad]		Angle of the pendulum from the upper vertical axis
R_r[m]	0.0605	Wheel radius
M_R[kg]	0.127	Wheel mass
I_R[kg m^2]	2.28e(-4)	Wheel inertia

$$\begin{aligned} \dot{\boldsymbol{x}}_s &= A_{c_{nl}}(\boldsymbol{x}_s)\,\boldsymbol{x}_s + B_{c_{nl}}(\boldsymbol{x}_s)\,u \\ y &= C_{c_{nl}}\boldsymbol{x}_s \end{aligned} \tag{2}$$

The variables x, $\dot{x}$ represent the cart's distance and the linear velocity respectively, while ϕ, $\dot{\phi}$ are the pendulum's angular displacement and angular velocity, respectively. Appendix A, explains how to derive a SDC matrix solution. The SDRE control technique, has been chosen for consistency of analysis, wishing to focus the paper on optimal control methods. By manipulating equations (2) following recommendations are given in Appendix A, the system matrices become the follows:

$$A_{c_{nl}} = \begin{bmatrix} 0, & 1, & 0, & 0 \\ 0, & -\frac{(b+\gamma)(J_p+mL^2)}{\eta_{nl}}, & -\frac{m^2L^2g\,s_3\,c_3}{x_3\eta_{nl}}, & \frac{mL\,x_4s_3(J_p+mL^2)}{\eta_{nl}} \\ 0, & 0, & 0, & 1 \\ 0, & \frac{mL(b+\gamma)c_3}{\eta_{nl}}, & \frac{mgLs_3(M+m+\xi)}{x_3\eta_{nl}}, & -\frac{m^2L^2\,x_4s_3\,c_3}{\eta_{nl}} \end{bmatrix}$$
$$B_{c_{nl}}(\boldsymbol{x}_s) = \begin{bmatrix} 0 \\ \frac{J_p+mL^2}{\eta_{nl}}\beta \\ 0 \\ -\frac{mL\,c_3}{\eta_{nl}}\beta \end{bmatrix}, \; C_{c_{nl}} = \left[\begin{smallmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{smallmatrix}\right] \tag{3}$$

where for reasons of space the dependence on the state ($A_{c_{nl}}(\boldsymbol{x}_s)$) of has been here omitted with s_3=sin (x_3), c_3=cos (x_3) and η_{nl}=$(J_p+mL^2)(M+m)-m^2L^2\cos^2(x_3)$. To ease the reading, the matrices $A_{c_{nl}}(\boldsymbol{x}_s)$, $B_{c_{nl}}(\boldsymbol{x}_s)$ can been arranged as:

$$A_{c_{nl}}(\boldsymbol{x}_s) = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & a_{22} & a_{23} & a_{24} \\ 0 & 0 & 0 & 1 \\ 0 & a_{42} & a_{43} & a_{44} \end{bmatrix} \quad B_{c_{nl}}(\boldsymbol{x}_s) = \begin{bmatrix} 0 \\ b_{21} \\ 0 \\ b_{41} \end{bmatrix} \tag{4}$$

where

$$a_{22} = -\frac{(b+\gamma)(J_p+mL^2)}{\eta_{nl}} \tag{5}$$

$$a_{23} = -\frac{\sin(x_3)\cos(x_3)m^2L^2g}{x_3\eta_{nl}} \tag{6}$$

$$a_{24} = \frac{mL\sin(x_3)x_4(J_p+mL^2)}{\eta_{nl}} \tag{7}$$

$$a_{42} = \frac{mL(b+\gamma)\cos(x_3)}{\eta_{nl}} \tag{8}$$

$$a_{43} = \frac{mgL\sin(x_3)(M+m)}{x_3\eta_{nl}} \tag{9}$$

$$a_{44} = -\frac{m^2L^2\sin{(x_3)}\cos{(x_3)}x_4}{\eta_{nl}} \tag{10}$$

$$b_{21} = \frac{J_p + mL^2}{\eta_{nl}}\,\beta \tag{11}$$

$$b_{41} = -\frac{mL\cos{(x_3)}}{\eta_{nl}}\,\beta \tag{12}$$

In order to implement the nonlinear controller on the proposed board for computational performance and WCET analysis, the discrete-time non linear model of (3) is derived by using the Euler method with sampling time ΔT, giving rise to following discrete-time model:

$$\begin{aligned} \boldsymbol{x}_s[k+1] &= A_{d_{nl}}(\boldsymbol{x}_s)\,\boldsymbol{x}_s[k] + B_{d_{nl}}(\boldsymbol{x}_s)\,u[k] \\ y[k] &= C\boldsymbol{x}_s[k] \end{aligned} \tag{13}$$

with $u[k] = E[k]$ and discrete-time matrices

$$\begin{aligned} A_{d_{nl}}(\boldsymbol{x}_s) &= \begin{bmatrix} 1, & \Delta T & 0, & 0 \\ 0, & 1{+}\Delta T a_{22}, & \Delta T a_{23}, & \Delta T a_{24} \\ 0, & 0, & 1, & \Delta T \\ 0, & \Delta T a_{42}, & \Delta T a_{43}, & 1{+}\Delta T a_{44} \end{bmatrix}, \\ B_{d_{nl}}(\boldsymbol{x}_s) &= \begin{bmatrix} 0 \\ \Delta T b_{21} \\ 0 \\ \Delta T b_{41} \end{bmatrix}, \; C = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \end{aligned} \tag{14}$$

In (14) the terms a_{ij},b_{ij} are the elements of continuous matrices $A_{c_{nl}}$, $B_{c_{nl}}$ described in the equations (5) to (12).

C. DISCRETE-TIME CART-PENDULUM LINEAR MODEL

This subsection, briefly introduces the discrete-time linear model here used to design the linear optimal control strategies described in the following Section. The model is obtained from the equations of motion (1) under small perturbation hypothesis $\phi \simeq 0$ around the upright position, and hence $cos(\phi) \simeq 1$, $sin(\phi) \simeq \phi$, $\dot{\phi}^2 \simeq 0$. The resulting linear dynamics is:

$$\begin{aligned} (m+M)\ddot{x} + (b+\,\gamma)\dot{x} + m\,L\ddot{\phi} &= \beta\,E \\ -\,m\,L\ddot{x} - (J_p + m\,L^2)\ddot{\phi} + m\,g\,L\phi &= 0 \end{aligned} \tag{15}$$

From (15), it is trivial to derive the discrete-time state space model using Euler method with sampling time ΔT. It has the form:

$$\begin{aligned} \boldsymbol{x}_s[k+1] &= A_d\,\boldsymbol{x}_s[k] + B_d\,u[k] \\ y[k] &= C\boldsymbol{x}[k] \end{aligned} \tag{16}$$

with $u[k] = E[k]$ and matrices

$$\begin{aligned} A_d &= \begin{bmatrix} 1, & \Delta T & 0, & 0 \\ 0, & 1{+}\Delta T a_{22}, & \Delta T a_{23}, & 0 \\ 0, & 0, & 1, & \Delta T \\ 0, & \Delta T a_{42}, & \Delta T a_{43}, & 1 \end{bmatrix} \\ B_d &= \begin{bmatrix} 0 \\ \Delta T b_{21} \\ 0 \\ \Delta T b_{41} \end{bmatrix}, C_d = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \end{aligned} \tag{17}$$

where

$$a_{22} = -\frac{(b+\gamma)(J_p + mL^2)}{\eta_{nl}} \tag{18}$$

$$a_{23} = -\frac{m^2L^2g}{\eta_{nl}} \tag{19}$$

$$a_{42} = \frac{mL(b+\gamma)}{\eta_{nl}} \tag{20}$$

$$a_{43} = \frac{mgL(M+m)}{\eta_{nl}} \tag{21}$$

$$b_{21} = \frac{J_p + mL^2}{\eta_{nl}}\beta \tag{22}$$

$$b_{41} = -\frac{mL}{\eta_{nl}}\beta \tag{23}$$

where k is the sampling instant and the terms a_{ij}, b_{ij} are elements of the matrices $A_{c_{nl}}, B_{c_{nl}}$ in (3).

IV. OPTIMAL CONTROL LAWS

The present section is devoted to briefly recall the theory behind the three different optimal control techniques considered in this work to regulate the upright pendulum position of the cart-pendulum robotic system. The controllers have been used both in simulation and on the experimental set-up. In order to ease the paper, some implementation details which require longer description are given in the appendix, while only necessary conceptual steps for a clear understanding are presented below. Moreover, considering the issues related to this work, different control features will be pointed out because useful for further performance's analysis. We refer to the discrete-time version of the three methods. The linear techniques LQR and MPC, make use of the model (16). The nonlinear SDRE control uses model (13).

A. THE DISCRETE-TIME LQR

The LQR design technique is well known in modern optimal control theory and has been widely used in many applications.

1) LQR fundamentals

In LQR theory a stabilizable discrete-time linear system:

$$\boldsymbol{x}[k+1] = \;A\boldsymbol{x}[k]\; + B\boldsymbol{u}[k] \tag{24}$$

where $\boldsymbol{x}[k]$ is the state vector and $\boldsymbol{u}[k]$ the system's input, the optimal inear control problem is to determine the linear optimal feedback matrix $K_{opt} \in R^{nxm}$ where n, m are the state and input vectors dimension respectively, such that:

$$\boldsymbol{u}_k = K_{opt}\,\boldsymbol{x}_k \tag{25}$$

we refer to (16) as the discrete-time linear plant model (24). In this paper, the state feedback controller is designed using the linear quadratic regulator and the discrete-time linear model of the system (16). Here, the infinite-time regulation problem has been solved to guarantee stability for further performance and WCET analysis. This solution leads to

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.3019776, IEEE Access

search for optimal control $\boldsymbol{u}_k$ which minimizes the following cost function:

$$J(\boldsymbol{u}) = \frac{1}{2}\sum_{k=0}^{\infty}(\boldsymbol{x}_k^T\, Q_k\, \boldsymbol{x}_k + \boldsymbol{u}_k^T\, R_k\, \boldsymbol{u}_k) \tag{26}$$

where Q_k, R_k are the state and input weighing matrices respectively, they are symmetric and positive defined, and k is the sampling instant. In order to ease the paper readability, the subscript s in the state vector $(\boldsymbol{x}_s)$ has been omitted. It has been demonstrated that the optimal control input $\boldsymbol{u}_k$ is given by:

$$\boldsymbol{u}_k = -R^{-1}{B_d}^T\, P\, \boldsymbol{x}_k = -K_{opt}\boldsymbol{x}_k \tag{27}$$

where the matrix P is the solution of the Discrete Algebraic Riccati Equation (DARE) problem:

$$\begin{aligned} &P = Q + {A_d}^T P A_d - \\ &{A_d}^T P B_d (R + {B_d}^T P B_d)^{-1} B^T P A_d \end{aligned} \tag{28}$$

Then, once computing off-line the DARE the linear optimal control input is obtained as well.

2) LQR features

In the conventional LQR design method, the DARE problem (28) is solved using numerical or iterative methods. For regulation problems, the DARE is solved once. Although, the LQR provides a simple solution its application is limited to linear systems or a linear approximation of nonlinear systems. Linear approximations are quite simple to handle in regulation problems, but for trajectory tracking problems, the LQR finds limits when dealing with not negligible nonlinearities. Often the solution can be found linearizing the system in a subset of the trajectory points along the whole trajectory but this is time-consuming whit respect to other techniques involving nonlinear controllers [39].

B. DISCRETE-TIME MPC

MPC has become a widely used control in the last decades, especially in industrial applications because of its versatility and capability to perform industrial process optimization. The linear MPC approach has been here considered to undertake a common comparison framework. A comprehensive theory of the MPC can be found in [12], but it is out of the scope of this paper. Thus, here the main implementation steps have been recalled, while more details are given in Appendix C.

1) MPC fundamentals

For discrete-time MPC implementation, we refer to the discrete time plant model (16). The MPC theory is based on the extended model (53) reported in Appendix C, and is derived by defining an augmented state $\boldsymbol{x}_e[k]$, wherein the variable $\Delta u[k]$ becomes the new input to be controlled in place of $u[k]$. How to derive the extended model and all the related mathematical manipulations is detailed in the appendix C. As for the other optimal controllers, the optimal control input $\Delta u[k]$ is obtained by minimizing a cost function J, defined as follow:

$$J = (\boldsymbol{R}_s - \boldsymbol{Y})^T(\boldsymbol{R}_s - \boldsymbol{Y}) + \Delta\boldsymbol{U}^T R \Delta\boldsymbol{U} \tag{29}$$

where, the first term $(\boldsymbol{R}_s - \boldsymbol{Y})^T(\boldsymbol{R}_s - \boldsymbol{Y})$ is related to the minimization between the predictive output $\boldsymbol{Y}$ and the set-point $\boldsymbol{R}_s$, while the second one $\Delta\boldsymbol{U}^T R\Delta\boldsymbol{U}$ is referred to the size of $\Delta\boldsymbol{U}$. Indeed, $R = r_\omega I$, and r_ω is used as a tuning parameter for closed-loop performance. The optimal $\Delta\boldsymbol{U}$ minimizing the cost function J holds:

$$\Delta\boldsymbol{U} = (\Phi^T\Phi + R)^{-1}\Phi^T(\boldsymbol{R}_s - F\boldsymbol{x}_e[k]) \tag{30}$$

where $(\Phi^T\Phi + R)^{-1}\Phi^T\boldsymbol{R}_s$ represents the set-point change, while $-(\Phi^T\Phi + R)^{-1}\Phi^T F$ is the state feedback control within the framework of predictive control, where the F matrix meaning is derived from the extended model formulation given in Appendix C.

2) MPC features

The optimal input $\Delta\boldsymbol{U}$ of equation (30), contains the controls $\Delta\boldsymbol{u}[k_i]$, $\Delta\boldsymbol{u}[k_i+1]$, ... $\Delta\boldsymbol{u}[k_i+N_c-1]$ with the receding horizon window (N_c). However, only the first sample of this sequence needs to be implemented, i.e., $\Delta u[k_i]$, while ignoring the rest of the sequence. When the next sample period arrives, the more recent measurement is taken to form the state vector $\boldsymbol{x}_e[k_{i+1}]$ for calculation of the new sequence of the control signal, as shown in the following:

$$\begin{aligned} \Delta\boldsymbol{u}(k_i) &= \overbrace{[1\,0\,..\,0]}^{N_c}\,(\Phi^T\Phi + R)^{-1}\Phi^T(\boldsymbol{R}_s - F\boldsymbol{x}_e[k]) \\ &= K_y r(k_i) - K_{mpc}x(k_i) \end{aligned} \tag{31}$$

where K_y is the first element of $(\Phi^T\Phi + R)^{-1}\Phi^T\boldsymbol{R}_s$, while K_{mpc} is the first row of $-(\Phi^T\Phi + R)^{-1}\Phi^T F$. This procedure, when iterated in real-time gives rise to the receding horizon control law. Moreover, the optimal input $\Delta u[k_i]$ implies that a discrete integrator is to be embedded in the closed-loop to derive the right $\boldsymbol{u}[k]$ for the plant. This integration acts as a low-pass filter on the controlled quantity making its transitions smoother; indeed the predicted optimal control input at time-instant k is obtained as $\boldsymbol{u}[k+1] = \boldsymbol{u}[k] + \Delta\boldsymbol{u}[k+1]$. The MPC key feature which should be highlighted is the versatility of control input implementation (31). Indeed, the derived implementation is valid both for stabilization and trajectory tracking problems. This means that in terms of computational effort and WCET of the code, having a constant or variable set-point which is changing over time does not affect the design and the computational effort required by the MPC control algorithm. Indeed, in (31) K_y is a constant weighting matrix multiplied for the reference value $r(k_i)$ at the k_i instant. Thus, for each next instant, a change in the set-point value is taken into consideration producing the respective optimal control input $\boldsymbol{u}[k+1]$.

C. THE DISCRETE-TIME SDRE

The SDRE strategy has become popular within the control community over the last decade, it provides an effective model-based method for nonlinear feedback control synthesis, by allowing nonlinearities in the system states while additionally offering good design flexibility through state-dependent weighting matrices. Different papers have been produced regarding the design and the implementation issues related to the non-uniqueness of the SDC matrix factorization [33], [34], this also for trajectory tracking control problems [35]. It has been demonstrated with a correct SDC factorization strong nonlinear behaviours can be handled as well [36]. However, achieving a new SDC matrix design is out of the scope of this paper and for this reason, a common approach has been used as proposed in [37], and the latter has been discussed in more detail in the dedicated appendix A.

1) SDRE fundamentals

The SDRE technique here recalled has been based on the SDC parametrization of the plant model obtained above in (3), which represents a linear-like state space form of the nonlinear system. The cost function to be minimized with respect to the control $u[k]$ is:

$$J(\boldsymbol{u}) = \frac{1}{2}\sum_{k=0}^{\infty}(\boldsymbol{x}[k]^T\, Q(x)\, \boldsymbol{x}[k] + \boldsymbol{u}[k]^T\, R(x)\, \boldsymbol{u}[k]) \tag{32}$$

One of the key advantages offered by SDRE is the trade-off between control effort and state errors, and it can be achieved through tuning of weighting matrices $Q(x)$ and $R(x)$. These weighting matrices can be chosen to be either constant or state dependent. However, in our application Q and R are assumed to be constant matrices to simplify the implementation and because constant matrices ensure the closed loop performance. In this case, taking into account the finite horizon problem, the sub-optimal control $\boldsymbol{u}[k]$ is given as:

$$\begin{aligned}\boldsymbol{u}(x[k]) &= -R^{-1}\, B_{d_{nl}}(x[k]))^T\, P(x[k])\, \boldsymbol{x}[k] \\ &= -K(x[k])\boldsymbol{x}[k]\end{aligned} \tag{33}$$

where $P(x[k])$ is computed at each sampling time k as the solution of the following DARE:

$$\begin{aligned}&P(x_k) = (Q + A_{d_{nl}})^T(x_k)P(x_k)A_{d_{nl}}(x_k) - \\ &A^T_{d_{nl}}(x_k)P(x_k)B_{d_{nl}}(x_k)(R + B^T_{d_{nl}}(x_k)P(x_k)B_{d_{nl}}(x_k))^{-1} \\ &\cdot B^T_{d_{nl}}(x(k)P(x_k)A_{d_{nl}}(x_k)\end{aligned} \tag{34}$$

Thus, unlike the linear case, the DARE equation has to be solved on-line at each control iteration.

2) SDRE features

Unlike the linear optimal control, the DARE equation must be solved on-line at sample time k, at least for each new value of $A_{d_{nl}}(x[k])$ and $B_{d_{nl}}(x[k])$. Indeed, in embedded applications such as in our experimental set-up, the equation (34) is solved on-line during the control actions by using iterative methods. Generally, the $P(\boldsymbol{x}[k])$ solution is first calculated off-line ($P(\boldsymbol{x}[k])_{off}$), with a huge number of backward iterations, until it reaches the convergence ($\overline{P}(\boldsymbol{x}[k])_{off}$). Then, an admissible convergence error is set (e.g. $\epsilon = 0.1$) in order to evaluate how many backward iterations the on-line DARE needs to obtain a solution $P(\boldsymbol{x}[k])_{on}$ which satisfies the norm error criteria $\|\overline{P}(\boldsymbol{x}[k])_{off} - P(\boldsymbol{x}[k])_{on}\| < \epsilon$. This procedure is repeated for different system's working conditions. Finally, it is found how many backward iterations are needed to the DARE for obtaining an almost convergence solution $\overline{P}(\boldsymbol{x}[k])_{on}$, considering the worst case as upper bound. The larger number of backward iterations, required by the worst case, are set and used to solve the DARE on-line for each sampling time k.

V. PROPOSING A NEW ON-LINE MINIMAL EXECUTION TIME SDRE CONTROL ALGORITHM

The present Section proposes a new Minimal Execution Time SDRE (MET-SDRE) for reducing the ET in the RT implementation of the SDRE nonlinear controller. It aims to maintain control performance by reducing the computational effort and therefore ET. In this work, the new solution has been developed for control regulation problems, the idea is based on the performance analysis experienced in this work in term of WCET of the SDRE algorithm by means of the HRT board. At present, limits for RT applications of the SDRE controller are generally related to the convergence time of the DARE (34) solving, which is desirable to be executed with few backward iteratively operations. As discussed in the previous subsection, the backward operations used to solve the DARE equation and to obtain the $P(\boldsymbol{x}[k])$ solution, are generally set experimentally in order to obtain a convergence solution in every possible system's working condition. This leads to having a large ET of the SDRE control algorithm because the WCS has to be considered as an upper bound to guarantee a correct regulation in every case, even if a convergence solution of (34) can be obtained sometimes with a smaller number of backward iterations.

In particular, in our experimental set-up the closed-loop cycle was set to 20ms which can be considered as our RT temporal constraint. Within that time (20ms) the microcontroller must be able to solve the DARE (34) iteratively until the $P(\boldsymbol{x}[k])$ solution reaches the convergence, calculate the right control law $u(\boldsymbol{x}[k])$ which depends on $P(\boldsymbol{x}[k])$, manage the data gathering from sensors and generate the PWM signal to drive the motors. In classic SDRE approaches, considering the RT constraint (20ms), to obtain a right control law a convergence criterion for the solution of the DARE (34) has to be established, as presented in Section IV-C2. Then, it can be concluded that to solve the DARE (34) and obtain a convergence solution $\overline{P}(\boldsymbol{x}[k])$ with an admissible convergence error ϵ, at least 20 times backward iterations are needed, considering as upper bound the WCS (when the system is working out of the linear range). The convergent solution $\overline{P}(\boldsymbol{x}[k])$ then, is used to produce the right control law $u(\boldsymbol{x}[k])$ for the system.

A. MET-SDRE BACKGROUND

The MET-SDRE algorithm aims to decrease the code-ET of the discrete-time SDRE control algorithm maintaining the same performance. This is achieved by reducing the computation time required by the DARE (34) in the SDRE algorithm in an adaptive way. This means the need to solve the DARE is evaluated adaptively by introducing a sort of system's distance from the linearity range, which is based on the degree of similarity of the SDC state-matrices between two consecutive sampling times. This result has been found analysing the convergence problem of the $\overline{P}\left(\boldsymbol{x}[k]\right)$ solution of the DARE (34) related to the SDRE nonlinear control problem. The convergence speed of the $\overline{P}(\boldsymbol{x}[k])$ depends on both the initial conditions, represented by which numerical value P the DARE's algorithm initializes, and on the degree of similarity of the SDC state-matrices between two consecutive sampling times k and $k+1$, i.e. between $A_{d_{nl}}(x[k])$ and $A_{d_{nl}}(x[k+1])$. Regarding the numerical P initialized by the algorithm, trivially, when the $P(\boldsymbol{x}[k+1])$ solution for a certain system's state $\boldsymbol{x}[k+1]$ is searched starting from a $P(\boldsymbol{x}[k]) = P$, the DARE solution of the previous system state $\boldsymbol{x}[k]$ is numerically close to that of $\boldsymbol{x}[k+1]$. Therefore the $P(\boldsymbol{x}[k+1])$ solution will converge with a less number of backward iterations with respect to initialize the searching with a P as identity matrix or else derived by previous sampling times $\boldsymbol{x}[k-i]$, with $i \geqslant 1$. Instead, concerning the similarity between two consecutive numerical SDC matrices, it is straightforward that, when two SDC matrices, namely $A_{d_{nl}}(x[k])$ and $A_{d_{nl}}(x[k+1])$, depend on two states $\boldsymbol{x}[k]$ and $\boldsymbol{x}[k+1]$, which are numerically close, it is expected that these SDC matrices are numerically similar. Thus, in this context, the convergence speed of the DARE solution $\overline{P}(x[k])$ also depends on the system's matrices numerical value $(A_{d_{nl}}, B_{d_{nl}}, C_{d_{nl}})$. Therefore it is expected that under state similarity condition $\boldsymbol{x}[k] \simeq \boldsymbol{x}[k+1]$, the DARE solution at instant $k+1$, namely $P(\boldsymbol{x}[k+1])$, converges with a smaller number of backward iterations if initialized as $P = P(\boldsymbol{x}[k])$ instead of the instants $k-i$, with $i > 1$. Taking into account these considerations, we propose to modify the DARE solver of the SDRE control algorithm making it adaptive with respect to the system's state. This means allowing it to have a faster convergence for the DARE solution. In the following, the implementation details of the proposed solution are explained. The computational solution implements two basic steps that are: 1) evaluation of matrices' similarity and 2) adaptive P initialization.

1) Evaluation of matrices' similarity

In order to evaluate the similarity (or distance) between two numerical matrices, the tools proposed in [40] have been used. Given a discrete-time nonlinear system represented in the linear-like form (3), considering two consecutive instants k and $k+1$, and the the corresponding consecutive numerical SDC matrices, namely $A_{d_{nl}}(x[k])$ and $A_{d_{nl}}(x[k+1])$, two similarity criteria (d_A; d_{λ_A}) can be established as follow:

$$\begin{aligned} d_A &= ||A_{d_{nl}}(\boldsymbol{x}[k]) - A_{d_{nl}}(\boldsymbol{x}[k+1])|| \\ d_{\lambda_A} &= ||\lambda_{A_{x[k]}} - \lambda_{A_{x[k+1]}}|| \end{aligned} \tag{35}$$

where d_A represents the pure distance between two numerical SDC matrices in two time intervals, while d_{λ_A} represents the distance between the eigenvalues of the frozen matrices in the two time intervals. These values (d_A, d_{λ_A}) quantify the similarity between SDC matrices which affects the convergence speed of the DARE's solution $\overline{P}(\boldsymbol{x}[k])$. In this work it has been experimentally proven that in the linear working-range of the experimental set-up, two consecutive states yield values having an order of magnitude $d_A \simeq 8 \cdot 10^{-2} \simeq -2$, $d_{\lambda_A} \simeq 0.5 \cdot 10^0 \simeq 0$. Differently, when the states belong to the nonlinear range the values become $d_A \simeq 3 \cdot 10^{-4} \simeq -4$; $d_{\lambda_A} \simeq 1.5 \cdot 10^{-5} \simeq -5$. Thus, out of the linear-range the biggest numerical difference between two consecutive matrices can be measured with the above introduced metrics, the bigger the distance the larger the number of backward iterations for solving the DARE.

2) adaptive P initialization

As explained in the above Section, the P chosen to initialize the SDRE algorithm affects the convergence speed of the DARE solution $\overline{P}(\boldsymbol{x}[k])$. For this reason, an adaptive P initialization based on the current system state $\boldsymbol{x}[k]$ can improve the ET required by the SDRE algorithm. A possible solution is to calculate off-line $\overline{P}(\boldsymbol{x}[k])$ for different working ranges (e.g. linear and nonlinear ranges) and to define different SDRE algorithm initialization conditions. Trivially, the proposed methodology should be evaluated depending on the application. In this paper, the proposed solution is applied to the cart-pendulum mobile robot and is detailed in the Experimental results Section VIII-C.

VI. HARDWARE SET-UP

In this Section are presented the hardware characteristics of the whole system and the control strategies implemented in the HRT board to stabilize the pendulum in upright position while the cart is maintaining the desired distance from obstacles.

A. CART-PENDULUM ROBOTIC SYSTEM

As discussed in the Introduction section, the hardware chosen to control the cart-pendulum system consists of the XMOS XK-1A board, a low cost development platform intended for exploring parallel computation. The XK-1A comprises 128KBytes SPI FLASH memory, four LEDs, and two press-button switches. An XTAG-2 debug adapter can be connected to a PC to debug the XK-1A operations. The XK-1A is based on a single XS1-L1 device in a 128TQFP package. The XS1-L1 hosts four deterministic cores operating at 100Mhz and it provides a tightly integrated general purpose I/O pins and 64 KBytes of on-chip RAM. The YUMO E6A2-CWZ3E Rotary Encoder sensor provides the measure of the pendulum angle while the HC-SR04 ultrasonic distance sensor provides precise, no contact distance measurements up to about 3 meters from the reference obstacle. It is simple to connect the sensor to micro-controllers requiring only one I/O pin. The DC motors are operated by the shared output voltage provided by a Sabertooth system 2x12 motor driver. This circuit can be powered with voltage ranging from 6 to 24 V, with up to 12 A continuous per channel. The board is used in R/C Mode/Microcontroller mode with a PWM from 1000us to 2000us, respectively corresponding from -12V to +12V. The whole system is powered by a LiPo battery, 2200mAh, 3S (3 cells), with discharge rate 65C and 12V.

B. HRT CONTROL STRATEGIES

The XK-1A board of the XMOS family has been used to implement the control algorithms described above. This technology was developed a few years ago, and currently new more performing boards are available from the XMOS company but the XK-1A hardware characteristics are still appropriated for this paper. The closed-loop controls are based on the direct measures of the cart position (x) and the pendulum angle (ϕ) provided by the sensors. At each sample time the controllers compute the feedback control law and the optimal voltage ($u[k]$) to be applied to the motors driver (i.e. the sabertooth device), in order to keep the pendulum in balance ($\phi = 0$) while maintaining the cart to the desired x position. The sensors are directly connected to different I/O pins of the board. The three different control strategies have been entirely coded on the XK-1A XMOS board: where at each of the four deterministic cores has been assigned a task. The core 1 runs the task 1 that manages the gathering of data from the position sensor. The core 2 hosts the task 2 that manages the gathering of data from the encoder for the feedback on the ϕ angle. Both tasks are in charge of making available their data, through the respective channels, to the task 3. This latter runs on core 3 and retrieves the sensor data from the proper channel every 20ms, within this time laps task 3 must compute the control law and send the message to the task 4 running on core 4 which generates the PWM to drive the motors. All the tasks are parallel and their ET is computed statically by means of the TimeAnalyzer, a feature provided by the development environment XTimeComposer of XMOS. The synchronization between the cores operations has been achieved by calculating statically the computing time of each task.

VII. CLASSIC SDRE AND MET-SDRE NONLINEAR IMPLEMENTATIONS

In this subsection the implementation of the algorithms regarding the classic SDRE and the proposed MET-SDRE algorithm solution, in the HRT board are explained in more detail.

A. CLASSIC SDRE ALGORITHM

Regarding the classic SDRE algorithm, common features have been already explained in Subsection IV-C2. In this part, the classic SDRE implementation used for stabilization problems on the cart-pendulum mobile robot is analyzed in more detail. Further implementation issues and feasibility aspects have been deeper discussed in [33], [38], where are given also the guidelines regarding the construction of the SDC matrix when the SDRE solvability condition is violated. The implementation steps are the following:

i) The DARE (34) is solved off-line until the $\overline{P}(\boldsymbol{x}[k])_{off}$ reaches the convergence imposing specific system condition (e.g. $\phi = 0.06\,rad$ and $\dot{\phi} = 0\,rad/s$).

ii) The necessary backward iterations for on-line implementation are found as discussed in Subsection IV-C2. It has been experimentally found that within 20 backward iterations the norm error criteria imposed as $\|\overline{P}(\boldsymbol{x}[k])_{off} - P(\boldsymbol{x}[k])_{20}\| < \epsilon$ with $\epsilon = 0.1$ was satisfied. This test, was repeated for different points of work of the cart-pendulum mobile robot to guarantee a consistent control law $\boldsymbol{u}(x[k])$ during the experimental phase.

iii) Every sample time k, the SDRE control algorithm solves the DARE (34) 20 times backwards iteratively.

B. MET-SDRE ALGORITHM

The idea of this MET-SDRE solution has been achieved through the WCET analysis experienced in this work and it aims to improve the classic SDRE approach in terms of ET. In the following, the implementation steps are listed below:

i) The linear working range of the system has been estimated through experimental tests, e.g. in our case the linear working range of the cart-pendulum is $\phi_m \leq \phi \leq \phi_M$, namely boundary conditions, where in our experimental setup $\phi_m = -4°$ and $\phi_M = 4°$.

ii) It has been calculated off-line a convergence solution $\overline{P}(x[k])_{1000}$ in 1000 backward iterations for the three different boundary conditions of the system $\phi = -4°, 0°, 4°$. These solutions have been named P_{ϕ_m}, $P_{0°}$, P_{ϕ_M}.

iii) The previous solutions have been used to initialize the adaptive DSDRE control algorithm to reach a faster convergence speed depending on the system's working range.

In Figure 2, the flow chart of the MET-SDRE control algorithm is shown. The initialization of the Algorithm is composed by the reading of the initial system's output (Y_0) and the computation of the state (X_0), then the matrix P_0 is chosen based on the initial state (X_0). The periodic control task in this implementation last $20ms$, it consists of the computation of the current state based on the output acquisition Y_k. In order to compute the DARE convergent solution $\overline{P}(\boldsymbol{x}[k])$, the number of backward iterations i is set based on the current pendulum angle ϕ. Then, if ϕ belongs to the linear range $i = I_m$, otherwise $i = I_M$. In our case, the values of $I_m = 3$ and $I_M = 10$ are found experimentally following the proposed procedure in Section V. Finally, the control input $u(X_k)$ is computed, and the periodic task stops upon termination decided by the user.

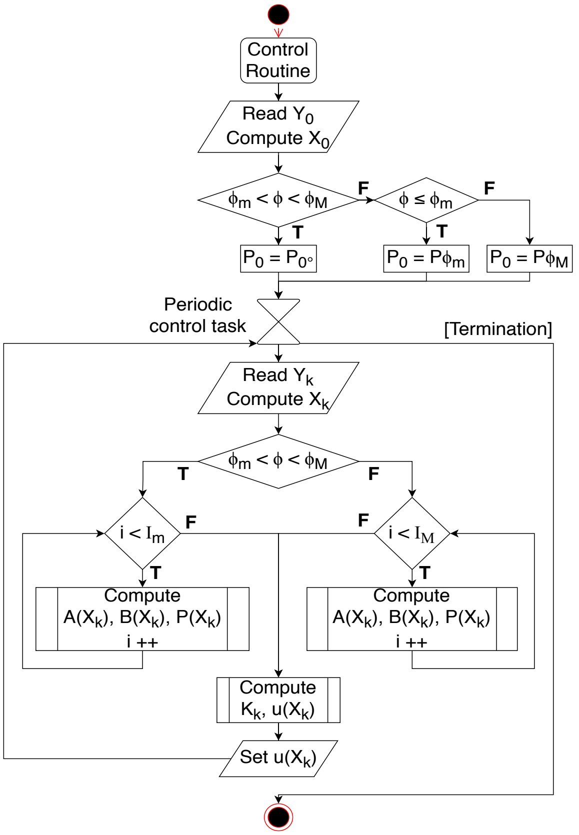

FIGURE 2: Flow chart of the MET-SDRE algorithm. In the Figure T and F state respectively for True and False.

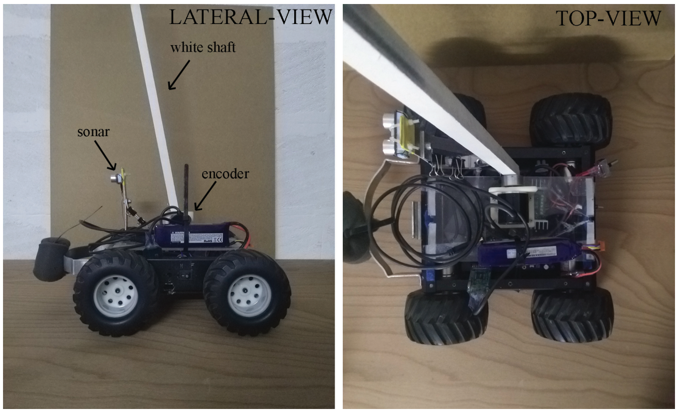

FIGURE 3: Cart-pendulum prototype developed by the authors.

VIII. PERFORMANCE'S ANALYSIS AND WCET OF THE OPTIMAL CONTROL ALGORITHMS

In this section, the simulation and experimental results regarding the cart-pendulum robotic system (Fig. 3) subject to the three above mentioned controllers are shown. The control performance is evaluated under different working condition of the real robot. The control parameters are introduced in Subsection VIII-A, while in Subsection VIII-B the simulations are compared with experimental results to demonstrate the validity of controllers' implementation. Lastly in Subsection VIII-C the experimental results are presented and discussed, with related performances and WCET analysis.

A. CONTROL PARAMETERS

Each controller has been designed and tuned to stabilize the cart-pendulum around the pendulum's vertical position while maintaining a certain distance from obstacles detected by the sonar and for disturbance rejection. All the tuning parameters are chosen to obtain a control system which first achieves as small as possible oscillations around the vertical position, secondly it maintains the desired x position and finally to have it also has a certain reactivity to disturbances. The values of the weighting matrices and tuned parameters used by the controllers in all the simulations and experimental tests are given below.

1) LQR

- The values of weighting matrices for the cost function (26) of the discrete-time LQR control are:

$$\begin{aligned} R &= 1; \\ Q &= \begin{bmatrix} 500.0, & 0.0, & 0.0, & 0.0 \\ 0.0, & 100.0, & 0.0, & 0.0 \\ 0.0, & 0.0, & 500.0, & 0.0 \\ 0.0, & 0.0, & 0.0, & 100.0 \end{bmatrix} \end{aligned} \tag{36}$$

2) MPC

- The controller is designed on the linear system plant in (16) with predicted horizon $N_p = 50$ and receding horizon $N_c = 3$, while the control weight is set to $r_\omega = 0.5$

3) SDRE

- The values of weighting matrices for the cost function (32) of the discrete-time SDRE control have been selected as those of LQR to better compare the controllers performance:

$$\begin{aligned} R &= 1; \\ Q &= \begin{bmatrix} 500.0, & 0.0, & 0.0, & 0.0 \\ 0.0, & 100.0, & 0.0, & 0.0 \\ 0.0, & 0.0, & 500.0, & 0.0 \\ 0.0, & 0.0, & 0.0, & 100.0 \end{bmatrix} \end{aligned} \tag{37}$$

B. SIMULATION RESULTS

This subsection briefly presents the simulation results of the cart-pendulum regulation problem. For showing the consistency of simulations with responses of the experimental setup, simulations have been compared with experimental results. For brevity, we reported the comparing results with only one of the controllers analyzed, that is the SDRE. Figure 4 shows the comparative results of the pendulum's ϕ angle when the SDRE controller is applied. In Figure 5 the comparative results for the cart position x are showed,

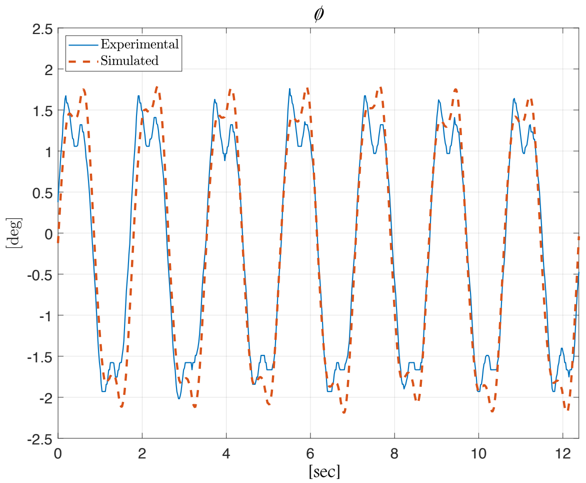

FIGURE 4: SDRE control of the cart-pendulum: comparison of simulated and experimental results of angle ϕ within similar boundary conditions.

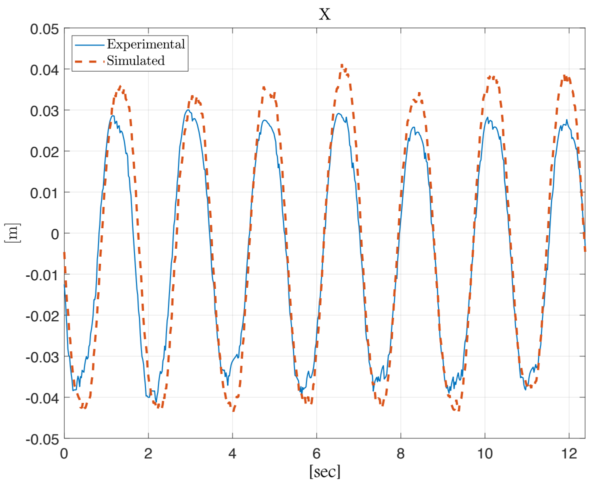

FIGURE 5: SDRE control of the cart-pendulum: comparison of simulated and experimental results of cart position x within same boundary conditions.

lastly in Figure 6, the control effort U in both cases are compared. The trend over time shown in the figures, highlights similar oscillations in term of amplitude, frequency and phase. Furthermore, small deviations between simulated and real system can be noticed in cart and pendulum direction changes, this may be attributed to the not perfect modelling of the tyre-floor contact point (tyre slip conditions). However, despite the small predictable mismatches between model and reality, by focusing on Figure 6 it can be noticed as the control input seems to be the same in both cases. This proves the effectiveness and consistency of the controller implementation in the HRT board. Besides, it is appropriate to point out that the achievement of perfect model matching with reality is out of the scope of this work.

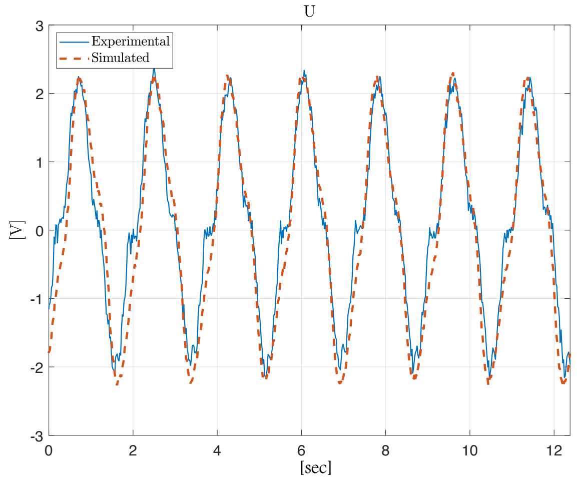

FIGURE 6: SDRE control of the cart-pendulum: comparison of simulated and experimental results of control input U within same boundary conditions.

C. EXPERIMENTAL RESULTS

This subsection is devoted to describe the experimental results obtained by applying optimal control laws in different working conditions of the experimental set-up in a RT scenario. The experiments concern the performance evaluation for the regulation problem of a cart-pendulum system. The challenge is: balancing the pendulum in a vertical position (pitch angle $\phi = 0$) while the cart maintains a desired position in space (x). In the design phase, the regulation around a fixed set-point is the primary goal to achieve by any control system. The second one is often the trajectory tracking, in which the control law forces the closed-loop response to follow a specified reference, generally time varying. Though the trajectory tracking is not the subject of this investigation, it will be shown how some considerations on it can be inferred in terms of WCET, considering the implementation of the algorithms adopted. For evaluating performance and WCET of the considered optimal controllers on regulation problems two experiments have been carried out. In the first one, the system works within the linearity range of the pendulum's vertical unstable equilibrium point, which is $\phi = 0$. For our system, it has been experimentally proofed that the range is $-4° \leq \phi \leq 4°$. The second experiment aims to evaluate the performance of the optimal controllers when nonlinear behaviours arise. In our case this happens by choosing the initial condition of the ϕ angle out of the linear region. During experiments, continuous monitoring of the WCET is performed by implementing the control Algorithms in the XMOS board. Furthermore, computational efforts required to the microcontroller by the control laws are taken into account.

1) Test 1 - linear working conditions

The Figures 7, 8, 9 show the comparison of control results respectively for the pitch angle ϕ, the cart position x and control input U in linear working conditions. Overall, all the controllers accomplish the regulation task stabilizing the system around $\phi = 0$ with small oscillations mainly due to physical limits of the experimental set-up. A comparative analysis of the controllers in terms of a) ET, and b) performance, is detailed as follows. Regarding the point a), Table 2 reports the code execution BCS and WCS values, which have been computed using the XMOS development environment xTimeComposer for each of the task executed by the board, including those related to the control algorithms. The WCS provides the values of the WCET for each task, this analysis gives the possibility of setting the temporal upper bound limit for RT or HRT applications. A first aspect to be noticed from Table 2, even if it is quite obvious, but here it has been quantified, is that for regulation problems LQR and MPC algorithms require less time to be executed. The SDRE requires instead almost four times the WCET in respect to the previous ones. This first result suggests that attention must be taken, in terms of ET, in using nonlinear algorithms, as the SDRE, for linear tasks. This becomes even more notable in higher order MIMO systems because the ET of the DARE solution is related to the system's order, as well as with the initial conditions. In classical applications where a single microcontroller must execute more tasks (e.g. sensors acquisition, control, etc.), a high WCET of the control algorithm could lead to critical situations. These temporal constraints are crucial in HRT applications. Differently, using the MET-SDRE controller previously introduced the algorithm ET decrease. In particular as long as the system is working within linear conditions the BCS value obtained can be comparable to the linear controllers and with respect to classic SDRE it decreases significantly. Regarding the point b), related to the performance analysis, it can be noticed that under linear conditions the LQR performs similar to the SDRE disregarding the ET, while the MET-SDRE keeps the same performance of SDRE as expected. For visualization reason, here the MET-SDRE performances, are not plotted because they match with the classic SDRE and they will be given for the second test. Figure 7 shows how the stabilization is reached for both the LQR controller and SDRE controller. However, focusing on Figure 8 it is possible to notice that, when using LQR controller, the cart position x is kept with a constant bias from the set-point. It happens because the LQR controller gives priority to the angle stabilization, indeed it has been demonstrated that in nonlinear systems when the state has coupling dynamics (as e.g. in our case where the ϕ and x behaviours are strictly related) a linear LQR cannot manage them perfectly [28]. However, some works sustain that in terms of performance SDRE approach provide at best a quite limited and case dependent, benefit over LQR [29]. Therefore LQR is often preferable due to its less computational effort especially in application where system's nonlinearities are negligible. Finally, the MPC takes more time to reach the steady-state condition with respect to the others and when reached is not able to well regulate the cart position x within 25 seconds. This may be because of determining the value of the control signal, the MPC analyzes the whole prediction horizon (N_p). As a result, the control effort is smoother but the system is also less reactive because the disturbance rejection task is averaged for the entire time horizon. Therefore the prediction horizon should not be too long, because it slows down the reaction of a controller to disturbances in the system, though this feature is interesting in some cases. Indeed, the control signal of MPC controller is smoother than the others (Fig. 9) and this may be important, especially in an industrial environment, during the start-up phases of plants and actuators, because these systems are adversely affected by sudden changes in the control signal. Moreover, this control strategy has less impact on energy consumption because the power required for a hypothetical actuator is less. Therefore, the prediction horizon N_p is an important tuning parameter that must be set case by case. Lastly, regarding trajectory tracking tasks, the MPC controller has advantages in terms of design and ET. This because the MPC control algorithm implementation is valid both for constant and variable system's reference signals. Therefore, the WCET of the MPC calculated in Table 2 for control regulation problem would remain the same also for the trajectory tracking problem. Differently, the LQR design phase in case of trajectory tracking issues is pursued [27] [39].

TABLE 2: Code execution BCS and WCS values computed for accomplished tasks

Algorithm	BCS	WCS
Encoder acquisition	$0.00331\,ms$	$0.00331\,ms$
Motor and PWM driver	$1.50025\,ms$	$2.00025\,ms$
Sonar acquisition	$1.58035\,ms$	$6.57300\,ms$
LQR	$1.90454\,ms$	$4.01384\,ms$
MPC	$2.06109\,ms$	$4.55033\,ms$
SDRE	$11.9710\,ms$	$12.5170\,ms$
MET-SDRE	$3.25336\,ms$	$8.15412\,ms$

2) Test 2 - nonlinear working conditions

Figures 10, 11, 12 show the trends over time of the pitch angle ϕ, the cart position x and the control input U. In this case, not all the controllers can accomplish the regulation task stabilizing the system around $\phi = 0$, for this reason in Figures 10, 11, 12 the divergent trends for the linear controllers are obtained. Also in this second test, a comparative analysis of the controllers in terms of a) ET, and b) performance, has been detailed. Regarding the point a) related to the ET values of the controllers, there are no changes from the previous test,

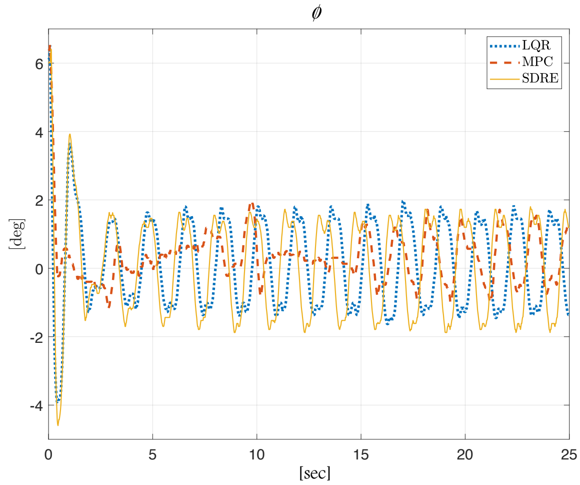

FIGURE 7: Experimental results of three optimal control strategies (LQR, MPC, SDRE) for the cart-pendulum in linear conditions; trend of pitch angle ϕ.

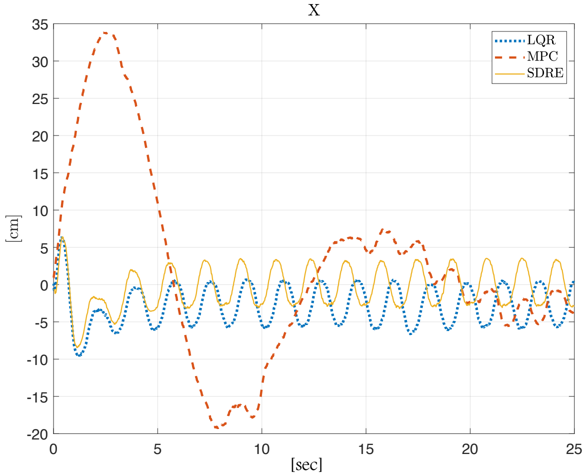

FIGURE 8: Experimental results of three optimal control strategies (LQR, MPC, SDRE) for the cart-pendulum in linear conditions; trend of cart position x.

but some insights regarding the MET-SDRE are given. For BCS and WCS values we always refer to 2. What should be pointed out is that the WCS value for the MET-SDRE has been obtained during this second test. Indeed, it is trivial to figure out that the Best Case Execution Time (BCET) is related to test no. 1, while the Worst Case Execution Time (WCET) is related to test no. 2. In the first case, under linear conditions (test no. 1), referring to the control performance, the MET-SDRE, being based on classic SDRE, can achieve superior performance (e.g. in terms of disturbance rejection and robustness) with respect to linear controllers such as LQR or MPC. At the same time, referring to the execution time (ET), the MET-SDRE has BCET of about $3\,ms$, similar to that of the LQR and MPC linear controllers, while the classic SDRE requires about $12\,ms$ under the same conditions, that means 4 times more. Secondly, in test no. 2 the MET-SDRE and the classic SDRE are proven to be the only control strategies able to stabilize the system while LQR and MPC lose control. At the same time, in the second scenario, the MET-SDRE decreases the algorithm's WCET of about 35% with respect to the classic SDRE (as in Table 2), by maintaining the same performance as well. This result is due to its adaptive nature, indeed, as discussed in section V, the number of backward iterations of the DARE (34) and then the ET of the control algorithm are based on the current state of the system. Then, when nonlinear behaviours arises the biggest difference between two consecutive SDC matrices $A_{d_{nl}}(x[k])$ and $A_{d_{nl}}(x[k+1])$ requires solving the DARE (34) with more iterations to obtain a convergent solution $\overline{P}(x[k])$. Regarding the point b) instead, which is related to the performance analysis, the system nonlinearities are triggered by imposing to the pendulum an angle $\phi = 11°$ out of the linear range. The linear controllers (LQR, MPC) are obviously not able to perform the regulation task. Indeed, in Figure 10 it can be noticed how the MPC tries to approach the reference value ($\phi = 0$) but the control law is not fast enough and the cart diverges from the position $x = 0$, as shown in Fig. 11. A similar result is obtained with the LQR even if the controller tries within 5 seconds to stabilize the pendulum but because of the large oscillations (Fig. 10) the control is lost. Differently, the SDRE and the MET-SDRE, within 2 seconds are able to bring the system near to the equilibrium point and stabilize the pendulum (with small oscillation around $\phi = 0$) in 5 seconds. The performance of these two nonlinear controllers are shown separately from the others to evaluate better the similarity of their behaviour. They are presented in Figures 13, 14, 15, where respectively are compared the

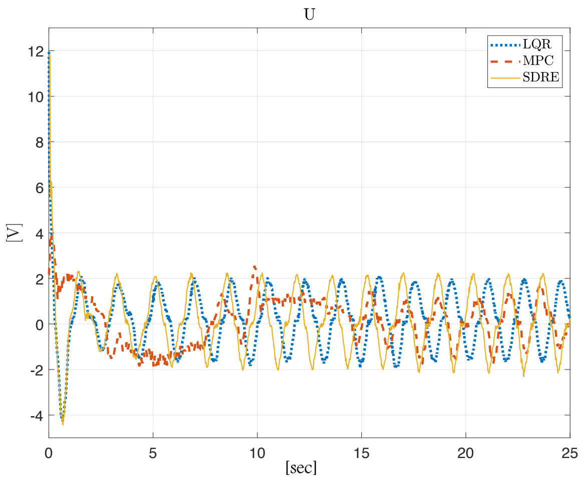

FIGURE 9: Experimental results of three optimal control strategies (LQR, MPC, SDRE) for the cart-pendulum in linear conditions; trend of control input U.

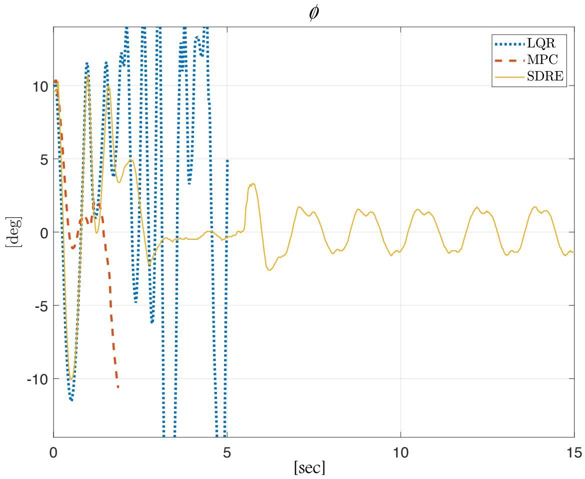

FIGURE 10: Experimental results of three optimal control strategies (LQR, MPC, SDRE) for the cart-pendulum in nonlinear conditions; trend of pitch angle ϕ.

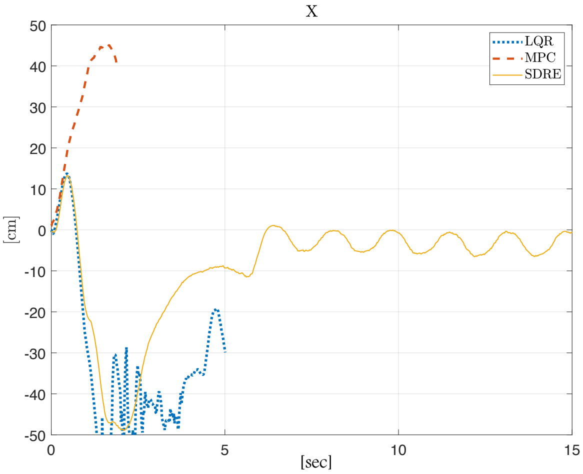

FIGURE 11: Experimental results of three optimal control strategies (LQR, MPC, SDRE) for the cart-pendulum in nonlinear conditions; trend of cart position x.

pitch angle ϕ, the cart position x and the control input U for the classic SDRE and the proposed MET-SDRE. It can be noticed in these figures, that the MET-SDRE achieves the same performance as classic SDRE implementation but decreases the algorithm's WCET. There is a small time-shift on the Figures 13, 14, 15 which depends on three factors: on the acquisition instant, on the small error which might be between the two initial angles ($\phi = 11°$) and on the possible reading lag introduced by the sonar. However, it is trivial to see how the MET-SDRE performances are consistent with its classic version. The performance analysis of SDRE, LQR and MPC instead, are shown in the Figures 10, 11, 12. Here, only the nonlinear controller (SDRE) is able to manage the increasing dynamics coupling of the system under nonlinear conditions. This is expected, indeed, when the systems dynamics become complex or the disturbances forces the system to work out of the linear range. In this case the MPC and LQR performance are overcome by the SDRE. Indeed, the SDRE taking into account system's nonlinearities can better handle the system dynamics as also experienced in [34] [41].

IX. CONCLUSION

The advancement in electronics, computing and communication technologies have made it feasible to extend the application of embedded systems to more critical applications, such as automotive, avionic and many others. Often, they involve Hard Real Time (HRT) requirements, where systems must be designed according to the resource adequacy policy and must provide sufficient computing resources to handle the specified worst-case computational load and fault scenario. The HRT embedded resources, as shown in this work, mostly are required by the control algorithms and the implementation of nonlinear control laws are often computationally demanding. For this reason in some cases, the performances are not suitable because of specification on application constraints. However, in order to achieve even more demanding objectives to control systems, the linear controllers have to be overcome because in most of control applications the PID is still the most used controller. In this work, a method has been provided to evaluate in a deterministic way the Execution Time required by the controllers' implementations for embedded devices with HRT characteristics. Implementation solutions for discrete-time linear and nonlinear optimal control techniques have been carried out with encouraging perspectives. Once evaluated both performance and ET for three of the most used optimal

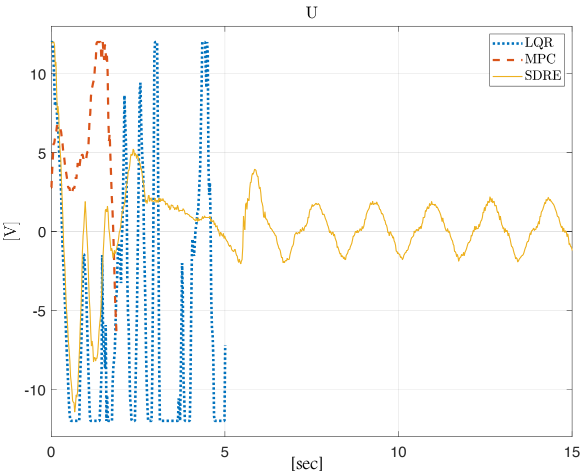

FIGURE 12: Experimental results of three optimal control strategies (LQR, MPC, SDRE) for the cart-pendulum in nonlinear conditions; trend of control input U.

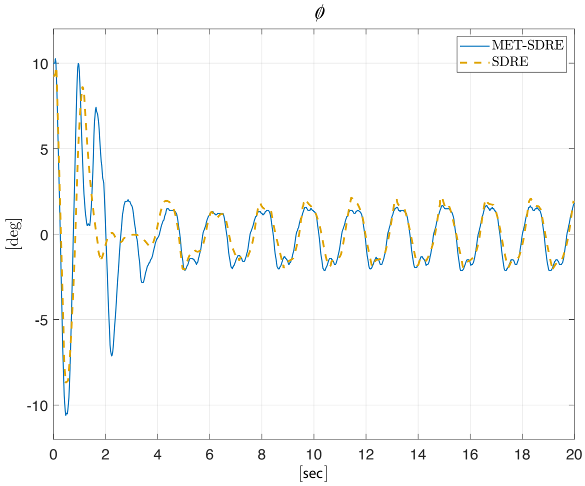

FIGURE 13: Experimental results of nonlinear optimal controllers (SDRE and MET-SDRE) for the cart-pendulum in nonlinear conditions; trend of pitch angle ϕ.

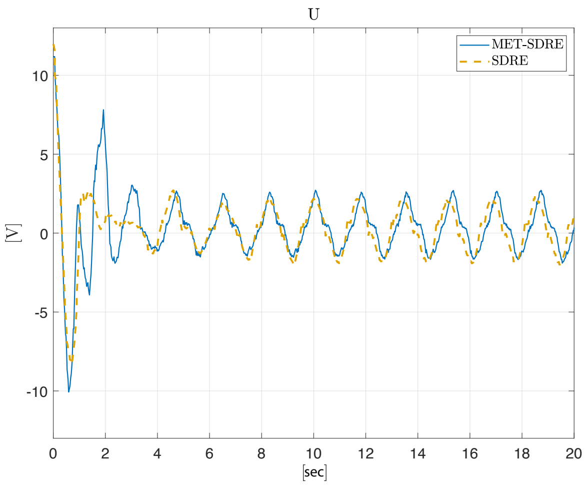

FIGURE 15: Experimental results of nonlinear optimal controllers (SDRE and MET-SDRE) for the cart-pendulum in nonlinear conditions; trend of control input U.

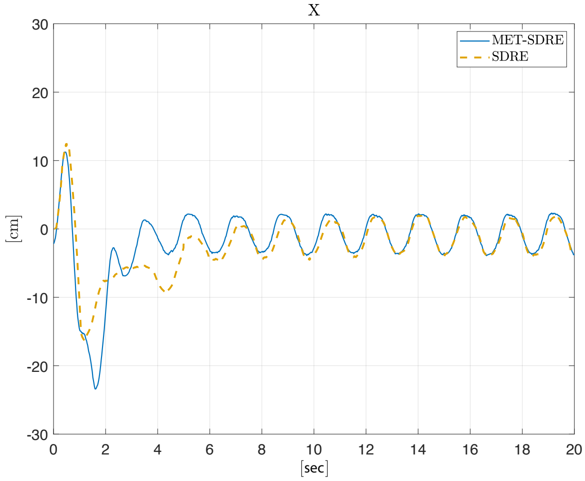

FIGURE 14: Experimental results of nonlinear optimal controllers (SDRE and MET-SDRE) for the cart-pendulum in nonlinear conditions; trend of cart position x.

controllers, this work shows how to improve the execution time of the SDRE nonlinear controller, by proposing a new method named MET-SDRE. With the MET-SDRE, in terms of ET, a Best Case Scenario (BCS) comparable with linear controllers have been achieved. The Worst Case Scenario (WCS) has been improved by 35% with respect to the common approach maintaining the same performance. In general, the results obtained, could be useful in the filed of control systems engineering especially during the design stage in RT and HRT systems, where temporal bounds have to be fulfilled also taking into account all the other functional specifications. In future, further investigation is required in order to prove the performance of these solutions on higher-order systems.

.

APPENDIX A SDC FACTORIZATION FOR SDRE DESIGN

In this appendix, the design flexibility for the SDC matrix is discussed, and general recommendations are provided to derive a correct factorization. The extended linearization, also known as SDC factorization is the process of factorizing a nonlinear system into a linear-like structure which contains SDC matrices. Considering the cart-pendulum model in (2), this can be generally written as:

$$\dot{x}(t) = f(x) + B(x)u(t) \tag{38}$$

where $f(x) = A(x)x$. Without loss of generality, the origin $x = 0$ can be assumed as an equilibrium point, such that $f(0) = 0$. Under this assumption, a continuous nonlinear matrix-valued function $A(x)$ always exists such that $f(x) = A(x)x$, where $A(x)$ is a $n \times n$ matrix found by mathematical factorization and it is nonunique when $n > 1$. Because of this non-uniqueness of the $A(x)$ matrix, in literature different works have been proposed methods to improve the system stability performances through the factorization process. For example, in [37] a method has been proposed to design the SDC matrix when working with a conforming system. In [36], it has been studied how a different factorization of the $A(x)$ matrix can affect stability performances. Another possible SDC parameterization solution is given in [34] and [41], where in both cases an inverted pendulum has been controlled with SDRE for both the swinging-up and stabilization. However, in this work, the focus is not to investigate a new parameterization method but to carry out a comparative analysis of control algorithms execution time, in particular LQR, MPC and SDRE. Consequently, a classic approach regarding the design flexibility has been followed, as proposed in [37]. The parameterization steps adopted to derive the SDC matrix in (3) are presented below with related motivations. Considering a nonlinear systems and under the assumption of $x = 0$ being an equilibrium point. The SDC $A(x)$ matrix terms can be state-independent or state-dependent. In case of state-independent terms, named also "bias", it can be handled to satisfy the assumption $f(0) = A(0)\,x(0) = 0$ by augmenting the system with a stable state:

$$z(\dot{t}) = -\lambda\, z(t) \tag{39}$$

so that the bias can be factorized as:

$$b(t) = \frac{b(t)\, z}{z} \tag{40}$$

which converges to zero only when $z = 0$.

Differently, regarding state-dependent terms non-unique solutions can be adopted. Defining $A(x, \alpha)x$ as an infinite family of SDC parameterization, in general, terms which do not converge to zero as the state approaches zero, violates the fundamental condition $f(0) = 0$ outlined above. Like biases, these terms prevent a direct factorization of $f(x)$ into $A(x, \alpha)x$ but they can be handled as discussed above. However, it is more desirable to capture their state dependency in the proper element of the matrix $A(x, \alpha)$. For example, supposing to have a system with two state variables, where equations are given by:

$$\begin{aligned} \dot{x_1} &= \beta(x_1 + x_2) \\ \dot{x_2} &= cos(x_1) \end{aligned} \tag{41}$$

where β can be considered as a bias term, discussed above, the SDC matrix $A(x)$ has the form:

$$A(x) = \begin{bmatrix} \beta & \beta \\ cos(x_1) & 0 \end{bmatrix} \tag{42}$$

The cosine term of the $A(x, \alpha)$ matrix it is desirable to be nonzero to reflect the state dependency. A solution could be to re-arrange the cosine function as:

$$cos(x_1) = [cos(x_1) \;-\; 1] \;+\; 1 \tag{43}$$

where $[cos(x_1) \;-\; 1]$ approaches to the origin when x_1 goes to zero and can be factorized as:

$$cos(x_1) = [\frac{cos(x_1) \;-\; 1}{x_1}]\, x_1 \tag{44}$$

and the remaining term +1 of equation (43) can be handled like a bias.

APPENDIX B THE MOTOR MODEL

The four-wheeled robot is powered by four identical motors which provide the total force F acting on it. The first order model in Laplace domain has been derived by means of an identification process which has provided the following transfer function:

$$\frac{\Omega(s)}{E(s)} = \frac{K_p}{1 + \tau_p s} = \frac{1.7469}{1 + 0.70738s} \tag{45}$$

where $\Omega(s), E(s)$, are respectively the Laplace transform of the angular velocity and the excitation voltage of the motor, while K_p,τ_p are respectively the gain constant and the mechanical time constant of the motor. In the time domain we have:

$$\dot{\omega}(t) = -\frac{1}{\tau_p}\,\omega(t) + \frac{K_p}{\tau_p}\,E(t) \tag{46}$$

Knowing that $\dot{\omega} = \frac{\tau_m}{J} = \frac{F\,R_r}{J}$ and $\omega = \frac{\dot{x}}{R_r}$ the (46) can be rewritten w.r.t. the force acting on the robot as:

$$F = \frac{(J\,K_p)}{(\tau_p\,R_r)}E \;-\; \frac{J}{\tau_p\,R_r^2}\dot{x} \tag{47}$$

Then, the force equation is replaced in the system model (1) where the term $\beta = \frac{(J\,K_p)}{(\tau_p\,R_r)}$, and $\gamma = \frac{J}{\tau_p\,R_r^2}$. The parameters K_p and τ_p are found through the identification process, R_r is the wheel radius. The rotor inertia J is calculated by adopting the reduced order model of the DC motor, where the electrical time constant is neglected. Since K_e and R_a are measured, and K_t is computed as a good approximation as $\frac{\tau_{stall}}{I_{stall}}$, it is possible to compute the viscous friction B_m with:

$$K_p = \frac{K_t}{K_t\,K_e +\,R_a\,B_m} \tag{48}$$

Then J can be computed with:

$$\tau_p = \frac{R_a\,J}{K_t\,K_e +\,R_a\,B_m} \tag{49}$$

the physical and electrical parameters are completely defined and listed in table 3.

TABLE 3: Physical and electrical parameters of the motor HN-GH35GMA

Parameter	Value	Description
τ_{stall}[N m]	0.44	Stall torque
I_{stall}[A]	1.4	Stall current
K_t[Nm/A]	$\frac{\tau_{stall}}{I_{stall}} = 0.3152$	Torque constant
K_e[V sec /rad]	0.4090	Back electromotive force constant
R_a[Ω]	7.49	Armature resistence
B_m[$N\,rad/\,s$]	0.007	viscous friction
J[km^2]	0.0018	Inertia rotor
E[V]		Excitation voltage

APPENDIX C MPC EXTENDED MODEL DERIVATION

In order to derive the extended model let consider the following extended state vector x_e:

$$\boldsymbol{x}_e[k] = [\,\Delta\boldsymbol{x}_s[k]^T\;\boldsymbol{y}[k]^T\,]^T \tag{50}$$

Using the iteration of the model (16) and by considering that $\boldsymbol{x}_s[k] = A_d\,\boldsymbol{x}_s[k-1] + B_d\,u[k-1]$, the following relationships can be easily obtained:

$$\Delta\boldsymbol{x}_s[k+1] = A_d\Delta\boldsymbol{x}_s[k] + B_d\Delta u[k] \tag{51}$$

$$\begin{aligned} \boldsymbol{y}[k{+}1]{-}\boldsymbol{y}[k] = C(\boldsymbol{x}_s[k{+}1]{-}\boldsymbol{x}_s[k]) = C\Delta\boldsymbol{x}_s[k{+}1] = \\ = CA_d\Delta\boldsymbol{x}_s[k] + CB_d\Delta u[k] \end{aligned} \tag{52}$$

equations (51) and (52) can be combined as below:

$$\begin{bmatrix} \Delta\, x_s[k{+}1] \\ y[k{+}1] \end{bmatrix} = \begin{bmatrix} A_d & o_m^T \\ CA_d & I_{qxq} \end{bmatrix} \begin{bmatrix} \Delta\, x_s[k] \\ y[k] \end{bmatrix} + \begin{bmatrix} B_d \\ CB_d \end{bmatrix} \Delta\, u \tag{53}$$

where for simplicity, it has been rewritten matrices as follow:

$$\begin{aligned} A_e &= \begin{bmatrix} A_d & o_m^T \\ CA_d & I_{qxq} \end{bmatrix} \quad B_e = \begin{bmatrix} B_d \\ CB_d \end{bmatrix} \\ C_e &= \begin{bmatrix} o_m & I_{qxq} \end{bmatrix} \end{aligned} \tag{54}$$

with o_m a $q\,\mathrm{x}\,dim(\boldsymbol{x}_s)$ zero matrix and q the number of the system outputs, i.e. the number of rows of y. Thus, equation (53) can be written in a compact form, and adding the equation to extract the $y[k]$ component, yields the extended system:

$$\begin{aligned} \boldsymbol{x}_e[k+1] &= A_e\, \boldsymbol{x}_e[k] + B_e\, \Delta u[k] \\ \boldsymbol{y}[k] &= C_e\, \boldsymbol{x}_e[k] \end{aligned} \tag{55}$$

Considering the extended model (50) Let us assume a sampling instant k^{th} and define the predicted output vector $\boldsymbol{Y}$ and the vector of the input trajectory $\Delta\boldsymbol{U}$ as follows:

$$\boldsymbol{Y} = \quad [\boldsymbol{y}[k+1|k]^T, \quad \boldsymbol{y}[k+2|k]^T, ..., \quad \boldsymbol{y}[k+N_p|k]^T\ ]^T \tag{56}$$

$$\Delta\boldsymbol{U} = \quad [\Delta u[k], \quad \Delta u[k+1],\ ..., \quad \Delta u[k+N_c-1]]^T \tag{57}$$

where the variable N_p is the length of the prediction window and N_c is the length of the receding window with $N_c \leq N_p$. It can be proven by using the model (54) iteratively that the output vector $\boldsymbol{Y}$ can be expressed in compact form as:

$$\boldsymbol{Y} = F\,\boldsymbol{x}_e[k] + \Phi\Delta\boldsymbol{U} \tag{58}$$

where

$$F = \begin{bmatrix} CA_e \\ CA_e{}^2 \\ CA_e{}^3 \\ ... \\ CA_e{}^{N_p} \end{bmatrix} \tag{59}$$

$$\Phi = \begin{bmatrix} CB_e & 0 & ... & 0 \\ CA_eB_e & CB_e & ... & 0 \\ CA_e{}^2B_e & CA_eB_e & ... & 0 \\ ... & ... & ... & 0 \\ CA_e{}^{N_p-1}B_e & CA_e{}^{N_p-2}B_e & ... & CA_e{}^{N_p-N_c}B_e \end{bmatrix} \tag{60}$$

For a given set-point signal $r(k_i)$ at sample time k_i, within a prediction horizon the objective of the predictive control system is to bring the predicted output as close as possible to the set-point signal, where we assume that the set-point signal remains constant in the optimization window. This objective is then translated into a design to find the 'best' control parameter vector ΔU such that an error function between the set-point and the predicted output is minimized. Assuming the data-vector which contains the set-point information as:

$$\boldsymbol{R}_s = \overbrace{[1\,1\,..\,1]}^{N_p}\ r[k_i]. \tag{61}$$

The cost function J that reflects the control objective is defined as:

$$J = (\boldsymbol{R}_s - \boldsymbol{Y})^T(\boldsymbol{R}_s - \boldsymbol{Y}) + \Delta\boldsymbol{U}^T R \Delta\boldsymbol{U} \tag{62}$$

where the first term is related to the minimization between the predictive output and the set-point, while the second is referred to the size of $\Delta\boldsymbol{U}$.

REFERENCES

[1] R. Obermaisser, (2004). Event-triggered and time-triggered control paradigms In Springer Science and Business Media, Vol. 22.

[2] A. Bonci, S. Longhi and G. A. Scala, (2020). "Towards an All-Wheel Drive Motorcycle: Dynamic Modeling and Simulation". In IEEE Access, vol. 8, Issue 1, pp. 112867-112882.

[3] A. Bonci, R. De Amicis, S. Longhi, E. Lorenzoni, G. A. Scala, (2016). "A motorcycle enhanced model for active safety devices in intelligent transport systems". In Proceedings of the 12th IEEE/ASME International Conference on Mechatronic and Embedded Systems and Applications (MESA), Auckland, New Zeland, Aug. 29-31, pp. 1-6.

[4] K. G. Shin and P. Ramanathan, (1994). "Real-time computing: A new discipline of computer science and engineering" In Proceedings of the IEEE.

[5] J. A. Stankovic, (1988). "Misconceptions about real-time computing: A serious problem for next-generation systems." In Computer. 21(10):10–19.

[6] M. Bohlin, Y. Lu, J. Kraft, P. Kreuger, and T. Nolte, (2009). "Simulation-based timing analysis of complex real-time systems". In In Embedded and Real-Time Computing Systems and Applications.

[7] G. Martins, D. Lacey, A. Moses, M. J. Rutherford, and K. P Valavanis, (2012). "A case for i/o response benchmarking of microprocessors.". In IECON, IEEE, 38th Annual Conference on IEEE Industrial Electronics Society, pp. 3018–3023.

[8] G. Martins, A. Moses, M. J. Rutherford, and K. P Valavanis, (2012). "Enabling intelligent unmanned vehicles through xmos technology.". In The Journal of Defense Modeling and Simulation, 9(1):71–82.

[9] P.L. Garoche, (2019). "Formal verification of control system soft-ware". In Princeton University Press Princeton, NJ, USA.

[10] I. D. Landau, and G. Zito, (2007). "Digital control systems: design, identification and implementation." In Springer Science and Business Media.

[11] A. Farmanbordar, N. Zaeri, and S. Rahimi, (2011). "Stabilizing a Driven Pendulum using DLQR Control". In Fifth Asia Modelling Symposium, Academic Press, New York, 3rd edition.

[12] M. N. Zeilinger, C. N. Jones, D. M. Raimondo, and M. Morari, (2009). "Real-time MPC-Stability through robust MPC design.". In Proceedings of the 48h IEEE Conference on Decision and Control , IEEE, pp. 3980-3986.

[13] S.C. Beeler, , H. T. Tran, (2000). "Feedback control methodologies for nonlinear systems". In Journal of optimization theory and applications.

[14] R.I. Davis, L. Cucu-Grosjean, M. Bertogna and A. Burns, (2016). "A review of priority assignment in real-time systems". In Journal of systems architecture.

[15] J. Hansen, S. A. Hissam, and G. A. Moreno, (2009). "Statistical-based wcet estimation and validation". In Proceedings of the 9th Intl. Workshop on Worst-Case Execution Time (WCET) Analysis.

[16] M. Liu, M. Behnam, and T. Nolte, (2013). "An evt-based worst-case response time analysis of complex real-time systems". In Industrial Embedded Systems (SIES), 8th IEEE International Symposium, pp. 249–258.

[17] Y. Lu, (2012). "Pragmatic approaches for timing analysis of real-time embedded systems". PhD thesis, Mälardalen University.

[18] C. Maxim, A. Gogonel, D. Maxim, and L. Cucu, (2013). "Estimation of probabilistic minimum inter-arrival times using extreme value theory".

[19] D. Maxim, F. Soboczenski, I. Bate, and Eduardo Tovar, (2015). "Study of the reliability of statistical timing analysis for real-time systems.". In Proceedings of the 23rd International Conference on Real Time and Networks Systems. pp 55-64.

[20] K. S. Refaat and P. E. Hladik, (2010). "Efficient stochastic analysis of realtime systems via random sampling". In Real- Time Systems (ECRTS). 22nd Euromicro Conference on, pp.175–183.

[21] B. Tanasa, U. D. Bordoloi, P. Eles, and Z. Peng, (2015). "Probabilistic response time and joint analysis of periodic tasks.". In Real-Time Systems, 27th Euromicro Conference IEEE, pp. 235–246.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.3019776, IEEE Access

[22] T. Ungerer, C. Bradatsch, M. Frieb, F. Kluge, J. Mische, A. Stegmeier, R. Jahr, M. Gerdes, P. Zaykov, L. Matusova, et al. (2016). "Parallelizing industrial hard real-time applications for the parmerasa multicore". In ACM Transactions on Embedded Computing Systems (TECS).

[23] R. Wilhelm, J. Engblom, A. Ermedahl, N. Holsti, S. Thesing, D. Whalley, G. Bernat, C. Ferdinand, R. Heckmann, T. Mitra, et al. (2008). "The worst-case execution-time problem—overview of methods and survey of tools". In ACM Transactions on Embedded Computing Systems, (TECS).

[24] K. M. Deliparaschos, K. Michail and A. Zolotas. (2017). "On the issue of LQG embedded control realization in a Maglev system". In 2017 25th Mediterranean Conference on Control and Automation (MED), pp. 1379-1384.

[25] A. Jezierski, J. Mozaryn and D. Suski, (2017). "A comparison of LQR and MPC control algorithms of an inverted pendulum". In Polish Control Conference, pp. 65-76.

[26] A. Bogdanov, (2004). "Optimal control of a double inverted pendulum on a cart.". Oregon Health and Science University, Tech. Rep. OGI School of Science and Engineering, Beaverton, OR.

[27] S. Dani,D. Sonawane, D. Ingole and S. Patil, (2017). "Performance evaluation of PID, LQR and MPC for DC motor speed control". In nd International Conference for Convergence in Technology (I2CT), pp. 348-354.

[28] P. Dang and L. Frank, (2005). "Controller for swing-up and balance of single inverted pendulum using SDRE-based solution.". In 31st Annual Conference of IEEE Industrial Electronics Society

[29] L. Felicetti and G.B. Palmerini, (2013). "A comparison among classical and SDRE techniques in formation flying orbital control". In IEEE Aerospace Conference. pp. 1-12.

[30] A. Bonci, M. Pirani, and S. Longhi, (2017). "An embedded database technology perspective in cyber-physical production systems". In Proceedings of the 27th International Conference on Flexible Automation and Intelligent Manufacturing, FAIM2017, (available on Procedia Manufacturing, Volume 11, Issue 1, 2017), Modena, Italy, pp. 830-837.

[31] A. Bonci, M. Pirani, A. Cucchiarelli, A. Carbonari, B. Naticchia, S. Longhi, (2018). "A Review of Recursive Holarchies for Viable Systems in CPSs" In IEEE 16th International Conference on Industrial Informatics(INDIN 2018), Porto, Portugal, pp. 37-42.

[32] A. Bonci, S. Longhi, G. Nabissi and F. Verdini, (2019). "Predictive Maintenance System using motor current signal analysis for Industrial Robot". In 2019 24th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA), (pp. 1453-1456).

[33] J. R. Cloutier and C. P. Mracek, (1997). "Parametric enhancement of state-dependent riccati equation based control". In Proceedings of the 1997 American Control Conference, Vol. 2, pp. 1072-1073.

[34] C. Huifeng, L. Hongxing, and Y. Peipei, (2009). "Swinging-up and stabilization of the inverted pendulum by energy well and SDRE control". In 2009 Chinese Control and Decision Conference, pp. 2222-2226.

[35] A. Khamis and D. Naidu, (2013). " Nonlinear optimal tracking using finitehorizon state dependent Riccati equation (SDRE)". In Proceedings of the 4th International Conference on Circuits, Systems, Control, Signals (WSEAS), pp. 37-42.

[36] S. Kim and S. Kwon, (2017). "Nonlinear optimal control design for underactuated two-wheeled inverted pendulum mobile platform". In IEEE/ASME Transactions on Mechatronics, 22(6), 2803-2808.

[37] T. Çimen, (2008). "State-dependent Riccati equation (SDRE) control: A survey." In IFAC Proceedings Volumes. pp. 3761-3775.

[38] Y. W. Liang and L. G. Lin, (2013). "Analysis of SDC matrices for successfully implementing the SDRE scheme". In Automatica, 49(10), 3120-3124.

[39] E. V. Kumar, and J. Jerome, (2013). "Robust LQR controller design for stabilizing and trajectory tracking of inverted pendulum". In Procedia Engineering. pp. 169-178.

[40] M. W. Trosset, (2000). "Distance matrix completion by numerical optimization." In Computational Optimization and Applications, 17(1), 11-22.

[41] P. Dang and F. L. Lewis, (2005). "Controller for swing-up and balance of single inverted pendulum using SDRE-based solution". In 31st Annual Conference of IEEE Industrial Electronics Society, IECON.

A. BONCI (M'15) holds the position of Assistant Professor in automatic control and heads the Automation Laboratory at Università Politecnica delle Marche. His research interests are modeling and control of dynamic systems, system and control theory, vehicle dynamics, automotive systems, modeling and control of autonomous systems, mechatronic, robotics, embedded systems applications, industrial automation, cyber-physical systems, smart manufacturing.

S. LONGHI (M'92-SM'04) holds the positions of Full Professor in Robot Technologies and of Rector at the Università Politecnica delle Marche. His main research interests include modeling, identification and control of linear and nonlinear systems, control of mobile robots, service robots for assistive applications supporting mobility and cognitive actions, home and building automation, automatic fault detection and isolation.

G. NABISSI received the master's degree (with honours) in Computer and Automation Engineering from Università Politecnica delle Marche, in 2019. Currently, he is a Ph.D student and he holds a research fellow contract for the Regional platform for Industry 4.0. His research of interest is focused on Industry 4.0 related issues. In particular, he works on predictive maintenance applied to electrical motor-operated devices and on robotics concerning in Modelling, Planning and Control.

G. A. SCALA received the Master degree in Computer and Automation Engineering from the Università Politecnica delle Marche, in 2018. At present he is attending the PhD in automation engineering from the Dipartimento di Ingegneria dell'Informazione. His main research interests focus on robotics, nonlinear system analysis and control, virtual simulations and computer science.