DOI: 10.48011/asba.v2i1.1523

Explicit Model Predictive Control for Inverted Pendulum Systems ⋆

Mateus Mussi Brugnolli, Bruno Augusto Angélico *

* *Department of Telecommunications and Control Engineering*
Escola Politécnica da USP, São Paulo, Brazil.
(e-mails: mateus_mmb@usp.br, angelico@lac.usp.br*)*

Abstract: Model Predictive Control is a control technique that has been greatly investigated in recent years. It has the versatility of different types of models for the prediction of the system and aptitude to handle the system constraints. In the last decade, the multi-parametric optimization has been applied to the control theory that allowed for the MPC optimization to be performed offline, which was denominated as explicit Model Predictive Control. This work investigates the application of this control technique in Inverted Pendulum systems, which are commonly used as didactic control systems. The complete control design is described considering its validation for two Inverted Pendulum systems through simulations.

Keywords: Model Predictive Control, Explicit solution, Multi-parametric optimization, Inverted Pendulum, Control Systems.

1. INTRODUCTION

Inverted pendulum systems are vastly used as didactic control tools (Sugaya et al. (2017)). This classic control problem relies on mechanical actuation to induce a torque sufficient to maintain a stem on its inverted position. This system has several mechanical configurations and adaptability on degrees of freedom. Therefore, a vast set of control theories can be applied to this multi-variable system, from the classic control, modern control, to the most advanced and recent controllers (Jmel et al. (2020)).

This work highlights the application of the Model Predictive Control (MPC) theory on inverted pendulum systems. This technique had its origins in the petrochemical industries in the '70s and appealed to many researchers. Nowadays, this theory has several configurations and flexibilities, for example, the usage of different sets of models as linear, hybrid, nonlinear, deterministic, or stochastic models (Qin and Badgwell (2003); Lenz et al. (2015)).

The key concept of the MPC is to optimize the system performance through the prediction of the system model in a closed-loop. Generally, the control signal is represented as discrete-time state feedback. Accordingly, the system states must be measured each time sampling to update the MPC optimization parameters (states). Additionally, the MPC theory includes states and control signal constraints for a feasibility guarantee. Thus, the optimization solution is heavily influenced by the system complexity. A system with a high dimension may demand high optimization solution time that exceeds the time sampling of the control system (Camacho and Alba (2013)).

⋆ This work was supported by Fundação de Amparo à Pesquisa do Estado de São Paulo (FAPESP), grant 2017/22130-4 and Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brazil (CAPES) - Finance Code 001.

Since the MPC with linear models results in a quadratic optimization problem, researchers started to apply the multi-parametric programming theory (Bemporad et al. (2002)). Briefly, in this approach a feasible parameter set is assembled, that is a representation of all possible states that the real system may reach. With this set, each optimization restriction creates a finite number of polytopes as a function of the parameters. Each region, that is related to a subset of parameters, can be interpreted as the set of active or inactive constraints. Therefore, the optimization can be solved offline and it designs the optimal MPC feedback control for each region. This approach has been known in the literature as the explicit MPC (eMPC) since the optimization solves the optimal control actions explicitly for each state subset (Borrelli et al. (2017)).

The implementation of the eMPC uses a look-up table. Iteratively, the states of the system are collected and the active region of the current states determines the respective MPC state feedback. Notice that if there is a considerable number of restrictions or if the system dimension is large enough, the multi-parametric optimization may result in a large number of regions. Hence, this work simulates the eMPC considering multiple system dimensions. The main contributions for this paper are the investigation of the eMPC for two inverted control systems and its validation through simulations.

This work is organized as follows: In Section 2, the modeling of two distinct inverted control systems are addressed, the Reaction Wheel Pendulum and the Rotary Inverted Pendulum (known as Furuta Pendulum). The next Section 3 follows with the specifications for the eMPC with its tuning parameters and system constraints. Section 4 develops the simulation setup using the nonlinear models and the discussion of the results. Lastly, Section 5 points out the conclusions and topics for future works.

2. INVERTED PENDULUM MODELING

This work considers two configurations that has been researched and built in the Laboratório de Controle Aplicado (LCA) (Angélico (2016)). The systems are the reaction wheel inverted pendulum (Block et al. (2007)) and the pendulum proposed by Katsuhisa Furuta, rotary inverted pendulum (Mori et al. (1976)).

2.1 Reaction Wheel Pendulum

A representation of the reaction wheel pendulum is shown in Figure 1. A stem that represents the pendulum rotates freely on the axis #1. In the opposite side of the pendulum, a DC motor on the stem. A second body, named as reaction wheel, is fixed and rotates on the axis #0 of the motor axis. The only actuator of the system is the DC motor, which directly applies a torque into the reaction wheel. In this case, the angular position of axes #0 and #1, α and θ_1, and their respective velocities are considered measured.

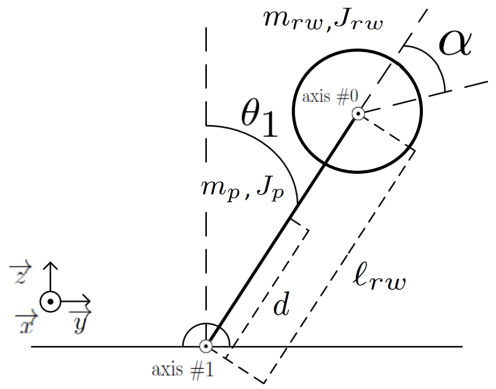

Figure 1. Illustration of the reaction wheel pendulum.

The control system objective is the application of a torque into the reaction wheel that results in a reaction torque on the pendulum that maintains it in the inverted position ($\theta_1 = 0$). The control input is the Pulse Width Modulation (PWM) signal applied to the DC motor. By the definition, the signal ranges between 1 (applying the maximum average voltage) and 0 (null voltage). Additionally, negative values imply that the motor rotates in its inverted nominal rotation. It is considered the Euler-Lagrange modeling method, with E_i as the kinematic energy and P_i as the potential energy of body i. This system was divided into two bodies: the reaction wheel rw and the pendulum p.

The potential energy of the reaction wheel P_{rw} is the displacement of the Center of Mass (CoM) of the reaction wheel body in the vertical plane and for the pendulum P_p is the displacement of its CoM in the same plane.

$$P_{rw} = m_{rw} g d \cos(\theta_1) \tag{1}$$

$$P_p = m_p g d \cos(\theta_1) \tag{2}$$

The kinematic energy of the pendulum E_p is related to the torque applied on the stem. For the reaction wheel, E_{rw} has three distinct components. First, the torque applied to the reaction wheel with both angular velocities. Additionally, there are vertical and horizontal translation displacements of the CoM of the reaction wheel.

$$\begin{aligned} E_{rw} = &\frac{1}{2}(\dot{\theta}_1 + \dot{\alpha}) J_{rw} (\dot{\theta}_1 + \dot{\alpha}) \\ &+ \frac{1}{2}(\dot{\alpha} \ell_{rw} \cos(\theta_1)) m_{rw} (\dot{\alpha} \ell_{rw} \cos(\theta_1)) \\ &+ \frac{1}{2}(\dot{\alpha} \ell_{rw} \sin(\theta_1)) m_{rw} (\dot{\alpha} \ell_{rw} \sin(\theta_1)) \end{aligned} \tag{3}$$

$$E_p = \frac{1}{2} \dot{\alpha} J_p \dot{\alpha} \tag{4}$$

Using the complete Lagrangian function L of Equation (5), Equation (6) is the Euler-Lagrange equation to obtain the nonlinear models for the reaction wheel and the pendulum. The generalized forces Q_i for each body are presented in Equations (7,8). For the reaction wheel, the control signal is the PWM signal for the 6 Volts DC motor and there is viscous damping in axis #0. For the pendulum, there is viscous damping in axis #1.

$$L = E_{rw} + E_p - P_{rw} - P_p \tag{5}$$

$$\frac{\mathrm{d}}{\mathrm{d}t}\left(\frac{\partial L}{\partial \dot{\delta}}\right) - \frac{\partial L}{\partial \delta} = Q_i; \text{for } \delta = \alpha, \theta_1 \tag{6}$$

$$Q_{rw} = \frac{K_t}{R}(6\,PWM - K_e \dot{\alpha}) - b_0 \dot{\alpha} \tag{7}$$

$$Q_p = -b_1 \dot{\theta}_1 \tag{8}$$

Table 1 shows the parameters of the reaction wheel pendulum built in the LCA.

Table 1. Parameters of the reaction wheel.

	Parameter	Value
m_{rw}	Reaction Wheel Mass $[Kg]$	0.144
g	Gravity Acceleration $[m/s^2]$	9.81
d	Distance of CoM of rw to axis #1 $[m]$	0.0987
m_p	Pendulum Mass $[Kg]$	0.149
J_{rw}	Inertia of the Reaction Wheel $[Kgm^2]$	9.456×10^{-4}
ℓ_{rw}	Distance of rw to axis #1 $[m]$	0.14298
J_p	Inertia of the Pendulum $[Kg]$	6.2533×10^{-4}
K_t	Motor Torque Constant $[Nm/A]$	0.307
R	Armature Resistance $[\Omega]$	5.325
K_e	Back EMF Constant $[Vs/rad]$	0.273
b_0	Viscous damping of axis #0 $[Nm/(rad/s]$	1.0×10^{-5}
b_1	Viscous damping of axis #1 $[Nm/(rad/s]$	1.0×10^{-6}

The nonlinear equations are linearized applying the jacobian with the state vector $x_{rw} = \begin{bmatrix}\delta\theta_1 & \delta\dot{\theta}_1 & \delta\dot{\alpha}\end{bmatrix}^T$, where δ represents the linearization close to a operation point. The chosen point was $\begin{bmatrix}\theta_1^R & \dot{\theta}_1^R & \dot{\alpha}^R\end{bmatrix}^T = [0\ 0\ 0]^T$. Note that the dimension of the system is three, since the reaction wheel angle α is not controllable in this configuration. Aiming the simulations using the MPC, that is a discrete-time control, a ZOH discretization is applied to the state-space, using $f_s = 50$ Hz. The discrete linear model for the control is shown in Equation (9).

$$x_{rw}(k+1) = A_{rw} x_{rw}(k) + B_{rw} PWM(k) \tag{9}$$

$$A_{rw} = \begin{bmatrix} 1.0189 & 0.0201 & 0.0008 \\ 1.8777 & 1.0189 & 0.0724 \\ -1.5951 & -0.0170 & 0.6563 \end{bmatrix}; B_{rw} = \begin{bmatrix} -0.0170 \\ -1.5908 \\ 7.5526 \end{bmatrix}$$

The objective of the control system is to regulate all states to zero since it is the pendulum in the inverted position. Also, note that the angular position of the reaction wheel

does not impact on the system stability, and the actuation on the system is limited from -1 to 1 (negative values of PWM represent opposite rotation). Because of the mechanical configuration, the reaction torque can not stabilize the pendulum at any other position other than the inverted position and the standby position.

2.2 Furuta Pendulum

The Figure 2 is a representation of the Furuta Pendulum. There is a DC motor accoupled to a fixed structure with a beam fixed on axis #0. This beam is referred as the body arm. There is also a stem that rotates around axis #2 on the other edge of the arm. This stem is referred as the body Furuta Pendulum f. The objective of this system is to apply a torque to rotate the arm and stabilize the pendulum in the inverted position ($\theta_2 = 0$).

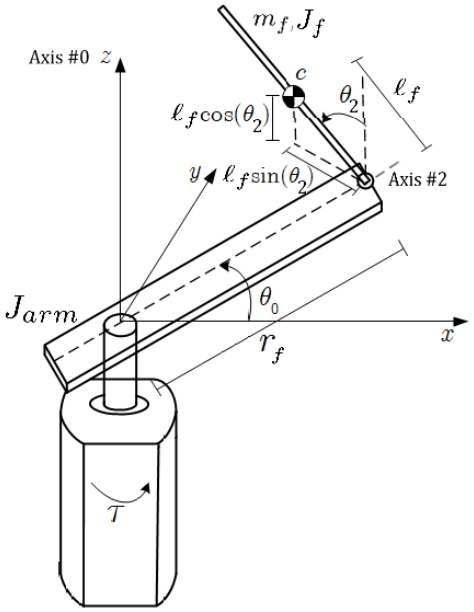

Figure 2. Illustration of the Furuta pendulum.

There are two bodies for this system: arm and pendulum f. The potential energy of the arm P_{arm} is equal to zero because of the CoM of the arm maintains in the horizontal plane. The potential energy of the pendulum P_f is the displacement of the CoM in the vertical plane.

$$P_{arm} = 0 \tag{10}$$

$$P_f = m_f g \ell_f \cos(\theta_2) \tag{11}$$

The kinematic energy of the arm E_{arm} is obtained from the torque of its rotation. The energy E_f for the pendulum can be described using its rotation, the velocity of the CoM in the directions of x, y, and z (Mori et al. (1976)).

$$E_{arm} = \frac{1}{2}(\dot{\theta}_0) J_{arm} (\dot{\theta}_0) \tag{12}$$

$$E_f = \frac{1}{2}(\dot{\theta}_2) J_f (\dot{\theta}_2) + \frac{m_f}{2}\left(\ell_f^2 \dot{\theta}_2^2 + r_f^2 \dot{\theta}_0^2 + \ldots + \ell_f^2 \dot{\theta}_0^2 \sin^2(\theta_2) + 2 r_f \ell_f \dot{\theta}_0 \dot{\theta}_2 \cos(\theta_2)\right) \tag{13}$$

With the Lagrangian function for the Furuta Pendulum in Equation (14), the nonlinear equations of this system are solved with Equation (15). The generalized forces Q_i for each body are defined in Equations (16,17). For the arm, the control signal is a PWM for the 12 Volts DC motor and there is viscous damping in axis #0. Additionally, there is a viscous damping in axis #2. Table 2 shows the parameters of the Furuta pendulum built in the LCA.

$$L = E_{arm} + E_f - P_{arm} - P_f \tag{14}$$

$$\frac{\mathrm{d}}{\mathrm{d}t}\left(\frac{\partial L}{\partial \dot{\delta}}\right) - \frac{\partial L}{\partial \delta} = Q_i; \text{for } \delta = \theta_0, \theta_2 \tag{15}$$

$$Q_{arm} = \frac{K_t}{R}\left(12\,PWM - K_e \dot{\theta}_0\right) - b_{arm}\dot{\theta}_0 \tag{16}$$

$$Q_f = -b_f \dot{\theta}_2 \tag{17}$$

Table 2. Parameters of the Furuta Pendulum.

	Parameter	Value
m_f	Pendulum Mass $[Kg]$	0.076
g	Gravity Acceleration $[m/s^2]$	9.81
ℓ_f	Distance of CoM of pendulum to axis #2 $[m]$	0.125
J_{arm}	Inertia of the arm $[Kgm^2]$	0.0046
J_f	Inertia of the Pendulum $[Kg]$	0.00039
r_f	Distance of pendulum to axis #0 $[m]$	0.25
K_t	Motor Torque Constant $[Nm/A]$	0.118
R	Armature Resistance $[\Omega]$	3.6
K_e	Back EMF Constant $[Vs/rad]$	0.118
b_{arm}	Viscous damping of axis #0 $[Nm/(rad/s]$	3.0×10^{-5}
b_f	Viscous damping of axis #2 $[Nm/(rad/s]$	5.0×10^{-6}

Afterwards, the nonlinear equations are linearized applying the jacobian with the state vector

$$x_f = \begin{bmatrix}\delta\theta_0 & \delta\theta_2 & \delta\dot{\theta}_0 & \delta\dot{\theta}_2\end{bmatrix}^T,$$

where δ represents the linearization close to a operation point. It was linearized for

$$\begin{bmatrix}\theta_0^R & \theta_2^R & \dot{\theta}_0^R & \dot{\theta}_2^R\end{bmatrix}^T = [0\ 0\ 0\ 0]^T.$$

Note that this system has dimension four since the angular positions of the arm, of the pendulum, and their respective velocities are controllable in this configuration. Applying a ZOH discretization in the state-space, the discrete model for the control is shown in Equation (18). This control system is able to track a reference signal for the arm position while stabilizing the pendulum on its inverted position.

$$x_f(k+1) = A_f x_f(k) + B_f PWM(k) \tag{18}$$

$$A_f = \begin{bmatrix}1.000 & -0.005 & 0.020 & 0.000\\ 0.000 & 1.019 & 0.001 & 0.020\\ 0.000 & -0.486 & 0.987 & -0.005\\ 0.000 & 1.914 & 0.020 & 1.019\end{bmatrix}; B_f = \begin{bmatrix}0.014\\ -0.021\\ 1.362\\ -2.051\end{bmatrix}$$

3. EXPLICIT MODEL PREDICTIVE CONTROL

The MPC is a discrete-time controller that uses the model to predict the optimal control sequence. For both systems, the control input is $u(k) = PWM(k)$. Consider the cost function of Equation (19).

$$J(k) = \sum_{j=0}^{N-1} (x(k+j))^T Q(x(k+j)) + (u(k+j))^T R(u(k+j)) \tag{19}$$

The element $x(k+j)$ is the prediction of the states after j samples, considering the linear model presented in Equation (9) for Reaction Wheel system and Equation (18) for Furuta Pendulum. Note that the measurement of $x(k)$ is a requirement for the model prediction. The control sequence $u(k+j)$ is the optimization vector with the structure $u(k+j) = F(j)x(k+j)+g_j(j)$. The element $g_j(j)$ is related to the input saturation. If $F(j)x(k+j)$ would saturate the control signal, $g_j(j)$ for the prediction j provides the saturated value for $u(k+j)$ to respect all input restrictions. For this approach, consider that N is finite and represents the prediction horizon. Therefore, there are N control signals for optimization for each input. The elements Q and R are semi-definite and definite matrices, respectively, for tuning the system performance. Now, observe the last term of $J(k)$, for $j = N-1$:

$$J(k)_{j=N-1} = (x(k+N-1))^T Q(x(k+N-1)) + (u(k+N-1))^T R(u(k+N-1)) \quad (20)$$

For $J(k)$ to be bounded, the term $J(k)_{j=N-1}$ should be as small as possible, representing that at the end of the prediction the system should be close to its desired states. For this bound certification, it is usual to include a terminal penalty at the end of prediction, as in Equation (21). The matrix P is called a terminal penalty and it is commonly computed using the Ricatti equation $P = A^T PA + Q$, where A is the state transition matrix and Q is the state weighting matrix.

$$J(k)_{MPC} = J(k) + (x(k+N))^T P(x(k+N)) \quad (21)$$

The MPC definition problem is stated in Lemma 1.

Lemma *1.* The MPC controller computes the following optimization (Camacho and Alba (2013)):

$$\min_{u(k+j)} \; J(k)_{MPC} \quad (22)$$

subject to

$$x(k+1) = Ax(k) + Bu(k) \quad (23)$$

$$x(k+j) \in \mathcal{X} \quad (24)$$

$$u(k+j) \in \mathcal{U} \quad (25)$$

for all $j = 0, \dots, N-1$.

The Equation (23) is the constraint for the model prediction. $\mathcal{X}$ is the parameters set that contains all feasible values for the states and $\mathcal{U}$ is the set of all input signals that satisfies its restrictions, for example, the input saturations.

For each time sampling, the controller in Lemma 1 obtains the present state $x(k)$ and solves the optimization problem. Hence, consider that the sets $\mathcal{X}$, $\mathcal{U}$, and the system model (Equation (23)) are known. Since all these restrictions are linear, they can be described as $g(u(k+j), x(k)) \leq 0$. Also, note that $u(k+j)$ is a function of $x(k)$ because of the feedback structure $u(k+j) = K(j)x(k+j)+g_j(j)$ and the discrete linear model. Therefore, the cost function at the instant k can be described as $J_{MPC} = J(u(k+j), x(k))$, with u as the optimization vector and x the parameter vector. Finally, the Lemma 2 defines the eMPC problem.

Lemma *2.* The eMPC controller computes the following multi-parametric optimization (Borrelli et al. (2017)):

$$\min_{u(k+j)} \; J(u(k+j), x(k)) \quad (26)$$

subject to

$$g(u(k+j), x(k)) \leq 0 \quad (27)$$

for all $x(k) \in \mathcal{X}$, $u(k+j) \in \mathcal{U}$, and $j = 0, \dots, N-1$. The restrictions of Equation (27) form a polytope with a finite number of partitions (regions). For each region, there is an optimal $u^*(k+j)$ vector that minimizes $J(u(k+j), x(k))$.

Since Lemma 2 considers the sets $\mathcal{X}$ and $\mathcal{U}$, this can be optimized offline and its solution has a structure of a look-up table. The implementation of the eMPC does not require any online optimization. At each time sampling, the state $x(k)$ checks which region of Equation (27) is active through the process of branch search. After detecting the active region, its respective $u^*(k+j)$ vector should be applied to the system. The current control action ($j = 0$) is chosen as the control signal and the controller holds for the next time sampling to update the states and repeat the process.

This control setup has been programmed in the Multi-Parametric Toolbox (MPT) for Matlab (Herceg et al. (2013)). This toolbox has a complete set of functions for parametric optimization, computational geometry, and especially a framework for MPC design. The MPT become a standard environment for eMPC since it has compatibility with YALMIP (Löfberg (2004)) for MPC design and algorithms to solve multi-parametric optimizations.

The MPT setup for MPC design has a list of requirements. First, a Linear Time-Invariant (LTI) System object selects a discrete linear model. Next, the set $\mathcal{X}$ is defined by inserting the upper and lower bounds of each state or declaring the polyhedra of constraints. The set $\mathcal{U}$ has similar requirements for the control input restrictions.

The matrices Q and R are included in the framework as quadratic functions for the states and control signals, respectively. One remark is that it is possible to use one norm or infinity norm functions for MPC design. Additionally, there is a function that computes the terminal penalty P using the usual LQR Penalty algorithm (Riccati Equation). Finally, the MPT designs the MPC similar to Lemma 1. Furthermore, there is a function to explicit the controller, applying Lemma 2.

4. SIMULATION SETUP AND DISCUSSIONS

The software Simulink was chosen for the environment setup. Each system was simulated using the nonlinear equations of Section 2 and the eMPC has the objective to control the system using the discrete linear model. The reaction wheel pendulum stabilizes the system on its inverted position $\theta_1 = 0$, subject to initial conditions and disturbances. The Furuta pendulum stabilizes the pendulum on its inverted position $\theta_2 = 0$ and tracks a reference signal for *arm* angular position $\theta_0 \rightarrow \theta_0^R$, also subject to initial conditions and disturbances.

Since output measurements are usually noisy, a white noise signal was added for each output signal. The white noise signal had variance of 5×10^{-6} for angular measurements and 3×10^{-5} for angular velocities. Each noise signal was generated with a distinct seed to minimize correlations between the signals.

Firstly, the parameter set $\mathcal{X}$ for the reaction wheel was defined. After few simulations without viscous damping and noise signals, the maximum value for θ_1 was 25°. Considering that the DC motor has a symmetric operation, the minimum value was $\theta_1^{min} = 25^\circ$ (0.43 rad). Also, $\dot{\theta}_1^{max} = -\dot{\theta}_1^{min} = 5\,rad/s; \dot{\alpha}^{max} = -\dot{\alpha}^{min} = 25\,rad/s$. The tuning parameters for the reaction wheel pendulum were:

$$Q = \begin{bmatrix} 25 & 0 & 0 \\ 0 & 10 & 0 \\ 0 & 0 & 1 \end{bmatrix}; R = 1.$$

Next, the parameter set $\mathcal{X}$ for the Furuta pendulum was defined similarly to the previous case. The *arm* angle position was considered as a full rotation $\theta_0^{max} = -\theta_0^{min} = 180^\circ (\pi\,rad)$. The maximum angle for the pendulum was $\theta_2^{max} = -\theta_2^{min} = 15^\circ$ (0.26 rad). The maximum angle velocities were

$$\dot{\theta}_0^{max} = -\dot{\theta}_0^{min} = \dot{\theta}_2^{max} = -\dot{\theta}_2^{min} = 5\,rad/s.$$

The tuning parameters for the Furuta pendulum were:

$$Q = \begin{bmatrix} 3000 & 0 & 0 & 0 \\ 0 & 50000 & 0 & 0 \\ 0 & 0 & 300 & 0 \\ 0 & 0 & 0 & 500 \end{bmatrix}; R = 1.$$

The control set $\mathcal{U}$ was the same for both systems since its signal was modularized. Therefore, the maximum value was 1 and the minimum value was -1. Both systems had a sampling frequency of $f_s = 50\,Hz$. The initial conditions were all zeros except for each pendulum position, with $\theta_1(0) = \theta_2(0) = 10^\circ$.

The last parameter of the eMPC is the prediction horizon N. For the reaction wheel, which has dimension 3, few configurations were investigated. With $N = 1$, it had 3 regions; $N = 2$ had 5 regions, $N = 3$ had 7 regions, up to $N = 30$ with 61 regions. However, since the eMPC only applies the first control signal ($j = 0$), it has been observed that all cases for this system had the same performance. This distinct behavior has a direct relation with the sets $\mathcal{X}$ and $\mathcal{U}$. For example, if a restrictive control signal would be applied, the set $\mathcal{U}$ would be smaller and a larger prediction horizon could improve the system performance. Since that it was not the case for this system, the prediction horizon $N = 1$ was chosen in the following simulations.

For the reaction wheel pendulum, Figure 3 shows the pendulum position subject to an external disturbance. The explicit MPC was able to stabilize the system with a initial condition and noisy measurements. The controller sustained with impulse disturbances and a small step signal in $t = 6\,s$.

The Figures 4 and 5 shows the angular velocities and the control signal computed by the eMPC. Note that the control signal had brief saturated impulses.

For the Furuta Pendulum, which has dimension 4, some configurations were investigated. $N = 1$ had 11 regions, $N = 3$ had 296 regions, up to $N = 5$ with 2108 regions. Note that the increase in system dimension strongly impacts the number of regions. For this system, all configurations also had the same performance, thus $N = 1$ was

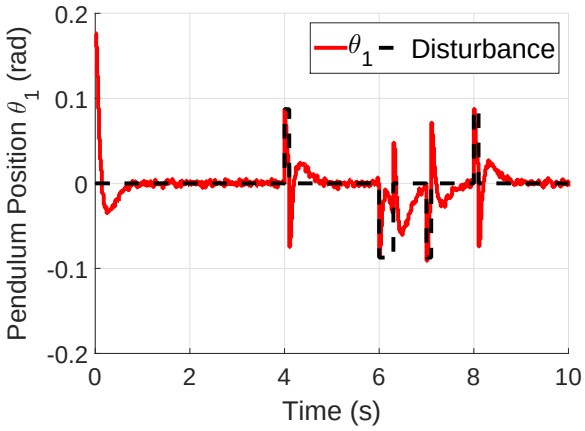

Figure 3. Reaction Wheel: Pendulum Position and Disturbance.

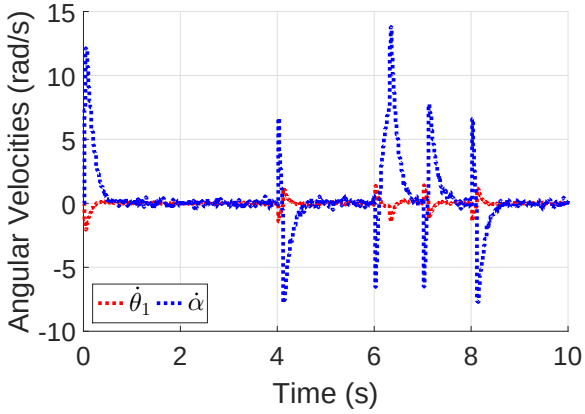

Figure 4. Reaction Wheel: Angular velocities.

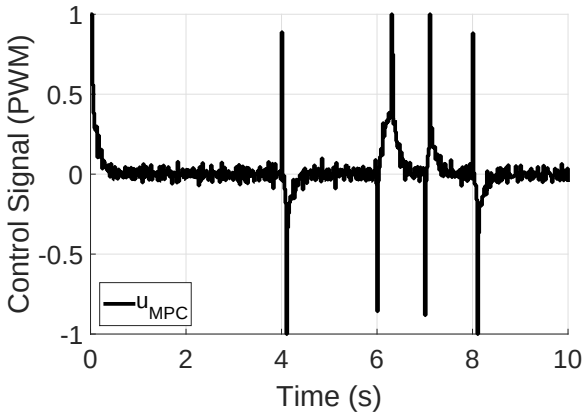

Figure 5. Reaction Wheel: Control Signal.

chosen for simulations. Figure 6 shows the arm position tracking the reference signal. In Figure 7, the pendulum position is stabilized considering the external disturbances.

Lastly, Figure 8 shows the angular velocites of the system and Figure 9 shows the control signal through the simulation.

5. CONCLUSIONS

This work investigated the application of the eMPC technique in two inverted pendulum control systems. The for-

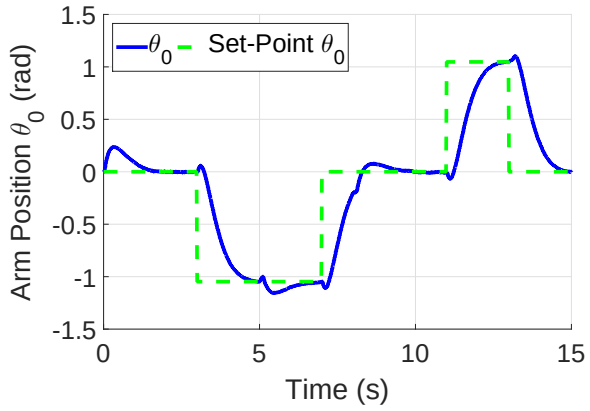

Figure 6. Furuta: Arm Position and Reference Tracking.

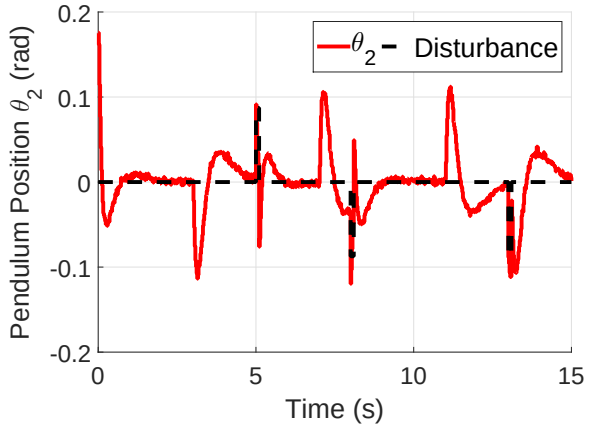

Figure 7. Furuta: Pendulum Position and Disturbance.

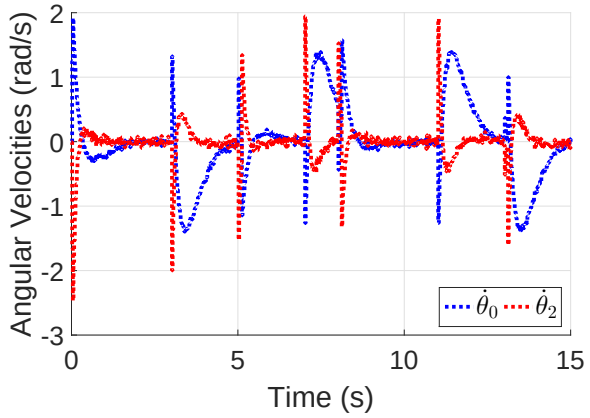

Figure 8. Furuta: Angular velocities.

mulation of the parameter set $\mathcal{X}$ and the $\mathcal{U}$ is a crucial step for eMPC design since unreachable states do not need to be considered for the control system, reducing the number of constraints regions. Additionally, the small increase in the system dimension greatly affected the number of regions. However, the increase of the prediction horizon N did not improve the control performance for these control systems.

REFERENCES

Angélico, B.A. (2016). Website of lca (laboratório de controle aplicado). `https://sites.usp.br/lca/`.

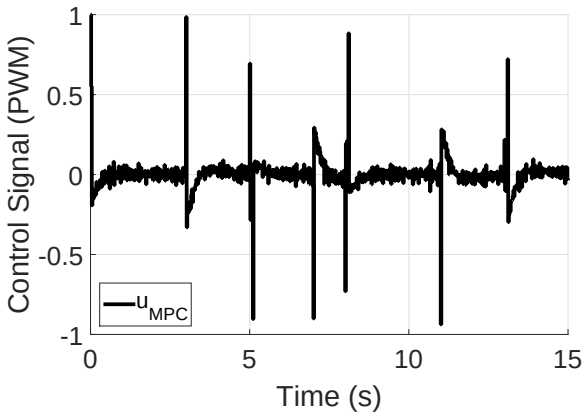

Figure 9. Furuta: Control Signal.

Bemporad, A., Morari, M., Dua, V., and Pistikopoulos, E.N. (2002). The explicit linear quadratic regulator for constrained systems. *Automatica*, 38(1), 3 – 20. doi: https://doi.org/10.1016/S0005-1098(01)00174-1.

Block, D.J., Astrom, K.J., and Spong, M.W. (2007). *The Reaction Wheel Pendulum*.

Borrelli, F., Bemporad, A., and Morari, M. (2017). *Predictive Control for Linear and Hybrid Systems*. Cambridge University Press. doi:10.1017/9781139061759.

Camacho, E.F. and Alba, C.B. (2013). *Model predictive control*. Springer Science & Business Media.

Herceg, M., Kvasnica, M., Jones, C., and Morari, M. (2013). Multi-Parametric Toolbox 3.0. In *Proc. of the European Control Conference*, 502–510. Zürich, Switzerland. `http://control.ee.ethz.ch/~mpt`.

Jmel, I., Dimassi, H., Hadj-Said, S., and M'Sahli, F. (2020). An adaptive sliding mode observer for inverted pendulum under mass variation and disturbances with experimental validation. *ISA Transactions*. doi:https://doi.org/10.1016/j.isatra.2020.02.029.

Lenz, D., Kessler, T., and Knoll, A. (2015). Stochastic model predictive controller with chance constraints for comfortable and safe driving behavior of autonomous vehicles. In *2015 IEEE Intelligent Vehicles Symposium (IV)*, 292–297.

Löfberg, J. (2004). Yalmip : A toolbox for modeling and optimization in matlab. In *In Proceedings of the CACSD Conference*. Taipei, Taiwan.

Mori, S., Nishihara, H., and Furuta, K. (1976). Control of unstable mechanical system control of pendulum†. *International Journal of Control*, 23(5), 673–692. doi: 10.1080/00207177608922192.

Qin, S.J. and Badgwell, T.A. (2003). A survey of industrial model predictive control technology. *Control Engineering Practice*, 11(7), 733 – 764. doi:https://doi.org/10.1016/S0967-0661(02)00186-7.

Sugaya, J., Ohba, Y., and Kanmachi, T. (2017). Simulation of standing upright control of an inverted pendulum using inertia rotor and the swing type inverted pendulum for engineering education. In *2017 9th International Conference on Information Technology and Electrical Engineering (ICITEE)*, 1–6.