POLYTECHNIC OF BARI

DEPARTMENT OF ELECTRICAL AND INFORMATION ENGINEERING

AUTOMATION ENGINEERING MASTER'S DEGREE

Course on MODEL PREDICTIVE CONTROL

Prof. Engr. Saverio **MASCOLO**
Prof. Engr. Vito Andrea **RACANELLI**

Project work:

Furuta Pendulum

Students:
D'ALESSANDRO Vito Ivano
VENEZIA Antonio

ACADEMIC YEAR 2019-2020

Abstract

This paper deals with certain options on controlling an rotary inverted pendulum also known as Furuta pendulum. Simulation and control of a Furuta pendulum using non-linear MPC (Model Predictive Control) are presented. Mathematical model used for the controller was obtained through the Lagrange-Euler formulation and its physical parameters were extracted from a mechanical CAD model. Finally, using non-linear MPC on Matlab, the increments applied to the control signal were calculated by minimizing a cost function considering constraints on the manipulated variables and the controlled variables, rejecting unmeasurable disturbance even if the presence of sensor noise. The results show a great robustness of the control strategy implemented since it is able to keep the system stable.

Contents

1 Introduction

Among the under-actuated systems, the Furuta Pendulum, also known as rotary inverted pendulum, is a classic example of non-linear controller. More in depth, it is a mechanism that has two degrees of freedom (DOF) and two revolute joints. It is essentially composed by three elements: a motor and two bars called arm and pendulum. The motor's shaft is connected to one end of the arm, which allows the arm to rotate in the horizontal plane, whereas the pendulum is joined to the free end of the arm through a revolute joint, allowing the rotation of the pendulum in the vertical plane. The objective of Furuta pendulum is to keep both pendulum angle and arm velocity null.

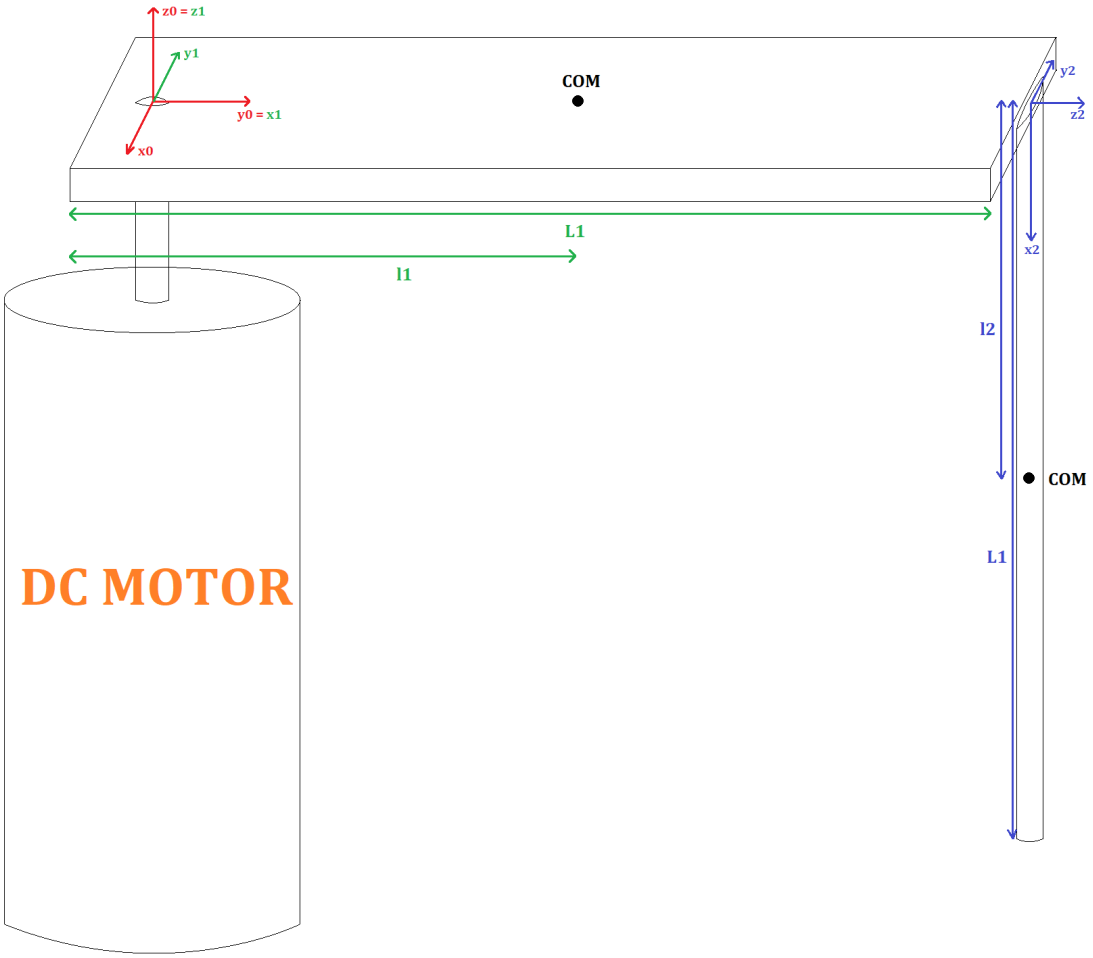


Figure 1.1: Schematic of rotary inverted pendulum

In Fig. 1.1 rotary inverted pendulum is shown. In particular, it has been defined frame attached to the arm and to the pendulum that will be used to determine the lagrangian equations of the rotary inverted pendulum.

2 Equations of motion

2.1 Definitions

Consider the rotational inverted pendulum mounted to a DC motor as shown in Fig.1.1. The DC motor is used to apply a torque τ_1 to Arm 1. The link between Arm 1 and Arm 2 is not actuated but free to rotate. The two arms have lengths L1 and L2. The arms have masses m1 and m2 which are located at l1 and l2, respectively, which are the lengths from the point of rotation of the arm to its center of mass. The arms have inertia tensors J1 and J2 (about the centre of mass of the arm). Each rotational joint is viscously damped with damping coefficients b1 and b2, where b1 is the damping provided by the motor bearings, and b2 is the damping arising from the pin coupling between Arm 1 and Arm 2. A right hand coordinate system has been used to define the inputs, states, and the Cartesian coordinate systems 1 and 2. The coordinate axes of Arm 1 and Arm 2 are the principal axes, such that the inertia tensors are diagonal of the form

$$J_1 = \begin{bmatrix} J_{1xx} & 0 & 0 \\ 0 & J_{1yy} & 0 \\ 0 & 0 & J_{1zz} \end{bmatrix} J_2 = \begin{bmatrix} J_{2xx} & 0 & 0 \\ 0 & J_{2yy} & 0 \\ 0 & 0 & J_{2zz} \end{bmatrix}$$

The angular rotation of Arm 1, θ_0, is measured in the horizontal plane where a counterclockwise direction is positive. The angular rotation of Arm 2, θ_1 is measured in the vertical plane where a counterclockwise direction is positive: in particular, when the Arm 2 is hanging down in the stable equilibrium position $\theta_1 = 0$ while in the unstable equilibrium position $\theta_1 = 180\,\text{deg}$.
The torque the servomotor applies to Arm 1,τ_1, is positive in a counterclockwise direction while the disturbance torque applied to Arm 2,τ_2 where a counterclockwise direction is positive.

2.2 Assumptions

Before deriving the dynamics of the system, a number of assumptions must be made:

- the motor shaft and Arm 1 are assumed to be rigidly coupled and infinitely stiff;
- Arm 2 is assumed to be infinitely stiff;
- the coordinate axes of Arm 1 and Arm 2 are the principal axes such that the inertia tensors are diagonal;
- the motor rotor inertia is assumed to be negligible. However, this term may be easily added to the moment of inertia of Arm 1;
- only viscous damping is considered. All other forms of damping (such as Coulomb) have been neglecte.

2.3 Lagrangian Formulation

In order to compute rotary inverted pendulum's equations of motions, the lagrangian formulation is used. First of all, the rotation matrix from the base frame to the frame attached to Arm 1 is

$$R_1 = \begin{bmatrix} \cos(\theta_0) & \sin(\theta_0) & 0 \\ -\sin(\theta_0) & \cos(\theta_0) & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{1}$$

The rotation matrix from Arm 1 to Arm 2 is given by

$$R_2 = \begin{bmatrix} 0 & \sin(\theta_1) & -\cos(\theta_1) \\ 0 & \cos(\theta_1) & \sin(\theta_1) \\ 1 & 0 & 0 \end{bmatrix} \tag{2}$$

The angular velocity of Arm 1 is given by

$$\omega_1 = \begin{bmatrix} 0 & 0 & \dot{\theta_0} \end{bmatrix}^T \tag{3}$$

The linear velocity of Arm 1 is 0, so

$$v_1 = \begin{bmatrix} 0 & 0 & 0 \end{bmatrix}^T \tag{4}$$

The velocity of the center of mass of Arm 1 is given by

$$v_1 = v_1 + \omega_0 \times \begin{bmatrix} l_1 & 0 & 0 \end{bmatrix}^T = \begin{bmatrix} 0 & \dot{\theta_0} \cdot l_1 & 0 \end{bmatrix}^T \tag{5}$$

The angular velocity of Arm 2 is given by

$$\omega_1 = R_2\omega_0 + \begin{bmatrix} 0 & 0 & \dot{\theta_1} \end{bmatrix}^T = \begin{bmatrix} -\cos(\theta_1)\dot{\theta_0} \\ \sin(\theta_1)\dot{\theta_0} \\ \dot{\theta_1} \end{bmatrix} \tag{6}$$

The linear velocity of Arm 2 respect to the reference frame is given by

$$v_2 = R_2(\omega_0 \times \begin{bmatrix} L_1 & 0 & 0 \end{bmatrix}^T) = \begin{bmatrix} \dot{\theta_0} L_1 \sin(\theta_1) \\ \dot{\theta_0} L_1 \cos(\theta_1) \\ 0 \end{bmatrix} \tag{7}$$

The velocity of the center of mass of Arm 2 is given by

$$v_{2_c} = v_2 + \omega_1 \times \begin{bmatrix} l_2 & 0 & 0 \end{bmatrix}^T) = \begin{bmatrix} \dot{\theta_0} L_1 \sin(\theta_1) \\ \dot{\theta_0} L_1 \cos(\theta_1) + \dot{\theta_1} l_2 \\ -\dot{\theta_0} l_2 \sin(\theta_1) \end{bmatrix} \tag{8}$$

The potential energy of Arm 1 is

$$E_{p_1} = 0 \tag{9}$$

and the kinetic energy is

$$E_{k_1} = \frac{1}{2}(v_{1_c}^T m_1 v_{1_c} + \omega_0^T J_1 \omega_0) = \frac{1}{2}\dot{\theta_0}^2(m_1 l_1^2 + J_{1_z z}) \tag{10}$$

The potential energy of Arm 2 is

$$E_{p_2} = g m_2 l_2 (1 - \cos(\theta_1)) \tag{11}$$

and the kinetic energy is

$$\begin{aligned} E_{k_2} &= \frac{1}{2}(v_{2_c}^T m_2 v_{2_c} + \omega_1^T J_2 \omega_1) = \\ &= \frac{1}{2}\dot{\theta_0}^2(m_2 L_2^2 + (m_2 l_2^2 + J_{2_y y})\sin(\theta_1)^2 + J_{2_x x}\cos(\theta_1)^2) + \frac{1}{2}\dot{\theta_1}^2(J_{2_z z} + m_2 l_2^2) \\ &+ m_2 L_1 l_2 \cos(\theta_1)\dot{\theta_0}\dot{\theta_1} \end{aligned} \tag{12}$$

So, the total potential and kinetic energies are given by

$$\begin{aligned} E_p &= E_{p_1} + E_{p_2} \\ E_k &= E_{k_1} + E_{k_2} \end{aligned} \tag{13}$$

The Lagrangian is the difference between kinetic and potential energies, so

$$L = E_k - E_p \tag{14}$$

After having computed the Lagrangian, the equations of motions are given by

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{q}_i}\right) + b_i\dot{q}_i - \frac{\partial L}{\partial q_i} = Q_i \tag{15}$$

where

- $q_i = [\theta_0, \theta_1]^T$ is the generalised coordinate;
- $b_i = [b_1, b_2]^T$ is a generalised viscous damping coefficient;
- $Q_i = [\tau_1, \tau_2]$ is the generalised torque.

Evaluating the terms in Eq.15 gives

$$\begin{aligned}
\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{\theta_0}}\right) &= \ddot{\theta_0}(J_{1_{zz}} + m_1 l_1^2 + m_2 L_1^2 + (m_2 l_2^2 + J_{2_{yy}})\sin(\theta_1)^2 + J_{2_{xx}}\cos(\theta_1)^2) \\
&\quad + m_2 L_1 l_2 \cos(\theta_1)\ddot{\theta_1} - m2L_1 l_2 \sin(\theta_1)\dot{(\theta_1)}^2 + \dot{\theta_0}\dot{\theta_1}\sin(2\theta_1)(m2l_2^2 + J_{2_{yy}} - J_{2_{zz}}) \\
\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{\theta_1}}\right) &= \ddot{\theta_0} m_2 L_1 l_2 \cos(\theta_1) + \ddot{\theta_1}(J_{2_{zz}} + m_2 l_2^2) - \dot{\theta_0}\dot{\theta_1} m_2 L_1 l_2 \sin(\theta_1) \\
-\frac{\partial L}{\partial \dot{\theta_0}} &= 0 \\
-\frac{\partial L}{\partial \dot{\theta_1}} &= -\frac{1}{2}\dot{\theta_0} sin(2\theta_1) = (m_2 l_2^2 + J_{2_{yy}} - J_{2_{xx}}) + \dot{\theta_0}\dot{\theta_1} m_2 L_1 l_2 \sin(\theta_1) + g m_2 l_2 \sin(\theta_1)
\end{aligned} \tag{16}$$

Substituting the previous terms into the Euler-Lagrange equation, the following differential equations are obtained:

$$\begin{bmatrix}
\ddot{\theta_0}(J_{2_{zz}} + m_1 l_1^2 + m_2 L_1^2 + (J_{2_{yy}} + m_2 l_2^2)\sin(\theta_1)^2 + J_{2_{xx}}\cos(\theta_1)^2) + \\
\ddot{\theta_1} m_2 L_1 l_2 \cos(\theta_1) - m_2 L_1 l_2 \sin(\theta_1)\dot{\theta_1}^2 + \dot{\theta_0}\dot{\theta_1}\sin(2\theta_1)(m_2 l_2^2 + J_{2_{yy}} - J_{2_{xx}}) + b_1\dot{\theta_0} \\
\ddot{\theta_0} m_2 L_1 l_2 \cos(\theta_1) + \ddot{\theta_1}(m_2 l_2^2 + J_{2_{zz}}) + \\
\frac{1}{2}\dot{\theta_0}^2 \sin(2\theta_1)(-m_2 l_2^2 - J_{2_{yy}} + J_{2_{xx}}) + b_2\dot{\theta_1} + g m_2 l_2 \sin(\theta_1)
\end{bmatrix} = \begin{bmatrix} \tau_1 \\ \tau_2 \end{bmatrix} \tag{17}$$

2.4 Simplifications

Most Furuta pendulums tend to have long slender arms, such that the moment of inertia along the axis of the arms is negligible. In addition, most arms have rotational symmetry, such that the moments of inertia in two of the principal axes are equal. Thus, the inertia tensors may be approximated as:

$$J_1 = \begin{bmatrix} J_{1_{xx}} & 0 & 0 \\ 0 & J_{1_{yy}} & 0 \\ 0 & 0 & J_{1_{zz}} \end{bmatrix} \simeq \begin{bmatrix} 0 & 0 & 0 \\ 0 & J_1 & 0 \\ 0 & 0 & J_1 \end{bmatrix}$$

$$J_2 = \begin{bmatrix} J_{2_{xx}} & 0 & 0 \\ 0 & J_{2_{yy}} & 0 \\ 0 & 0 & J_{2_{zz}} \end{bmatrix} \simeq \begin{bmatrix} 0 & 0 & 0 \\ 0 & J_2 & 0 \\ 0 & 0 & J_2 \end{bmatrix}$$

Because of these simplifications, the total moment of inertia of Arm 1 about the origin of attached frame is

$$\hat{J}_1 = J_1 + m_1 l_1^2$$

while the moment of inertia of Arm 2 about the origin of attached frame is

$$\hat{J}_2 = J_2 + m_2 l_2^2$$

Finally,the total moment of inertia the motor rotor experiences when Arm 2 is in its equilibrium position (downward position) is

$$\hat{J}_0 = \hat{J}_1 + m_2 L_1^2 = J_1 + m_1 l_1^2 + m_2 L_1^2$$

Substituting these results in Eq.17, equations of motions will be as follow:

$$\begin{bmatrix} \ddot{\theta_0}(\hat{J}_0 + \hat{J}_2 \sin(\theta_1)^2 + \ddot{\theta_1} m_2 L_1 l_2 \cos(\theta_1) - m_2 L_1 l_2 \sin(\theta_1) \dot{\theta_1}^2 + \dot{\theta_0}\dot{\theta_1} \hat{J}_2 \sin(2\theta_1) + b_1 \dot{\theta_0} \\ \ddot{\theta_0} m_2 L_1 l_2 \cos(\theta_1) + \ddot{\theta_1} \hat{J}_2 - \frac{1}{2} \dot{\theta_0}^2 \hat{J}_2 \sin(2\theta_1) + b_2 \dot{\theta_1} + g m_2 l_2 \sin(\theta_1) \end{bmatrix} = \begin{bmatrix} \tau_1 \\ \tau_2 \end{bmatrix} \tag{18}$$

2.5 State Space Equations

Equations of motions of Furuta pendulum can be solved in terms of the angular acceleration of Arm 1 and Arm 2, as given by [1]:

$$\ddot{\theta_0} = \frac{\begin{bmatrix} -\hat{J}_2 b_1 \\ m_2 L_1 l_2 \cos(\theta_1) b_2 \\ -\hat{J}_2^2 \sin(2\theta_1) \\ -\frac{1}{2} \hat{J}_2 m_2 L_1 l_2 \cos(\theta_1) \sin(2\theta_1) \\ \hat{J}_2 m_2 L_1 l_2 \sin(\theta_1) \end{bmatrix}^T \cdot \begin{bmatrix} \dot{\theta_0} \\ \dot{\theta_1} \\ \dot{\theta_0}\dot{\theta_1} \\ \dot{\theta_0}^2 \\ \dot{\theta_1}^2 \end{bmatrix} + \begin{bmatrix} \hat{J}_2 \\ -m_2 L_1 l_2 \cos(\theta_1) \\ \frac{1}{2} m_2^2 l_2^2 L_1 \sin(2\theta_1) \end{bmatrix}^T \cdot \begin{bmatrix} \tau_1 \\ \tau_2 \\ g \end{bmatrix}}{\hat{J}_0 \hat{J}_2 + \hat{J}_2^2 \sin(\theta_1) - m_2^2 L_1^2 l_2 \cos(\theta_1)^2} \tag{19}$$

$$\ddot{\theta_1} = \frac{\begin{bmatrix} m_2 L_1 l_2 \cos(\theta_1) b_1 \\ -b_2(\hat{J}_0 + \hat{J}_2 \sin(\theta_1)^2) \\ m_2 L_1 l_2 \hat{J}_2 \cos(\theta_1) \sin(2\theta_1) \\ -\frac{1}{2} \sin(2\theta_1)(\hat{J}_0 \hat{J}_2 + \hat{J}_2^2 \sin(\theta_1)^2) \\ -\frac{1}{2} m_2 L_1^2 l_2^2 \sin(2\theta_1) \end{bmatrix}^T \cdot \begin{bmatrix} \dot{\theta_0} \\ \dot{\theta_1} \\ \dot{\theta_0}\dot{\theta_1} \\ \dot{\theta_0}^2 \\ \dot{\theta_1}^2 \end{bmatrix} + \begin{bmatrix} -m_2 L_1 l_2 \cos(\theta_1) \\ \hat{J}_0 + \hat{J}_2 \sin(\theta_1)^2 \\ -m_2 l_2 \sin(\theta_1)(\hat{J}_0 + \hat{J}_2 \sin(\theta_1)^2) \end{bmatrix}^T \cdot \begin{bmatrix} \tau_1 \\ \tau_2 \\ g \end{bmatrix}}{\hat{J}_0 \hat{J}_2 + \hat{J}_2^2 \sin(\theta_1) - m_2^2 L_1^2 l_2 \cos(\theta_1)^2} \tag{20}$$

Consider that:

- $x_1 = \theta_0$;
- $x_2 = \dot{\theta_0}$;
- $x_3 = \theta_1$;
- $x_4 = \dot{\theta_1}$;

So, the state space model of Furuta pendulum is given by:

$$\begin{bmatrix} \dot{x_1} \\ \dot{x_2} \\ \dot{x_3} \\ \dot{x_4} \end{bmatrix} = \begin{bmatrix} x_2 \\ \ddot{\theta_0} \\ x_4 \\ \ddot{\theta_1} \end{bmatrix} \tag{21}$$

3 Furuta Pendulum Mechanical Parameters

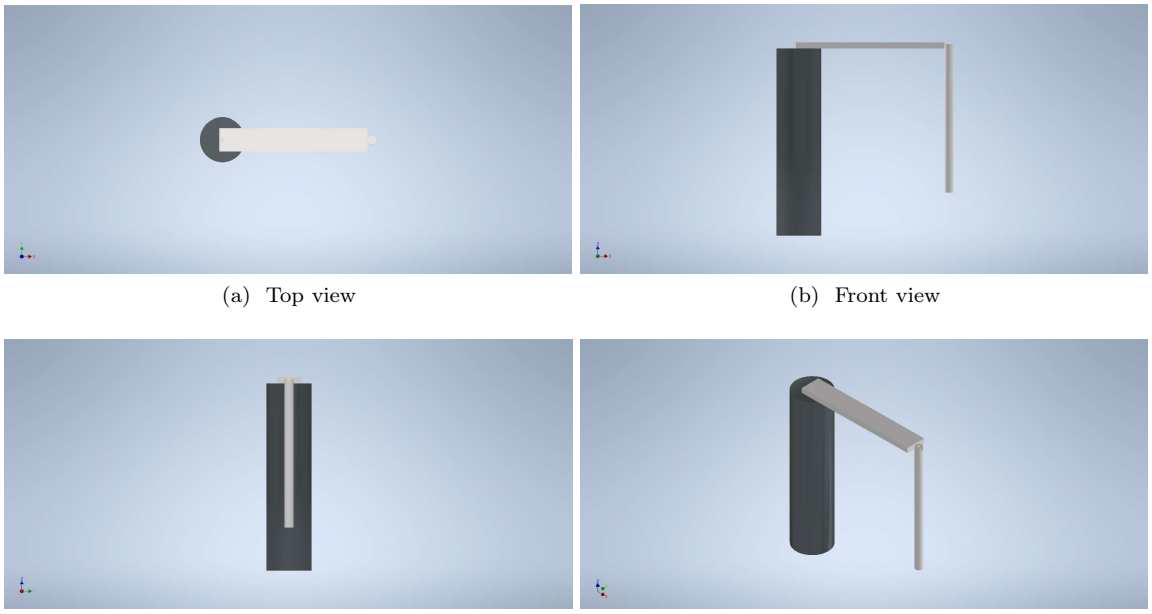
The following table shows the mechanical parameters of Furuta pendulum considered.

	Physical parameter	**Symbol**	**Value**	**Unit**
Arm parameters	Mass	m_1	0.05	kg
	Length	L_1	0.25	m
	Damping coefficient	b_1	0	Nm/rad/s
	Center of mass	l_1	0.125	m
	Thickness	d_1	0.01	m
	Inertia	J_1	0.0002671	kg m^2
Pendulum parameters	Mass	m_2	0.02195	kg
	Length	L_2	0.25	m
	Damping coefficient	b_2	0.00025	Nm/rad/s
	Center of mass	l_2	0.125	m
	Inertia	J_2	0.00011462	kg m^2
Motor parameters	Maximum torque	τ_1	0.35	Nm

Table 3.1: Physical parameters of Furuta pendulum

4 3D CAD Model

In order to simulate the behaviour of the Furuta pendulum controlled by a non-linear MPC, a computer-aided design application for 3D mechanical design, Autodesk Inventor Professional 2020, has been used. In particular, Inventor allows 2D and 3D data integration in a single environment, creating a virtual representation of the final product that enables users to validate the form, fit, and function of the product before it is ever built.



(a) Top view (b) Front view

(c) Lateral view (d) Prospectic view

Figure 4.1: 3D CAD model

In Fig.4.1 different views of Furuta pendulum 3D CAD model are shown. In addition, it can be seen the reference frame in each of view.

5 Non-Linear MPC Controller

Model Predictive Control (MPC), also known as moving horizon control or receding horizon control, refers to a class of algorithms which make explicit use of a process model to optimize the future predicted behavior of a plant [2]. The main items in the design of a predictive controller are:

- the process model;
- a performance index reflecting the reference tracking error and the control action;
- an optimization algorithm to compute a sequence of future control;
- signals that minimizes the performance index subject to a given set of constraints;
- the receding horizon strategy, according to which only the first element of the optimal control sequence is applied on-line.

At each sampling time T_s, a finite horizon optimal control problem is solved over a prediction horizon H_p, using the current state x of the process as the initial state. The on-line optimization problem takes account of system dynamics, constraints and control objectives. The optimization yields an optimal control sequence, and only the control action for the current time is applied while the rest of the calculated sequence is discarded. At the next time instant the horizon is shifted one sample and the optimization is restarted with the information of the new measurements.
As in traditional linear MPC, non-linear MPC calculates control actions at each control interval H_c using a combination of model-based prediction and constrained optimization. The key differences are:

- the prediction model can be nonlinear and include time-varying parameters;
- the equality and inequality constraints can be nonlinear;
- The scalar cost function to be minimized can be a non-quadratic (linear or nonlinear) function of the decision variables.

The standard cost function to be minimized is:

$$J = \sum_{i=1}^{H_p} ||(\hat{x}(k+i|k) - r(k+i|k)||_Q^2 + \sum_{i=0}^{H_c-1} ||\Delta u(k+i|k)||_R^2 \tag{22}$$

where

- $\hat{x}(k+i|k)$ is the prediction, made at time k, of variable x at time instant $k+i$;
- $r(k+i|k)$ is the reference outputs of the system at time instant $k+i$;
- Δu is the variation of the control input variables at time instant $k+i$;
- Q is the weights' matrix of controlled outputs;
- R is the weights' matrix of variation of the manipulated variables;
- H_p is the prediction horizon;
- H_c is the control horizon.

6 Simulations and Results

In the following sections, different simulations are presented with the aim of stabilizing the pendulum in the upward position, even if an unmeasurable disturbance occurs. For each simulation, the basic parameters of the non-linear MPC are the following:

- sampling time $T_s = 50\ ms$;
- prediction horizon $H_p = 10\ samples$;
- control horizon $H_c = 2\ samples$;
- $n_u = 1$, number of measurable variables;
- $n_x = 4$, number of states;
- $n_y = 4$, number of measurable output;
- $n_d = 1$, number of unmeasurable disturbance (which is the disturbance torque applied to the pendulum).

For the input, the maximum rate has been set in order to obtain an aggressive control or not:

$$\Delta u = 0.01\ Nm$$

while the remaining parameters are introduced through each simulations in order to improve the perfomance of the whole sysytem. Fig.6.1 shows the whole control system implemented in SIMULINK.

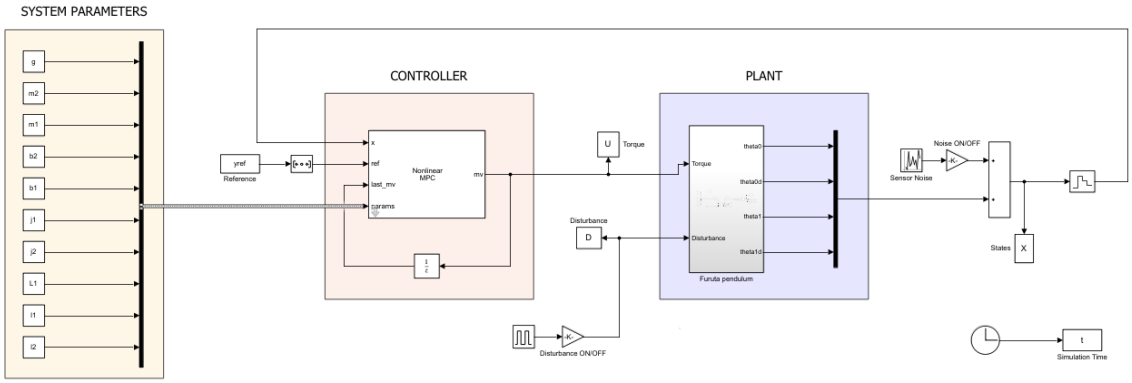


Figure 6.1: Control System on Simulink

In the following sections, different simulations are presented in which the non-linear MPC parameters are tuned in order to obtain an improvement on the behaviour of the whole system. Notice that all the improvement done in each simulation are used for the next one.

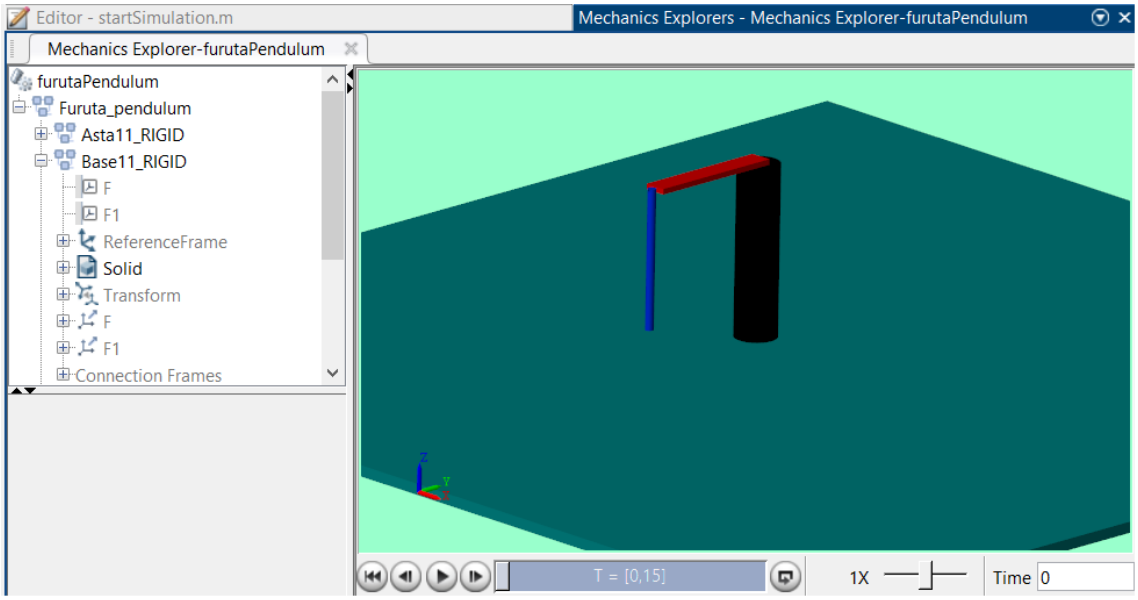


Figure 6.2: Simscape view

Fig.6.2 shows the dialog window of Simscape where the Furuta pendulum behaviour is simulated.

6.1 Simulation without Boundaries on Manipulated Variable

In this simulation the controller has the aim to swing-up the pendulum in the upward position without boundaries on the maximum and minimum value of the manipulated variable (recall torque applied to Arm 1). In addition, the Q matrix has been tuned as follow:

$$Q_1 = \begin{bmatrix} 0.1 & 0 & 0 & 0 \\ 0 & 0.9 & 0 & 0 \\ 0 & 0 & 7 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Fig.6.1.1 and Fig.6.1.2 shows states and manipulated variable MV over time.

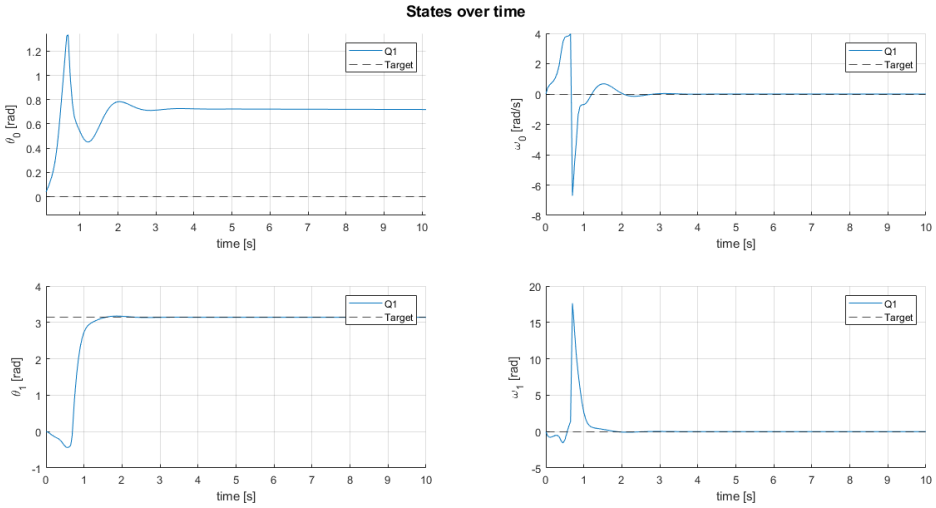


Figure 6.1.1: States of the system over time

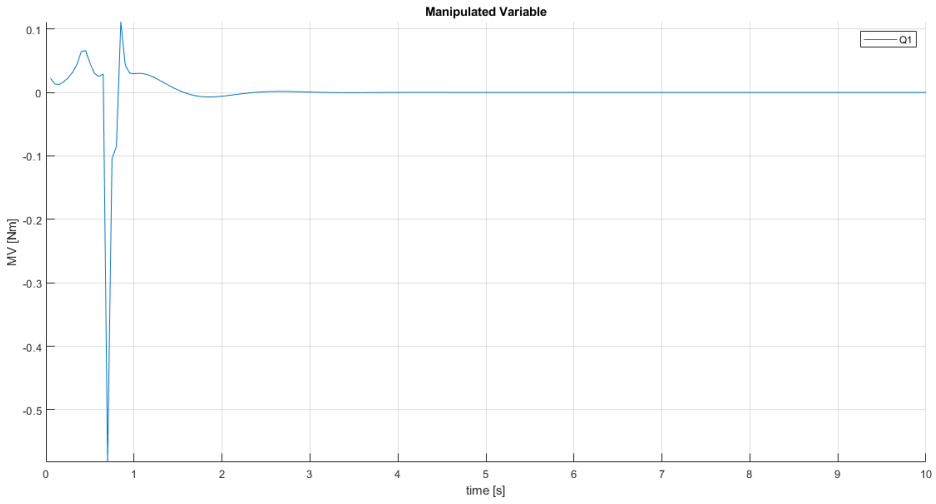


Figure 6.1.2: Manipulated variable over time

As can be noticed, the MV assumes values needed to swing-up the pendulum. As the pendulum reaches the target, the MV is almost null, as expected.
Due to a low weight assigned to θ_0 in the weights' matrix Q_1, a steady state error occurs, while for the the remaining states the targets have been reached.

6.2 Simulation with Boundaries on Manipulated Variable

For a real test-bed application, boundaries on the control action variable must be taken into account in order to match the hardware specifications. For this reason, in this simulation boundaries on the MV are applied, simulating the availability of a DC motor that provides a maximum torque:

$$\tau_{max} = 0.35\ Nm$$

Fig.6.2.1 and Fig.6.2.2 shows the results of the simulation done.

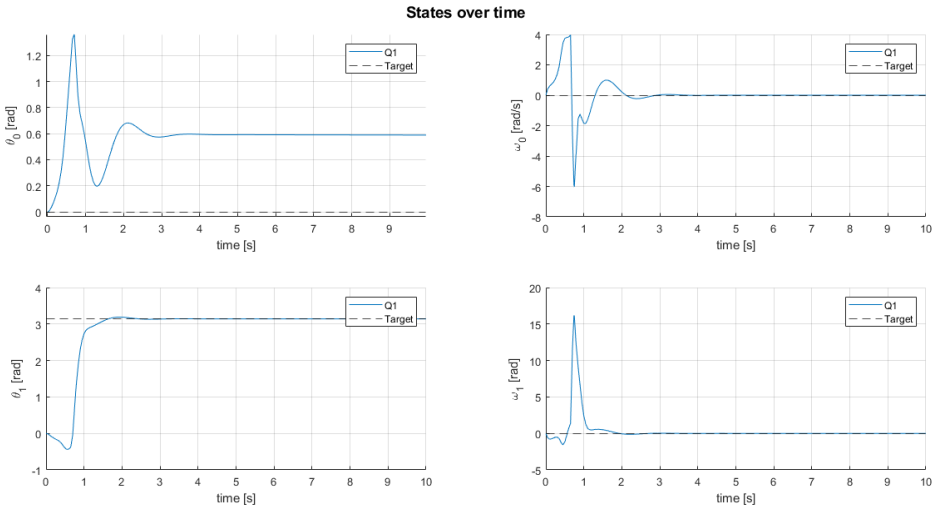


Figure 6.2.1: States of the system over time

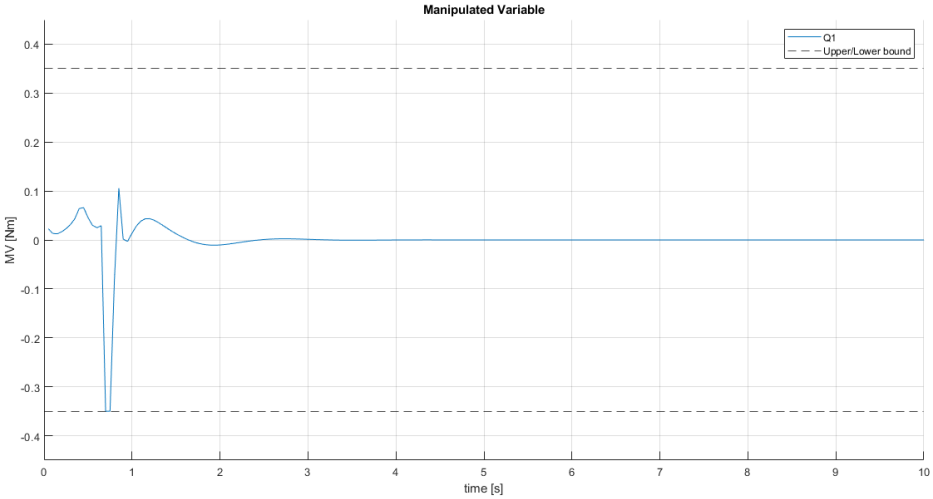


Figure 6.2.2: Manipulated variable over time

As can be seen, the controller limits the MV within the boundaries, compared to Fig.6.1.2, keeping the performance as in the previous simulation while decreasing the energy consumption. In addition, it can be noticed that also in this case the angular position of the motor, θ_0, has not reached the reference position.

6.3 Simulation by Tuning the Q Matrix

Time response of the system can be improved by tuning the weights' matrix Q: in particular, in this case increasing the weights related to angular position of the motor and pendulum. Fig.6.3.1 and Fig.6.3.2 show the performance comparison when the following Q matrices are chosen:

$$Q_2 = \begin{bmatrix} 2.5 & 0 & 0 & 0 \\ 0 & 0.9 & 0 & 0 \\ 0 & 0 & 7 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad Q_3 = \begin{bmatrix} 4 & 0 & 0 & 0 \\ 0 & 0.9 & 0 & 0 \\ 0 & 0 & 9 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

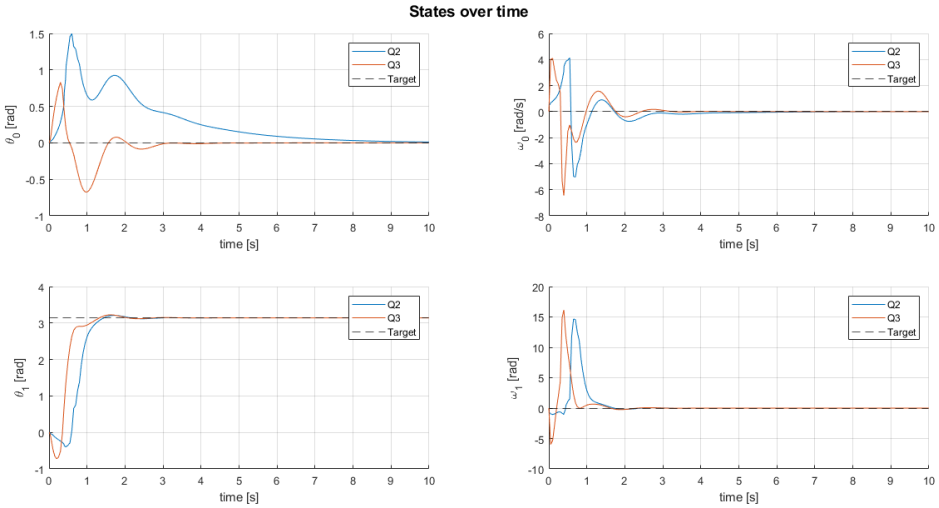


Figure 6.3.1: States of the system over time

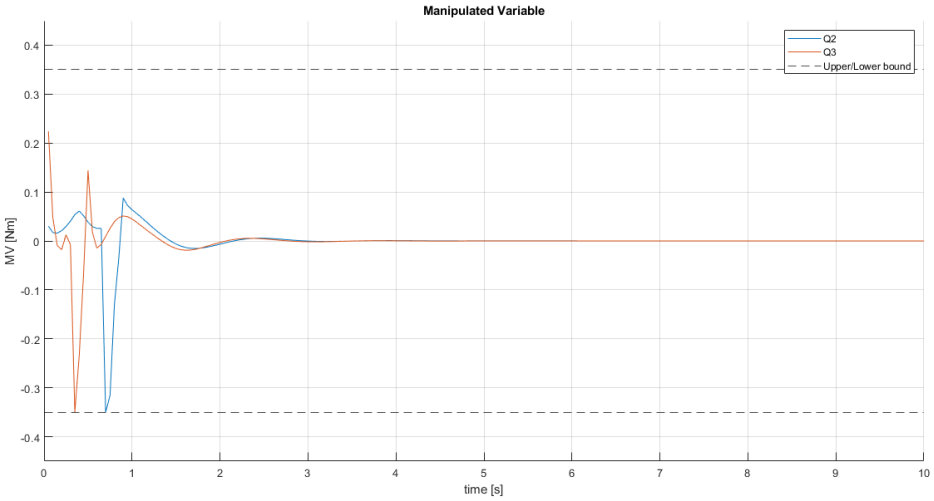


Figure 6.3.2: Manipulated variable over time

As can be notice, by increasing the weights corresponding to θ_0 and θ_1, the settling time for each states has been reduced drastically, improving the performance of the system, wherease the energy consumption is almost the same, as can be seen in Fig.6.3.2.

6.4 Simulation with Unmeasurable Disturbance

The controller has to reject disturbance even if they are unmeasurable. Thus, the system has been tested by applying a pulse torque disturbance on the pendulum at time $t = 10s$, when it is in the upward position. Fig.6.5.1 and Fig.6.5.2 shows the results.

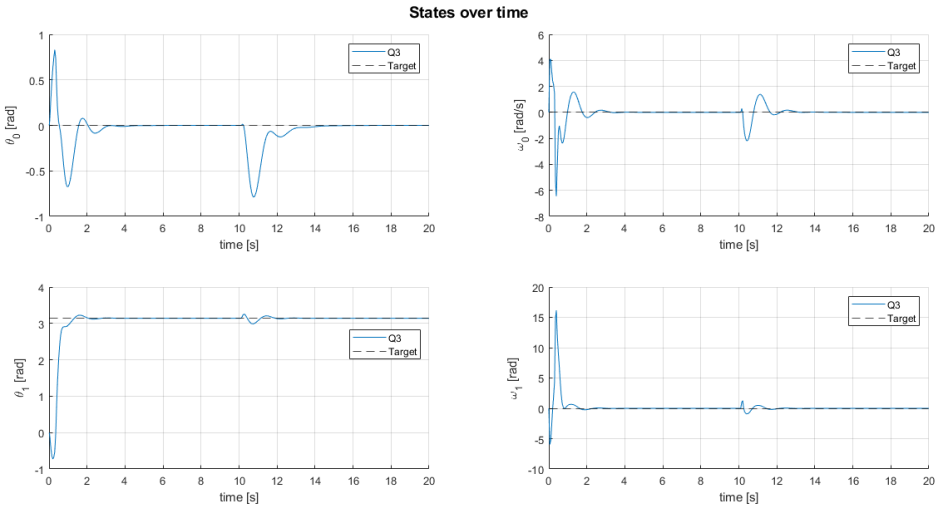


Figure 6.4.1: States of the system over time

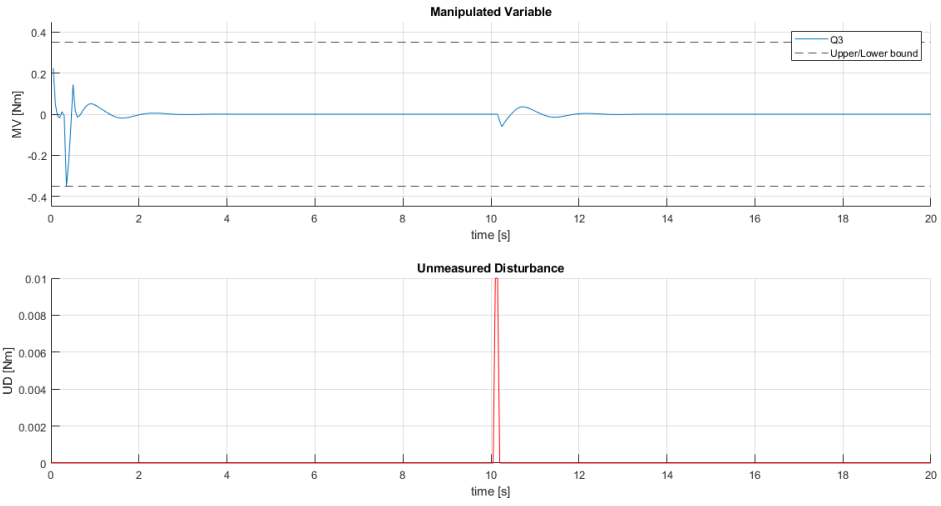


Figure 6.4.2: Manipulated variable and unmeasured disturbance over time

As can be seen, when an unmeasured disturbance is applied to pendulum at time instant $t = 10s$,

the controller computes the required torque in order to keep the pendulum in the upward position according to the targets.

6.5 Simulation with Sensor Noise

A real test-bed could include sensors to measure the states (for example rotary incremental encoder) that are affected by noises and low accuracy. Fig.6.5.1 and Fig.6.5.2 show the performance of the system when a uniformly distributed random noise is added to the full states with upper/lower limits:

$$\epsilon = \pm 0.01$$

that allows to simulate the use of a rotary incremental encoder with accuracy equal to ± 0.57 *deg*.

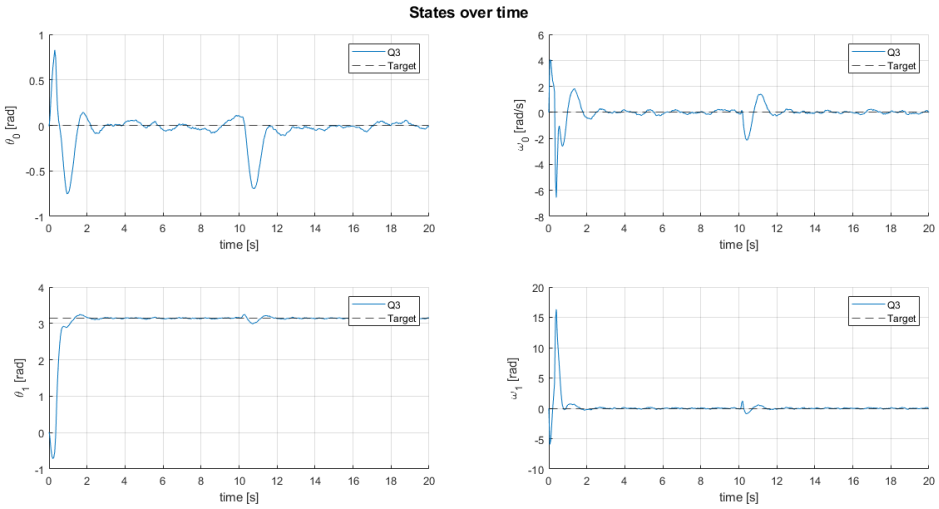


Figure 6.5.1: States of the system over time

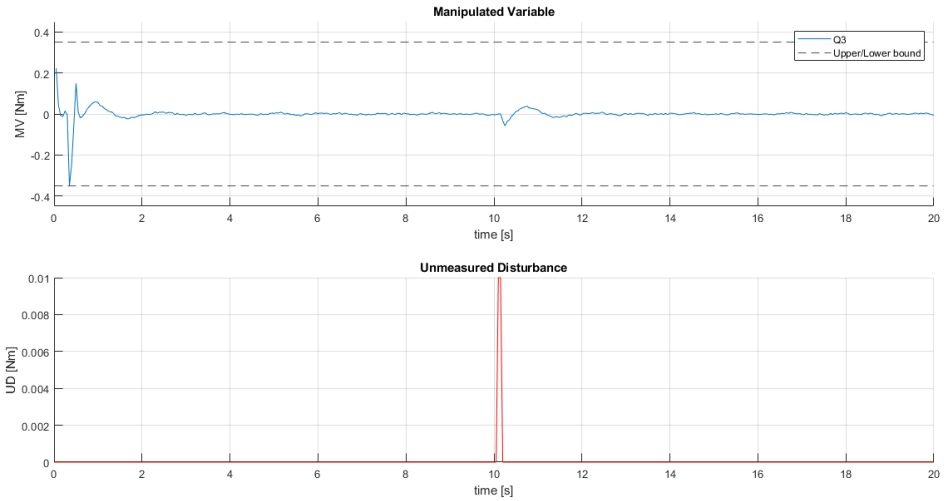


Figure 6.5.2: Manipulated variable and unmeasured disturbance over time

The simulation shows that by adding the sensors noise, the controller still provide the required torque to swing-up the pendulum and stabilize it in the upward position, but the states of the system oscillate around the corresponding target.

6.6 Simulation with Boundaries on Output Variables

In many applications, output variables of a system cannot assume arbitrary values during the transient, so hard constraints on the output must be considered. In the study case an hard constraint on the pendulum angular velocity has been set, such that:

$$-14.5\ rad/s < \dot{\theta}_1 < 14.5\ rad/s$$

Fig.6.6.1 and Fig.6.6.2 show the results of the simulation done.

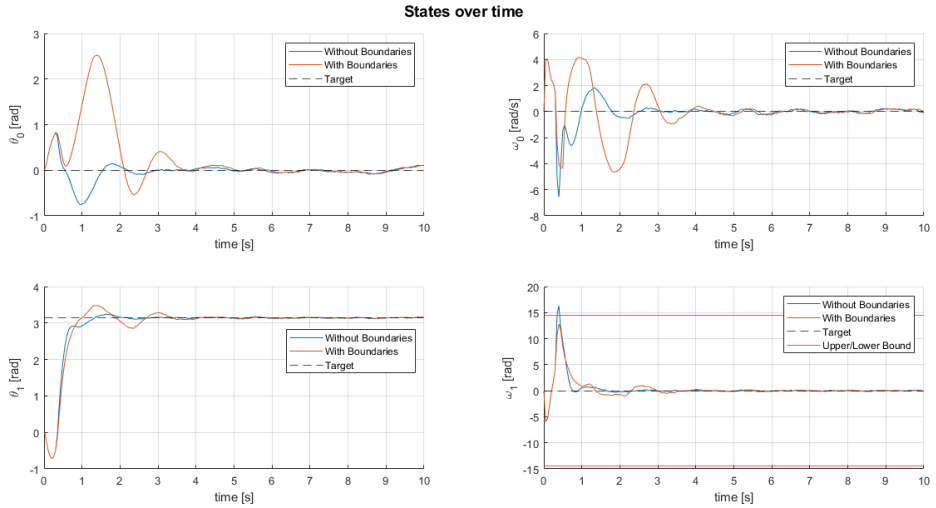


Figure 6.6.1: States of the system over time

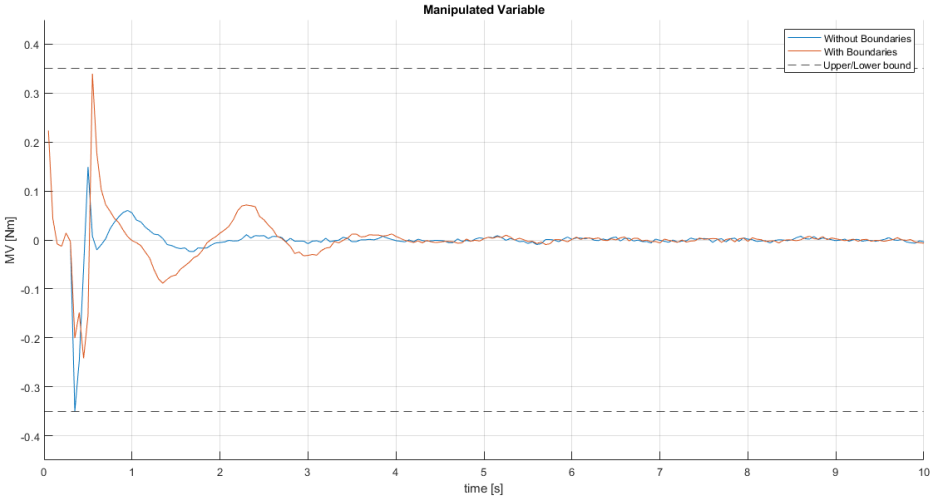


Figure 6.6.2: Manipulated variable over time

As can be seen, the controller limits $\dot{\theta}_1$ within the boundaries. The latter also limits the kinetic energy introduced in the system and useful to swing-up the pendulum, so that overshoots on the other states have been increased due to the objective of the controller to stabilize the system.

7 Conclusions

In this project the implementation of non-linear MPC controller for a Furuta pendulum has been presented. After having created the 3D model of rotary inverted pendulum, it has been imported in Matlab using the Add-On Simscape Multibody in order to obtain a more realistic non-linear model which could take into account all the physical parameters of the system (such as damping coefficients of DC motor's shaft, inertia tensors, ecc..). The non-linear state space model of rotary inverted pendulum is defined and different simulations have been done varying the non-linear MPC controllers' parameters (weights, constraints) in order to observe the different behaviour of the system. As can be seen in Sec.6.6, the resulting non-linear MPC parameters allows the system to reach the desired performances.

References

[1] Zebb Prime Benjamin Seth Cazzolato. "On the Dynamics of the Furuta Pendulum". In: *School of Mechanical Engineering, University of Adelaide, Adelaide, SA 5005, Australia* (2011).

[2] J.M.Maciejowski. *Predictive control with constraints*. Prentice Hall.

List of Figures

List of Tables