POLITECNICO DI TORINO

Master Degree

in Mechatronic Engineering

Modelling, design and control of a cart inverted pendulum using electrical and pneumatic actuation

Supervisor
prof. Luigi Mazza

Authors
Vittorio Cordasco
Andrea Di Maggio

2019/2020

Abstract

This thesis presents the control of a Cart Inverted Pendulum system, starting from the modelling and design of a test bench. The Cart Inverted Pendulum system is actuated via a DC Motor and a transmission mechanism composed by a screw and a linear guide. The electric motor is controlled by a servo drive system and a PLC. Matlab and Simulink are used to asses the performance of the system in the presence of different controllers: PID controllers in cascade and parallel architectures, Static State feedback via pole placement or LQR.

The first part of the thesis deal with the definition of a mathematical model of the system. It is implemented by Simulink in a block diagram representation. Different techniques are used to control the system, basing on PID and Static State Feedback control theories. The PID controllers are implemented in two different architectures, cascade and parallel. The cascade architecture is obtained through two loop, inner and outer. The inner loop stabilizes the pendulum angle, using the first PID, while the outer one stabilizes the position of the cart, using the second one. The parallel architecture is obtained by two parallel loop, that separately stabilize the pendulum angle and the position of the cart; the parallel architecture provides an input to the plant, using the sum of the outputs of the two PID controllers. The Static State Feedback control theory is based on a negative feedback branch; the states of the system are multiplied by a gains matrix $[K]$ and then they are fed back to the control input. The considered ways, in which the gains matrix $[K]$ can be computed, are the pole placement and the LQR. Moreover in the Static State Feedback technique, an asymptotic observer is needed due to the limited number of the available measurements (states). The observer is able to provide an estimation of all the states, basing on the measurements of the input and only one output. Afterwards, the same control techniques are applied to a Simulink model based on a real test bench designed and built at the Laboratory of the Politecnico di Torino described in [4], that uses a pneumatic actuation. All the controllers are computed in order to guarantee the specifications on the time response and the physical constraint of the plant (supply voltage in the electrical actuation and internal chamber pressure on the pneumatic one).

Finally, the last chapter presents the comparison between the proposed control techniques and between the two different types of actuation. Principally, the comparisons highlighted opposite behaviours in the two types of actuation; the pneumatic actuation allows a very low values of settling time in the response of the cart position, causing, however, an oscillating behaviour of the tilt angle of the pendulum, while the electrical actuation provides a smooth trend in both output responses with high values of settling time. All the results are commented to guarantee a detailed report on the entire study, carried out on this classical control theory problem.

The entire development of this dissertation is carried out in parallel by both students, Vittorio Cordasco and Andrea Di Maggio. However, the draft chapters 1, 4 and 5 is accomplished by Andrea Di Maggio, while the draft chapters 2 and 3 is realized by Vittorio Cordasco.

Contents

List of figures

List of tables

List of symbols

x	Linear position of the cart
$\dot{x}$	Linear velocity of the cart
$\ddot{x}$	Linear acceleration of the cart
α	Angle of the rod with respect to the vertical axis
$\dot{\alpha}$	Angular velocity of the rod with respect to the vertical axis
$\ddot{\alpha}$	Angular acceleration of the rod with respect to the vertical axis
C_u	Torque generated by the linear acceleration of the cart
J_p	Moment of inertia of the mass with respect to fulcrum
m_p	Concentrated mass of the pendulum
g	Gravity acceleration
l_p	Rod length of the pendulum
θ	Angular position of the motor shaft
$\dot{\theta}$	Angular velocity of the motor shaft
$\ddot{\theta}$	Angular acceleration of the motor shaft
p	Pitch of the screw
η	Efficiency of the transmission system
F_p	Horizontal component of constraint reaction transmitted by pendulum to cart
C_p	Torque of constraint reaction transmitted by pendulum to cart
x_p	Relative position of the concentrated mass with respect to the vertical axis
V_a	Armature voltage of the motor
E	Back EMF by self-inductance
I_a	Armature current of the motor
C_m	Output torque generated by the motor
R_a	Armature resistance
K	Torque constant
J_{tot}	Total inertia moment of the system with respect to the motor
J_m	Moment of inertia of the rotor shaft
J_v	Moment of inertia of the transmission system
J_{cart}	Moment of inertia of the assembly of the cart and pendulum components
m_v	Mass of the screw
r_v	Radius of the screw
m_c	Mass of the cart
$G_{V_a\alpha}$	Transfer function between input V_a and output α
$G_{V_a x}$	Transfer function between input V_a and output x
$G_{\alpha x}$	Transfer function between output α and output x
G	Generic transfer function of the plant
F	Denominator of the closed loop transfer function

H	Closed loop transfer function
Z_p	Number of poles of the closed loop transfer function
P_p	Number of poles of the generic transfer function of the plant
K_p	Proportional coefficient of the PID controller
K_i	Integrative coefficient of the PID controller
K_d	Derivative coefficient of the PID controller
N	Pre-Filter coefficient of the PID controller
x_{ref}	Reference signal for the output x
α_{ref}	Reference signal for the output α
x_{feed}	Feedback signal of the output x
α_{feed}	Feedback signal of the output α
x_{err}	Error signal of the output x
α_{err}	Error signal of the output α
C_α	Controller for the compensation of α error
C_x	Controller for the compensation of x error
$G_{u\alpha_{cl}}$	Closed loop transfer function of the inner loop in cascade architecture
$G_{ux_{ol}}$	Plant transfer function of the outer loop in cascade architecture
u_α	Output of C_α in parallel architecture
u_x	Output of C_x in parallel architecture
u	Input signal for the plant
$G_{x_{cl}}$	Closed loop transfer function between x_{ref} and x_{feed} in parallel architecture
$G_{x_{ol,un}}$	Open loop transfer function of the uncontrolled system for x in parallel architecture
$G_{x_{ol}}$	Open loop transfer function of the controlled system for x in parallel architecture
N_{cl}	Numerator of the closed loop transfer function
D_{cl}	Denominator of the closed loop transfer function
$N_{ol,un}$	Numerator of the open loop transfer function of the uncontrolled system
$D_{ol,un}$	Denominator of the open loop transfer function of the uncontrolled system
N_{ol}	Numerator of the open loop transfer function
D_{ol}	Denominator of the open loop transfer function
K_c	Largest gain at which the output of the control loop has stable and consistent oscillations
T_c	Oscillation period of the output response using K_c
A	State Space matrix
B	State Space matrix
C	State Space matrix
D	State Space matrix
X	States vector in the State Space representation
$\dot{X}$	States vector derivative in the State Space representation
Y	Outputs vector in the State Space representation

K	State feedback gain matrix
K_i	Gains vector of K
K_o	Gains vector of K
Q_c	Controllability matrix
Q_o	Observability matrix
J	Cost function
Q	Weighting matrix of the states
R	Weighting matrix of the input
P	Matrix that satisfies the algebraic Riccati equation
N	Gain correction
q	State of the augmented states vector x_{tot}
x_{tot}	Augmented states vector in the State Space representation
$\dot{x}_{tot}$	Augmented states vector derivative in the State Space representation
A_{tot}	Augmented State Space matrix
B_{tot}	Augmented State Space matrix
C_{tot}	Augmented State Space matrix
D_{tot}	Augmented State Space matrix

Introduction

In 1990, the International Federation of Automatic Control Theory Committee (IFAC) identified several practical problems useful to compare new and existing control methods, so that a meaningful comparison can be derived. The federation decided to highlight these problems as "benchmark control problems", like the Cart Inverted Pendulum Control problem. It was defined as a highly unstable and under-actuated system, with a non-minimum phase rotation. The complexity of this control problem increases with the number of links constituting the systems. For these reasons, the Cart Inverted Pendulum proves to be a challenging control problem. The relevance of this problem is due to the fact that it can be associated with practical applications:

- Robotics: the vertical stabilization of humanoid robots used in domotics and industrial application
- Construction industry: buildings behaviour and resistance under earthquake phenomena
- Transports: automatic transports like overboard, segway, bicycle, pendulum robot, e.g.
- Aerospace: control of the attitude in rocket launching

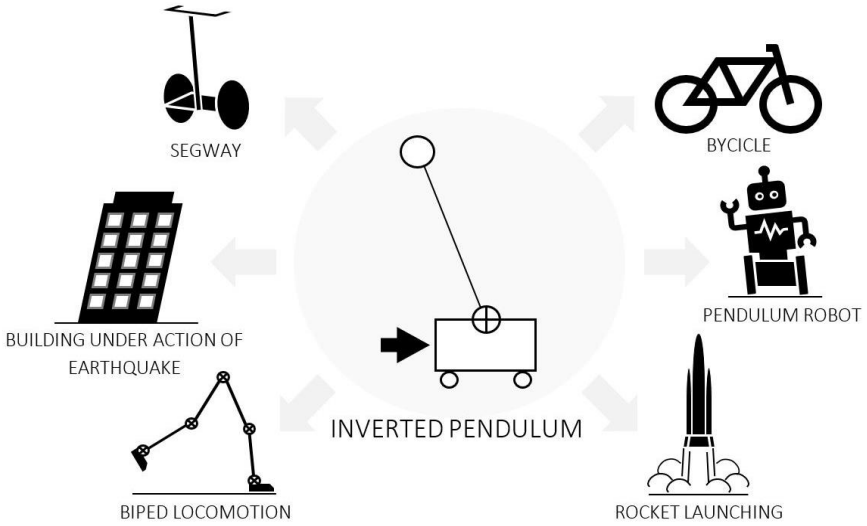


Fig. a Actual industry applications of the inverted pendulum

As it can be seen from figure a, the simplest type of inverted pendulum is constituted by a translating cart which is connected through a cylindrical hinge to a rod which presents a mass on its extremity. As already mentioned, the goal of the control algorithm is to maintain the pendulum rod in vertical position.

To do this, it is necessary to compensate for the inherent limitations of the system:

- Instability: the inverted vertical position is the unstable equilibrium point of the system
- Non-linearity: the dynamic equations describing the inverted pendulum have non-linear terms
- Non-minimum phase: looking at the transfer functions, they contain non-minimum phase zeros (positive value), which affect the stability margins, including the robustness, and influence the time response of the systems
- Under-actuated: there are two control variables and only one control input, that is the force applied to the cart. This means that the system has a lower number of actuators than the degrees of freedom. Differently from other mechanical systems like segways or overboards that can exploit any arbitrary trajectory for their stabilization, the cart of the inverted pendulum can employ only a limited and linear trajectory to get stability

Moreover, depending on the application and the selected actuation system can be many other issues, like the space constraints of the designed system or the physical constraints of the used actuation. Looking at literature, there are two types of actuation, pneumatic and electrical. The pneumatic actuation is obtained by a cylinder that can extend and retract its rod, that is directly connected to the cart of the pendulum (in same cases the cylinder rod is linked to a connection plate, that mounts on the top a hinge connected to the pendulum rod). The electrical actuation is obtained through an electric motor (commonly DC motor) with a rotational outer shaft connected through different types of motion transmission systems to the connection plate and the hinge of the inverted pendulum (linear guide with different type of transmission systems, like screw-screw nut or belt-pulley assemblies [3]). The motor provides the torque to the motion transmission system, that converts the torque into linear force used to move the inverted pendulum cart. In both cases, the draw backs are due to the physical (supply pressure in pneumatic cylinder and supply voltage/current in the electric motor) and inherent (e.g., low dynamics) limitations of the actuation systems. These contributes render the control problem even more complex.

Firstly, this thesis considers the design of a control system for a cart inverted pendulum test bench using an electric motor as actuator and a screw-screw nut linear guide to control the position of the cart. The control algorithms adopted here in can be broken down in two different control strategies:

- PID controllers with cascade and parallel architectures
- Static state feedback control with a gain feedback obtained by pole placement technique or by LQR algorithm

Afterwards, the same techniques are applied to a system that employs a pneumatic actuation [4]. Finally, the last part of the work presents the comparison of considered control techniques by highlighting pros and cons of each method.

Current trends

The state-of-art inverted pendulum systems boast many design models, which differ principally in the available physical controller and in the actuation and motion transmission systems. Before beginning the analysis of the problem presented in this dissertation, an explanation of the models taken as reference is needed to understand which are the basis of our study.

[1] and [2] are the texts taken as reference for everything concerning the control theory, from the analysis and the definition of a model to the stability topics (stability criteria, frequency response, controller design). Text [3] provides the needed knowledge to the definition of the model used in the simulation environment. Here, the implemented system is composed by an electric motor connected to the inverted pendulum cart thanks to a belt-pulley transmission system. The control techniques applied to this system are

- PID compensators in cascade architecture
- Frequency design: loop shaping technique
- Static state feedback: pole-placement technique

All these control strategies have proved to be very powerful in terms of output response and robustness. These performances have been taken as reference to test the reliability of the techniques used in this dissertation.

The information about the model with pneumatic actuation, that is considered in section 3.3, was taken from [4]. The data provided by [4] are used to apply the control techniques adopted in the model with electrical and pneumatic actuations and, in chapters 4 and 5, to compare the two types of actuations. Moreover, this dissertation provides the description of the real test bench, preserved in Lab DIMEAS of the Politecnico di Torino, built by the authors. The test bench is composed by a cart integrated with an inverted pendulum. These are mounted on plate (of a linear guide) and are directly connected to a translating rod of a pneumatic cylinder. The double acting cylinder is used as a pneumatic actuator, that is activated by a series of simple on-off 2/2 ways electro-pneumatic valves ($\mathrm{AP}-7211-\mathrm{LR2\_U7}$), modulated by PWM signal, while the controller is implemented by a PLC code. The control strategy considered in [4] is a PID compensators in cascade architecture. Firstly, a classical control theory criterion (Nyquist criterion, like in our study) is chosen to verify the achieved stability considering the linearized model of the plant. Then, the software Amesim is used to perform a non-linear analysis of the system and to set up the final controllers.

Another control applied to the same test bench that exploits different simulation's software is presented in [5]. The designed control system is composed by two PID compensators placed into two-parallel loop in order to individually control the tilt angle of the pendulum and the position of the cart. The system simulation is carried out using DSHplus and Matlab/Simulink environments.

The same control architecture and model with pneumatic actuation of text [4] are analyzed in [6]. However, the controller implementation is different because an Arduino board is adopted, instead PLC, using a PWM generation technique programmed by Fuzzy logic. This choice leads to a very good control performances, in terms of output response, and to low costs because of implementation of the control algorithm on Arduino.

Regarding to the other control techniques used in our dissertation, several papers have been analyzed to understand the state-of-art modelling of the plant and to learn about the techniques adopted for control purposes. The [7] introduces another control method, the Static State Feedback using pole placement technique, that has proven to be very powerful in terms of output response. Moreover, this paper is useful to solve the practical issue of the available output measures, because the Static State Feedback control needs the feedback of all the states to respect the assigned control law (the states defined in the state vector of the system, using the State Space Representation). So, the asymptotic state observer is used to solve this problem, estimating all the states by knowing only one of them (from real measurement) and the control input of the considered plant. Therefore, the final control architecture is composed by a Static State Feedback, with pole placement technique, and by an asymptotic state observer, that estimates all the states. The results show that the proposed controller provides excellent performances. Also, the system presents a good robustness, it can overcome any external disturbances on cart or on pendulum rod.

The [8] and [9] exploit the design of the two parallel-loop PIDs' architecture and, in particular, the [8] shows the implementation of the PID controllers to allow the stabilization and the tracking control of three types of inverted pendulum

- Inverted pendulum with one Degree of Freedom (DOF), the cart can freely move only in one direction (x-axis)
- Inverted pendulum with two DOF, the cart can freely move in a plane (x-y plane)
- Inverted pendulum with three DOF, the cart is freely to move in the three-dimensional space (x-y-z)

Simulation results prove that the way to design the PID controllers is very simple and effective. Although the system design can realize stabilization and tracking control of three types of inverted pendulum, also it provides robustness to outer large and fast disturbances. Conversely to [8], the [9] considers only the one DOF inverted pendulum model, but two control technique have been applied that are an innovative double PID control method and a modern LQR (liner quadratic regulator) control method. Dynamic and steady state performances are investigated and compared in the two types of controllers. This paper proves that the LQR controller can guarantee a faster and smoother stabilizing process for the inverted pendulum, providing better robustness and less oscillations than the double PID controller.

Finally, the [10] describes a test bench composed by an electric DC motor, as actuator, and a belt-pulley mechanism, as motion transmission system, to control the cart inverted pendulum. Here, several complex algorithms are designed and implemented into a PC with a PCI-1711 card.

The control algorithms, presented in [10], can be resumed as

- Systematic algorithm for weight selection of LQR state feedback
- Two-loop PID controller designed by pole placement approach. The design is based on dominant LQR poles. This leads to an improvement in the cart response with damped oscillations
- State feedback control design by sub-optimal LQR subjected to H∞ constraint
- Integral Sliding Mode (ISM) via pole placement algorithm in order to provide better robustness on the output channel and superior cart position response than LQR

Looking at the obtained simulation and at the experiments, the ISM exhibits a good robustness in presence of output channel gain perturbations. The efficiency of the developed techniques is tested in numerical simulations. It has been also observed that the two-loop PID controller yields satisfactory response of cart position and robust property. In the event of sensor fault, the ISM provides best performance out of all the techniques.

1 The Test bench

The test bench was designed using the SolidWorks environment. This chapter provides a description of the test bench components. All the parts were designed in order to provide an easy assembly between them and to respect the specifications in terms of length of the path ($500\ mm$) and angle amplitude of the pendulum ($\pm 25°$ respect the vertical axis).

Looking at figure 1-1, the seven notes indicate the most important assemblies, that creates the overall test bench. The components of the test bench can be subdivided in three different branches related to the application environment of the system (mechanical, actuation and electrical). In the mechanical environment there are the pendulum system "1" and the cart system "2". Regarding to the actuation environment there are the motion transmission system "3" and the motor "4". Finally, the PLC "5" and the servo drive system "6" constitute the electrical environment, while the element "7" is the panel designed to fulfil the assembly of all components.

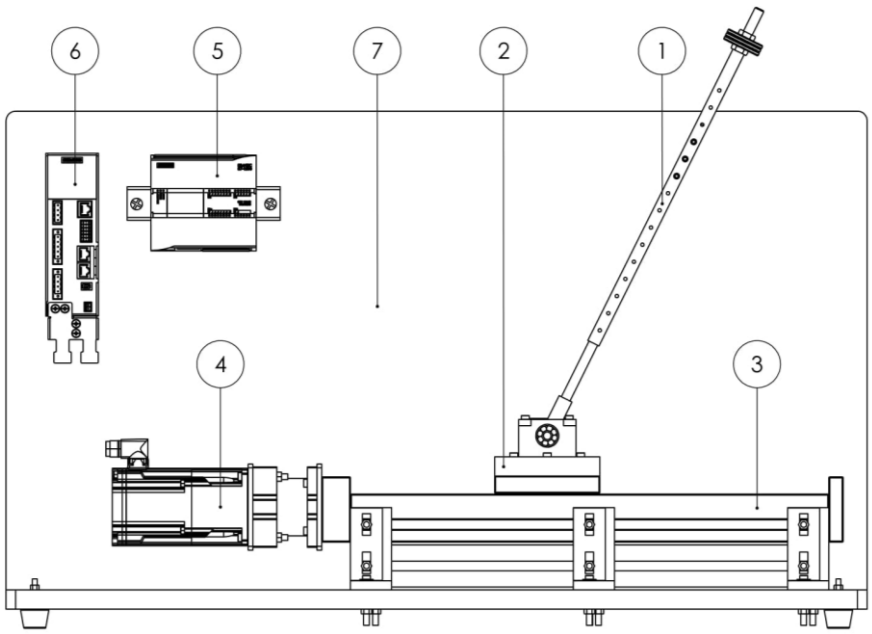


Fig. 1-1 The Test bench

In the next sections, the design of the pendulum and cart assembly and the other test bench components are described.

1.1 Cart Inverted Pendulum

In this section, the design and assembly methods of the components, releated to the mechanical environment, are provide. The technical drawings of the described components are presented in Appendix A.

1.1.1 Pendulum design

The pendulum system, shown in figure 1.1-2, is designed in order to provide the variation of the rod length and of the mass weight, and to guarantee its integration with the cart.

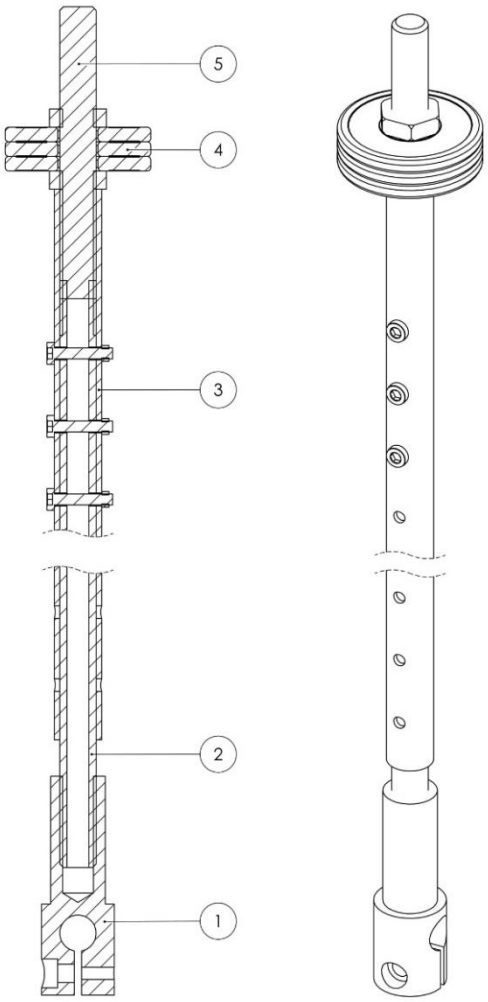


Fig. 1.1-1 Pendulum system

As it can be seen in figure 1.1-1, the length can be modified thanks to the telescopic property of the rod design, that consists of element "2" and "3", that are the internal and external parts of the rod. The rod length can assume values between $400\ mm$ and $700\ mm$ by increments of $20\ mm$ (choosing the suitable coupling holes of "2" and "3"). The pendulum mass can be modified using the coupling between "2" and "5" (internal rod and tip cylinder), that gives the possibility to change the number of weights fixed at the end of the pendulum "4". The separation of the pendulum system from the cart system (explained in the next section) is allowed by the threaded coupling between "2" and "1" (internal rod and two-axis joint). In order to accomplish the concentrated mass hypothesis the pendulum was built in a light aluminium structure.

1.1.2 Cart assembly design

The cart system, shown in figure 1.1-2, provides the way in which the pendulum system is connected to the motion transmission system.

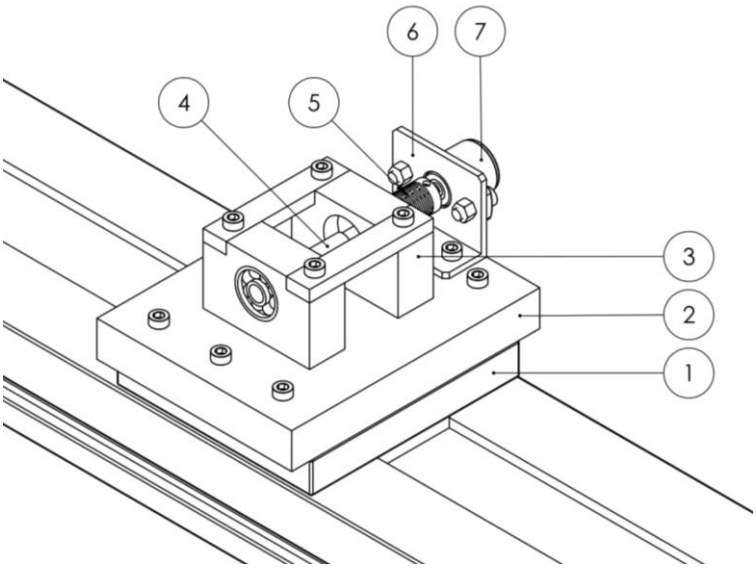


Fig. 1.1-2 Cart system

The connection plate "2" is designed to connect the plate of the linear guide "1" to the two bearing supports "3" and the mounting bracket "6" for the angular sensor "7". The pendulum system is connected to the cart using by the two bearing supports "3" and the shaft "4". The assembly "3"-"4" with the element "1" (of the pendulum system) forms the rotational joint used to allow the rotation of the pendulum. Finally the joint "5" provides an elastic coupling between the shaft "4" and the angular sensor "7", in order to compensate alignment offset.

1.2 Actuation system and sensors

An overview on the components specifications, that constitute the actuation environment and the electrical one, is given in the next sections.

1.2.1 Motor

The actuation used in this system is provided by SIEMENS with the product code SIMOTICS S-1FK2104-4AK10-1MA0, shown in figure 1.1-3.

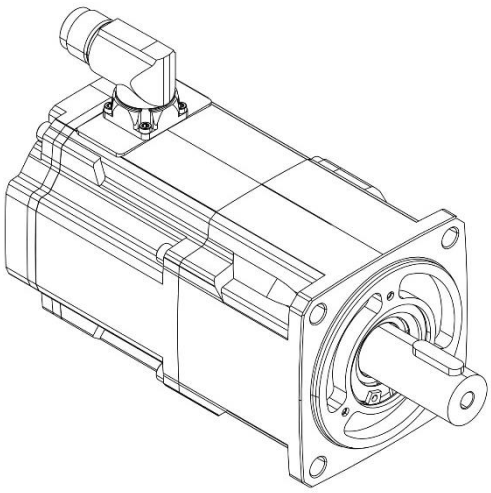


Fig. 1.2-1 The motor

It is a permanent-magnet synchronous motor connected to the linear guide shaft by a coupling flange with bearings. The motor is able to provide the information about the angle position of the shaft thanks to its absolute encoder 22 bit. The information about the shaft position allows the position computation of the cart by the transmission ratio provided by the linear guide. The table 1.2-1 summarizes some of the motor specifications.

Static torque	1.27 Nm
Static current	2.4 A
Rotor moment of inertia	0.430 $Kgcm^2$
Rated speed	3000 rpm
Rated torque	1.27 Nm
Rated current	2.4 A
Rated power	0.40 KW
Torque constant	0.53 Nm/A
Armature resistance	1,7 Ω

Table 1.2-1 Motor specifications

1.2.2 Servo drive system

The linking element between the motor and the PLC is the servo drive system provided by SIEMENS with the product code SINAMICS S210, shown in figure 1.2-2.

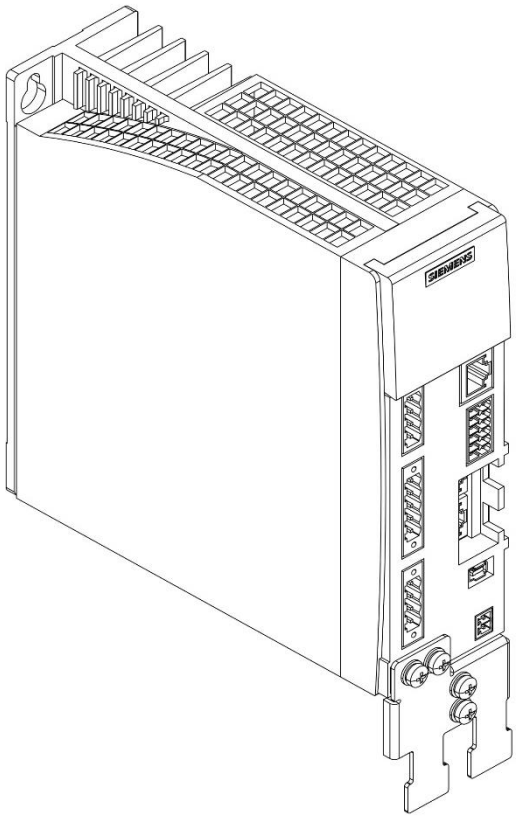


Fig. 1.2-2 The servo drive system

The motor, the PLC and the servo drive system create the overall electrical actuation (straddling the actuation environment and the electrical one of the test bench, mentioned before). The servo drive system is a converter that gives a communications interface between the controller unit (PLC) and the physic actuation (motor), allowing a complete motion control and an easy plug-and-play commissioning with one-cable connection.

1.2.3 PLC

Also the used controller, shown in figure 1.2-3, is provided by SIEMENS with the product code SIMATIC S7-1200, the version is the one with the CPU 1214C DC/DC/DC.

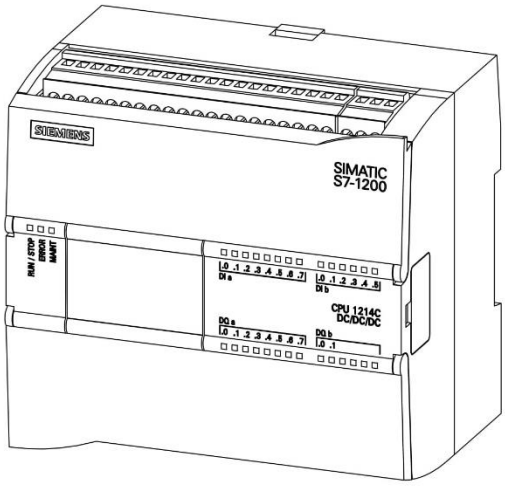


Fig. 1.2-3 The PLC

The table 1.2-2 summarizes the main specifications of the PLC.

CPU code	$CPU\ 1214\ DC/DC/DC$
Digital input ports	14
Fast digital input ports	6 $(of\ the\ 14\ digital)$
Digital output ports	10
PWM output ports	4 $(of\ the\ 10\ digital)$
Analog input ports	2 $(0 \div 10\ V_{DC})$
Supply voltage	24 V_{DC}

Table 1.2-2 PLC specifications

1.2.4 Linear guide

The adopted motion transmission system is provided by AUTOMATIONWARE with the product code ML120PRO-22, shown in figure 1.2-4. The transformation from the rotational motion of the motor to the linear motion of the cart is allowed by the screw-screw nut mechanism of the transmission system. The linear guide is connected to the motor thanks to a coupling flange with bearings, and, on the other side, the cart is connected to the transmission system using its plate (element "1" in figure 1.1-2).

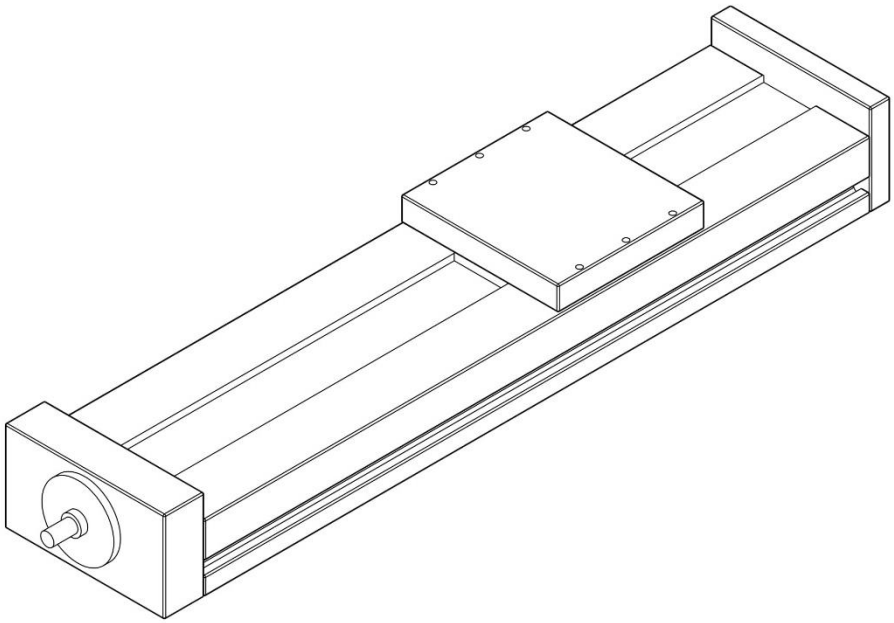


Fig. 1.2-4 The linear guide

The table 1.2-3 summarizes the main specifications of the linear guide.

Screw standard	$ISO\ 7$
Screw specifications	$\oslash\ 10mm\ -\ p\ 10mm$
Carts number	2
Path	$520\ mm$
Output shaft	$\oslash\ 10\ mm$
Maximum velocity	$1.2\ m/s$
Maximum acceleration	$20\ m/s^2$

Table 1.2-3 Linear guide specifications

1.2.5 Angular sensor

Finally, the angular sensor used for the tilt angle measurements of the pendulum is provided by VISHAY with the product code Model 981 HE Series, it is a single turn hall effect type and it is shown in figure 1.2-5.

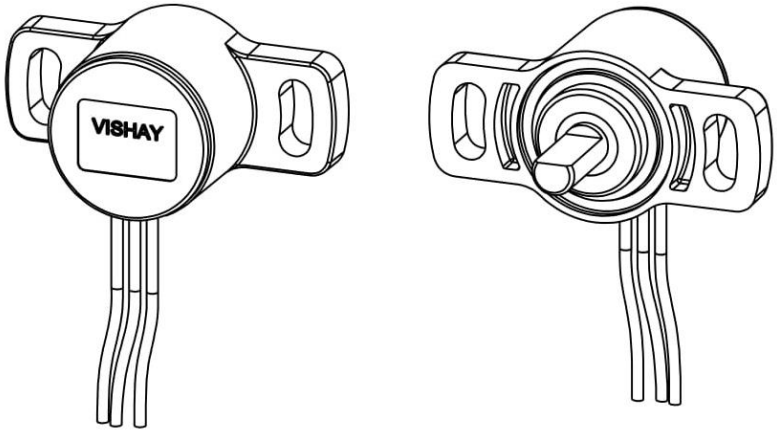


Fig. 1.2-5 The angular sensor

The table 1.2-4 summarizes the main specifications of the angular sensor.

Type	$Single\ turn\ Hall\ effect$
Supply voltage	$5\ V_{DC} \pm 10\%$
Supply current	$10\ mA\ /\ 16\ mA\ max$
Output signal	$Analog\ ratiometric, 10\%\ to\ 90\%\ of\ V_{supply}$ $PWM, 1KHz\ 10\%\ to\ 90\%\ duty\ cycle$
Electrical angle	$90°, 120°, 180°, 270°, 360°$
Output shaft	$\varnothing\ 10\ mm$
Maximum velocity	$1.2\ m/s$
Maximum acceleration	$20\ m/s^2$

Table 1.2-4 Angular sensor specifications

2 Mathematical model

The implementation of the mathematical model of the system represents the first step for the design of its controller. Once the model is done, the transfer functions, that describe the behaviour of the system in time domain, are obtained. The goal of the controllers is the stabilization of the transfer function between the input u and the outputs of the system. Referring to this case of study, the armature voltage V_a of the motor is chosen as command input, while a suitable choice of the outputs is the position of the cart x and the angle α between the rod of pendulum and the vertical axis. These variables can be directly measured by the motor encoder and the angular sensor, respectively. According to this choice, the system becomes SIMO (single-input multiple-output) but, as can be seen in next chapters, the entire system will be approximated as SISO when the static state feedback control is used. The goal is the stabilization of the cart position x in the center of the linear guide (like the cascade control model of the pneumatic actuation studied in [4]).

Therefore, the first step to obtain the model of the whole system is to writing down the equilibrium and kinematic equations of each component. Matlab and Simulink are used to implement the model and to validate and test the system as software in the loop. For practical reasons, the whole system can be subdivided into two different subsystems

- the Inverted Pendulum that is the controlled plant.
- the Actuation Systems constituted by the motor and the transmission system (screw-screw nut).

Next sections describe the equations constituting the model of each subsystems.

2.1 Inverted pendulum

Figure 2.1-1 shows a typical configuration of the system

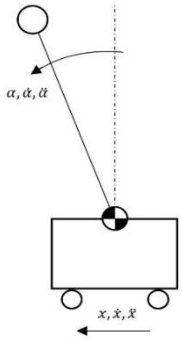


Fig. 2.1-1 Classical representation of the cart inverted pendulum system

As shown in figure 2.1-1, the system is constituted by a mass linked to a rod which is hinged at one end, while the other one presents a concentrated mass m_p. The rod is freely to rotate around the hinge connected to a cart. The cart is constrained to move backward and forward along a limited horizontal axis. Once defined the kinematic constraints, let's write down the dynamic equation that describes the system. Figure 2.1-2 shows the free body diagram of the pendulum rod

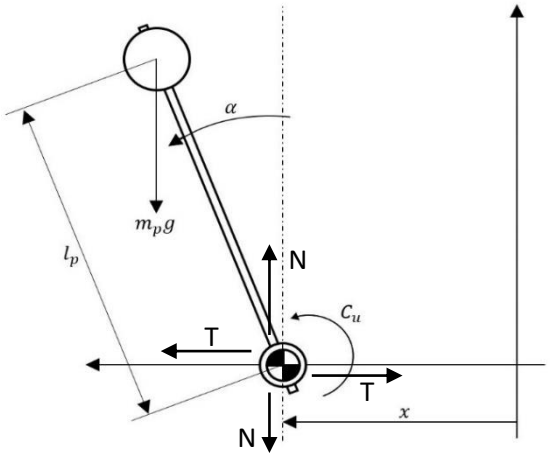


Fig. 2.1-2 Free body diagram of the pendulum

From now on, let's consider the following hypothesis

- mass of the rod negligible
- mass of the system (rod and final mass) concentrated in the top extremity of the pendulum
- no friction forces

The table (2.1-1) resumes the list of variables and parameters used to create the model

$\ddot{x}$	Linear acceleration of the cart	m/s^2
α	Angle of the rod with respect to the vertical axis	rad
$\ddot{\alpha}$	Angular acceleration of the rod	rad/s^2
C_u	Torque generated by the linear acceleration of the cart	Nm
J_p	Moment of inertia of the mass with respect to fulcrum	0,05 kgm^2
m_p	Concentrated mass of the pendulum	0,2 kg
g	Gravity acceleration	9,8 ms^2
l_p	Rod length of the pendulum	0,5 m

Table 2.1-1 Variables and parameters of the inverted pendulum

With these approximations, the equilibrium of the torques at the hinge is

$$J_p\ddot{\alpha}(t) = \underbrace{m_p g l_p \sin\alpha(t)}_{(1)} + \underbrace{C_u(t)}_{(2)} \tag{2.1-1}$$

The right term represents the time derivative of the angular momentum of the mass $\frac{dL_p}{dt} = \sum_{i=1}^{n} C_i$ and is equal to the product of the angular acceleration of the rotational system and the inertia moment $J_p = m_p l_p^2$.

The right terms, (2.1-1) and (2.1-2), are

- $m_p g l_p \sin\alpha(t)$, torque produced by the gravity, that is equal to the product of the gravitational force $m_p g$ and its arm $l_p \sin\alpha(t)$
- C_u, torque produced by the acceleration of the cart

Explicating the torque C_u in terms of the linear acceleration of the cart

$$C_u(t) = -m_p \ddot{x}(t) l_p \cos\alpha(t) \tag{2.1-2}$$

The equation obtained, substituting the relation (2.1-2) in equation (2.1-1), is

$$J_p\ddot{\alpha}(t) = m_p l_p^2 \ddot{\alpha}(t) = m_p g l_p \sin\alpha(t) - m_p \ddot{x}(t) l_p \cos\alpha(t) \tag{2.1-3}$$

Collecting in the terms on the right the factors $m_p l_p$ and simplifying them, the equation (2.1-3) becomes

$$l_p\ddot{\alpha}(t) = g \sin\alpha(t) - \ddot{x}(t) \cos\alpha(t) \tag{2.1-4}$$

The function that relates the torque induced by the gravity with respect to the angle α is nonlinear (there are the sin and cos functions). This nonlinearity can be neglected by considering only small values at the angle α. Therefore, under this assumption, α stays near the unstable equilibrium point $\alpha = \alpha_0 = 0$. The same approximation lays for the position $x = x_0 = 0$. In this way, the nonlinear terms of the equation (2.1-4) becomes

$$\sin\alpha(t) \sim \alpha(t)$$

$$\cos\alpha(t) = 1$$

Finally, the equation that describes the linearized system of the pendulum is

$$l_p\ddot{\alpha}(t) = g\alpha(t) - \ddot{x}(t) \tag{2.1-5}$$

2.2 Electromechanical System

The Electromechanical system is composed by the actuation and transmission system represented by motor, linear guide and his screw-screw nut mechanism, as can be seen in figure 2.2-1.

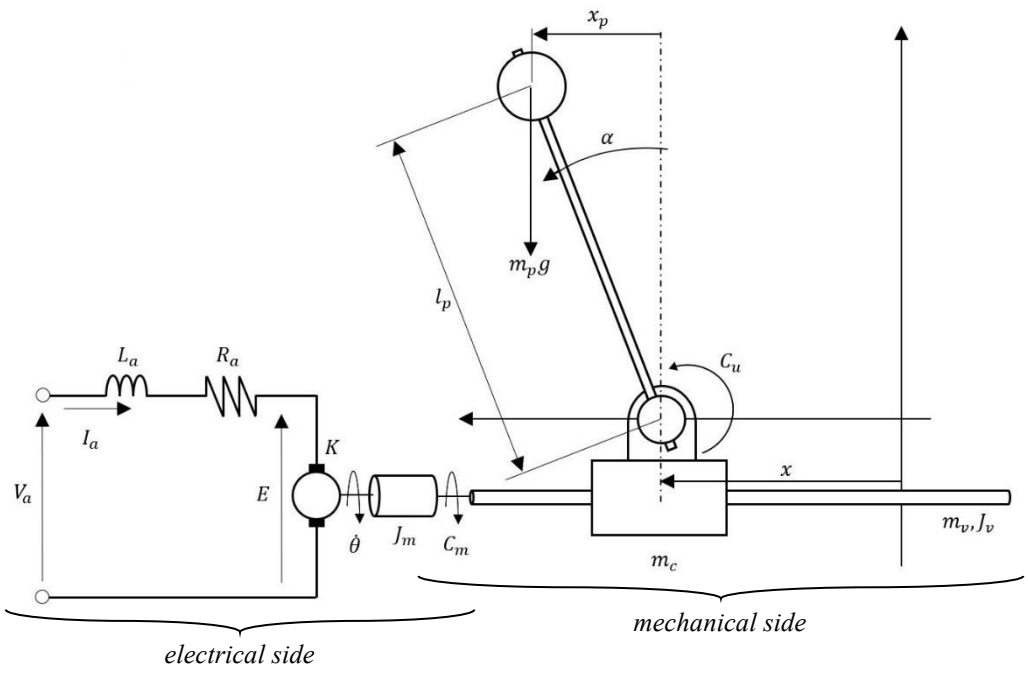


Fig. 2.2-1 Electromechanical system diagram

In the table (2.2-1), the variables and parameters of the transmission system are shown

x	Position of the cart	m
$\dot{x}$	Linear velocity of the cart	m/s^2
θ	Angular position of the motor shaft	rad
$\dot{\theta}$	Angular velocity of the motor shaft	rad/s
n	Angular velocity of the motor shaft	rpm
p	Pitch of the screw	$10\ mm$
η	Efficiency of the transmission system	0,96

Table 2.2-1 Variables and parameters of the transmission system

This is a linear equation, that depends on the two constant parameters p and η and on the transformation between the respective units of measurement, rpm and rad/s

$$n(t)[rpm] = \frac{30\,\dot{\theta}(t)[rad/s]}{\pi} \tag{2.2-1}$$

$$\dot{x}(t) = \frac{n(t)\,p}{60 \cdot 1000} = \frac{p}{2\pi \cdot 1000}\dot{\theta}(t) \tag{2.2-2}$$

The transmission system also imposes the relation between the torque generated by the motor and the linear force applied to the cart, that generates the linear acceleration $\ddot{x}$. This relation depends on the parameter η , that represents the efficiency of the transmission. The relation is

$$C_p(t) = F_p(t)\frac{p}{2\pi\eta}\,, \quad with \quad F_p(t) = m_p\ddot{x_p}(t) \quad and \quad x_p(t) = l_p \sin(\alpha) \tag{2.2-3}$$

The other part of the handling system is the motor, that constitutes the effective actuator of the whole system. In the table 2.2-2 the variables and parameters of the actuation system are shown

V_a	Armature voltage of the motor (command)	V
E	Back EMF (electromotive force) by self-inductance	V
I_a	Armature voltage of the motor	A
C_m	Output torque generated by the motor	Nm
F_p	Horizontal component of constraint reaction transmitted by pendulum to cart (due to the relative motion)	N
R_a	Armature resistance	$1{,}7\ \Omega$
K	Torque constant	$0{,}53\ Nm/A$
J_{tot}	Total inertia moment of the system with respect to the motor	$1{,}4727 \cdot 10^{-4}\ Kgm^2$
l_p	Rod length of the pendulum	$0{,}5\ m$

Table 2.2-2 Variables and parameters of the actuation system

The electric motor type is a synchronous brushless motor with permanent magnet, as presented in the previous chapter. From now on, the following assumptions are considered for the derivation of the model of the plant

1. Magnetic circuit of the electric motor is linear
2. Magnetic power losses in the iron are negligible
3. Friction forces in the motor are neglected
4. The armature inductance L_a, and all the transients are neglected in the model of the motor
5. Torque constant k_t equal to the back EMF constant k_e, and so $K = 0{,}53Nm/A$ is used

With these assumptions, the output torque of the DC motor linearly depends by the current armature I_a by the torque constant K through the following equation

$$C_m(t) = KI_a(t) \tag{2.2-4}$$

Thanks to the hypothesis 1, 2 and 3, the electric power of the input can be entirely converted into mechanical power by the electromechanics coupling.

Considering these assumptions, the following power balance can be derived

$$E(t)I_a(t) = C_m\dot{\theta}(t) \Rightarrow E(t)I_a(t) = KI_a(t)\dot{\theta}(t) \tag{2.2-5}$$

$$E(t) = K\dot{\theta}(t) \tag{2.2-6}$$

Therefore, thanks to the hypothesis 5, the proportional constant K, between torque and current, is worth also for the relation between the back EMF E and the angular velocity of the rotor shaft $\dot{\theta}$. In this way, the equations that represent the electrical side, of the electromechanical system, are

$$V_a(t) = R_aI_a(t) + E(t) \tag{2.2-7}$$

$$E(t) = K\dot{\theta}(t) \tag{2.2-8}$$

$$C_m(t) = KI_a(t) \tag{2.2-9}$$

On the other hand, the mechanical side, of the electromechanical system, can be modeled by the rotational force balance at the output shaft by the Euler formula:

$$C_m(t) - J_{tot}\ddot{\theta}(t) = C_p(t) \tag{2.2-10}$$

$$\text{with} \quad C_p(t) = F_p(t)\frac{p}{2\pi\cdot 1000\cdot\eta} = m_p\ddot{x}_p(t)\frac{p}{2\pi\cdot 1000\cdot\eta} \tag{2.2-11}$$

The parameter J_{tot} is derived as

$$J_{tot} = J_m + J_v + J_{cart} \tag{2.2-12}$$

The right terms, in equation (2.2-12) are respectively

- $J_m = 4.3 \cdot 10^{-4}\ Kgm^2$, moment of inertia of the rotor shaft
- $J_v = \frac{m_v r_v^2}{2} = 9{,}92 \cdot 10^{-5}\ Kgm^2$, moment of inertia of the transmission system (m_v, r_v mass and radius of the screw)
- $J_{cart} = (m_c + m_p) \cdot \left(\frac{p}{2\pi\cdot 1000}\right)^2 = 5{,}066 \cdot 10^{-6}\ Kgm^2$, moment of inertia of the assembly of the cart and pendulum components with respect to rotational axis of the electromechanical system, that is constituted by motor shaft and the screw

Finally, inserting the relation (2.2-11) in (2.2-10) and assuming $x_p(t) = l_p \sin(\alpha) \cong l_p \alpha(t)$ (considering the unstable equilibrium point, $\alpha = \alpha_0 = x = x_0 = 0$, as initial condition), the final equation of the mechanical side is

$$C_m(t) - J_{tot}\ddot{\theta}(t) = \left(m_p \frac{p}{2\pi \cdot 1000 \cdot \eta}\right) \ddot{x}_p(t) = \left(m_p l_p \frac{p}{2\pi \cdot 1000 \cdot \eta}\right) \ddot{\alpha}(t) \quad (2.2\text{-}13)$$

2.3 State space representation

With the equations obtained in the section 2.2, the problem can be traced to the solution of the following pair of linear systems differential equations, using the state space representation

$$\dot{\vec{X}}(t) = [A]\vec{X}(t) + [B]u(t)$$

$$\vec{Y}(t) = [C]\vec{X}(t) + [D]u(t)$$

The vector $\vec{X}(t) = \left[\, \alpha \; \dot{\alpha} \; \dot{\theta} \; x \; \dot{x} \,\right]^T$ is defined as the vector that contains the state variables of the system, while $\dot{\vec{X}}(t)$ is its time derivative. On the other hand, the term $u(t) = V_a(t)$ is the input vector and $\vec{Y}(t) = [\, \alpha \; x \,]$ the vector that contains the outputs x and α of the systems (this choice is understandable because these two quantities are directly measured with sensors). The matrices [A], [B], [C] and [D] are obtained by writing in matrix form the equations computed in the previous section, that are resumed below

$$lpl_p\ddot{\alpha}(t) = g\alpha(t) - \ddot{x}(t) \quad (2.3\text{-}1)$$

$$\dot{x}(t) = \frac{p}{2\pi \cdot 1000}\dot{\theta}(t) \quad (2.3\text{-}2)$$

$$V_a(t) = R_a I_a(t) + E(t) \quad (2.3\text{-}3)$$

$$E(t) = K\dot{\theta}(t) \quad (2.3\text{-}4)$$

$$C_m(t) = KI_a(t) \quad (2.3\text{-}5)$$

$$C_m(t) - J_{tot}\ddot{\theta}(t) = \left(m_p l_p \frac{p}{2\pi \cdot 1000 \cdot \eta}\right) \ddot{\alpha}(t) \quad (2.3\text{-}6)$$

The matrices [A] and [B] are obtained by the relation of $\dot{\vec{X}}(t)$. Firstly, using the equations 2.3-1 → 6, the relation of each state derivative is obtained as function of the states and input. Once that all the relations, for each state derivative, are computed, they are organized in order to highlight the coefficients that multiply the states and the input. The computed coefficients represent the matrices [A] and [B].

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ \frac{g}{W_c} & 0 & \frac{W_a \cdot K^2}{J_{tot} \cdot R_a \cdot W_c} & 0 & 0 \\ \frac{g}{W_a \cdot W_d} & 0 & \frac{l_p \cdot K^2}{W_a \cdot W_b \cdot W_d \cdot R_a} & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ \frac{g}{W_d} & 0 & -\frac{l_p \cdot K^2}{W_b \cdot W_d \cdot R_a} & 0 & 0 \end{bmatrix} \qquad B = \begin{bmatrix} 0 \\ -\frac{W_a \cdot K}{J_{tot} \cdot R_a \cdot W_c} \\ -\frac{l_p \cdot K}{W_a \cdot W_b \cdot W_d \cdot R_a} \\ 0 \\ \frac{l_p \cdot K}{W_b \cdot W_d \cdot R_a} \end{bmatrix}$$

Instead, computing the relation of each output as function of states and input, the matrices [C] and [D] can be obtained

$$C = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix} \qquad D = [\,0\,]$$

The parameters W_a, W_b, W_c and W_d are

$$W_a = \frac{p}{2\pi \cdot 1000}$$

$$W_b = \frac{m_p \cdot l_p \cdot W_a}{\eta}$$

$$W_c = l_p - \frac{W_a \cdot W_b}{J_{tot}}$$

$$W_d = 1 - \frac{l_p \cdot J_{tot}}{W_a \cdot W_b}$$

This is the most useful representation that describes a system in terms of input (armature voltage V_a) and output (position x and pendulum angle α).

However, at this point, a clarification is necessary. The state space representation, as mentioned above, is a useful mathematical representation of a system because it offers a description of the system in terms of the defined states and input. Therefore, the stability analysis can be initially performed by looking at the eigenvalues of matrix [A] and in this case it is called “Internal Stability” and it describes the behavior of a system neglecting any kind of external input. The state space representation is useful because it allows to study contemporarily the “Internal Stability” and the “External stability” by looking at the matrix [B] that multiplies the input of the system (the forced response). An Internal Stability analysis of the obtained state space representation has been carried out.

The eigenvalues λ_i of [A] matrix, shown in table 2.3-1, are the poles of the system

λ_1	$0 + i0$
λ_2	$0 + i0$
λ_3	$-1126 + i0$
λ_4	$-4{,}427 + i0$
λ_5	$+4{,}472 + i0$

Table 2.3-1 A matrix eigenvalues

The position of the eigenvalues, in the complex plane, determines if an eigenvalue can make the system stable or not. If at least one of the eigenvalues, of matrix [A], has a positive real part so this eigenvalue makes the system not internally stable. As can be seen in figure 2.3-1, one eigenvalue has real part greater than zero, so the stability condition is not verified and the system results unstable.

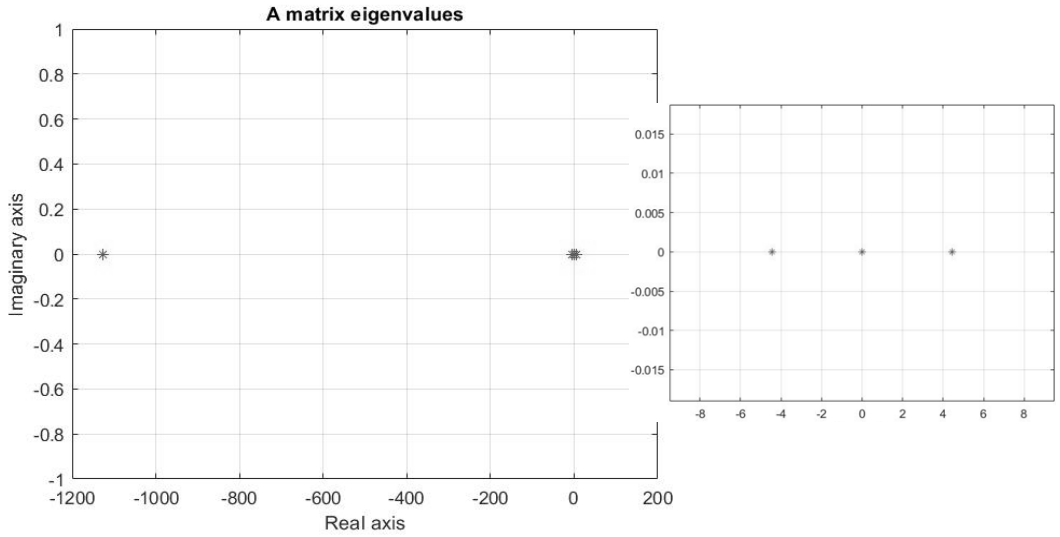


Fig. 2.3-1 Eigenvalues of the A matrix. Zoom on the right

In the next section, the transfer functions of the system are obtained from the state space representation and they describes the behavior of the system in terms of forced response by analyzing the outputs in presence of a given input (and so not considering the initial conditions for the states). However, it is important to remember that the transfer functions representation of a system is equal to the state space one if and only if the system is represented in minimal form and so only the minimal amount of variables that could describes the system's behavior is considered. The minimal form of the state space representation can be obtained by the Matlab command `'minimal'` applied to the command `'ss'` that gives to the designer the minimal realization of a system.

2.4 Simulink model and validation

Once compute the constitutive equations and obtained the state space representation, the Simulink model has been obtained. Figure 2.4-1 shows the block diagram of the system, obtained from equations 2.3-1 to 2.3-6.

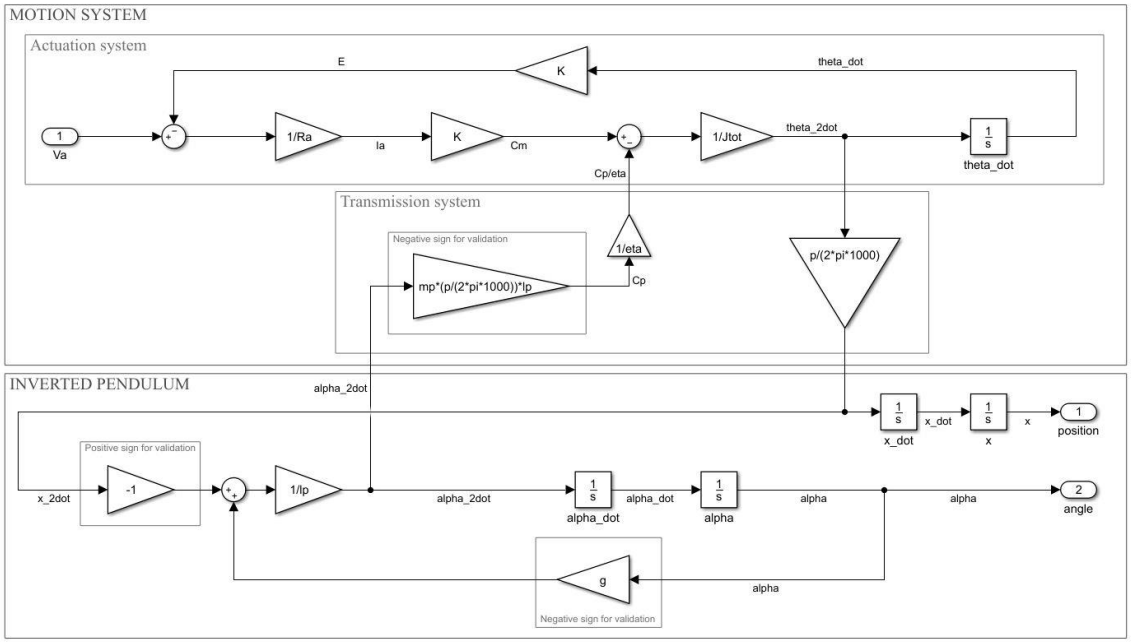


Fig. 2.4-1 Simulink model of the inverted pendulum system with electrical actuation

The system transfer functions can be obtained from the state space representation, using the relation $H(s) = [C](s[I] - [A])^{-1}[B] + [D]$. The H terms gives the possibility to evaluate the transfer functions between the input u and any of the outputs y ($H = y/u$). In this case, the transfer functions between the input V_a and the outputs α and x are important because they model the device's outputs for each possible input and they are used to design the proper controller that will stabilize the overall controlled system. Moreover it is possible to obtain the same results by using the Simulink command called 'Model linearizer' (in APPS section) that automatically evaluates the transfer function from a desired input to a desired output. The transfer functions $G_{V_a\alpha}$ and G_{V_ax} are

$$G_{V_a\alpha} = \frac{(2\pi 1000Kp^2\eta)s}{(m_p p^3 l_p R_a - J_{tot}(2\pi 1000)^2\eta l_p p R_a)s^3 - ((2\pi 1000)^2\eta l_p p K^2)s^2 + (J_{tot}(2\pi 1000)^2\eta g p R_a)s + ((2\pi 1000)^2 g K^2 p\eta)} =$$

$$= -\frac{6{,}7629s}{(s+1126)(s+4{,}427)(s-4{,}427)}$$

$$G_{V_ax} = \frac{(2\pi 1000Kp^2\eta)s}{(m_p p^3 l_p R_a - J_{tot}(2\pi 1000)^2\eta l_p p R_a)s^3 - ((2\pi 1000)^2\eta l_p p K^2)s^2 + (J_{tot}(2\pi 1000)^2\eta g p R_a)s + ((2\pi 1000)^2 g K^2 p\eta)} \cdot \frac{g - l_p s^2}{s^2} =$$

$$= \frac{3{,}3815}{s\,(s+1126)}$$

The analytical steps performed to evaluate $G_{V_a\alpha}$ and G_{V_ax}, are shown in Appendix C. In the next chapter a set of control techniques are used to stabilize and control the system. The transfer function $G_{\alpha x}$ was necessary to obtain the cascade control architecture. It is obtained through the relation $G_{V_ax}/G_{V_a\alpha}$ and its zero-pole gain form is

$$G_{\alpha x} = \frac{G_{V_ax}}{G_{V_a\alpha}} = \frac{g - l_p s^2}{s^2} = -\frac{0{,}5\,(s + 4{,}427)(s - 4{,}427)}{s^2}$$

The Bode diagrams of the $G_{V_a\alpha}$, G_{V_ax} and $G_{\alpha x}$ are shown in figures 2.4-2 and 3.

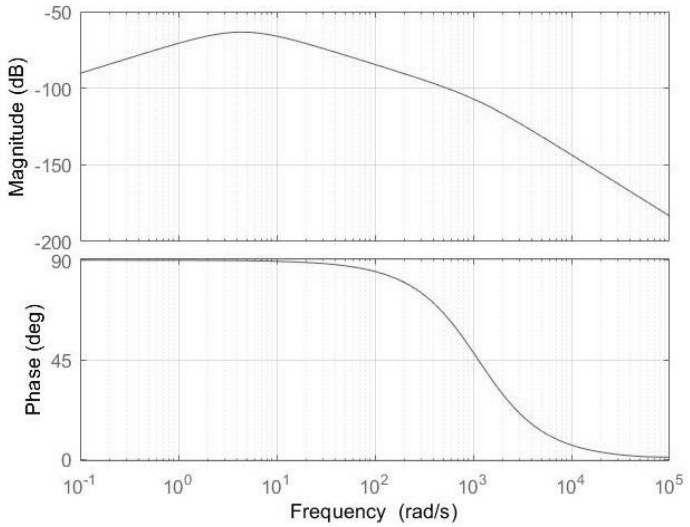


Fig. 2.4-2 Bode diagram of $G_{V_a\alpha}$

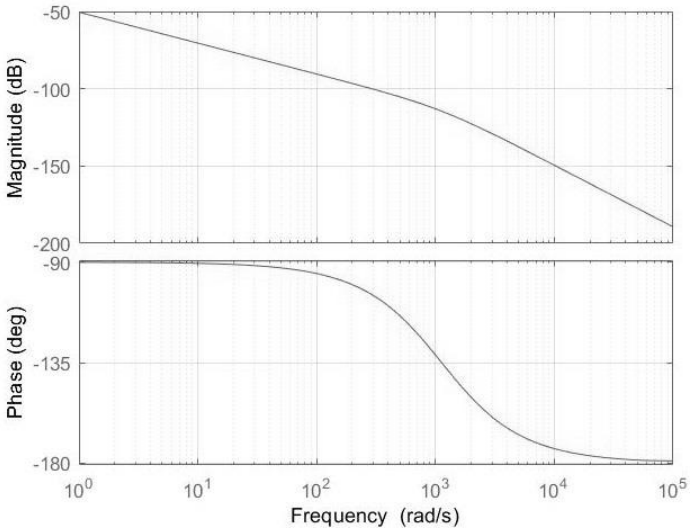


Fig. 2.4-3 Bode diagram of G_{V_ax}

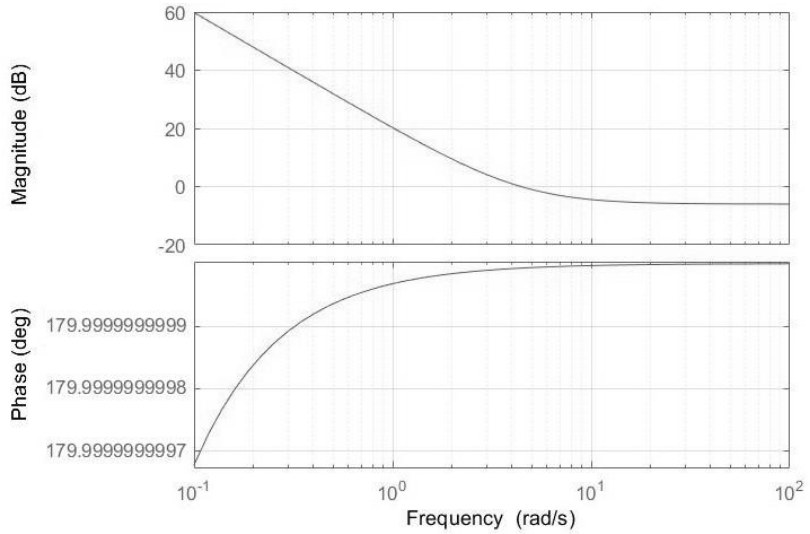


Fig. 2.4-4 Bode diagram of $G_{\alpha x}$

After this, to prove that the model and its constitutive equations are correct a validation test was performed. Looking at the Simulink model, in figure 2.4-1, there are three gains that contain the information about sign (- or +), this means that the test validation executes the check of the model in an inverted configuration. The validation Simulink model is shown in figure 2.4-5.

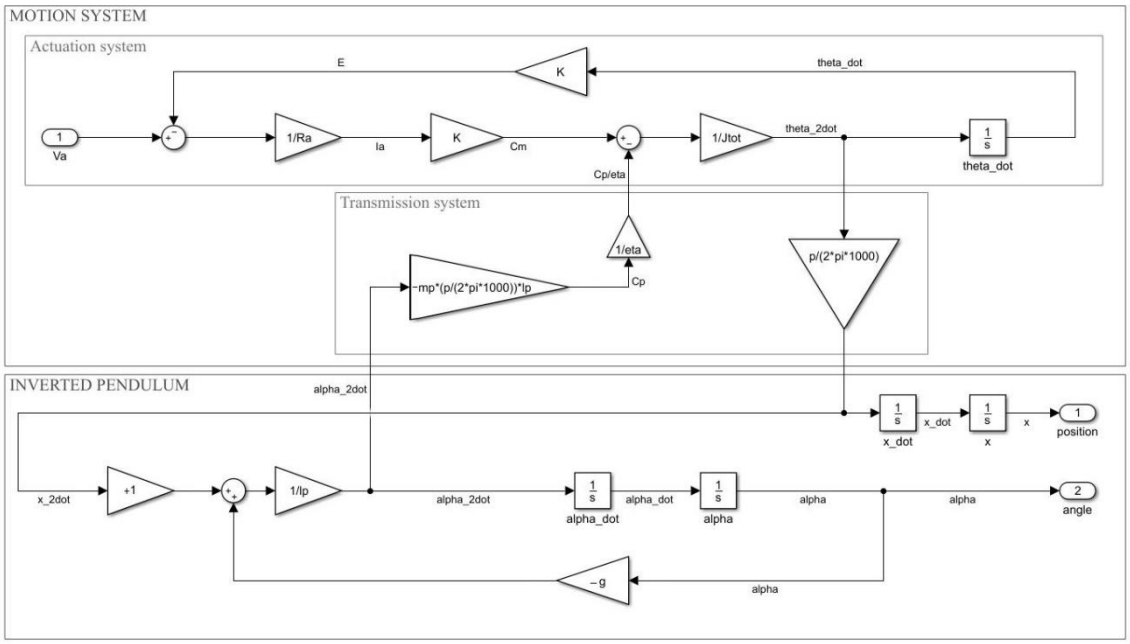


Fig. 2.4-5 Simulink model of the classical pendulum system with electrical actuation

The sign change leads to modify the constitutive equations (2.3-1 and 2.3-6)

$$l_p\ddot{\alpha}(t) = -g\alpha(t) + \ddot{x}(t)$$

$$C_m(t) - J_{tot}\ddot{\theta}(t) = -\left(m_p l_p \frac{p}{2\pi \cdot 1000 \cdot \eta}\right)\ddot{\alpha}(t)$$

and so the pendulum configuration, where the rod is under the cart like in figure 2.4-5.

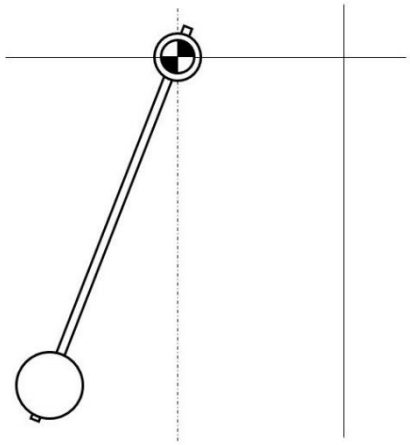


Fig. 2.4-6 Classical pendulum diagram for the test validation

As can be seen in figure 2.4-6, the test is done using an impulsive input V_a of $20V$ for one second, that acts at time zero.

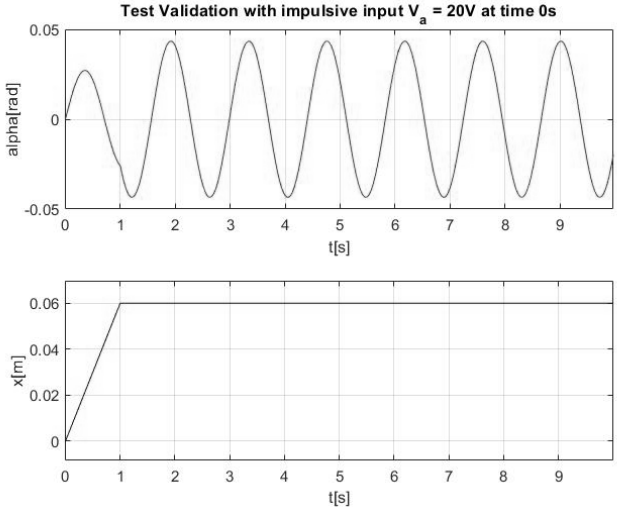


Fig. 2.4-7 Test validation with impulsive input V_a of $20V$ at time zero

The model results correct because, in the absence of friction forces, the pendulum moves along the x axis until the input returns to zero and so the rod stars to swing along the vertical axis in a perpetual motion.

This validation can be used also on the real Test bench. These types of experimental simulations allow to characterize the real damping of the pendulum.

3 Controllers design

Once obtained (state space and transfer functions) the mathematical representation of the system is obtained in Laplace domain a proper controller was designed in order to modify the whole transfer function and to control the behavior of the system. The goal is to suitably changes the value of poles in the transfer function (the roots of the denominator). Knowing the control stability theory, the characteristic polynomial of matrix [A] is stable if and only if its polynomial, on the denominator, is Hurwitz (the denominator has roots lying on the left half of the s-plane, practically all their real parts must have values equal or less than zero [1]). For the purpose of this work, this criterion and the Nyquist criterion were sufficient. The Nyquist criterion is used to obtain the final goal of this dissertation: the stabilization of the system. The following section allows a better comprehension of this stability criterion.

3.1 Nyquist stability criterion

In control theory and stability theory, the Nyquist's stability criterion is a graphical technique for determining the stability of a dynamical system. This criterion is very powerful because of its case of application. It is a graphical criterion that doesn't need an explicit computation of the poles and zeros of both the closed-loop or open-loop transfer functions of the system. A brief overview of this principle is explained in this section. For further explanations refer to control theory texts like [1].

This criterion is based on the Cauchy's argument principle applied to a specific transfer function of the form

$$F(s) = 1 + G(s)$$

where $G(s)$ represents a generic transfer function. In this case, $G(s)$ is the transfer function of the plant. $F(s)$ represents the polynomial at denominator of the closed loop transfer function of $G(s)$, that has the form

$$W(s) = \frac{G(s)}{1 + G(s)}$$

Therefore, the roots of $F(s)$ coincide with the poles of the closed loop system $W(s)$, while the roots of $F(s)$ coincides with the poles of the open loop system that are the poles of $G(s)$. By applying now the principle of the argument and noting that the Nyquist path of the $F(s)$ is the function $G(s)$ translated in $(1, i0)$, the following relation can be obtained

$$N_{-1}^{\circlearrowright} = Z_P - P_P$$

where $N_{-1}^{\circlearrowright}$ specifies the number of clockwise encirclements of the critical point $(1, i0)$ done by the directed Nyquist path of the function $G(s)$, while Z_P and P_P represent the number of closed

loop ($H(s)$) and open loop ($G(s)$) poles with positive real part. Negative values of $N_{-1}^{\circlearrowleft}$ corresponds to counterclockwise encirclements.

Nyquist's stability criterion states that, given a transfer function $G(s)$, if its transfer function of the closed loop system is stable if and only if the following relationship exists

$$N_{-1}^{\circlearrowleft} = -P_P$$

or alternatively, imposing that the closed loop system must has only poles with negative real part, the above equation becomes

$$N_{-1}^{\circlearrowleft} = P_P$$

that is equivalent to say that a system is stable in closed loop if the Nyquist diagram of the open loop transfer function of the controlled system performs a number of counterclockwise encirclements about the critical point $(1, i0)$ equal to the number of poles with a positive real part of the open loop transfer function of the uncontrolled system.

The following sections describe the application of this criterion on the pendulum transfer functions, in order to demonstrate the stabilization of the controlled system.

3.2 Controllers of electrical actuation model

Once that the stability criterion was chosen, different control techniques were tested in order to control the inverted pendulum system. The next sections present the architecture of the proposed control algorithms: PID and Static State feedback.

3.2.1 PID design: a brief introduction

The Proportional-Integrative-Derivative control (PID) is a negative feedback system widely used in industrial applications because of its simple integration and calibration in real systems, like PLC or microcontrollers. The controller acquires the value of the difference between a reference value and a measured one of the physical parameters that have to be controlled (commonly called tracking error). The controller reacts to positive or negative error by forcing it to asymptotically tend to zero. Moreover, the calibration of this compensator is easy and it is regulated by some empiric methods, like Ziegler-Nichols presented in the following sections. The classical form of the PID controllers is

$$C = K_p + K_i \frac{1}{s} + K_d \frac{N}{1 + N\frac{1}{s}}$$

where K_p, K_i and K_d are the Proportional, Integrative and Derivative coefficient, respectively. The N terms is called pre-Filter coefficient and it is needed because of the reliability of the derivative terms (originally the derivative term introduce a zero in the compensator transfer function but it is an unreliable controller because it has an improper transfer function).

The following sections (3.2.2 and 3.2.3) explain how these coefficients can be found. The final form of the controller was achieved thanks to the PID tuning tool of Simulink, being careful both on the physical limits (motor angular velocity) of the actuator and on the frequency response of the entire system (Bode diagrams), in terms of bandwidth and phase margin. For clearance of contents, let's remember that the stability margins are indexes able to express the robustness of a design control system from outer disturbances not considered in the initial model. They can be graphically evaluated from the Nyquist plot of $G(j\omega)$ considering the "distance" (in terms of linear distance along the real axis and in terms of angular phase) from the critical point $(-1; 0)$.

In this case of study, the desired gain margin has been fixed around 500 rad/s (for C_α calibration) or 1÷10 rad/s (for C_x calibration) and the desired phase margin at least equal to 40°.

3.2.2 PID design: cascade architecture

The system to be implemented needs a double control, C_x and C_α, on the two measured output, that are the pendulum angle and the cart position. The transfer function, that are implemented in the inner and outer loops, are $G_{V_a\alpha}$ and $G_{\alpha x}$. The problem can be solved by providing a suitable constant acceleration to the cart, to stabilize the unbalance by its inertial reactions. This principle, brilliantly exploited in the case of the Segway, requires an infinite trajectory, but in this case the available path before that the cart collides with the end of the linear guide, is 500 mm.

Therefore, the design of the controllers starts with the stabilization of the angle alpha. That constitutes the inner control loop of the whole final architecture. Whereas the position transfer function will be considered for the outer loop. Before starting, let's assume the cart is in the middle of the linear guide and the pendulum is perfectly vertical, $(x = 0, \alpha = 0)$, that is the unstable equilibrium point of the system.

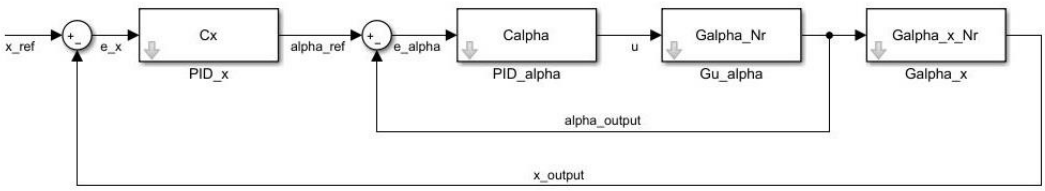


Fig. 3.2-1 PID cascade architecture of the control model

Looking at the figure above (3.2-1) when the pendulum is in vertical position, the error at the input of the first compensator (PID_x) is zero, and such as the output from the same block. The

output of PID_x is called α_{ref} and it constitutes the reference signals for the inner loop, that has to stabilize the pendulum angle α. The signal α_{ref} is compared with that one coming from the angular transducer in order to compute the resulting error, that is compensated by PID_α. The output of PID_α will be the command $u = V_a$ for the motor. Consequently, whenever there is a position error on the cart ($x_{ref} - x_{feed} \neq 0$) there will be a resulting error on the input of the inner loop $\alpha_{ref} \neq 0$. In this condition the rod position is modified to restore its vertical position, thus achieving $x_{ref} - x_{feed} = 0$ and therefore $\alpha_{ref} = 0$.

Design of the inner loop: angle compensation

Hence, knowing the generalities of the cascade control and considering the inner loop of the architecture let's control the transfer function $G_{u\alpha}$. The poles of the transfer function are

$$p_1 = -1126$$

$$p_2 = +4.427$$

$$p_3 = -4.427$$

$G_{u\alpha}$ has one pole with a positive real value, that proves the instability of the uncontrolled system. Looking at the figure 3.2-2, the Nyquist path of $G_{u\alpha}$ does not present encirclements around the critical point $(-1; 0)$, and, since it presents an unstable pole, the closed loop stability condition is not satisfied.

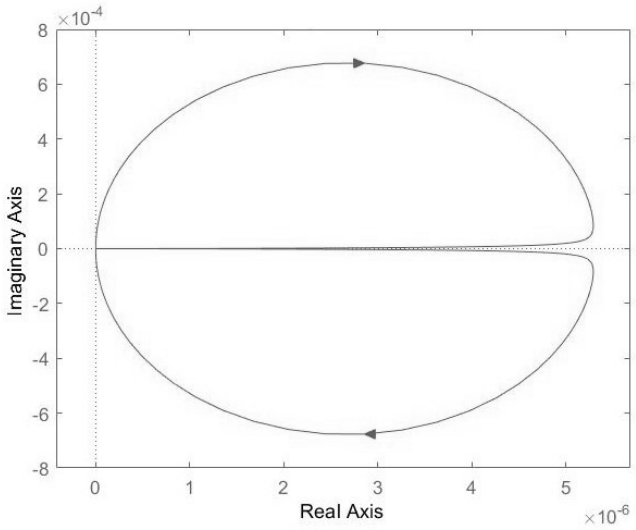


Fig. 3.2-2 Nyquist diagram of $G_{u\alpha}$

The relationship $N_{-1}^{\curvearrowleft} = -P_P = -1$ must be satisfied to stabilize the system, so that given a controller C_α, the Nyquist path of $C_\alpha G_{u\alpha}$ performs a counterclockwise encirclement around the critical point $(-1; 0)$.

Starting with a P compensator, K_p, let's analyze the Nyquist's path deformation of the open loop transfer function $K_p G_{u\alpha}$. Figure 3.2-3 shows several Nyquist path for different values of K_p.

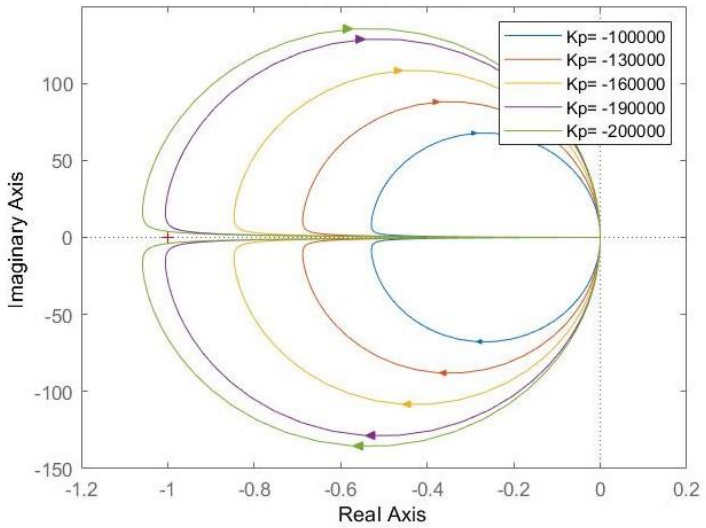


Fig. 3.2-3 Nyquist diagram of $K_p G_{u\alpha}$ for different values of K_p

As can be seen in figure 3.2-3, it is impossible to achieve the system stability with only the proportional coefficient K_p. The same occurs with a PD compensator. As it can be seen in figure 3.2-4, in the second case, two clockwise encirclements are obtained instead of one counterclockwise.

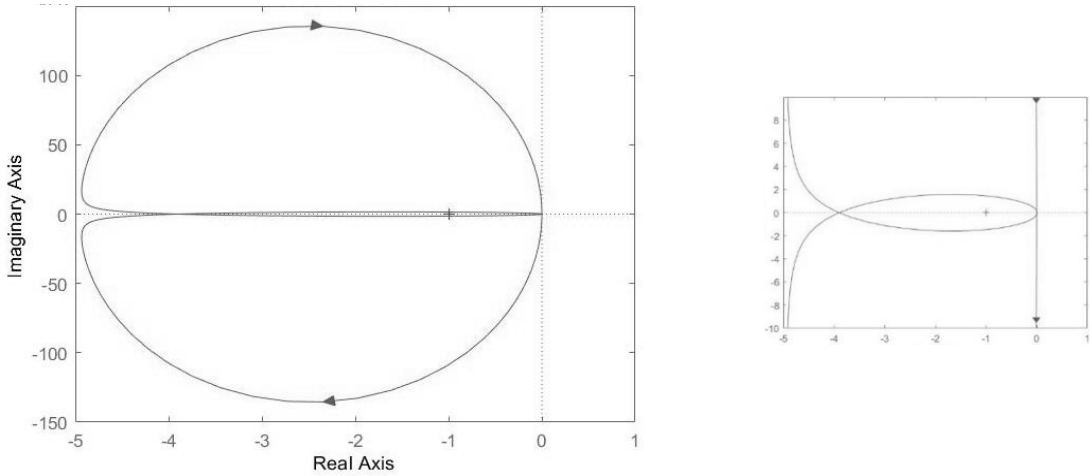


Fig. 3.2-4 Nyquist diagram of $C_{K_p K_d} G_{u\alpha}$, with $K_p = -200000, K_d = -650$ and $N = 550000$, on the left. Zoom around the critical point $-1 + i0$, on the right

The same problem can be observed using a PI compensator, $C_{K_pK_i}$, that generates a clockwise encirclement of the critical point as can be seen in figure 3.2-5.

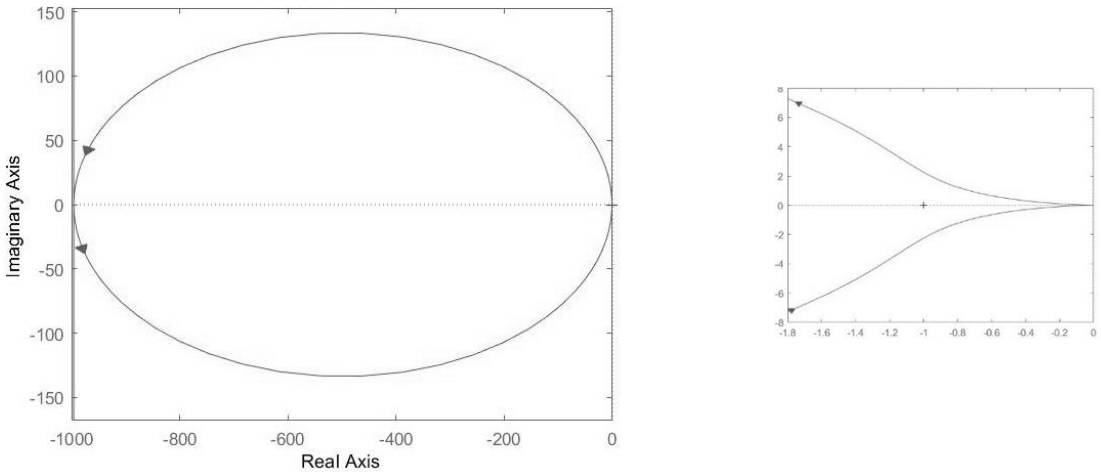


Fig. 3.2-5 Nyquist diagram of $C_{K_pK_i}G_{u\alpha}$, with $K_p = -200000, K_i = -3250000$, on the left. Zoom around the critical point $-1 + i0$, on the right

These tests reveal that a PID compensator is necessary. The final values of the three coefficients, K_p, K_d and K_i are chosen thanks to the PID tuning tool of Simulink, and so the final form of the controller is

$$C_\alpha = -13061.51 - \frac{204997.97}{s} + 60.47\frac{215.97s}{s + 215.97}$$

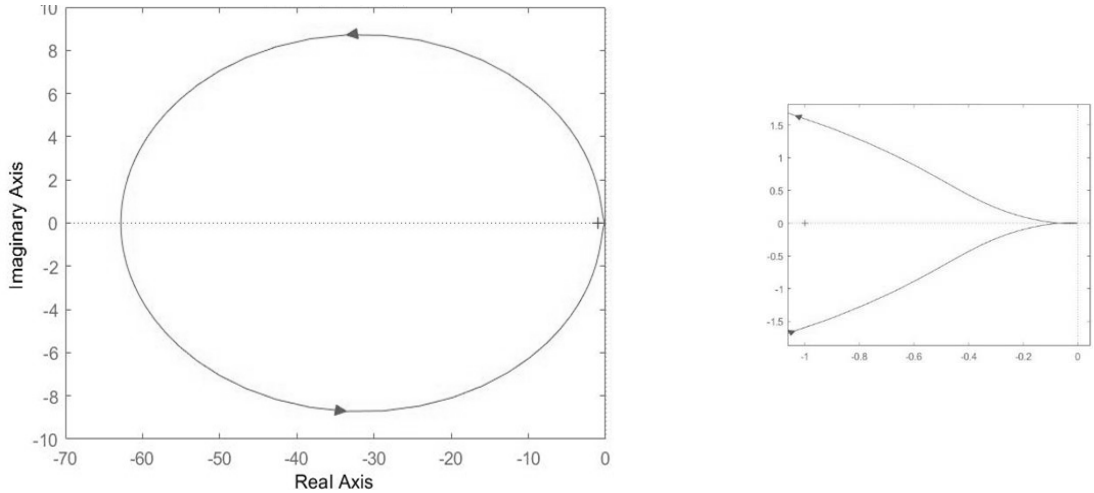


Fig. 3.2-6 Nyquist diagram of $C_\alpha G_{u\alpha}$, on the left. Zoom around the critical point $-1 + i0$, on the right

As can be seen in figure 3.2-6, the PID compensator is able to modify the Nyquist path in order to circulate the critical point one time in the counterclockwise direction. This fact is the proof of the system stability but, remembering the angle transfer function

$$G_{u\alpha} = \frac{-6.7629s}{(s+1126)(s+4.427)(s-4.427)}$$

and multiplying $G_{u\alpha}$ by the PID compensator,

$$C_\alpha = -13061.51 - \frac{204997.97}{s} + 60.47\frac{215.97s}{s+215.97}$$

the controller pole in 0 simplifies the root in zero at the numerator of $G_{u\alpha}$. This cancellation could compromise the stability. Therefore is necessary to analyze the poles of the closed loop transfer function of $C_\alpha G_{u\alpha}$

$$p_1 = -1145$$

$$p_2 = -89.5756 - i89.0353$$

$$p_3 = -89.5756 + i89.0353$$

$$p_4 = -17.834$$

All the poles have negative real part, so the system is asymptotically stable. Looking at the Bode diagrams of the open loop transfer function $C_\alpha G_{u\alpha}$, the frequency response, in terms of stability margins, can be evaluated (see figure 3.2-7).

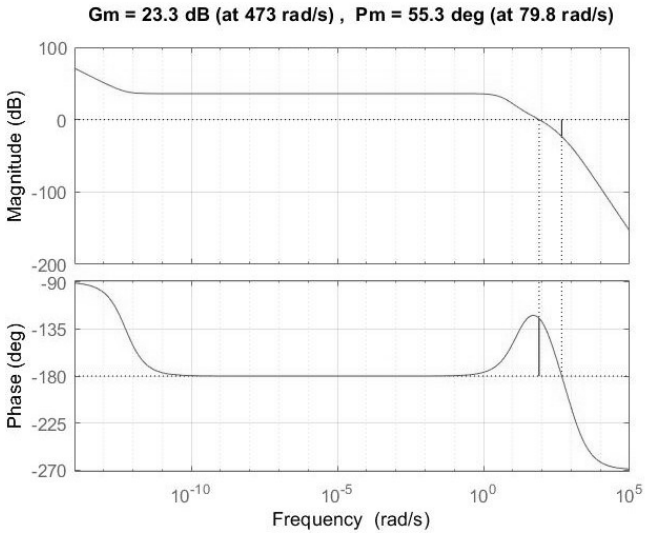


Fig. 3.2-7 Bode diagrams of $C_\alpha G_{u\alpha}$

As can be grasp from the Bode diagrams, the system achieves good performances in terms of frequency response. The system presents a good phase margin ($55{,}3°$) and an optimal gain margin ($23{,}3dB$) at $473\ rad/s$. The obtained results respect the specification presented in section 3.2.1.

Design of the outer loop: position compensation

Once the inner loop is stabilized, the outer loop transfer functions have to be stabilized. The first step is to substitute the inner loop with the respective closed loop transfer function $G_{u\alpha_{cl}}$, that will multiply the transfer function $G_{\alpha x}$ as can be seen in the Simulink model presented in 3.2-8.

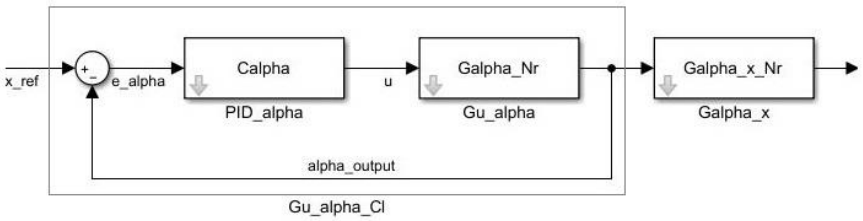


Fig. 3.2-8 Uncontrolled outer loop architecture

The outer closed loop transfer function $C_\alpha G_{u\alpha}$ can be compute by using the Matlab command `'feedback'`

$$G_{u\alpha_{cl}} = \frac{C_\alpha G_{u\alpha}}{1 + C_\alpha G_{u\alpha}} = \frac{8.6112 * 10^{-11}(s + 2.37 * 10^{17})(s + 14.63)}{(s + 1145)(s + 17.83)(s^2 + 179.2 + 14330)}$$

Therefore, the plant transfer function for the outer loop has to be computed and it is done multiplying $G_{u\alpha_{cl}}$ by $G_{\alpha x}$

$$G_{ux_{ol}} = G_{u\alpha_{cl}} G_{\alpha x} = \frac{-4.3065 \cdot 10^{-11}(s + 2.37 * 10^{17})(s + 14.63)(s + 4.427)(s - 4.427)}{s^2(s + 1145)(s + 17.83)(s^2 + 179.2 + 14330)}$$

The poles of the new transfer function are

$$p_1 = -1145$$

$$p_2 = -89.5756 - i89.0353$$

$$p_3 = -89.5756 + i89.0353$$

$$p_4 = -17.834$$

$$p_5 = 0$$

$$p_6 = +5{,}616 \cdot 10^{-13}$$

Although p_6 is very close to zero, it is a pole with positive real part and so it could create instability in the new plant $G_{ux_{ol}}$. A proof is the Nyquist path of $G_{ux_{ol}}$in figure 3.2-9.

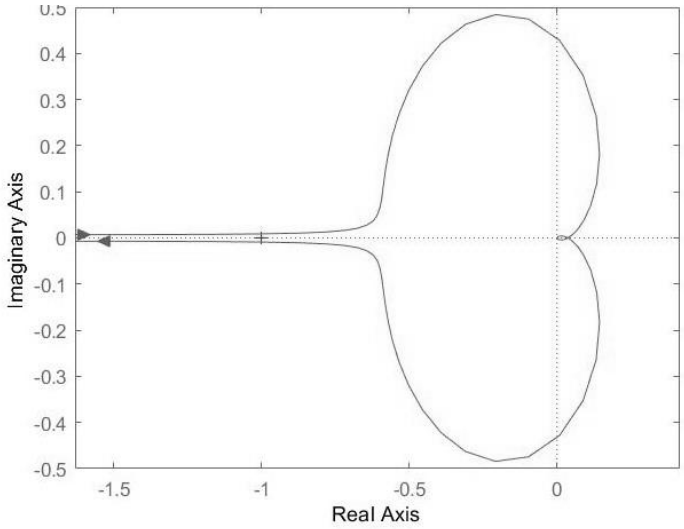


Fig. 3.2-9 Nyquist diagram of $G_{ux_{ol}}$

As confirmed by the Nyquist path, the stability criterion is violated, because the path circulates two times clockwise around the critical only one time in the counterclockwise direction, thus a PID compensator is needed.

The PID coefficients are chosen in order to guarantee the frequency response specification (gain margin at 1÷10rad/s and a phase margin equal or greater than 40°). A controller, stabilizing the plant transfer function and to respect the frequency specification, is obtained by the PID tuning tool of Simulink and it has the form

$$C_x = \frac{0.073639(s + 0.06284)(s + 0.05082)}{s(s + 0.9375)}$$

with the PID coefficients equal to

$$K_p = 8.6599 \cdot 10^{-3} \qquad K_d = 69.3088 \cdot 10^{-3}$$
$$K_i = 250.8324 \cdot 10^{-6} \qquad N = 937.5267 \cdot 10^{-3}$$

The Nyquist path of the new open loop transfer function $C_x G_{ux_{ol}}$ and its relative Bode diagrams are presented in 3.2-10 and 3.2-11, respectively.

$$C_x G_{ux_{ol}} = \frac{-3.1 \cdot 10^{-13}(s + 0.0628)(s + 0.0528)(s + 4.427)(s - 4.427)(s + 14.63)(s + 2.376 \cdot 10^{17})}{s^3(s + 0.9375)(s + 1145)(s + 17.83)(s^2 + 179.2 + 14330)}$$

$$p_1 = -1145 \qquad p_5 = -0.9375$$

$$p_2 = -89.5756 - i89.0353 \qquad p_6 = 0$$

$$p_3 = -89.5756 + i89.0353 \qquad p_7 = 0$$

$$p_4 = -17.834 \qquad p_8 = 0$$

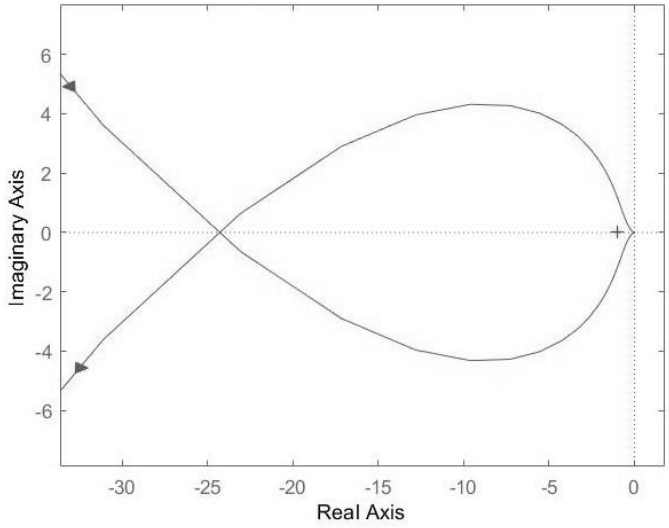


Fig. 3.2-10 Nyquist diagram of $C_x G_{ux_{ol}}$

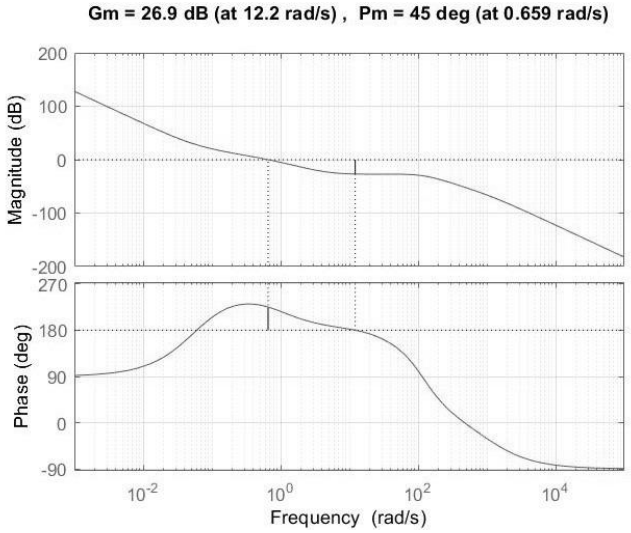


Fig. 3.2-11 Bode diagrams of $C_x G_{ux_{ol}}$

Looking at figure above (3.2-10), the stability of the system is achieved because the Nyquist path presents two encirclement, respectively clockwise and counterclockwise, that result in zero encirclements of the critical point, according with the number of unstable poles presented in the open loop transfer function. Moreover, also the frequency response specifications are achieved as can be seen in figure 3.2-11.

3.2.3 PID design: parallel architecture

As declared in the previous chapters, different control techniques have been tested for the plant transfer function. The next architecture is the parallel one.

In the parallel architecture, two loops compare the α_{feed} and x_{feed} (output feedbacks) with x_{ref} and α_{ref} (reference signals). Therefore, the two resulting errors (α_{err} and x_{err}) become the inputs of the PID controllers and their outputs are summed to generate the input command u of the plant. In this way, the characteristic equations of the control scheme are necessary to find the PID constants K_p, K_i and K_d.

The Simulink model of the parallel architecture is shown in figure 3.2-12.

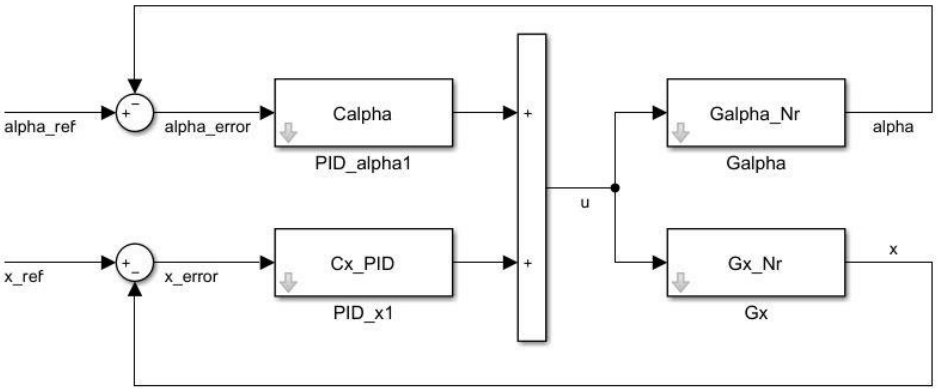


Fig. 3.2-12 PID parallel architecture of the control model

Let's call the PID controllers as C_α and C_x , the relations between the inputs of the controllers and the plant input u are presented below

$$u_\alpha = C_\alpha \cdot \alpha_{err} \qquad (3.2.3\text{-}1)$$

$$u_x = C_x \cdot x_{err} \qquad (3.2.3\text{-}2)$$

$$u = u_x + u_\alpha \qquad (3.2.3\text{-}3)$$

with $\alpha_{err} = \alpha_{ref} - \alpha_{feed} = -\alpha_{feed}$ and $x_{err} = x_{ref} - x_{feed}$. Remembering the plant transfer functions

$$G_{u\alpha} = \frac{\alpha_{feed}(S)}{u(S)} \qquad (3.2.3\text{-}4)$$

$$G_{ux} = \frac{x_{feed}(S)}{u(S)} \qquad (3.2.3\text{-}5)$$

and substituting the equations 3.2.3-4 and 5 in the equation 3.2.3-3, the relation that expresses the input u as a function of the two references is obtained

$$u = \frac{C_\alpha \alpha_{ref} + C_x x_{ref}}{1 + C_\alpha G_{u\alpha} + C_x G_{ux}} \qquad (3.2.3\text{-}6)$$

A system with two references is considered as MIMO system (multiple-input multiple-output), that is difficult to stabilize with common control techniques. For this reason, an useful assumption was done to reduce the complexity of the control problem. A constant reference for the α loop is considered and it is set to zero ($\alpha_{ref} = 0$ is the pendulum equilibrium point). In this way, the system become a SIMO system, and its input is x_{ref}.

Therefore, using the equations 3.2.3-5 and 3.2.3-6 and assuming $\alpha_{ref} = 0$, the closed loop transfer function between x_{ref} and x_{feed} is

$$G_{x_{cl}} = \frac{N_{cl}}{D_{cl}} = \frac{C_x G_{ux}}{1 + C_\alpha G_{u\alpha} + C_x G_{ux}} \qquad (3.2.3\text{-}7)$$

The characteristic equations of the parallel architecture are useful to understand the PID coefficients selection process. A good starting point is the design of the controller C_α, considering the loop shown in figure 3.2-13.

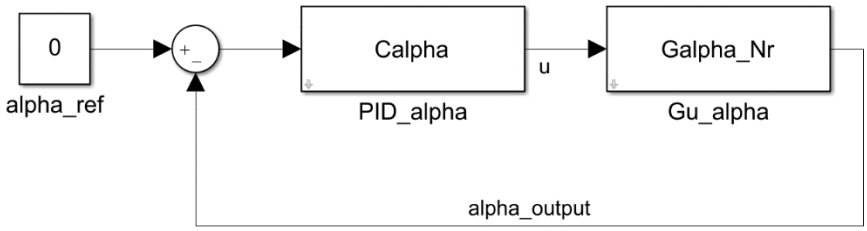


Fig. 3.2-13 Simulink scheme for C_α design

The scheme shown in figure 3.2-13 is equal to the inner loop of the cascade control studied in the previous section. For simplicity, the same controller is adopted in this case and its form is

$$C_\alpha = -13061.51 - \frac{204997.97}{s} + 60.47\frac{215.97s}{s + 215.97}$$

Now, considering only the controller C_α, let's analyze the open loop transfer function of the uncontrolled system for x, that is

$$G_{x_{ol,un}} = \frac{N_{ol,un}}{D_{ol,un}} = \frac{G_{ux}}{1+C_\alpha G_{u\alpha}} \qquad (3.2.3\text{-}8)$$

In this way, the stability problem is related to the roots of $D_{ol,un}$, that represent the poles of the uncontrolled system $G_{x_{ol,un}}$

$$p_1 = -1145$$

$$p_2 = -89.5756 - i89.0353$$

$$p_3 = -89.5756 + i89.0353$$

$$p_4 = -17.834$$

$$p_5 = 0$$

Looking at the poles, $G_{x_{ol,un}}$ has a pole in zero, that could compromise the stability. A proof is the Nyquist path of $G_{x_{ol,un}}$ in figure 3.2-14.

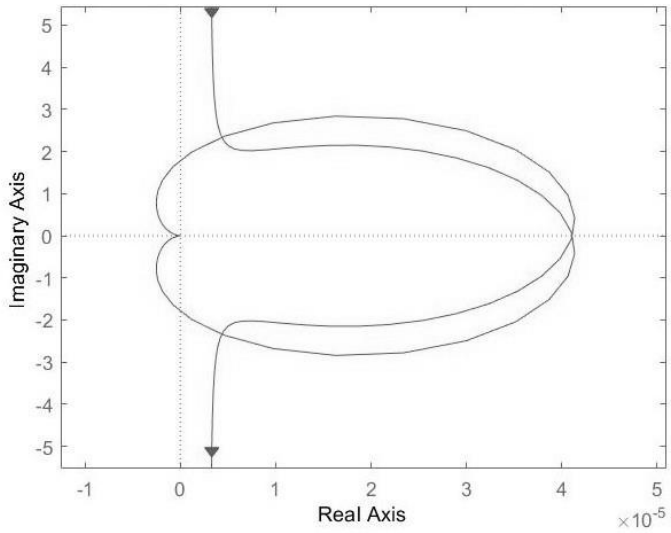


Fig. 3.2-14 Nyquist diagram of $G_{x_{ol,un}}$

Looking at the Nyquist path in figure above, $G_{x_{ol,un}}$ results unstable because the presence of p_5 creates an clockwise encirclement about the critical point $(-1; 0)$, that makes the system unstable on the basis the Nyquist criterion.

Therefore, the goal of the PID controller C_x is to moves all the poles, of the closed loop transfer function $G_{x_{cl}}$, to the left half of the s-plane in order to guarantee stability.

Noting that the closed loop transfer function $G_{x_{cl}}$ can be stabilized by a proportional coefficient $K_p < 0$, this coefficient can be found with the aid of Ziegler-Nichols method.

The Ziegler-Nichols method is a heuristic technique used for the PID controllers tuning. It was developed by John G. Ziegler and Nathaniel B. Nichols and it is performed by setting the K_i and K_d gains to zero. The K_p gain is then increased (from zero) until it reaches the ultimate critical gain K_c which is the largest gain at which the output of the control loop has stable and consistent oscillations. Higher gains, than the critical one K_c, have diverging oscillation and the oscillation period T_c is then used to set the K_p, K_i, and K_d coefficients, depending on the controller type used and the desired behavior. The table below shows the Ziegler-Nichols relation between K_p, K_i, and K_d with K_c and T_c.

Controller type	K_p	T_i	T_d	K_i	K_d
P	$0.5K_c$	-	-	-	-
PI	$0.45K_c$	$T_c/1.2$	-	K_p/T_i	-
PD	$0.8K_c$	-	$T_c/8$	-	K_dT_d
PID	$0.6K_c$	$T_c/2$	$T_c/8$	K_p/T_i	K_dT_d

Table 3.2-1 Ziegler-Nichols parameters for the desired controller type

Therefore, following the procedure explained above, several attempts have been carried out for values of $K_c \in \{0, -\infty\}$ thanks to the simulation environment of Matlab/Simulink. The value of the critical gain is reached around $K_c = -24000$. Using this value, the output response of the system performs constants oscillation with a period $T_c = 0.15\ s$, the resulting coefficients for the

$$K_p = -14400 \qquad K_d = 270$$
$$K_i = -192000 \qquad N = 355.5556$$

controller C_x are

The resulting controller is able to stabilize the closed loop transfer function $G_{x_{cl}}$

$$G_{x_{cl}} = \frac{2.7593 \cdot 10^5 (s + 4.427)(s - 4.427)(s + 10.99)(s - 76.09)(s + 216)}{(s + 182.3)(s^2 + 0.2934s + 19.46)(s^2 + 24.07s + 403.2)(s^2 + 1491s + 6.833 \cdot 10^5)}$$

A graphical proof is given by the Nyquist path in figure 3.2-15, of the open loop transfer function of the controlled system

$$G_{x_{ol}} = \frac{C_x G_{ux}}{1 + C_\alpha G_{u\alpha}} = \frac{2.7593 \cdot 10^5 (s + 4.427)(s - 4.427)(s + 10.99)(s - 76.09)(s + 216)}{s^2 (s + 355.6)(s + 1145)(s + 17.83)(s^2 + 179.2s + 1.443 \cdot 10^4)}$$

$$p_1 = 0$$

$$p_2 = 0$$

$$p_3 = -89.5756 - i89.0353$$

$$p_4 = -89.5756 + i89.0353$$

$$p_5 = -17.834$$

$$p_6 = -355.56$$

$$p_7 = -1145$$

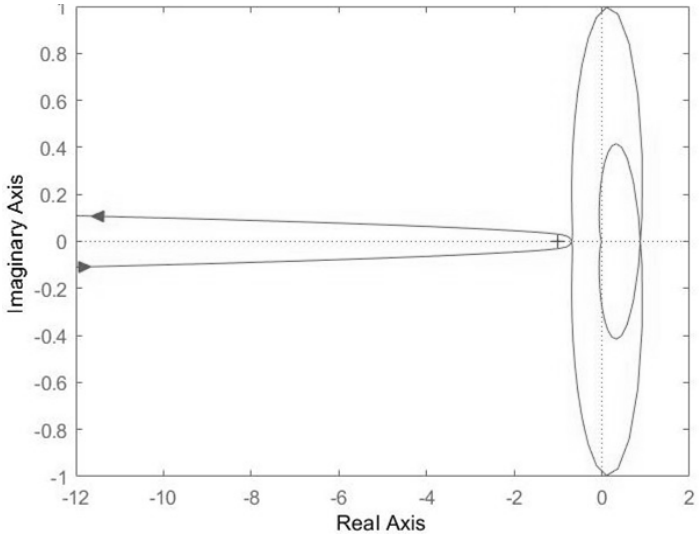


Fig. 3.2-15 Nyquist diagram of $G_{x_{ol}}$

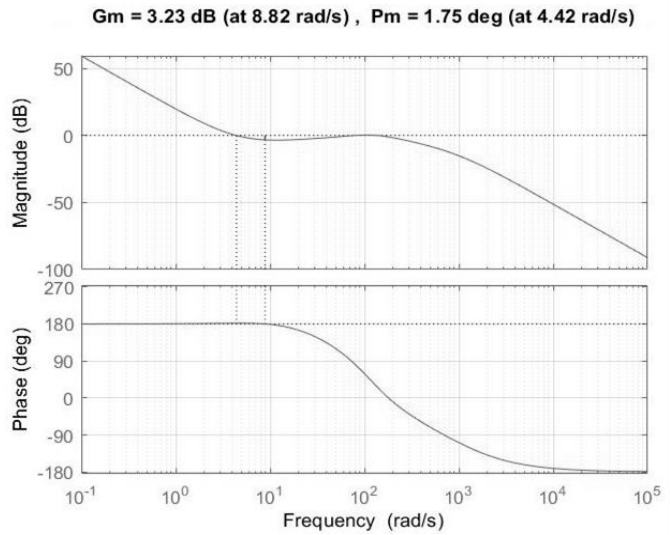


Fig. 3.2-16 Bode diagrams of $G_{x_{ol}}$

As can be seen in figure 3.2-15, there are no encirclements of the critical point $-1 + i0$ and so the system results stable for the Nyquist criterion.

So, let's have a look on the Bode diagrams of $G_{x_{ol}}$ in order to verify if the specifications are achieved with the Ziegler-Nichols control. As can be seen in figure 3.2-16, the controller C_x is not able to achieve good performances in terms of frequency response, due to low stability margins. An optimal controller, able to achieve the desired specifications, can be obtained using the PID tuning tool of Simulink. The resulting final controller C_x, the Nyquist and Bode diagrams of the open loop transfer function are

$$C_x = -19875.018 - \frac{4067.0793}{s} + 7058.012\frac{2.816s}{s + 2.816}$$

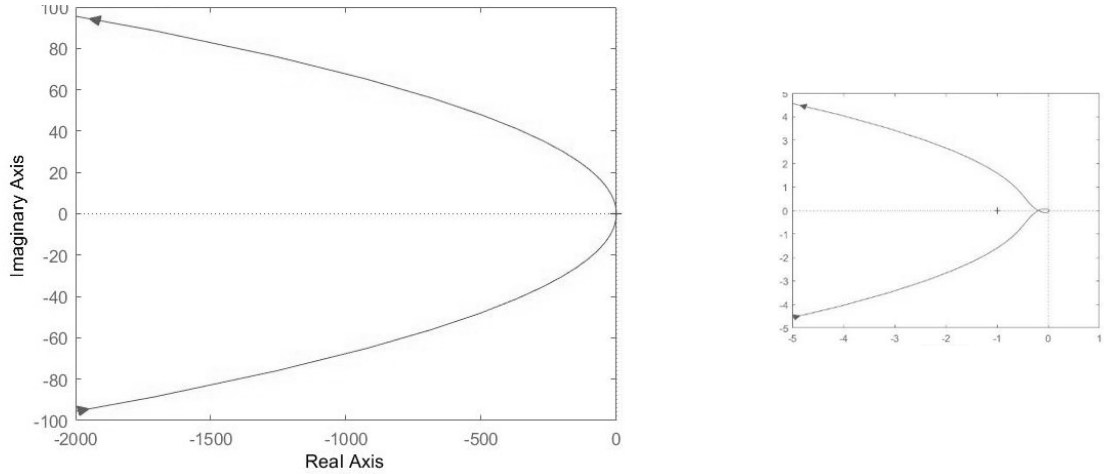


Fig. 3.2-17 Nyquist diagram of $G_{x_{ol}}$, on the left. Zoom around the critical point $-1 + i0$, on the right

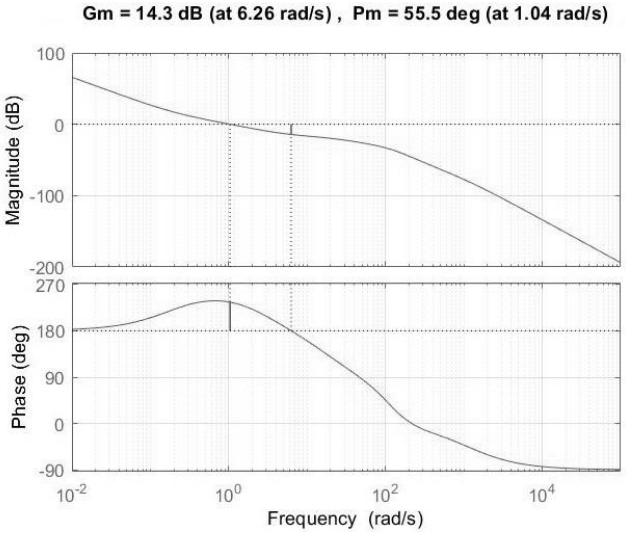


Fig. 3.2-18 Bode diagrams of $G_{x_{ol}}$

3.2.4 Static state feedback design: a brief introduction

Before starting to design a static state feedback control, some clarifications, on how this technique can be applied to a continuous time system, must be provide. Let's resume the state space representation of the system presented in chapter 2.3

$$\dot{\vec{X}}(t) = [A]\vec{X}(t) + [B]u(t)$$

$$\vec{Y}(t) = [C]\vec{X}(t)$$

with $\vec{X}(t) = \left[\, \alpha \; \dot{\alpha} \; \dot{\theta} \; x \; \dot{x} \,\right]^T$ the states vector and $A \in \mathbb{R}^{n\times n}$, $B \in \mathbb{R}^{n\times 1}$ and $C = \mathbb{R}^{2\times n}$, with $n = 5$, the state space matrices (in this case $[D] = 0$ is imposed). This particular representation of a system is very useful for control design purposes, the state space representation concerns a description of the time behavior for a dynamic system in term of matrices A, B, C and the available output variables. These equations can be translated in Simulink's block scheme as shown in figure 3.2-19.

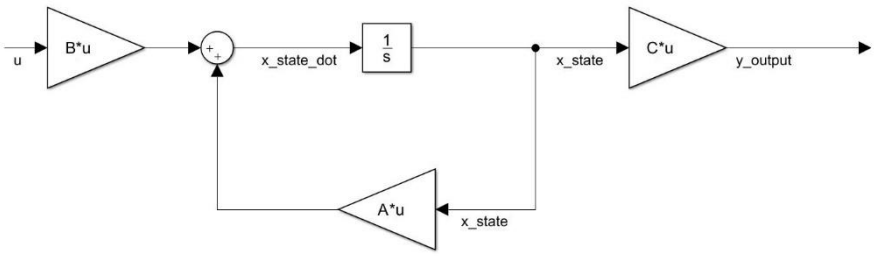


Fig. 3.2-19 Simulink block's scheme of the state space representation

Here a question comes. Can the poles be modified, in terms of their location on the s-plane, in order to modify the behavior of a system? Or even, can a system, initially unstable, be forced to a stability condition through a modification of its poles? The static state feedback, using the pole placement technique, help us to solve this design problem. This control strategy is based on a negative feedback branch, the states of the system are fed back, as the name implies, to the control input u, and they are multiplied by a gains matrix K computed in order to force the closed loop poles to lie in the desired positions. With this assumption, the control law of the architecture becomes

$$u(t) = -[K]\vec{X}(t) = -\left(\sum_{i=1}^{n} K_i X_i(t)\right)$$

The $1 \times n$ matrix K is called state feedback gain matrix.

Initially, let's assume that all state variables are available for feedback. Before starting the design phase, some considerations on the state space representation must be done. Firstly, the system, as it's defined, contains five states, but looking the equations described in section 2.3, it easily be noted that only four variables are independent. Two variables, the position of the cart x and the angular velocity of the motor $\dot{\theta}$, are linearly dependent due to the coefficient induced by the transmission mechanism between the motor and the cart-pendulum system. A confirmation of that can be achieved by the attribute `'minimal'` applied to the command `'ss'` of Matlab that gives to the designer the minimal realization of a system. Once clarified that, let's start the design phase.

Now a second question comes. Is it always possible to force the poles of a system in a desired position? Naturally the answer to this question is negative in fact arbitrary pole placement for a given system is possible if and only if the system is completely state controllable. The controllability condition can be summarized as follows, a system is completely controllable if and only if the controllability matrix defined as

$$Q_c = [\, B \;\; AB \;\ldots\; A^{n-1}B \,]$$

has $\rho(Q_c) = n$, with ρ the rank of Q_c matrix. If the system is not completely state controllable, there are eigenvalues of the matrix $(A - BK)$ that cannot be controlled by state feedback, otherwise all the eigenvalues of A can be arbitrarily placed.

However, this technique implies that all the states are available for feedback (all the states can be measured), but in this case, as described in chapter 1, only two states are available from outputs measurement, the position x and the angle α. This problem can be solved by using an asymptotic state observer. In real applications it is difficult to measure all the states, but an estimation of them can be achieved. The state observer, or estimator, is a dynamical system that estimates the state variables using the input signal and the output data of the plant as inputs to its system. A typical block diagram of the observer is shown in figure 3.2-20.

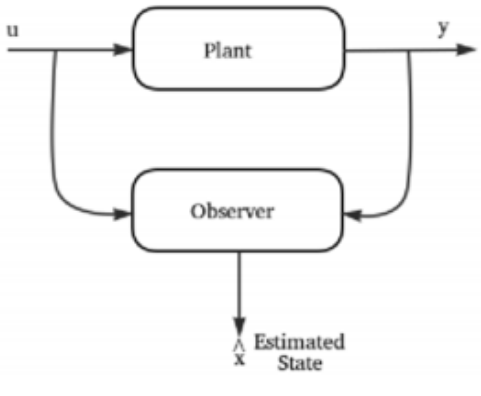


Fig. 3.2-20 Typical block's scheme of the asymptotic state observer

The method presented below is referred to the Luenberger observer design. Considering a plant described by a state space representation in minimal form (so the system is completely observable), the observer representation can be written in the following form

$$\dot{\tilde{x}} = [A]\tilde{x} + [B]u + [L](y - \tilde{y})$$

$$\tilde{y} = [C]\tilde{x}$$

with $\tilde{x}$ the estimated states vector, $\tilde{y}$ the output computed by $\tilde{x}$ and L a gain matrix. Now, let's perform a subtraction between the state space equations of the plant and observer

$$\left(\dot{x} - \dot{\tilde{x}}\right) = [A](x - \tilde{x}) - [L](y - \tilde{y}) \qquad (3.2.4\text{-}1)$$

$$(y - \tilde{y}) = [C](x - \tilde{x}) \qquad (3.2.4\text{-}2)$$

These equations describe the error between the real and estimated values of the states and the output. Substituting 3.2.4-2 in 3.2.4-1

$$\left(\dot{x} - \dot{\tilde{x}}\right) = ([A] - [L][C])(x - \tilde{x}) \rightarrow \dot{e_x} = ([A] - [L][C])e_x$$

$$(y - \tilde{y}) = [C](x - \tilde{x}) = [C]e_x$$

Therefore, the error dynamics equations are obtained and they must converge to zero as the time evolves. The zero convergence in the equations is achieved if the values of design variable matrix L is chosen so that the eigenvalues of the state matrix $([A] - [L][C])$ have negative real parts or, in other words, to make it asymptotically stable. Moreover, the observation error must converge to zero faster than the closed-loop response (the observer system must have poles faster than the closed loop poles). As the case of state feedback, is it always possible to estimate the matrix L so that $([A] - [L][C])$ has eigenvalues of arbitrarily choice? Like in the previous case, the answer is yes if and only if the system is completely observable. The observability condition for a system can be summarized as follows, a system is completely observable if and only if the observability matrix defined as

$$Q_o = [C^T \;\; AC^T \; \ldots \; A^{Tn-1}C^T \,]$$

has $\rho(Q_c) = n$, with ρ rank of matrix Q_o. In other words, a given state description can be transformed to an observer canonical form if (and only if) the observability matrix Q_o is non-singular (full rank).

Another powerful control technique, related to the static state feedback, is the Linear Quadratic Regulator LQR control that belong to the optimal control family. The idea is that the desired performances on the system output can be achieved thanks to two design parameters chosen to minimize the following cost function, along a time interval $\Delta t = t_f - 0$,

$$J = inf_{u(t)\in L(0,t_f)} \int_0^{t_f} x^t(\tau)Qx(\tau) + u(\tau)Ru(\tau)\, d\tau$$

The parameters $Q \geq 0$ and $R \geq 0$ are chosen to reflect the relative importance of the states x_i, in terms of "error loss", and the "energy loss" of the actuation command u (V_a in case of the electrical actuation model), respectively. Q and R are the weighting matrices that penalize certain states and the control inputs of the system. Q is a positive semi-definite matrix, while R is a positive definite matrix. If this system is disturbed and loses the zero state condition, the control input u should return the system to the zero state condition while, at the same time, J is minimal [7]. This condition is achieved if the following control law is applied to the system

$$u(t) = -R^{-1}B^T P x(t) = -Kx(t)$$

$$K = R^{-1}B^T P \quad \text{, with } P = P^T > 0 \text{ constant symmetric positive definite matrix}$$

and P satisfies the following algebraic Riccati equation

$$A^T P + PA - PBR^{-1}B^T + Q = 0$$

Once that the matrix P is obtained, the feedback gain K, that stabilizes the system, is finally achieved. The algebraic Riccati equation may be solved numerically for given values of A, B, R and Q.

In conclusion, the static state feedback control needs at least a simulator (in this case Simulink), where the dynamic system response can be evaluated in order to choose the best parameters K_i and guarantee the best output performances.

3.2.5 Static state feedback design: pole placement

The design of the controlled system, following the rules described in the previous section, is shown in figure 3.2-21.

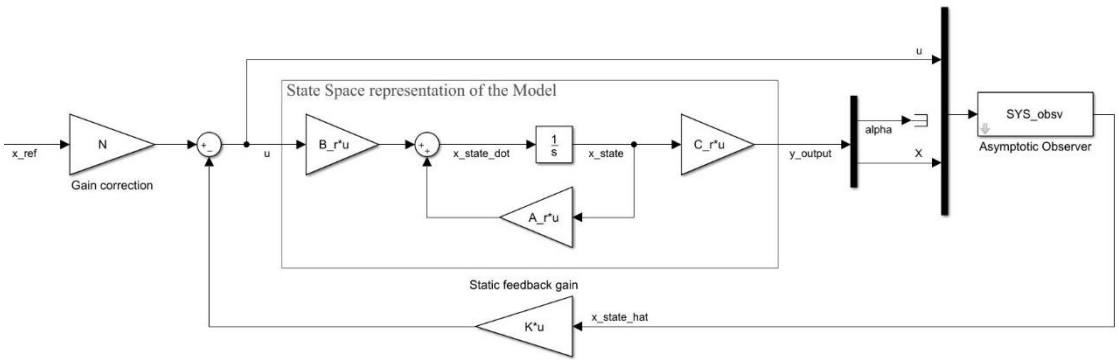


Fig. 3.2-21 Simulink's block scheme of the static state feedback control with asymptotic observer

The control law of the designed system is

$$u(t) = -Kx(t) + Nr(t)$$

where N is a filter coefficient that multiplies a reference signal $r(t)$, (x_{ref}), N is chosen to make unitary the dc-gain of the controlled system in order to obtain, besides asymptotic stability, good damping and rapidity properties of the transient. In this case $r(t)$ is a step reference signal. Using this architecture, it's possible to stabilize the response by placing the poles of the closed loop system in the left-half s-plane. The desired pole location are

$$p_{des} = [\,-10 - 6 - 10 - 0.0046\,]$$

that are used to calculate the static state feedback gain K, using the command 'acker' in Matlab. The resulting value of K is

$$K = \texttt{acker(A,B,}p_{des}\texttt{)} = [\,-35.4775 - 8.3793 - 0.0416 - 342.0771\,]$$

The N filter coefficient is computed as follows and its value is

$$N = [C(I - (A - BK))^{-1})B]^{-1} = -13.5288$$

The value of the parameter N depends on matrices A, B and C, but in presence of some uncertainty in these matrices the asymptotic tracking error couldn't converge to zero. The same results holds when a (step) disturbance is summed to either the input or the output. To solve this issue, it is possible to introduce, within the system equations, the information of a continuous time integral of the tracking error

$$\dot{q}(t) = q(t) + T\big(r(t) - y_i(t)\big) = q(t) + Te(t)$$

$$\lim_{t\to\infty} Te(t) = 0$$

In this way, a variable is controlled by an integrator filter in an outer loop. This variable constitutes the input for an inner closed loop controlled by static state feedback. The chosen variable, that is controlled by the integrator, is the position of the cart x because of limited path, otherwise the cart starts to move until it collides at end of the available path. The final Simulink scheme, including the integral action, is shown in figure 3.2-22.

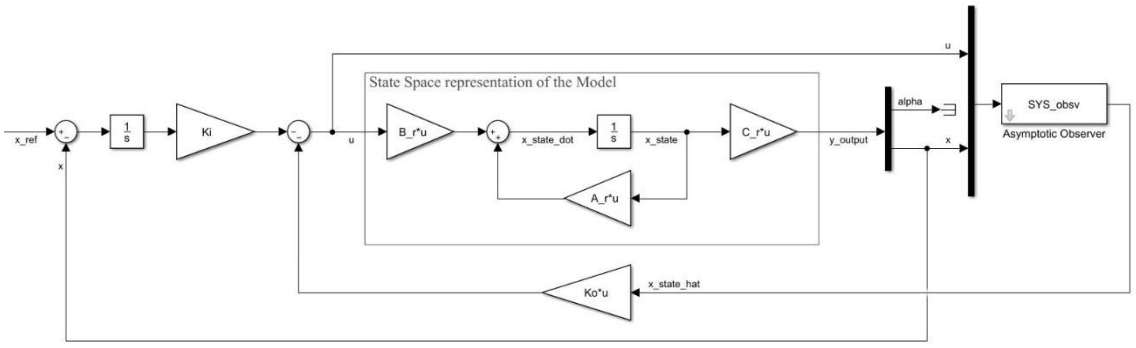


Fig. 3.2-22 Simulink's block scheme of the static state feedback control with integral action and asymptotic observer

In this way the system can be approximated to a SISO one. By adding to the system states the continuous time integrator of the tracking error, an augmented state is introduced and so the new state vector $\vec{x}_{tot}$ and the new state space matrices A_{tot}, B_{tot} and C_{tot} are defined as follows

$$\vec{x}_{tot}(t) = \begin{bmatrix} q(t) \\ \vec{x}(t) \end{bmatrix}$$

$$A_{tot} = \begin{bmatrix} 1_{1\times1} & -TC_x \\ 0_{n\times1} & A \end{bmatrix}, \quad \text{with } C_x = [\,0\ 0\ 1\ 0] \text{ second row of } C \text{ matrix}$$

$$B_{tot} = \begin{bmatrix} 0 \\ B \end{bmatrix}$$

$$B_r = \begin{bmatrix} T \\ 0 \end{bmatrix}$$

$$C_{tot} = [0_{1\times1} \quad C]$$

and the system equations becomes

$$\dot{\vec{x}}_{tot}(t) = A_{tot}\vec{x}_{tot}(t) + B_{tot}u(t) + B_r r(t)$$
$$y(t) = C_{tot}\vec{x}_{tot}(t)$$

The selected control law is modified as

$$u(t) = -K\vec{x}_{tot}(t) = -[K_i \ K_o]\vec{x}_{tot}(t) = -[K_i \ K_o]\begin{bmatrix} q(t) \\ \vec{x}(t) \end{bmatrix}$$

with K_i and K_o the gain coefficients for the integral action and the static state feedback. The resulting parameters K_i and K_o to obtain the p_{des} placements of poles are computed as follows

$$p_{des} = [\,-10 - 8 - 6 - 4 - 0.0046\,]$$

$$K = \texttt{place}(\texttt{A}_{\texttt{tot}},\texttt{B}_{\texttt{tot}},\texttt{p}_{\texttt{des}}) = [55.3452 \ -76.8162 \ -17.3672 \ -55.2119 \ -359.1657]$$

$$K_i = [55.3452]$$

$$K_o = [-76.8162 \ -17.3672 \ -55.2119 \ -359.1657]$$

The Matlab scripts used to support the Simulink environment are shown in Appendix B1. The resulting system is stable and provides good performances. Figures 3.2-23 and 24 show the Nyquist and Bode diagrams of the obtained open loop system.

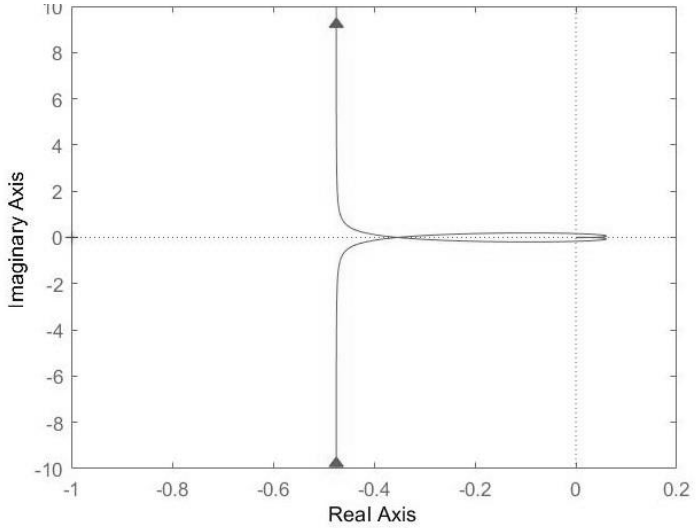


Fig. 3.2-23 Nyquist diagram of the open loop system obtained by static state feedback control and integral action

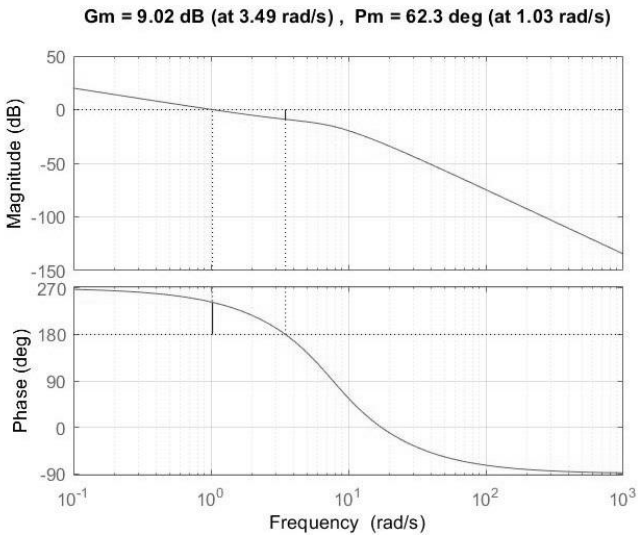


Fig. 3.2-24 Bode diagrams of the open loop system obtained by static state feedback control and integral action

3.2.6 Static state feedback design: LQR

This control technique is similar to the previous case. The LQR control technique is an algorithm that is useful to force the closed loop poles to lie in the stable region of the s-plane (negative real side). Moreover, it provides an important improvement on the stability achieving. This technique allows to stabilize the system accordingly to the specified parameters Q and R, that are useful to understand which states or command need to be penalized, by weights, in order to achieve best output performances. The Simulink block's scheme used by this control architecture is shown in figure 3.2-25.

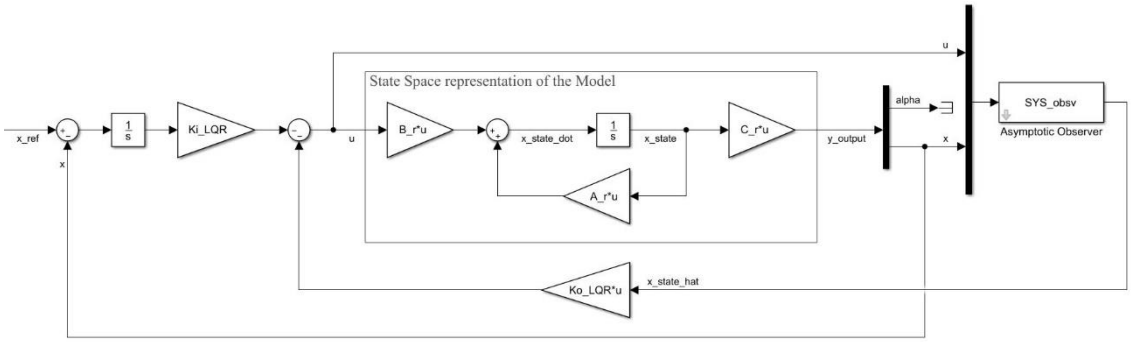


Fig. 3.2-25 Simulink block's scheme of the static state feedback control with integral action and asymptotic observer, using the LQR optimal control algorithm

Also in this case, an integral action on the outer loop is needed. The controlled system is not able to guarantee an asymptotic zero tracking error in presence of disturbances or step input signals.

Once that the weighting matrices are assigned, the closed loop poles are computed using the command `'lqr'`. Then, the gain K can be computed thanks to the Matlab command `'place'`, as shown in Appendix B1.

The values of Q and R that guarantee best output performances are

$$Q = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \in \mathbb{R}^{n \times n}$$

$$R = [\,1\,], \in \mathbb{R}^{1 \times 1}$$

The resulting gain matrix K, obtained with these assigned parameters, is

$$K = [K_i \;\; K_o] = [\,531.83 \;\; -1915.3 \;\; -432.61 \;\; -531.82 \;\; -862.30\,]$$

The obtained control systems achieves the stability with very good performances in terms of time and frequency behavior. The Nyquist and Bode diagrams of the open loop system are shown in figures 3.2-26 and 27, while the time responses are presented in the chapter 4.

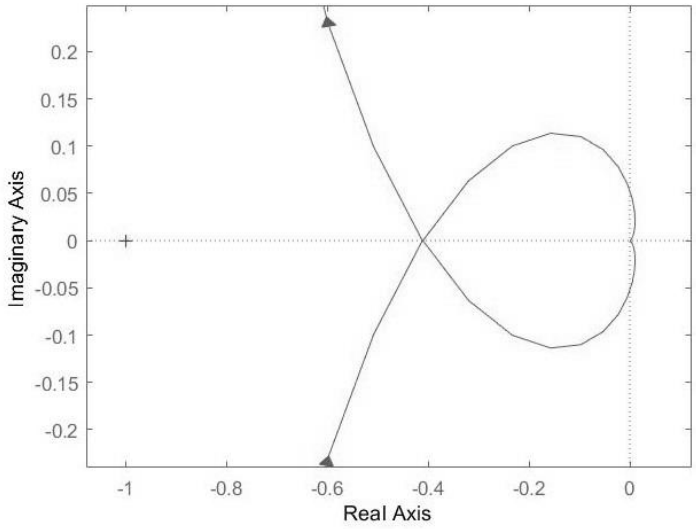


Fig. 3.2-26 Nyquist diagram of the open loop system obtained by LQR control and integral action

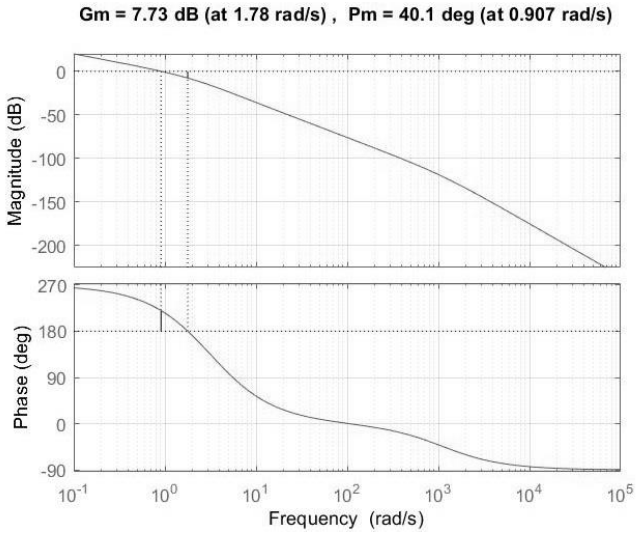


Fig. 3.2-27 Bode diagrams of the open loop system obtained by LQR control and integral action

3.3 Controllers of pneumatic actuation model

The mathematical model obtained with the pneumatic actuation is studied in [4], where a cascade PIDs' cascade architecture is designed to control the system. The mathematical model is reclaimed in order to study the same techniques realized for the model with electrical actuation.

Initially, the transfer functions $G_{u\alpha}$ and G_{ux} are translated into the Simulink's block scheme so that the simulation model is obtained like in figure 3.3-1.

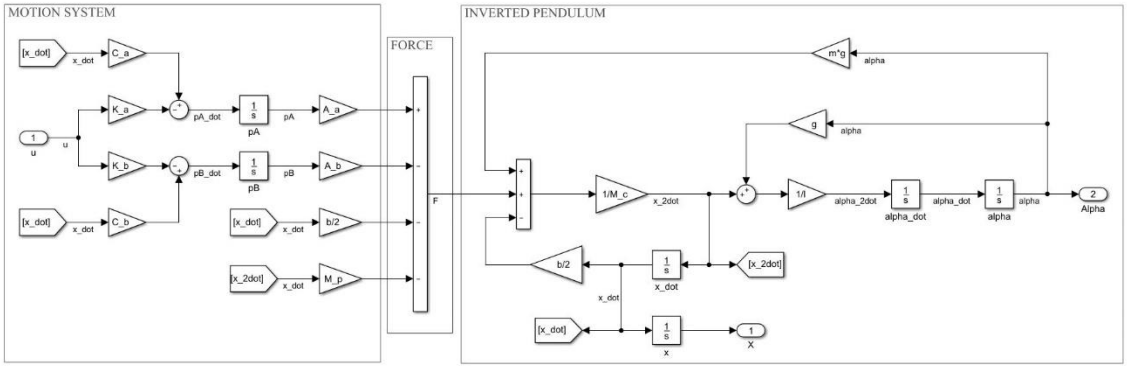


Fig. 3.3-1 Simulink model of the inverted pendulum system with pneumatic actuation

The transfer functions of the system are

$$G_{u\alpha} = \frac{(K_A A_A + K_B A_B)s}{(M_c + M_P)ls^4 + (b_c + b_P)ls^3 - (m + M_c + M_p)gs^2 + (A_A C_A + A_B C_B)ls^2 - (b_c + b_P)gs - (A_A C_A + A_B C_B)g} =$$

$$= \frac{796.33s}{(s + 4.449)(s - 4.442)(s^2 + 9.902s + 202.3)}$$

$$G_{ux} = \frac{(K_A A_A + K_B A_B)s}{(M_c + M_P)ls^4 + (b_c + b_P)ls^3 - (m + M_c + M_p)gs^2 + (A_A C_A + A_B C_B)ls^2 - (b_c + b_P)gs - (A_A C_A + A_B C_B)g} \frac{ls^2 - g}{s^2} =$$

$$= \frac{398.16(s + 4.427)(s - 4.427)}{s(s + 4.449)(s - 4.442)(s^2 + 9.902s + 202.3)}$$

Moreover, another useful transfer function is the one that connects the state x with the state α

$$G_{\alpha x} = \frac{G_{ux}}{G_{u\alpha}} = \frac{ls^2 - g}{s^2} = \frac{0.5(s - 4.427)(s + 4.427)}{s^2}$$

In the table 3.1-1, the experimental values of the model variables are shown.

l	rod length	$0.5\ m$
m	pendulum mass	$0.2\ kg$
M_c	cart mass	$1.8\ Kg$
M_p	piston mass	$0.4\ Kg$
b	viscous damping coefficient	$20.83\ N\frac{s}{m}$
A_A	area of the rear piston chamber	$2.01 \cdot 10^{-4}\ m^2$
A_B	area of the front piston chamber	$1.73 \cdot 10^{-4}\ m^2$
C_A	$\frac{A_A p_{supply}}{2V_{A0}}$	$1200000\frac{Pa}{m}$
C_B	$\frac{A_B p_{supply}}{2V_{B0}}$	$1200000\frac{Pa}{m}$
K_A	$\frac{p_0}{2\rho_0}\frac{Kp_{supply}}{2V_{A0}}$	$2177000\frac{Pa}{sV}$
K_B	$\frac{p_0}{2\rho_0}\frac{Kp_{supply}}{2V_{B0}}$	$2534000\frac{Pa}{sV}$

Table 3.3-1 Experimental values of the model variables

Once that the model is obtained in the Simulink environment, let's start with the design phase of the controllers but, for clarity of the contents, also the PIDs' cascade architecture obtained in [4] is presented in the next section.

3.3.1 PID design: cascade architecture

Knowing that this architecture is equal to the one shown in section 3.2.2, this section shows the results obtained in [4]. The final controllers C_α (inner loop) and C_x (outer loop), that are able to stabilize the plant with pneumatic actuation, are

$$C_\alpha = 2.09 + \frac{6.31}{s} + \frac{0.103s}{0.000875s + 1}$$

$$C_x = -0.000802 + \frac{-4.68 * 10^{-6}}{s} - \frac{0.0306s}{0.00875s + 1}$$

The resulting system provides good performances in terms of frequency response, as shown in figures 3.3-2 and 3.3-3.

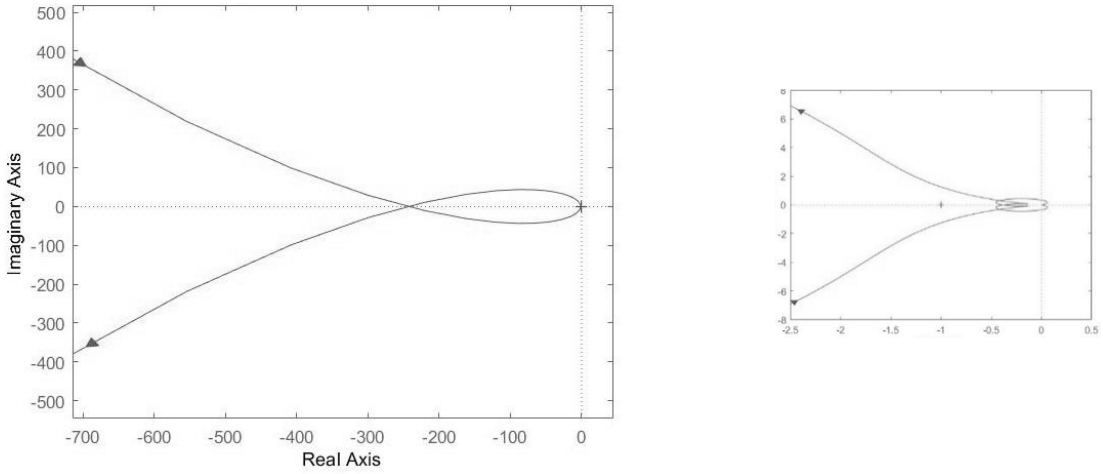


Fig. 3.3-2 Nyquist diagram of the pneumatic model controlled by the PIDs' cascade architecture, on the left. Zoom around the critical point $-1 + i0$, on the right

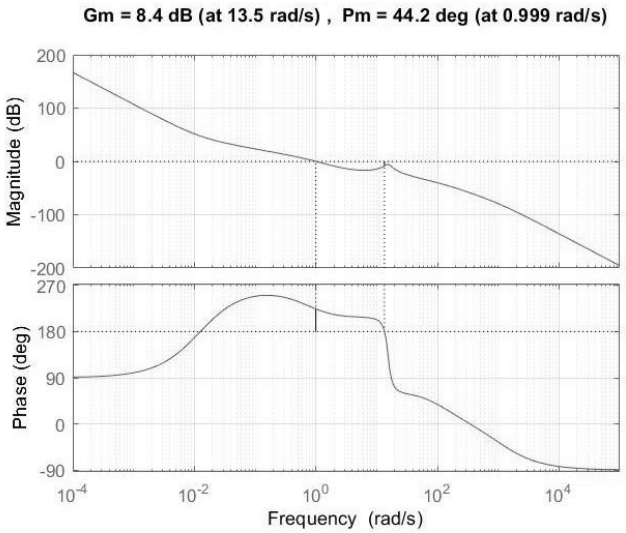


Fig. 3.3-3 Bode diagrams of the pneumatic model controlled by the PIDs' cascade architecture

3.3.2 PID design: parallel architecture

Knowing that this architecture is equal to the one shown in section 3.2.3 the same considerations are still valid. Therefore, also the hypothesis of SIMO system are considered in this case that leads to the computation of the open loop and closed loop transfer functions, of the overall model, as follows

$$G_{x_{ol}} = \frac{N_{ol}}{D_{ol}} = \frac{C_x G_{ux}}{1 + C_\alpha G_{u\alpha}}$$

$$G_{x_{cl}} = \frac{N_{cl}}{D_{cl}} = \frac{C_x G_{ux}}{1 + C_\alpha G_{u\alpha} + C_x G_{ux}}$$

Like in section 3.2.3, the controller C_α is equal to the one that is able to stabilize the inner loop of the PIDs' cascade architecture (section 3.3.1), therefore let's start to analyze the open loop transfer function of the uncontrolled system (without the controller C_x)

$$G_{x_{ol,un}} = \frac{N_{ol,un}}{D_{ol,un}} = \frac{G_{ux}}{1 + C_\alpha G_{u\alpha}} = \frac{398.16(s + 4.427)(s - 4.427)(s + 1143)}{s(s + 1143)(s + 5.306)(s + 0.8142)(s^2 + 3.716s + 237.6)}$$

In this way, the stability problem is related to the roots of $D_{ol,un}$, that represent the poles of the uncontrolled system

$$p_1 = 0$$

$$p_2 = -1143$$

$$p_3 = -1.8581 - i15.3$$

$$p_4 = -1.8581 + i15.3$$

$$p_5 = -0.8142$$

$$p_6 = -5.306$$

Looking at the poles, the uncontrolled system has a pole in zero, that could compromise the stability. A proof is the Nyquist path of $G_{x_{ol,un}}$ in figure 3.3-4.

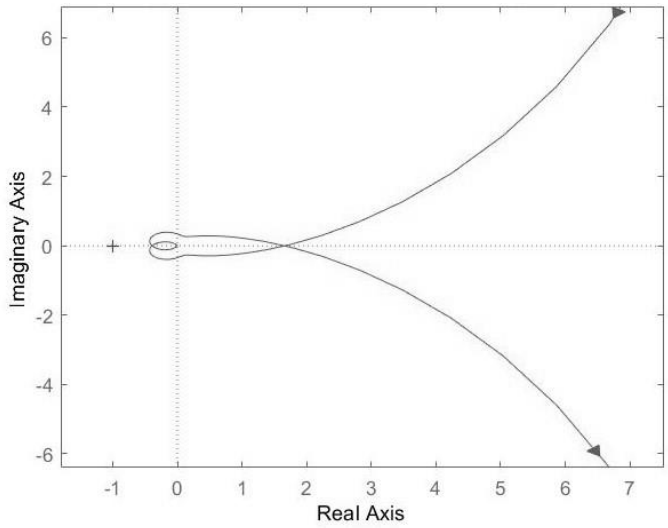


Fig. 3.3-4 Nyquist diagram of $G_{x_{ol,un}}$

Looking at the Nyquist path in figure above, $G_{x_{ol,un}}$ results unstable because the presence of p_1 creates a clockwise encirclement of the critical point $(-1; 0)$, that makes the system unstable violating the Nyquist criterion.

Luckily, also this transfer function ($G_{x_{ol,un}}$) can be stabilized using only a proportional coefficient $K_p < 0$, so the Ziegler-Nichols empiric method can be used to improve the stability performances. Initially, the critical gain K_c must be determined and in this case it is reached at -0.6. The resulting critical oscillation period is $T_c = 4.15s$ (but evaluating the output performances, of the overall controlled system, a value of 5 is successively chosen).

So, assuming $K_c = -0.6$ and $T_c = 5s$, the resulting coefficients for the controller C_x are

$$K_p = -0.36 \qquad K_d = -0.225$$
$$K_i = -0.144 \qquad N = 0.0938$$

The resulting controller is able to stabilize the closed loop transfer function $G_{x_{cl}}$

$$G_{x_{cl}} = -\frac{1098.9(s + 4.427)(s - 4.427)(s + 1143)(s^2 + 1.433s + 0.5565)}{(s + 1143)(s + 1.335)(s^2 + 2.019s + 1.312)(s^2 + 9.041s + 34.13)(s^2 + 8.108s + 200.6)}$$

A graphical proof is given by the Nyquist path in figure 3.3-5, of the open loop transfer function of the controlled system

$$G_{x_{ol}} = -\frac{1098.9(s+4.427)(s-4.427)(s+1143)(s^2+1.433s+0.5565)}{s^2(s+10.67)(s+5.306)(s+0.8142)(s+1143)(s^2+3.716s+237.6)}$$

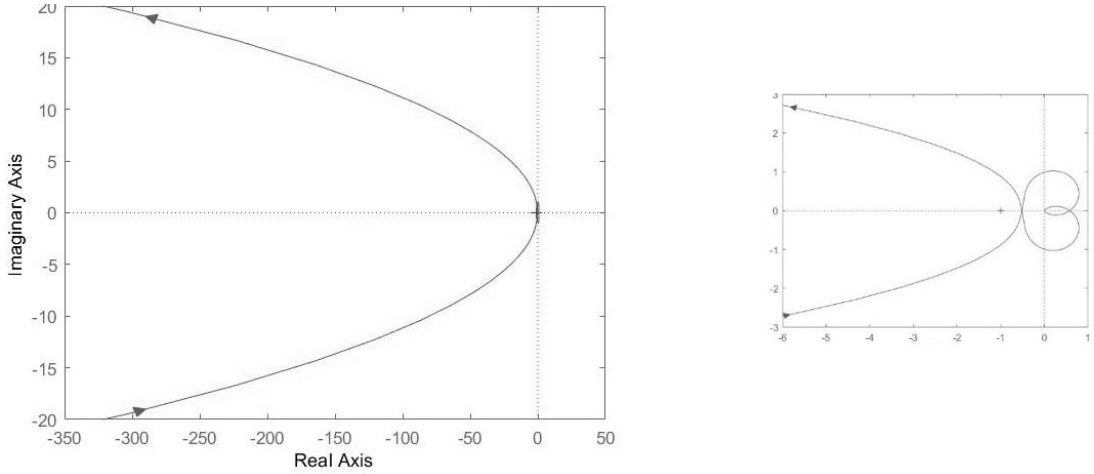


Fig. 3.3-5 Nyquist diagram of $G_{x_{ol}}$, on the left. Zoom around the critical point $-1+i0$, on the right

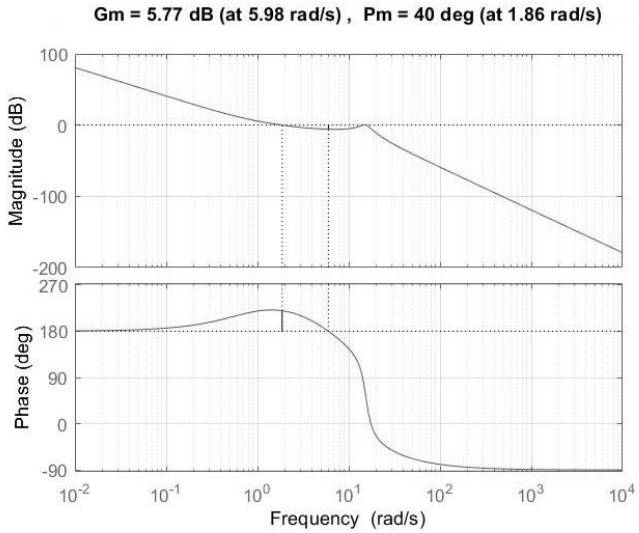


Fig. 3.3-6 Bode diagrams of $G_{x_{ol}}$

As can be seen, there are no encirclements of the critical point $(-1;0)$ and so the system results stable for the Nyquist criterion.

Unlike the electric case, the using of the PID tuning tool is not necessary because the system achieves good performances with the PID computed by Ziegler-Nichols approach. As can be seen in figure 3.3-6, the Bode diagrams show a gain margin of $5.77dB$ at $5.98rad/s$ and a phase margin of $40°$, so the final PID coefficients for the simulations has been assigned accordingly the Ziegler-Nichols method.

3.3.3 Static state feedback design: pole placement

The same consideration and approximation considered in sections 3.2.5 holds also for the pneumatic model. The reader is invited to consult the Appendix B2 for an exhaustive and complete understanding of the procedures and calculations done in the Simulink environment.

The final architecture, that uses the pole placement technique, is equal to the one shown in figure 3.2-22, which is the static state feedback architecture (inner loop) with integral action (outer loop) and asymptotic observer.

Knowing that the control law used in this type of architecture is equal to

$$u(t) = -K\vec{x}_{tot}(t) = -[K_i \;\; K_o]\vec{x}_{tot}(t) = -[K_i \;\; K_o]\begin{bmatrix} q(t) \\ \vec{x}(t) \end{bmatrix}$$

the resulting parameters K_i and K_o are computed starting with the choice of p_{des}

$$p_{des} = [\,-10 - 10 - 10 - 10 - 10 - 4.6\,]$$

$$K = \texttt{acker}(\texttt{A}_{\texttt{tot}}, \texttt{B}_{\texttt{tot}}, \texttt{p}_{\texttt{des}}) = [\,121.7804 \;\; 52.8375 \;\; 11.9164 \;\; 2.0988 \cdot 10^{-5} \;\; -61.0605 \;\; -22.1997\,]$$

$$K_i = [121.7804]$$

$$K_o = [\,52.8375 \;\; 11.9164 \;\; 2.0988 \cdot 10^{-5} \;\; -61.0605 \;\; -22.1997\,]$$

The command `'acker'` is used instead of `'place'` because it supports a repeating value for the pole locations.

Therefore, adopting the previous results of K_i and K_o, the system results stable, providing good performances. Figures 3.3-7 and 8 show the Nyquist and Bode diagrams of the obtained open loop system.

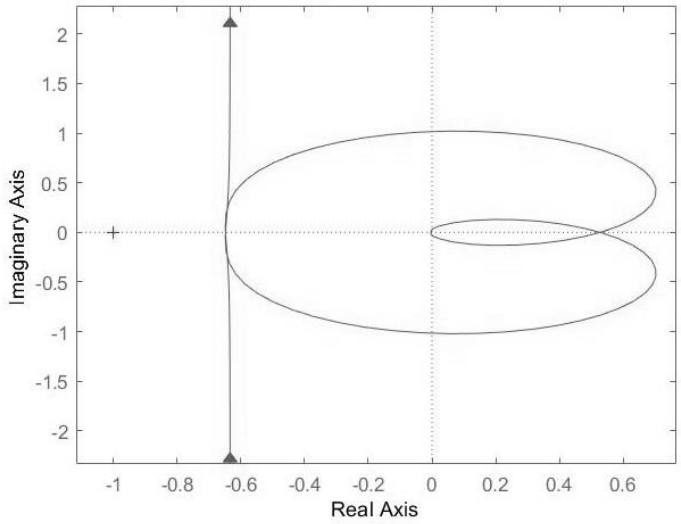


Fig. 3.3-7 Nyquist diagram of the open loop system obtained by static state feedback control and integral action

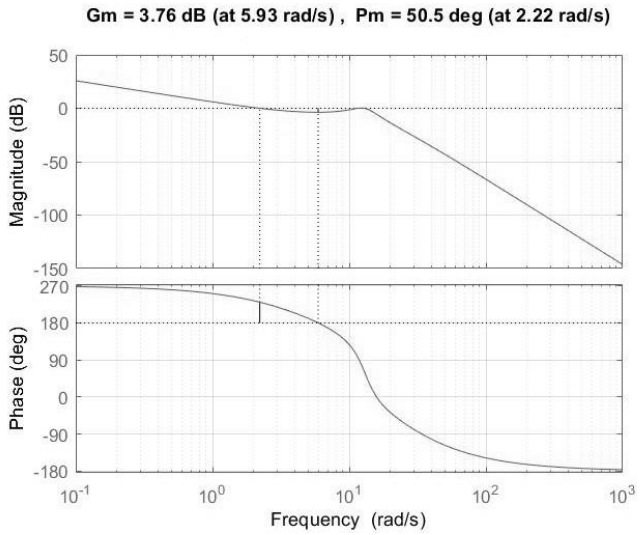


Fig. 3.3-8 Bode diagrams of the open loop system obtained by static state feedback control and integral action

3.3.4 Static state feedback design: LQR

The control architecture used in this section is equal to the one shown in figure 3.2-25, which is the static state feedback architecture (inner loop) with integral action (outer loop) and asymptotic observer, using the LQR optimal control algorithm.

In this case the values of Q and R that guarantee best output performances are

$$Q = \begin{bmatrix} 10 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 100 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}, \in \mathbb{R}^{n \times n}$$

$$R = [\, 1 \cdot 10^{9} \,], \in \mathbb{R}^{1 \times 1}$$

Unlike the electrical actuation model, different weights are assigned to the states. A high importance is given to the α state (multiplicative factor of 10) and to the x one (multiplicative factor of 100). Moreover, a very huge weight for the input signal is assigned and it is equal to $1 \cdot 10^{9}$.

The resulting gain matrix K, obtained with these assigned parameters, is

$$K = [K_i \;\; K_o] = [\, 385.07 \;\; 249.85 \;\; 56.3763 \;\; 5.74 \cdot 10^{-5} \;\; -257.13 \;\; -101.71 \,]$$

The obtained control systems achieves the stability with very good performances in terms of time and frequency behavior. The Nyquist and Bode diagrams of the open loop system are shown in figures 3.3-9 and 10, while the time responses are presented in the chapter 4.

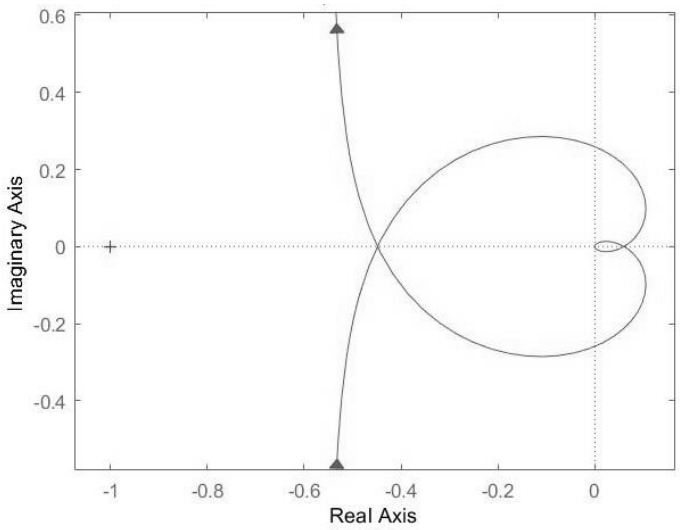


Fig. 3.3-9 Nyquist diagram of the open loop system obtained by LQR control and integral action

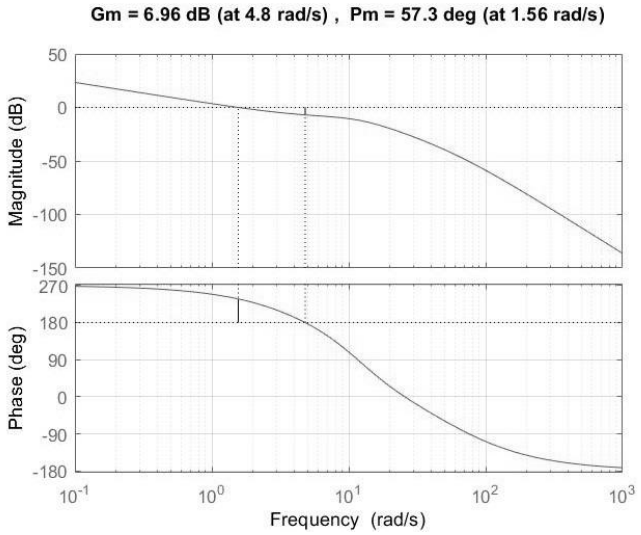


Fig. 3.3-10 Bode diagrams of the open loop system obtained by LQR control and integral action

4 Numerical simulations

This chapter presents the numerical simulations performed on the controlled system, taking into account the control techniques described in chapter 3 (cascade, parallel and LQR control architectures). The simulations were performed in order to observe the time response of the system, providing as inputs different signals (step, square and sine waves).

Moreover, the system was modified, in terms of rod length l_p and pendulum mass m_p, in order to highlight differences in the behaviour of the step responses. These parameters modifications, m_p and l_p, do not involve any modification in the PID controllers for the cascade and parallel architecture, but, in the LQR control technique, a change in the state space representation implies a change on the desired eigenvalues needed for the computation of K_{LQR}. Therefore, the parameter changes involve different computations for the gain K_{LQR}, in order to achieve the stability of the system.

The performances of the step responses can be classified in terms of settling time and over/under shoot. The settling time represents the amount of time required by the step response to reach and stay within the $\pm\,\beta$ % ($\beta = 5\%$ in this case) of the steady state value y_∞ (output value at infinite time). While, the over/under shoot represent the maximum, positive or negative, deviation from y_{ref} (reference signal for the output one).

4.1 Step response

Firstly, a step input signals, of 200mm amplitude, was imposed to the controlled system. The step responses were performed on three different configurations of the system, taking into account three couples of different values of m_p and l_p (180gr-550mm, 180gr-400mm and 550mm-1Kg, respectively).

4.1.1 Step response: electrical actuation model

In this section the step responses of the electrical model are analyzed. Let's have a look at the step responses shown in figure 4.1-1, of the system controlled by the PIDs' cascade architecture.

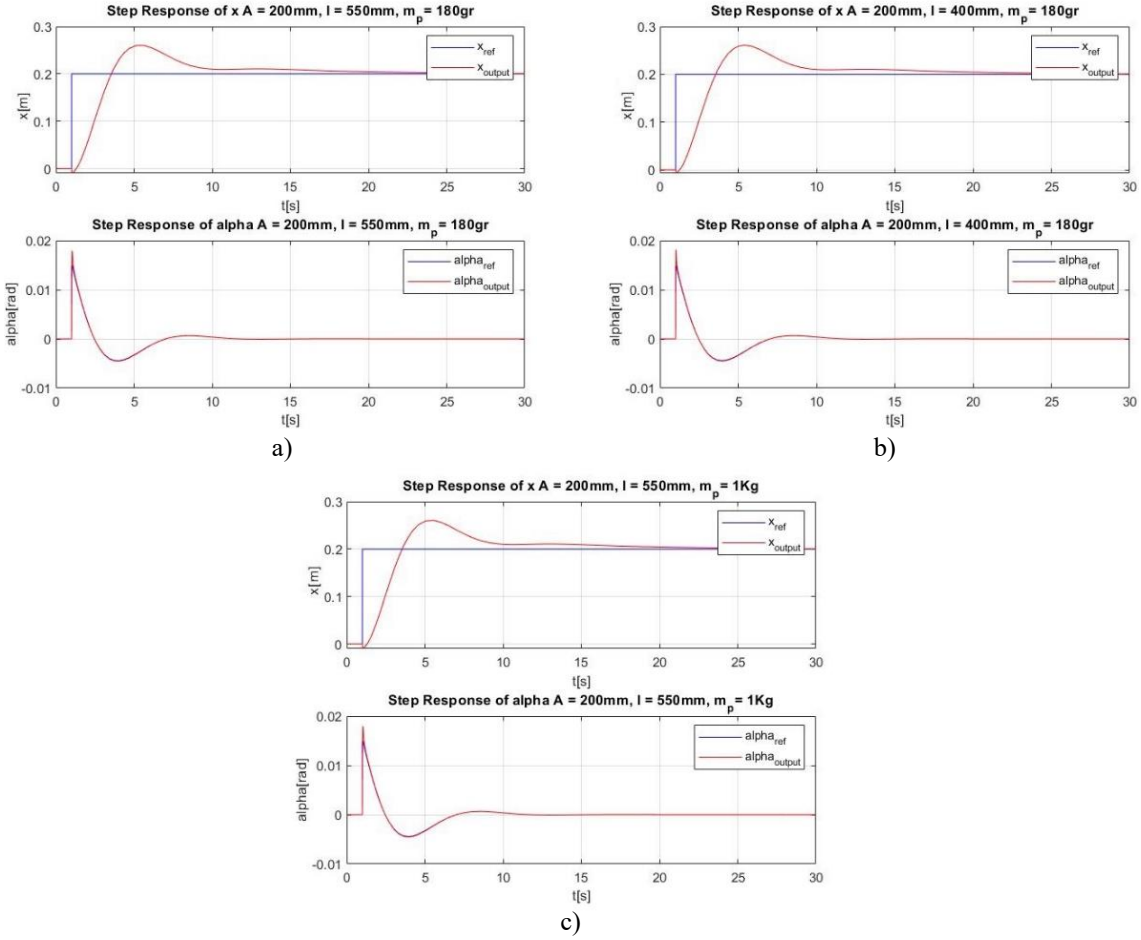


Fig. 4.1-1 Step responses of the electrical model controlled by the PIDs' cascade architecture. In each figure, from top to bottom, the time responses of x and α. Fig. a m_p=180gr and $l_p = 550mm$. Fig. b m_p=180gr and $l_p = 440mm$. Fig. c m_p=1Kg and $l_p = 550mm$.

The system reacts to the changes, on m_p and l_p, in the same way for the three configurations. Although there is a difference on m_p around 800 gr and on l_p around 100 mm, the PID cascade control is able to generates the same time response. This means that the cascade architecture results very robust about changes in the rod length and pendulum mass.

The values related to settling time and over/undershoot of the responses in figure 4.1-1 are resumed in table 4.1-1.

		Settling time 5% $[s]$	Overshoot $x[m], \alpha[rad]$	Undershoot $x[m], \alpha[rad]$
$l_p = 550mm$ $m_p = 180gr$	x_{output}	10.20	0.260	-0.009
	α_{output}	–	0.017	-0.004
$l_p = 400mm$ $m_p = 180gr$	x_{output}	10.18	0.261	-0.007
	α_{output}	–	0.018	-0.004
$l_p = 550mm$ $m_p = 1Kg$	x_{output}	10.10	0.260	-0.009
	α_{output}	–	0.017	-0.004

Table 4.1-1 Values of settling time and over/undershoot of step responses in figure 4.1-1

Figure 4.1-2 shows the step responses related to the system controlled by the PIDs' parallel architecture.

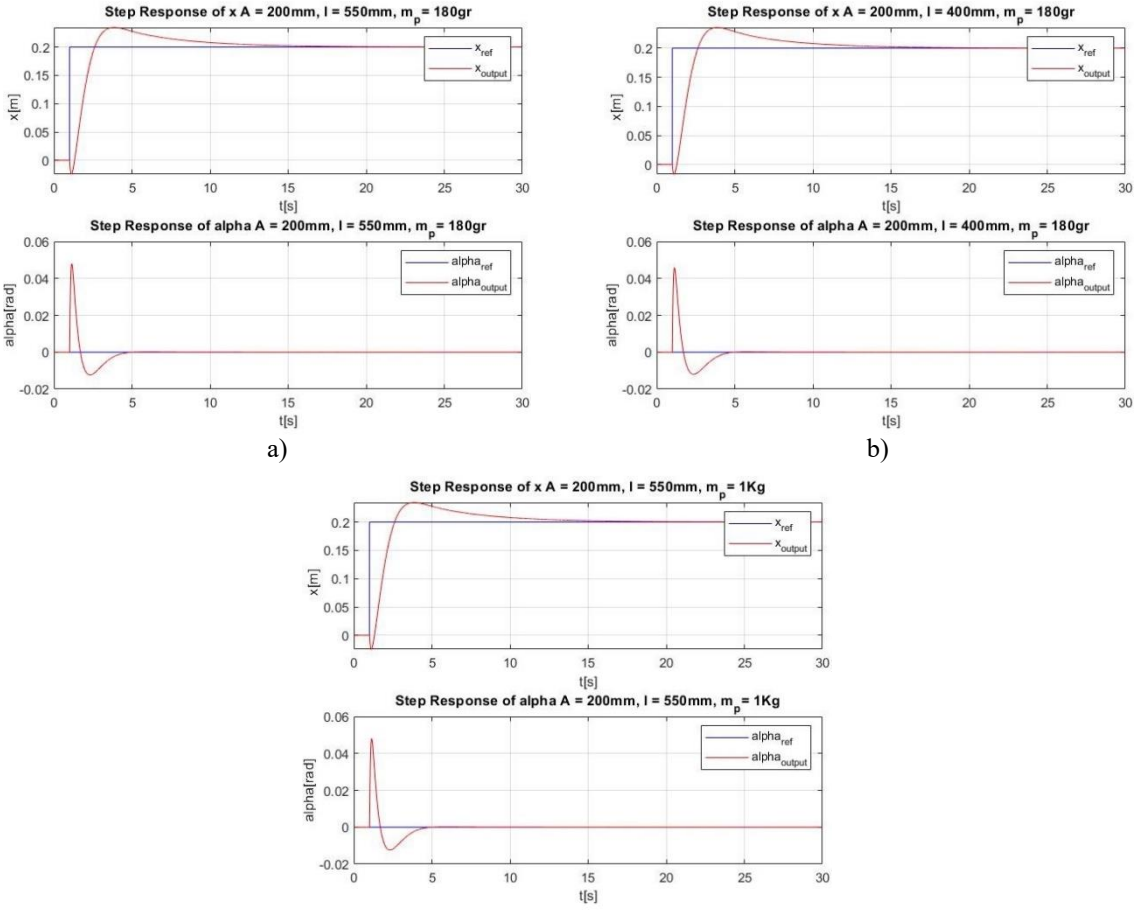


Fig. 4.1-2 Step responses of the electrical model controlled by the PIDs' parallel architecture.
In each figure, from top to bottom, the time responses of x and α. Fig. a m_p=180gr and $l_p = 550mm$. Fig. b m_p=180gr and $l_p = 440mm$. Fig. c m_p=1Kg and $l_p = 550mm$.

As can be seen in figure 4.1-2, also in the parallel architecture as in the cascade one, for all the configurations the system reacts in the same way to the step input signal. The same overshoot, undershoot and settling time are performed by the system response in each graph.

The values related to settling time and over/undershoot of the responses in figure 4.1-2 are resumed in table 4.1-2.

		Settling time 5% $[s]$	Overshoot $x[m], \alpha[rad]$	Undershoot $x[m], \alpha[rad]$
$l_p = 550mm$ $m_p = 180gr$	x_{output}	9.07	0.234	-0.024
	α_{output}	–	0.047	-0.012
$l_p = 400mm$ $m_p = 180gr$	x_{output}	9.06	0.235	-0.016
	α_{output}	–	0.045	-0.011
$l_p = 550mm$ $m_p = 1Kg$	x_{output}	9.04	0.234	-0.024
	α_{output}	–	0.047	-0.012

Table 4.1-2 Values of settling time and over/undershoot of step responses in figure 4.1-2

Moreover, the step responses performed on the system controlled by the LQR architecture are shown in figure 4.1-3.

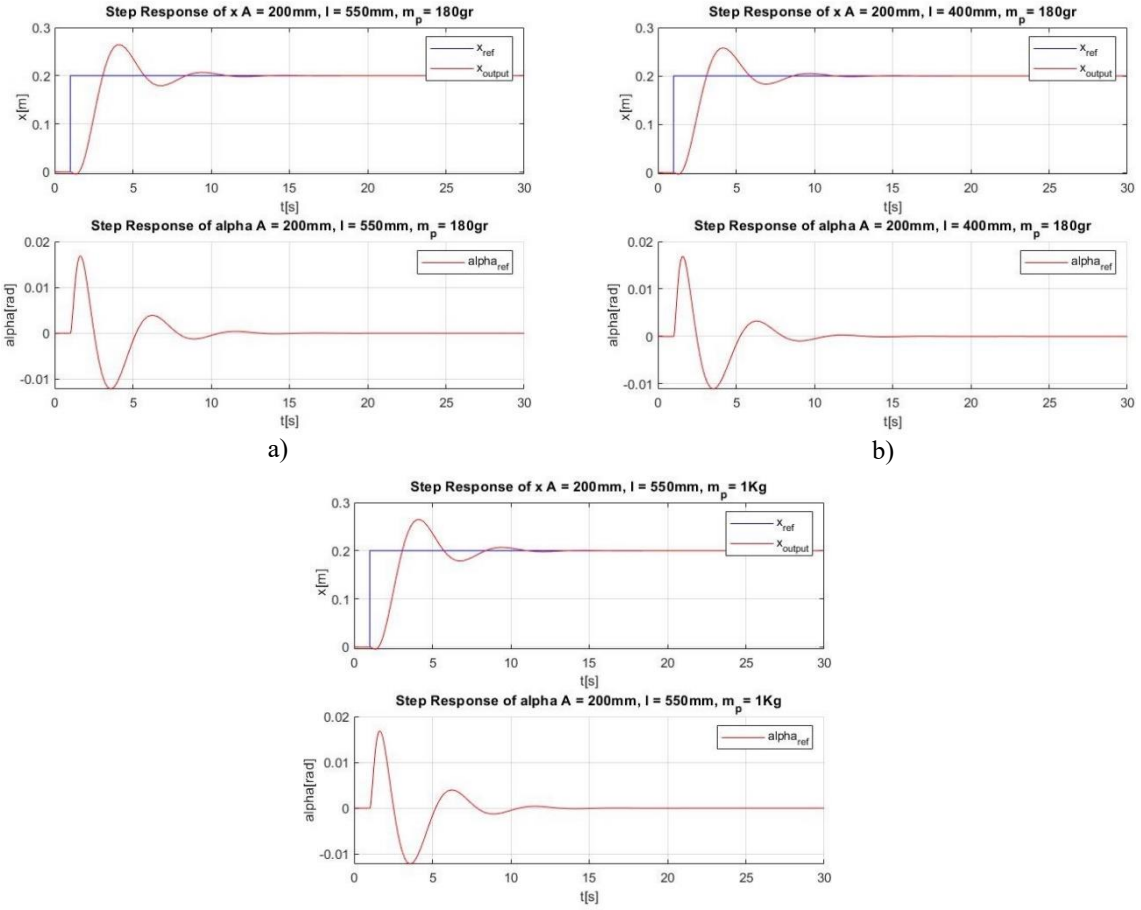


Fig. 4.1-3 Step responses of the electrical model controlled by the LQR architecture. In each figure, from top to bottom, the time responses of x and α. Fig. a m_p=180gr and $l_p = 550mm$. Fig. b m_p=180gr and $l_p = 440mm$. Fig. c m_p=1Kg and $l_p = 550mm$.

Although some low differences in the overshoots of the x time response, the graphs a), b), and c) in figure 4.1-3 are very similar.

The results obtained prove that the electrical actuation system, regardless of the control used, is able to react in the same way against changes in the rod length and pendulum mass.

The values related to settling time and over/undershoot of the responses in figure 4.1-3 are resumed in table 4.1-3.

		Settling time 5% $[s]$	Overshoot $x[m], \alpha[rad]$	Undershoot $x[m], \alpha[rad]$
$l_p = 550mm$ $m_p = 180gr$	x_{output}	5.38	0.264	-0.004
	α_{output}	–	0.016	-0.012
$l_p = 400mm$ $m_p = 180gr$	x_{output}	5.27	0.253	-0.002
	α_{output}	–	0.019	-0.009
$l_p = 550mm$ $m_p = 1Kg$	x_{output}	5.39	0.264	-0.004
	α_{output}	–	0.015	-0.013

Table 4.1-3 Values of settling time and over/undershoot of step responses in figure 4.1-3

4.1.2 Step response: pneumatic actuation model

The step responses shown in section 4.1.1 were also performed in the case of the pneumatic model.

In figure 4.1-4, the step responses of the system controlled by the PIDs' cascade architecture are shown.

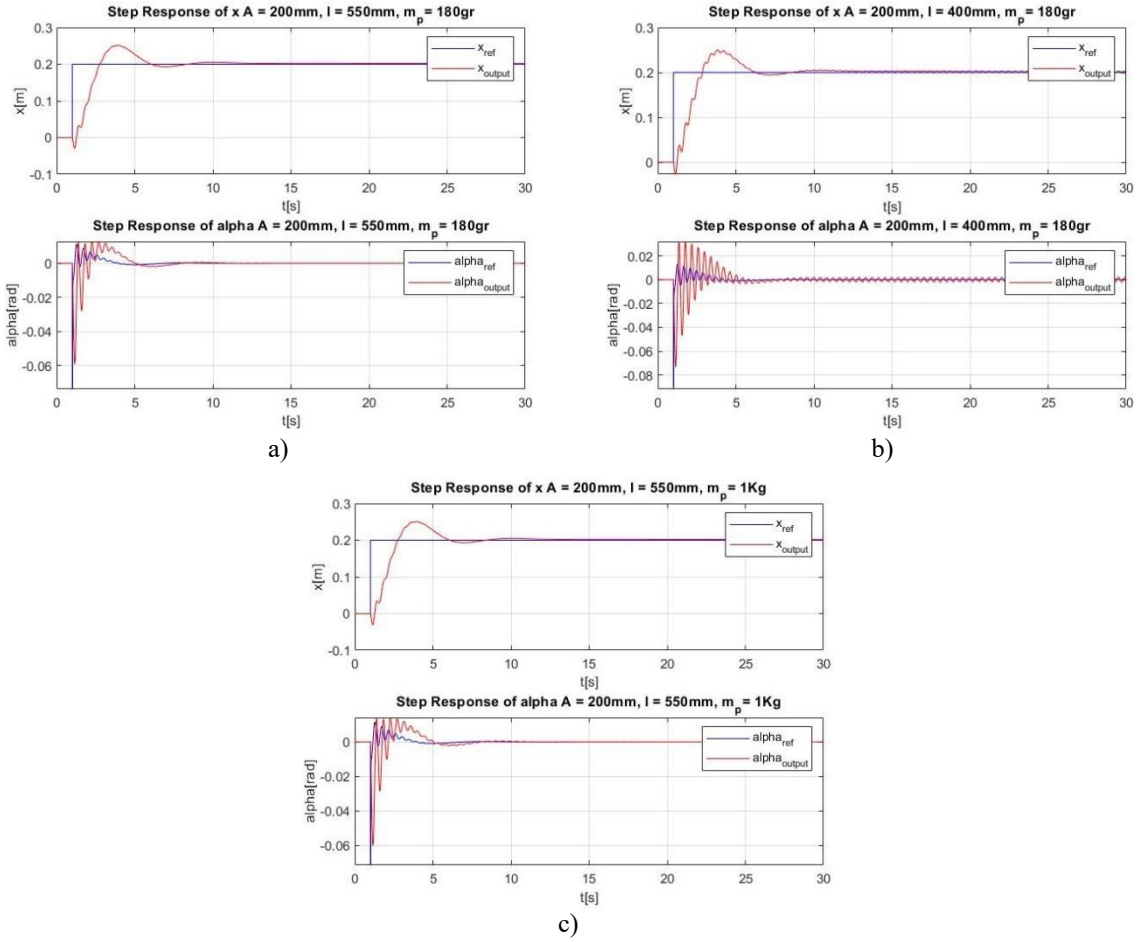


Fig. 4.1-4 Step responses of the pneumatic model controlled by the PIDs' cascade architecture. In each figure, from top to bottom, the time responses of x and α. Fig. a m_p=180gr and $l_p = 550mm$. Fig. b m_p=180gr and $l_p = 440mm$. Fig. c m_p=1Kg and $l_p = 550mm$.

Unlike the electrical model, in this case some response differences are present in the three configurations. As can be seen in figures 4.1-4 a) and c), the time responses of x and α are very similar, so the difference of 820 gr in the pendulum mass doesn't involve a change in the reaction

of the system to the step signal. However, looking at figure 4.1-4 b), the time responses of x and α exhibits more oscillations than in figures 4.1-4 a) and c). This means that a shortest rod length causes an hardest scenario for the control problem.

The values related to settling time and over/undershoot of the responses in figure 4.1-4 are resumed in table 4.1-4.

		Settling time 5% $[s]$	Overshoot $x[m], \alpha[rad]$	Undershoot $x[m], \alpha[rad]$
$l_p = 550mm$ $m_p = 180gr$	x_{output}	5.56	0.251	-0.030
	α_{output}	–	0.012	-0.059
$l_p = 400mm$ $m_p = 180gr$	x_{output}	5.65	0.250	-0.026
	α_{output}	–	0.031	-0.073
$l_p = 550mm$ $m_p = 1Kg$	x_{output}	5.60	0.250	-0.030
	α_{output}	–	0.014	-0.060

Table 4.1-4 Values of settling time and over/undershoot of step responses in figure 4.1-4

Let's have a look at the step responses shown in figure 4.1-5, of the system controlled by the PIDs' parallel architecture.

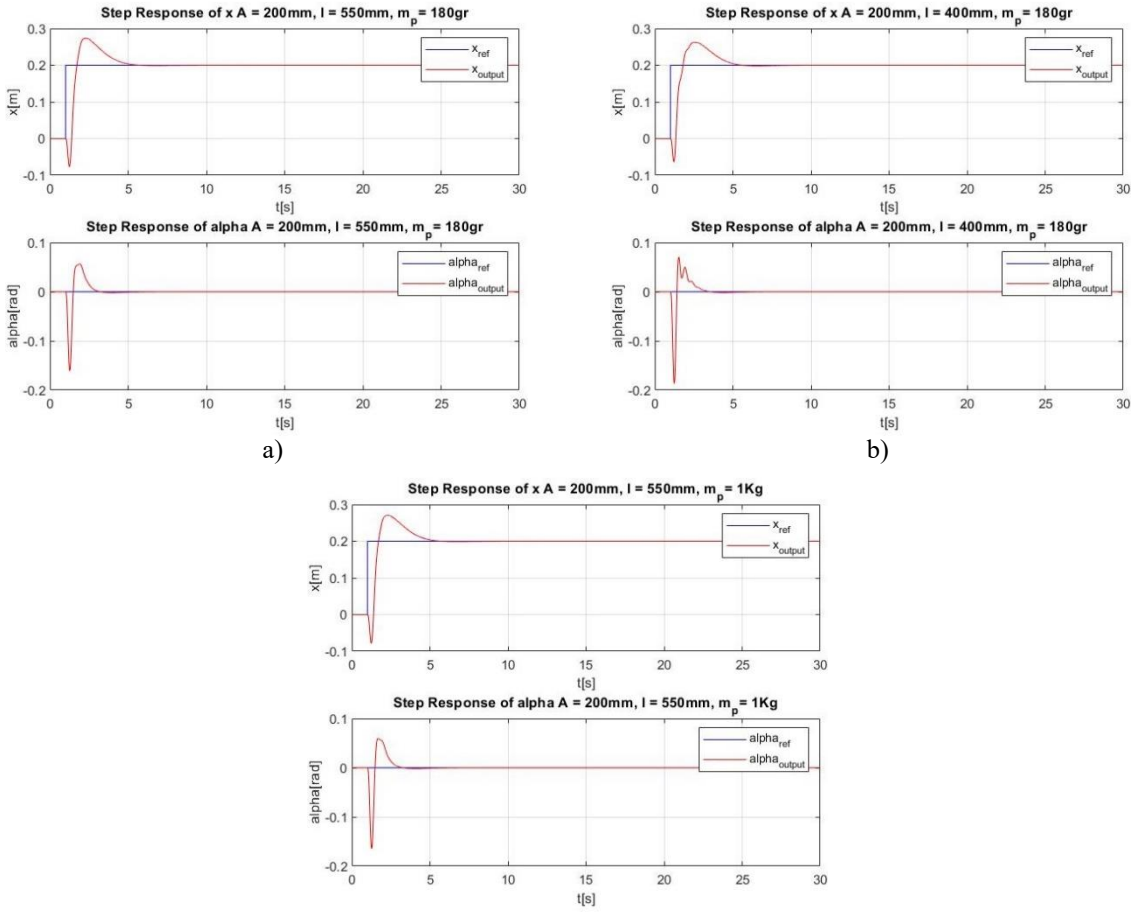


Fig. 4.1-5 Step responses of the pneumatic model controlled by the PIDs' parallel architecture. In each figure, from top to bottom, the time responses of x and α. Fig. a m_p=180gr and $l_p = 550mm$. Fig. b m_p=180gr and $l_p = 440mm$. Fig. c m_p=1Kg and $l_p = 550mm$.

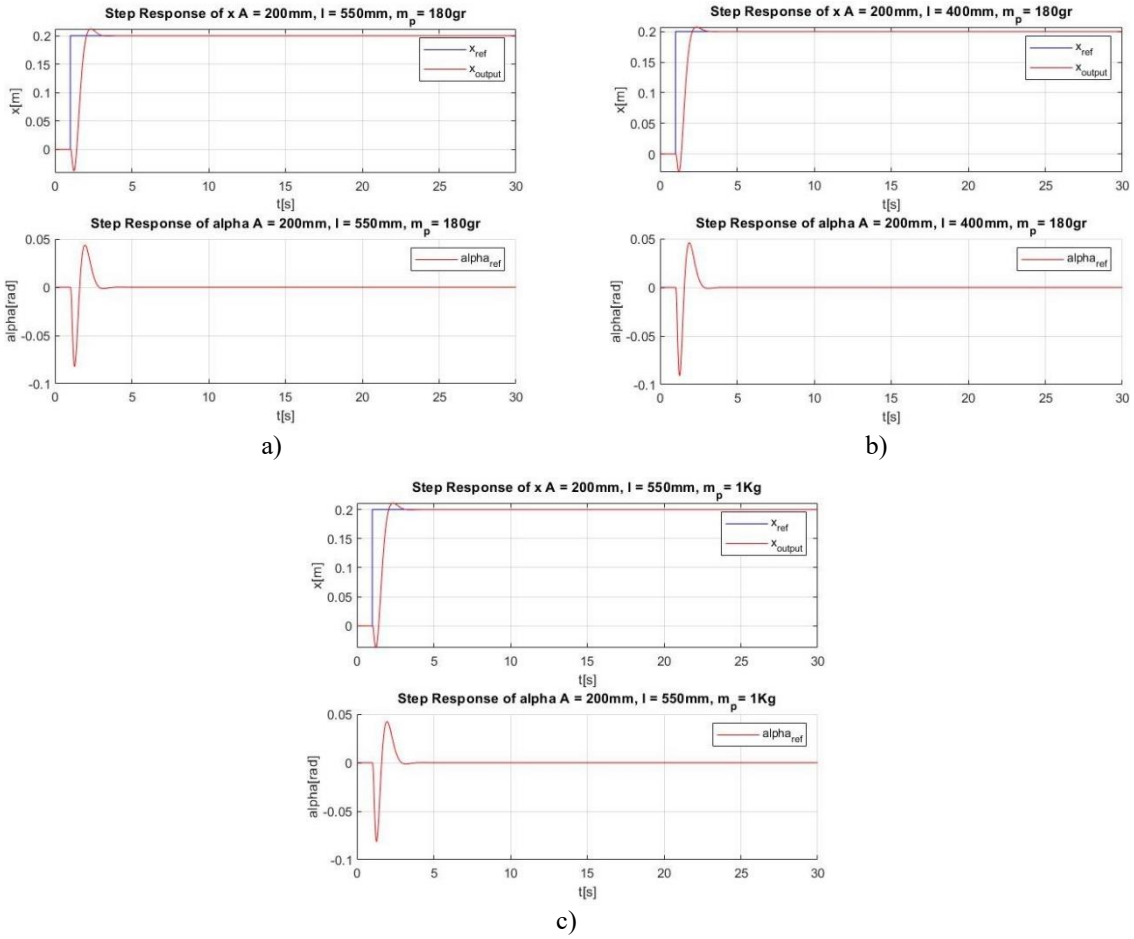
Also in this case, like in the cascade control (figure 4.1-4), the most important difference is in figure 4.1-5 b) that highlights an oscillated overshoot in both time responses of x and α, while in figures 4.1-5 a) and c), although some very low differences are present, the time responses are practical the same. Also in this case a shortest rod length involves some oscillations in the time response.

The values related to settling time and over/undershoot of the responses in figure 4.1-5 are resumed in table 4.1-5.

		Settling time 5% $[s]$	Overshoot $x[m], \alpha[rad]$	Undershoot $x[m], \alpha[rad]$
$l_p = 550mm$ $m_p = 180gr$	x_{output}	4.40	0.273	-0.076
	α_{output}	–	0.056	-0.160
$l_p = 400mm$ $m_p = 180gr$	x_{output}	4.60	0.262	-0.063
	α_{output}	–	0.070	-0.186
$l_p = 550mm$ $m_p = 1Kg$	x_{output}	4.45	0.270	-0.078
	α_{output}	–	0.059	-0.164

Table 4.1-5 Values of settling time and over/undershoot of step responses in figure 4.1-5

Figure 4.1-6 shows the step responses related to the system controlled by the LQR architecture.



a) b)

c)

Fig. 4.1-6 Step responses of the pneumatic model controlled by the LQR architecture.
In each figure, from top to bottom, the time responses of x and α. Fig. a m_p=180gr and $l_p = 550mm$. Fig. b m_p=180gr and $l_p = 440mm$. Fig. c m_p=1Kg and $l_p = 550mm$.

As can be seen in figure 4.1-6, the LQR control architecture is able to perform a good time response in all the configurations a), b) and c). Unlike the cascade and parallel architectures scenario shown in figures 4.1-4 and 5, in this case, there are no differences in the three step responses because the LQR control architecture manages the variations on m_p and l_p in a better way than the cascade and parallel ones.

The values related to settling time and over/undershoot of the responses in figure 4.1-6 are resumed in table 4.1-6.

		Settling time 5% $[s]$	Overshoot $x[m], \alpha[rad]$	Undershoot $x[m], \alpha[rad]$
$l_p = 550mm$ $m_p = 180gr$	x_{output}	2.49	0.211	-0.037
	α_{output}	–	0.043	-0.082
$l_p = 400mm$ $m_p = 180gr$	x_{output}	2.35	0.205	-0.026
	α_{output}	–	0.052	-0.094
$l_p = 550mm$ $m_p = 1Kg$	x_{output}	2.49	0.212	-0.038
	α_{output}	–	0.043	-0.082

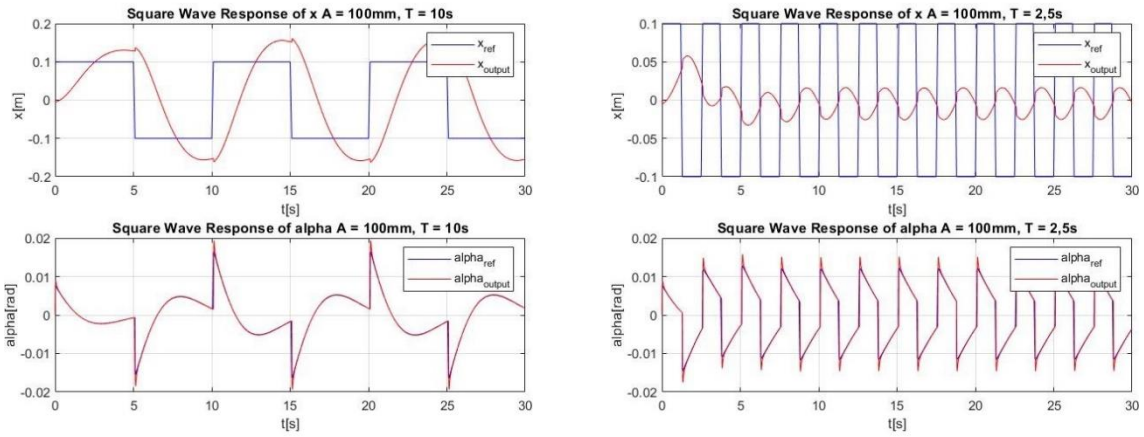
Table 4.1-6 Values of settling time and over/undershoot of step responses in figure 4.1-6

4.2 Square wave response

Another set of the performed numerical simulation concerns a square wave as input signal for the controlled system. The providing input signals is a square wave with a peak amplitude of 100 mm and a variable period of 10 s and 2.5 s, in order to analyze the behaviour of the system changing the frequency of the wave. As in the step response case, all the control architectures (cascade, parallel and LQR) are tested for these two type of input signals.

4.2.1 Square wave response: electrical actuation model

In this section the square wave responses of the electrical model are analyzed. Let's have a look at the square wave responses shown in figure 4.2-1, of the system controlled by the PIDs' cascade architecture.



a) b)

Fig. 4.2-1 Square wave responses of the electrical model controlled by the PIDs' cascade architecture. In each figure, from top to bottom, the time responses of x and α. Fig. a $A = 100mm$ and $T = 10s$. Fig. b $A = 100mm$ and $T = 2.5s$.

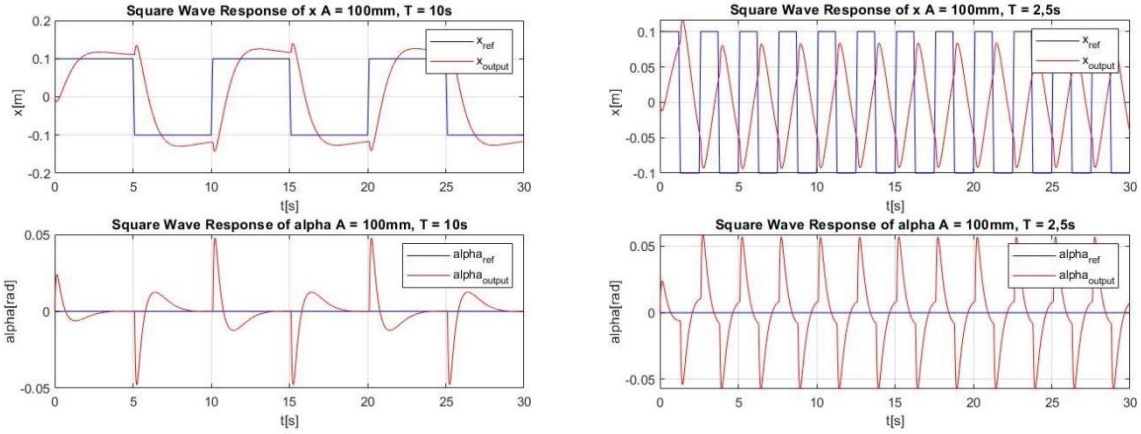
In both configurations ($T = 10\ s$ and $T = 2.5\ s$) the system immediately reacts against the change of the signal. Looking at the time response of the position x in both the configurations (figure 4.2-1 a) and b)), when the blue line, that represents the reference signal for the position (x_{ref}), changes its direction, the red line (x_{output} signal) quickly follows the path change. Unlike the similar behaviour for the two configuration about the rapidity of the system to follow the input signal, a change in the period of the square wave highlights a difference in the two x_{output} signals. In the case of a square wave with $T = 10\ s$ (figure 4.2-1a), the x_{output} signal (red line) exceeds the peak value of the x_{ref} signal; while in the case at $T = 2.5\ s$ (figure 4.2-1b), the red line is not able to reach the peak value of the blue line. This means that, a decreasing value of the period (increasing frequency), around 2.5 s, implies that the output signal will not reach the reference signal but it will quickly able to follow the direction change of the reference signal.

The peak values related to the responses in figure 4.2-1 are resumed in table 4.2-1.

		Peak value $[m]$
$T = 10s$	x_{output}	0.137
$T = 2.5s$	x_{output}	0.015

Table 4.2-1 Peak values of the step responses in figure 4.2-1

In figure 4.2-2, the square wave responses of the system controlled by the PIDs' parallel architecture are shown.



a) b)

Fig. 4.2-2 Square wave responses of the electrical model controlled by the PIDs' parallel architecture. In each figure, from top to bottom, the time responses of x and α. Fig. a $A = 100mm$ and $T = 10s$. Fig. b $A = 100mm$ and $T = 2.5s$.

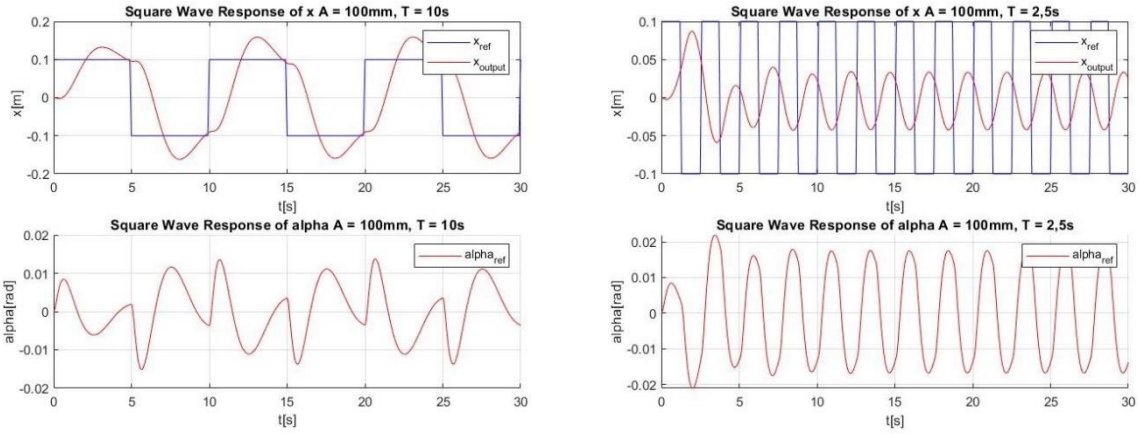
Unlike the previous case (figure 4.2-1), the parallel architecture gives the possibility to the system to reach the peak value of the reference signal x_{ref} also in the case at $T = 2.5\ s$ (figure 4.2-2 b). Although there is a low difference between the peaks of the blue and red lines, the peak value of the x_{output} signal (red) is very close to the reference one (blue, x_{ref}). Therefore, the parallel architecture proves to be quicker than the cascade one to follows the reference signal in terms of time and amplitude.

The peak values related to the responses in figure 4.2-2 are resumed in table 4.2-2.

		Peak value $[m]$
$T = 10s$	x_{output}	0.13
$T = 2.5s$	x_{output}	0.08

Table 4.2-2 Peak values of the step responses in figure 4.2-2

Also the system controlled with the LQR architecture was tested with a square wave input signals, and its time responses are shown in figure 4.2-3.



a) b)

Fig. 4.2-3 Square wave responses of the electrical model controlled by the LQR architecture. In each figure, from top to bottom, the time responses of x and α. Fig. a $A = 100mm$ and $T = 10s$. Fig. b $A = 100mm$ and $T = 2.5s$.

As in the cascade architecture, also in this case the LQR architecture doesn't provide to the system the ability to reach the peak value of the reference signal at low value of the period (figure 4.2-3 b)).

The peak values related to the responses in figure 4.2-3 are resumed in table 4.2-3.

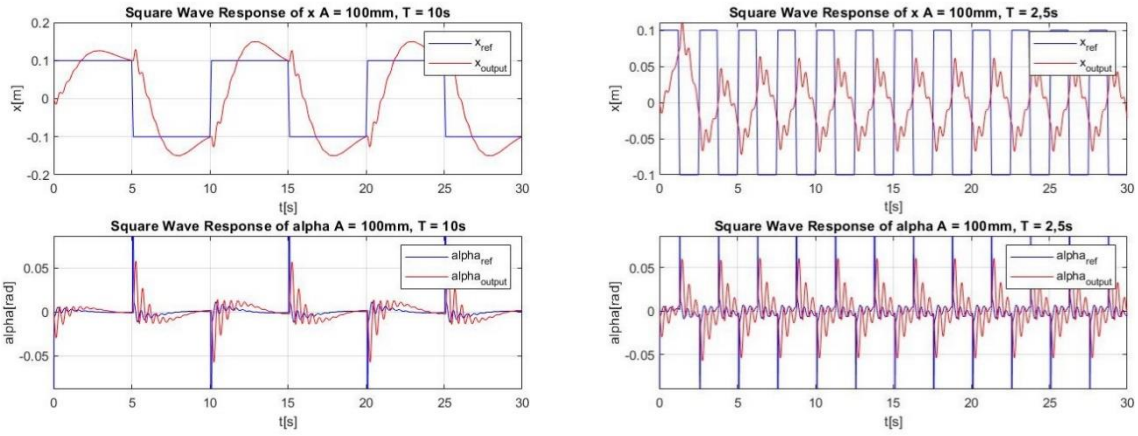
		Peak value $[m]$
$T = 10s$	x_{output}	0.15
$T = 2.5s$	x_{output}	0.04

Table 4.2-3 Peak values of the step responses in figure 4.2-3

4.2.2 Square wave response: pneumatic actuation model

In this section the numerical simulation with a square wave input signal are repeated for the pneumatic actuation model.

Figure 4.2-4 shows the square wave responses of the system controlled by the PIDs' cascade architecture.



a) b)

Fig. 4.2-4 Square wave responses of the pneumatic model controlled by the PIDs' cascade architecture. In each figure, from top to bottom, the time responses of x and α. Fig. a $A = 100mm$ and $T = 10s$. Fig. b $A = 100mm$ and $T = 2.5s$.

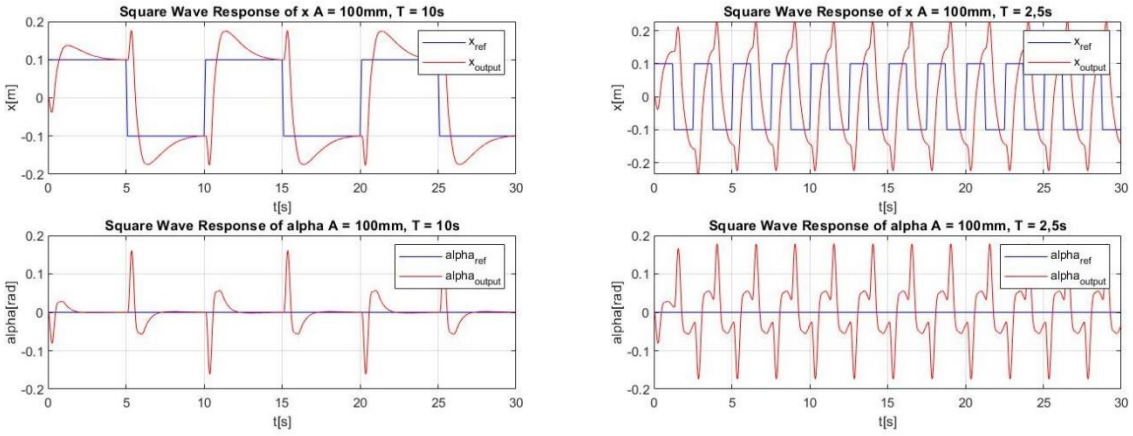
As can be seen in figure 4.2-4 a) and b), although there are some oscillations in the time responses of x and α, the system has a similar behaviour like the one in figure 4.2-1 in terms of time and amplitude. Like in the electrical model, also in the pneumatic one the cascade architecture doesn't provide to the system the ability to follows the reference signal in terms of amplitude. Looking at figure 4.2-4 b), the red line (x_{output} signal) approximately reaches the half amplitude of the blue line (x_{ref} signal), before x_{ref} changes direction.

The peak values related to the responses in figure 4.2-4 are resumed in table 4.2-4.

		Peak value $[m]$
$T = 10s$	x_{output}	0.14
$T = 2.5s$	x_{output}	0.06

Table 4.2-4 Peak values of the step responses in figure 4.2-4

The square wave responses of the system controlled by the PIDs' parallel architecture are shown in figure 4.2-5.



a) b)

Fig. 4.2-5 Square wave responses of the pneumatic model controlled by the PIDs' parallel architecture. In each figure, from top to bottom, the time responses of x and α. Fig. a $A = 100mm$ and $T = 10s$. Fig. b $A = 100mm$ and $T = 2.5s$.

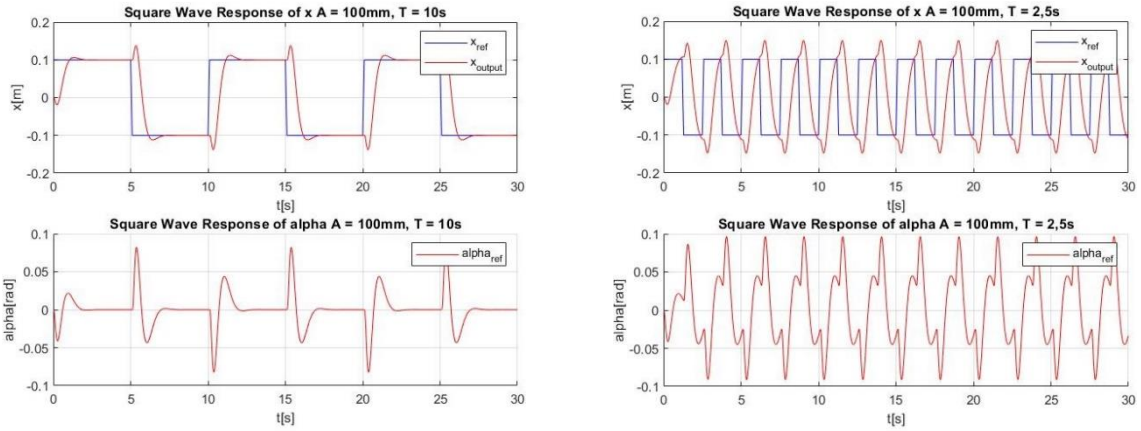
In this case the behaviour of the system (figure 4.2-5 b)) gives the proof about the ability of the parallel architecture to provide a quick response, in terms of time and amplitude, like in the electrical model controlled by this type of control technique.

The peak values related to the responses in figure 4.2-5 are resumed in table 4.2-5.

		Peak value $[m]$
$T = 10s$	x_{output}	0.17
$T = 2.5s$	x_{output}	0.22

Table 4.2-5 Peak values of the step responses in figure 4.2-5

Let's have a look at the square wave responses shown in figure 4.2-6, of the system controlled by the LQR architecture.



a) b)

Fig. 4.2-6 Square wave responses of the pneumatic model controlled by the LQR architecture. In each figure, from top to bottom, the time responses of x and α. Fig. a $A = 100mm$ and $T = 10s$. Fig. b $A = 100mm$ and $T = 2.5s$.

Like in the previous case (figure 4.2-5), also with the LQR architecture the system output signal (x_{output}) is able to follow the reference one (x_{ref}) as it shown in figure 4.2-6 a) and b). Moreover, as can be seen in figure 4.2-6 a), the LQR architecture provides a better response, in terms of overshoot and settling time, than the parallel one.

The peak values related to the responses in figure 4.2-6 are resumed in table 4.2-6.

		Peak value $[m]$
$T = 10s$	x_{output}	0.13
$T = 2.5s$	x_{output}	0.14

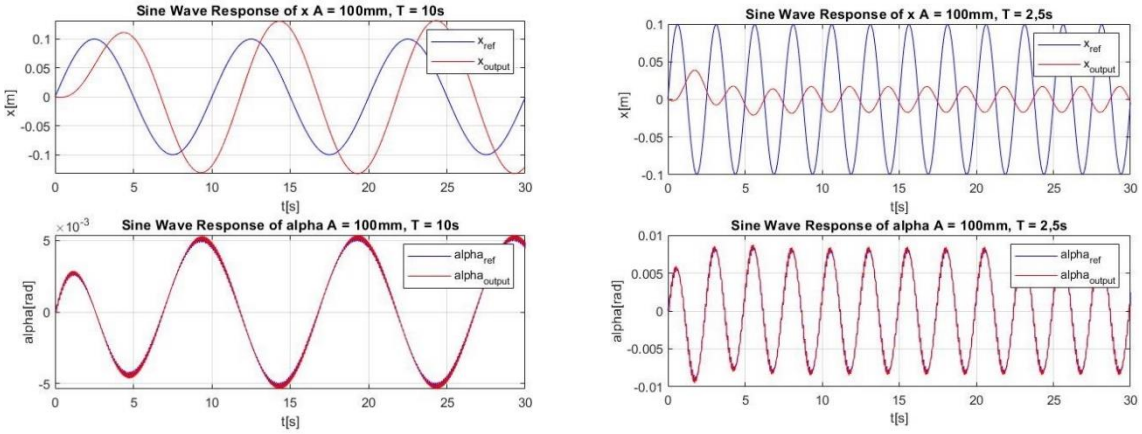
Table 4.2-6 Peak values of the step responses in figure 4.2-6

4.3 Sine wave response

The last set of the performed numerical simulation concerns a sine wave as input signal for the controlled system. The providing input signals is a sine wave with a peak amplitude of 100 mm and a variable period of 10 s and 2.5 s, like the square wave signals in section 4.2. As in the previous cases, all the control architectures (cascade, parallel and LQR) are tested for the sine wave input signals.

4.3.1 Sine wave response: electrical actuation model

Let's have a look at the sine wave responses shown in figure 4.3-1, of the system controlled by the PIDs' cascade architecture.



a) b)

Fig. 4.3-1 Sine wave responses of the electrical model controlled by the PIDs' cascade architecture. In each figure, from top to bottom, the time responses of x and α. Fig. a $A = 100mm$ and $T = 10s$. Fig. b $A = 100mm$ and $T = 2.5s$.

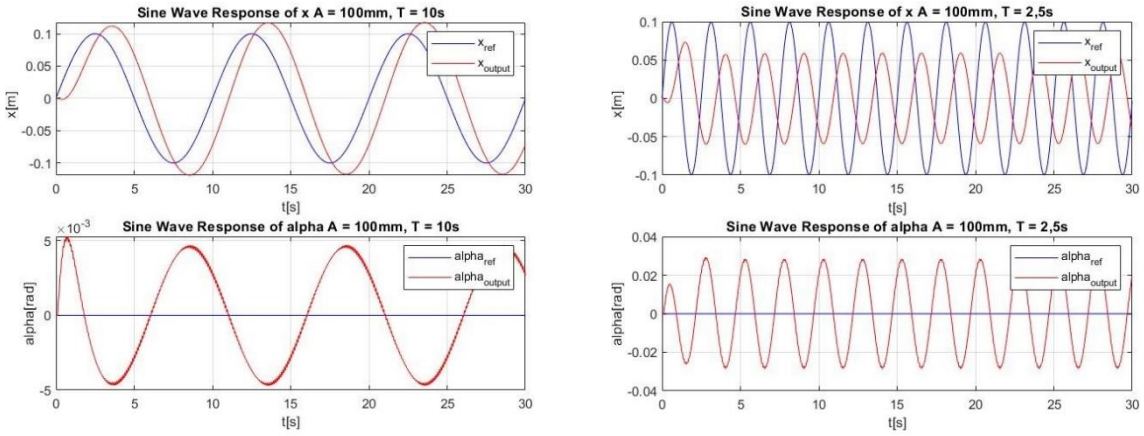
Unlike the square wave responses, the sine wave ones gives an easy understanding about the time response delay. Looking at figure 4.3-1 a), the x_{output} signal (red line) has a time delay around two seconds respect to the x_{ref} signal (blue line). As can be seen in figure 4.3-1 b), this delay increases by decreasing the period of the wave signal. In the case at $T = 2.5s$, the delay produces a sort of phase inversion between the two signals (x_{ref} and x_{output}).

The time delay values related to the responses in figure 4.3-1 are resumed in table 4.3-1.

		Time delay $[s]$
$T = 10s$	x_{output}	2.1
$T = 2.5s$	x_{output}	1.1

Table 4.3-1 Time delay values of the step responses in figure 4.3-1

In figure 4.3-2, the sine wave responses of the system controlled by the PIDs' parallel architecture are shown.



a) b)

Fig. 4.3-2 Sine wave responses of the electrical model controlled by the PIDs' parallel architecture. In each figure, from top to bottom, the time responses of x and α. Fig. a $A = 100mm$ and $T = 10s$. Fig. b $A = 100mm$ and $T = 2.5s$.

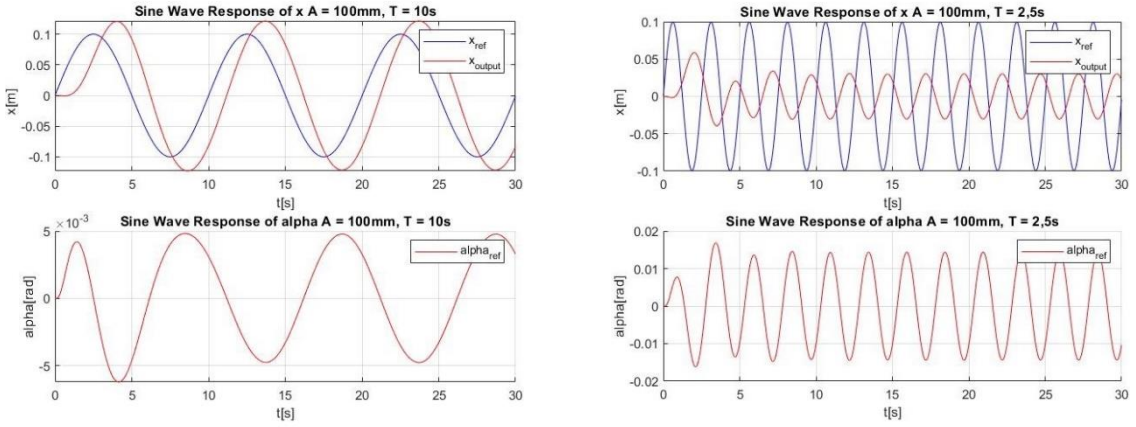
Also in the parallel architecture the time delay between x_{ref} and x_{output} is present, but looking at figure 4.3-2 it is lower than the time delay caused by the cascade architecture. However, the parallel architecture, contrary to the cascade one, provides to the system the ability to almost reach the peak amplitude of the x_{ref} signal thanks to the low time delay of x_{output} signal.

The time delay values related to the responses in figure 4.3-2 are resumed in table 4.3-2.

		Time delay $[s]$
$T = 10s$	x_{output}	1.1
$T = 2.5s$	x_{output}	1.0

Table 4.3-2 Time delay values of the step responses in figure 4.3-2

The sine wave responses of the system controlled by the LQR architecture are shown in figure 4.3-3.



a) b)

Fig. 4.3-3 Sine wave responses of the electrical model controlled by the LQR architecture. In each figure, from top to bottom, the time responses of x and α. Fig. a $A = 100mm$ and $T = 10s$. Fig. b $A = 100mm$ and $T = 2.5s$.

At high values of the wave period ($T = 10\ s$, figure 4.3-3 a), the LQR architecture provides to the system a good capability to follow the reference signal, producing a short time delay between x_{ref} and x_{output}. However, in the case at $T = 2.5\ s$ (figure 4.3-3 b), the time delay of x_{output} signal becomes really high, because it results in a phase inversion between the reference and the output signals.

The time delay values related to the responses in figure 4.3-3 are resumed in table 4.3-3.

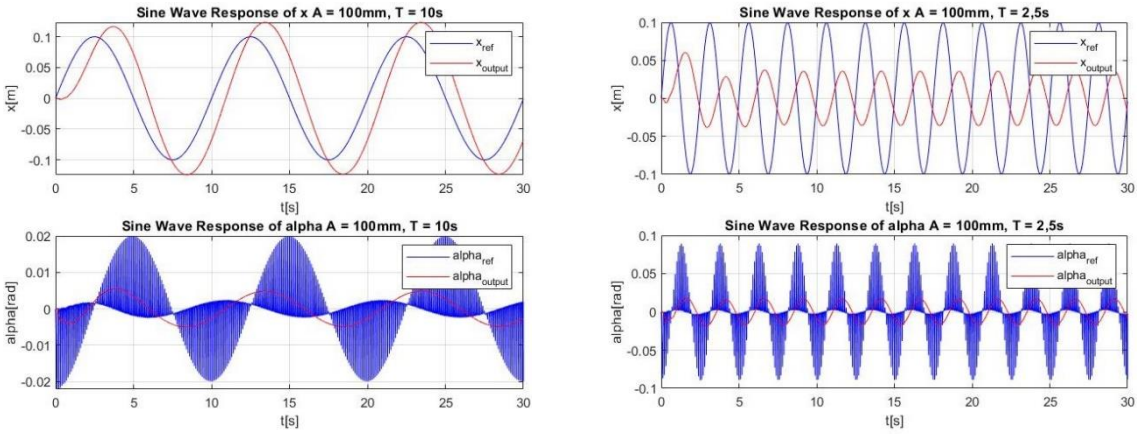
		Time delay $[s]$
$T = 10s$	x_{output}	1.1
$T = 2.5s$	x_{output}	1.5

Table 4.3-3 Time delay values of the step responses in figure 4.3-3

4.3.2 Sine wave response: pneumatic actuation model

Like in the sections 4.1 and 4.2, the responses are performed also on the pneumatic model.

Figure 4.3-4 shows the sine wave responses of the system controlled by the PIDs' cascade architecture.



a) b)

Fig. 4.3-4 Sine wave responses of the pneumatic model controlled by the PIDs' cascade architecture. In each figure, from top to bottom, the time responses of x and α. Fig. a $A = 100mm$ and $T = 10s$. Fig. b $A = 100mm$ and $T = 2.5s$.

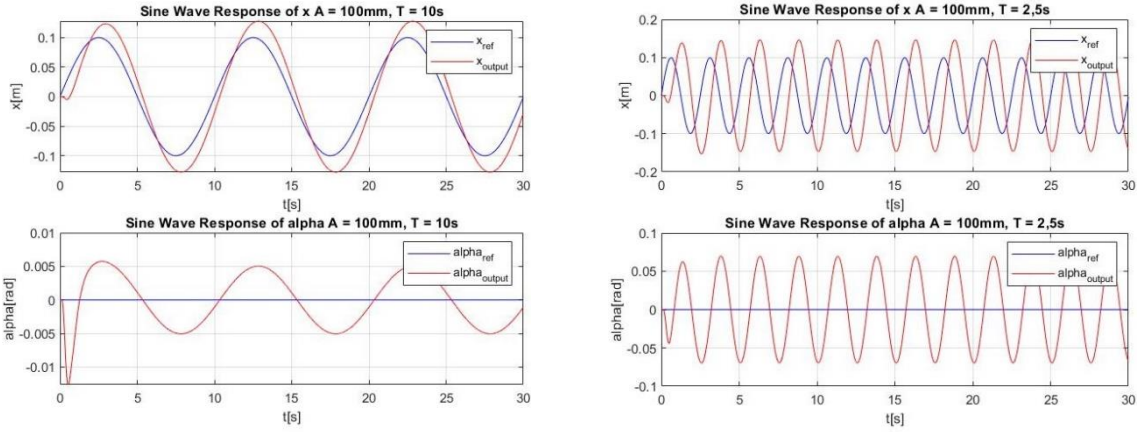
Looking at figures 4.3-4 a) and b), the x_{output} signals (red lines) have more or less the same behaviour shown in figures 4.3-1 a) and b) of the electrical model controlled by the PIDs' cascade architecture. There is a time delay between the two signals (x_{ref} in blue and x_{output} in red), that increases by decreasing the value of the wave period. Moreover, also in the case at $T = 2.5\ s$ the output signals doesn't reach the peak amplitude of the reference one, before x_{ref} changes direction.

The time delay values related to the responses in figure 4.3-4 are resumed in table 4.3-4.

		Time delay $[s]$
$T = 10s$	x_{output}	0.9
$T = 2.5s$	x_{output}	1.0

Table 4.3-4 Time delay values of the step responses in figure 4.3-4

In figure 4.3-5, the sine wave responses of the system controlled by the PIDs' parallel architecture are shown.



a) b)

Fig. 4.3-5 Sine wave responses of the electrical model controlled by the PIDs' parallel architecture. In each figure, from top to bottom, the time responses of x and α. Fig. a $A = 100mm$ and $T = 10s$. Fig. b $A = 100mm$ and $T = 2.5s$.

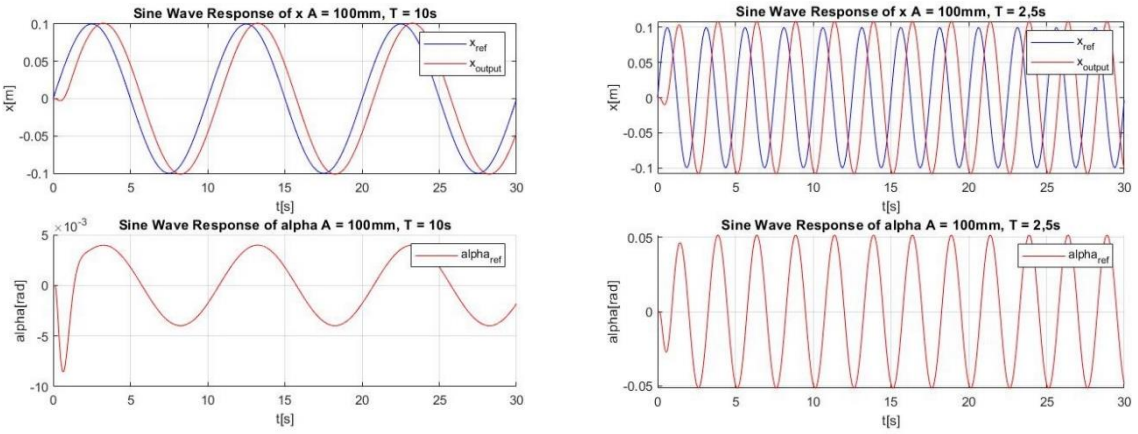
As can be seen in figures 4.3-5 a) and b), the parallel architecture provides the system with a good capability to follow the reference signal, producing a very short time delay between x_{ref} and x_{output}. Moreover, looking at figure 4.3-5 b), the red line (x_{output}) is able to reach and exceed the maximum amplitude of the blue line (x_{ref}) due to the low time delay between the output and reference signals.

The time delay values related to the responses in figure 4.3-5 are resumed in table 4.3-5.

		Time delay $[s]$
$T = 10s$	x_{output}	0.3
$T = 2.5s$	x_{output}	0.7

Table 4.3-5 Time delay values of the step responses in figure 4.3-5

The last simulation is shown in figure 4.3-6, that concerns the sine wave responses of the system controlled by the LQR architecture.



a) b)

Fig. 4.3-6 Sine wave responses of the electrical model controlled by the LQR architecture. In each figure, from top to bottom, the time responses of x and α. Fig. a $A = 100mm$ and $T = 10s$. Fig. b $A = 100mm$ and $T = 2.5s$.

Figures 4.3-6 a) and b) highlight the optimal control properties of the LQR architecture. As can be seen, the time delay of the output signals is very low in both cases, period equal to 10 and 2.5 s respectively. Moreover, the x_{output} signals respects the amplitude value provided by the x_{ref} signals.

The time delay values related to the responses in figure 4.3-6 are resumed in table 4.3-6.

		Time delay $[s]$
$T = 10s$	x_{output}	0.7
$T = 2.5s$	x_{output}	0.7

Table 4.3-6 Time delay values of the step responses in figure 4.3-6

5 Results

The aim of this chapter is to give multiple comparisons between the used control techniques (cascade, parallel and LQR) and between the used actuations in the two models (electrical and pneumatic).

Initially, in section 5.1, the comparison between the control techniques in the model with electrical actuation is provide, while the same comparison is done in section 5.2 for the model with pneumatic actuation. Finally, in section 5.3 a comparison between the two actuation is shown. All the comparisons use the results of the step responses presented in chapter 4, in particular, only the configuration with $m_p = 180$ gr and $l_p = 550\ mm$ is considered.

The performances of the step responses can be classified in terms of settling time and over/under shoot.

5.1 Control techniques comparison in electrical actuation

Figure 5.1-1 shows the step responses of the model, with electrical actuation, controlled by PIDs' cascade and parallel architectures and by the LQR one.

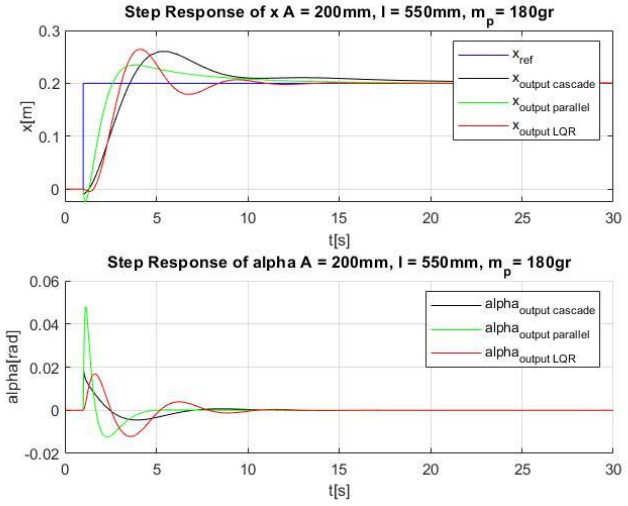


Fig. 5.1-1 Step responses of the system with electrical actuation controlled by the three techniques. Top, the x response. Bottom, the α response.

Looking at the x step responses, the technique that performs the best settling time is the LQR, while in the worst case is the cascade one. On the other hand, α responses, the technique that performs the best settling time is the parallel architecture, while in the worst case is the LQR one.

The results changes in terms of over/under shoot, the LQR and the parallel technique create the highest overshoots in the x response and in the α response, respectively. While the lowest peaks are created by the parallel technique in the x response and by the LQR technique in the α response. The values related to settling time and over/undershoot of the responses in figure 5.1-1 are resumed in table 5.1-1.

		Settling time 5% $[s]$	Overshoot $x[m], \alpha[rad]$	Undershoot $x[m], \alpha[rad]$
Cascade	x_{output}	10.20	0.260	-0.009
	α_{output}	–	0.017	-0.004
Parallel	x_{output}	9.07	0.234	-0.024
	α_{output}	–	0.047	-0.012
LQR	x_{output}	5.38	0.264	-0.004
	α_{output}	–	0.016	-0.012

Table 5.1-1 Values of settling time and over/undershoot of step responses in figure 5.1-1

5.2 Control techniques comparison in pneumatic actuation

Figure 5.1-1 shows the step responses of the model with electrical actuation controlled by PIDs' cascade and parallel architectures and by the LQR one.

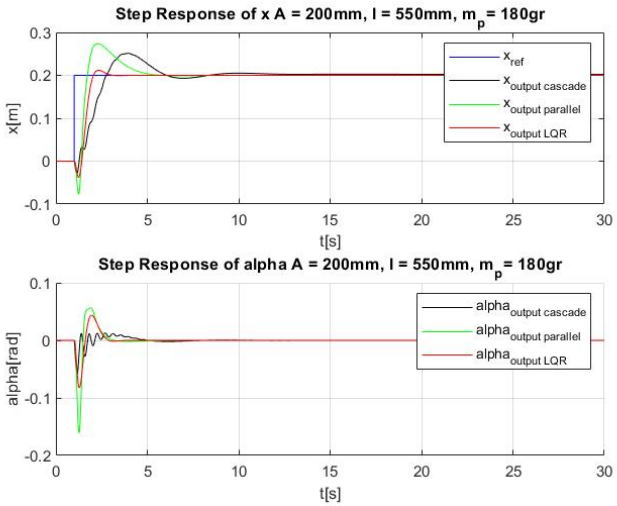


Fig. 5.2-1 Step responses of the system with pneumatic actuation controlled by the three techniques. Top, the x response. Bottom, the α response.

Looking at figure above (5.2-1), the LQR technique provides the best settling time performances for both responses, x and α respectively, while the worst cases are performed by the cascade technique, that also performs some oscillations in both responses.

On the other hand, the parallel technique performs the highest over/under shoots in the x and α responses, in contrast to the LQR and cascade ones that perform the lowest over/under shoots for the x and α responses, respectively.

The values related to settling time and over/undershoot of the responses in figure 5.2-1 are resumed in table 5.2-1.

		Settling time 5% $[s]$	Overshoot $x[m], \alpha[rad]$	Undershoot $x[m], \alpha[rad]$
Cascade	x_{output}	5.56	0.251	-0.030
	α_{output}	–	0.012	-0.059
Parallel	x_{output}	4.40	0.273	-0.076
	α_{output}	–	0.056	-0.160
LQR	x_{output}	2.49	0.211	-0.037
	α_{output}	–	0.043	-0.082

Table 5.2-1 Values of settling time and over/undershoot of step responses in figure 5.2-1

5.3 Electrical and pneumatic actuation comparison

Once that the control techniques are separately compared in the systems controlled by the electrical and pneumatic actuation, the two system are compared between them in order to highlight the best features of the two actuation types. So, let's start to analyze the step responses of the systems, with the two actuations, controlled by the PIDs' cascade architecture.

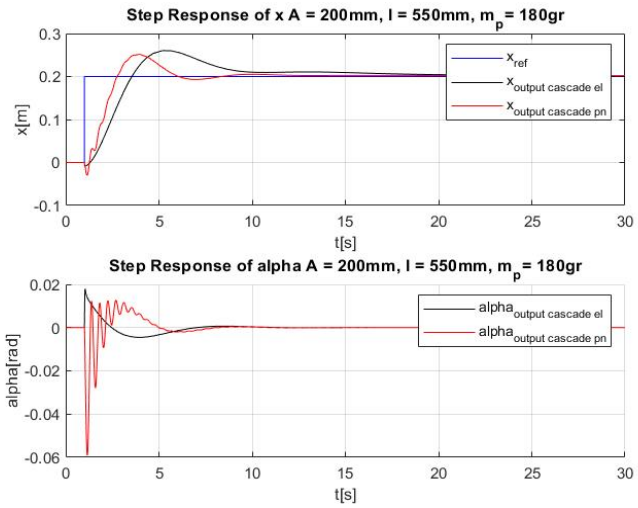


Fig. 5.3-1 Step responses of the systems with electrical and pneumatic actuation controlled by the PIDs' cascade architecture

Figure 5.3-1 highlights the smooth behaviour of the time responses of the system with electrical actuation, in contrary to the oscillating behaviours provided by the one with pneumatic actuation. Despite the oscillating behaviour, however, the pneumatic actuation provides to the system a lower settling time than the one obtained by the electrical actuation.

The values related to settling time and over/undershoot of the responses in figure 5.3-1 are resumed in table 5.3-1.

		Settling time 5% $[s]$	Overshoot $x[m], \alpha[rad]$	Undershoot $x[m], \alpha[rad]$
Electrical	x_{output}	10.20	0.260	-0.009
	α_{output}	–	0.017	-0.004
Pneumatic	x_{output}	5.56	0.251	-0.030
	α_{output}	–	0.012	-0.059

Table 5.3-1 Values of settling time and over/undershoot of step responses in figure 5.3-1

The step responses of the systems, with the two actuations, controlled by the PIDs' parallel architecture are shown in figure 5.3-2.

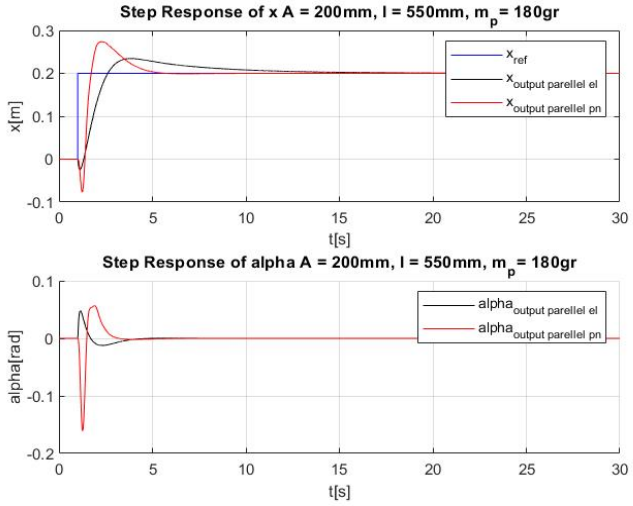


Fig. 5.3-2 Step responses of the systems with electrical and pneumatic actuation controlled by the PIDs' parallel architecture

As can be seen in figure 5.3-2, also the parallel architecture, like the cascade one, provides to the system with pneumatic actuation faster settling time than the system with electrical actuation for both the time responses, x and α respectively. However, the electrical system controlled by the parallel architecture performs a very smooth behaviour in the x and α responses, providing less oscillation and lower over/under shoots than the pneumatic actuation.

The values related to settling time and over/undershoot of the responses in figure 5.3-2 are resumed in table 5.3-2.

		Settling time 5% $[s]$	Overshoot $x[m], \alpha[rad]$	Undershoot $x[m], \alpha[rad]$
Electrical	x_{output}	9.07	0.234	-0.024
	α_{output}	–	0.047	-0.012
Pneumatic	x_{output}	4.40	0.273	-0.076
	α_{output}	–	0.056	-0.160

Table 5.3-2 Values of settling time and over/undershoot of step responses in figure 5.3-2

Finally, figure 5.3-3 provides the step responses of the systems, with the two actuations, controlled by the LQR architecture.

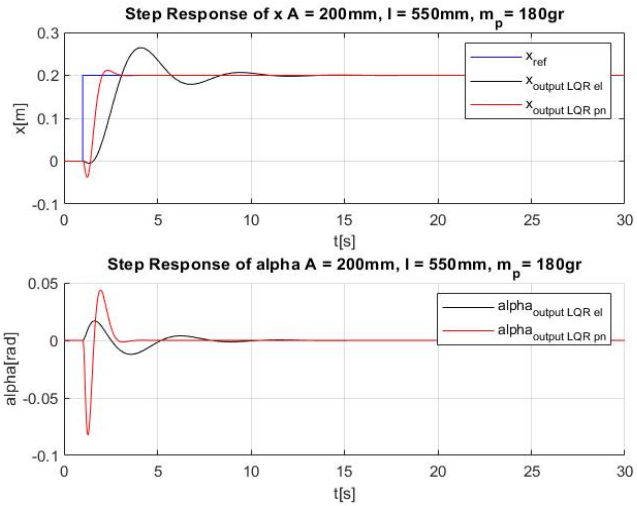


Fig. 5.3-3 Step responses of the systems with electrical and pneumatic actuation controlled by the LQR architecture

Looking at figure 5.3-3, the system with pneumatic actuation performs, thanks to the LQR technique, good value of settling time (2 seconds after the activation of x_{ref}), that is lower than the one performed by the system with electrical actuation (around 12 seconds after the activation of x_{ref}). However, the fast behaviour of the system, with pneumatic actuation, causes higher over/under shoots, in the α response, than the system with electrical actuation.

Looking at figures 5.3-1, 2 and 3, the same behaviour is more or less repeated, the fast time response of the cart position x, of the system with pneumatic actuation, causes an oscillating behaviour in the time response of the pendulum angle α, while the electrical system provides a smooth behaviour of the time response of the angle α, producing higher value of settling time for the position x than the system with pneumatic actuation.

The values related to settling time and over/undershoot of the responses in figure 5.3-3 are resumed in table 5.3-3.

		Settling time 5% $[s]$	Overshoot $x[m], \alpha[rad]$	Undershoot $x[m], \alpha[rad]$
Electrical	x_{output}	5.38	0.264	-0.004
	α_{output}	–	0.016	-0.012
Pneumatic	x_{output}	2.49	0.211	-0.037
	α_{output}	–	0.043	-0.082

Table 5.3-3 Values of settling time and over/undershoot of step responses in figure 5.3-3

Figure 5.3-4 provides support to what mentioned before, it shows the behaviours of the velocities and accelerations of the systems carts, with electrical and pneumatic actuation, controlled by the PIDs' cascade architectures.

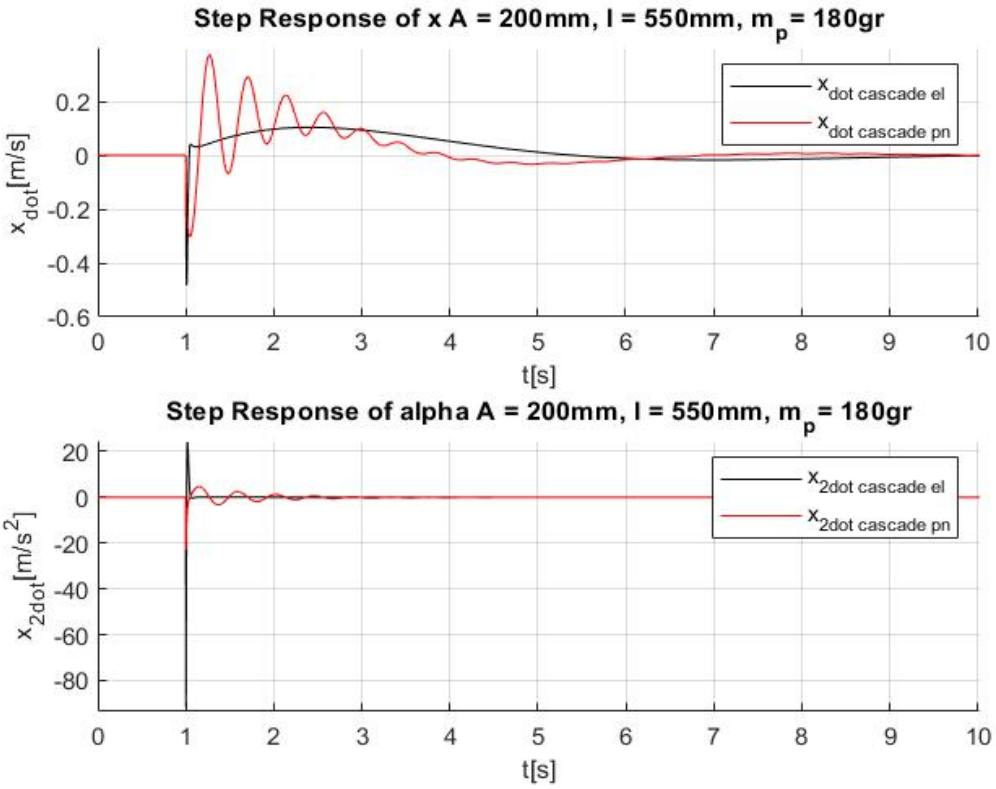


Fig. 5.3-4 Velocities (top) and accelerations (bottom) of the systems carts, with electrical and pneumatic actuation, controlled by the PIDs' cascade architecture

As can be seen, the smooth responses of x and α (black lines in figure 5.3-1), in the system with electrical actuation, are allowed thanks to a very relaxed behaviours of the velocities and accelerations of the cart (black lines), as shown in figure 5.3-4. Just after considerable undershoots in the trends of the velocity and the acceleration, these responses gently approach the zero value, that represents the equilibrium condition of the system. In contrary to this case, the oscillating responses of x and α (red lines in figure 5.3-1), in the system with the pneumatic actuation, are caused by oscillating behaviours in the velocities and accelerations trends (red lines in figure 5.3-4), that undulating approach to the equilibrium condition.

Conclusions and future works

The dissertation describe the use of several control techniques, applied to an inverted pendulum system: PIDs' cascade and parallel architectures and Static State Feedback with Pole placement or with LQR. These methods successfully achieve the stabilization of the system by satisfying the physical constraints of the plant. The aim of this is to compare the adopted control techniques and the performance of actuations, electrical and pneumatic.

Regarding the variations of the pendulum mass and rod length, the three control techniques exploiting electrical actuation show the same performances exhibiting a good robustness. On the other hand, pneumatic actuation presents similar performance only in the LQR technique.

Further considerations on the controller performance can be made on the basis of the system response to square wave. The x output signal of the system has been compared with the square wave reference signals, that have been generated with two different periods, 10 and 2.5 s. This simulations have been done to understand if the system is fast enough to reach the peak value of the reference signal when the square wave period is $10\ s$ and when the period is $2.5\ s$. The peak value of the square wave reference signal is reached by the parallel architecture with the electrical actuation and by the LQR one with the pneumatic actuation, even if the wave period change between 10 and 2,5 seconds. On contrary, the same feature is not achieved by the cascade technique applied to both actuations and by the LQR technique applied to the electrical actuation.

Moreover, considerations about the time delay between the output signal and the reference one can be made on the basis of the system response to sine wave. The pneumatic system exhibits a very low time delay for all the considered techniques. Conversely the electrical one exhibits a low time delay only with the parallel architecture.

For all these reasons, the comparison between the two types of actuation can be resumed as

- the pneumatic actuation exhibits very low values of settling time in the response of the cart position, but this generates permanent oscillations of the tilt angle of the pendulum
- the electrical actuation provides a smooth trend in both output responses due to high values of settling time

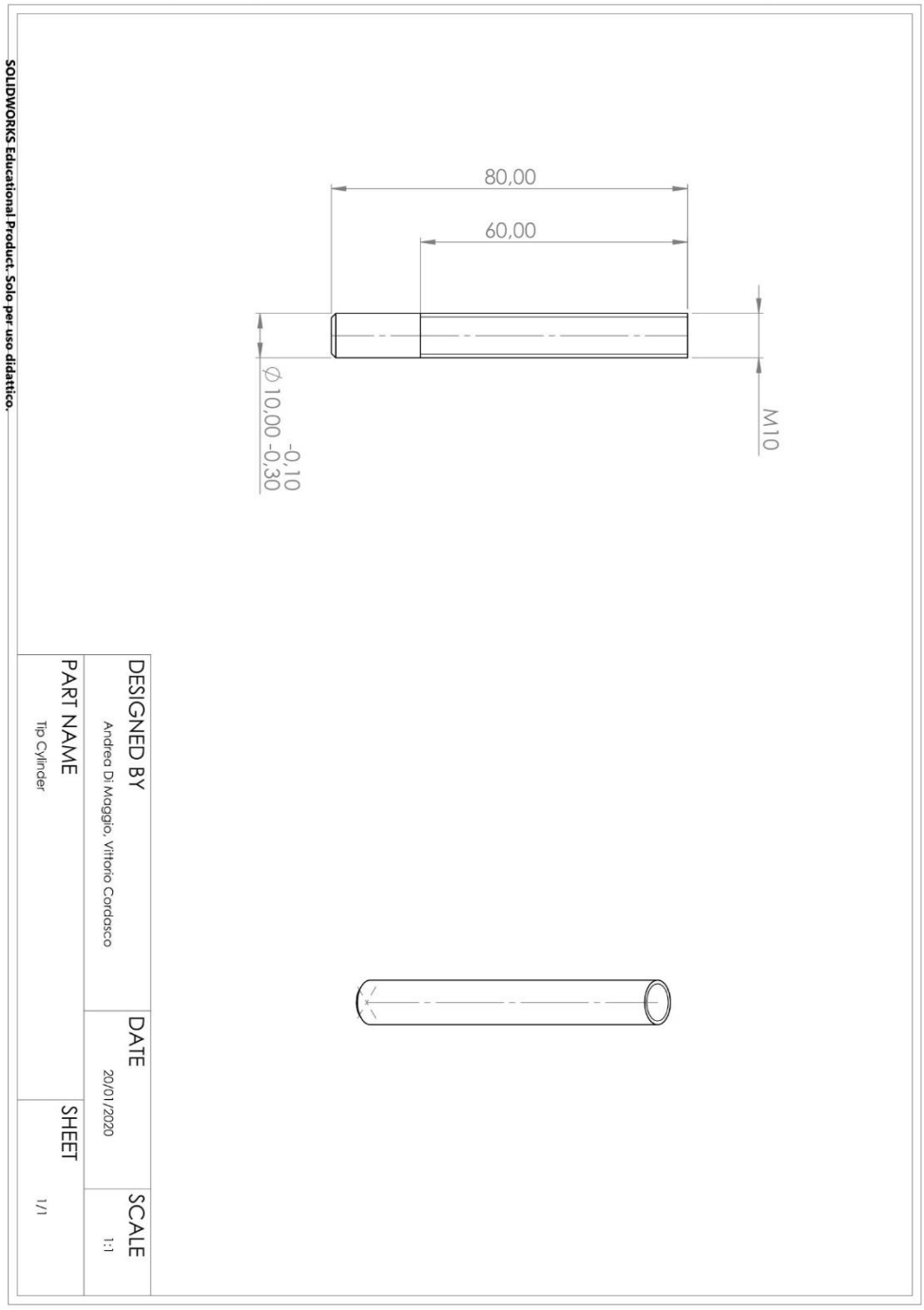
These considerations can be justified by analyzing the nature of the used actuation signals. The pneumatic system is based on the binary logic implemented by the opening and closure of the control valves, this mechanism causes oscillations in the rapid achievement of the steady state conditions. On the other hand given the reference signal, the electrical actuator is slower but it makes it possible to obtain a "smoother control" of x and α than the pneumatic cylinder. This "smoother control" can be attributed to the lower dynamics of the electric motor.

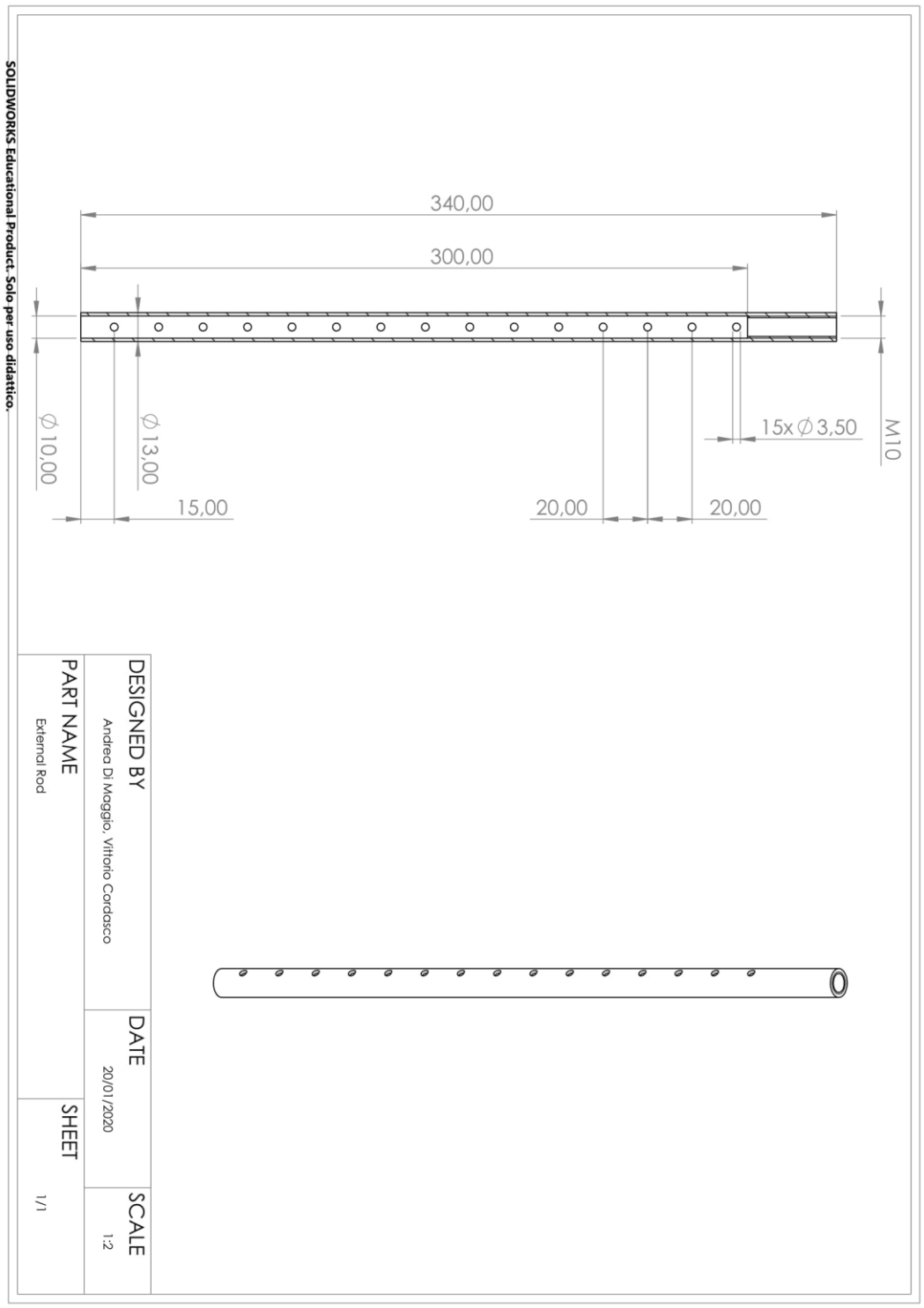
All the developed control methods are applied to the linear model of the system with electrical and pneumatic actuation. Future developments of the present study can be

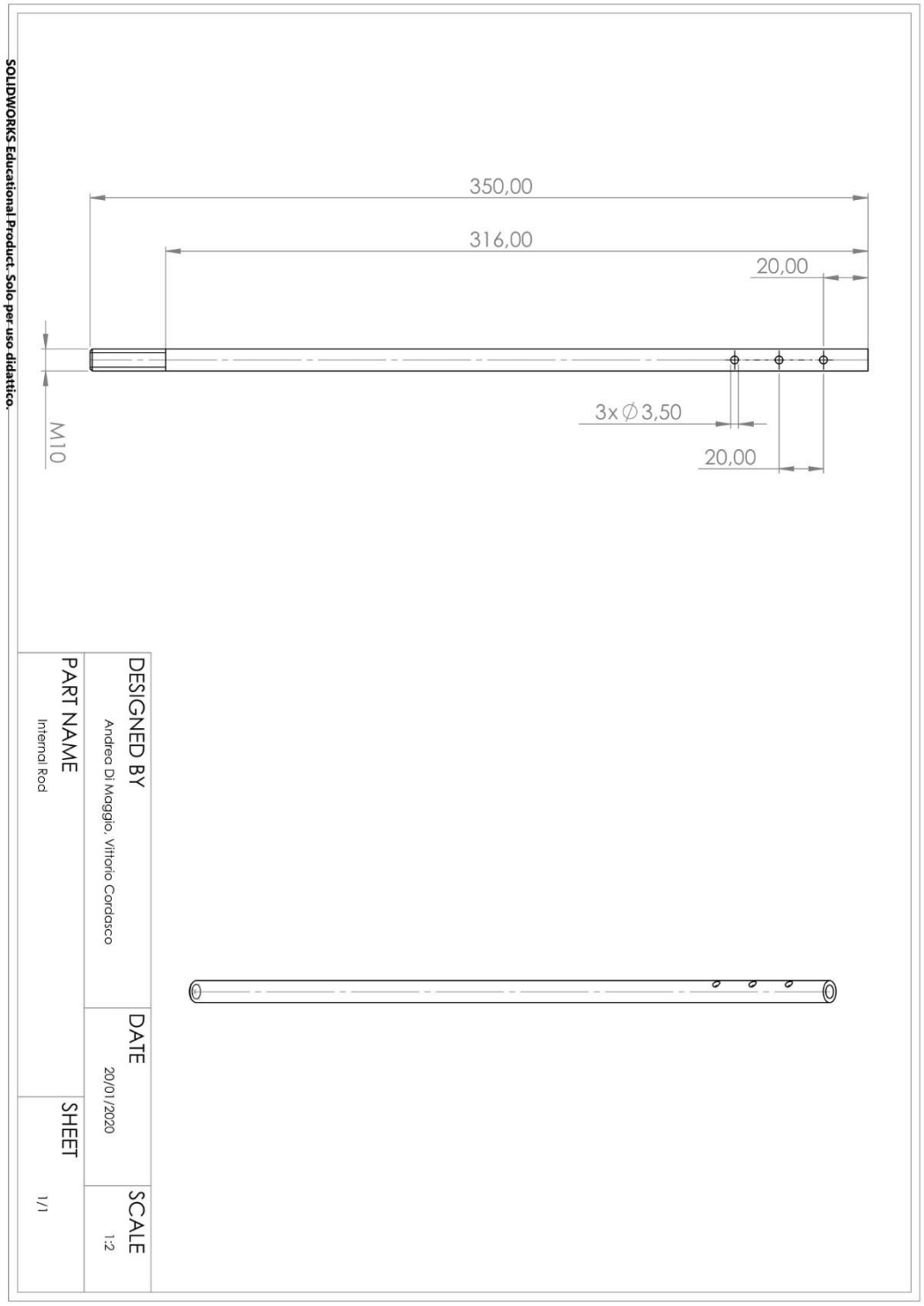
- the analysis and adaptation of these control methods also for the non-linear system, considering in the mathematical model new elements like: nonlinear functions of the pendulum dynamics (sin and cos), friction forces in all the movable parts, transients in plant dynamics and generally all other features that makes the system increasingly like the real plant
- the considered compensators could become digital controller thanks to the used of PC-based algorithm or by Microcontrollers implemented on evaluation boards, that allow the design of more complex algorithms.

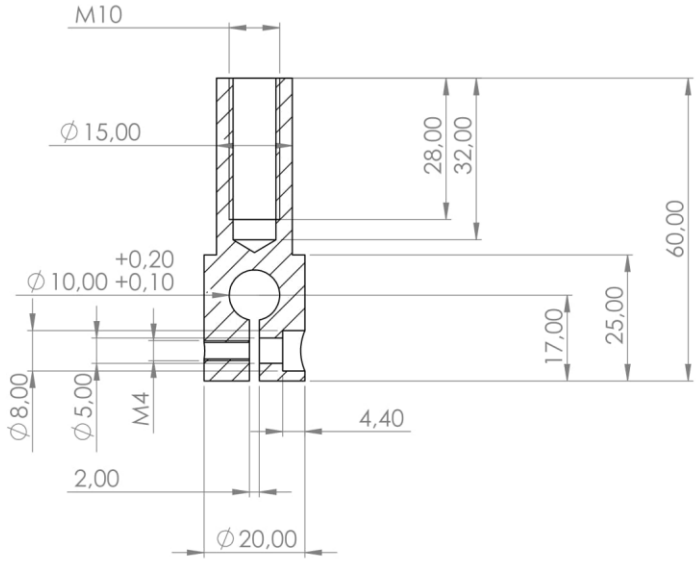
Appendix A

A Technical drawings

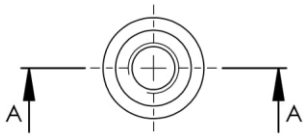


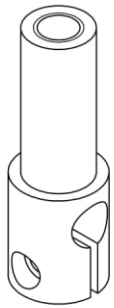

SOLIDWORKS Educational Product. Solo per uso didattico.
340,00
300,00
∅10,00
∅13,00
15,00
20,00
20,00
15x∅3,50
M10
DESIGNED BY
Andrea Di Maggio, Vittorio Cordasco
DATE
20/01/2020
SCALE
1:2
PART NAME
External Rod
SHEET
1/1


SOLIDWORKS Educational Product. Solo per uso didattico.
350,00
316,00
20,00
M10
3x ∅3,50
20,00
DESIGNED BY
Andreo Di Maggio, Vittorio Cordasco
DATE
20/01/2020
SCALE
1:2
PART NAME
Internal Rod
SHEET
1/1

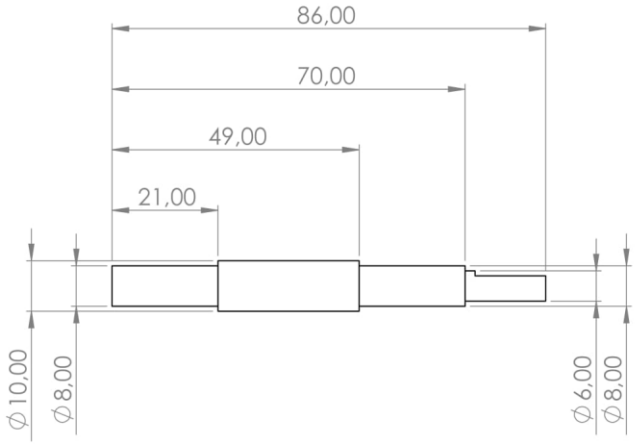


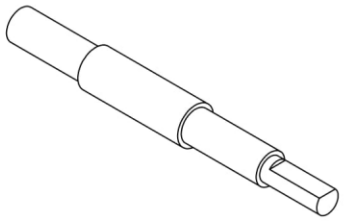
SEZIONE A-A



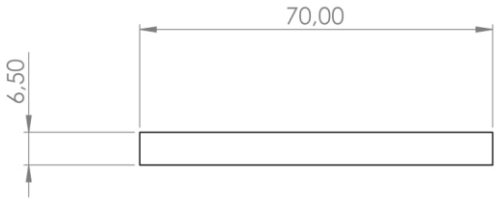


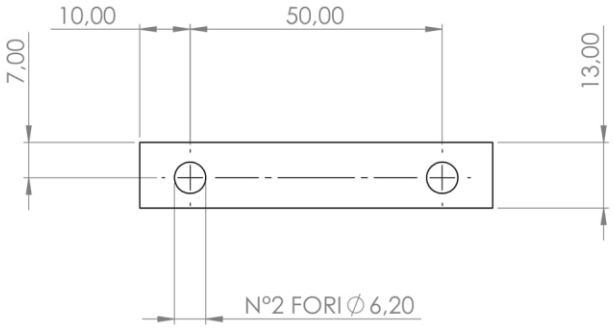
DESIGNED BY	DATE	SCALE
Andrea Di Maggio, Vittorio Cordasco	20/01/2020	1:1
PART NAME		SHEET
Two-axis Joint		1/1

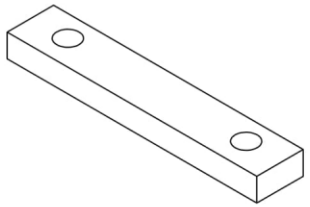




DESIGNED BY	DATE	SCALE
Andrea Di Maggio, Vittorio Cordasco	20/01/2020	1:1
PART NAME	SHEET	
Shaft	1/1	

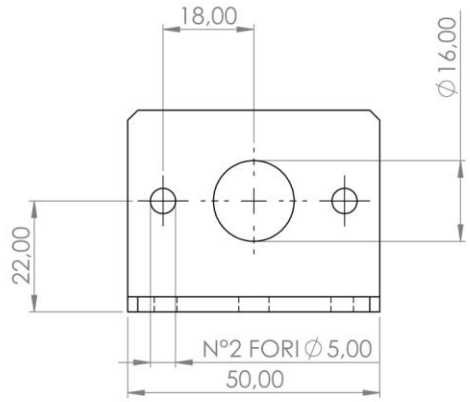


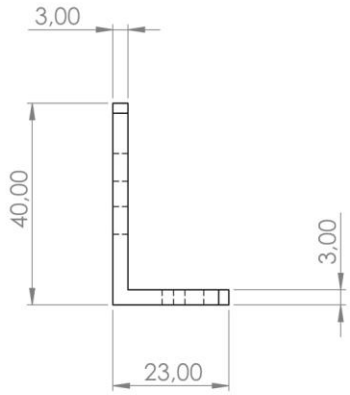


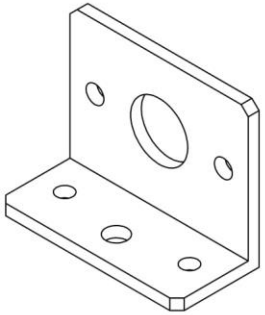


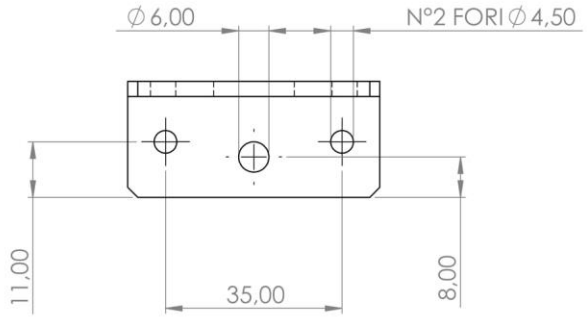
DESIGNED BY	DATE	SCALE
Andrea Di Maggio, Vittorio Cordasco	20/01/2020	1:1
PART NAME		SHEET
Support Brackets x 2		1/1

Dove non indicato, si suppongano i fori posizionati simmetricamente rispetto al centro del pezzo.





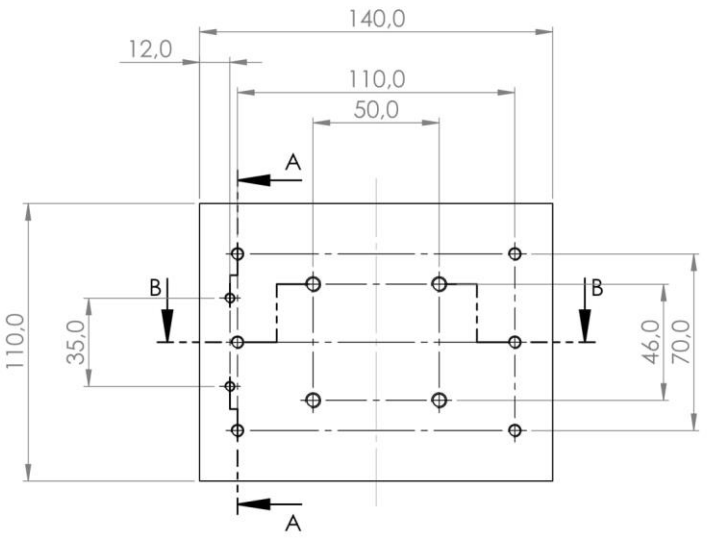


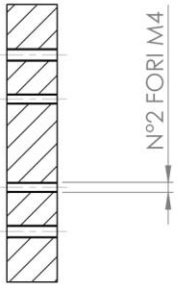


Smussi non quotati: 2x45°.

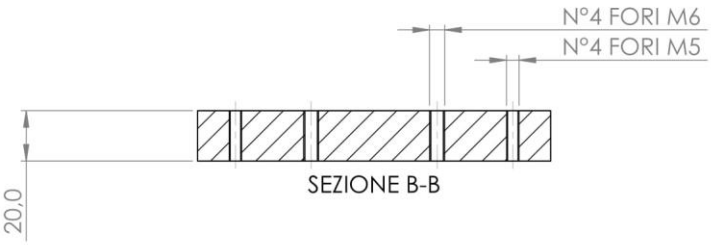
Dove non indicato, si suppongano i fori posizionati simmetricamente rispetto al centro del pezzo.

DESIGNED BY	DATE	SCALE
Andrea Di Maggio, Vittorio Cordasco	20/01/2020	1:1
PART NAME		SHEET
Mounting Bracket		1/1

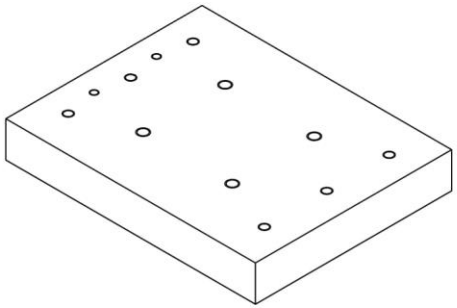




SEZIONE A-A



SEZIONE B-B



DESIGNED BY	DATE	SCALE
Andrea Di Maggio, Vittorio Cordasco	20/01/2020	1:2
PART NAME		SHEET
Connection Plate		1/1

Quotatura in scala 1:1.

Dove non indicato, si suppongano i fori posizionati simmetricamente rispetto al centro del pezzo.

Appendix B

B Matlab scripts

B1 System with electrical actuation

Model, state space representation and transfer functions

```
s = tf('s');                              % declaration transfer function 's'

% MoI = Moment of Inertia

%% Model data (Electrical Actuation)
mp   = 0.2;                               % [kg]    Concentrated mass of the pendulum
lp   = 0.5;                               % [m]     Rod length of the pendulum
Jp   = mp*(lp^2);                         % [kgm^2] MoI of mp wrt fulcrum
g    = 9.8;                               % [ms^2]  Gravity acceleration
p    = 10;                                % [mm]    Pitch of the screw
Ra   = 1.7;                               % [Ohm]   Armature resistance
K    = 0.53;                              % [Nm/A]  Torque constant
mv   = 3.1;                               % [kg]    Screw mass
r    = 0.008;                             % [m]     Screw radius
mc   = 1.8;                               % [kg]    Cart mass
Jm   = 4.3e-5;                            % [kgm^2] MoI of the rotor shaft
Jv   = (mv*(r^2))/2;                      % [kgm^2] MoI of the transmission system
X    = 2*pi*1000;                         % [-]     Auxiliary variable
Jcar = (mp+mc)*((p/X)^2);                 % [kgm^2] MoI (mp+mc) wrt rotational axis
Jtot = Jm+Jv+Jcar;                        % [kgm^2] Total MoI of the system wrt the motor
eta  = 0.96;                              % [-]     Efficiency of the transmission system

%% Integrators initial condition for the Simulink Model
x0 = [0 0 0 0 0];

%% State Space representation

load('sistema_raggiungibile.mat')         % State Space, controllable,   4 states
SYS_r = ss(sistema_raggiungibile);
FDT_r = minreal(zpk(tf(SYS_r)),1e-06);
A_r = SYS_r.a;
B_r = SYS_r.b;
C_r = SYS_r.c;
D_r = SYS_r.d;

load('sistema_NONraggiungibile.mat')      % State Space, uncontrollable, 5 states
SYS_Nr = ss(sistema_NONraggiungibile);
FDT_Nr = minreal(zpk(tf(SYS_Nr)),1e-06);
A_Nr = SYS_Nr.a;
B_Nr = SYS_Nr.b;
C_Nr = SYS_Nr.c;
D_Nr = SYS_Nr.d;

%% Controllable   SYS transfer functions

Galpha_r = zpk(FDT_r(1));                                  % alpha/Va transfer function
Gx_r     = zpk(FDT_r(2));                                  % x/Va     transfer function

%% Uncontrollable SYS transfer functions

Galpha_Nr = zpk(FDT_Nr(1));                                % alpha/Va transfer function
Gx_Nr     = zpk(FDT_Nr(2));                                % x/Va     transfer function
Galpha_x_Nr = minreal(zpk(Gx_Nr/Galpha_Nr),1e-06);         % x/alpha  transfer function
```

Script of the PIDs' cascade architecture for the Electrical model

```
%% PID for the INNER LOOP of the PIDs' cascade architecture
%% P CONTROLLER
for kp_alpha = [-100000 -130000 -160000 -190000 -200000]
   figure(6), nyquist(kp_alpha*Galpha_Nr), legend('Kp= -100000','Kp= -130000','Kp= -
   160000','Kp= -190000','Kp= -200000'),hold on
end
%% PD CONTROLLER
kd_alpha = -650;
Nd_alpha = 550000;
% PD_G_ol = open   loop tf of the inner loop with PD controller
PD_G_ol = minreal(zpk(-(200000+((kd_alpha*Nd_alpha*s)/(s+Nd_alpha)))*Galpha_Nr));
poles_PD_G_ol = pole(PD_G_ol);
figure(7), nyquist(PD_G_ol)
% PD_G_cl = closed loop tf of the inner loop with PD controller
PD_G_cl = minreal(zpk(PD_G_ol/(1+PD_G_ol)));
poles_PD_G_cl = pole(PD_G_cl);
%% PI CONTROLLER
ki_alpha = -3250000;
% PI_G_ol = open   loop tf of the inner loop with PI controller
PI_G_ol = minreal(zpk((-200000+(ki_alpha/s))*Galpha_Nr));
poles_PI_G_ol = pole(PI_G_ol);
figure(8), nyquist((-200000+(ki_alpha/s))*Galpha_Nr)
% PI_G_cl = closed loop tf of the inner loop with PI controller
PI_G_cl = minreal(zpk(PI_G_ol/(1+PI_G_ol)));
poles_PI_G_cl = pole(PI_G_cl);

%% PID CONTROLLER for the inner loop, using PID tuning tool parameters
kp_alpha_PID = -13061.5181780994;
ki_alpha_PID = -204997.97380483;
kd_alpha_PID = 60.4766490604993;
N_alpha_PID  = 215.976221913899;
Calpha = kp_alpha_PID+(ki_alpha_PID/s)+((kd_alpha_PID*N_alpha_PID*s)/(s+N_alpha_PID));

%% PID for the OUTER LOOP of the PIDs' cascade architecture
%% PID CONTROLLER for the outer loop, using PID tuning tool parameters
kp_x_PID = 0.00865988352689662;
ki_x_PID = 0.000250832346456376;
kd_x_PID = 0.0693087704565265;
N_x_PID  = 0.937526767376307;
Cx = kp_x_PID+(ki_x_PID/s)+((kd_x_PID*N_x_PID*s)/(s+N_x_PID));

%% Stability Analysis of inner and outer loops, PIDs' cascade architecture
%% open   loop transfer function of the inner loop
FDT_ol_il = minreal(zpk(Calpha*Galpha_Nr),1e-06);
p_ol_il = pole(FDT_ol_il);
figure(1), nyquist(FDT_ol_il)
% magnitude and phase margin of the open loop FDT_ol_il
figure(2), margin(Calpha*Galpha_Nr), grid on
%% closed loop transfer function of the inner loop
FDT_cl_il = minreal(zpk(FDT_ol_il/(1+FDT_ol_il)),1e-06);
p_cl_il = pole(FDT_cl_il);
%% open   loop transfer function of the outer loop
figure(9), nyquist(FDT_cl_il*Galpha_x_Nr)
FDT_ol_el = minreal(zpk(Cx*FDT_cl_il*Galpha_x_Nr),1e-06);
p_ol_el = pole(FDT_ol_el);
figure(3), nyquist(FDT_ol_el)
% magnitude and phase margin of the open loop FDT_ol_el
figure(4), margin(Cx*FDT_cl_il*Galpha_x_Nr), grid on
%% closed loop transfer function of the outer loop
FDT_cl_el = minreal(zpk(FDT_ol_el/(1+FDT_ol_el)),1e-06);
p_cl_el = pole(FDT_cl_el);
```

Script of the PIDs' parallel architecture for the Electrical model

```
%% PID CONTROLLER for the alpha/Va transfer function,  using PID tuning tool parameters

kp_alpha_PID = -13061.5181780994;
ki_alpha_PID = -204997.97380483;
kd_alpha_PID = 60.4766490604993;
N_alpha_PID  = 215.976221913899;
Calpha = kp_alpha_PID+(ki_alpha_PID/s)+((kd_alpha_PID*N_alpha_PID*s)/(s+N_alpha_PID));

%% PID CONTROLLER for the total parallel architecture

%% PID CONTROLLER for the total parallel architecture, using Ziegler-Nichols method

% proportional gain stable limit for Ziegler-Nichols optimization
kc = -24000;
% oscillation period [seconds] at P = -24279
Pc = 0.15;
% Ziegler-Nichols method
kp_x_ZN = 0.6*kc;
Tr      = Pc*0.5;
ki_x_ZN = kp_x_ZN/Tr;
Td      = Pc/8;
kd_x_ZN = -kp_x_ZN*Td;
Tf_x_ZN = 0.15*Td;
N_x_ZN  = Tf_x_ZN^-1;

Cx_ZN = kp_x_ZN+(ki_x_ZN/s)+((kd_x_ZN*s)/((Tf_x_ZN*s)+1));

%% PID CONTROLLER for the total parallel architecture, using PID tuning tool parameters

kp_x_PID = -19875.0181340588;
ki_x_PID = -4067.07933875287;
kd_x_PID = 7058.0119971262;
N_x_PID  = 2.81595131067379;

Cx_PID = kp_x_PID+(ki_x_PID/s)+((kd_x_PID*N_x_PID*s)/(s+N_x_PID));

%% Stability Analysis for the alpha/Va tf and the total parallel architecture

%% open   loop transfer function of the alpha/Va tf
FDT_ol_al = minreal(zpk(Calpha*Galpha_Nr));
p_ol_al = pole(FDT_ol_al);
figure(1), nyquist(FDT_ol_al)

%% closed loop transfer function of the alpha/Va tf
FDT_cl_al = minreal(zpk(FDT_ol_al/(1+FDT_ol_al)));
p_cl_al = pole(FDT_cl_al);

%% open   loop transfer function of the total parallel architecture
figure(3), nyquist((Gx_Nr)/(1+(Galpha_Nr*Calpha)))
FDT_ol_tl = minreal(zpk((Gx_Nr*Cx_ZN)/(1+(Galpha_Nr*Calpha))));        % Cx_ZN or Cx_PID
p_ol_tl = pole(FDT_ol_tl);
figure(2), nyquist(FDT_ol_tl)
% magnitude and phase margin of the open loop FDT_ol_tl
figure(4), margin(FDT_ol_tl), grid on

%% closed loop transfer function of the total parallel architecture
FDT_cl_tl = minreal(zpk(FDT_ol_tl/(1+FDT_ol_tl)));
p_cl_tl = pole(FDT_cl_tl);
```

Script of the Static State feedback architecture for the Electrical model

```
%% Controllability condition
Mr     = ctrb(A_r,B_r);       % Controllability matrix
rho_Mr = rank(Mr);            % Rank of controllability matrix

%% Observability   condition
Mo     = obsv(A_r,C_r);       % Observability   matrix
rho_Mo = rank(Mo);            % Rank of observability   matrix

%% Observer design
others_obsv     = -100;
lambda_obsv_des = [others_obsv others_obsv];
C_r_SO = [0 0 1 0];           % second row of system matrix C, referred to x
L      = acker(A_r',C_r_SO',lambda_obsv_des)';

SYS_obsv = ss(A_r-L*C_r_SO,[B_r L],eye(4),zeros(4,2));

x0_hat = [0 0 0 0];           % Observer initial condition for the Simulink Model

%% STATIC-STATE-FEEDBACK: Pole placement, N-K

% continuous time requirements and eigenvalues
s_hat = 0.1;                                                  % max overshoot
zeta  = abs(log(s_hat))/(sqrt(pi^2+(log(s_hat))^2));          % damping coefficient
t_s_1 = 1;                                                    % 1% settling time
wn    = 4.6/(t_s_1*zeta);                                     % natural frequency
sigma = -zeta*wn;                                             % eigenvalue real and
omega = wn*sqrt(1-zeta^2);                                    % imaginary part tau=-1/sigma
tau   = 1/(zeta*wn);                                          % time constant
% discrete time eigenvalues for control law
lambda_1 = sigma+j*omega;
lambda_2 = sigma-j*omega;
lambda_3 = sigma;

% state feedback gain K
others_K1  = -10;
others_K2  = -6;
lambda_des = [others_K1 others_K2 others_K1 0.001*lambda_3];
K = acker(A_r,B_r,lambda_des); % 'acker' for equal values of poles

% gain correction N
N = inv(C_r_SO*inv(eye(4)-(A_r-B_r*K))*B_r);

%% Stability Analysis of Pole placement, N-K
%% open loop transfer function
load('OpenLoopTF_StaticStateFeedback.mat')
SYS_OL_ssf = ss(OpenLoopTF_StaticStateFeedback);
FDT_OL_ssf = minreal(zpk(tf(SYS_OL_ssf)),1e-06);
poles_FDT_OL_ssf = pole(FDT_OL_ssf);
figure(1), nyquist(FDT_OL_ssf)
% magnitude and phase margin of the open loop
figure(2), margin(FDT_OL_ssf), grid on
%% closed loop transfer function
load('ClosedLoopTF_StaticStateFeedback.mat')
SYS_CL_ssf = ss(ClosedLoopTF_StaticStateFeedback);
FDT_CL_ssf = minreal(zpk(tf(SYS_CL_ssf)),1e-06);
poles_FDT_CL_ssf = pole(FDT_CL_ssf);
figure(3), nyquist(FDT_CL_ssf)
% magnitude and phase margin of the open loop
figure(4), margin(FDT_CL_ssf), grid on
```

```
%% STATIC-STATE-FEEDBACK: Pole placement with integral action, Ki-Ko

% State Space representation with the augmented state, n=5
Atot = [1 -C_r_SO; zeros(4,1) A_r];
Btot = [0;B_r];

lambda_IntAct_des  = [-10 -8 -6 -4 0.001*lambda_3];
K_IntAct = place(Atot,Btot,lambda_IntAct_des); % 'place' for different values of poles
Ki = K_IntAct(1);
Ko = K_IntAct(2:5);

%% Stability Analysis of Pole placement with integral action, Ki-Ko
%% open loop transfer function
load('OpenLoopTF_StaticStateFeedback_IntAct.mat')
SYS_OL_ssf_Ia = ss(OpenLoopTF_StaticStateFeedback_IntAct);
FDT_OL_ssf_Ia = minreal(zpk(tf(SYS_OL_ssf_Ia)),1e-06);
poles_FDT_OL_ssf_Ia = pole(FDT_OL_ssf_Ia);
figure(5), nyquist(FDT_OL_ssf_Ia)
% magnitude and phase margin of the open loop
figure(6), margin(FDT_OL_ssf_Ia), grid on
%% closed loop transfer function
load('ClosedLoopTF_StaticStateFeedback_IntAct.mat')
SYS_CL_ssf_Ia = ss(ClosedLoopTF_StaticStateFeedback_IntAct);
FDT_CL_ssf_Ia = minreal(zpk(tf(SYS_CL_ssf_Ia)),1e-06);
poles_FDT_CL_ssf_Ia = pole(FDT_CL_ssf_Ia);
figure(7), nyquist(FDT_CL_ssf_Ia)
% magnitude and phase margin of the open loop
figure(8), margin(FDT_CL_ssf_Ia), grid on

%% STATIC-STATE-FEEDBACK: LQR with integral action, Ki-Ko

% Weighting matrices Q and R
Q = [ 1   0   0   0 ; 0   1   0   0 ; 0   0   1   0 ; 0   0   0   1 ];
R =   1;
[K,S,CLP] = lqr(SYS_r,Q,R);   % lqr command for closed-loop-poles computation
CLP_LQR = [CLP];

% State Space representation with the augmented state, n=5
C_r_SO = [0 0 1 0];
Atot = [1 -C_r_SO; zeros(4,1) A_r];
Btot = [0;B_r];

lambda_LQR_des = [CLP_LQR' 0.001*lambda_3];
K_LQR = place(Atot,Btot,lambda_LQR_des); % 'place' for different values of poles
Ki_LQR = K_LQR(1);
Ko_LQR = K_LQR(2:5);

%% Stability Analysis of LQR with integral action, Ki-Ko (lqr)
%% open loop transfer function
load('OpenLoopTF_LQR.mat')
SYS_OL_LQR = ss(OpenLoopTF_LQR);
FDT_OL_LQR = minreal(zpk(tf(SYS_OL_LQR)),1e-06);
poles_FDT_OL_LQR = pole(FDT_OL_LQR);
figure(1), nyquist(FDT_OL_LQR)
% magnitude and phase margin of the open loop
figure(2), margin(FDT_OL_LQR), grid on
%% closed loop transfer function
load('ClosedLoopTF_LQR.mat')
SYS_CL_LQR = ss(ClosedLoopTF_LQR);
FDT_CL_LQR = minreal(zpk(tf(SYS_CL_LQR)),1e-06);
poles_FDT_CL_LQR = pole(FDT_CL_LQR);
figure(3), nyquist(FDT_CL_LQR)
% magnitude and phase margin of the open loop
figure(4), margin(FDT_CL_LQR), grid on
```

B2 System with pneumatic actuation

Model, state space representation and transfer functions

```
s = tf('s');                              % declaration transfer function 's'

%% Model data (Pneumatic Actuation)

l   = 0.5;                % [m]     Rod length of the pendulum
m   = 0.2;                % [kg]    Concentrated mass of the pendulum
M_c = 1.8;                % [kg]    Cart mass
M_p = 0.4;                % [kg]    Piston mass
b   = 21.8;               % [N*s/m] Viscous damping coefficient
A_a = 2.01*10^-4;         % [m^2]   Area of the rear  piston chamber
A_b = 1.73*10^-4;         % [m^2]   Area of the front piston chamber
C_a = 1200000;            %         A_a*P_supply/(2*Va0)
C_b = 1200000;            %         A_b*P_supply/(2*Vb0)
K_a = 2177000;            %         (p0/(2*rho0))*(k*P_supply/Va0)
K_b = 2534000;            %         (p0/(2*rho0))*(k*P_supply/Vb0)
g   = 9.8;                % [m/s^2] Gravity acceleration

%% Integrators initial condition for the Simulink Model

x0 = [0 0 0 0 0 0];

%% State Space representation

load('sistema_raggiungibile.mat')        % State Space, controllable,   5 states
SYS_r = ss(sistema_raggiungibile);
FDT_r = minreal(zpk(tf(SYS_r)),1e-06);
A_r = SYS_r.a;
B_r = SYS_r.b;
C_r = SYS_r.c;
D_r = SYS_r.d;

load('sistema_NONraggiungibile.mat')     % State Space, uncontrollable, 6 states
SYS_Nr = ss(sistema_NONraggiungibile);
FDT_Nr = minreal(zpk(tf(SYS_Nr)),1e-06);
A_Nr = SYS_Nr.a;
B_Nr = SYS_Nr.b;
C_Nr = SYS_Nr.c;
D_Nr = SYS_Nr.d;

%% Controllable   SYS transfer functions

Galpha_r = zpk(FDT_r(1));                                % alpha/u transfer function
Gx_r     = zpk(FDT_r(2));                                % x/u     transfer function

%% Uncontrollable SYS transfer functions

Galpha_Nr = zpk(FDT_Nr(1));                              % alpha/u transfer function
Gx_Nr     = zpk(FDT_Nr(2));                              % x/u     transfer function
Galpha_x_Nr = minreal(zpk(Gx_Nr/Galpha_Nr),1e-06);       % x/alpha transfer function
```

Script of the PIDs' cascade architecture for the Pneumatic model

```
%% PID for the INNER LOOP of the PIDs' cascade architecture

%% PID CONTROLLER for the inner loop, using PID tuning tool parameters
kp_alpha = 2.09;
ki_alpha = 6.31;
kd_alpha = 0.103;
Tf_alpha = 0.000875;
N_alpha  = Tf_alpha^-1;

Calpha = kp_alpha+(ki_alpha/s)+((kd_alpha*s)/((Tf_alpha*s)+1));

%% PID for the OUTER LOOP of the PIDs' cascade architecture

%% PID CONTROLLER for the outer loop, using PID tuning tool parameters
kp_x = -0.000802;
ki_x = -4.68e-06;
kd_x = -0.0306;
Tf_x = 0.00875;
N_x  = Tf_x^-1;

Cx = kp_x+(ki_x/s)+((kd_x*s)/((Tf_x*s)+1));

%% Stability Analysis of inner and outer loops, PIDs' cascade architecture

%% open loop transfer function of the inner loop
FDT_ol_il = minreal(zpk(Calpha*Galpha_Nr),1e-06);
p_ol_il = pole(FDT_ol_il);
figure(1), nyquist(FDT_ol_il)
% magnitude and phase margin of the open loop FDT_ol_il
figure(2), margin(Calpha*Galpha_Nr), grid on

%% closed loop transfer function of the inner loop
FDT_cl_il = minreal(zpk(FDT_ol_il/(1+FDT_ol_il)),1e-06);
p_cl_il = pole(FDT_cl_il);

%% open loop transfer function of the external loop
figure(8), nyquist(FDT_cl_il*Galpha_x_Nr)
FDT_ol_el = minreal(zpk(Cx*FDT_cl_il*Galpha_x_Nr),1e-06);
p_ol_el = pole(FDT_ol_el);
figure(3), nyquist(FDT_ol_el)
% magnitude and phase margin of the open loop FDT_ol_el
figure(4), margin(Cx*FDT_cl_il*Galpha_x_Nr), grid on

%% closed loop transfer function of the external loop
FDT_cl_el = minreal(zpk(FDT_ol_el/(1+FDT_ol_el)),1e-06);
p_cl_el = pole(FDT_cl_el);
```

Script of the PIDs' parallel architecture for the Pneumatic model

```
%% PID CONTROLLER for the alpha/Va transfer function,  using PID tuning tool parameters

kp_alpha = 2.09;
ki_alpha = 6.31;
kd_alpha = 0.103;
Tf_alpha = 0.000875;
N_alpha  = Tf_alpha^-1;

Calpha = kp_alpha+(ki_alpha/s)+((kd_alpha*s)/((Tf_alpha*s)+1));

%% PID CONTROLLER for the total parallel architecture

%% PID CONTROLLER for the total parallel architecture, using Ziegler-Nichols method

% proportional gain stable limit for Ziegler-Nichols optimization
kc = -0.6;
% oscillation period [seconds] at P = -0.6
Pc = 3.15;     % 5 for optimization
% Ziegler-Nichols method
kp_x_ZN = 0.6*kc;
Tr      = Pc*0.5;
ki_x_ZN = kp_x_ZN/Tr;
Td      = Pc/8;
kd_x_ZN = -kp_x_ZN*Td;
Tf_x_ZN = 0.15*Td;
N_x_ZN  = Tf_x_ZN^-1;

Cx = kp_x_ZN+(ki_x_ZN/s)+((kd_x_ZN*s)/((Tf_x_ZN*s)+1));

%% Stability Analysis for the alpha/Va tf and the total parallel architecture

%% open   loop transfer function of the alpha/Va tf
FDT_ol_al = minreal(zpk(Calpha*Galpha_Nr));
p_ol_al = pole(FDT_ol_al);
figure(1), nyquist(FDT_ol_al)

%% closed loop transfer function of the alpha/Va tf
FDT_cl_al = minreal(zpk(FDT_ol_al/(1+FDT_ol_al)));
p_cl_al = pole(FDT_cl_al);

%% open   loop transfer function of the total parallel architecture
figure(3), nyquist((Gx_Nr)/(1+(Galpha_Nr*Calpha)))
FDT_ol_tl = minreal(zpk((Gx_Nr*Cx_ZN)/(1+(Galpha_Nr*Calpha))));
p_ol_tl = pole(FDT_ol_tl);
figure(2), nyquist(FDT_ol_tl)
% magnitude and phase margin of the open loop FDT_ol_tl
figure(4), margin(FDT_ol_tl), grid on

%% closed loop transfer function of the total parallel architecture
FDT_cl_tl = minreal(zpk(FDT_ol_tl/(1+FDT_ol_tl)));
p_cl_tl = pole(FDT_cl_tl);
```

Script of the Static State feedback architecture for the Pneumatic model

```
%% Controllability condition
Mr     = ctrb(A_r,B_r);       % Controllability matrix
rho_Mr = rank(Mr);            % Rank of controllability matrix

%% Observability   condition
Mo     = obsv(A_r,C_r);       % Observability   matrix
rho_Mo = rank(Mo);            % Rank of observability   matrix

%% Observer design
others_obsv     = -100;
lambda_obsv_des = [others_obsv others_obsv];
C_r_SO = [0 0 0 1 0];         % second row of system matrix C, referred to x
L      = acker(A_r',C_r_SO',lambda_obsv_des)';

SYS_obsv = ss(A_r-L*C_r_SO,[B_r L],eye(5),zeros(5,2));

x0_hat = [0 0 0 0 0];         % Observer initial condition for the Simulink Model

%% STATIC-STATE-FEEDBACK: Pole placement, N-K

% continuous time requirements and eigenvalues
s_hat = 0.1;                                                  % max overshoot
zeta  = abs(log(s_hat))/(sqrt(pi^2+(log(s_hat))^2));          % damping coefficient
t_s_1 = 1;                                                    % 1% settling time
wn    = 4.6/(t_s_1*zeta);                                     % natural frequency
sigma = -zeta*wn;                                             % eigenvalue real and
omega = wn*sqrt(1-zeta^2);                                    % imaginary part tau=-1/sigma
tau   = 1/(zeta*wn);                                          % time constant
% discrete time eigenvalues for control law
lambda_1 = sigma+j*omega;
lambda_2 = sigma-j*omega;
lambda_3 = sigma;

% state feedback gain K
others_K   = -5;
lambda_des = [others_K lambda_3];
K = acker(A_r,B_r,lambda_des); % 'acker' for equal values of poles

% gain correction N
N = inv(C_r_SO*inv(eye(5)-(A_r-B_r*K))*B_r);

%% Stability Analysis of Pole placement, N-K
%% open loop transfer function
load('OpenLoopTF_StaticStateFeedback.mat')
SYS_OL_ssf = ss(OpenLoopTF_StaticStateFeedback);
FDT_OL_ssf = minreal(zpk(tf(SYS_OL_ssf)),1e-06);
poles_FDT_OL_ssf = pole(FDT_OL_ssf);
figure(1), nyquist(FDT_OL_ssf)
% magnitude and phase margin of the open loop
figure(2), margin(FDT_OL_ssf), grid on
%% closed loop transfer function
load('ClosedLoopTF_StaticStateFeedback.mat')
SYS_CL_ssf = ss(ClosedLoopTF_StaticStateFeedback);
FDT_CL_ssf = minreal(zpk(tf(SYS_CL_ssf)),1e-06);
poles_FDT_CL_ssf = pole(FDT_CL_ssf);
figure(3), nyquist(FDT_CL_ssf)
% magnitude and phase margin of the open loop
figure(4), margin(FDT_CL_ssf), grid on
```

```
%% STATIC-STATE-FEEDBACK: Pole placement with integral action, Ki-Ko

% State Space representation with the augmented state, n=5
Atot = [1 -C_r_SO; zeros(5,1) A_r];
Btot = [0;B_r];

lambda_IntAct_des  = [-10 -10 -10 -10 -10 lambda_3];
K_IntAct = acker(Atot,Btot,lambda_IntAct_des); % 'acker' for equal values of poles
Ki = K_IntAct(1);
Ko = K_IntAct(2:6);

%% Stability Analysis of Pole placement with integral action, Ki-Ko
%% open loop transfer function
load('OpenLoopTF_StaticStateFeedback_IntAct.mat')
SYS_OL_ssf_Ia = ss(OpenLoopTF_StaticStateFeedback_IntAct);
FDT_OL_ssf_Ia = minreal(zpk(tf(SYS_OL_ssf_Ia)),1e-06);
poles_FDT_OL_ssf_Ia = pole(FDT_OL_ssf_Ia);
figure(5), nyquist(FDT_OL_ssf_Ia)
% magnitude and phase margin of the open loop
figure(6), margin(FDT_OL_ssf_Ia), grid on
%% closed loop transfer function
load('ClosedLoopTF_StaticStateFeedback_IntAct.mat')
SYS_CL_ssf_Ia = ss(ClosedLoopTF_StaticStateFeedback_IntAct);
FDT_CL_ssf_Ia = minreal(zpk(tf(SYS_CL_ssf_Ia)),1e-06);
poles_FDT_CL_ssf_Ia = pole(FDT_CL_ssf_Ia);
figure(7), nyquist(FDT_CL_ssf_Ia)
% magnitude and phase margin of the open loop
figure(8), margin(FDT_CL_ssf_Ia), grid on

%% STATIC-STATE-FEEDBACK: LQR with integral action, Ki-Ko

% Weighting matrices Q and R
Q = [ 10 0 0 0 0 ; 0 1 0 0 0 ; 0 0 1 0 0 ; 0 0 0 100 0 ; 0 0 0 0 1 ];
R =   1e9;
[K,S,CLP] = lqr(SYS_r,Q,R);   % lqr command for closed-loop-poles computation
CLP_LQR = [CLP];

% State Space representation with the augmented state, n=5
C_r_SO = [0 0 1 0 0];
Atot = [1 -C_r_SO; zeros(5,1) A_r];
Btot = [0;B_r];

lambda_LQR_des = [CLP_LQR' lambda_3*1e01];
K_LQR = place(Atot,Btot,lambda_LQR_des); % 'place' for different values of poles
Ki_LQR = K_LQR(1);
Ko_LQR = K_LQR(2:6);

%% Stability Analysis of LQR with integral action, Ki-Ko (lqr)
%% open loop transfer function
load('OpenLoopTF_LQR.mat')
SYS_OL_LQR = ss(OpenLoopTF_LQR);
FDT_OL_LQR = minreal(zpk(tf(SYS_OL_LQR)),1e-06);
poles_FDT_OL_LQR = pole(FDT_OL_LQR);
figure(1), nyquist(FDT_OL_LQR)
% magnitude and phase margin of the open loop
figure(2), margin(FDT_OL_LQR), grid on
%% closed loop transfer function
load('ClosedLoopTF_LQR.mat')
SYS_CL_LQR = ss(ClosedLoopTF_LQR);
FDT_CL_LQR = minreal(zpk(tf(SYS_CL_LQR)),1e-06);
poles_FDT_CL_LQR = pole(FDT_CL_LQR);
figure(3), nyquist(FDT_CL_LQR)
% magnitude and phase margin of the open loop
figure(4), margin(FDT_CL_LQR), grid on
```

Appendix C

C Computation of $G_{V_a\alpha}$, G_{V_ax} and $G_{\alpha x}$ transfer functions

This Appendix provides the computation steps used to obtain $G_{V_a\alpha}$, G_{V_ax} and $G_{\alpha x}$, starting from the equations (2.3-1), 2), 3), 4), 5) and 6) shown in section 2.3, which are repeated here

$$l_p\ddot{\alpha} = g\alpha - \ddot{x} \quad (1)$$

$$\dot{x} = \frac{p}{2\pi\cdot 1000}\dot{\theta} \quad (2)$$

$$V_a = R_a I_a + E \quad (3)$$

$$E = K\dot{\theta} \quad (4)$$

$$C_m = KI_a \quad (5)$$

$$C_m - J_{tot}\ddot{\theta} = \left(m_p l_p \frac{p}{2\pi\cdot 1000\cdot\eta}\right)\ddot{\alpha} \quad (6)$$

Summarizing, the equation (1) describes the linearized model of the inverted pendulum, the equation (2) gives the ratio of the transmission system, the equations (3), (4) and (5) are the constitutive equations that describes the motor and, finally, the equation (6) is the torques balance at the output shaft of the motor.

Applying the Laplace transform to these equations, the new relations become

$$l_p\bar{\alpha}s^2 = g\bar{\alpha} - \bar{x}s^2 \quad (s1)$$

$$\bar{x} = \frac{p}{2\pi\cdot 1000}\bar{\theta} \quad (s2)$$

$$\overline{V_a} = R_a\overline{I_a} + \overline{E} \quad (s3)$$

$$\overline{E} = K\bar{\theta}s \quad (s4)$$

$$\overline{C_m} = K\overline{I_a} \quad (s5)$$

$$\overline{C_m} - J_{tot}\bar{\theta}s^2 = \left(m_p l_p \frac{p}{2\pi\cdot 1000\cdot\eta}\right)\bar{\alpha}s^2 \quad (s6)$$

Let's assume, from now on, the marked variables equal to the ones that are time-dependent, in order to simplify the next writing procedures.

Considering the equation (s1) and isolating the x and α terms, the transfer function obtained is

$$G_{\alpha x} = \frac{x}{\alpha} = \frac{g - l_p s^2}{s^2}$$

Starting from equation (s3) and substituting, for I_a and E, the equations (s4) and (s5) the follow relation is obtained

$$V_a = R_a \frac{C_m}{K} + K\theta s \qquad \text{(A)}$$

Others substitution can be applied to the relation (A), the following steps describe the procedure to obtain the $G_{V_a\alpha}$ transfer function, highlighting the needed substitutions

(s2) - (s6) → (A) $$V_a = \frac{R_a}{K}\left(\left(m_p l_p \frac{p}{2\pi\cdot 1000\cdot\eta}\right)\alpha s^2 + J_{tot}\theta s^2\right) + \frac{K2\pi 1000}{p} x s \qquad \text{(B)}$$

(s1) - (s2) → (B) $$V_a = \frac{R_a}{K}\left(\left(m_p l_p \frac{p}{2\pi\cdot 1000\cdot\eta}\right)\alpha s^2 + \frac{J_{tot}K2\pi 1000}{p} x s^2\right) + \frac{K2\pi 1000}{p}\left(\frac{g - l_p s^2}{s^2}\right)\alpha s \qquad \text{(C)}$$

(s1) → (C) $$V_a = \frac{R_a}{K}\left(\left(m_p l_p \frac{p}{2\pi\cdot 1000\cdot\eta}\right)\alpha s^2 + \frac{J_{tot}K2\pi 1000}{p}\left(\frac{g - l_p s^2}{s^2}\right)\alpha s^2\right) + \frac{K2\pi 1000}{p}\left(\frac{g - l_p s^2}{s^2}\right)\alpha s \qquad \text{(D)}$$

Simplifying the relation (D) and highlighting α in the right side, the $G_{V_a\alpha}$ can be obtained. The relation becomes

$$G_{V_a\alpha} = \frac{(2\pi 1000 K p^2 \eta)s}{(m_p p^3 l_p R_a - J_{tot}(2\pi 1000)^2 \eta l_p p R_a)s^3 - ((2\pi 1000)^2 \eta l_p p K^2)s^2 + (J_{tot}(2\pi 1000)^2 \eta g p R_a)s + ((2\pi 1000)^2 g K^2 p\eta)}$$

Knowing $G_{V_a\alpha}$ and $G_{\alpha x}$, the $G_{V_a x}$ transfer function is computed as follows

$$G_{V_a x} = G_{V_a\alpha} G_{\alpha x} = \frac{(2\pi 1000 K p^2 \eta)s}{(m_p p^3 l_p R_a - J_{tot}(2\pi 1000)^2 \eta l_p p R_a)s^3 - ((2\pi 1000)^2 \eta l_p p K^2)s^2 + (J_{tot}(2\pi 1000)^2 \eta g p R_a)s + ((2\pi 1000)^2 g K^2 p\eta)} \cdot \frac{g - l_p s^2}{s^2}$$

Bibliography

[1] G. Calafiore "*Appunti di Controlli Automatici"*, Edizioni CLUT, Torino (2006)

[2] G. Calafiore "*Elementi di Automatica",* Edizioni CLUT, Torino (2004)

[3] C. Greco, M. Rulla, L. Spagnolo "*Laboratorio sperimentale di automatica. Applicazioni di modellistica, analisi e controllo",* McGraw-Hill Education Companies (2003)

[4] M. Pontin "*Modellazione, realizzazione e controllo mediante PLC di un sistema a pendolo inverso ad attuazione pneumatica*", Politecnico di Torino (2018)

[5] Shailesh S Hedge "*Cart-inverted pendulum with pneumatic actuation: simulation and experiments*", Politecnico di Torino (2018)

[6] S. Di Natale "*Model-based design of a fuzzy logic controller for an inverse pendulum*", Politecnico di Torino (2019)

[7] Nivedita Rajak "*Stabilization of Cart-Inverted Pendulum using Pole-Placement Method"* International Journal of Advanced Engineering Research and Studies (2015)

[8] Wende Li, Hui Ding & Kai Cheng "*An Investigation on the Design and Performance Assessment of double-PID and LQR Controllers for the Inverted Pendulum",* UKACC International Conference on Control, Cardiff, UK (2012)

[9] Jia-Jun Wang "*Simulation studies of inverted pendulum based on PID controllers"* Institute of Automation, Hangzhou Dianzi University, Hangzhou, 310018 Zhejiang, PR China (2010)

[10] T Rakesh Krishnan "*On stabilization of Cart-Inverted Pendulum System: An Experimental Study",* National Institute of Technology, Rourkela-769008, India (2012)