

POLITECNICO DI TORINO

Master's Degree in Computer Engineering



Master's Degree Thesis

**Design and Testing of a Robust Control
System For Industrial Manipulators
using Matlab and Simulink**

A Robust H_∞ Approach through Nested Loops

Supervisor

Candidate

Prof. Marina INDRI

Martina BELLISSIMO

Supervisor
at Comau S.p.A
Ing. Stefano PESCE

April 2021

Alla mia Fabiola

*"If you want it, you can fly,
you just have to trust you a lot"*
Steve Jobs

Abstract

The Thesis project has been developed in COMAU S.p.A, one of the most important leading companies in the industrial automation field.

The purpose of the activity is the development and simulation of a robust control system for an industrial six axes manipulator, the most common mechanical structure. This is a very important goal to reach, since robustness to disturbances and uncertainties has always been the central focus in feedback control theory.

To fulfill this objective, the H_∞ approach has been used. Moreover, the development of the controller has given the chance to investigate the advantages and disadvantages of this method when it is applied to a more realistic and complex field, such as the robotics one. Matlab has been adopted as computing environment for the development of the design. The analysis of the obtained results has been carried out through Simulink toolbox.

After the derivation of the model describing the dynamics of the robot under analysis, a simple feedback control system has been implemented to control the position of a single axis of the manipulator. To enhance the robustness of the system stability, the previous structure has been subsequently improved by introducing an inner loop aimed at controlling the velocity. Once the controllers have been derived, their robustness with respect to variations, in a given range of uncertainty, of the parameters characterizing the system has been studied. Finally, a parametric controller adapting to the changes of the link inertia has been built.

Table of Contents

List of Tables	IV
List of Figures	V
I Introduction	1
1 A Background on Robotics and Control	3
1.1 The birth of Robotics	3
1.2 Robots Classification	4
1.3 The Structure of a Robot	5
1.4 What is a Control System?	5
1.5 System Models	6
1.5.1 Mass-Spring-Damper model	7
1.6 Control of Mechanical Manipulators	8
1.7 H_∞ Control Approach	9
1.7.1 Specifications Handling	11
1.7.2 The Weighting Functions	11
1.7.3 LMI Approach	16
1.7.4 Advantages and Disadvantages of the H_∞ Control System	17
II A Benchmark Problem for the Control of an Industrial Manipulator	18
2 Feedback Control System	20
2.1 The relevance of a Benchmark Problem	20
2.2 The industrial Manipulator	21
2.3 Dynamic Model Derivation	21
2.3.1 Model Reduction	23
2.4 Plant Definition	24

2.5	H_∞ Control Implementation	27
2.5.1	Weighting Functions Definition	28
2.5.2	The Controller	33
2.5.3	Time Simulation	33
3	Nested Control System	35
3.1	Requirements to Guarantee	35
3.2	Velocity Control	36
3.2.1	Plant Definition	36
3.2.2	Velocity Specifications Handling	37
3.2.3	Weighting Functions Definition	38
3.2.4	G_{c_v} Controller	38
3.2.5	Time Simulation	40
3.3	Position Control	41
3.3.1	Plant Definition	41
3.3.2	Position Specifications Handling	42
3.3.3	Weighting Functions Definition	42
3.3.4	G_{c_p} Controller	42
3.3.5	Time Simulation	43
III	Robust Control of a Comau Industrial Manipulator	45
4	Motor and Link Control	47
4.1	Control Requirements	48
4.2	Velocity Control	49
4.2.1	Dynamical Model Derivation	49
4.3	Plant Definition	50
4.4	Velocity Specifications Handling	50
4.5	Weighting Functions Definition	52
4.6	G_{c_v} Controller	54
4.7	Time Simulation	56
4.8	Position Control	57
4.9	Plant Definition	57
4.10	Position Specifications Handling	57
4.11	Weighting Functions Definition	58
4.12	G_{c_p} Controller	59
4.13	Time Simulation	61
4.14	Link Control	62
4.14.1	Plant Definition	62
4.14.2	Time Simulation	63

5	Parameters Variation	65
5.1	Single Parameters Variation	65
5.2	A Robust Choice of Gc_v and Gc_p	68
5.2.1	Single Parameters Variation	68
5.2.2	Multi-Parameters Variation	72
6	Changing Controller	74
6.1	Velocity Controllers	74
6.1.1	Velocity Design Optimization	78
6.2	Position Controllers	80
6.2.1	Position Time Responses	81
6.2.2	Position Design Optimization	82
6.2.3	Final Time Responses	84
7	Conclusions and Future Works	85
	Bibliography	90

List of Tables

2.1	Parameter values	23
5.1	Results from parameters variation	66
5.2	Results from the I Solution	69
5.3	Results from the II Solution	70

List of Figures

1.1	Rossum's Universal Robots [1]	3
1.2	General Control System	6
1.3	Mass-Spring-Dumper System [3]	7
1.4	Solution Procedure through Laplace and Anti-Laplace Transform	8
1.5	Feedback Control System	9
1.6	Standard Feedback Configuration [4]	11
1.7	$W_S(s)$ Weighting Function	13
1.8	$W_T(s)$ Weighting Function	14
2.1	Feedback Control System	20
2.2	Industrial Manipulator [7]	21
2.3	Four-masses Model [7]	21
2.4	Spring torque as a nonlinear function of $(q_m - q_1)$ [7]	22
2.5	Bode Plot of the $G(s)$ transfer function	25
2.6	Bode Plot of the Notch Filter $G_N(s)$	26
2.7	Bode Plot of $G(s)$ and $G_p(s)$	27
2.8	Weighting Function on $S(s)$	29
2.9	Weighting Function on $T(s)$	30
2.10	Step Time Simulation	31
2.11	$W_S(s)$ and $W_1(s)$	32
2.12	$W_T(s)$ and $W_2(s)$	32
2.13	Nichols Plot of the Open-Loop function corresponding to $G_c(s)$ (2.36)	34
2.14	Time Response	34
3.1	Nested Control System	35
3.2	Bode Plot of $G(s)$	36
3.3	Bode Plot of $G(s)$ and $G_p(s)$	37
3.4	Nichols of the Loop Function $L(s)$	39
3.5	Increase of the crossover frequency on the Nichols Plot	40
3.6	Time Response	40
3.7	Bode Plot of $G_p(s)$	41

3.8	Time Response	43
3.9	Nichols Plot of the Loop Functions derived from $G_c(s)$ and $G_{c_p}(s)$	44
4.1	A Comau Robot	47
4.2	Motor Velocity Plant	48
4.3	Motor - Link Plant	48
4.4	Motor - Link Control System	48
4.5	Simulink Scheme for the model of the robot	50
4.6	Bode Plot of the Plant	51
4.7	Weighting Function on the Sensitivity	52
4.8	Weighting Function on the Complementary Sensitivity	53
4.9	Loop functions derived from the original controller and the modified controller	55
4.10	Loop functions derived from $G_c(s)$ and $G_{c_v}(s)$	55
4.11	Loop functions derived from $G_c(s)$ and $G_{c_v}(s)$	56
4.12	Motor-Link Control System	57
4.13	Bode Plot of the Plant	58
4.14	Loop functions derived from $G_c(s)$	59
4.15	Loop functions derived from $G_c(s)$ and $G_{c_p}(s)$	60
4.16	Time Response	61
4.17	Bode Plot of $G_{pl}(s)$	62
4.18	Simulink Model for the derivation of the link transfer function	63
4.19	Simulink Model for the description of the motor control and the link control	63
4.20	Time Link Performances	64
5.1	Nichols Plot	66
5.2	Time Response	67
5.3	Nichols plot of the Open Loop functions computed for the variation K_r parameter (I Solution)	71
5.4	Nichols plot of the Open Loop functions computed for the variation K_r parameter (II Solution)	71
6.1	Nichols plot of the Open Loop functions computed for the input values J_l	77
6.2	Time responses	77
6.3	Bode Plots of the Original Plant	78
6.4	Nichols of $L(s)$ functions	79
6.5	Time Responses	79
6.6	Bode Plot of the Plants	80
6.7	Nichols Plant	81
6.8	Position Time Responses	82

6.9	Bode Plot of $G_{p,orig}(1)$ and $G_{p,app}(1)$	83
6.10	Nichols Plot of the Open Loop functions	83
6.11	Step Responses	84
7.1	Comau Robots	85

Part I

Introduction

Chapter 1

A Background on Robotics and Control

1.1 The birth of Robotics

In 1920, a Czechoslovakian playwright, Karl Kapek, published R.U.R (Rossum's Universal Robots), a drama where a scientist named Rossum discovers the secret of creating a human-like machine called *robot*. This is a term deriving from *robota*, a Slav word that means executive labour.

In his work, Kapek conceives robots as figures that replace and exterminate the human race.



Figure 1.1: Rossum's Universal Robots [1]

The first person who defined *robotics* from a scientific point of view was Isaac Asimov. In this context, robotics was born as the science that aims at the study of robots. Asimov reversed the Kapek idea and defined a robot not as a machine that can harm humans, but that can help them in tasks that might be dangerous or impossible to perform. He underlined this concept in the "Three Laws of Robotics" (1942).

The First Law states that "A robot may not injure a human being or, through inaction, allow a human being to come to harm".

The Second Law states that "A robot must obey the orders given it by human beings except where such orders would conflict with the First Law".

The Third Law states that "A robot must protect its own existence as long as such protection does not conflict with the First or Second Law".

These laws have made an important impact on the research and development of the actual robots, since they can be considered as the basic rules for their design.

1.2 Robots Classification

In the years, industrial manipulators have been built to replace people in dangerous tasks, to increase productivity and to improve quality. But, despite what people think, not always a robot can be seen as a reproduction of the human body. With the evolution of the concept of robotics, many types of robots, used in different fields and for different purposes, have been produced:

Industrial Robots: they usually consist of a manipulator arm used in the industrial environment for lifting heavy components, moving parts from one place to another one or performing some repetitive actions.

Humanoids: they look like humans and they are able to interact with the environment.

Surgical Robots: they are used for performing surgeries with the remote control of doctors.

Service Robots: they substitute humans in everyday tasks giving them the change to handle with more cognitive actions.

Army Robots: they are used in military fields.

At the same time, from a structural point of view, it is possible to point out three classes of robots:

Fixed Robots: they are formed by a fixed base that prevents them from moving during their operations. They generally manipulate the environment by controlling the position and orientation of an end-effector.

Wheeled Robots: they are able to change their position thanks to the presence of the wheels.

Legged Robots: they are very similar to the wheeled ones, but they move their structure thanks to mechanical legs.

Today, the most popular class is the one of the fixed robots. They can perform precise mechanical and physical tasks and, for this reason, spread through many areas of the modern industrial automation.

1.3 The Structure of a Robot

Robotics is commonly defined as the science studying the "*intelligent connection between perception and action*" [2].

According to this statement, it is possible to consider a robot as a complex system formed by:

1. A mechanical structure that involves a locomotion apparatus (wheels, mechanical legs) and a manipulator apparatus (mechanical arms, end-effectors, artificial hands)
2. Actuators that allow the movements of the mechanical structure
3. Sensors that guarantee the interaction with the surrounding environment
4. A Control System that controls the actions the robot needs to make in order to complete its jobs.

1.4 What is a Control System?

It is possible to speak of *control system* every time it is needed to impose a *desired behavior* to a *physical system*, or every time it is needed to impose a *desired evolution* to some *variables* associated with a *physical system*.

A *control system* consists of:

1. a real process, called *plant*, that represents the system that has to be controlled, according to some predetermined objectives
2. sensors and transducers to measure the variables that have to be controlled
3. actuators to act on the variables to control
4. algorithms that realize a *control law* that gives orders to the actuators about what they have to do

A simple representation of a control system and its basic components is depicted in Figure 1.2

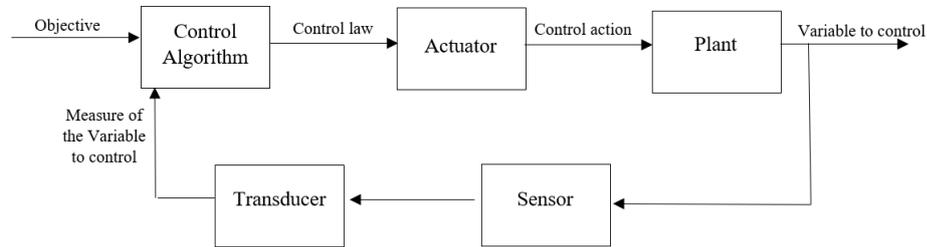


Figure 1.2: General Control System

In other words, one of its purposes is to make the output y (that represents the variable to control) behave in a desired way by manipulating the plant input u (that is defined by the control law), also considering the effects produced by some disturbances.

However, the process of designing a control system is not easy and usually requires many steps:

1. Study the system that has to be controlled and decide the kinds of actuators and sensors to adopt
2. Model the system to be controlled and simplify it as much as possible
3. Analyze the properties of the resulted model
4. Set the performance specifications
5. Define the type of controller to implement
6. Design a controller able to meet the control objectives and specifications
7. Simulate the controlled system
8. Tune the controller if necessary

1.5 System Models

Modeling of systems is an essential task to accomplish.

A *model* is a mathematical description of the connection between the *control variables* and *variables to control*. It may be derived from physical laws or experimental data, it allows to analyze and to study a system, but not in a perfect way. There will always be some uncertainties to take into account.

1.5.1 Mass-Spring-Damper model

Any mechanical system can be described, in a very compact form, by means of the *mass-spring-damper model* (Figure 1.3), since its basic elements are precisely the mass, the spring and the damper.

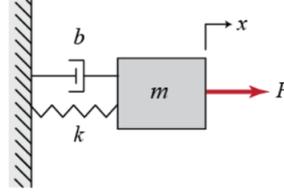


Figure 1.3: Mass-Spring-Dumper System [3]

m is a rigid body that slides on the x direction under the action of a force F . The rigid body is connected to a vertical plan through a *spring* and a *damper*. The spring force k is proportional to the displacement of the mass, x , while the damping force is proportional to the velocity of the mass \dot{x} . According to the Newton's 2nd law:

$$\sum F_x = ma_x \quad (1.1)$$

F_x is given by the sum of all the forces acting on the system

$$F_x = F + kx + b\dot{x} \quad (1.2)$$

The initial equation (1.1) becomes

$$F - kx - b\dot{x} = m\ddot{x} \quad (1.3)$$

that is

$$m\ddot{x} + b\dot{x} + kx = F \quad (1.4)$$

The second order differential equation (1.4) can be difficult to solve by means of the canonical resolution methods. To reach this objective in a fast and rigorous way, it is better to use another instrument: the *Laplace Transform* \mathcal{L} . The *Laplace Transform* is a mathematical tool also used to solve a linear differential equation in a simple way.

$$\mathcal{L}(f(t)) = F(s) = \int_0^{\infty} f(t)e^{-st} dt = \lim_{\tau \rightarrow \infty} \int_0^{\tau} f(t)e^{-st} dt \quad (1.5)$$

Applying the *Laplace Transform* to (1.4), where F is the input and x is the output

$$Ms^2X(s) + BsX(s) + KX(s) = FU(s) \quad (1.6)$$

$$X(s) = \frac{1}{Ms^2 + Bs + K}FU(s) \quad (1.7)$$

Starting from the computed solution in the Laplace domain $X(s)$ and applying the *Anti-Laplace Transform* \mathcal{L}^{-1} , the solution $x(t)$ can be obtained.

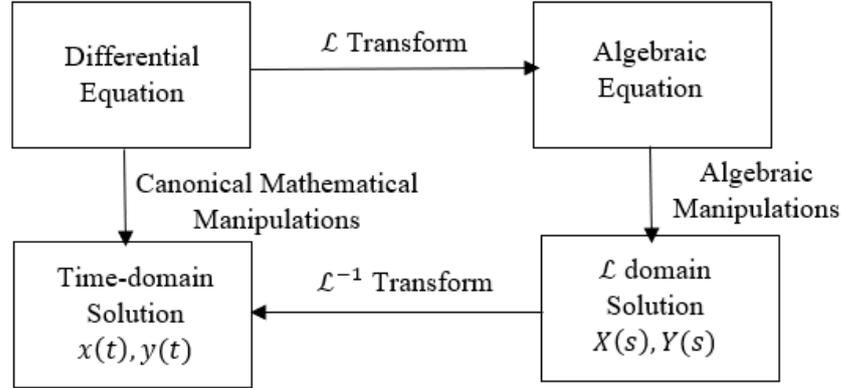


Figure 1.4: Solution Procedure through Laplace and Anti-Laplace Transform

1.6 Control of Mechanical Manipulators

A very important task is to control the movement of a robot manipulator. For doing this, it is possible to implement a variety of control methods and controllers, each one with its own advantages and disadvantages. Since the dynamics of robots are highly nonlinear and may contain uncertain parameters, linear controls (PD control, PID control, Inverse Dynamics control, Lyapunov-based control) are not usually suitable. In fact, they do not guarantee performance and robustness over the full operating range. Therefore, nonlinear controllers with robust performance which can cope with uncertainties and variation in the parameters are recommended. Among the most frequently used control methods, the thesis work has been focused on the H_∞ one.

1.7 H_∞ Control Approach

The H_∞ control approach is a very powerful instrument because it guarantees robustness in the presence of disturbances and uncertainties.

The task of the algorithm is to find a controller G_c that assures good performances in terms of steady-state output errors, overshoot, rise time and settling time.

The design procedure is based on the involvement of loop shaping techniques of SISO control systems. A first analysis of the results is carried out through the adoption of the Nichols plane, on which the constant magnitude loci M_T of the complementary sensitivity function $T(s)$ and the constant magnitude loci M_S of the sensitivity function $S(s)$ are represented. This solution helps to investigate on the relationship between the closed loop $T(s)$ and $S(s)$ and the open loop $L(s)$ frequency responses.

To start with the implementation phase, it is indispensable the presence of a linear model of the system. Otherwise, a linearization around a suitable operating point has to be performed.

Consider the class of controllers G_c^{stab} able at stabilizing the considered system, G_c is chosen as the one that minimizes the H_∞ norm of the closed loop transfer function. However, the derivation of the controller is not a simple task because it involves the definition of some weighting functions, chosen to reflect the design objectives, the disturbances and sensor noises. They are $W_S(s)$, $W_T(s)$ and $W_U(s)$, respectively. The first ones are used to define the frequency requirements, while the latter one is used to sum up all the possible uncertainties afflicting the model. Since the former part of the work is based on the hypothesis that the plant is not affected by uncertainties in the parameters, only $W_S(s)$ and $W_T(s)$ are exploited.

The general structure of the control is illustrated in Figure 1.5

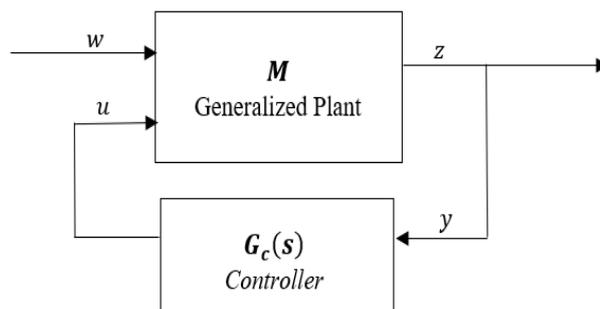


Figure 1.5: Feedback Control System

M is the *generalized plant*. It involves everything that is known at the beginning of the control design, such as the plant, the actuators, the sensors, and it is built taking into account all the requirements of the control problem. $G_c(s)$ is the controller: its derivation represents the purpose of the design. The inputs of the generalized plant, w , represent references, disturbances, sensor noises, while its outputs, z , represent all the signals that have to be controlled and the plant outputs. The vector y contains the sensor output and u contains the controller inputs to the generalized plant. Both the plant and the controller are assumed to be real rational and proper functions. Furthermore, the state-space models of M and $G_c(s)$ are assumed to be available and their realizations are assumed to be stabilizable and detectable. [4]

Optimal and Sub-Optimal H_∞ Control

The purpose of the **Optimal** H_∞ Control is:

"to find all admissible controllers $G_c(s)$ such that $\|T_{wz}\|_\infty$ is minimized", where T_{wz} is a transfer function to be properly selected [4].

The controller $G_c(s)$ is computed by solving an optimization problem:

$$G_c(s) = \arg \min_{G_c \in G_c^{stab}} \|T_{wz}\|_\infty \quad (1.8)$$

G_c^{stab} denotes the set of all the controllers that make the system internally stable. From this set, G_c is selected. It produces a control signal u that counteracts the influence of w and z , minimizing the norm $\|T_{wz}\|_\infty$. [5]

The problem is that deriving an optimal H_∞ controller is complicated and sometimes it is not necessary. In practice it is preferred to compute controllers that are very close, in the norm sense, to the optimal one. They are known as *sub-optimal controllers*.

The purpose of the **Sub-optimal** H_∞ Control is:

to find all admissible controllers $G_c(s)$, if there are any, such that $\|T_{wz}\|_\infty < \gamma$, with $\gamma > 0$ ", where γ represents the best H_∞ performance [4].

For the latter reasons, the attention will be focused on suboptimal control.

Derivation of the controller

The controller is derived by following three steps:

1. Selection of the T_{wz} transfer function
2. Representation of the generalized plant corresponding to the T_{wz} transfer function

3. Resolution of a minimization problem.

In particular, $T_{wz}(s)$ can be selected as:

$$T_{wz}(s) = \begin{bmatrix} W_1(s)S_n(s) \\ W_2(s)T_n(s) \end{bmatrix} \quad (1.9)$$

In (1.9), $S_n(s)$ and $T_n(s)$ are respectively the *nominal sensitivity function* and the *nominal complementary sensitivity function*. $W_1(s)$ and $W_2(s)$ are defined as

$$\begin{bmatrix} W_1(s) = W_S(s) \\ W_2(s) = W_T(s) \end{bmatrix} \quad (1.10)$$

1.7.1 Specifications Handling

The control design starts with the manipulation of the most common types of specifications. Starting from them, it is possible to derive

1. a constraint on the type of the controller (definition of ν : numbers of poles at the origin of $G_c(s)$)
2. a constraint on the gain of the $G_c(s)$ controller
3. a constraint on the value of the damping coefficient ζ
4. a constraint on the value of the resonant peaks on the sensitivity function and on the complementary sensitivity function
5. a constraint on the natural frequency ω_n
6. a constraint on the crossover frequency ω_c

They are very useful for the definition of the weighting functions.

1.7.2 The Weighting Functions

Let us consider a general feedback system

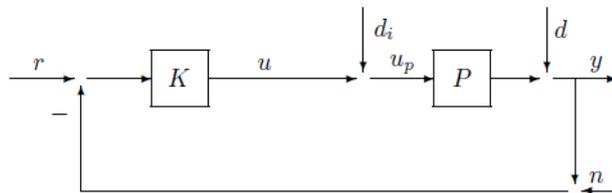


Figure 1.6: Standard Feedback Configuration [4].

Despite the simplicity of the representation, feedback design is challenging, because it is fundamental to assure that the "loop gains are not arbitrarily high over arbitrarily large frequency ranges." [4] They also have to satisfy performance trade-off and design limitations. In this sense, an important deal is to reduce commands and disturbance error in order to guarantee stability in presence of model uncertainties.

A key step is the selection of weighting functions that allow to guarantee and to verify robustness and stability of the system. The problem is that this task is very difficult to accomplish: there are no fixed criteria or mathematical procedures that demonstrate how they have to be derived. Their representation depends on the requirements of the problem under consideration. Even though there are no general rules to obey, it is possible to follow some guideline to simplify the process. These will be here underlined.

The most important variables to estimate the quality of the step time response are the *rise time* t_r , the *settling time* t_s and the percent *overshoot* \hat{s} . Starting from them, the corresponding specifications in frequency domain can be described in terms of requirements on the sensitivity function and the complementary sensitivity function.

To give an example, starting from the condition on the maximum overshoot of the response, it is possible to compute the requirements on the resonant peaks on both the sensitivity function and complementary sensitivity function, under the assumption that the closed loop system is characterized by a dominant second order dynamics:

$$\begin{cases} \hat{s} \leq \hat{s}_0 \implies \zeta \geq |\ln \hat{s}_0| \frac{|\ln \hat{s}_0|}{\sqrt{\pi^2 + \ln^2 \hat{s}_0}} \\ S_p \leq \frac{2\zeta \sqrt{2+4\zeta^2+2\sqrt{1+8\zeta^2}}}{\sqrt{1+8\zeta^2+4\zeta^2-1}} = S_{p0} \implies \text{requirement on } S \\ T_p \leq \frac{1}{2\zeta\sqrt{1-\zeta^2}} \implies \text{requirement on } T \end{cases} \quad (1.11)$$

To approximate the frequency domain constraints on S and T it is possible to use the weighting functions W_S and W_T .

$W_S(s)$ Weighting Function Description

Generally, a possible choice for $W_S(s)$ is

$$W_S(s) = \frac{1 + 1.414 \frac{s}{\omega_2} + \frac{s^2}{\omega_2^2}}{as(1 + \frac{s}{\omega_1})} \quad (1.12)$$

Both the numerator and the denominator are given by a *butterworth polynomial*, respectively of the first and second order to have a frequency response as flat as

possible in the pass-band.

In order to satisfy the requirement on the resonant peak of the sensitivity function, the behavior of $W_S(s)$ can be described as in Figure 1.7.

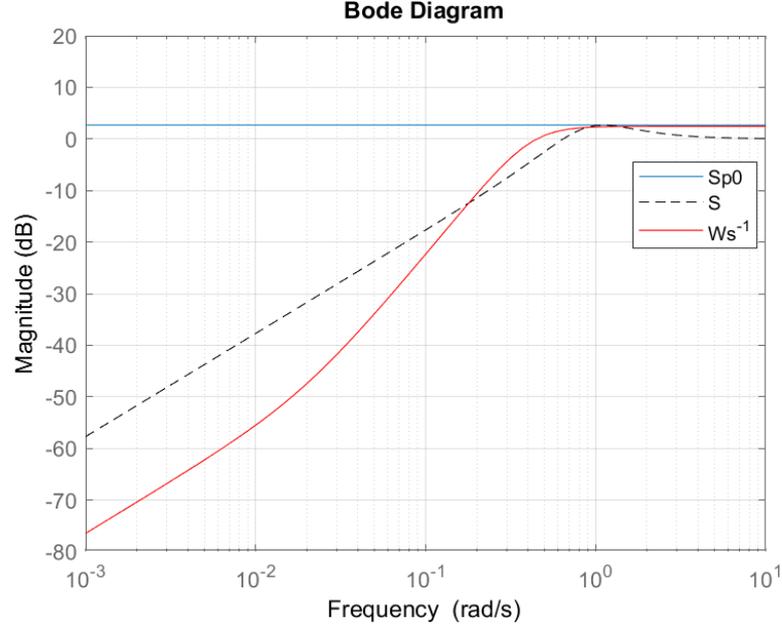


Figure 1.7: $W_S(s)$ Weighting Function

The coefficients a, ω_1, ω_2 have to be derived

$$a = \lim_{s \rightarrow 0} \frac{1}{s} W_S^{-1}(s) = S^*(0) \quad (1.13)$$

$S^*(0)$ describes the slope of the weighting function and it is computed starting from a different formulation of the sensitivity function

$$S(s) = s^{\nu+p} S^*(s) \quad (1.14)$$

In (1.14), the power $(\nu + p)$ indicates the system type: ν is the number of poles at the origin of the controller and p is the number of poles at the origin of the plant. From a practical point of view, the numerical value of $S^*(0)$ comes from the analysis of the requirements on the steady-state output error behavior. By the way, during the development of the project a concrete proof of its calculation will be provided.

ω_1 is chosen by trial and error

$$\omega_2 = \sqrt{\frac{\omega_1 S_{p0}}{a}} \quad (1.15)$$

By manipulating the ω_1 and consequently ω_2 coefficients, it is possible to move the function closer or farther to S .

$W_T(s)$ Weighting Function Description

In order to satisfy the requirement on the resonant peak of the complementary sensitivity function, the behavior of $W_T(s)$ can be described as in Figure 1.8

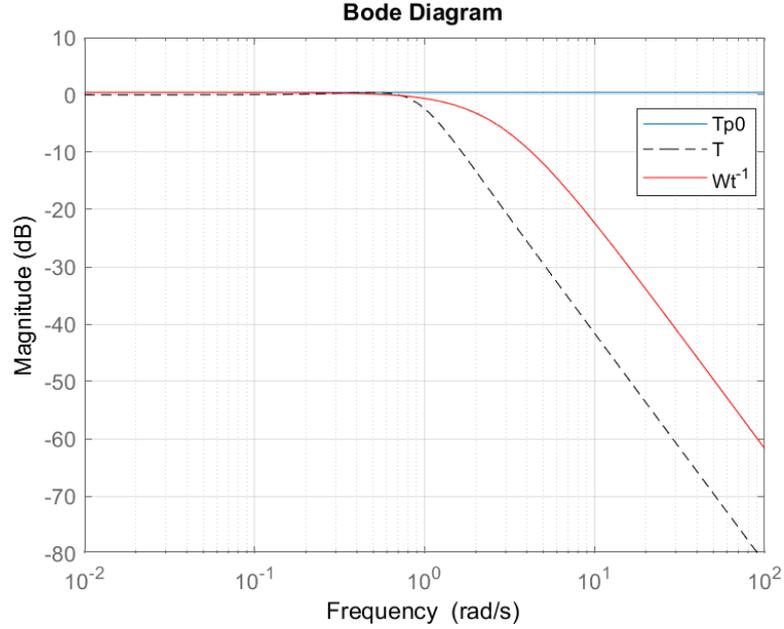


Figure 1.8: $W_T(s)$ Weighting Function

The purpose is to build a function whose values are never greater than the maximum resonant peak T_{p0} and that is closer to $T(s)$. To reach this scope, two alternative representations can be used

$$W_T(s) = \frac{1 + \frac{2\zeta}{\omega_3 s} + \frac{s^2}{\omega_3^2}}{T_{p0}} \quad (1.16)$$

$$W_T(s) = \frac{\left(1 + \frac{s}{\omega_3}\right)^2}{T_{p0}} \quad (1.17)$$

Expression (1.17) is the preferred one and ω_3 is chosen by trial and error. The greater is its values, the farther is the function from T .

Final Considerations on the Weighting Functions

As a result of the latter definitions, the weighting functions have to be designed in such a way that the following conditions are satisfied

$$\begin{cases} \max_{\omega} |W_S^{-1}| \leq S_{p0} \\ \max_{\omega} |W_T^{-1}| \leq T_{p0} \end{cases} \quad (1.18)$$

If frequency disturbance attenuation requirements are specified, further constraints can be added to the formulation of the weighting functions [6]. This means that:

if a disturb d_p acts on the plant of the system

$$d_p = a_p \sin \omega_p t \quad \forall \omega_p \leq \omega_p^+ \quad (1.19)$$

with ω_p^+ and a_p given, and the output error due to this disturb d_p is required to be bounded by a given ρ_p

$$|W_S^{-1}(j\omega)| \leq \frac{\rho_p}{a_p} = M_S^{LF} \quad \forall \omega_p \leq \omega_p^+ \quad (1.20)$$

M_S^{LF} represents the needed low frequency attenuation.

In the same way, if a disturb d_s acts on the sensor

$$d_s = a_s \sin \omega_s t \quad \forall \omega_s \geq \omega_s^- \quad (1.21)$$

with ω_s^- and a_s given, and the output error due to this disturb d_s is required to be bounded by a given ρ_s

$$|W_T^{-1}(j\omega)| \leq \frac{\rho_s}{a_s} = M_T^{HF} \quad \forall \omega_s \geq \omega_s^- \quad (1.22)$$

M_T^{HF} represents the needed high frequency attenuation.

Nominal Performance, Robust Performance and Robust Stability

$W_S(s)$ and $W_T(s)$ have an important role and have to be designed in a correct shape because they also allow to check the *nominal performance*, that is the performance conditions in the uncertainty-free case:

$$\begin{cases} \|W_S S_n\|_{\infty} < 1 \\ \|W_T T_n\|_{\infty} < 1 \end{cases} \quad (1.23)$$

Otherwise, to guarantee robust stability and robust performance, the $W_U(s)$ weighting function is needed

$$\|W_U T_n\|_\infty < 1 \quad (1.24)$$

for the robust stability

$$\| |W_S S_n| + |W_U T_n| \|_\infty < 1 \quad (1.25)$$

for the robust performance

Not always the original functions $W_S(s)$ and $W_T(s)$ are able to give satisfactory results in time domain. In this case, they need to be modified to obtain a control system that fulfills the initial requirements. To avoid the loss of their original definition, two new functions, $W_1(s)$, $W_2(s)$ as in (1.10), are involved. They are used to obtain good results, while $W_S(s)$ and $W_T(s)$ are used to verify the performances of the system.

1.7.3 LMI Approach

The solution of (2.6) can be achieved by applying different methodologies. The *LMI Approach* will be used. According to it, the generalized plant is described as

$$M : \begin{cases} \dot{x}_M = Ax_M + B_1 w + B_2 u \\ z = C_1 x_M + D_{11} w + D_{12} u \\ v = C_2 x_M + D_{21} w + D_{22} u \end{cases} \quad (1.26)$$

In (1.26), x_M is the state of the plant given by the union of the G_p , W_S and W_T state variables.

M can be internally stabilized by a LTI controller $G_c(s)$ only if $W_S(s)$ and $W_T(s)$ are stable transfer functions. But, considering (1.12) and (1.17), $W_S(s)$ has a pole at the origin, while $W_T(s)$ is an improper function. They cannot be used.

Because of this

$$W1_{mod} = W_1 \frac{s^{\nu+p}}{(s + \lambda)^{\nu+p}} \quad (1.27)$$

$$W2_{mod} = \frac{1}{T_{p0}} \quad (1.28)$$

are introduced to replace them in the definition of the generalized plant. λ is chosen as $\lambda = 0.1w_c$. This is a possible option, but not the only one. It substitutes the pole at the origin that makes the function $W_S(s)$, and consequently $W_1(s)$, unstable with a low frequency one, $(\nu + p)$ is the order of the system deriving from the study of the specifications.

The *hinf* Matlab toolbox implements the LMI-based approach. It computes the best H_∞ performance and an H_∞ controller G_c that internally stabilizes the plant and generates a closed-loop gain that is no greater than the computed performance. The calculated controller G_c can be, then, modified by removing poles at high frequencies or couples of zeros-poles at low frequencies. In the definition of $W2_{mod}$, the zeros of W_2 have been eliminated. Actually, they will be further compensated by means of the Matlab command

$$dsys = sderiv(sys, chan, pd)$$

It multiplies an input or output channel of the LTI system "sys" for a PD component $Ns + D$ where the coefficients N and D are specified by the pd vector $pd = [N, D]$.

1.7.4 Advantages and Disadvantages of the H_∞ Control System

H_∞ is a robust control method, so it is designed to behave correctly also in presence of modeling errors. The most important advantage deriving from the algorithm is the possibility of including the uncertainties of the system in the design process. Clearly, there are also disadvantages in its use. As it has been said, two weighting functions are useful to group together and synthesize all the performance requirements. They are selected using trial and error techniques and this does not always assure that the controller computed by the *hinf* toolbox is the optimal one in terms of achievement of the control objectives. This depends on the way in which they have been shaped. For this reason, it very difficult to build an algorithm that, given some inputs, automatically returns as output the best $G_c(s)$ that allows to fulfill all the control objectives. The definition of the $W_S(s)$, $W_T(s)$ and $W_u(s)$ is a slow process that can require various iterations before obtaining functions that will yield good results.

Part II

A Benchmark Problem for the Control of an Industrial Manipulator

Chapter 2

Feedback Control System

The implemented feedback control system is showed in Figure 2.1.

$G_c(s)$ is the controller computed by means of the H_∞ algorithm, $G_p(s)$ is the plant expressing the relationship between the input u_m and the output q_m .

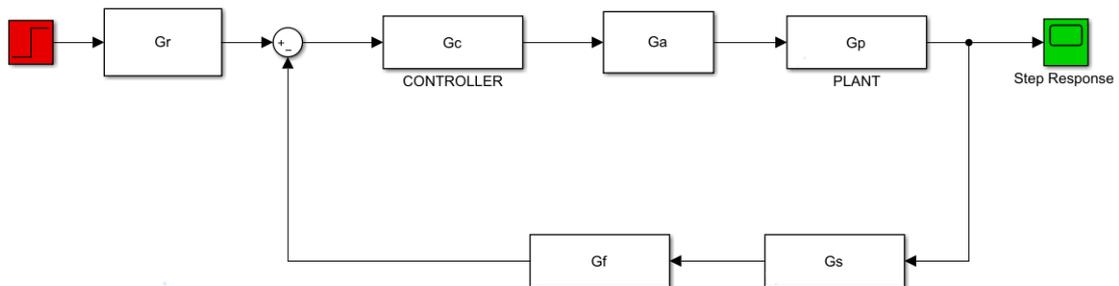


Figure 2.1: Feedback Control System

2.1 The relevance of a Benchmark Problem

In control researches, experiments have a big relevance. Through years, it has been proved that scientific methods have a larger potential to work compared to theoretical ones.

A *benchmark problem* can play as a substitute for real control experiments, since it can be presented in a sufficiently realistic, complete and not too complex form.

2.2 The industrial Manipulator

The most common industrial manipulator (Figure 2.2) is formed by six links controlled by electrical motors via gears. It can be described as a nonlinear multi-variable dynamic system with six inputs (motor currents) and six outputs (motor angles).

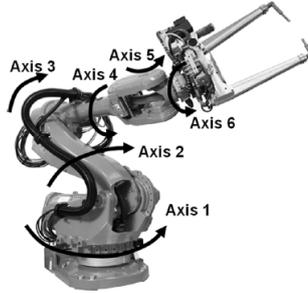


Figure 2.2: Industrial Manipulator [7]

Hereafter in the discussion, only the first axes of the manipulator will be analyzed. The other ones are set in a fixed configuration in order to be able to neglect their nonlinear dynamics.

2.3 Dynamic Model Derivation

First axes dynamics can be described by means of a four-masses model(Figure 2.3)

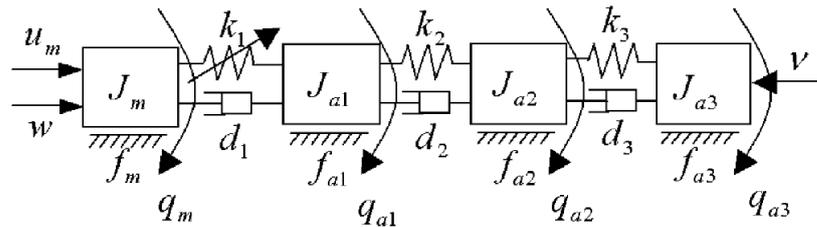


Figure 2.3: Four-masses Model [7]

The moment of inertia of the arm is divided into three components J_1, J_2, J_3 . J_m is the moment of inertia of the motor. f_m, f_1, f_2, f_3 represent the viscous frictions in the motor and the arm structure. u_m is the motor torque, the input of the system. v, w are disturbances acting on the motor and the arm. q_1, q_2, q_3 are the

angles of the three masses, useful to define the end-effector position, while q_m is the angle of the motor.

The manipulator dynamics can be implemented through four differential equations

$$J_m \ddot{q}_m = u_m + w - f_m \dot{q}_m - \tau_{gear} - d_1(\dot{q}_m - \dot{q}_1) \quad (2.1)$$

$$J_1 \ddot{q}_1 = -f_1 \dot{q}_1 + \tau_{gear} + d_1(\dot{q}_m - \dot{q}_1) - k_2(q_1 - q_2) - d_2(\dot{q}_1 - \dot{q}_2) \quad (2.2)$$

$$J_2 \ddot{q}_2 = -f_2 \dot{q}_2 + k_2(q_1 - q_2) + d_2(\dot{q}_1 - \dot{q}_2) - k_3(q_2 - q_3) - d_3(\dot{q}_2 - \dot{q}_3) \quad (2.3)$$

$$J_3 \ddot{q}_3 = v - f_3 \dot{q}_3 + k_3(q_2 - q_3) - d_3(\dot{q}_2 - \dot{q}_3) \quad (2.4)$$

The rotating masses are linked via spring-damper pairs. The first one, corresponding to the gear, has linear damping d_1 and nonlinear elasticity τ_{gear} . In particular, the spring torque τ_{gear} is a function of the deflection $(q_m - q_1)$ (Figure 2.4). The second and third masses, instead, are both linear and represented by d_2, k_2, d_3, k_3 .

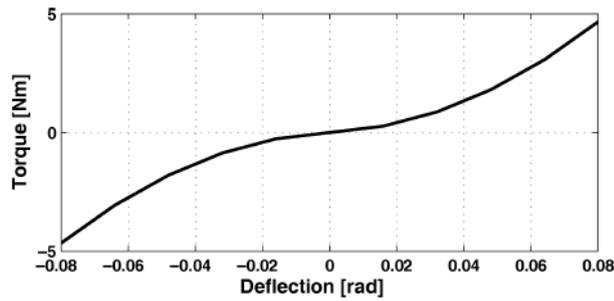


Figure 2.4: Spring torque as a nonlinear function of $(q_m - q_1)$ [7]

The numerical values of the above quantities, needed to define the model, as well to simulate it, are taken from the scientific article: "A Benchmark Problem For Robust Feedback Control of a Flexible Industrial Manipulator"[7] and listed in Table 2.1.

Parameter	Value	Unit
J_m	$5 \cdot 10^{-3}$	kgm^2
J_1	$2 \cdot 10^{-3}$	kgm^2
J_2	0.02	kgm^2
J_3	0.02	kgm^2
k_1	100	Nm/rad
k_2	110	Nm/rad
k_3	80	Nm/rad
d_1	0.08	Nms/rad
d_2	0.06	Nms/rad
d_3	0.08	Nms/rad
f_m	$6 \cdot 10^{-3}$	Nms/rad
f_1	$1 \cdot 10^{-3}$	Nms/rad
f_2	$1 \cdot 10^{-3}$	Nms/rad
f_3	$1 \cdot 10^{-3}$	Nms/rad
u_m	25	Nm

Table 2.1: Parameter values

2.3.1 Model Reduction

The differential equations have been modified to reduce the complexity of the problem: the spring torque τ_{gear} is described by a linear function $k_1(q_m - q_1)$, the disturbances (v, w) and d_2, d_3, k_2, k_3 are considered equal to zero. On the basis of the previous considerations, the equations modeling the system become

$$J_m \ddot{q}_m = u_m - f_m \dot{q}_m - k_1(q_m - q_1) - d_1(\dot{q}_m - \dot{q}_1) \quad (2.5)$$

$$J_1 \ddot{q}_1 = -f_1 \dot{q}_1 + k_1(q_m - q_1) + d_1(\dot{q}_m - \dot{q}_1) \quad (2.6)$$

$$J_2 \ddot{q}_2 = -f_2 \dot{q}_2 \quad (2.7)$$

$$J_3 \ddot{q}_3 = -f_3 \dot{q}_3 \quad (2.8)$$

Furthermore, to derive the plant only (2.5) and (2.6) are useful. The final model is represented by

$$J_m \ddot{q}_m = u_m - f_m \dot{q}_m - k(q_m - q) - d(\dot{q}_m - \dot{q}) \quad (2.9)$$

$$J \ddot{q} = -f \dot{q} + k(q_m - q) + d(\dot{q}_m - \dot{q}) \quad (2.10)$$

where , $J = J_1$, $q = q_1$, $k = k_1$ and $f = f_1$.

2.4 Plant Definition

Starting from (2.9) and (2.10), it is possible to compute the transfer function of the plant. Its expression has been calculated by following a mathematical procedure that involves the definition of the state equations and the state matrices. At the end, the result has been validated through Simulink.

State Equations and State Matrices

By manipulating (2.9) and (2.10)

$$\ddot{q}_m = \frac{1}{J_m}(u_m - f_m \dot{q}_m - k(q_m - q) - d(\dot{q}_m - \dot{q})) \quad (2.11)$$

$$\ddot{q} = \frac{1}{J}(-f \dot{q} + k(q_m - q) + d(\dot{q}_m - \dot{q})) \quad (2.12)$$

It is possible to identify four states

$$x = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} q_m \\ \dot{q}_m \\ q \\ \dot{q} \end{bmatrix} \quad (2.13)$$

The state equations are

$$\begin{cases} \dot{x}_1 = x_2 \\ \dot{x}_2 = \frac{1}{J_m}(u_m - f_m x_2 - k(x_1 - x_3) - d(x_2 - x_4)) \\ \dot{x}_3 = x_4 \\ \dot{x}_4 = \frac{1}{J}(-f x_4 + k(x_1 - x_3) + d(x_2 - x_4)) \end{cases} \quad (2.14)$$

The state matrices are

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 \\ -\frac{k}{J_m} & -\frac{f_m+d}{J_m} & \frac{k}{J_m} & \frac{d}{J_m} \\ 0 & 0 & 0 & 0 \\ \frac{k}{J} & \frac{d}{J} & -\frac{k}{J} & -\frac{(f+d)}{J} \end{bmatrix} \quad (2.15)$$

$$B = \begin{bmatrix} 0 \\ \frac{1}{J_m} \\ 0 \\ 0 \end{bmatrix} \quad (2.16)$$

$$C = [1 \quad 0 \quad 0 \quad 0] \quad (2.17)$$

$$D = [0] \quad (2.18)$$

$G(s)$ Transfer Function

An initial expression of the plant is given by

$$G(s) = \frac{200(s^2 + 40.5s + 5 \cdot 10^4)}{s(s + 1)(s^2 + 56.7s + 7 \cdot 10^4)} \quad (2.19)$$

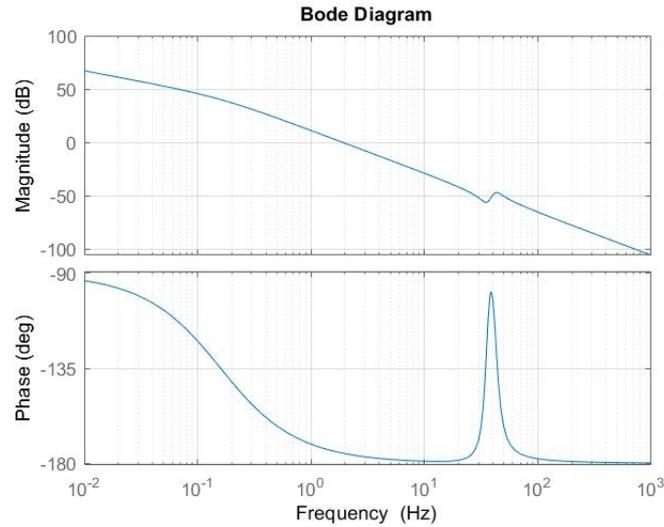


Figure 2.5: Bode Plot of the $G(s)$ transfer function

The zeros are

$$\begin{cases} z_1 = 1.0 \cdot 10^2(-0.2025 + 2.2269i) \\ z_2 = 1.0 \cdot 10^2(-0.2025 - 2.2269i) \end{cases} \quad (2.20)$$

The poles are

$$\begin{cases} p_1 = 0 \\ p_2 = -1 \\ p_3 = 1.0 \cdot 10^2(-0.2835 + 2.6305i) \\ p_4 = 1.0 \cdot 10^2(-0.2835 - 2.6305i) \end{cases} \quad (2.21)$$

Considering $(s^2 + 56.7s + 7 \cdot 10^4)$ at the denominator of (2.19), the damping factor and the natural frequency can be deduced:

it is always possible to write a not modular trinomial in the form

$$s^2 + 2\zeta\omega_n s + \omega_n^2 \quad (2.22)$$

The damping factor and the natural frequency can be obtained by solving the system

$$\begin{cases} 2\zeta\omega_n = 56.7 \\ \omega_n^2 = 7 \cdot 10^4 \end{cases} \quad (2.23)$$

It results

$$\begin{cases} \zeta = 0.1072 \\ \omega_n = 264.5751 \end{cases} \quad (2.24)$$

Notch Filter Application

A *Notch Filter* is applied to (2.19) to remove the complex poles given by the polynomial $(s^2 + 56.7s + 7 \cdot 10^4)$.

A Notch Filter is a dynamic system defined by the rational function

$$G_N(s) = \frac{s^2 + 2\zeta_1\omega_n s + \omega_n^2}{s^2 + 2\zeta_2\omega_n s + \omega_n^2} \quad (2.25)$$

By selecting

$$\begin{cases} \omega_n = 264.5751 \\ \zeta_1 = 0.1072 \\ \zeta_2 = 1.001 \end{cases} \quad (2.26)$$

$$G_N(s) = \frac{s^2 + 56.72s + 7 \cdot 10^4}{(s + 268.3)(s + 260.9)} \quad (2.27)$$

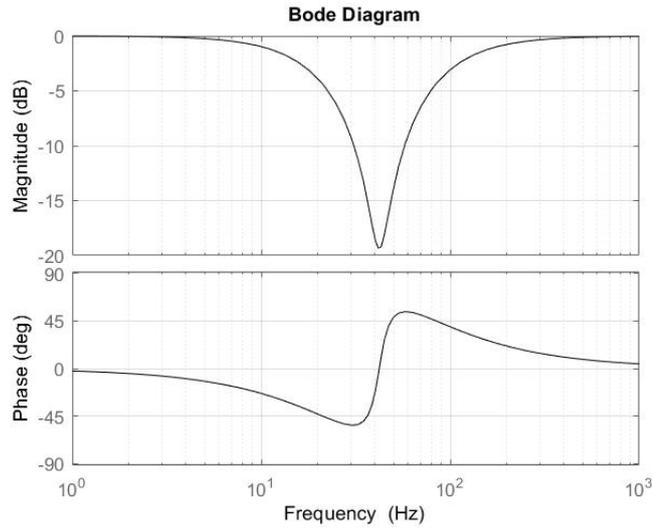


Figure 2.6: Bode Plot of the Notch Filter $G_N(s)$

With the application of $G_N(s)$, the final expression of the plant under analysis is

$$G_p(s) = G(s)G_N(s) = \frac{200(s^2 + 40.5s + 5 \cdot 10^4)}{s(s+1)(s+260.9)(s+268.3)} \quad (2.28)$$

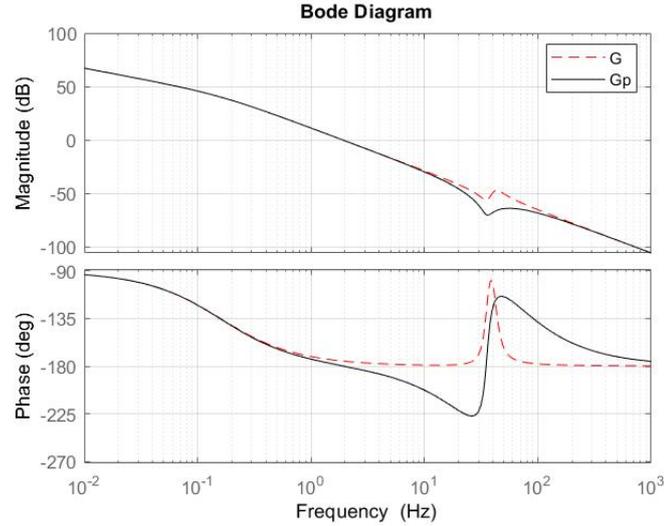


Figure 2.7: Bode Plot of $G(s)$ and $G_p(s)$

2.5 H_∞ Control Implementation

The starting point in the H_∞ algorithm is the manipulation of the specifications. In this sense, the requirements chosen to build a controller that guarantees a response characterized by a small rise time ($t_r \leq 200ms$) and a null overshoot are:

1. Steady-state Gain of the Feedback Control system: $K_d = 1$
2. Steady-state output error when the reference is a ramp ($R_0 = 1$):
 $|e_r^\infty| \leq 0.0001$
3. Step response overshoot $\hat{s} \leq 0.1$
4. Rise time: $t_r \leq 0.5s$
5. Settling time: $t_{s,5\%} \leq 1.5s$

Mathematical procedures will be performed in order to transform the specifications in data useful to shape the weighting functions and to derive $G_c(s)$.

$$G_r = G_a = G_s = 1$$

$$1. K_d = \lim_{s \rightarrow 0} G_{ry} = \lim_{s \rightarrow 0} \frac{G(s)}{1+G(s)H(s)}$$

From Figure 2.1: $G(s) = G_a(s)G_p(s)G_c(s)$ and $H(s) = G_s(s)G_f(s)$.

$$\text{Since } (\nu + p) \geq 1 \implies K_d = \frac{1}{G_s G_f} \implies G_f = \frac{1}{K_d G_s}$$

$$2. |e_r^\infty| = |\lim_{t \rightarrow \infty} e_r(t)| =$$

By applying the *final value theorem*

$$\begin{aligned} &= |\lim_{s \rightarrow 0} s e_r(s)| = |\lim_{s \rightarrow 0} s K_d S(s) R(s)| = |\lim_{s \rightarrow 0} s s^{\nu+p} S^*(s) \frac{R0}{s^2}| = \\ &= |\lim_{s \rightarrow 0} s s^p s^\nu S^*(s) \frac{R0}{s^2}| = |\lim_{s \rightarrow 0} s^2 s^\nu S^*(s) \frac{R0}{s^2}| = |\lim_{s \rightarrow 0} s^\nu S^*(s) R0| \end{aligned}$$

Since $|e_r^\infty|$ has to be **finite** and **different from zero**: $\nu \geq 0$.

Under this condition:

$$|e_r^\infty| = |K_d S^*(0) R0| \leq 0.0001 \implies |S^*(0)| \leq \frac{0.0001}{K_d R0} = 0.0001$$

$$3. \zeta \geq |\ln \hat{s}_0| \frac{\ln \hat{s}_0}{\sqrt{\pi^2 + \ln^2 \hat{s}_0}} = 0.5912$$

$$T_p \leq T_{p0} = \frac{1}{2\zeta \sqrt{1-\zeta^2}} = 1.0487 = 0.4127dB$$

$$S_p \leq S_{p0} = \frac{2\zeta \sqrt{2+4\zeta^2+2\sqrt{1+8\zeta^2}}}{\sqrt{1+8\zeta^2+4\zeta^2-1}} = 1.3611 = 2.6775dB$$

$$4. \omega_{n,1} = \frac{1}{t_r \sqrt{1-\zeta^2}} = 2.4797 \frac{rad}{s} = 0.39Hz$$

$$\omega_{c,1} = \frac{1}{t_r \sqrt{1-\zeta^2}} \sqrt{\sqrt{1-4\zeta^4} - 2\zeta^2} = 1.79 \frac{rad}{s} = 0.28Hz$$

$$5. \omega_{n,2} = \frac{-\log \alpha}{t_s \zeta} = 3.3784 \frac{rad}{s} = 0.54Hz$$

$$\omega_{c,2} = \frac{-\log \alpha}{t_s \zeta} \sqrt{\sqrt{1-4\zeta^4} - 2\zeta^2} = 2.4388 \frac{rad}{s} = 0.39Hz$$

$$\omega_n = \max(\omega_{n,1}, \omega_{n,2}) = 3.3784 \frac{rad}{s}, \omega_c = \max(\omega_{c,1}, \omega_{c,2}) = 2.4388 \frac{rad}{s}$$

2.5.1 Weighting Functions Definition

The choice of $W_S(s)$ and $W_T(s)$ represents the most important task in the definition of the controller.

Starting from the original formulation of $W_S(s)$ in (1.12), it follows that

$$W_S^{-1}(s) = \frac{as(1 + \frac{s}{\omega_1})}{1 + 1.414 \frac{s}{\omega_2} + \frac{s^2}{\omega_2^2}} = \frac{1.3611s(s + 0.0003)}{s^2 + 2.857s + 4.083} \quad (2.29)$$

For (1.13),(1.15)

$$\begin{cases} a = S^*(0) = 0.0001 \\ \omega_1 = 0.0003 \implies \omega_2 = 2.0207 \end{cases} \quad (2.30)$$

The rational function (2.29) includes the requirement on the maximum resonant peak and takes into account the value of $S^*(0)$ derived from the constraint on the steady-state output error in front of a ramp reference.

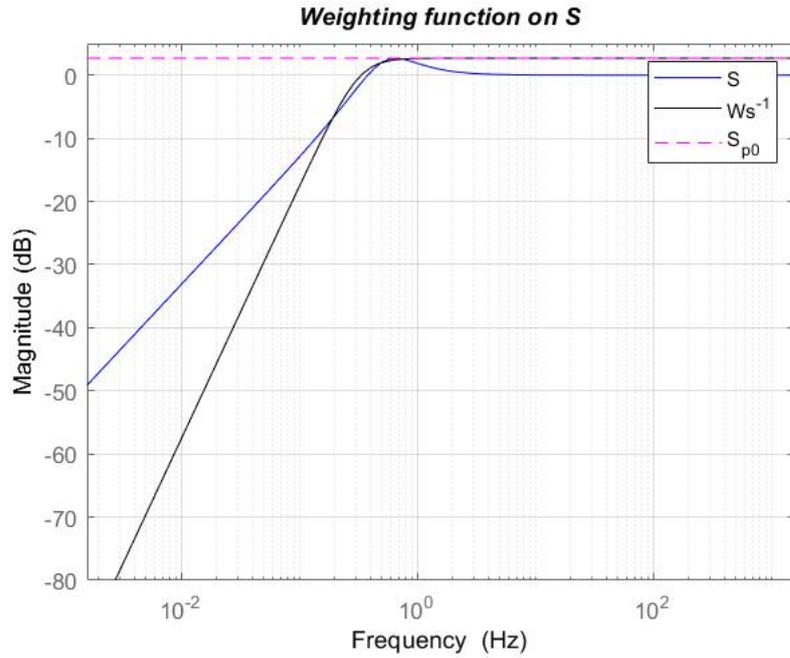


Figure 2.8: Weighting Function on $S(s)$

In the same way, it is possible to build the second weighting function $W_T(s)$. Given (1.17) and choosing $\omega_3 = 10 \frac{rad}{s}$, it results:

$$W_T^{-1}(s) = \frac{T_{p0}}{(s + \omega_3)^2} = \frac{1.0487}{(s + 10)^2} \quad (2.31)$$

that incorporates the constraint on the resonant peak T_{p0} .

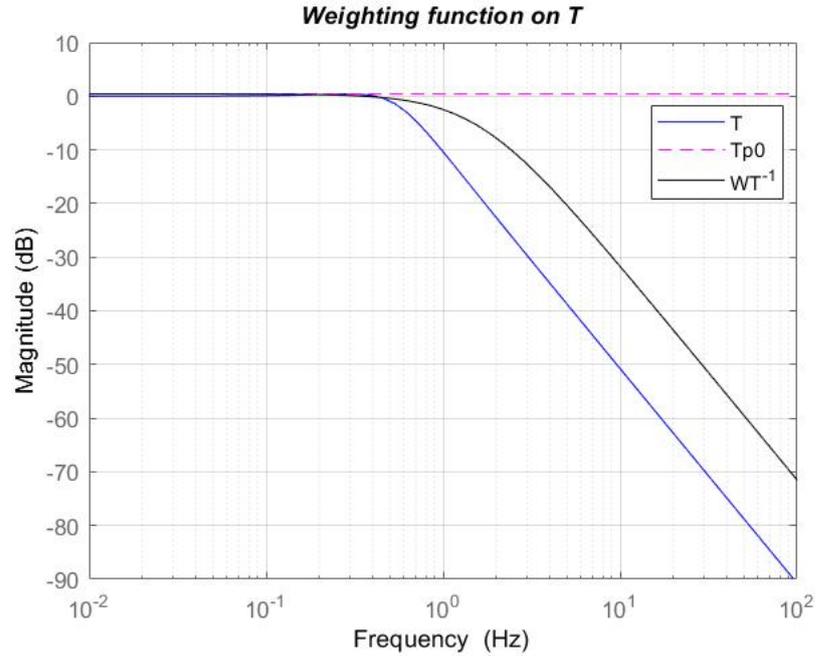


Figure 2.9: Weighting Function on $T(s)$

Both $W_S(s)$ and $W_T(s)$ fulfil the condition (1.18), as can be seen in Figure 2.8 and 2.9.

Unfortunately, the adoption of the original weighting functions does not provide good results. From the time simulation in Figure 2.10, the numerical values of \hat{s} and t_r are computed:

$$\begin{cases} \hat{s} = 13.1\% \\ t_r = 0.422s \end{cases} \quad (2.32)$$

Modified Weighting Functions

As described in Chapter 1, sometimes it is necessary to change the original form of the weighting functions to get better outcomes.

The final expressions for $W_1(s)$ and $W_2(s)$ are

$$\begin{cases} W_1(s) = \frac{0.73472(s^2+0.165s+0.01361)}{s(s+10^{-6})} \\ W_2(s) = \frac{(s+32)^2}{1.0487} \end{cases} \quad (2.33)$$

Following a *trial and error* procedure, in fact, it has been observed that:

- By selecting greater zeros of $W_T(s)$, the overshoot of the step response decreases and the crossing frequency ω_c increases, making the system faster
- By selecting a smaller ω_1 frequency of $W_S(s)$, the overshoot of the step response decreases but the crossing frequency ω_c increases, making the system slower.

For the latter reasons, it is important to balance the variations on the weighting functions in such a way to guarantee the total satisfaction of the requirements.

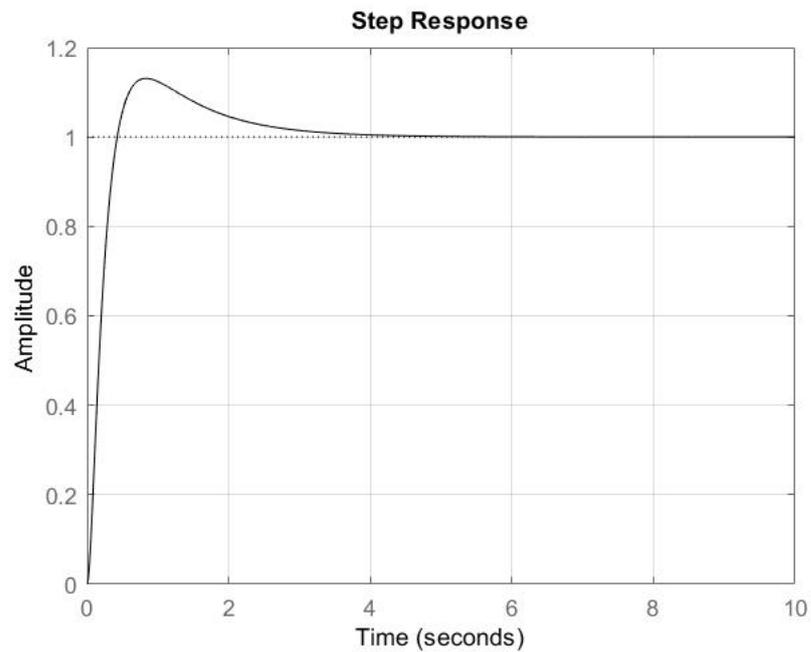


Figure 2.10: Step Time Simulation

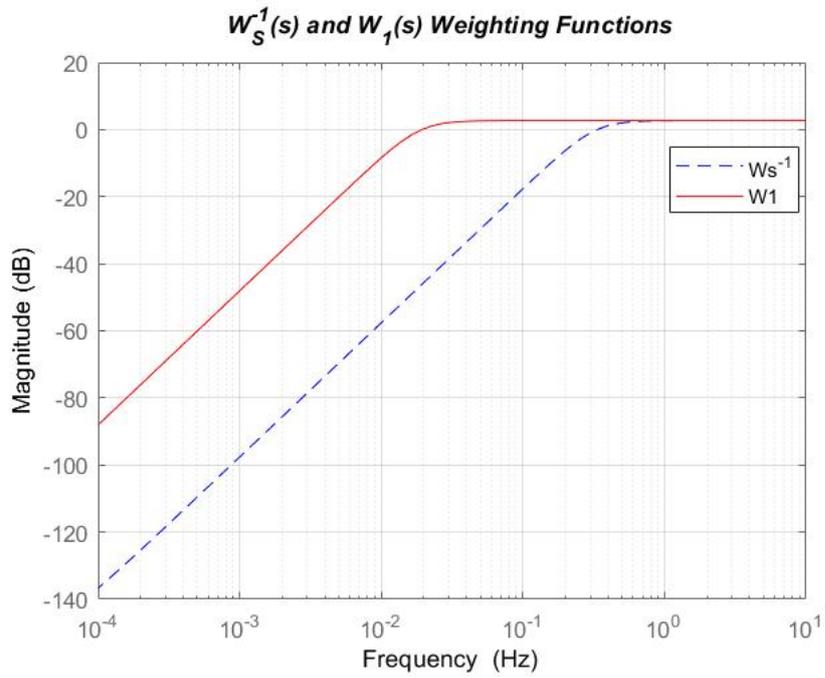


Figure 2.11: $W_S(s)$ and $W_1(s)$

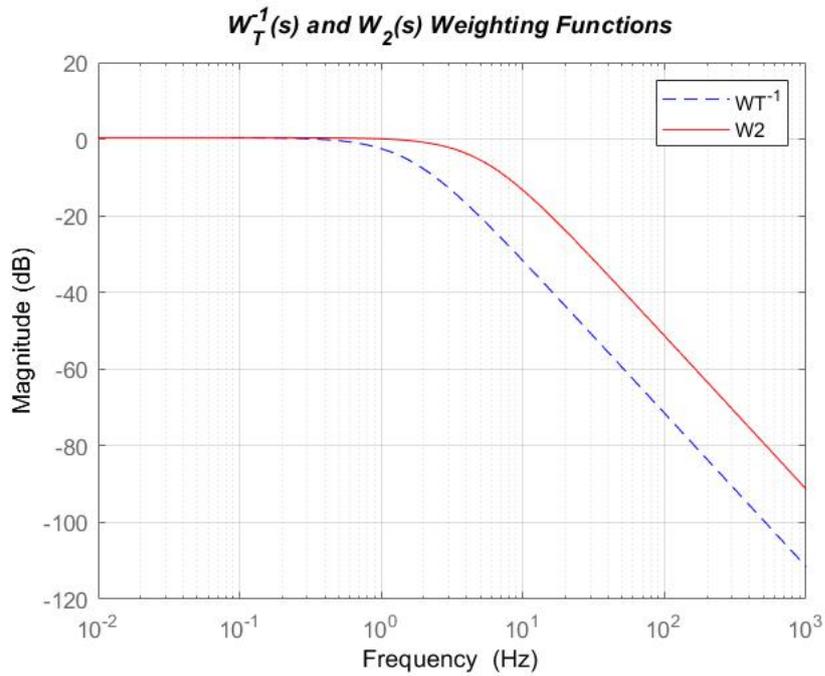


Figure 2.12: $W_T(s)$ and $W_2(s)$

2.5.2 The Controller

The controller calculated by the *hinflmi* Matlab toolbox is

$$G_c(s) = \frac{302.26(s + 268.3)(s + 260.9)(s + 0.9926)(s + 0.03279)(s + 0.0004242)}{(s + 0.02439)(s + 1.318 \cdot 10^{-7})(s^2 + 178.8s + 9192)(s^2 + 42.69s + 5.001 \cdot 10^4)} \quad (2.34)$$

However, since the zeros and poles of $G_c(s)$ affecting the behavior of the system under analysis are the ones that belong to the **middle region frequency**, the couple zero-pole at low frequency can be removed without altering the dynamics and the performances.

By applying such manipulations, the function becomes

$$G_c(s) = \frac{302.26(s + 268.3)(s + 260.9)(s + 0.9926)(s + 0.03279)}{(s + 0.02439)(s^2 + 178.9s + 9204)(s^2 + 42.69s + 5.001 \cdot 10^4)} \quad (2.35)$$

This is not the final expression for $G_c(s)$. To obtain results that fully reflect the control desires: $G_{c_p}(s) = K_c G_c(s) = 1.88 G_c(s)$

$$G_c(s) = \frac{568.25(s + 268.3)(s + 260.9)(s + 0.9926)(s + 0.03279)}{(s + 0.02439)(s^2 + 178.9s + 9204)(s^2 + 42.69s + 5.001 \cdot 10^4)} \quad (2.36)$$

The Nichols Plot of the Open Loop function computed starting from the expression of the controller defined by (2.36) is described in Figure 2.13

2.5.3 Time Simulation

To analyze the time performances of the system, the closed loop transfer function has to be computed.

$$L_{cl} = \frac{1.1365 \cdot 10^5 (s^2 + 40.5s + 5 \cdot 10^4)}{(s + 93.46)(s + 67.2)(s + 18.09)(s^2 + 42.7s + 5.002 \cdot 10^4)} \quad (2.37)$$

The time response (Figure 2.14) proves the complete fulfillment of the imposed requirements

$$\begin{cases} \hat{s} = 0\% \\ t_r = 0.132s \end{cases} \quad (2.38)$$

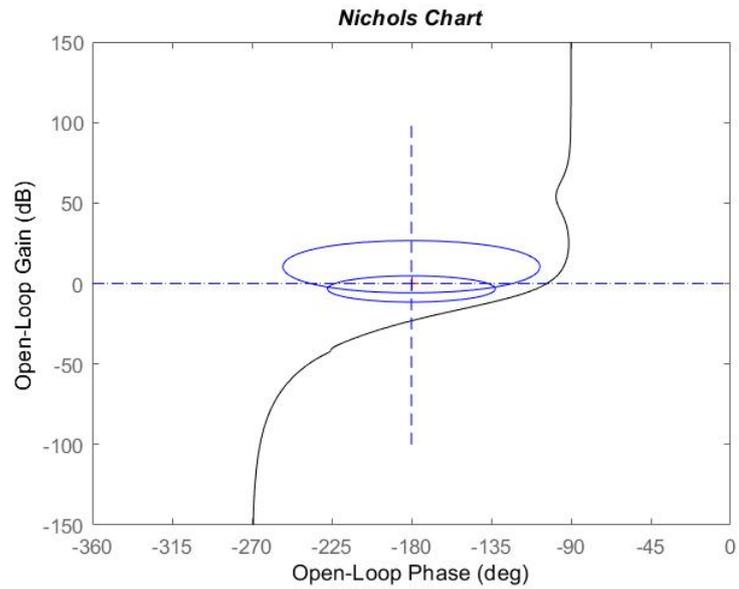


Figure 2.13: Nichols Plot of the Open-Loop function corresponding to $G_c(s)$ (2.36)

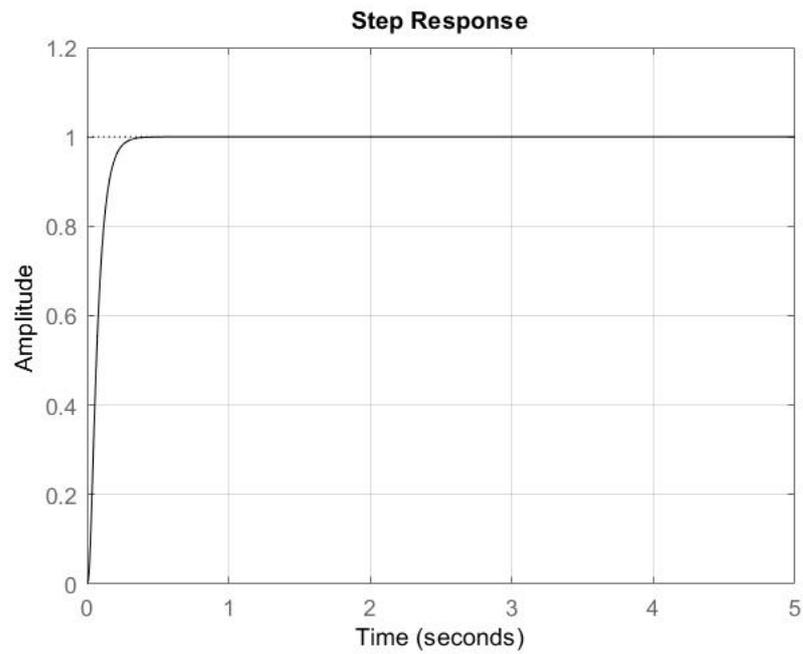


Figure 2.14: Time Response

Chapter 3

Nested Control System

To make the control system more robustly stable, an inner feedback loop has been inserted.

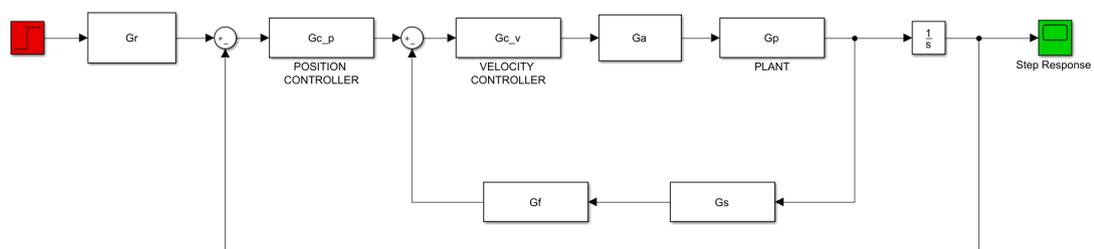


Figure 3.1: Nested Control System

The introduction of two different loops follows a *divide et impera* technique, because it permits to split up a large problem into two smaller ones: the first one implements the *velocity control*, the second one implements the *position control*.

3.1 Requirements to Guarantee

The two typologies of control pursue the same aims:

1. Steady-state Gain of the Feedback Control system: $K_d = 1$
2. Steady-state output error when the reference is a ramp ($R_0 = 1$):
 $|e_r^\infty| \leq 0.0001$
3. Step response overshoot $\hat{s} \leq 0.1$
4. Rise time: $t_r \leq 0.5s$

5. Settling time: $t_{s,5\%} \leq 1.5s$

3.2 Velocity Control

3.2.1 Plant Definition

Since the controlled variable is the velocity, the plant expresses the relationship between the input u_m and the output \dot{q}_m .

Starting from (2.9) and (2.10)

$$G(s) = \frac{200(s^2 + 40.5s + 5 \cdot 10^4)}{(s + 1)(s^2 + 56.7s + 7 \cdot 10^4)} \quad (3.1)$$

Its Bode plot is illustrated in Figure 4.6

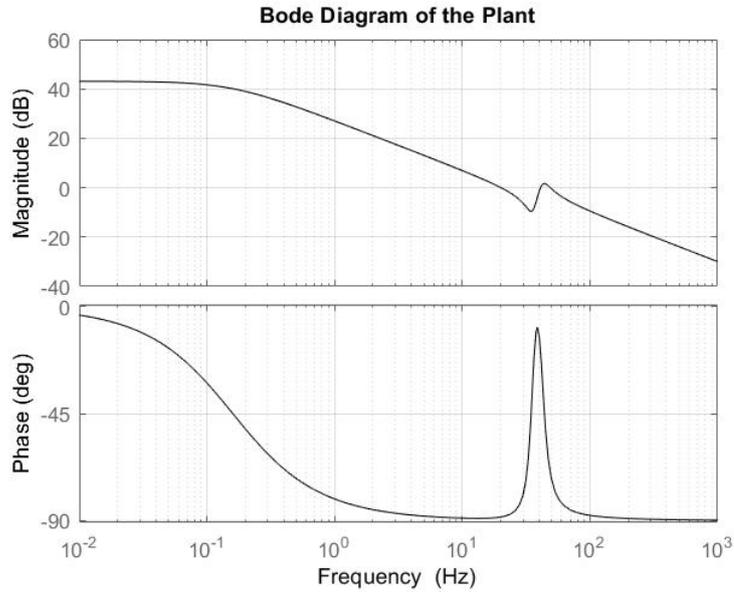


Figure 3.2: Bode Plot of $G(s)$

However, this transfer function has not been used for the realization of the inner control loop: the two complex poles at the denominator of $G(s)$ have been eliminated via the application of the Notch Filter (2.25).

The consequence is a rational function composed by three distinct real poles and two complex zeros

$$G_p(s) = \frac{200(s^2 + 40.5s + 5 \cdot 10^4)}{(s + 1)(s + 260.9)(s + 268.3)} \quad (3.2)$$

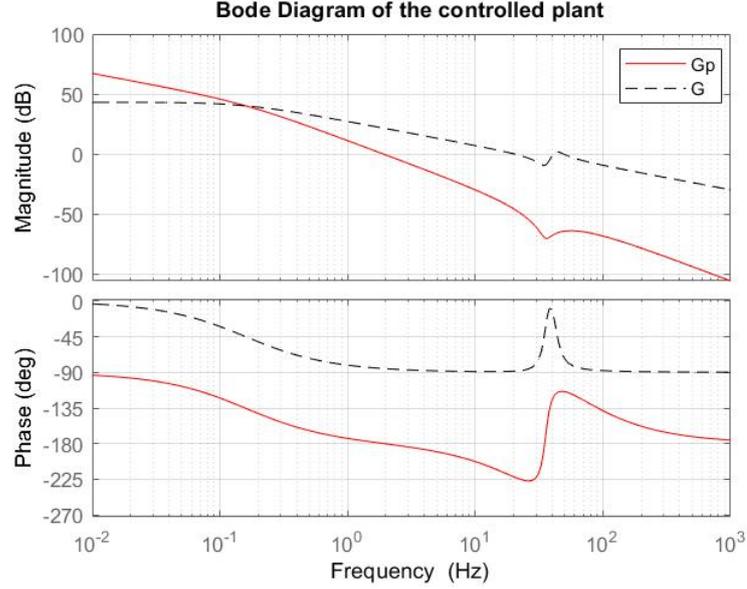


Figure 3.3: Bode Plot of $G(s)$ and $G_p(s)$

3.2.2 Velocity Specifications Handling

$$1. K_d = \lim_{s \rightarrow 0} G_{ry} = \lim_{s \rightarrow 0} \frac{G(s)}{1+G(s)H(s)}$$

From Figure 2.1: $G(s) = G_a G_p G_c$ and $H(s) = G_s G_f$.

$$\text{Since } (\nu + p) \geq 0 \implies K_d = \frac{1}{G_s G_f} \implies G_f = \frac{1}{K_d G_s}$$

$$2. |e_r^\infty| = |\lim_{t \rightarrow \infty} e_r(t)| =$$

By applying the *final value theorem*

$$\begin{aligned} &= |\lim_{s \rightarrow 0} s e_r(s)| = |\lim_{s \rightarrow 0} s K_d S(s) R(s)| = |\lim_{s \rightarrow 0} s s^{\nu+p} S^*(s) \frac{R0}{s^2}| = \\ &= |\lim_{s \rightarrow 0} s s^p s^\nu S^*(s) \frac{R0}{s^2}| = |\lim_{s \rightarrow 0} s s^\nu S^*(s) \frac{R0}{s^2}| = |\lim_{s \rightarrow 0} s^\nu S^*(s) \frac{R0}{s}| \end{aligned}$$

Since $|e_r^\infty|$ has to be **finite** and **different from zero**: $\nu \geq 1$.

Under this condition:

$$|e_r^\infty| = |K_d S^*(0) R0| \leq 0.0001 \implies |S^*(0)| \leq \frac{0.0001}{K_d R0} = 0.0001$$

$$3. \zeta \geq |\ln \hat{s}_0| \frac{\ln \hat{s}_0}{\sqrt{\pi^2 + \ln^2 \hat{s}_0}} = 0.5912$$

$$Tp \leq T_{p0} = \frac{1}{2\zeta \sqrt{1-\zeta^2}} = 1.0487 = 0.4127dB$$

$$Sp \leq S_{p0} = \frac{2\zeta\sqrt{2+4\zeta^2+2\sqrt{1+8\zeta^2}}}{\sqrt{1+8\zeta^2+4\zeta^2-1}} = 1.3611 = 2.6775dB$$

$$4. \omega_{n,1} = \frac{1}{t_r\sqrt{1-\zeta^2}} = 2.4797\frac{rad}{s} = 0.39Hz$$

$$\omega_{c,1} = \frac{1}{t_r\sqrt{1-\zeta^2}}\sqrt{\sqrt{1-4\zeta^4}-2\zeta^2} = 1.79\frac{rad}{s} = 0.28Hz$$

$$5. \omega_{n,2} = \frac{-\log\alpha}{t_s\zeta} = 3.3784\frac{rad}{s} = 0.54Hz$$

$$\omega_{c,2} = \frac{-\log\alpha}{t_s\zeta}\sqrt{\sqrt{1-4\zeta^4}-2\zeta^2} = 2.4388\frac{rad}{s} = 0.39Hz$$

$$\omega_n = \max(\omega_{n,1}, \omega_{n,2}) = 3.3784\frac{rad}{s}, \omega_c = \max(\omega_{c,1}, \omega_{c,2}) = 2.4388\frac{rad}{s}$$

3.2.3 Weighting Functions Definition

The final form of the weighting functions is

$$\begin{cases} W_1(s) = \frac{s^2+2.857s+4.083}{1.3611s(s+0.0003)} \\ W_2(s) = \frac{(s+0.5)^2}{1.0487} \end{cases} \quad (3.3)$$

3.2.4 G_{c_v} Controller

$$G_c(s) = \frac{1.729(s+268.4)(s+260.8)(s+0.9986)(s+0.6345)}{(s+216.6)(2+0.02435)(s+0.0002641)(s^2+40.5s+5\cdot 10^4)} \quad (3.4)$$

It can not be considered the definitive version of the controller. From the study of the requirements, in fact, it has been obtained a pole at the origin has to be inserted in the controller expression. To get it, it is possible to translate the pole at low frequency ($s+0.0002641$) in zero.

$$G_c(s) = \frac{1.729(s+268.4)(s+260.8)(s+0.9986)(s+0.6345)}{s(s+216.6)(2+0.02435)(s^2+40.5s+5\cdot 10^4)} \quad (3.5)$$

The open loop function is given by

$$L(s) = \frac{345.79(s+0.6345)}{s(s+216.6)(s+0.02435)} \quad (3.6)$$

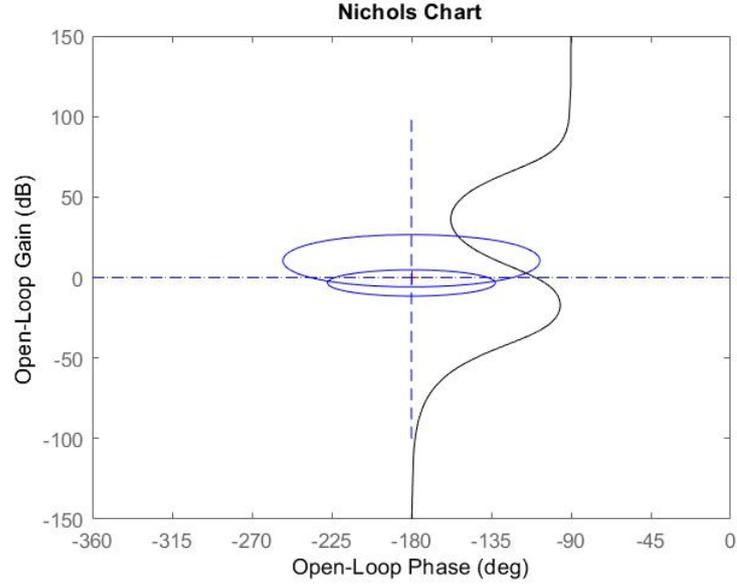


Figure 3.4: Nichols of the Loop Function $L(s)$

The crossover frequency is $\omega_c = 1.7 \frac{rad}{s} = 0.271 Hz$.
 To make the response faster, the ω_c value is increased. This has been made by multiplying $G_c(s)$ for a constant value $K_c > 1$.

$$G_{c_v}(s) = 35G_c(s) = \frac{60.514(s + 268.4)(s + 260.8)(s + 0.9986)(s + 0.6345)}{s(s + 216.6)(s + 0.02435)(s^2 + 40.5s + 5 \cdot 10^4)} \quad (3.7)$$

$$L(s) = \frac{12103(s + 0.6345)}{s(s + 216.6)(s + 0.02435)} \quad (3.8)$$

As expected, the ω_c value changes: $\omega_c = 54.2 \frac{rad}{sec} = 8.63 Hz$.
 The increment of the crossover frequency is also visible from the Nichols plot of $L(s)$ that is shifted towards up.

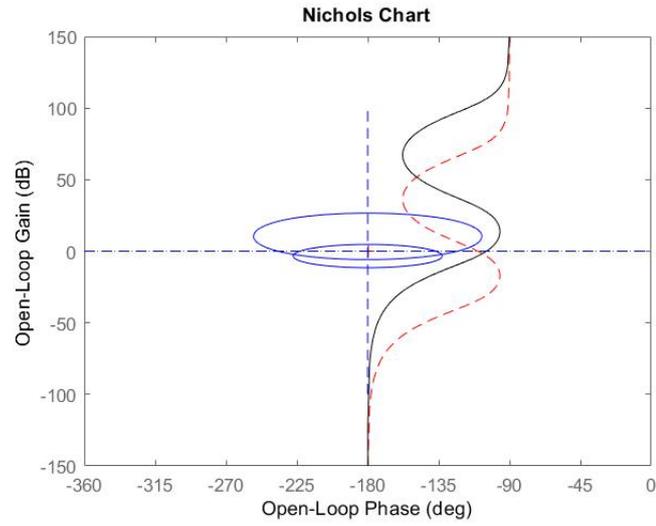


Figure 3.5: Increase of the crossover frequency on the Nichols Plot

3.2.5 Time Simulation

$$T(s) = \frac{12103(s + 0.6345)}{(s + 0.6415)(s^2 + 216s + 1.197 \cdot 10^4)} \quad (3.9)$$

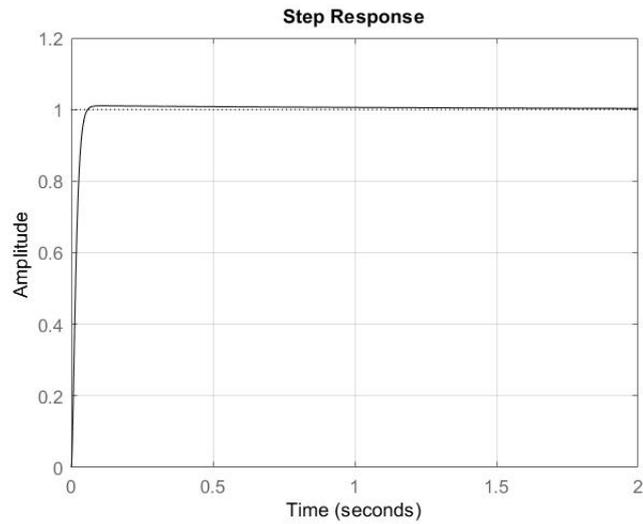


Figure 3.6: Time Response

The response shows that

$$\begin{cases} \hat{s} = 1.05\% \\ t_r = 0.0597s \end{cases} \quad (3.10)$$

The system is quite fast, but there is a minimum overshoot that has to be removed. These improvements will be done with the introduction of the external loop.

3.3 Position Control

3.3.1 Plant Definition

The open loop function (4.14) describes the internal loop. With the addition of an integral factor it shapes the transfer function of the new plant useful in the position controller.

$$G_p(s) = \frac{12103(s + 0.6345)}{s(s + 0.6415)(s^2 + 216s + 1.197e04)} \quad (3.11)$$

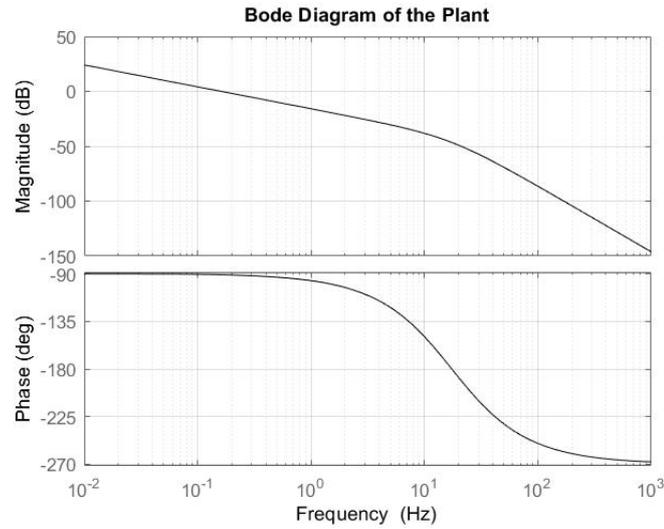


Figure 3.7: Bode Plot of $G_p(s)$

3.3.2 Position Specifications Handling

The conversion of the specifications from the time domain to the frequency one gives the same effects illustrated in the velocity control. There is only an important difference, regarding the number of poles at the origin of the controller. It derives from the manipulation of the constraint concerning the steady state output error.

$$2. |e_r^\infty| = |\lim_{t \rightarrow \infty} e_r(t)| =$$

$$\begin{aligned} & \text{By applying the } \textit{final value theorem} \\ & = |\lim_{s \rightarrow 0} s e_r(s)| = |\lim_{s \rightarrow 0} s K_d S(s) R(s)| = |\lim_{s \rightarrow 0} s s^{\nu+p} S^*(s) \frac{R0}{s^2}| = \\ & |\lim_{s \rightarrow 0} s s^p s^\nu S^*(s) \frac{R0}{s^2}| = |\lim_{s \rightarrow 0} s^2 s^\nu S^*(s) \frac{R0}{s^2}| = |\lim_{s \rightarrow 0} s^\nu S^*(s) R0| \end{aligned}$$

Since $|e_r^\infty|$ has to be **finite** and **different from zero**: $\nu \geq 0$.

Under this condition:

$$|e_r^\infty| = |K_d S^*(0) R0| \leq 0.0001 \implies |S^*(0)| \leq \frac{0.0001}{K_d R0} = 0.0001$$

3.3.3 Weighting Functions Definition

After different trials, $W_1(s)$ and $W_2(s)$ (3.12) seem to be the best option, since they allow to compute a BIBO stable controller whose shape remarks the one of a prototype of the second order

$$\begin{cases} W_1(s) = \frac{0.73472(s^2+0.5217s+0.1361)}{s(s+10^{-5})} \\ W_2(s) = \frac{(s+44)^2}{1.0487} \end{cases} \quad (3.12)$$

3.3.4 G_{c_p} Controller

The resulting controller is

$$G_c(s) = \frac{4959.1(s + 0.6415)(s + 0.3756)(s + 2.634 \cdot 10^{-5})(s^2 + 216s + 1.197 \cdot 10^4)}{(s + 1142)(s + 0.6345)(s + 0.3658)(s + 8.895 \cdot 10^{-6})(s^2 + 144.3s + 7129)} \quad (3.13)$$

The couple zero-pole at low frequency does not have a big impact on the time results.

$$G_c(s) = \frac{4959.1(s + 0.6415)(s + 0.3756)(s^2 + 216s + 1.197e04)}{(s + 1142)(s + 0.6345)(s + 0.3658)(s^2 + 144.3s + 7129)} \quad (3.14)$$

However, the best results are obtained by multiplying the controller for a constant $K_c = 2.2$.

$$G_c(s) = \frac{10910(s + 0.6415)(s + 0.3756)(s^2 + 216s + 1.197 \cdot 10^4)}{(s + 1142)(s + 0.6345)(s + 0.3658)(s^2 + 144.3s + 7129)} \quad (3.15)$$

3.3.5 Time Simulation

The closed loop function is given by

$$T(s) = \frac{1.3204 \cdot 10^8 (s + 0.3756)}{(s + 1142)(s + 50.84)(s + 0.3758)(s^2 + 93.53s + 2273)} \quad (3.16)$$

This results in

$$\begin{cases} \hat{s} = 0.054\% \\ t_r = 0.0848s \end{cases} \quad (3.17)$$

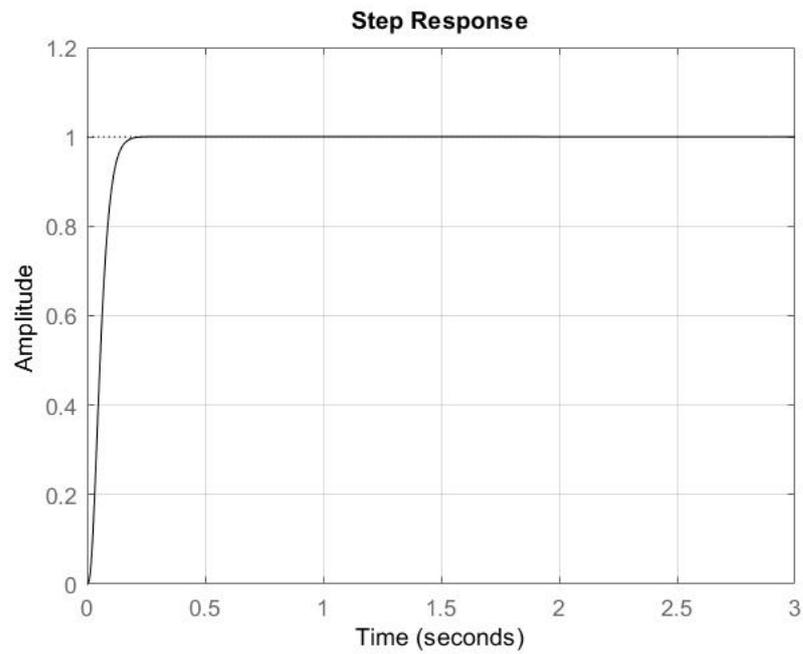


Figure 3.8: Time Response

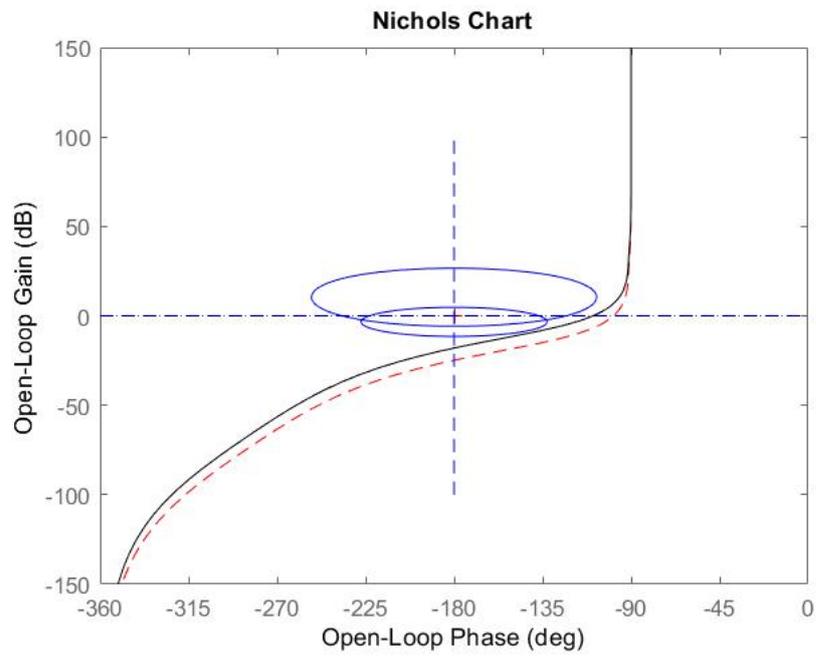


Figure 3.9: Nichols Plot of the Loop Functions derived from $G_c(s)$ and $G_{c_p}(s)$

Part III

Robust Control of a Comau Industrial Manipulator

Chapter 4

Motor and Link Control

The central focus of this chapter is the realization of **motor control** and **link control** of the first axis of a 6DOF Comau Robot.



Figure 4.1: A Comau Robot

The above types of control merge into a single complex and robust feedback structure:

The *motor control* is formed by two nested loops.

In the internal loop, the variable under control is the motor velocity. The plant expresses the relationship between u_m and \dot{q}_m .

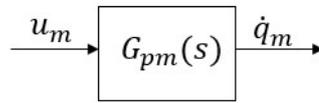


Figure 4.2: Motor Velocity Plant

In the external loop, the variable under control is the motor position q_m . In this case, the plant is defined by the closed loop function describing the internal feedback system with the addition of an integral factor.

The *link control* has a simpler configuration. It is only made of a transfer function expressing the relationship between the motor angle q_m and the link angle q_l .

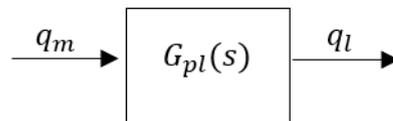


Figure 4.3: Motor - Link Plant

The complete scheme putting together **motor** and **link** control is depicted in Figure 4.4

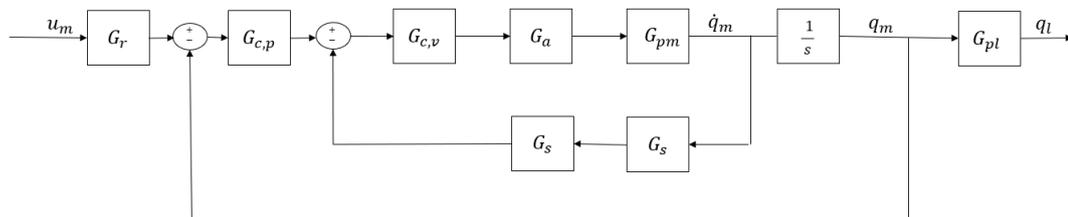


Figure 4.4: Motor - Link Control System

A deep explanation of each component of the proposed scheme will be provided in the next sections.

4.1 Control Requirements

The requirements useful to start the implementation of the velocity and position controls are

1. Steady-state Gain of the Feedback Control system: $K_d = 1$
2. Steady-state output error when the reference is a ramp ($R_0 = 1$):
 $|e_r^\infty| \leq 0.0001$
3. Step response overshoot $\hat{s} \leq 0.1$
4. Rise time: $t_r \leq 0.5s$
5. Settling time: $t_{s,5\%} \leq 1.5s$

4.2 Velocity Control

4.2.1 Dynamical Model Derivation

The differential equations that describe the manipulator under analysis derive from (2.11) and (2.12)

$$\begin{cases} \ddot{q}_m = \frac{1}{J_m}(u_m - f_m \dot{q}_m - \frac{T_l}{K_r}) \\ \ddot{q} = \frac{1}{J_l}(-f \dot{q} + T_l) \end{cases} \quad (4.1)$$

The first equation represents motor dynamics, the second one represents link dynamics.

The term T_l symbolizes the transmission. It is given by

$$T_l = K\left(\frac{q_m}{K_r} - q\right) + d\left(\frac{\dot{q}_m}{K_r} - \dot{q}\right) \quad (4.2)$$

Each parameter relative to motor, link and transmission has a precise physical meaning. The adopted numerical values are the ones characterizing a Comau industrial manipulator:

1. J_m represents the *motor inertia moment*: $J_m = 0.0136 \quad Kg \cdot m^2$
2. J_l represents the *link inertia moment*: $J_l = 463.214956910686 \quad Kg \cdot m^2$
3. f_m represents the *viscous friction in the motor*: $f_m = 0.0017482 \quad \left[\frac{Nms}{rad}\right]$
4. f represents the *viscous friction in the link*: $f = 0 \quad \left[\frac{Nms}{rad}\right]$
5. K represents the *gearbox stiffness coefficient*: $K = 6737983 \quad \left[\frac{Nm}{rad}\right]$
6. K_r represents the *gear ratio*: $K_r = 191.35$
7. d represents the *damping factor*: $d = 26951 \quad \left[\frac{Nms}{rad}\right]$

The Simulink scheme defining (4.1) and useful to compute the expression of the plant is illustrated in Figure 4.5

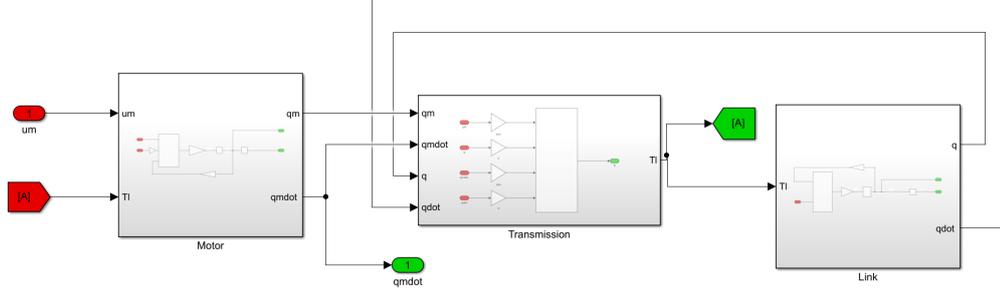


Figure 4.5: Simulink Scheme for the model of the robot

4.3 Plant Definition

The plant is described by a transfer function having a couple of complex zeros, a real pole and a couple of complex poles.

$$G_p(s) = \frac{73.529(s^2 + 58.18s + 1.455 \cdot 10^4)}{(s + 0.3794)(s^2 + 112.7s + 2.808 \cdot 10^4)} \quad (4.3)$$

The roots of the numerator of (4.3) are

$$s_{1,2} = -29.09 + -117.05i \quad (4.4)$$

The roots of the denominator of (4.3) are

$$\begin{cases} s_{1,2} = -56.18 + -157.86i \\ s_3 = -0.0666 \end{cases} \quad (4.5)$$

4.4 Velocity Specifications Handling

The control requisites need to be manipulated in order to obtain constraints on the sensitivity and complementary sensitivity functions, such as the maximum peak, the crossover frequency and the natural frequency.

It is important to synthesize the control objectives in $S(s)$ and $T(s)$ because they give the chance to shape the weighting functions that will influence the performances of the system in the time domain.

Each requirement needs to be analyzed, given $G_r = 1$, $G_a = Kt$, $G_s = \frac{1}{2pi}$

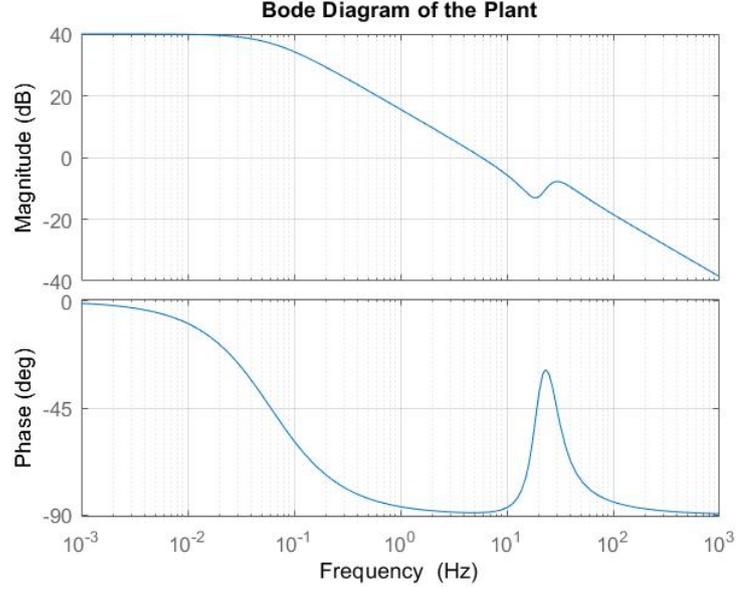


Figure 4.6: Bode Plot of the Plant

$$1. K_d = \lim_{s \rightarrow 0} G_{ry} = \lim_{s \rightarrow 0} \frac{G(s)}{1+G(s)H(s)}$$

From Figure 2.1: $G(s) = G_a G_p G_c$ and $H(s) = G_s G_f$.

$$\text{Since } (\nu + p) \geq 0 \implies K_d = \frac{1}{G_s G_f} \implies G_f = \frac{1}{K_d G_s}$$

$$2. |e_r^\infty| = |\lim_{t \rightarrow \infty} e_r(t)| =$$

By applying the *final value theorem*

$$= |\lim_{s \rightarrow 0} s e_r(s)| = |\lim_{s \rightarrow 0} s K_d S(s) R(s)| = |\lim_{s \rightarrow 0} s s^{\nu+p} S^*(s) \frac{R0}{s^2}| = |\lim_{s \rightarrow 0} s s^p s^\nu S^*(s) \frac{R0}{s^2}| = |\lim_{s \rightarrow 0} s s^\nu S^*(s) \frac{R0}{s}|$$

Since $|e_r^\infty|$ has to be **finite** and **different from zero**: $\nu \geq 1$.

Under this condition:

$$|e_r^\infty| = |K_d S^*(0) R0| \leq 0.0001 \implies |S^*(0)| \leq \frac{0.0001}{K_d R0} = 0.0001$$

$$3. \zeta \geq |\ln \hat{s}_0| \frac{\ln s_0}{\sqrt{\pi^2 + \ln^2 s_0}} = 0.5912$$

$$Tp \leq T_{p0} = \frac{1}{2\zeta \sqrt{1-\zeta^2}} = 1.0487 = 0.4127dB$$

$$Sp \leq S_{p0} = \frac{2\zeta \sqrt{2+4\zeta^2+2\sqrt{1+8\zeta^2}}}{\sqrt{1+8\zeta^2+4\zeta^2-1}} = 1.3611 = 2.6775dB$$

$$4. \omega_{n,1} = \frac{1}{t_r \sqrt{1-\zeta^2}} = 2.4797 \frac{rad}{s} = 0.39 Hz$$

$$\omega_{c,1} = \frac{1}{t_r \sqrt{1-\zeta^2}} \sqrt{\sqrt{1-4\zeta^4} - 2\zeta^2} = 1.79 \frac{rad}{s} = 0.28 Hz$$

$$5. \omega_{n,2} = \frac{-\log \alpha}{t_s \zeta} = 3.3784 \frac{rad}{s} = 0.54 Hz$$

$$\omega_{c,2} = \frac{-\log \alpha}{t_s \zeta} \sqrt{\sqrt{1-4\zeta^4} - 2\zeta^2} = 2.4388 \frac{rad}{s} = 0.39 Hz$$

$$\omega_n = \max(\omega_{n,1}, \omega_{n,2}) = 3.3784 \frac{rad}{s}, \omega_c = \max(\omega_{c,1}, \omega_{c,2}) = 2.4388 \frac{rad}{s}$$

4.5 Weighting Functions Definition

At this point, two weighting functions satisfying the conditions (1.18) can be written as:

$$W_S^{-1}(s) = \frac{1.3611s(s + 0.0003)}{(s^2 + 2.857s + 4.083)} \quad (4.6)$$

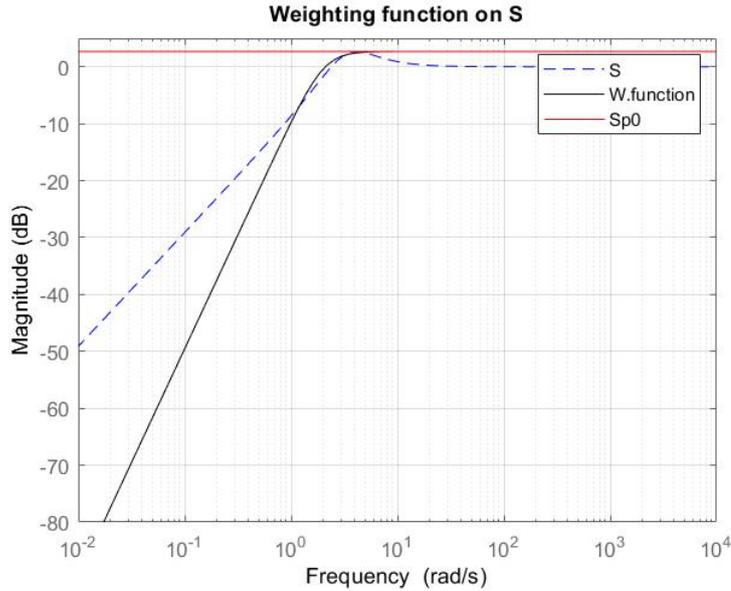


Figure 4.7: Weighting Function on the Sensitivity

$$W_T^{-1}(s) = \frac{235.95}{(s + 15)^2} \quad (4.7)$$

As it has been said in the introductory chapter, $W_S(s)$ and $W_T(s)$ are built for verifying the nominal performances. They suggest a starting point in the research

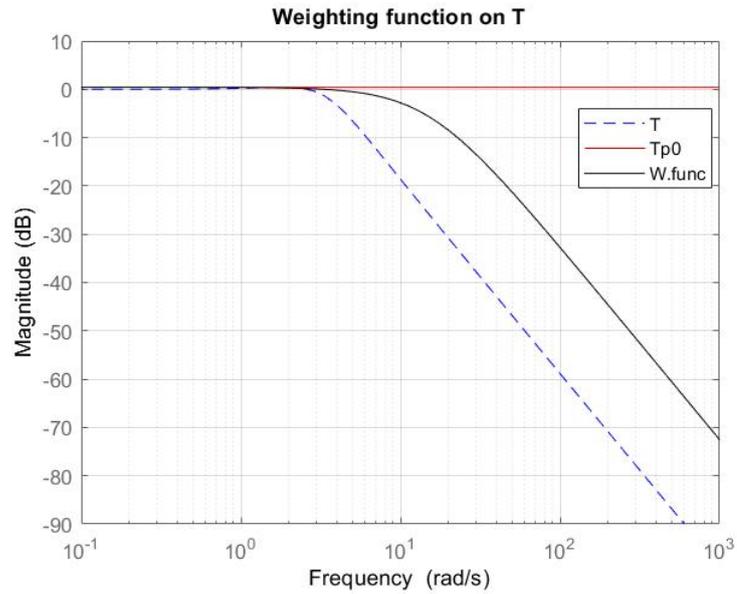


Figure 4.8: Weighting Function on the Complementary Sensitivity

of $W_1(s)$ and $W_2(s)$ that guarantee the desired system behavior. The final choice comes from a trial and error action. Depending on the complexity of the system, finding good expressions can also require some time. But, sometimes, the original expressions of $W_S(s)$ and $W_T(s)$ are good enough to satisfy the control purposes.

In this implementation, it is possible to impose:

$$\begin{cases} W_1(s) = W_S(s) \\ W_2(s) = W_T(s) \end{cases} \quad (4.8)$$

However, it is not possible to use $W_1(s)$ and $W_2(s)$ in this form, since they are not BIBO stable and not proper, respectively. To overcome this problem, two new functions are introduced. $W1_{mod}$ removes the pole at the origin with the introduction of a low frequency one defined by $0.01 \cdot \omega_c$. $W2_{mod}$ temporary removes the zeros that make W_2 not proper.

$$\begin{cases} W1_{mod} = \frac{0.73472(s^2+2.857s+4.083)}{(s+0.0003)(s+0.2439)} \\ W2_{mod} = \frac{1}{1.0487} \end{cases} \quad (4.9)$$

4.6 Gc_v Controller

The output computed by *hinflmi* toolbox is

$$G_c(s) = \frac{1324.6(s + 0.8279)(s + 0.3794)(s^2 + 112.6s + 2.808 \cdot 10^4)}{(s + 9474)(s + 0.2439)(s + 0.0002906)(s^2 + 58.18s + 1.455 \cdot 10^4)} \quad (4.10)$$

This expression has to be modified. Some changes in the form of the expression have to be applied. First of all, a pole at the origin needs to be placed, according to what it has been derived from the manipulation of the control requirements. For doing this, it is possible to move the low-frequency pole ($s + 0.0002906$). Then, in order to optimize its design, the pole in the high-frequency region ($s + 9474$) can be substituted by a lower one, i.e ($s + 800$), without altering the time outcomes.

$$G_c(s) = \frac{111.85(s + 0.8279)(s + 0.3794)(s^2 + 112.6s + 2.808 \cdot 10^4)}{s(s + 800)(s + 0.2439)(s^2 + 58.18s + 1.455 \cdot 10^4)} \quad (4.11)$$

The loop functions coming from the control system realized by means of $G_c(s)$ (4.10) and $G_c(s)$ (4.11) are plotted in a Nichols plane (Figure 4.10).

The introduction of a pole at the origin and of a lower pole at high frequency makes the shape of $L(s)$ similar to a second order prototype (continuous line).

The time results obtained with the definition of a control system by means of $G_c(s)$ are

$$\begin{cases} \hat{s} = 3.83\% \\ t_r = 0.285s \end{cases} \quad (4.12)$$

The control requirements are already met since $\hat{s} \leq 10\%$ and $t_r \leq 0.5s$. To further improve performances in terms of overshoot and rise time, the controller gain is incremented as follows:

$$\begin{aligned} Gc_v(s) &= KcGc(s) = 1.2Gc(s) \\ Gc_v(s) &= \frac{134.22(s + 0.8279)(s + 0.3794)(s^2 + 112.6s + 2.808 \cdot 10^4)}{s(s + 800)(s + 0.2439)(s^2 + 58.18s + 1.455 \cdot 10^4)} \end{aligned} \quad (4.13)$$

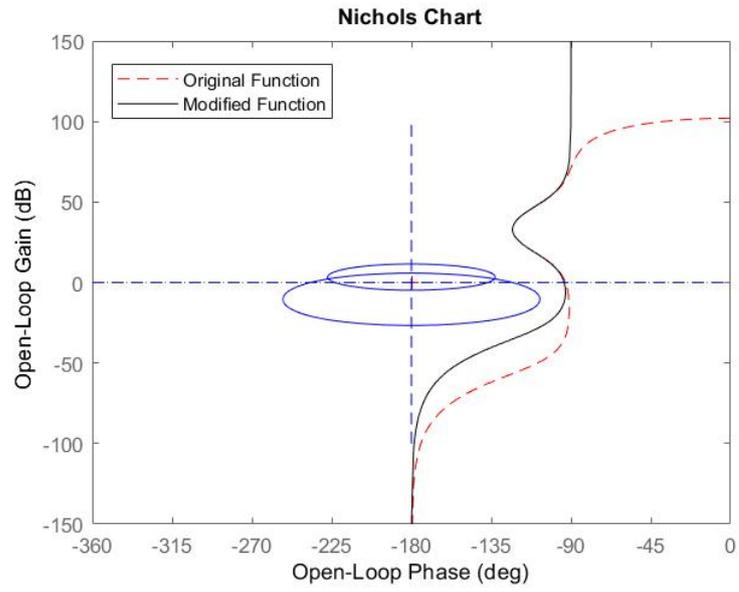


Figure 4.9: Loop functions derived from the original controller and the modified controller

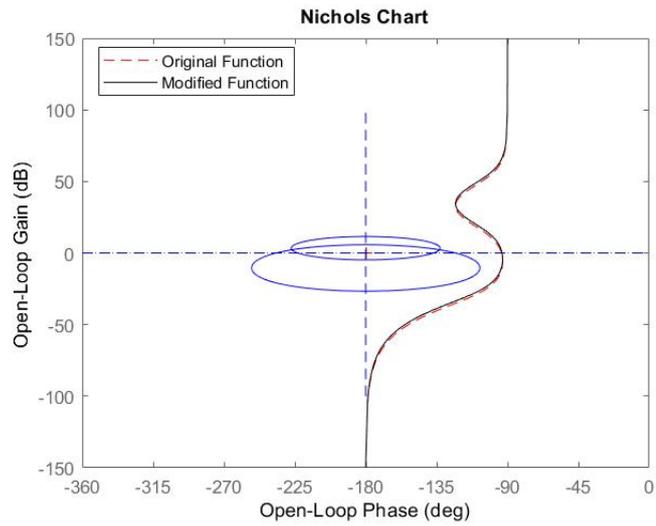


Figure 4.10: Loop functions derived from $G_c(s)$ and $G_{c_v}(s)$

4.7 Time Simulation

The closed loop function is

$$T(s) = \frac{L(s)}{1 + L(s)} = \frac{10398(s + 0.8279)}{(s + 786.8)(s + 12.58)(s + 0.8697)} \quad (4.14)$$

$L(s)$ is the open loop function:

$$L(s) = G_r(s)G_a(s)G_p(s)G_{c_v}(s)G_s(s)G_f(s)$$

$$L(s) = \frac{10398(s + 0.8279)}{s(s + 0.2439)(s + 800)} \quad (4.15)$$

$$\begin{cases} \hat{s} = 2.18\% \\ t_r = 0.17s \end{cases} \quad (4.16)$$

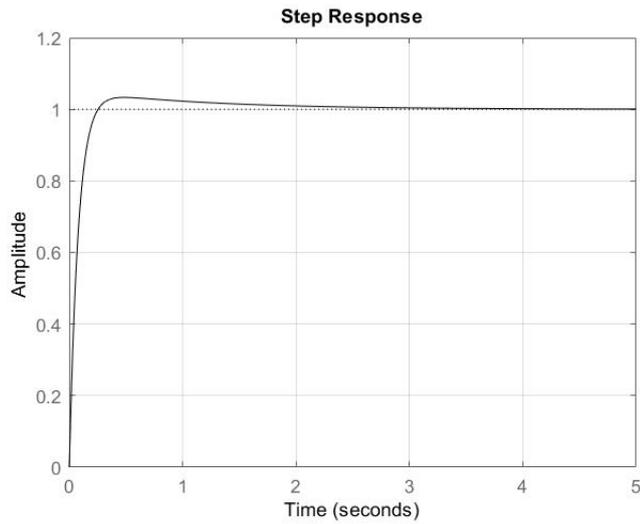


Figure 4.11: Loop functions derived from $G_c(s)$ and $G_{c_v}(s)$

As expected, the multiplication of the control function for K_c results into a reduction both in the overshoot and in the rise time.

4.8 Position Control

In position control, the control variable is the motor position.

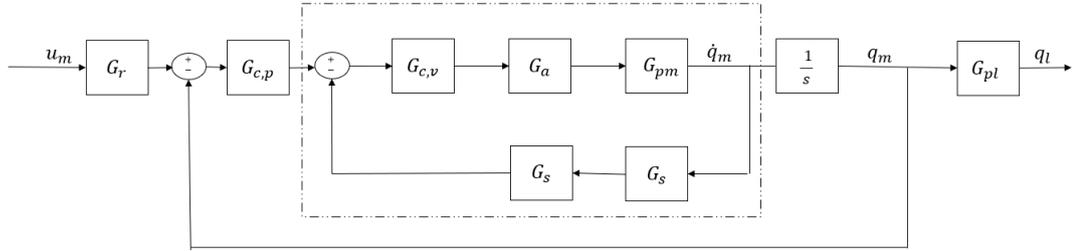


Figure 4.12: Motor-Link Control System

The dashed part in Figure 4.12 highlights the inner loop, i.e the closed loop function of the velocity controller, already computed. Starting from \dot{q}_m , q_m can be simply obtained by applying a integral action.

4.9 Plant Definition

The position plant has a simpler structure than the velocity one. It is only composed by real zeros and poles, while the complex values have been removed by the inner loop.

$$G_p(s) = \frac{T(s)}{s} \quad (4.17)$$

where $T(s)$ is described by (4.14)

$$G_p(s) = \frac{10398(s + 0.8279)}{s(s + 786.8)(s + 12.58)(s + 0.8697)} \quad (4.18)$$

4.10 Position Specifications Handling

From the analysis of the imposed requirements, it results that the position controller must not to have a pole at the origin. In fact, it is already present in the plant definition. However, this can also be derived from mathematical procedures applied to the constraint on the steady state output error:

2. $|e_r^\infty| = |\lim_{t \rightarrow \infty} e_r(t)| =$
By applying the *final value theorem*

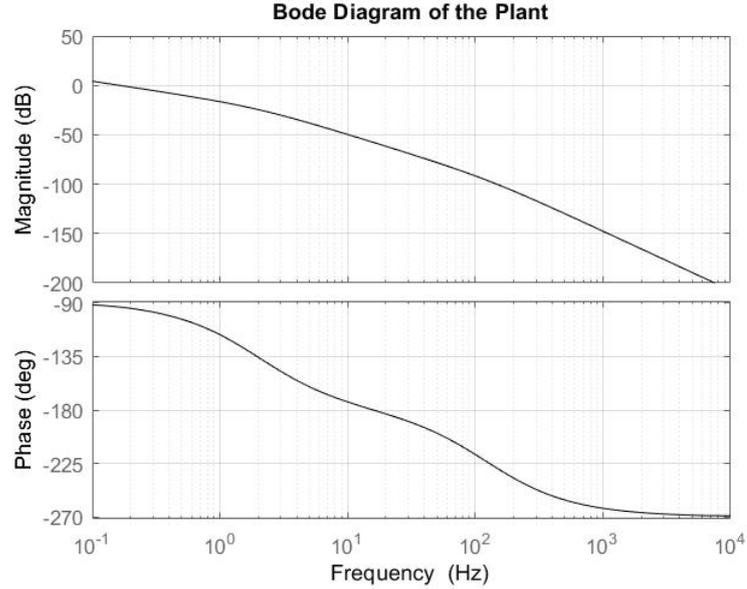


Figure 4.13: Bode Plot of the Plant

$$= |\lim_{s \rightarrow 0} s e_r(s)| = |\lim_{s \rightarrow 0} s K_d S(s) R(s)| = |\lim_{s \rightarrow 0} s s^{\nu+p} S^*(s) \frac{R0}{s^2}| =$$

$$|\lim_{s \rightarrow 0} s s^p s^{\nu} S^*(s) \frac{R0}{s^2}| = |\lim_{s \rightarrow 0} s^2 s^{\nu} S^*(s) \frac{R0}{s^2}| = |\lim_{s \rightarrow 0} s^{\nu} S^*(s) R0|$$

Since $|e_r^\infty|$ has to be **finite** and **different from zero**: $\nu \geq 0$.

Under this condition:

$$|e_r^\infty| = |K_d S^*(0) R0| \leq 0.0001 \implies |S^*(0)| \leq \frac{0.0001}{K_d R0} = 0.0001$$

In addition, it is important to state that, in this case, the control requirements are only useful to build $S(s)$ and $T(s)$ and to start the implementation problem.

Actually, the goal is to set up a position control able to reach better results with respect to the velocity one. In particular, it has to guarantee the presence of a very fast response, with a rise time whose value must be lower than 200ms, and a null overshoot.

4.11 Weighting Functions Definition

In contrast with the previous control, the original weighting functions are not going to produce acceptable results. In this case, $W_1(s)$ and $W_2(s)$ can not be a copy of $W_S(s)$ and $W_T(s)$, but they have to be modified:

1. In $W_1(s)$ weighting function, a lower ω_1 frequency is chosen

2. In $W_2(s)$ weighting function, a greater ω_3 frequency is chosen.

$$\begin{cases} W_1(s) = \frac{0.73472(s^2+1.166s+0.6805)}{s(s+5 \cdot 10^{-5})} \\ W_2(s) = \frac{(s+55)^2}{1.0487} \end{cases} \quad (4.19)$$

4.12 Gc_p Controller

The best controller computed by means of the *hinflmi* Matlab toolbox is

$$Gc(s) = \frac{1.079 \cdot 10^6 (s + 786.8)(s + 12.58)(s + 0.8697)(s + 0.3299)(s + 0.0005211)}{(s + 1.19 \cdot 10^4)(s + 510.1)(s + 136.9)(s + 0.8279)(s + 0.2439)(s + 4.28 \cdot 10^{-5})} \quad (4.20)$$

The couple zero-pole at low frequency $\frac{(s+0.0005211)}{(s+4.281 \cdot 10^{-5})}$ can be removed and the pole at high frequency $(s + 1.192 \cdot 10^4)$ can be substituted by a lower one.

$$Gc(s) = \frac{81438(s + 786.8)(s + 12.58)(s + 0.8697)(s + 0.3299)}{(s + 510.1)(s + 900)(s + 136.9)(s + 0.8279)(s + 0.2439)} \quad (4.21)$$

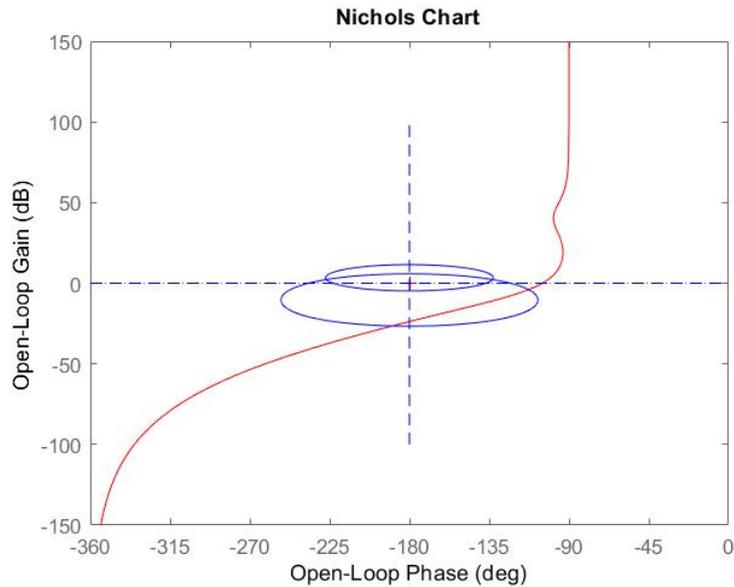


Figure 4.14: Loop functions derived from $Gc(s)$

The results obtained starting from (4.21) can not be considered satisfactory, since

$$\begin{cases} \hat{s} = 0.52\% \\ t_r = 0.129s \end{cases} \quad (4.22)$$

These values can be acceptable in the case of velocity control. As it has been said at the beginning of the description implementation under analysis, position control has to ensure a step response that gets the steady state value in a short time, without oscillations or overshoot.

To make this possible, the controller gain is increased as follows:

$$G_{c_p}(s) = K_c G_c(s) = 1.8G_c(s).$$

$$G_{c_p}(s) = \frac{1.4659 \cdot 10^5 (s + 786.8)(s + 12.58)(s + 0.8697)(s + 0.3299)}{(s + 510.1)(s + 900)(s + 136.9)(s + 0.8279)(s + 0.2439)} \quad (4.23)$$

$K_c = 1.8$ is the maximum gain applicable to the controller in order to improve the results. It has been noticed that, by using a greater constant, the time response begins to oscillate and the overshoot begins to show up.

The consequence of applying (4.23) is an upward shift of the loop function and an increment of the crossover frequency, that results in a faster system. As showed

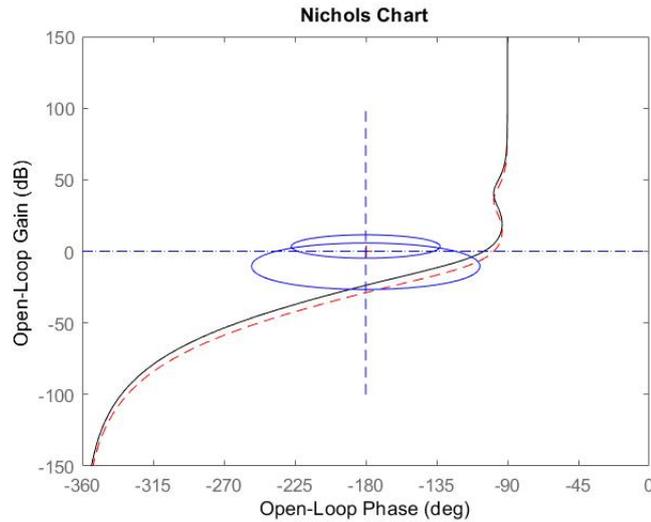


Figure 4.15: Loop functions derived from $G_c(s)$ and $G_{c_p}(s)$

by Figure 4.15, the loop functions $L(s)$ remain inside the constant magnitude loci. In a simple control situation, in order to guarantee stability of the system, it is preferred to make $L(s)$ tangent to the circumferences, at least. This is not the case, as the stability and robustness of the system is strengthened by the presence of two feedback loops.

4.13 Time Simulation

By computing the closed loop function, it is possible to verify the improvements gained.

$$T(s) = \frac{L(s)}{1 + L(s)} = \frac{1.6059 \cdot 10^9}{(s + 893.8)(s + 530.9)(s + 80.02)(s + 42.15)} \quad (4.24)$$

$L(s)$ is the open loop function:

$$L(s) = G_r(s)G_a(s)G_c(s)G_p(s)G_s(s)G_f(s)$$

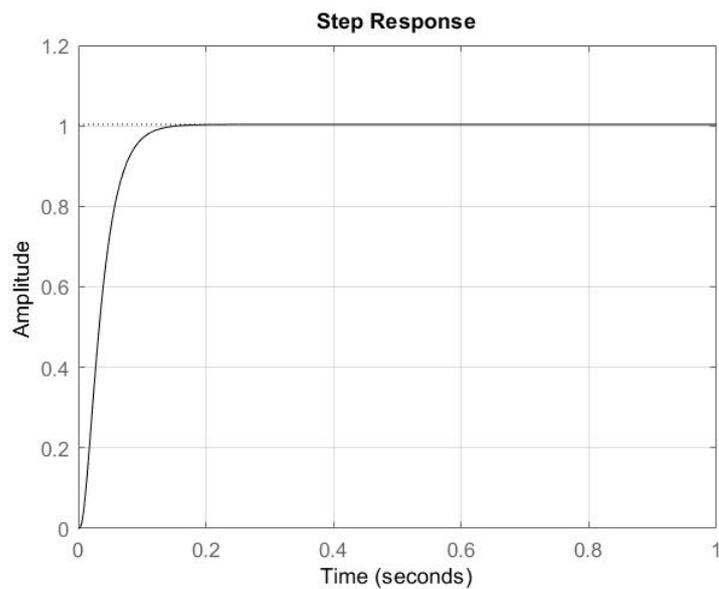


Figure 4.16: Time Response

From the time Response in Figure 4.16, it results:

$$\begin{cases} \hat{s} = 0\% \\ t_r = 0.0626s \end{cases} \quad (4.25)$$

4.14 Link Control

4.14.1 Plant Definition

To analyze the link performances, a new transfer function between the q_m parameter and the q_l one has been calculated, as illustrated at the beginning of this Chapter.

$$G_{pl}(s) = \frac{0.30406(s + 250)}{(s^2 + 58.18s + 1.455 \cdot 10^4)} \quad (4.26)$$

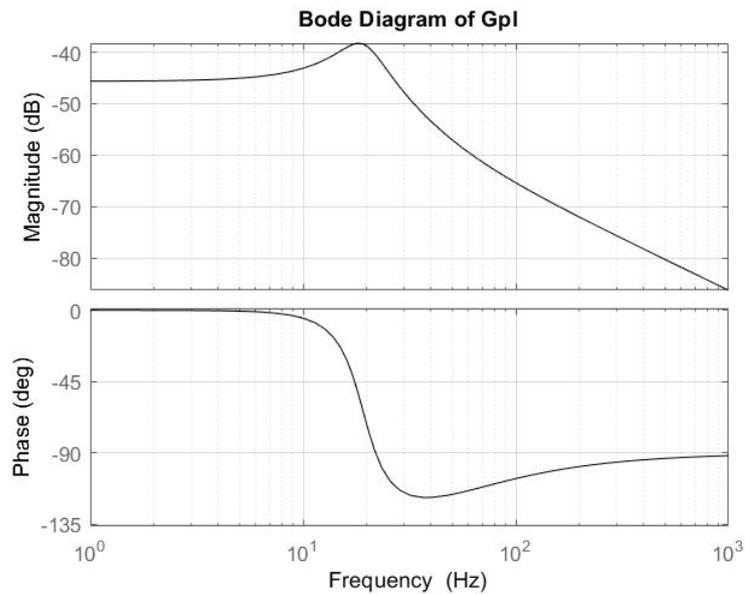


Figure 4.17: Bode Plot of $G_{pl}(s)$

The starting point for deriving G_{pl} is the manipulation of the differential equations (4.1) describing the dynamics of the manipulator. Only the link equation and the transmission one are useful. The scheme is sketched in Figure 4.18

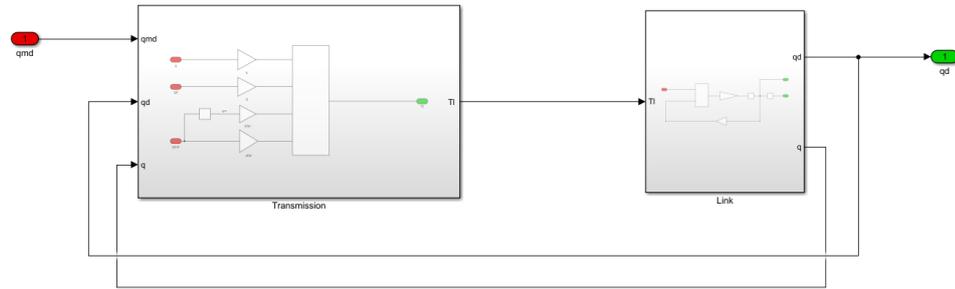


Figure 4.18: Simulink Model for the derivation of the link transfer function

The plant function has been computed through the derivation of the state equations and the state matrices, at first. In a second time, the Simulink scheme has been built in order to check the accuracy of the realized plant form.

4.14.2 Time Simulation

The examination of the motor and link performances has been carried out via Simulink. The scheme in Figure 4.19 illustrates in a more compact and schematic way the control structure reported in (4.4), where all the contributes are exploited. In fact, in Figure 4.19, the velocity control system is hidden inside the "Plant" box, in order to improve the readability of the overall scheme.

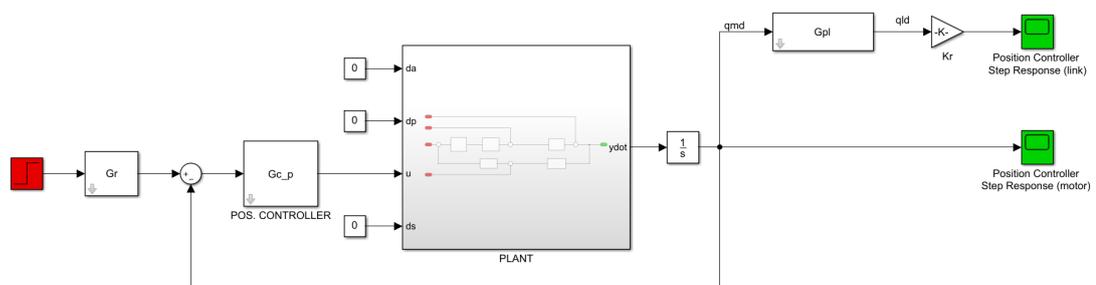


Figure 4.19: Simulink Model for the description of the motor control and the link control

By focusing only on link control, the achieved performances demonstrate a good behavior of the time response: the system arrives at steady state in 0.06s and does not present any overshoot (Figure 4.20).

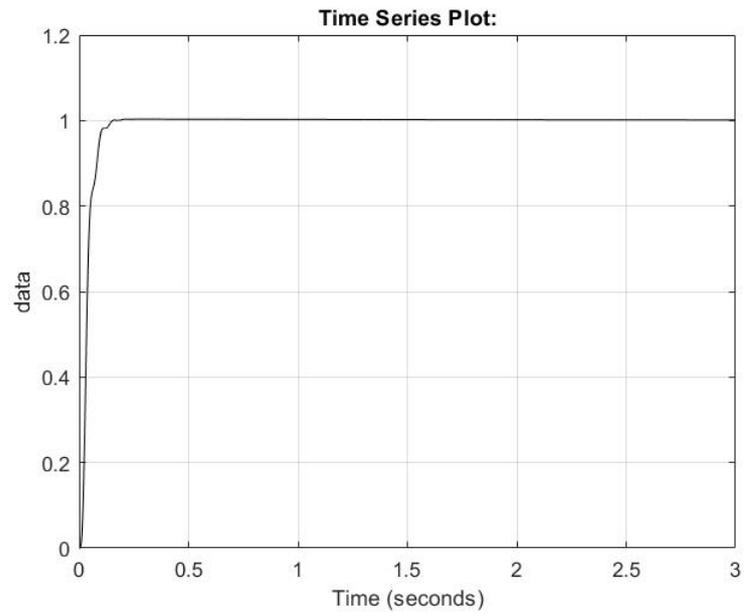


Figure 4.20: Time Link Performances

In general, the complete complex control structure plenty reaches the desired characteristics, by fulfilling all the control imposed objectives.

Chapter 5

Parameters Variation

After the derivation of two controllers $G_{c_v}(s)$ and $G_{c_p}(s)$, it is important to study how much robust they result when the parameters characterizing the dynamics of the model vary in a range of uncertainty. It is known, in fact, that the numerical values chosen to simulate the system and to study its behavior can change according to the position and the orientation of the industrial robot.

So, maintaining the controllers [(4.13) & (4.23)] immutable and computing different expressions for the plants depending on the parameters values, it is possible to understand in which conditions the performances of the nested control system degrade.

5.1 Single Parameters Variation

A first analysis has been carried out by supposing that only a unique parameter changes. This is not a realistic case, in which, on the contrary, more parameters are simultaneously subject to a variation.

The obtained results in terms of overshoot for each studied case are grouped in Table 5.1, in order to facilitate their comparison.

The most critical situation appears in correspondence of K_r , where the maximum value of the overshoot is equal to 1.42%.

As can be seen from Table 5.1, K_r varies in a range between $K_{r,min}$ and $K_{r,max}$. Since the nominal value for K_r is 191.35

$$172.2150 \leq K_r \leq 210.4850$$

By reporting the Nichols plot of the *position open loop functions* $L(s)$ computed for each K_r value, it is possible to observe that they move inside the constant magnitude loci (Figure 5.1).

Range of Variation	Overshoot
$J_m - 5\% \cdot J_m \leq J \leq J_m + 5\% \cdot J_m$	$0\% \leq \hat{s} \leq 0.202\%$
$J_l - 5\% \cdot J_l \leq J \leq J_l + 5\% \cdot J_l$	$0\% \leq \hat{s} \leq 0.208\%$
$K_t - 5\% \cdot K_t \leq K_t \leq K_t + 5\% \cdot K_t$	$\hat{s} = 0\%$
$d - 30\% \cdot d \leq d \leq d + 30\% \cdot d$	$\hat{s} = 0\%$
$f_m - 10\% \cdot f_m \leq f_m \leq f_m + 10\% \cdot f_m$	$\hat{s} = 0\%$
$0.05 \leq f \leq 0.1$	$\hat{s} = 0\%$
$K_t - 30\% \cdot K_t \leq K_t \leq K_t + 30\% \cdot K_t$	$0\% \leq \hat{s} \leq 0.0094\%$
$K_r - 10\% \cdot K_r \leq K_r \leq K_r + 10\% \cdot K_r$	$0\% \leq \hat{s} \leq 1.42\%$

Table 5.1: Results from parameters variation

As explained in Chapter 4, in front of nominal parameters that do not vary in time, this behavior does not impact on the final time results, since the robustness and the stability of the system is covered by the nested loops. The actual situation is quite different, because the central point is a plant that changes at each step. To overcome this issue and to define a more powerful structure, a controller that allows to compute open loop functions tangent to the circumferences, at least, has been shaped.

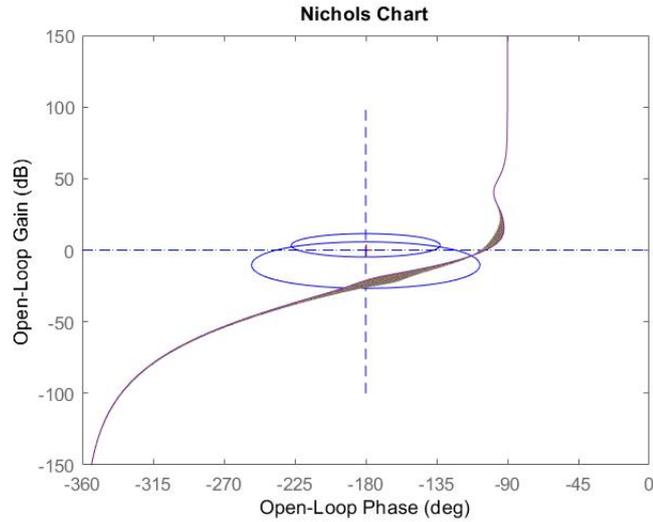


Figure 5.1: Nichols Plot

Moreover, the step response shows how the peak response decreases when K_r moves from the maximum value $K_{r,max}$ to the minimum one $K_{r,min}$.

$$\begin{cases} \hat{s} = 1.42\% & \text{for } K_r = K_{r,min} \\ \hat{s} = 0\% & \text{for } K_r = K_{r,max} \end{cases} \quad (5.1)$$

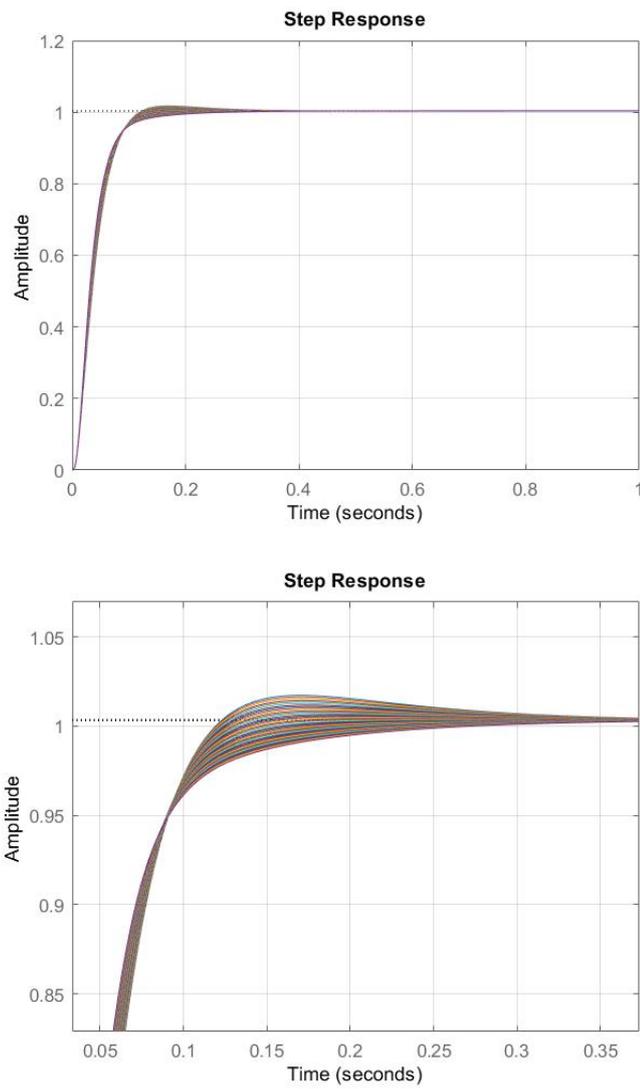


Figure 5.2: Time Response

5.2 A Robust Choice of G_{c_v} and G_{c_p}

In the development process of new velocity and position controllers that ensure good time performances for all the variations of all the parameters, an important conclusion has been made.

The optimal solution has to take into account two different aspects: safety, described by means of the overshoot, and velocity, described by means of the rise time value.

All the control structures implemented in the thesis project aim at building a system that includes the satisfaction of both the conditions. In front of such setting, where the parameters are not fixed, it has not been possible to meet both the requirements at the same time.

For this reason, two separate solutions will be presented:

In the **first solution**, the *velocity and position controllers* allow to reach robust results in terms of **overshoot** for all the parameters variations.

In the **second solution**, the *velocity and position controllers* allow to get a low value of the rise time for all the parameters variations.

5.2.1 Single Parameters Variation

The first solution uses the velocity and position controllers reported in (5.2) and (5.3)

I: Overshoot Solution

$$G_{c_v}(s) = \frac{75.217(s + 0.3794)(s + 0.3536)(s^2 + 112.6s + 2.808 \cdot 10^4)}{s(s + 500)(s + 0.2439)(s^2 + 58.18s + 1.455 \cdot 10^4)} \quad (5.2)$$

$$G_{c_p}(s) = \frac{8081(s + 488.1)(s + 11.82)(s + 0.3571)(s + 0.2663)}{(s + 170.6)(s + 0.3536)(s + 0.2429)(s^2 + 231.5s + 5.418 \cdot 10^4)} \quad (5.3)$$

Both $G_{c_v}(s)$ and $G_{c_p}(s)$ have been derived by means of the H_∞ approach and the definition of $W_1(s)$ and $W_2(s)$ weighting functions.

In particular, in the case of velocity controller:

$$\begin{cases} W_1(s) = \frac{0.73472(s^2 + 1.166s + 0.6805)}{s(s + 5 \cdot 10^{-5})} \\ W_2(s) = \frac{(s + 15)^2}{1.0487} \end{cases} \quad (5.4)$$

In order to obtain a faster response, the controller obtained from the Matlab H_∞ toolbox has been multiplied for a constant $K_c = 1.2$.

In the case of the position controller, the weighting functions $W_1(s)$ and $W_2(s)$ have been derived starting from the frequencies:

$$\begin{cases} \omega_1 = 0.00002 \frac{rad}{s} \\ \omega_3 = 15 \frac{rad}{s} \end{cases} \implies \omega_2 = 0.5217 \frac{rad}{s} \quad (5.5)$$

The controller obtained from the Matlab H_∞ toolbox has been multiplied for a constant $K_c = 0.4$. This allows to apply a downshift of the open loop function in the Nichols plane and to place it outside the two constant magnitude loci.

Range of Variation	Overshoot	Rise Time
$J_m - 5\% \cdot J_m \leq J \leq J_m + 5\% \cdot J_m$	$\hat{s} = 0\%$	$t_r = 0.385s : 0.392s$
$J_l - 5\% \cdot J_l \leq J \leq J_l + 5\% \cdot J_l$	$\hat{s} = 0$	$t_r = 0.385s : 0.392s$
$K_t - 5\% \cdot K_t \leq K_t \leq K_t + 5\% \cdot K_t$	$\hat{s} = 0\%$	$t_r = 0.388s$
$d - 30\% \cdot d \leq d \leq d + 30\% \cdot d$	$\hat{s} = 0\%$	$t_r = 0.388s : 0.389s$
$f_m - 10\% \cdot f_m \leq f_m \leq f_m + 10\% \cdot f_m$	$\hat{s} = 0\%$	$t_r = 0.386s : 0.39s$
$0.05 \leq f \leq 0.1$	$\hat{s} = 0\%$	$t_r = 0.388s$
$K_t - 30\% \cdot K_t \leq K_t \leq K_t + 30\% \cdot K_t$	$\hat{s} = 0\%$	$t_r = 0.388s$
$K_r - 10\% \cdot K_r \leq K_r \leq K_r + 10\% \cdot K_r$	$\hat{s} = 0\%$	$t_r = 0.374s : 0.4s$

Table 5.2: Results from the I Solution

As showed by Table 5.2, the response is slow, but the overshoot remains always null, even in the critical state previously occurred because of the variation of K_r .

II: Rise-Time Solution

The second solution uses the velocity and position controllers reported in (5.6) and (5.7)

$$G_{c_v}(s) = \frac{151(s + 0.8279)(s + 0.3794)(s^2 + 112.6s + 2.808 \cdot 10^4)}{s(s + 900)(s + 0.2439)(s^2 + 58.18s + 1.455 \cdot 10^4)} \quad (5.6)$$

$$G_{c_p}(s) = \frac{5.5517 \cdot 10^5(s + 886.8)(s + 145.6)(s + 12.56)(s + 0.2663)}{(s + 170.6)(s + 0.3536)(s + 0.2429)(s^2 + 231.5s + 5.418 \cdot 10^4)} \quad (5.7)$$

$G_{c_v}(s)$ has been obtained by means of the H_∞ algorithm, while $G_{c_p}(s)$ has been obtained by applying a lead compensator to the output of the *hinflmi* toolbox.

A lead compensator, with the introduction of a dominant zero and a pole to the controller transfer function, is useful to increase the phase angle margin at low frequency and to improve the dynamic performance of the system [8].

Range of Variation	Overshoot	Rise Time
$J_m - 5\% \cdot J_m \leq J \leq J_m + 5\% \cdot J_m$	$\hat{s} = 0\%$	$t_r = 0.189s : 0.193s$
$J_l - 5\% \cdot J_l \leq J \leq J_l + 5\% \cdot J_l$	$\hat{s} = 0$	$t_r = 0.19s : 0.193s$
$K_t - 5\% \cdot K_t \leq K_t \leq K_t + 5\% \cdot K_t$	$\hat{s} = 0\%$	$t_r = 0.191s$
$d - 30\% \cdot d \leq d \leq d + 30\% \cdot d$	$\hat{s} = 0\%$	$t_r = 0.191s$
$f_m - 10\% \cdot f_m \leq f_m \leq f_m + 10\% \cdot f_m$	$\hat{s} = 0\%$	$t_r = 0.19 : 0.192s$
$0.05 \leq f \leq 0.1$	$\hat{s} = 0\%$	$t_r = 0.191s$
$K_t - 30\% \cdot K_t \leq K_t \leq K_t + 30\% \cdot K_t$	$\hat{s} = 0\%$	$t_r = 0.191s$
$K_r - 10\% \cdot K_r \leq K_r \leq K_r + 10\% \cdot K_r$	$\hat{s} = 0\% : 0.119\%$	$t_r = 0.185s : 0.197s$

Table 5.3: Results from the II Solution

In this case, the overshoot remains equal to zero except that in the case of the variation of the K_r parameter, while the rise time assumes lower values than with the first proposed solution.

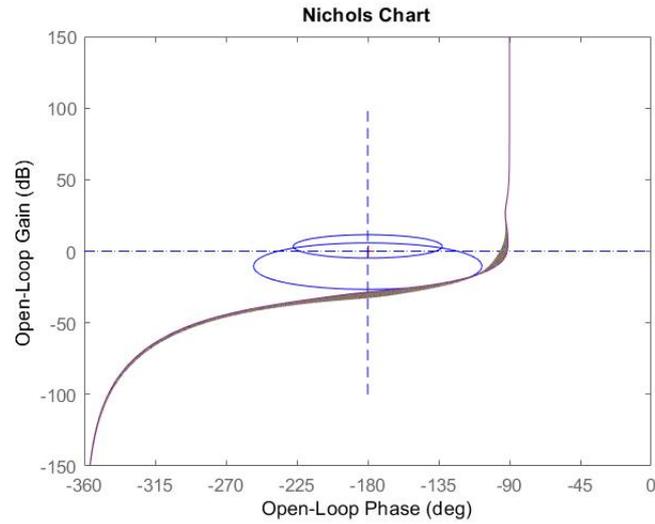


Figure 5.3: Nichols plot of the Open Loop functions computed for the variation K_r parameter (I Solution)

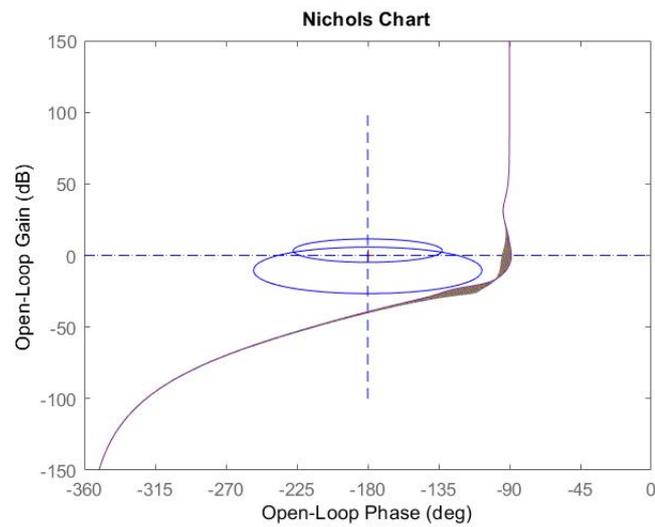


Figure 5.4: Nichols plot of the Open Loop functions computed for the variation K_r parameter (II Solution)

From Figure 5.4 it is possible to see that the controllers give the chance to describe loop functions that always lie outside the two constant magnitude loci.

5.2.2 Multi-Parameters Variation

In a realistic situation, it is impossible to have a unique parameter varying and the remaining ones unchanged.

For this reason, it is worth notice to check how the introduced couple of controllers behaves when many parameters change at the same time.

Case I: Variation of K_t and K_r

In order to check if improvements have been introduced with the proposed velocity and position solutions, the variation of K_t , that is not a critical one, and K_r is considered. Four different situations have been simulated, to cover all the possible cases: initially, both the parameters move from their minimum values to their maximum ones or from their maximum values to their minimum ones, then the parameters vary in an opposite direction.

$$\begin{cases} K_{t,min} \leq K_t \leq K_{t,max} \\ K_{r,min} \leq K_r \leq K_{r,max} \end{cases} \quad (5.8)$$

I Solution	II Solution
$\hat{s} = 0\%$	$\hat{s} = 0.119\% : 0\%$
$t_r = 0.374s : 0.4s$	$t_r = 0.18s : 0.197s$

$$\begin{cases} K_{t,min} \leq K_t \leq K_{t,max} \\ K_{r,max} \leq K_r \leq K_{r,min} \end{cases} \quad (5.9)$$

I Solution	II Solution
$\hat{s} = 0\%$	$\hat{s} = 0.119\% : 0\%$
$t_r = 0.4s : 0,374s$	$t_r = 0.197s : 0.18s$

$$\begin{cases} K_{t,max} \leq K_t \leq K_{t,min} \\ K_{r,min} \leq K_r \leq K_{r,max} \end{cases} \quad (5.10)$$

I Solution	II Solution
$\hat{s} = 0\%$	$\hat{s} = 0\% : 0.119\%$
$t_r = 0.374s : 0.4s$	$t_r = 0.18s : 0.197s$

$$\begin{cases} K_{t,max} \leq K_t \leq K_{t,min} \\ K_{r,max} \leq K_r \leq K_{r,min} \end{cases} \quad (5.11)$$

I Solution	II Solution
$\hat{s} = 0\%$	$\hat{s} = 0\% : 0.119\%$
$t_r = 0.4s : 0.374s$	$t_r = 0.197s : 0.18s$

For all the simulated conditions, the overshoot value remains always equal to zero.

Case II: Variation of K_t , K_r , f_m and f

A more complex situation has been performed. Four parameters vary at the same moment. Also in this case, the presence of K_r is necessary to check how much the controllers can be considered high performing.

$$\begin{cases} K_{t,min} \leq K_t \leq K_{t,max} \\ K_{r,min} \leq K_r \leq K_{r,max} \\ f_{m,max} \leq f_m \leq f_{m,min} \\ f_{max} \leq f \leq f_{min} \end{cases} \quad (5.12)$$

I Solution	II Solution
$\hat{s} = 0\%$	$\hat{s} = 0.106\% : 0\%$
$t_r = 0.376s : 0.398s$	$t_r = 0.186s : 0.197s$

$$\begin{cases} K_{t,max} \leq K_t \leq K_{t,min} \\ K_{r,min} \leq K_r \leq K_{r,max} \\ f_{m,max} \leq f_m \leq f_{m,min} \\ f_{min} \leq f \leq f_{max} \end{cases} \quad (5.13)$$

I Solution	II Solution
$\hat{s} = 0\%$	$\hat{s} = 0.106\% : 0\%$
$t_r = 0.376s : 0.398s$	$t_r = 0.186s : 0.197s$

By comparing these results, it is possible to state that the intention to find *velocity and position controller* robustly stable in a more realistic situation and not only in an ideal one has been fulfilled.

Chapter 6

Changing Controller

The link inertia changes depending on the movements performed by the robot.

For each J_l value, good time performances have to be reached. The two parameters that reflect the behavior of robot are the *overshoot* and the *rise time*. Acceptable situations are met when \hat{s} is close to zero and t_r is small ($t_r \leq 200ms$).

In order to get similar time outcomes whatever the J_l value, a unique controller is not good enough.

Purpose of this Chapter is to find a standard form that allows to shape automatically a controller that can fulfill the time objectives on \hat{s} and t_r starting from random values of the link inertia.

Also in this case, the control problem is divided into two smaller ones. The analysis is focused mostly on the *velocity control* as it is directly affected by the manipulator dynamics. The *position control* is a consequence of the first one and tries to further improve its results.

6.1 Velocity Controllers

By making use of the H_∞ approach, three controllers starting from realistic values of J_l have been computed.

$$J_l = [122.082355609927, 463.214956910686, 1647.77302847327] Kg \cdot m^2$$

The chosen values of J_l correspond to three different positions that the robot can assume.

$$\begin{cases} J_l = J_{l,min} = 122.082355609927 \implies q = [0^\circ, 0^\circ, 0^\circ, 0^\circ, 0^\circ, 0^\circ] \\ J_l = J_{l,med} = 463.214956910686 \implies q = [0^\circ, 0^\circ, 90^\circ, 0^\circ, 0^\circ, 0^\circ] \\ J_l = J_{l,max} = 1647.77302847327 \implies q = [0^\circ, 90^\circ, 0^\circ, 0^\circ, 0^\circ, 0^\circ] \end{cases} \quad (6.1)$$

In particular

$$G_{cJlmin}(s) = \frac{475.5(s + 0.396)(s + 0.5882)(s^2 + 275s + 6.872 \cdot 10^4)}{s(s + 800)(s + 0.2439)(s^2 + 220.8s + 5.519 \cdot 10^4)} \quad (6.2)$$

$$G_{cJlmed}(s) = \frac{474.2(s + 0.3969)(s + 0.3794)(s^2 + 112.7s + 2.808 \cdot 10^4)}{s(s + 800)(s + 0.2439)(s^2 + 58.18s + 1.455 \cdot 10^4)} \quad (6.3)$$

$$G_{cJlmax}(s) = \frac{474.91(s + 0.4015)(s + 0.17)(s^2 + 71.04s + 1.762 \cdot 10^4)}{s(s + 800)(s + 0.2439)(s^2 + 16.36s + 4089)} \quad (6.4)$$

For each controller, the open loop function $L(s)$ and the closed loop function $T(s)$ have been computed, in order to study the performances.

The three systems report the same satisfactory results, since

$$\forall JI = [J_{l,min}, J_{l,med}, J_{l,max}]:$$

$$\begin{cases} \hat{s} = 0\% \\ t_r = 0.0451s \end{cases} \quad (6.5)$$

This is the best situation that can occur starting from the chosen values of the link inertia.

An analysis of the similarities and the differences between the computed controllers, both in the form and in the numerical values corresponding to the zeros and the poles, has carried out to a general definition.

$$G_c(s) = K \frac{(s - z_{fix})}{s(s - p_{1,fix})(s - p_{2,fix})} G_p^{-1}(s) \quad (6.6)$$

$G_c(s)$ is composed by a **fixed** part, a **variable** part, and also by the **plant**.

The fixed part consists of

1. a pole at the origin that comes from the mathematical manipulation of the specifications
2. a zero placed at medium frequency, ($z_{fix} = -0.4$)
3. a pole placed at high frequency ($p_{1,fix} = 800$)
4. a pole placed at medium frequency coming from the weighting function on the sensitivity ($p_{2,fix} = 0.2439$)

The variable part consists of the gain controller K . For each input J_l , it is computed via a lookup table, starting from the known values that it assumes in correspondence of J_{lmin} , J_{lmed} , J_{lmax} .

The most interesting part is the one that involves the inverse function of the plant.

$$G_{p_{J_{lmin}}} = \frac{73.529(s^2 + 220.8s + 5.519 \cdot 10^4)(s^2 + 275s + 6.872 \cdot 10^4)}{(s + 0.5882)(s^2 + 275s + 6.872 \cdot 10^4)} \quad (6.7)$$

$$G_{p_{J_{lmed}}} = \frac{73.529(s^2 + 58.18s + 1.455 \cdot 10^4)}{(s + 0.3794)(s^2 + 112.7s + 2.808 \cdot 10^4)} \quad (6.8)$$

$$G_{p_{J_{lmax}}} = \frac{73.529(s^2 + 16.36s + 4089)}{(s + 0.17)(s^2 + 71.04s + 1.762 \cdot 10^4)} \quad (6.9)$$

It has been noticed that the poles of $G_p(s)$ becomes part of the zeros of $G_c(s)$ and vice versa.

By following this implementation, the obtained results are satisfactory.

The Nichols plot of the loop function (Figure 6.1) is a second order prototype tangent to the two constant magnitude loci, while the time response reaches in a short time the steady state value (Figure 6.2).

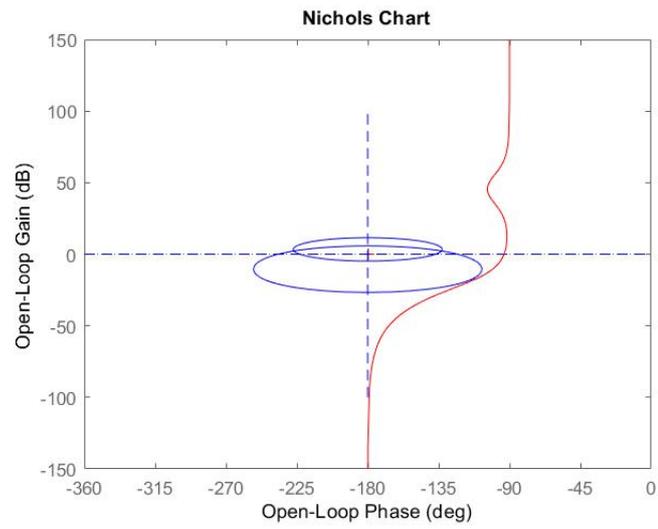


Figure 6.1: Nichols plot of the Open Loop functions computed for the input values J_l

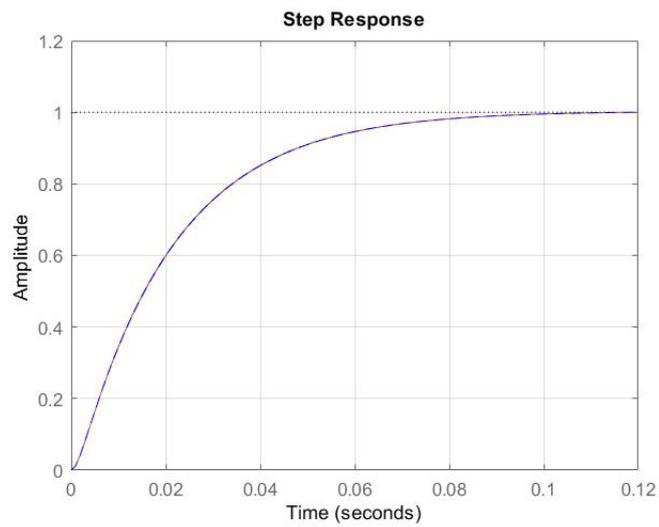


Figure 6.2: Time responses

6.1.1 Velocity Design Optimization

Since the plant has complex zeros and complex poles, according to the way in which the controller has been shaped, also G_c has, in its definition, complex zeros and complex poles.

This is not good for the software implementation including the control system, because of the uncertainties hidden behind the complex values.

The following steps will be focused on the substitution of the complex structure in the controller definition with a real one. The major purpose is to assure that this new form of the controller allows to reach the same time conditions guaranteed by the original one.

From the graphical Bode representation of the plants (Figure 6.3), it results that the complex zeros and complex poles act in the same frequency region of the natural frequencies of the polynomial at the numerator and the polynomial at the denominator, respectively.

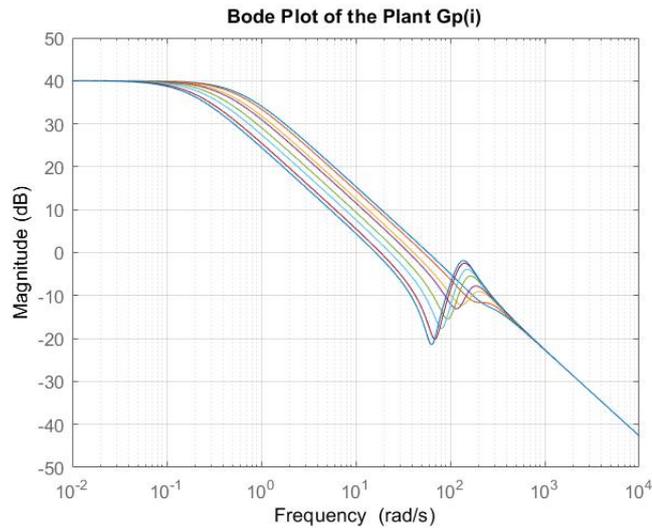


Figure 6.3: Bode Plots of the Original Plant

From the Nichols plot (Figure 6.4), the presence of the complex structure in the plant is visible.

$$L(s) = G_r(s)G_a(s)G_c(s)G_p(s)G_f(s)G_s(s)$$

In the initial case (6.6), the contribution of the complex poles and complex zeros does not appear because of the product between $G_p(s)$ and $G_c(s)$ that results in the elimination of them.

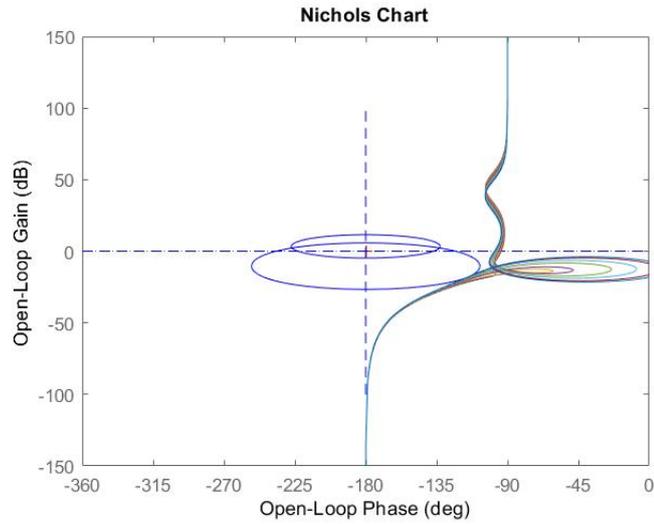


Figure 6.4: Nichols of $L(s)$ functions

By testing the control system performances in time domain, good results can be highlighted

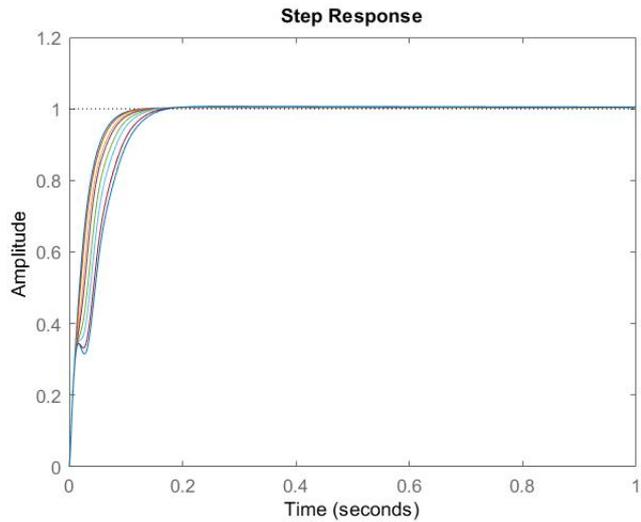


Figure 6.5: Time Responses

The time response has to be improved. At the first time instants, it is possible to observe a slight delay before reaching the steady state.

6.2 Position Controllers

Having described the velocity controllers, it is possible to close the external loop and realize the position controllers. They have been computed by means of the H_∞ approach and the Matlab toolbox supporting it.

The starting point is the definition of two weighting functions that can guarantee better time performances with respect to the ones observed in the velocity implementation, for each J_I value.

In order to understand which functions better approximate the desired behavior in time, several simulations have been performed. This is a time consuming operation, since it is necessary to execute all the design steps to understand if the candidate functions fit with the control objective.

The final choice has been

$$\begin{cases} W_1(s) = \frac{0.73472(s^2+0.165s+0.01361)}{s(s+10^{-6})} \\ W_2(s) = \frac{(s+85)^2}{1.0487} \end{cases} \quad (6.10)$$

The plants have been derived starting from the closed loop functions of the internal loop.

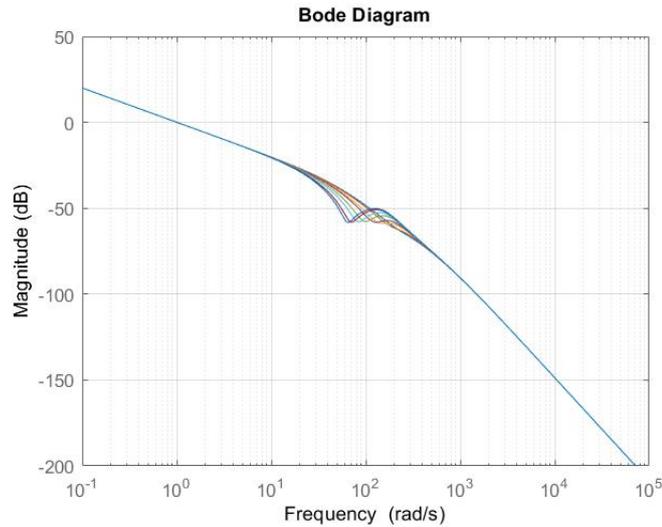


Figure 6.6: Bode Plot of the Plants

They have a high order. This will result in a high order controller structure. In fact

$$G_p(1) = \frac{36832(s+272.2)(s+0.4)(s^2+220.8s+5.519 \cdot 10^4)}{s(s+750.6)(s+234.7)(s+44.71)(s+0.4015)(s^2+289.8s+6.998 \cdot 10^4)}$$

$$G_{c_p}(1) = \frac{21377(s+750.6)(s+234.7)(s+44.65)(s+0.4014)(s+0.02928)(s+0.008132)(s^2+289.8s+6.998 \cdot 10^4)}{(s+9541)(s+235.4)(s+0.4)(s+0.02439)(s+8.508 \cdot 10^{-7})(s^2+109.8s+5201)(s^2+290.1s+7.003 \cdot 10^4)}$$

The interesting thing is that

$$\forall J_l = [J_{l,min}, \dots, J_{l,max}]$$

the Optimal H_∞ performance is around $9.556 \cdot 10^{-1}$

The Nichols plots for all the obtained controllers are illustrated in Figure 6.7

They show the characteristics of the system under analysis and reflect a real situation where it is very difficult to have low-order transfer function that reflect a second order model.

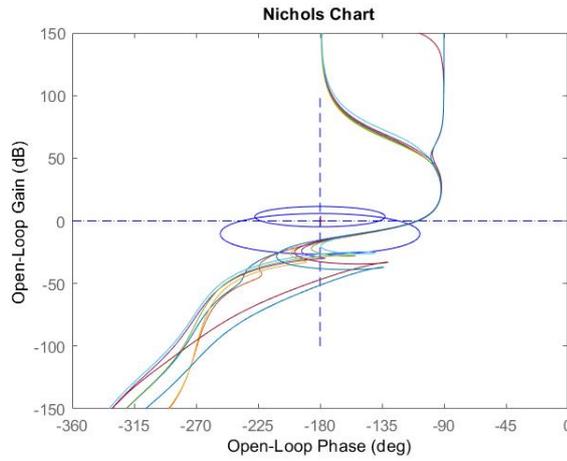


Figure 6.7: Nichols Plant

6.2.1 Position Time Responses

Although the Nichols plot of the Open loop functions demonstrate a complex evolution, the time steps (Figure 6.8) show that the system is acting in a proper way. The delay at the beginning of the responses in the velocity control disappears. The systems quickly reach the steady state and without overshoot.

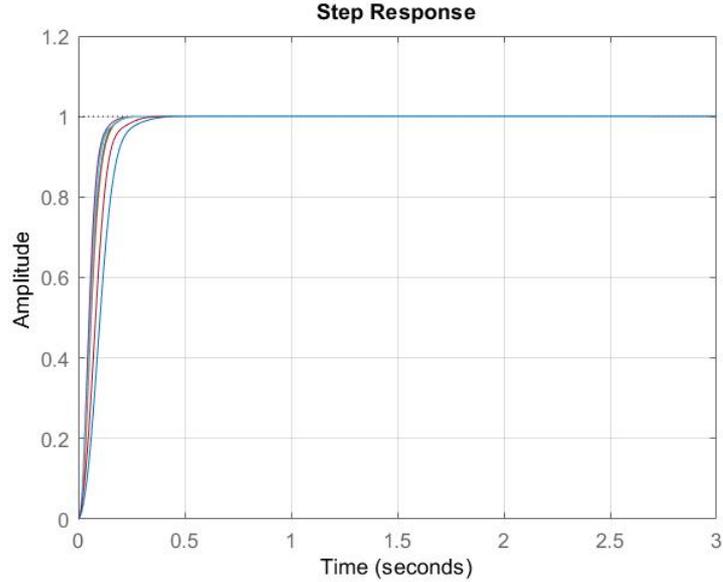


Figure 6.8: Position Time Responses

6.2.2 Position Design Optimization

The final goal is trying to reduce the complexity in the expression of the controller. In order to apply this improvement, the starting point is the approximation of the transfer function of the plants with simpler ones.

The dominant complex zeros and complex poles in $G_p(s)$ are substituted with the correspondent natural frequencies.

$$G_{p,orig}(1) = \frac{36832(s+272.2)(s+0.4)(s^2+220.8s+5.519 \cdot 10^4)}{s(s+750.6)(s+234.7)(s+44.71)(s+0.4015)(s^2+289.8s+6.998 \cdot 10^4)}$$

$$G_{p,app}(1) = \frac{32709(s+272.2)(s+234.9)(s+0.4)}{s(s+750.6)(s+264.5)(s+234.7)(s+44.71)(s+0.4015)}$$

The Bode Plot representing $G_{p,orig}(1)$ and $G_{p,app}(1)$ (Figure 6.9) proves that the two plants are very similar, although $G_{p,app}(1)$ has simpler characteristics.

The development project of the controllers uses an approximated plant. Otherwise, to test the timing performances, the original plants have to be used. In fact, it is important to remember that the plant can not be changed, because it reflects the physical characteristics of the manipulator under analysis, as well as its dynamics. This means that

$$L_i(s) = G_r(s)G_a(s)G_{c_{p,i}}(s)G_{p,orig,i}(s)G_f(s)G_s(s)$$

From the Nichols Plot reporting $L_i(s)$ (Figure 6.10), it is possible to see that they are loosing the characteristics of a second order model and they move inside the constant magnitude loci.

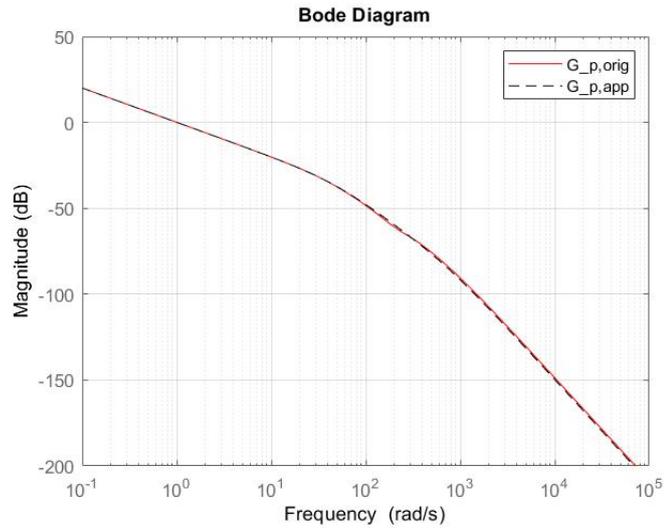


Figure 6.9: Bode Plot of $G_{p,orig}(1)$ and $G_{p,app}(1)$

Also in this case, a real situation is reported where it is difficult to obtain a prototype of the second order, because of the influence of many external factors and uncertainties that make complex the structure of the loop functions.

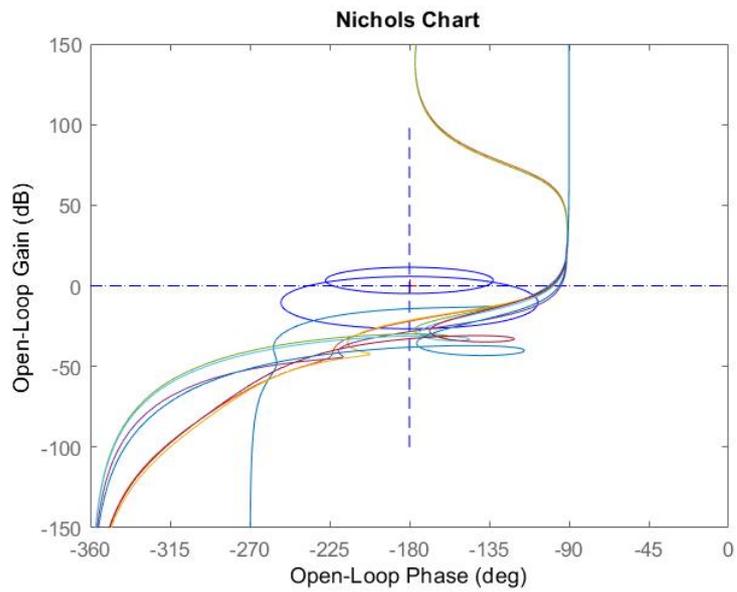


Figure 6.10: Nichols Plot of the Open Loop functions

6.2.3 Final Time Responses

From Time Response, it is possible to state that the overshoot always remains close to zero, but, in order to have a good behavior for all the input Jl parameters, it is necessary to slow down the system. The rise time increases, especially for the system computed starting from the maximum value of the link inertia.

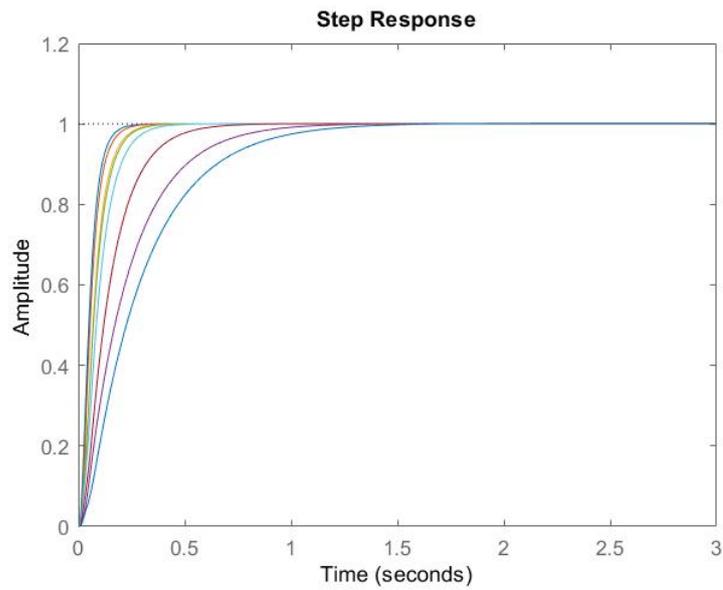


Figure 6.11: Step Responses

At this point, it is possible to conclude that the computed control systems are able to react to the reduction of complexity in the plants, and consequently in the controllers structure, without generating a negative impact in the desired performances.

This procedure shows how much the control structure tries to react in a robust way in each situation that can occur.

Chapter 7

Conclusions and Future Works

The Thesis project can be considered an upward path in the implementation of a complex and robust control system. The starting point has been the realization of a control structure for an ideal robot, in order to discover the characteristics of a specific control algorithm, the H_∞ one, when used in a industrial application. Then, the central focus has been the development of a complex and robust system for a real Comau manipulator. All the implementation phases have been covered: starting from the definition of a model describing the dynamics of the robot up to the realization of standard form of the velocity and position controllers able to guarantee optimal performances whatever the position and orientation of the robot is. The latter implementation can be considered a basic idea to further optimizations. Purpose of future works is the realization of an automatic toolbox that, by following the general rules of the H_∞ approach, can directly produce the desired results. This would represent a revolution in the design project, since most of the implementation work is based on the time consuming trial and error action useful to find the good shape of the weighting functions.

Moreover, it has not been possible to test how the achieved structure reacts on the modeled Comau robot. A future real test can prove if the results obtained through the Simulink toolbox are replicated in practice.

This would confirm the robustness of the project.



Figure 7.1: Comau Robots

Acknowledgements

16 Aprile 2021.

Questo viaggio volge al termine.

Ho dato tutta me stessa per raggiungere i miei obiettivi e realizzare i miei sogni. Oggi sorrido per la strada che ho percorso e per ciò che ho conquistato. Non è sempre stato tutto rose e fiori: gli esami da portare avanti, la lontananza da casa, i continui viaggi per le lezioni in Conservatorio. Ho avuto la fortuna di avere accanto le persone giuste che mi hanno sostenuta e mi hanno aiutata a non perdermi mai. Se mi ritrovo qui a scrivere queste ultime pagine di un importante capitolo è anche per merito loro.

La prima persona che vorrei ringraziare è la Professoressa *Marina Indri*. E' stata lei, con i suoi ricchi insegnamenti, a farmi avvicinare alla robotica e a darmi la possibilità di realizzare un progetto di tesi che mi ha aiutato a crescere professionalmente e personalmente.

Grazie a chi mi ha seguito dall'inizio alla fine nella realizzazione di questo lavoro, dedicandomi il proprio tempo con gioia e premura:

All'Ing. *Stefano Pesce* per aver creduto in me, per aver reso ogni momento stimolante, per i tantissimi consigli a qualsiasi ora e in qualsiasi giorno della settimana. Grazie per aver condiviso con me la tua passione e soprattutto per avermi dimostrato sempre tanto entusiasmo. Questo mi ha aiutato a non scoraggiarmi mai.

All'Ing. *Valerio Perna* per avermi insegnato tantissime cose e avermi dato la possibilità, grazie alla sua professionalità, di toccare con mano una realtà che mi entusiasma così tanto.

E' stato un onore per me condividere questo intero anno con persone che stimo da un punto di vista professionale e soprattutto personale.

Alla mia mamma e al mio papà. Non si possono elencare e descrivere tutte le cose che mi avete dato nella vita. Voi siete il meglio per me. Vi ringrazio per essere ciò che siete e per avermi reso ciò che sono. Grazie per non avermi mai fatto mancare nulla, per avermi insegnato ciò che conta nella vita e per aver sostenuto ogni mia scelta.

Alla mia roccia, *Fabiola*. Non credo esista sentimento più forte di quello che mi lega a te. Sei la mia fortuna più grande, il mio senso di tutto. Sono fiera di guardarti crescere e accompagnarti. Grazie perchè credi in me e questo mi da' sempre forza. La cosa più difficile è stata averti lontana. Mi manchi ogni volta che non sei con me. Avrai sempre

me, così come io avrò sempre te. Questa è la mia certezza.

Ad *Ivan*. Grazie per esserti preso cura di me. Per aver capito i miei sogni e averli resi anche tuoi. Per avermi consigliato il meglio ed avermi seguito sempre con soddisfazione. Per esserti emozionato insieme a me. Per avermi dimostrato che con te accanto non avrei mai dovuto aver paura di cadere. Abbiamo condiviso così tanto. Ci sei sempre stato e so che sempre ci sarai, perchè come dici sempre tu: "qualunque cosa accada, sarai sempre la mia piccolina".

Grazie alle mie amiche:

Ad *Irene*, per essere entrata nella mia vita e aver deciso di rimanere. Abbiamo affrontato questi anni insieme, scegliendo di essere l'una la spalla dell'altra, nel bene e nel male. Non credevo potesse nascere così tanto in così poco. Mi emoziono nel vedere che siamo così vicine, così simili. Io e te siamo una squadra.

A *Mery*, perchè la nostra non è un'amicizia ovvia, ma una continua scelta, una sicurezza. Sei quella persona che c'è nei momenti in cui ne ho più bisogno, che rimane ore e ore insieme a me al telefono, solo per tenermi compagnia, che mi dimostra il suo affetto in modi che solo noi possiamo capire. Ormai non servono parole, ci basta solo uno sguardo. Grazie per essere rimasta accanto a me nei momenti più bui della mia vita.

A *Maria Rosaria*, perchè non importa la distanza, ma la consapevolezza che qualsiasi cosa accada posso contare su di te. La nostra amicizia è iniziata tra i banchi del Conservatorio. Guarda quanta strada abbiamo fatto. Siamo diverse, eppure ci completiamo.

A *Mary*, che anche da lontano ha continuato a seguire i miei passi, giorno dopo giorno. Sembra ieri che percorrevamo insieme i corridoi dell'Unical, che, sfinite dalle lunghissime lezioni, ci buttavamo sul letto una accanto all'altra e condividevamo emozioni, paure, dubbi.

Ai miei due cavalieri: *Mario* e *Antonio*, che mi hanno sempre trattata come se fossi speciale.

Grazie al mio *Maestro*, *L. Genitori*, per avermi dato la possibilità di concludere i miei studi senza dover mai mettere da parte la passione più grande della mia vita: la musica. Grazie per avermi seguita in tutti questi dieci anni con soddisfazione e stima.

Grazie alle *mie nonne*, semplicemente per esserci: mia nonna *Graziella*, la mia mamma bis, e mia nonna *Annina*. Vedere l'emozione sui vostri volti è il regalo più grande per me.

I miei pilastri, purtroppo, oggi non ci sono più. Avrei voluto che avessero continuato a guidarmi nei passi più importanti della mia vita, come solo loro sapevano fare: con amore, dedizione, bontà e dolcezza. E oggi li vedo qui con me, a festeggiare: mio *nonno Pino*, che sorride con orgoglio e mio *nonno Elam*, a cui ho affidato il mio cuore, con i suoi occhi lucidi. Spero che guardandomi possano essere fieri di me e sappiano che **tutto questo lo dedico a loro**.

Ad augusta per angusta

Bibliography

- [1] Bob Williams. «An introduction to robotics». In: *Mech. and Cont. of Robotic Manipulators* (2014).
- [2] Bruno Siciliano, Lorenzo Sciavicco, Luigi Villani, and Giuseppe Oriolo. *Robotics: modelling, planning and control*. Springer Science & Business Media, 2010.
- [3] B Messner and D Tilbury. *Control Tutorials for MATLAB and Simulink-Introduction: PID Controller Design.[online]* Ctrms. engin. umich. edu. 2015.
- [4] Kemin Zhou. «ESSENTIALS OF ROBUST CONTROL». In: (1998).
- [5] Ankit Bansal and Veena Sharma. «Design and analysis of robust H-infinity controller». In: *Control theory and informatics* 3.2 (2013), pp. 7–14.
- [6] Vito Cerone, Massimo Canale, and D Regruto. «Loop-shaping design with constant magnitude loci in control education». In: *International Journal of Engineering Education* 24.1 (2008), p. 127.
- [7] Stig Moberg, Jonas Ohr, and Svante Gunnarsson. «A benchmark problem for robust feedback control of a flexible manipulator». In: *IEEE Transactions on Control Systems Technology* 17.6 (2009), pp. 1398–1405.
- [8] Zhou Yun-fei, Song Bao, and Chen Xue-dong. «Position/force control with a lead compensator for PMLSM drive system». In: *The International Journal of Advanced Manufacturing Technology* 30.11 (2006), pp. 1084–1092.