Politecnico di Torino

Master Degree in Mechatronic Engineering

Master Degree Thesis

Control of a Dual Dry Clutch Transmission System: comparison of different techniques

---

Supervisor:

Prof. Carlo Novara
Ing. Emanuel Corigliano

Author:

Domenico Ippolito

December 14, 2018

"Quello che per un uomo è 'magia', per un altro è ingegneria"

"One man's 'magic' is another man's engineering"

Robert Anson Heinlein

Ringraziamenti

Il mio primo ringraziamento va senza dubbio alla mia famiglia, in particolar modo a mio padre, mia madre e mia sorella che con il loro instancabile supporto, sia morale che economico hanno contribuito al raggiungimento di questo importante traguardo.

Un sincero ringraziamento va ai miei cugini Pietro e Pietro, per aver creduto sempre in me e con i quali ho un rapporto fraterno.

Un grazie particolare va a tutti gli amici che mi hanno supportato e sopportato durante questo percorso, facendomi sentire meno lontano da casa. In particolar modo Francesco, mio coinquilino per 5 anni.

Un sentito grazie a tutti i colleghi con cui ho affrontato e superato le difficoltà in questa eperienza universitaria.

Ringrazio i miei relatori, il Prof. Novara per i consigli che mi ha dato durante questi mesi di lavoro e l'Ing. Corigliano che mi ha permesso di svolgere questo progetto di tesi con il Centro Ricerche Fiat.

Abstract

Transmission is the mechanism that brings the power produced by the engine to the wheels of a vehicle according to the driver's request. Two types of transmissions are commonly used in automotive industry: Manual Transmission (MT) and Automatic Transmission (AT). They both present advantages and disadvantages. The main problems of the MT are the not smooth gear shifting that induces oscillations in the driveline, torque transmission interruption and low comfort. For what concerns the AT, the comfort is enhanced to the detriment of an increase of both fuel consumption and manufacturing costs. Hence, the Automatic Manual Transmission (AMT) and then the Dual Clutch Transmission (DCT), were introduced to reduce the weak points of MT and AT.

There are several issues concerning DCT, such as vehicle launch, effective gear shifting, smooth clutch engagement and driving comfort. This thesis is concerned with the problem of controlling the dual clutch during its slipping phase. This phase is critical because of the clutch engagement during motion. In particular, the clutch engagement should occur smoothly, reducing oscillation and jerks, improving the drivability and driver's comfort.

The thesis is in collaboration with Centro Ricerche Fiat (CRF). The overall aim of the thesis project is to design a controller for the dual dry clutch transmission (DDCT). Two methods are first considered: Loop-shaping and H-infinity. A third method, based on a more general control structure and on a Model Predictive Control (MPC) approach, is also investigated. The obtained simulation results show that the designed controllers can provide a satisfactory tracking performance, yielding a smooth engagement of the clutch, with acceptable oscillations on the longitudinal acceleration and on the drive shafts.

The thesis is structured as follows. In the first chapter, an overview of the DCT system and its components is presented. This section includes the description of the clutch Simulink model provided by CRF and the related differential equations which describe the clutch system dynamics. It also includes the characterization

of the used control system architectures.

The second chapter analyzes the Loop-shaping approach and the workflow to design a controller using such a frequency-domain approach.

The third chapter discusses the H-infinity (H_∞) approach. The structure is similar to the second chapter.

The fourth chapter shows an overview of the Model Predictive Control approach.

The fifth chapter illustrates a comparison between the used methods and the obtained simulation results.

The sixth chapter exhibits conclusions and additional considerations.

Contents

1 Overview on Dual Clutch Transmission **1**

1.1 Introduction 1

1.1.1 C635 DDCT Transmission & Dry Dual Clutch Unit 2

1.1.2 Electro-hydraulic actuation system & control unit 3

1.2 Micro-slip problem 6

1.3 State of art 7

1.4 Detailed model configuration 8

1.4.1 Reference generator 9

1.4.2 Actuator 11

1.4.3 Engine 11

1.4.4 Load 12

1.4.5 DDCT system 13

1.4.5.1 Control 14

1.5 Simplified model 14

1.5.1 Plant architecture 17

1.6 Control system architecture 20

1.6.1 Base 20

1.6.2 Feedforward 20

1.6.3 Feedforward with a new reference input 22

2 Loop-shaping approach **24**

2.1 Overview 24

2.2 Design 25

2.2.1 Sign of Kc 27

2.2.2 Lead and Lag Compensators 29

2.3 Loop shaping controller on Base architecture 35

2.4 Loop shaping controller on Feedforward architecture 36

2.5 Loop shaping controller on Feedforward architecture with a new reference input . . . 39
2.6 Loop shaping controller on CRF's detailed model . . . 43

3 H-infinity (H_∞) approach **47**
3.1 Overview . . . 47
3.1.1 Robust control . . . 50
3.2 Design . . . 50
3.2.1 Weighting functions computation . . . 51
3.2.1.1 $W_s(s)$ function . . . 51
3.2.1.2 $W_t(s)$ function . . . 52
3.2.1.3 $W_u(s)$ function . . . 52
3.2.2 LMI optimization approach . . . 53
3.3 H_∞controller on Base architecture . . . 59
3.4 H_∞controller on Feedforward architecture . . . 60
3.5 H_∞controller on Feedforward architecture with a new reference input 62
3.6 H_∞controller on CRF's detailed model . . . 64

4 Model Predictive Control approach **67**
4.1 Overview . . . 67
4.2 Design . . . 69
4.3 MPC with two command input . . . 72

5 Comparison of used methods **75**
5.1 Robust analysis . . . 76

6 Conclusions **77**
6.1 Future Works . . . 78

Bibliography **79**

List of Figures

1.1 C635 MT and DDCT version. 2
1.2 Cross section of the C635 DDCT. 3
1.3 Hydraulic Power Unit (PU), Complete Actuation Module (CAM). . 4
1.4 Complete Actuation System (CAS). 5
1.5 Schematic model of a dry clutch. 6
1.6 Control scheme from [2]. 7
1.7 Decoupled control scheme [3]. 8
1.8 MPC control scheme [4]. 8
1.9 Blocks scheme of entire DDCT system. 8
1.10 Reference Generator block. 9
1.11 Inner part of Reference Generator block. 10
1.12 Reference signal. 10
1.14 C_m torque signal. 11
1.13 Inner part of actuator block. 11
1.15 Inner part of DDCT system. 13
1.16 DDCT system Simulink scheme. 13
1.17 Static controller block. 14
1.18 Driveline scheme. 15
1.19 Table containing driveline parameters. 17
1.20 Plant represented with elementary blocks. 18
1.21 Plant represented with state space blocks. 18
1.22 Plant represented with LTI blocks. 19
1.23 Slipping speed w_d for both detailed and simplified model. 19
1.24 Base architecture. 20
1.25 General feedforward scheme. 21
1.26 Feedforward control scheme. 22
1.27 tanh shape reference input. 23

2.1 General control system scheme. 24
2.2 Relation between $t_r * \omega_c$ and ζ (from [8]) 27
2.3 Nyquist plot with $K_c = 1$. 28
2.4 Nyquist plot with $K_c = -1$. 29
2.5 Lead network magnitude contribution. 30
2.6 Lag network magnitude contribution. 31
2.7 Nichols chart with $G_c(s) = \frac{K_c}{s}$. 32
2.8 Nichols chart with $G_{c1}(s)$. 33
2.9 Nichols plot with $G_{c2}(s)$. 34
2.10 Step response with $G_{c2}(s)$. 34
2.11 w_d responses with the first three controllers. 36
2.12 w_d responses with the last six controllers. 37
2.13 w_{sr} outputs with first three controllers. 37
2.14 A_x values with the first three controllers. 38
2.15 w_{sr} outputs with last six controllers. 38
2.16 A_x values with the last six controllers. 39
2.17 w_d responses with new reference (first three controllers). 40
2.18 w_{sr} outputs with new reference (first three controllers). 40
2.19 A_x values with new reference (first three controllers). 41
2.20 w_d responses with new reference (last six controllers). 41
2.21 w_{sr} outputs with new reference (last six controllers). 42
2.22 A_x values with new reference (last six controllers). 42
2.23 w_d responses with CRF's model (first three controllers and PI). . . 43
2.24 w_{sr} outputs with CRF's model (first three controllers and PI). . . . 44
2.25 A_x values with CRF's model (first three controllers and PI). 44
2.26 w_d responses with CRF's model (last six controllers and PI). 45
2.27 w_{sr} outputs with CRF's model (last six controllers and PI). 45
2.28 A_x values with CRF's model (last six controllers and PI). 46

3.1 General feedback system. 48
3.2 General feedback system with Weighting functions. 49
3.3 Plot of the relevant errors. 53
3.4 *'generalized_plant'* Simulink file. 55
3.5 Sensitivity function $S_n(s)$ and weighting function $W_s^{-1}(s)$. 57
3.6 Complementary Sensitivity function $T_n(s)$ and weighting function $W_t^{-1}(s)$. 57

3.7 Complementary Sensitivity function $T_n(s)$ and weighting function $W_u^{-1}(s)$. 58
3.8 Step response of the system with $G_c(s)$. 58
3.9 Nichols plot with $G_c(s)$. 59
3.10 w_d response with $H_{\infty 2}$ in Base architecture. 60
3.11 w_d responses using Feedforward architecture. 61
3.12 w_{sr} outputs using Feedforward architecture. 61
3.13 A_x values using Feedforward architecture. 62
3.14 w_d responses using new reference architecture. 63
3.15 w_{sr} outputs using new reference architecture. 63
3.16 A_x values using new reference architecture. 64
3.17 w_d responses with CRF's detailed model. 65
3.18 w_{sr} outputs with CRF's detailed model. 65
3.19 A_x values with CRF's detailed model. 66

4.1 Control horizon T_c and prediction horizon T_{pr}. 68
4.2 MPC Simulink block. 70
4.3 w_d responses with MPC. 71
4.4 w_{sr} outputs with MPC. 71
4.5 A_x values with MPC. 72
4.6 w_d response (MPC with two command input). 73
4.7 w_{sr} output (MPC with two command input). 73
4.8 A_x value (MPC with two command input). 74

5.1 w_d responses with $H_{\infty 2}$ in Feedforward architecture with two different K_{usura} values. 76

List of Tables

2.1 Loop Shaping designed controllers. 36

3.1 H_∞ controllers overview. 60

4.1 MPC controller values. 70

5.1 Recap of best controllers. 75
5.2 Robust analisys comparison. 76

Chapter 1

Overview on Dual Clutch Transmission

1.1 Introduction

Nowadays there are several kind of transmissions, the first to be introduced was the Manual Transmission (MT) which exhibits high efficiency, low cost and allows full control to the driver. At a later time was introduced the Automatic Transmission, where comfort and ergonomics are privileged but to the detriment of higher cost and lower efficiency. So a trade-off between these two kind of transmissions was necessary, thus the Automated Manual Transmission (AMT) system arrived on the market. The AMT guarantees a combination of high transmission efficiency with optimal use of the engine fuel consumption characteristics, ensuring at the same time low add-on cost and good ergonomics. Its drawback is the lack of comfort in automatic mode due to torque interruption during the automated gear shifting which can be felt by some drivers. To overcome this issue the Dual Clutch Transmission was proposed, as matter of fact it eliminates the torque interruption of the AMT, improving significantly the efficiency compared to the previous transmissions optimizing the use of the engine fuel consumption. The DCT is composed by two independents sub-gearboxes, one for the even gear sets and the other for odd gear sets, each one activated by separate clutches: the on-coming clutch and the off-going clutch to ensure a shift without traction interruption. DCT has also the ability to allow the driver to gear selection in manual mode because the transmission operates in fully automatic mode. There are two fundamental kinds of clutches used in DCTs: either two wet multi-disc clutches which are bathed in oil for cooling (WDCT), or two dry single-disc clutches (DDCT).

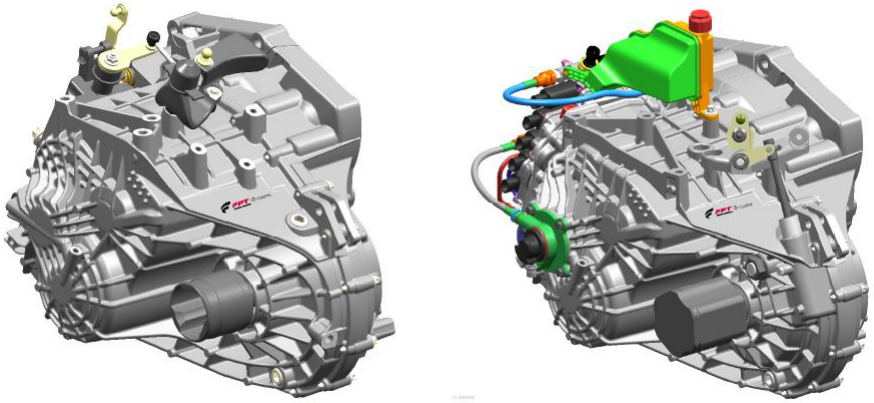

Figure 1.1: C635 MT and DDCT version.

This thesis takes into account the Dual Dry Clutch Transmission (DDCT) developed by Fiat Power-train Technologies. The C635 transmissions are transversal front wheel drive and are characterized by a compact three shaft architecture with a maximum input torque of 350 Nm. The research introduces and takes into consideration the control strategies to develop a controller using different methods: Loop-shaping, H-infinity and Model Predictive Control. The control law has to be able to track a particular reference for clutch slipping speed, assuring a low oscillation both on longitudinal acceleration of the vehicle and on drive shaft during motion.

1.1.1 C635 DDCT Transmission & Dry Dual Clutch Unit

As reported in [1], the 3-shaft transmission architecture is contained in a 2-piece aluminium housing with an intermediate support plate for the shaft bearings. The gear set housing is characterized by a reduced upper secondary shaft length, a feature was also necessary to ensure packaging in the lower segment vehicles, where the longitudinal crash beam imposes serious installation constraints. The most important feature of this transmission, in terms of packaging characteristics is the adoption of a coaxial pull-rod for the actuation of the odd-gear clutch (K1), while the even gear clutch (K2) is actuated with a rather conventional hydraulic Concentric Slave Cylinder (CSC). This pull-rod is connected to a hydraulic piston actuator located on the rear face of the transmission housing in a manner identical to the one adopted in the past in an earlier FPT (Fiat Powertrain Technologies) technical demonstrator. We can see in the figure the relative cross section.

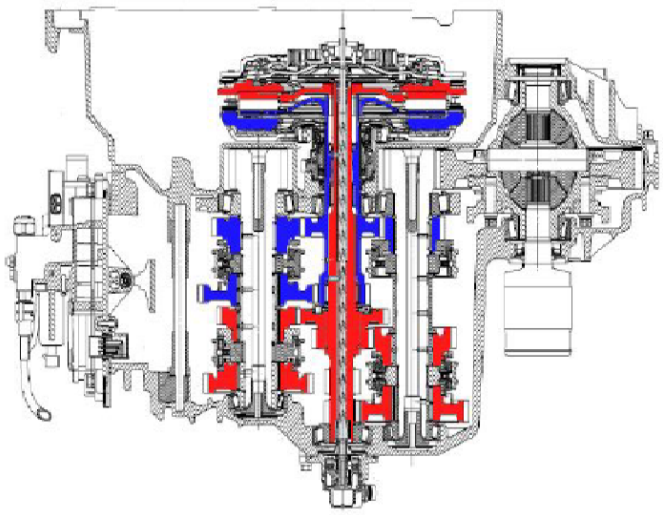

Figure 1.2: Cross section of the C635 DDCT.

The Dual Clutch Unit, as mentioned in [1], is an engineering benchmark in terms of packaging, for this reason, together with the K1 actuator solution are the main contributors to the transmission's compactness. As in conventional MT the clutch K1 is normally closed while the even gear clutch K2 is normally open. The K1 position is controlled by means of a contact-less linear position sensor integrated in the rear hydraulic piston actuator whereas K2 is controlled in force, i.e., through hydraulic pressure. The two clutches act on a center plate together with the two pressure plates. The whole dual clutch unit is installed on the clutch housing by means of a single main support bearing. The adoption of the specific actuation system of the clutch K1 allows the space for such a bearing to be mounted.

1.1.2 Electro-hydraulic actuation system & control unit

In C635 DDCT clutches and gear shifting mechanisms are electro-hydraulically actuated through a dedicated, sealed, hydraulic oil circuit [1]. The system is composed of a hydraulic power unit (PU, Figure 1.3), consisting of an electrically driven high pressure pump and accumulator, and an Actuation Module (CAM, figure) which includes the control solenoid valves, gear shift actuators and sensors.

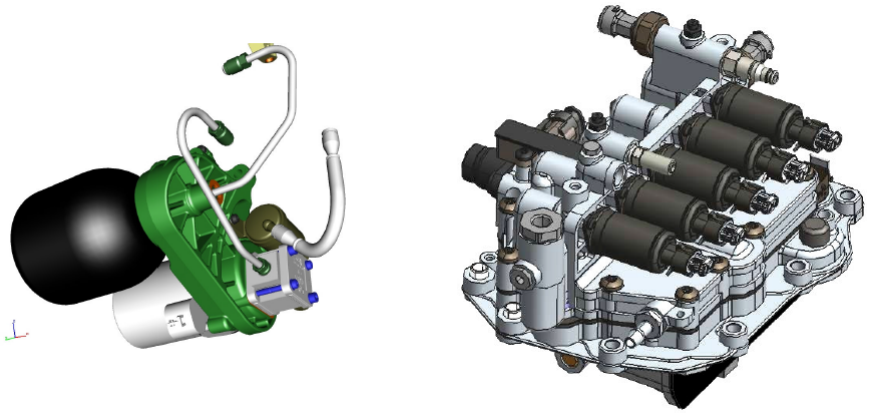

Figure 1.3: Hydraulic Power Unit (PU), Complete Actuation Module (CAM).

The clutch and gear actuation module (CAM), consists of 4 distinct double action pistons operating the gear engagement forks, one shifter spool which selects the piston to be actuated and 5 solenoid valves of which 4 are pressure proportional (PPV) and one flow proportional (QPV). Two of the PPVs actuate the gear engagement piston which is selected by the spool valve operated by the third PPV. The fourth PPV is used for the control of the K2 clutch CSC. The QPV is used for the position control of the K1 clutch. All solenoid valves are direct derivatives of those currently used in FPTs AMT systems and, therefore, employ well proven technology and guarantee robustness. The Actuation Module also comprises 5 non-contact linear position sensors, one for each shifting piston and one for the shifter spool, as well as two speed sensors reading the speed of the two primary shafts. Finally, one pressure sensor is used for the control of the K2 clutch and one for the system pressure monitoring and control. Figure 1.4 represents the hydraulic circuit of the complete actuation system (CAS).

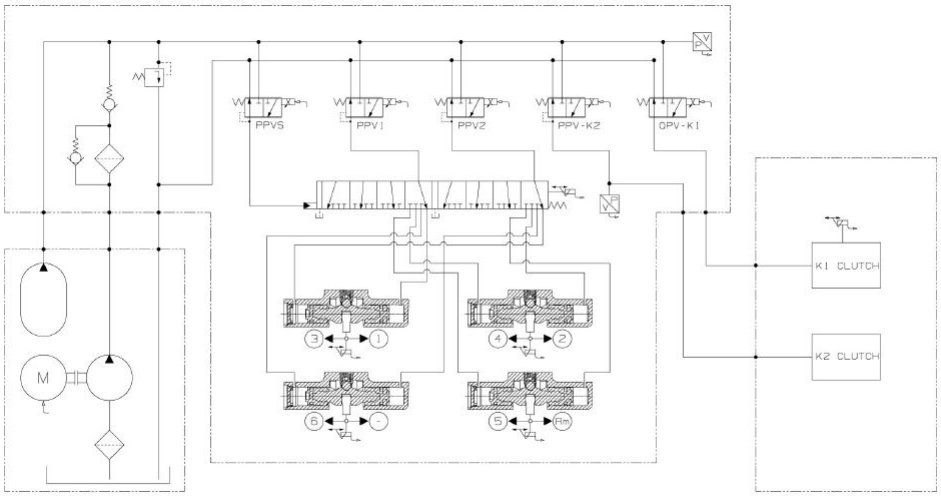

Figure 1.4: Complete Actuation System (CAS).

The C635 DDCT control strategies run in a multitasking environment preserving the Main Micro Controller resources. As described in [1], the strategies can be grouped as:

- Actuator Control, the actuator control strategies exploit the high performance attainable with electro-hydraulic actuators. The principal control strategies are:
  - Engagement actuators control: based on a force/speed control concept.
  - Shifter (selector) control: hydraulic power to the required engagement actuator is guaranteed by a fast and precise control of the shifter.
  - Odd gears clutch controls: the normally closed clutch (K1) is controlled by a position closed loop. This is the clutch of the first and of the reverse gear; therefore, this control strategy is essential also for the vehicle starting performance.
  - Even gears clutch: the normally open clutch (K2) is controlled in force with a pressure feedback signal delivered by one of the CAM sensors.

- Self-tuning controls: The main self-tuning control algorithms concern the conversion of the requested clutch transmitted torque to K1 position and K2 pressure.

- Launch and gear shift strategies: The C635 DDCT implements various driving modes, depending on the desired performance and Brand/OEM requirements, both in manual and in automatic mode.

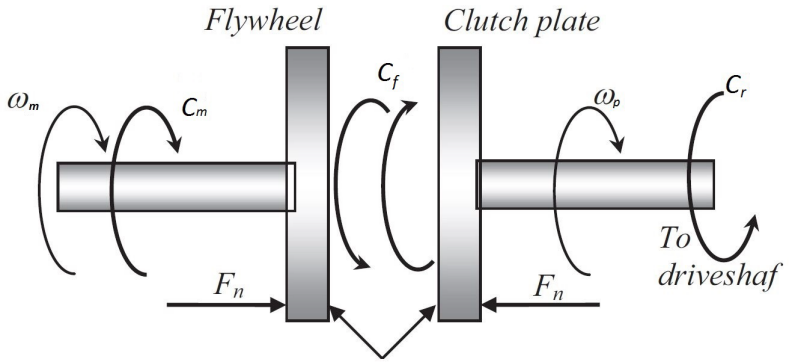

Figure 1.5: Schematic model of a dry clutch.

1.2 Micro-slip problem

A simple explanation of the clutch mechanism is that it consists of two rotating plates (flywheel-clutch and plate-clutch) that can be pressed together. When this happens friction will arise and transmit a torque between the plates, which acts to reduce the speed difference. The micro-slip affects one clutch at a time, thus in this thesis project, the control design will refer to only one clutch. In Figure 1.5 is shown the schematic model of a dry-clutch, where the normal force F_n produced by the clutch actuator presses the disks against each other, in this way the friction due to their contact allows the transmission of the torque called C_f. Hence varying F_n, the transmitted torque C_f can be controlled. There are also w_m and w_p which are respectively the angular velocity given by the engine and the clutch angular velocity. In figure it is also shown C_m (engine torque) and C_r (torque reacting by the driveline). Basically we can find the clutch in one of these phases:

- open clutch phase ($C_f = 0$), when the two disks are separated;
- slipping phase, where the speed difference between engine speed and primary shaft is not zero, $w_d = w_m - w_p$;
- closed clutch phase ($C_f = C_m$), where there is the full transmission of torque between engine and primary shaft.

The micro-slip problem is indeed related to the slipping phase. This is the principal problem of this thesis: guarantee a smooth engagement between the two disks in order to reduce torque oscillation on driveline, improving comfort and drivability.

1.3 State of art

In literature are presents several research works that propose different methods to solve clutch slipping control and torque oscillation problem in a dry clutch transmission system. In [2] it is presented a smooth control algorithm based on measurements of engine speed, clutch speed and on estimation of the dual-clutch engaging torque. In the Figure 1.6 is shown the control scheme . Using this

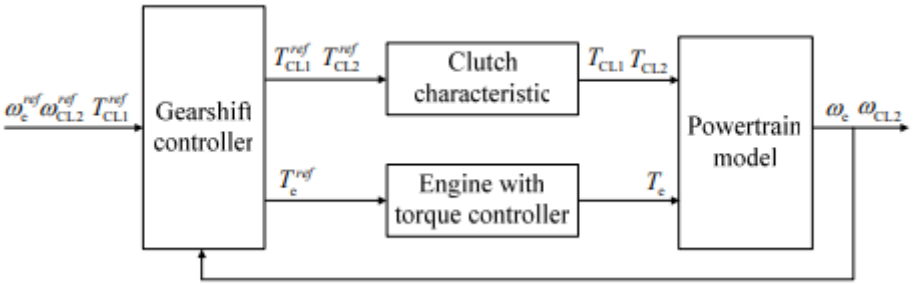

Figure 1.6: Control scheme from [2].

approach the control objective are fulfilled, ensuring that engine and dual clutch speeds track desired reference signals. The gearshift controller inputs are the references to track, while the outputs are the reference torques of engine and two clutches. These two blocks produce the torques entering in the powertrain model. The loop provides the desired difference between engine torque and clutch torque.

In [3] a proposal to generate the optimal engine and clutch reference speeds based on the optimal control theory. A decoupled controller is used to study the vehicle launch process, a decoupled PID is derived for smooth clutch engagement. The system is modeled as two-input and two-output and both engine and slip speeds can be controlled independently through the two PID. Both clutch and vehicle model provide a speed feedback in order to improve reference tracking. The whole system scheme is shown in Figure 1.7.

A different kind of strategy is presented in [4], where as control variable is used only the clutch torque, considering the engine torque as a known non-controllable input. The engine control unit (ECU) outputs the engine torque Γ_e based on the throttle pedal position x_p. t_r is the time control horizon given as a function of the total engagement time computed for a certain pedal position. The MPC obtains the clutch torque solving the optimal control problem with a suitable cost function. The control structure is presented in Figure 1.8.

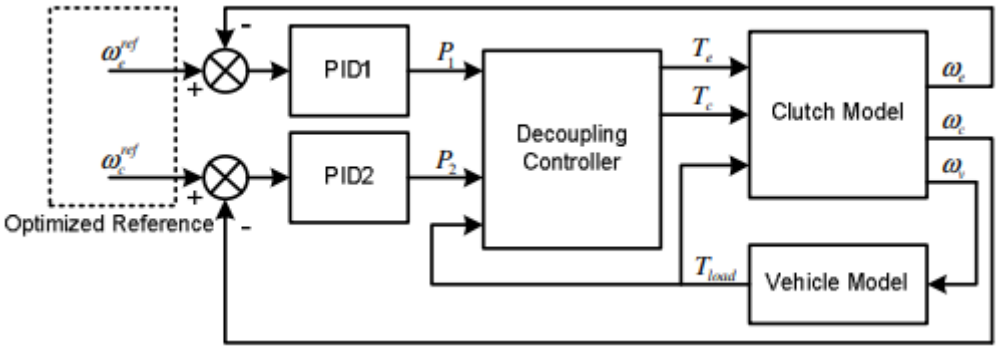

Figure 1.7: Decoupled control scheme [3].

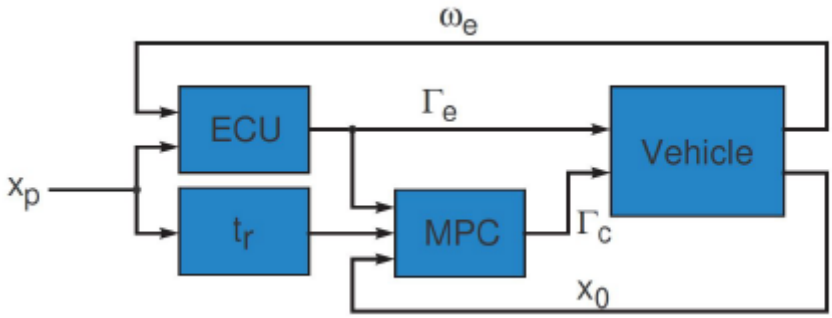

Figure 1.8: MPC control scheme [4].

1.4 Detailed model configuration

This chapter provides information about the detailed driveline model designed by Centro Ricerche Fiat (CRF), using Matlab/Simulink. It is very important understand this model in order to develop a proper controller. The model under investigation is divided in the following blocks:

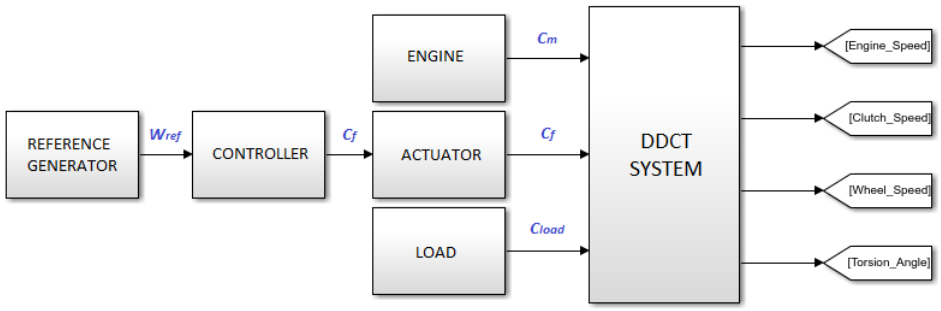

Figure 1.9: Blocks scheme of entire DDCT system.

- Reference generator: provides the reference signal that the slipping speed w_d

has to follow;

- Actuator: gives the requested torque C_f;
- Engine: provides the driving torque C_m;
- Load: produces the load torque C_{load} which includes the air, roll and slope resistances;
- DDCT system: includes the clutch and the transmission model.

These blocks are explained in a specified way in the following.

1.4.1 Reference generator

As shown in Figure 1.10, the reference generator has five inputs where *Signal 1* is always equal to 0, $Om\_MisRpm = w_m$, $Oc\_MisRpm = w_p$, there is also the reset input that enables the control after 0.6 seconds.

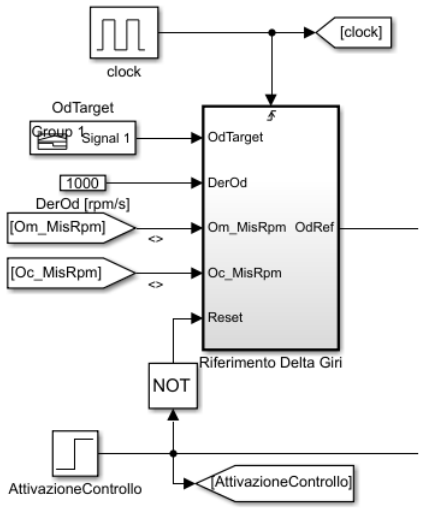

Figure 1.10: Reference Generator block.

To understand the relationship between inputs and output we can see inside that block (Figure 1.11).

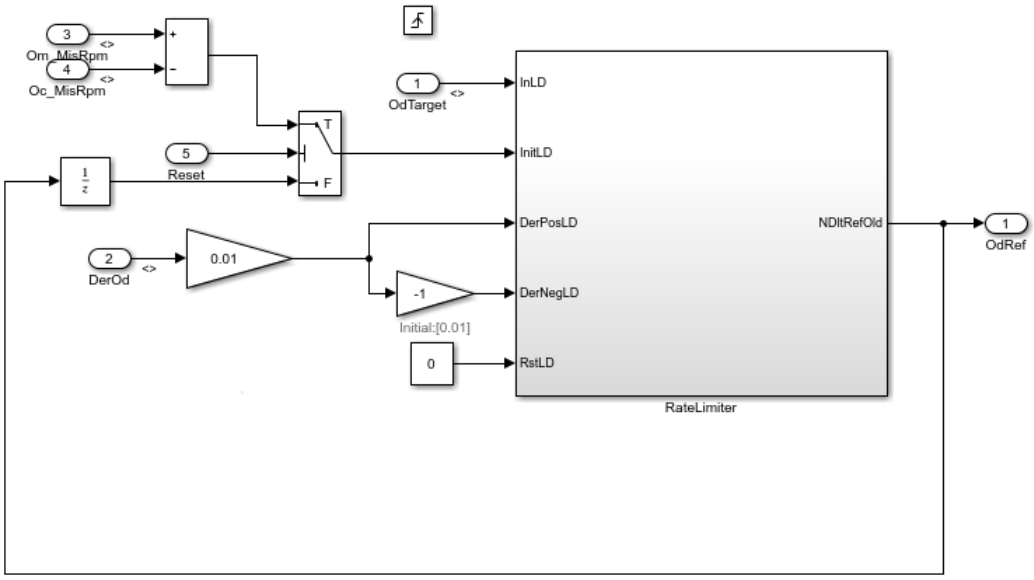

Figure 1.11: Inner part of Reference Generator block.

We can notice that the output of the generator depends on $w_d = w_m - w_p$ just for 0.6 seconds, then the control becomes active and the new value of the output depends on its last value.

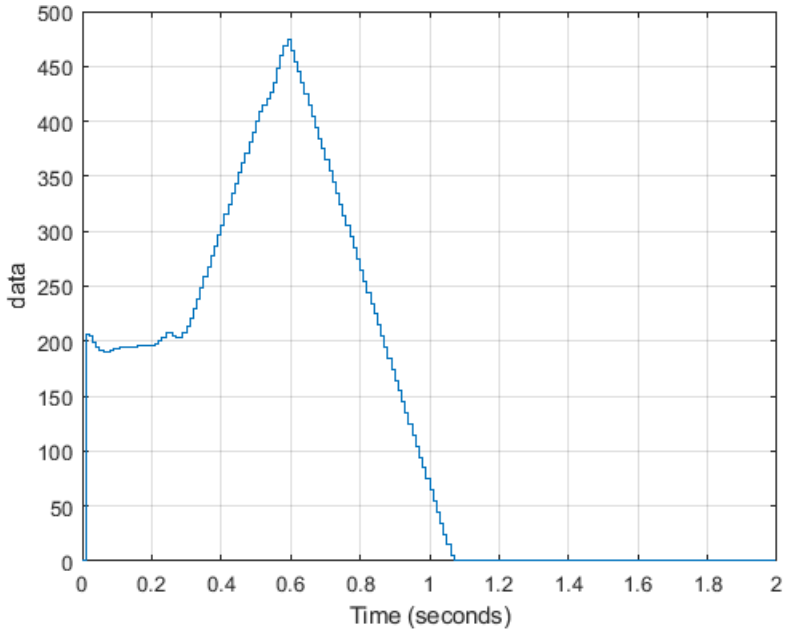

Figure 1.12: Reference signal.

In the figure above is shown the output of the reference generator, the part where the signal grows is done by the open-loop contribute of the control, while the slope part (where the control starts) should bring the slipping speed w_d to 0.

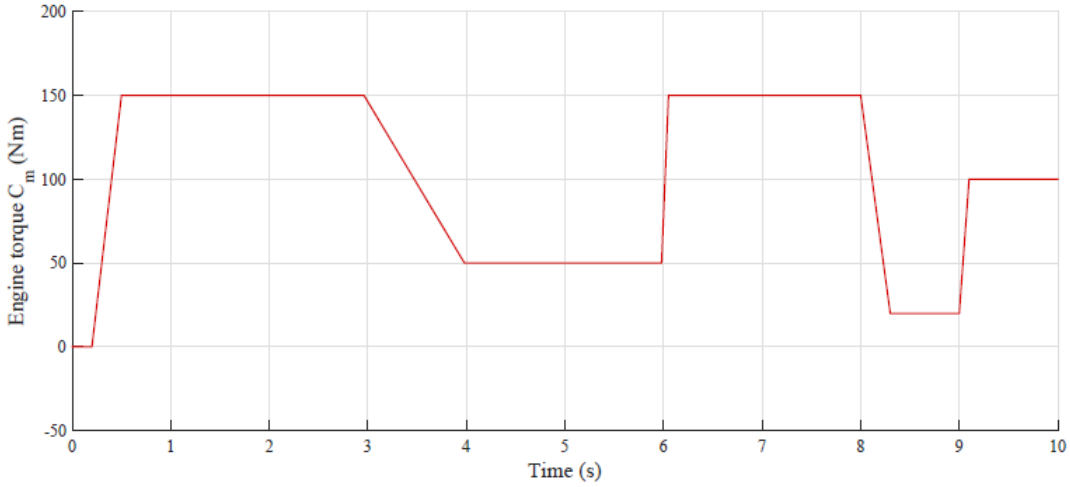

Figure 1.14: C_m torque signal.

1.4.2 Actuator

Actuator is a fundamental part of the system, it is composed by the following blocks:

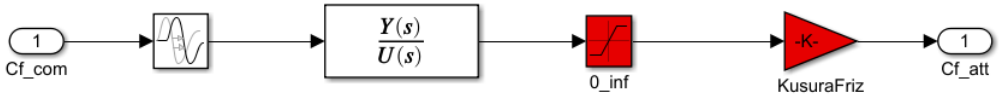

Figure 1.13: Inner part of actuator block.

- *0_inf* is a saturation block having a range between 0 and ∞;
- *KusuraFriz* is a gain that takes into account the clutch wear;
- a delay block that represents the actuator delay equal to 10 ms;
- a transfer function that describes the actuator's dynamic:

$$G_a(s) = \frac{25.305(s+138)(s^2-161.3s+3.134e04)}{(s^2+82.23s+2563)(s^2+134.1s+4.27e04)}$$

1.4.3 Engine

The engine is simply model as a fixed signal having output torque (C_m) shape in Figure 1.14.

1.4.4 Load

The load torque is composed of three elements: the aerodynamic resistance F_a, the rolling resistance force F_r and the force due to the gravity F_g when driving in a non-horizontal road. The aerodynamic resistance force is modeled as:

$$F_a = 0.5\rho_a A_f C_a (v_a + v_v)^2$$

where:

- A_f is the frontal area of the vehicle,
- C_a is the aerodynamic drag coefficient,
- v_v is the vehicle speed,
- v_a is the wind speed,
- ρ_a is the density of the air.

The rolling resistance force is:

$$F_r = m_v g \mu_r \cos(\beta)$$

where:

- μ_r is the rolling friction coefficient,
- α is the slope angle of the road,
- m_v is the vehicle mass,
- g is the gravity acceleration.

The uphill driving force is:

$$F_g = m_v g \sin(\alpha)$$

For simplicity we consider to drive in a horizontal road ($\alpha = 0$), deleting the F_g contribution, so considering also the wheels radius r_w, the vehicle resistance torque C_{load} becomes:

$$C_{load} = (F_a + F_r) r_w$$

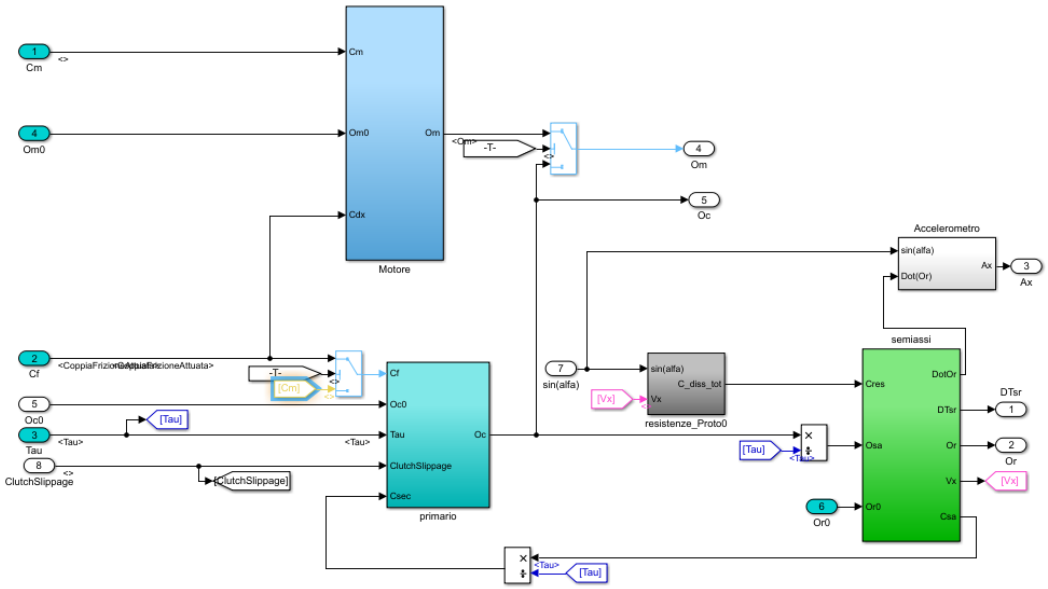

Figure 1.15: Inner part of DDCT system.

1.4.5 DDCT system

This block describes the driveline, if we look inside (Figure 1.15) there are three sub-blocks which model the engine, the primary shaft and the drive shaft. Each sub-block has its relative inertia and its velocity initial conditions. Here are also computed the longitudinal velocity V_x and acceleration A_x. There is even the *tau* value which represents the gear engaged, in this thesis we are supposing to have a moving vehicle with $tau = 2$. The whole system is shown in the following picture.

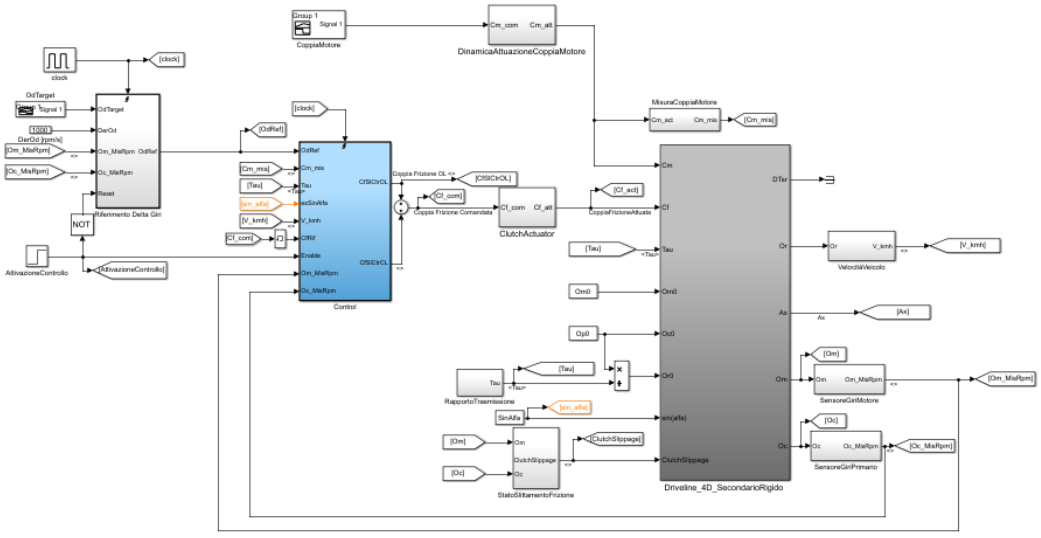

Figure 1.16: DDCT system Simulink scheme.

1.4.5.1 Control

The control block is composed of two different controllers: a static controller, derived from the plant differential equations, and a closed loop controller which we will substitute with the designed controller. Before starting design, we choose as closed loop controller a preliminary PI. In the following figure is shown the static controller which inputs are C_m ,V_x , w_m (measured in RPM), tau and $\sin(\alpha)$. The output is C_{fOL}.

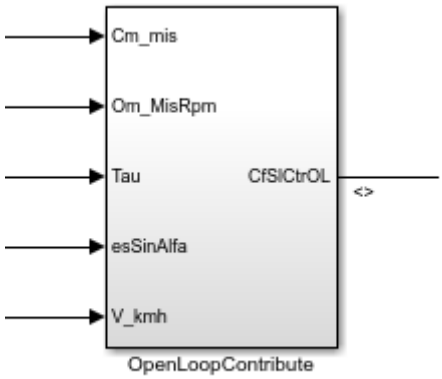

Figure 1.17: Static controller block.

The final control torque is $C_f = C_{fOL} + C_{fCL}$. Where C_{fCL} is the control torque provided by closed loop designed controllers.

1.5 Simplified model

In order to design a controller a model of the plant it is required. The detailed model seen before is not suitable to accomplish the design project, therefore a simplified model is necessary. CRF provides us differential equations which describe the system, derived from the following driveline structure:

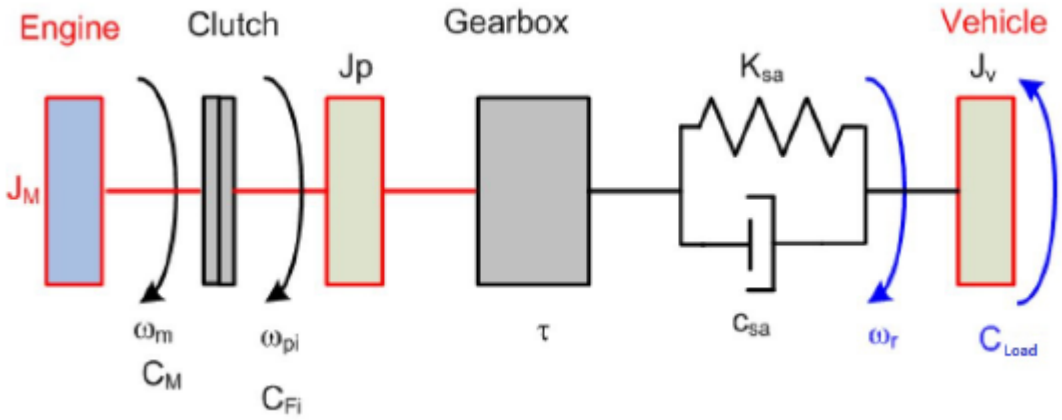

Figure 1.18: Driveline scheme.

The differential equations are:

$$J_m \dot{w}_m = C_m - c_m w_m - C_f$$

$$J_p \dot{w}_p = C_f + \frac{1}{\tau}(-c_{sa} w_{sr} - k_{sa} \theta_{sr}) \tag{1.4}$$

$$J_v \dot{w}_r = c_{sa} w_{sr} + k_{sa} \theta_{sr} - C_{load}$$

$$\dot{\theta}_{sr} = w_{sr}$$

where:

$$\theta_{sr} = \theta_{sa} - \theta_r$$

$$w_{sr} = \frac{wp}{\tau} - w_r$$

Finally:

$$\dot{w}_m = \frac{1}{J_m}(C_m - C_f) - \frac{c_m}{J_m} w_m$$

$$\dot{w}_p = \frac{C_f}{Jp} - (\frac{c_{sa}}{J_p \tau^2} + \frac{c_{prim}}{J_p}) w_p + \frac{c_{sa}}{J_p \tau} w_r - \frac{k_{sa}}{J_p \tau} \theta_{sr}$$

$$\dot{w}_r = \frac{c_{sa}}{J_p \tau} w_p - \frac{c_{sa}}{J_v} w_r + \frac{k_{sa}}{J_v} \theta_{sr} - \frac{C_{load}}{J_v}$$

$$\dot{\theta}_{sr} = \frac{w_p}{\tau} - w_r$$

We can rewrite this system in the state space representation:

$$\dot{x}(t) = Ax(t) + Bu(t)$$
$$y(t) = Cx(t) + Du(t)$$

Making the following assumptions:

- C_f is the manipulated input;
- C_m and C_{load} are seen as disturbances;
- the state vector is $x(t) = [w_m(t), w_p(t), w_r(t), \theta_{sr}(t)]^T$;
- the input vector is $u(t) = [C_m(t), C_f(t), C_{load}(t)]^T$;
- the outputs vector is $y(t) = [w_d(t), w_{sr}(t)]$, where $w_d(t) = w_m(t) - w_p(t)$ is the slipping velocity and $w_{sr}(t) = \frac{w_p(t)}{\tau} - w_r(t)$ is the torsion speed.

The matrices are:

$$A = \begin{bmatrix} -\frac{c_m}{J_m} & 0 & 0 & 0 \\ 0 & -(\frac{c_{sa}}{J_p\tau^2} + \frac{c_{prim}}{J_p}) & \frac{c_{sa}}{J_p\tau} & -\frac{k_{sa}}{J_p\tau} \\ 0 & \frac{c_{sa}}{J_v\tau} & -\frac{c_{sa}}{J_v} & \frac{k_{sa}}{J_v} \\ 0 & \frac{1}{\tau} & -1 & 0 \end{bmatrix}$$

$$B = \begin{bmatrix} \frac{1}{J_m} & -\frac{1}{J_m} & 0 \\ 0 & \frac{1}{J_p} & 0 \\ 0 & 0 & -\frac{1}{J_v} \\ 0 & 0 & 0 \end{bmatrix}$$

$$C = \begin{bmatrix} 1 & -1 & 0 & 0 \\ 0 & \frac{1}{\tau} & -1 & 0 \end{bmatrix}$$

$$D = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

CRF provides also a table containing the parameters used to model the system with their unit of measurement:

Torsional damper constant	C_{sa}
Torsional spring constant	K_{sa}
Axle gear ratio	τ
Rolling radius	r_{rot} (m)
Wheel inertia	J_r (Kg.m^2)
Vehicle inertia	J_v (Kg.m^2)
Motor damper constant	c_m
Motor inertia	J_m (Kg.m^2)
Road slope	$\sin(\alpha)$
Gravity	g (m/s^2)
Primary shaft inertia	j_p (Kg.m^2)

Figure 1.19: Table containing driveline parameters.

1.5.1 Plant architecture

As we can see in the previous section, the system is composed of two outputs (w_d, w_{sr}). In this thesis, during design we will consider only w_d as output. Basically we used three plant architectures in order to understand which one provided the best approximation compared with the detailed model:

- Plant built using elementary blocks

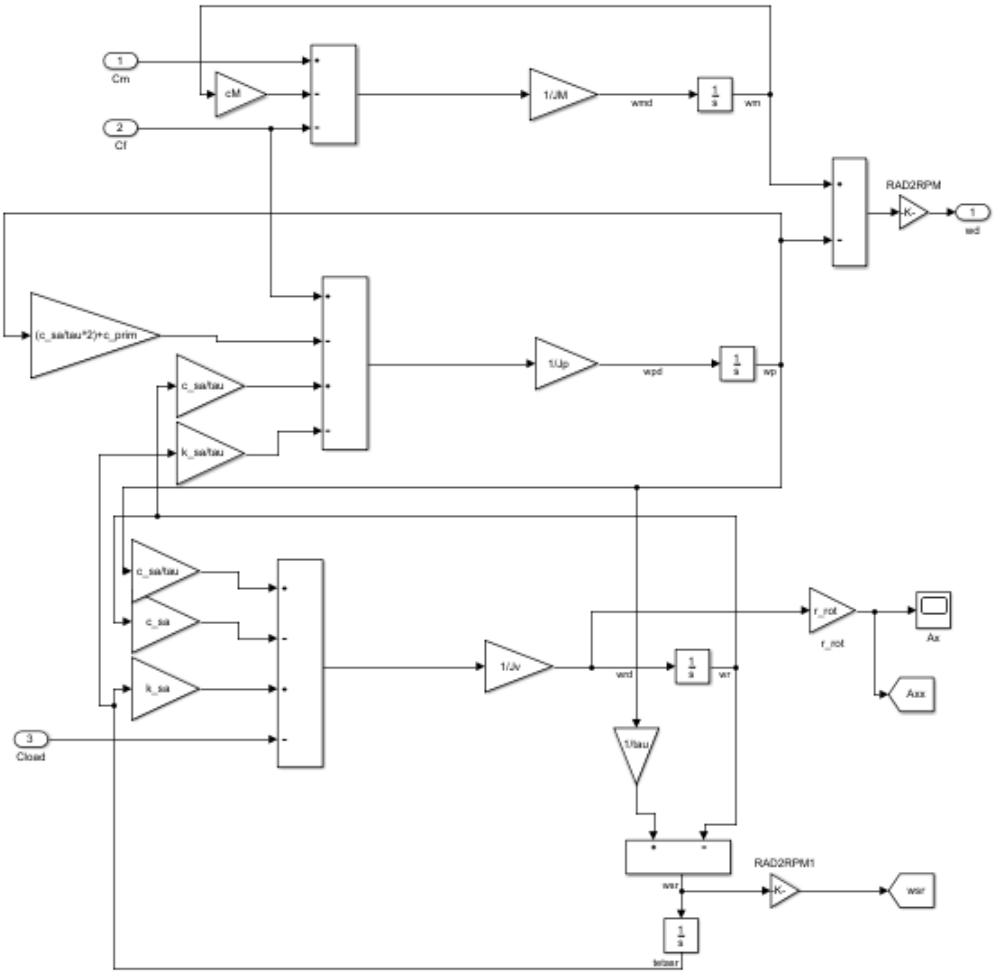

Figure 1.20: Plant represented with elementary blocks.

- Plant in state-space representation

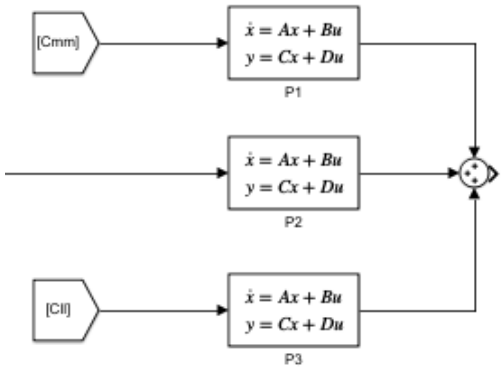

Figure 1.21: Plant represented with state space blocks.

where P_1 is the state space model with the following matrices: $A, B(:,1), C, D(:$

, 1), P_2 is composed by: $A, B(:, 2), C, D(:, 2)$ and P_3 contains: $A, B(:, 3), C, D(:$, 3). Using this architecture it's possible to insert even the initial condition.

- Plant constructed by the use of linear time invariant (LTI) block

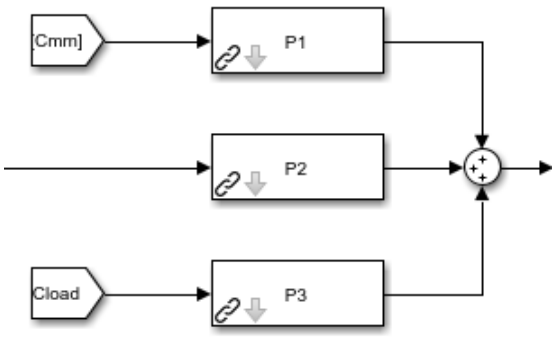

Figure 1.22: Plant represented with LTI blocks.

in this case P_1, P_2 and P_3 are LTI transfer function, we will see their value in the following section.

Before starting to design the controller, we have to check if these simplified linear models are able to guarantee a good approximation of the detailed model. In the below figure we can see the outputs of the models obtained after simulation.

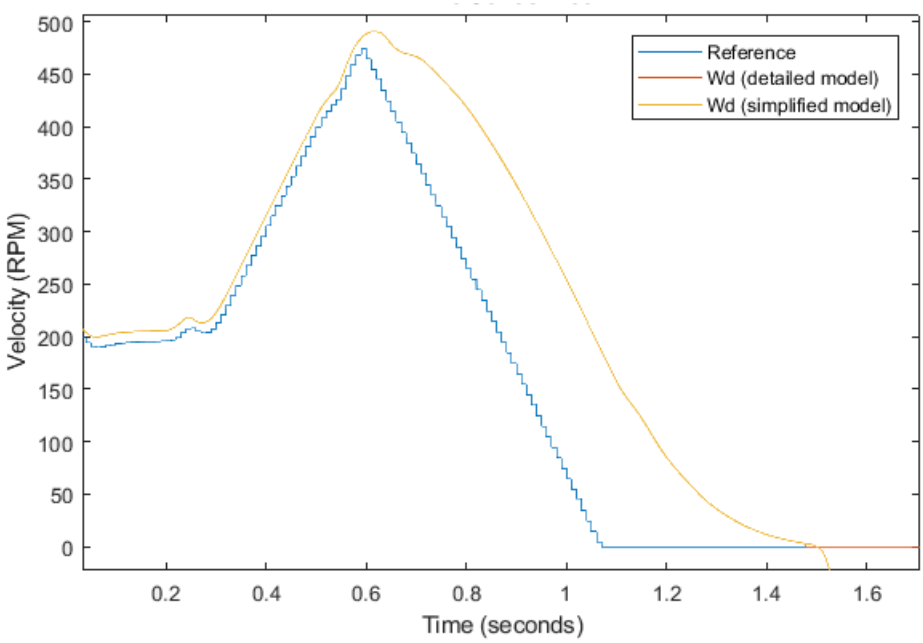

Figure 1.23: Slipping speed w_d for both detailed and simplified model.

They are all equal during the slipping phase, hence we conclude that the linear model can be used to design a controller.

1.6 Control system architecture

Basically we used three different control system architecture in order to find the best solution:

- Base,
- Feedforward,
- Feedforward with a new reference input.

1.6.1 Base

The Base control system architecture is composed as follows.

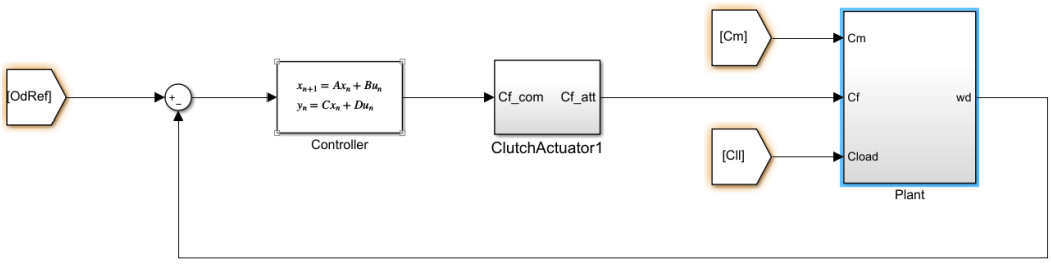

Figure 1.24: Base architecture.

Using the reference input provided by CRF.

1.6.2 Feedforward

The feedforward concept is to compensate the contribution of the disturbances in order to guarantee a better tracking of the reference input. A general scheme of feedforward control is presented in the figure below

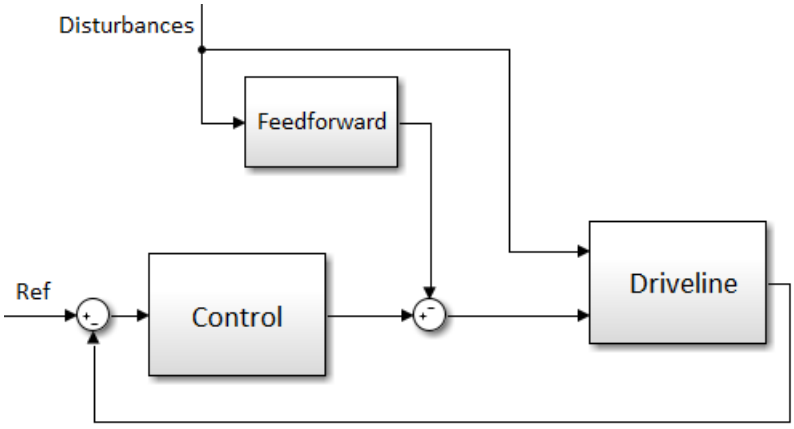

Figure 1.25: General feedforward scheme.

Feedforward control can be used to improve significantly a control loop's response to disturbance. The advantage are:

- reacts the moment when the disturbance occurs;
- does not alter the system stability;

while the disadvantage are:

- disturbance must be measurable;
- requests the system model.

In order to provide a Feedforward control system architecture we have to split the plant into three parts, one for each input. The Matlab command used in this case is *linmod*. We obtain:

1. $P_1 = \frac{4}{s}$
2. $P_2 = \frac{-103.43(s+0.005905)(s^2+5.344s+334.4)}{s(s+0.006733)(s^2+121.4s+7696)}$
3. $P_3 = \frac{7.2217(s+63.99)}{(s+0.006733)(s^2+121.4s+7696)}$

Where P_1, P_2, and P_3 are LTI transfer functions respectively between the inputs C_m, C_f, C_{load} and the output w_d.

We have the following situation:

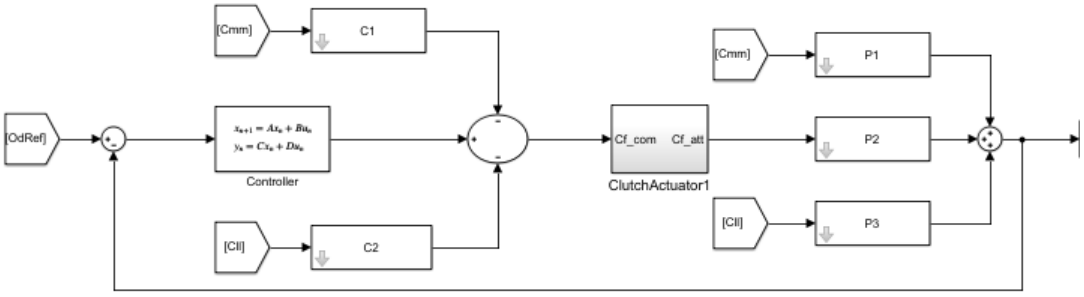

Figure 1.26: Feedforward control scheme.

where $C_1$and C_2 are transfer functions used to compensate the disturbance. We have to find their value using a simple mathematical equation:

$$y = C_m P_1 + C_{load} P_3 + P_2 G_a(-C_m C_1 - C_{load} C_2 + G_c Ref - G_c y);$$

$$y = C_m P_1 + C_{load} P_3 + P_2 G_a(-C_m C_1 - C_{load} C_2 + G_c Ref) - P_2 G_a G_c y;$$

$$y(1 + P_2 G_a G_c) = C_m P_1 + C_{load} P_3 - P_2 G_a C_1 C_m - P_2 G_a C_{load} C_2 + P_2 G_a G_c Ref;$$

$$y = [C_m(P_1 - P_2 G_a C_1) + C_{load}(P_3 - P_2 G_a C_2) + P_2 G_a G_c Ref] \frac{1}{1 + P_2 G_a G_c}$$

To get rid C_m and C_{load} contributions:

$$\begin{cases} P_1 - P_2 G_a C_1 = 0 & \rightarrow \begin{cases} C_1 = \frac{P_1}{P_2 G_a} \end{cases} \\ P_3 - P_2 G_a C_2 = 0 & \rightarrow \begin{cases} C_2 = \frac{P_3}{P_2 G_a} \end{cases} \end{cases}$$

1.6.3 Feedforward with a new reference input

In this system architecture we change the reference input using a new signal. This new reference input has the shape of an hyperbolic tangent, modified in order to get a desired shape.

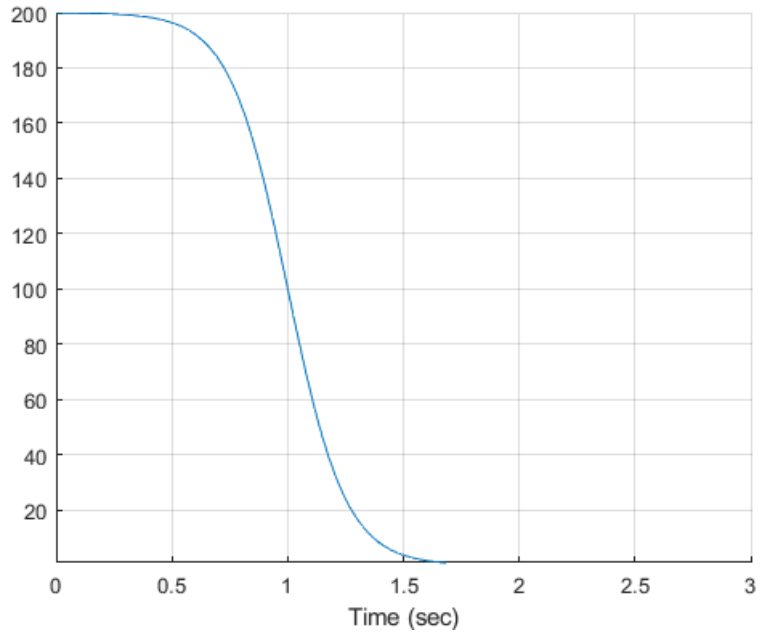

Figure 1.27: tanh shape reference input.

As shown in figure above, this reference input goes to zero smoother with respect to CRF's input. Furthermore we can decide the time where this signal goes to 0, obtaining different results.

Chapter 2

Loop-shaping approach

2.1 Overview

Loop-shaping (LS) is a method to design a controller using frequency approach. To understand the main concept of LS we have to introduce the Nyquist Stability Theorem [5]. It allows us to determine whether a system is stable or unstable furnishing a measure of the degree of stability through the definition of stability margin. Nyquist criterion also indicates how an unstable system should be changed to make it stable. LS approach leads to stabilize the unstable systems. Considering a general system in Figure 2.1

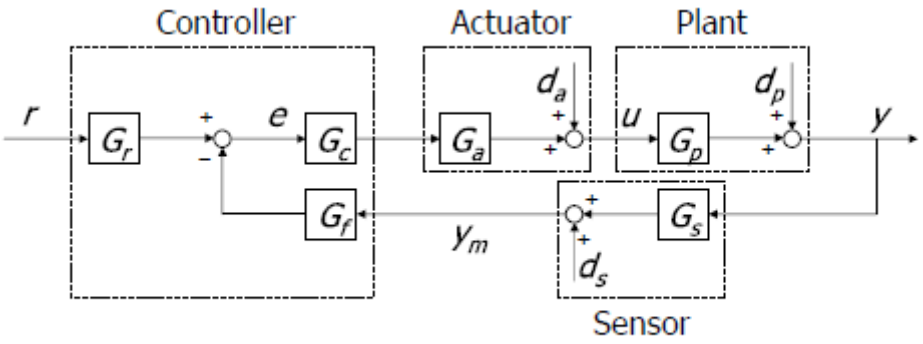

Figure 2.1: General control system scheme.

the loop transfer function $L(s)$ is by definition the product of all blocks inside the loop, $L(s) = G_c(s)G_a(s)G_p(s)G_s(s)G_f(s)$. A powerful concept of the Nyquist approach is that it allows to study the stability of the feedback system by looking at properties of the loop transfer function. In this way it is easy to see how the controller $G_c(s)$ should be chosen to obtain a desired loop transfer function. In

particular the designer can modify controller $G_c(s)$ adding zeros and poles properly even using particular networks: lead and lag. Another chart that make easier the design of the controller is the Nichols plot. It is a plot of the logarithmic magnitude in decibels versus the phase angle for a frequency range of interest. Nichols plot provides a better representation of the stability system, thus the compensation can be worked out easily. We will see how to do that later in the following section. For other info on Nyquist and Nichols plot see the reference in bibliography. Considering the general control scheme in the above figure, the control system in this thesis is composed by $G_c(s)$ that is the controller to design, $G_a(s)$ that is the actuator having transfer function seen in the first chapter and $G_p(s)$, the plant under investigation.

2.2 Design

Before starting the design it is better to introduce the sensitivity function $S(s)$ and the inverse sensitivity function $T(s)$. They are strictly related to the loop function $L(s)$ and play a significant role in the design:

$$S(s) = \frac{1}{1+L(s)},\ T(s) = \frac{L(s)}{1+L(s)} \Rightarrow S(s) + T(s) = 1$$

Another fundamental parameter in a control system is the crossover frequency ω_c of the loop function, because discriminates the low and high frequency range:

- $\omega \ll \omega_c \rightarrow$ low frequency range, typically $|L(j\omega)| \gg 1$,
- $\omega \gg \omega_c \rightarrow$ high frequency range, typically $|L(j\omega)| \ll 1$,
- $\omega \approx \omega_c \rightarrow$ medium frequency range.

Further it is very important to underline that the phase behavior of $L(j\omega)$ ($\angle L(j\omega)$) at medium frequency determines the stability characteristic of the closed loop system. During design, will be considered two steady-state gain K_c and K_p defined as:

- $\lim_{s\to 0} s^{\mu} G_c(s) = K_c$, so the controller has μ poles at $s = 0$;
- $\lim_{s\to 0} s^{p} G_p(s) = K_p$, so the plant has p poles at $s = 0$.

We assume to want to reach the following time specifications:

- Rise time: $t_r < 0.25\ s$
- Overshoot: $\hat{s} < 14\%$
- Steady-state output error in presence of $C_m$and C_{load}: $|e_{d_p}^{\infty}| = 0$

The second order prototype model is used to translate time requirements into the relevant indices of the frequency response of the T(s), S(s) and L(s) functions [6].

$$T(s) = \frac{w_n^2}{s^2+2\zeta w_n s+w_n^2}$$

Translation of the specifications:

- $\hat{s} < 14\% \rightarrow \zeta \geq \frac{|\ln(\hat{s})|}{\sqrt{\pi^2+\ln^2(\hat{s})}} \rightarrow \zeta = 0.53$
- S_p : Maximum sensitivity $\rightarrow \underset{\omega\in[0,\infty]}{max}|S(j\omega)| \leq \frac{2\zeta\sqrt{2+4\zeta^2+2\sqrt{1+8\zeta^2}}}{\sqrt{1+8\zeta^2}+4\zeta^2-1} = 1.43$
- T_p : Maximum complementary sensitivity $\rightarrow \underset{\omega\in[0,\infty]}{max}|T(j\omega)| \leq \frac{1}{2\zeta\sqrt{1-\zeta^2}} = 1.11$
- $t_r = 0.25 \rightarrow t_r * \omega_c = 2.02$ (value taken using the graph $t_r * \omega_c$ versus ζ (Figure 2.2)) $\rightarrow \omega_c = 8$ rad/s.
- $|e_{d_p}^{\infty}| = 0 \rightarrow$Applying the final-value theorem, just considering C_m as disturbance:

$$|e_{d_p}^{\infty}| = \lim_{t\to\infty}|e^{dp}(t)| = \lim_{s\to 0}s|e^{dp}(s)| = \lim_{s\to 0}s|y^{dp}(s)| = \lim_{s\to 0}s|\frac{1}{1+L}d_p(s)| =$$

$$= \lim_{s\to 0}\ s\big|P_1(s)\frac{s^{\mu+p}}{s^{\mu+p}+K_cK_aK_p}\frac{D_{p0}}{s^{h_p+1}}\big|$$

where $h_p = 0$ because the disturbance is seen as a step. So to have $|e_{d_p}^{\infty}| = 0$ the following relation must be true $\mu + p \geq 2$. p is already equal to 1 because the plant has a pole in the origin, hence $\mu \geq 1$. We choose $\mu = 1$. Furthermore there is no constraint on $|K_c|$, this allows us to change K_c freely taking into account that an increase of that gain leads to move $L(s)$ upward, while a decrease involves a movement downward. Making the same computation using C_{load} instead C_m the result in terms of constraints does not change. Note that the rise time t_r provides a lower bound on crossover frequency ω_c. The crossover frequency of a system is an essential parameter because describes stability and speed. We can see the value of ω_c in the Nichols plot, when the loop function crosses the 0 dB axis. The

computed maximum values S_p and T_p will be converted in circle constraints visible in the Nichols chart and are very useful for the graphically design. To ensure good performance the loop function has no cross the circles.

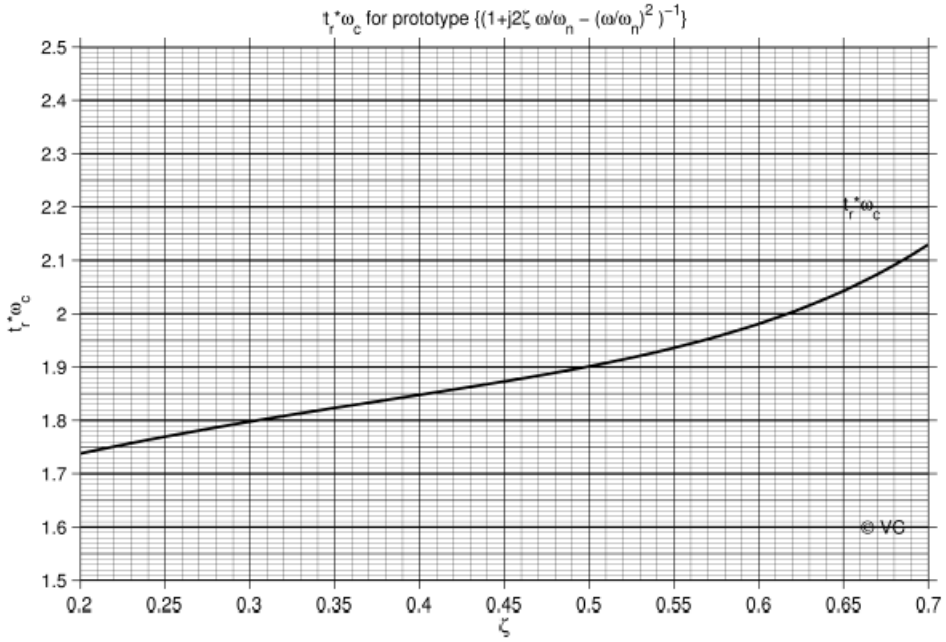

Figure 2.2: Relation between $t_r * \omega_c$ and ζ (from [8]) .

2.2.1 Sign of Kc

To guarantee the stability of the feedback system the Nyquist stability is used as mentioned in [5]. We have to pay attention to this relation:

$$P_{cl} = P_{ol} + N$$

where:

- P_{cl} is the number of roots of the equation $1 + L(s) = 0$ with positive real part;
- P_{ol} is the number of poles of $L(s)$ with real part > 0;
- N is the number of encirclements of the Nyquist plot of $L(j\omega)$ around the point $-1 + j0$, computed as the difference between the number of clockwise encirclements and number of counterclockwise encirclements.

From translation of specifications we have not found constraints on the controller dc gain K_c, so as first choice we put $K_c = 1$. Plotting the Nyquist chart using the Matlab command *nyquist* (Figure 2.3)

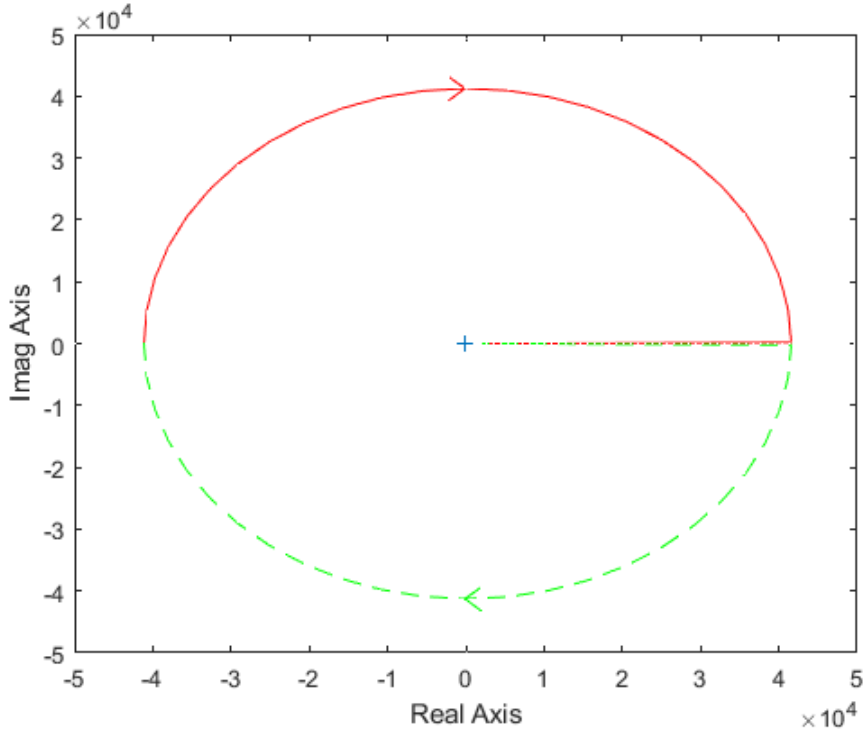

Figure 2.3: Nyquist plot with $K_c = 1$.

where the blue cross indicates the critical point $-1 + j0$, we can see that $N = 1$, $P_{ol} = 0$ thus $P_{cl} = 1$. Hence, taking K_c positive the system is unstable. With $K_c = -1$ instead we obtain the plot in Figure 2.4

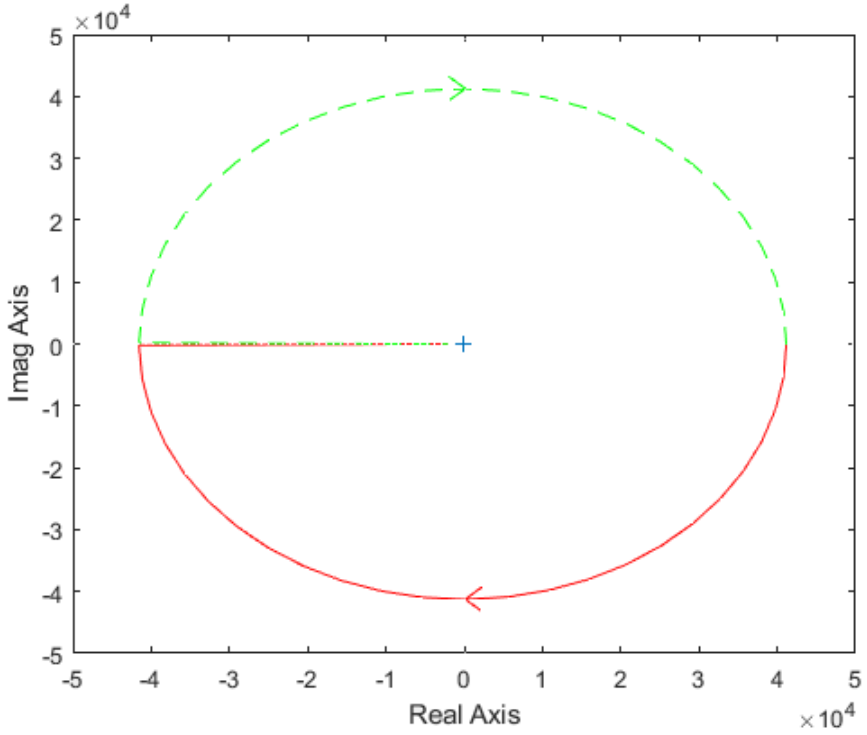

Figure 2.4: Nyquist plot with $K_c = -1$.

Zooming the figure around the blue cross it is easier to see that $P_{ol} = 0, N = 2$ so $P_{cl} = 2$, hence the system is not stable but can be stabilizable according to Nyquist stability criterion. We can conclude that the sign of K_c must be negative.

2.2.2 Lead and Lag Compensators

The lead and lag networks are particular transfer functions utilized to improve an undesirable frequency response in a feedback control system. Both introduce a pole-zero pair into the open loop transfer function:

- Lead network has the following structure:

$$R_d(s) = \frac{1+\frac{s}{z_d}}{1+\frac{s}{m_d z_d}}, m_d > 1$$

The effects are: phase lead and magnitude increase, we can see the graphic contribution in the figure below.

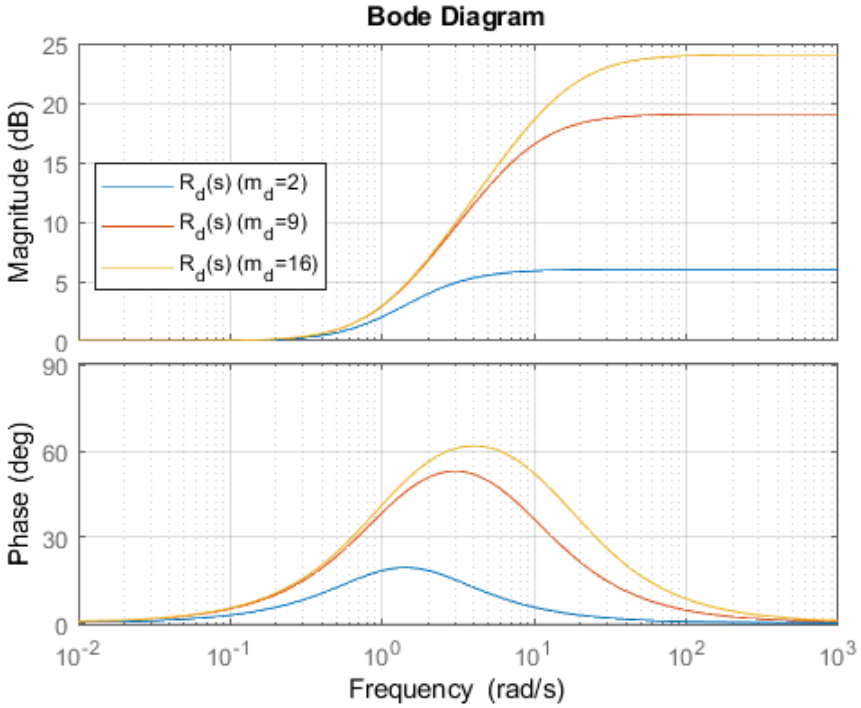

Figure 2.5: Lead network magnitude contribution.

In the lead network the contribution of the zero comes before than the contribution of the pole.

- Lag network with this structure:

$$R_i(s) = \frac{1+\frac{s}{m_i p_i}}{1+\frac{s}{p_i}}, m_i > 1$$

The effects are: magnitude attenuation and phase lag, the contribution is graphically shown in the following figure:

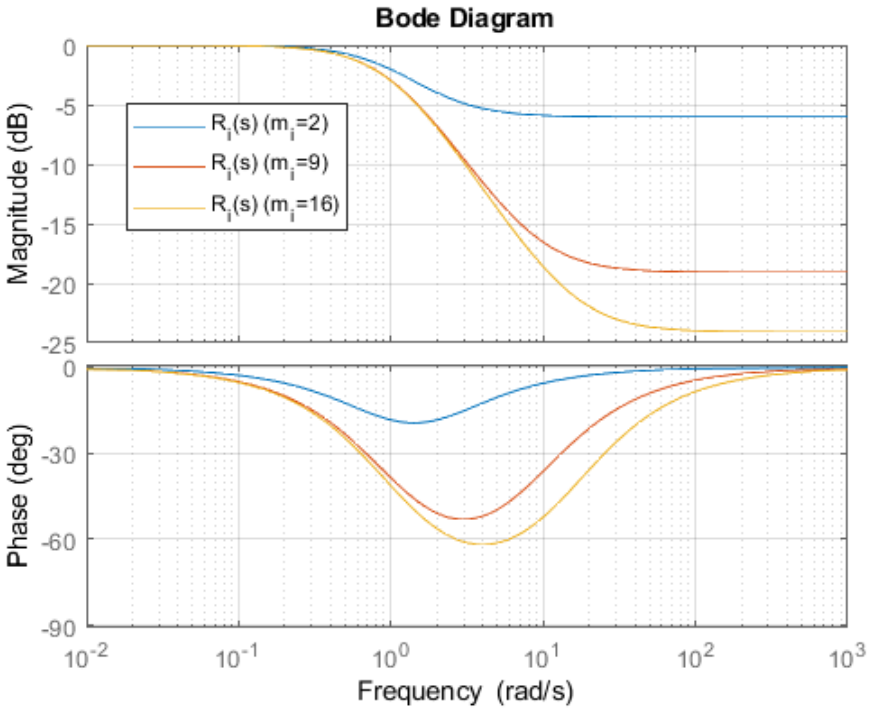

Figure 2.6: Lag network magnitude contribution.

In the lag network instead, the contribution of the zero comes after than the pole contribution.

As already said, Nyquist plot provides gain and phase margins, gain margin tells how much the gain has to be increased before the closed loop system becomes unstable, while the phase margin tells how much the phase lag has to be increased to make the closed loop system unstable. Plotting the Nichols chart of the loop function $L(s)$ with the Matlab command *nichols*

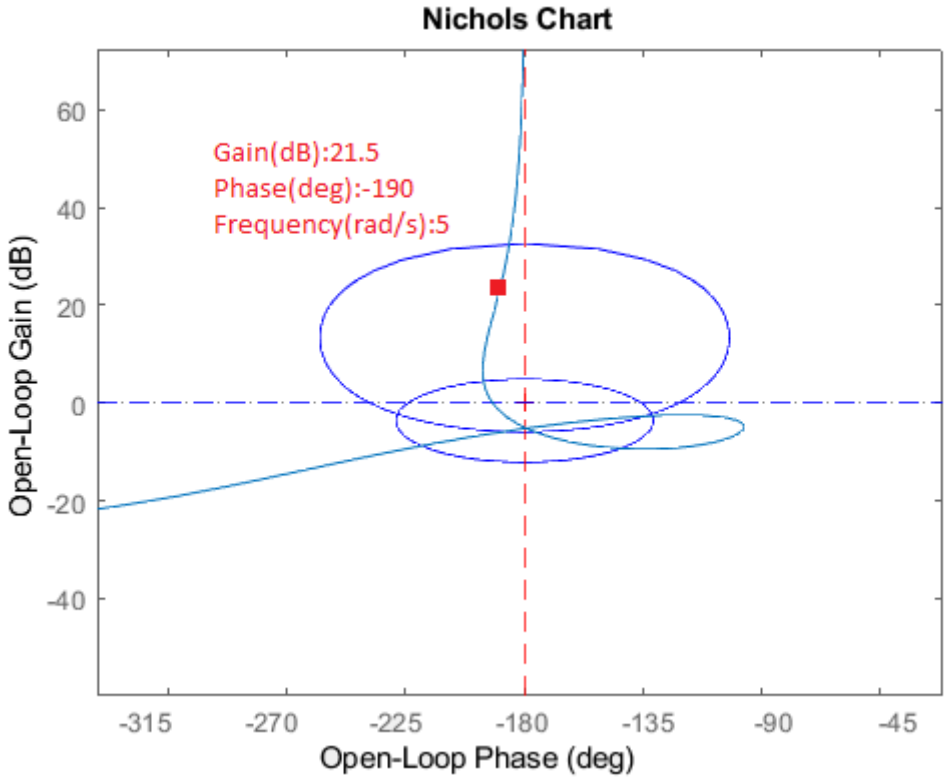

Figure 2.7: Nichols chart with $G_c(s) = \frac{K_c}{s}$.

we can see that the $L(s)$ is very close to the point $-1 + j0$. We can also notice that the system has a crossover frequency different from the desired one. To ensure that the system has the desired crossover frequency with a good phase and gain margins we have to add some compensators to the controller. In figure 2.7 we can see that an increase of phase (70°) is needed in order to move the loop function far from the critical point and the circles. Usually it is convenient to use more than one lead network with a medium value instead just one with the greater phase value. In the follows the computation of the first lead network in order to gain 40°:

- $m_d = 16$;
- Normalized frequency $\frac{\omega}{z_d} = 10^0 = 1 \rightarrow z_d = \frac{\omega_c}{1}$ to stay on the left part of the bell (Figure 2.5) avoiding to add a significant quantity of gain.

Moving on in this way the final result is in Figure 2.8. The final controller is composed by two lead network, one lag network and $|K_c| = 0.3$:

$$G_{c1}(s) = \frac{-4.8(s+9.6)(s+1)^2}{s(s+16)^2(s+0.6)} \qquad (2.1)$$

Looking at Nichols chart we can see that the loop function is far from the critical point but crosses the circle:

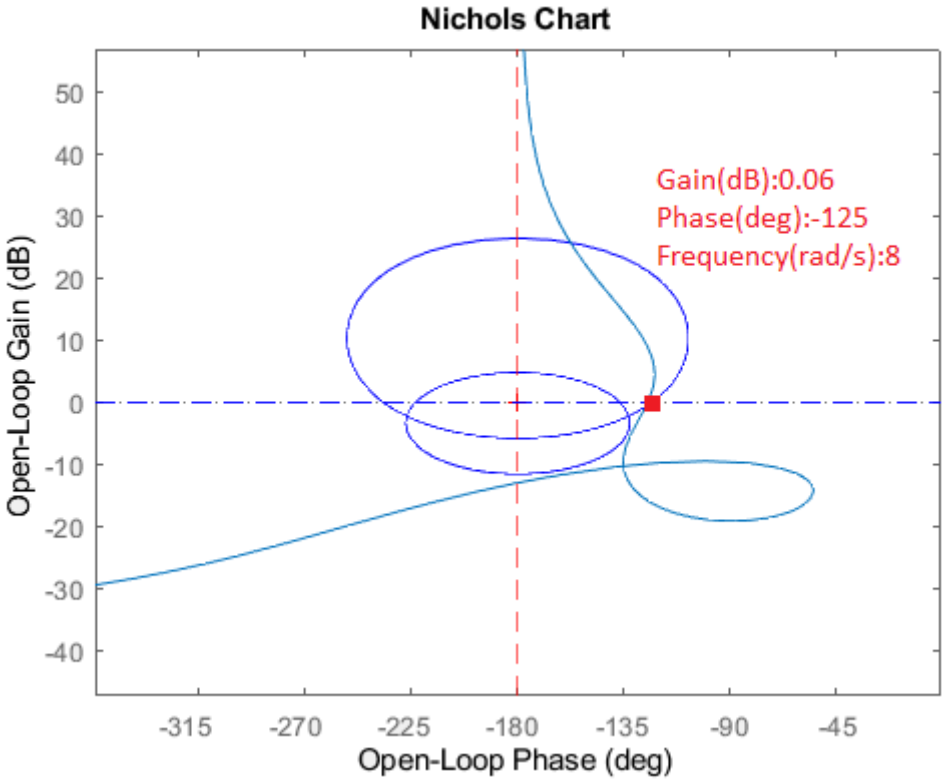

Figure 2.8: Nichols chart with $G_{c1}(s)$.

In this case only the rise time fulfills the fixed time specifications. The order of $L(s)$ is very high, so it is difficult to find a good shape for the loop function using only these two networks. For this reason we used the tool already present in Matlab: *Control System Designer*. In this case we try to have $t_r = 0.6\ s \rightarrow \omega_c = 3.5\ rad/s$, obtaining:

$$G_{c2}(s) = \frac{-0.012057(s+0.8361)(s^2+74.29s+2393)}{s(s^2+5.325s+339.3)} \tag{2.2}$$

The relative Nichols plot and step response are:
We can see that the specifications on the step time are fulfilled:

- $t_r = 0.52 < 0.6$
- $\hat{s} = \frac{y_{max}-y_\infty}{y_\infty} = \frac{1.13-1}{1} = 0.13 = 13\% < 14\%$

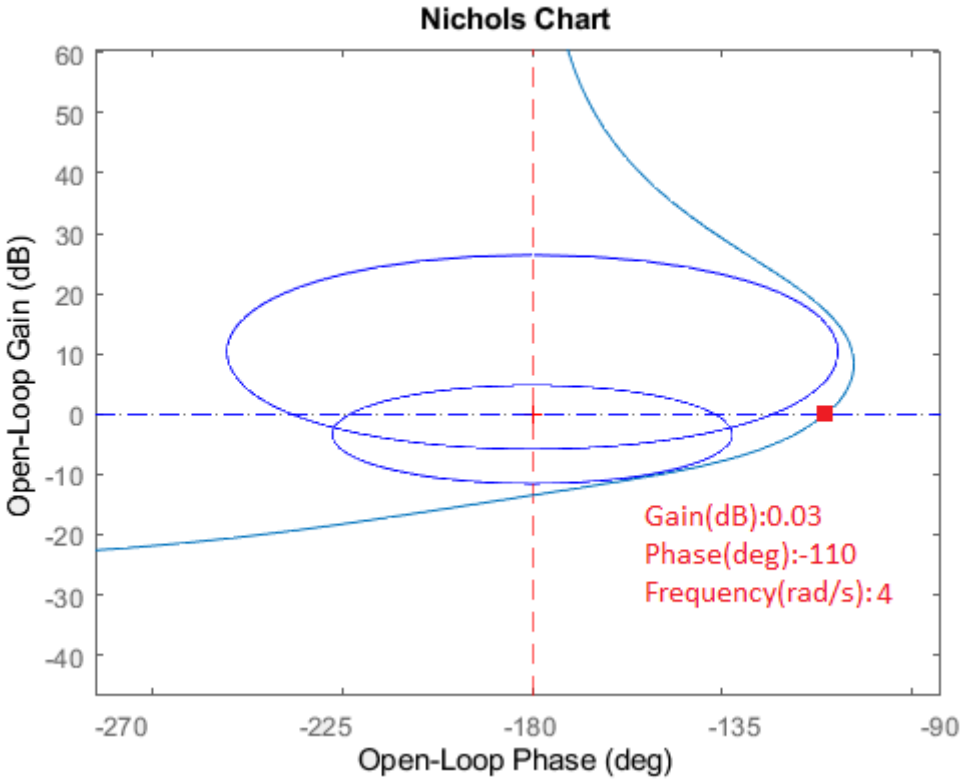

Figure 2.9: Nichols plot with $G_{c2}(s)$.

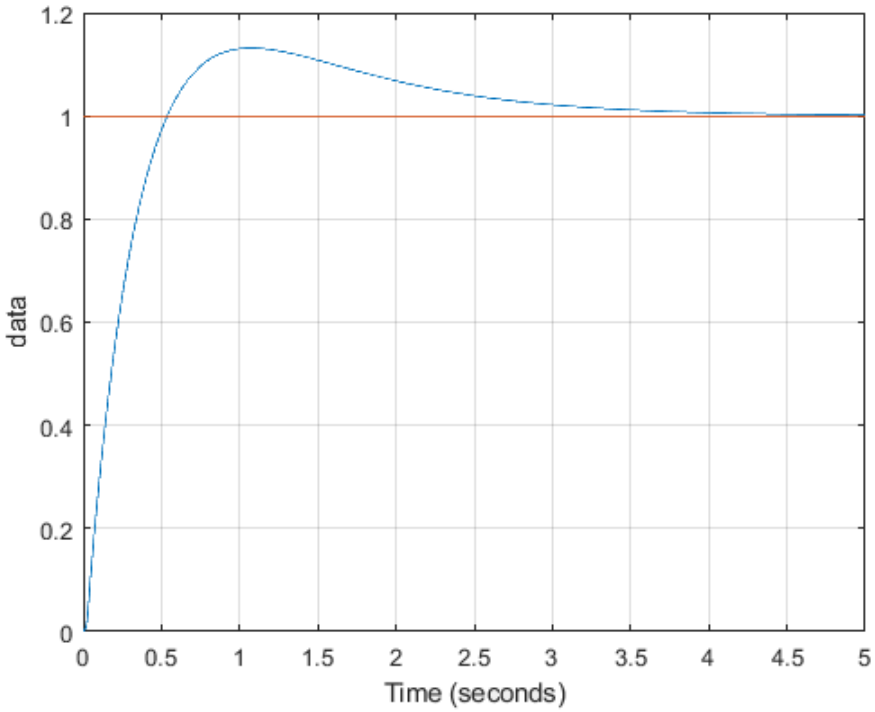

Figure 2.10: Step response with $G_{c2}(s)$.

The obtained controller is expressed in the s domain, so we have a continuous-time transfer function. The control unit works in discrete time, hence we have to discretize the controller with a sampling time equal to $T_s = 10\ ms$. After applying the *c2d* command on Matlab we obtain the controller in z domain:

$$G_c(z) = \frac{-0.012057(z-0.9917)(z^2-1.144z+0.3769)}{(z-1)(z^2-1.915z+0.9481)}$$

This is the workflow to obtain a controller, we made several attempts getting different controllers and results. The most useful controllers will be show in the following.

2.3 Loop shaping controller on Base architecture

Now we want to test the designed controllers through the Base architecture (Figure 1.23). Note that in all figures the time axis starts before 0.6 seconds because we are not interested in the initial transient values, furthermore we are focused on the response after 0.6, seconds that is the time in which the control will be activated in the detailed model. The controllers employed in these simulations are nine separated in two groups, three derived from (2.1) and six derived from (2.2):

- $G_{cLS1}(s) = \frac{-1.6(s+9.6)(s+1)^2}{s(s+16)^2(s+0.6)}$
- $G_{cLS2}(s) = \frac{-4.8(s+9.6)(s+1)^2}{s(s+16)^2(s+0.6)}$
- $G_{cLS3}(s) = \frac{-10(s+9.6)(s+1)^2}{s(s+16)^2(s+0.6)}$
- $G_{cLS4}(s) = \frac{-0.0060285(s+0.8361)(s^2+74.29s+2393)}{s(s^2+5.325s+339.3)}$
- $G_{cLS5}(s) = \frac{-0.0072342(s+0.8361)(s^2+74.29s+2393)}{s(s^2+5.325s+339.3)}$
- $G_{cLS6}(s) = \frac{-0.0096456(s+0.8361)(s^2+74.29s+2393)}{s(s^2+5.325s+339.3)}$
- $G_{cLS7}(s) = \frac{-0.012057(s+0.8361)(s^2+74.29s+2393)}{s(s^2+5.325s+339.3)}$
- $G_{cLS8}(s) = \frac{-0.014468(s+0.8361)(s^2+74.29s+2393)}{s(s^2+5.325s+339.3)}$
- $G_{cLS9}(s) = \frac{-0.018086(s+0.8361)(s^2+74.29s+2393)}{s(s^2+5.325s+339.3)}$

Plot name	Controller	t_r(sec)	$\hat{s}$
LS1	$G_{cLS1}(s)$	0.55	27%
LS2	$G_{cLS2}(s)$	0.25	23%
LS3	$G_{cLS3}(s)$	0.16	21%
LS4	$G_{cLS4}(s)$	0.8	20%
LS5	$G_{cLS5}(s)$	0.75	18%
LS6	$G_{cLS6}(s)$	0.6	15%
LS7	$G_{cLS7}(s)$	0.52	13%
LS8	$G_{cLS8}(s)$	0.47	11%
LS9	$G_{cLS9}(s)$	0.4	10%

Table 2.1: Loop Shaping designed controllers.

For what concern this architecture the results are not good because the disturbance C_m is quite significant.

2.4 Loop shaping controller on Feedforward architecture

In the follows we show the results obtained using the Feedforward architecture. Here the results are better because the contribution of C_m is compensated.

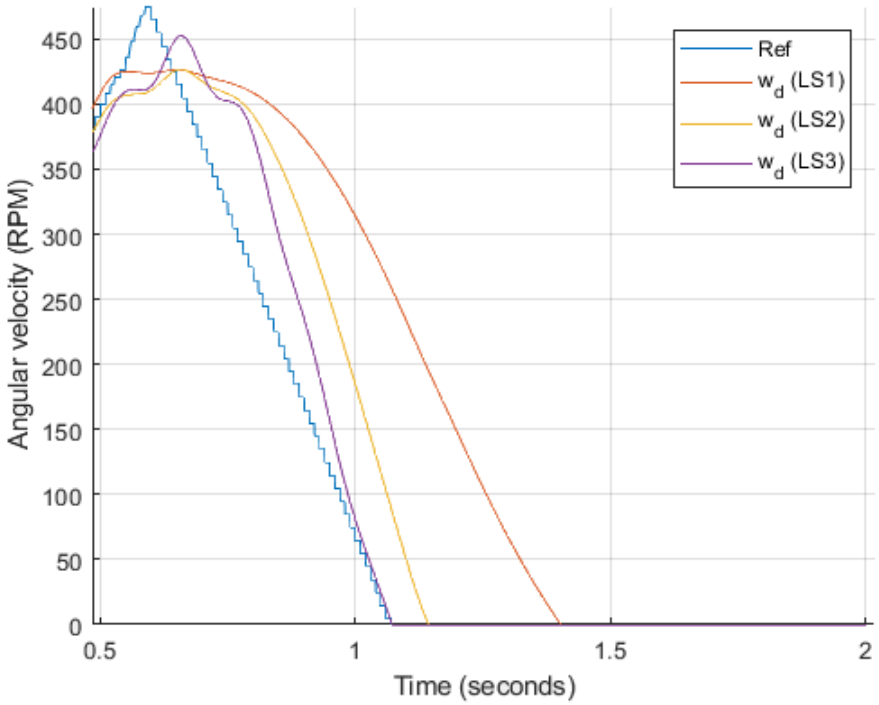

Figure 2.11: w_d responses with the first three controllers.

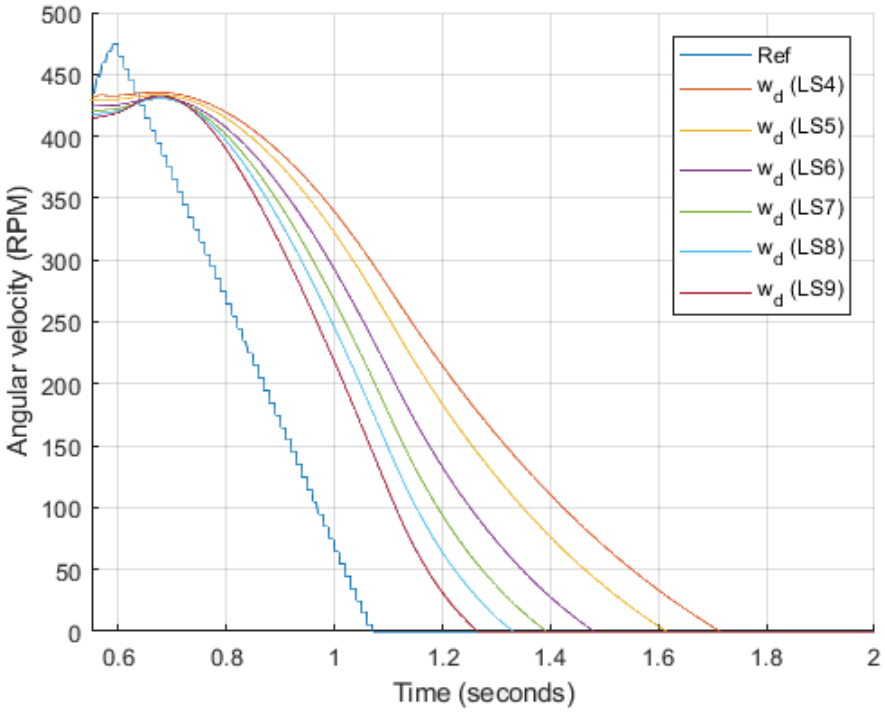

Figure 2.12: w_d responses with the last six controllers.

the relative values of w_{sr} and A_x are:

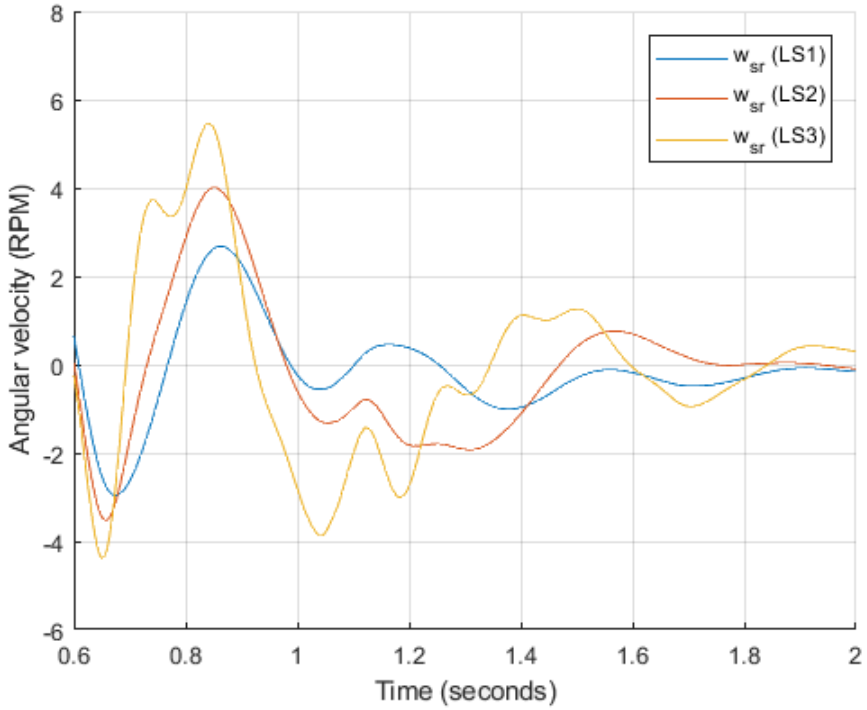

Figure 2.13: w_{sr} outputs with first three controllers.

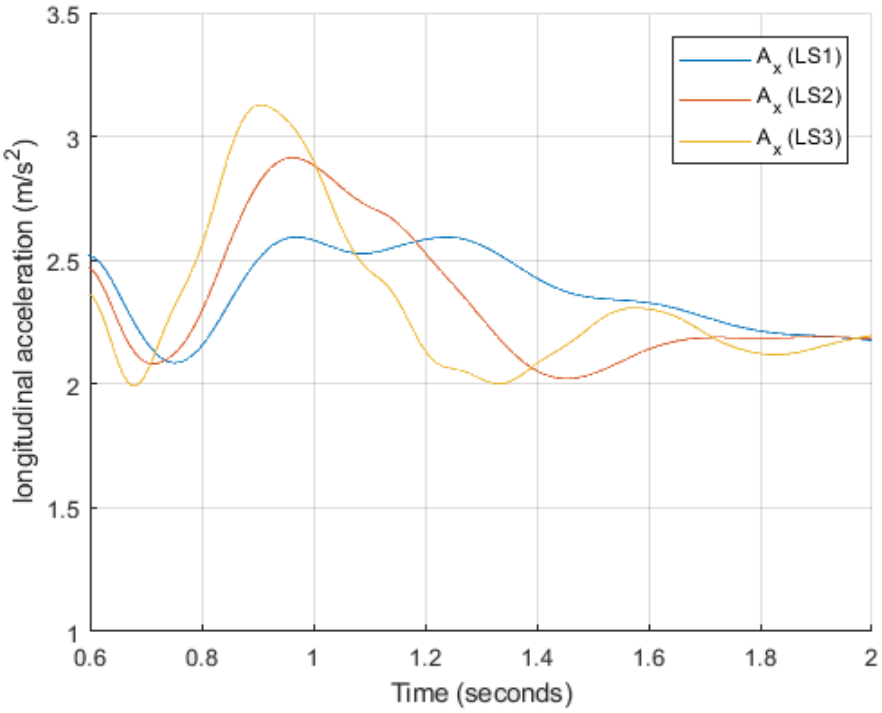

Figure 2.14: A_x values with the first three controllers.

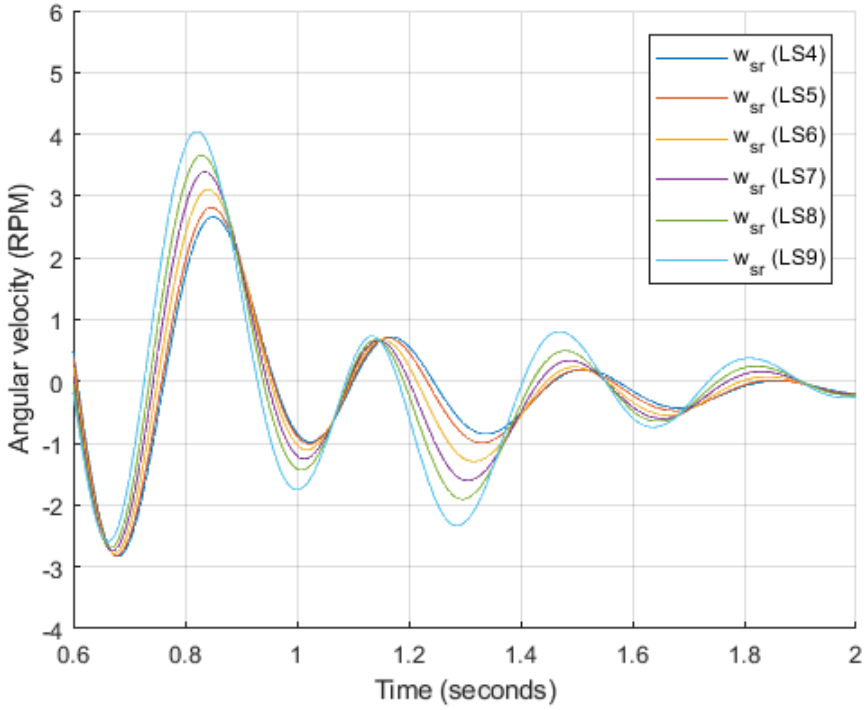

Figure 2.15: w_{sr} outputs with last six controllers.

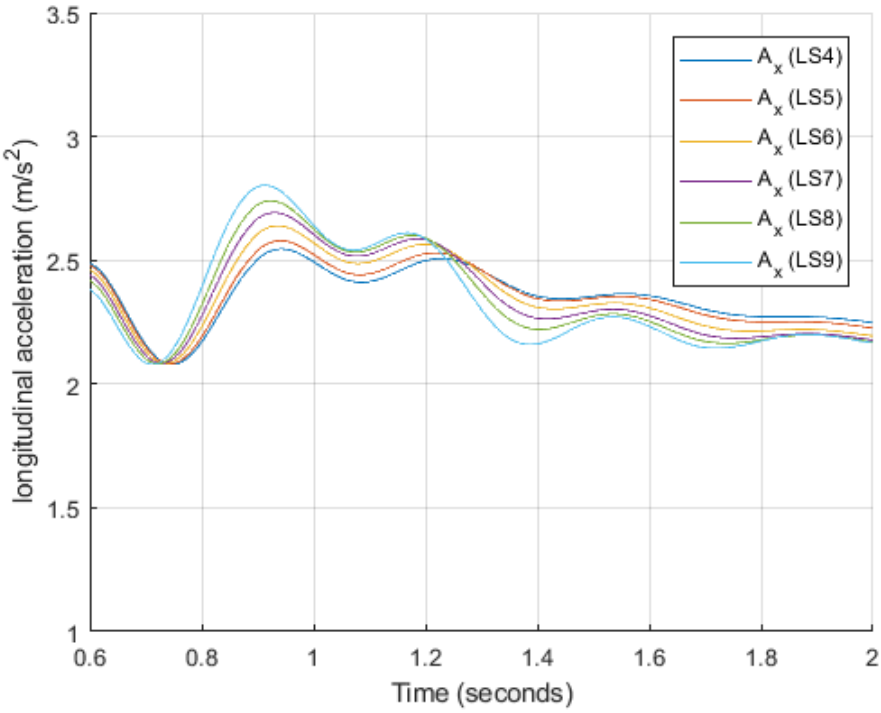

Figure 2.16: A_x values with the last six controllers.

Although the controllers provide a good w_d values, in w_{sr} and A_x are present consistent oscillations

2.5 Loop shaping controller on Feedforward architecture with a new reference input

In this section there are the results obtained using this last architecture.

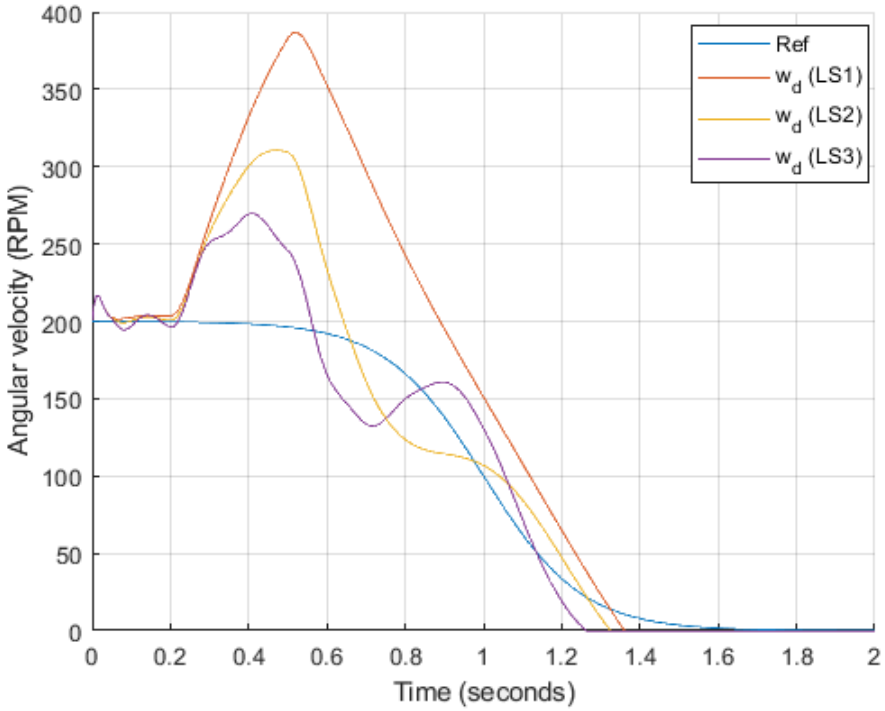

Figure 2.17: w_d responses with new reference (first three controllers).

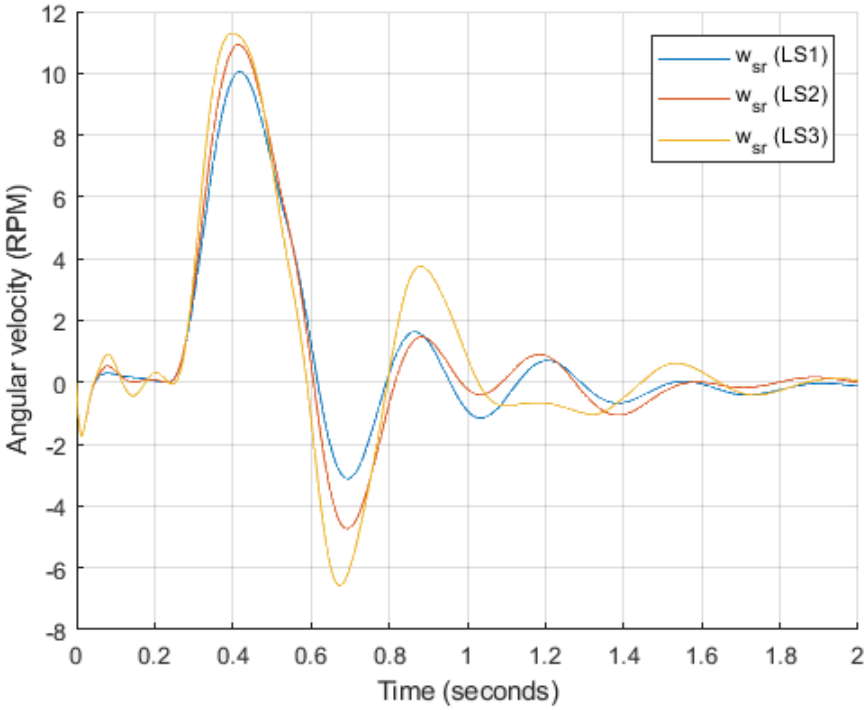

Figure 2.18: w_{sr} outputs with new reference (first three controllers).

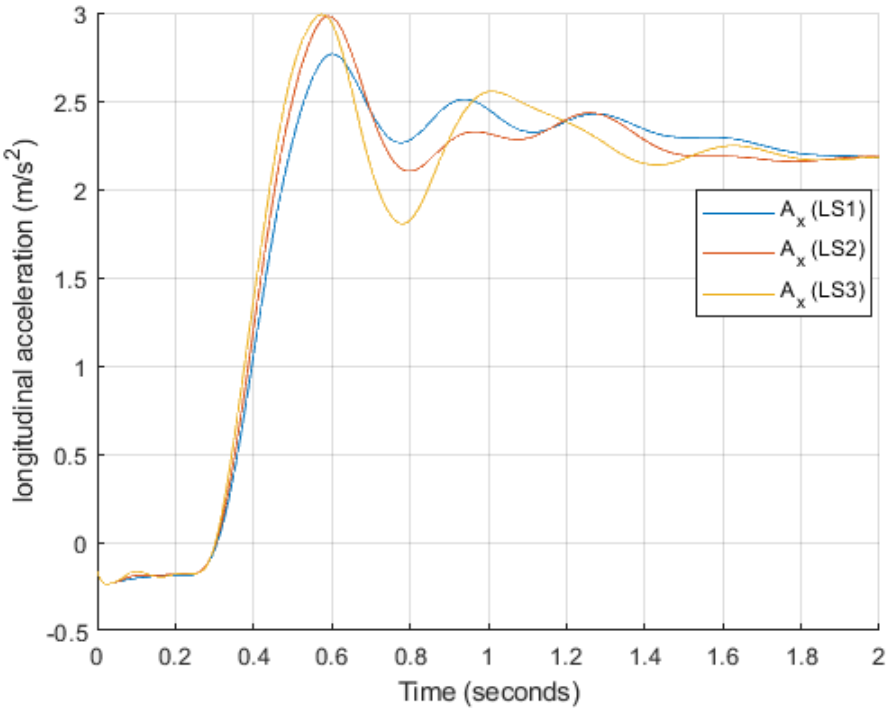

Figure 2.19: A_x values with new reference (first three controllers).

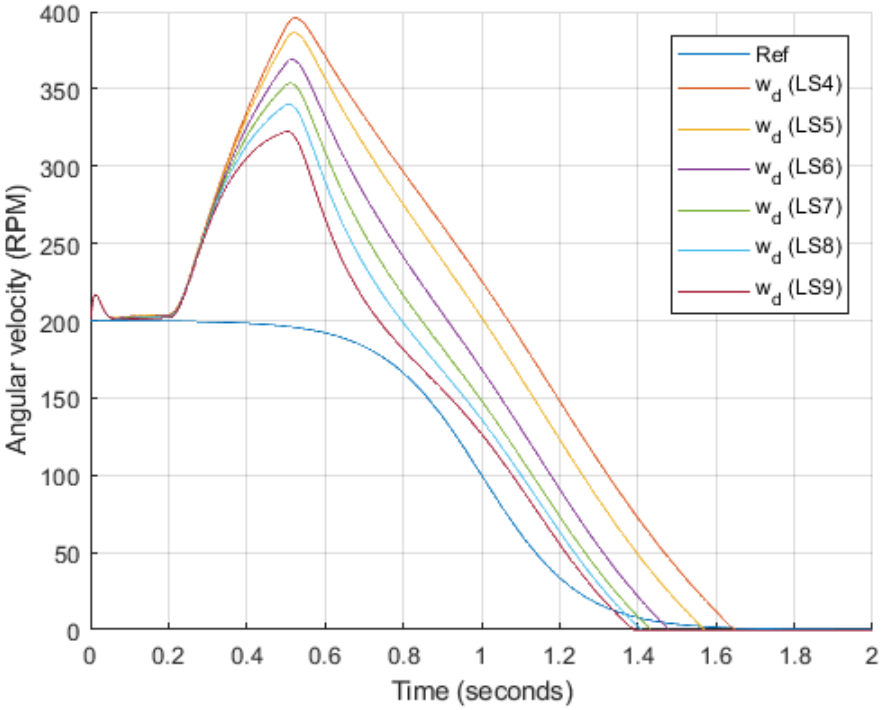

Figure 2.20: w_d responses with new reference (last six controllers).

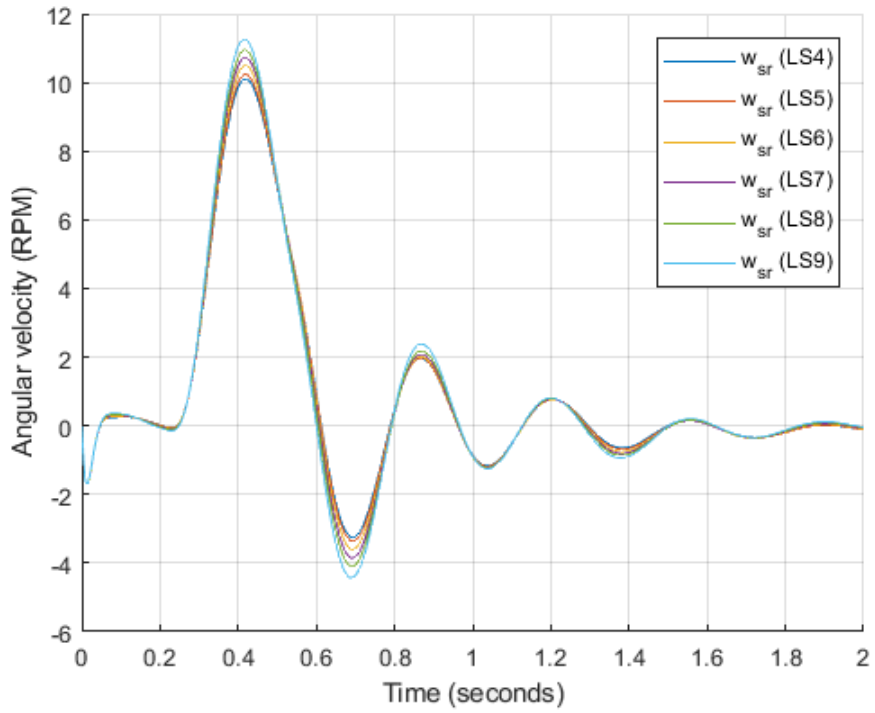

Figure 2.21: w_{sr} outputs with new reference (last six controllers).

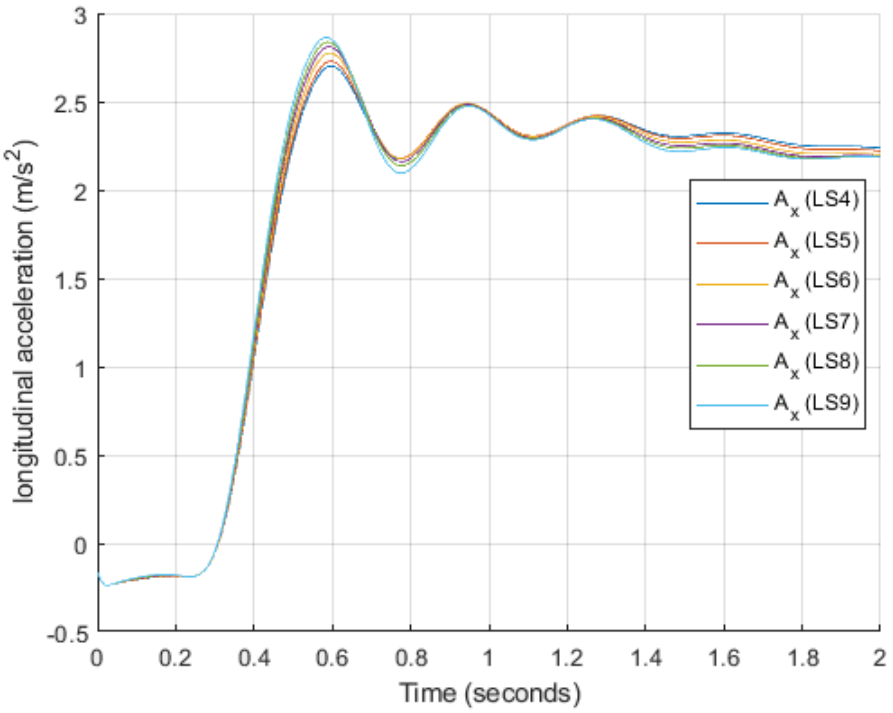

Figure 2.22: A_x values with new reference (last six controllers).

. Even with this architecture the results are not good.

2.6 Loop shaping controller on CRF's detailed model

As a final step we introduced the designed controller in the detailed model. Here is present a preliminary PI, so we show all results obtained in the following pictures:

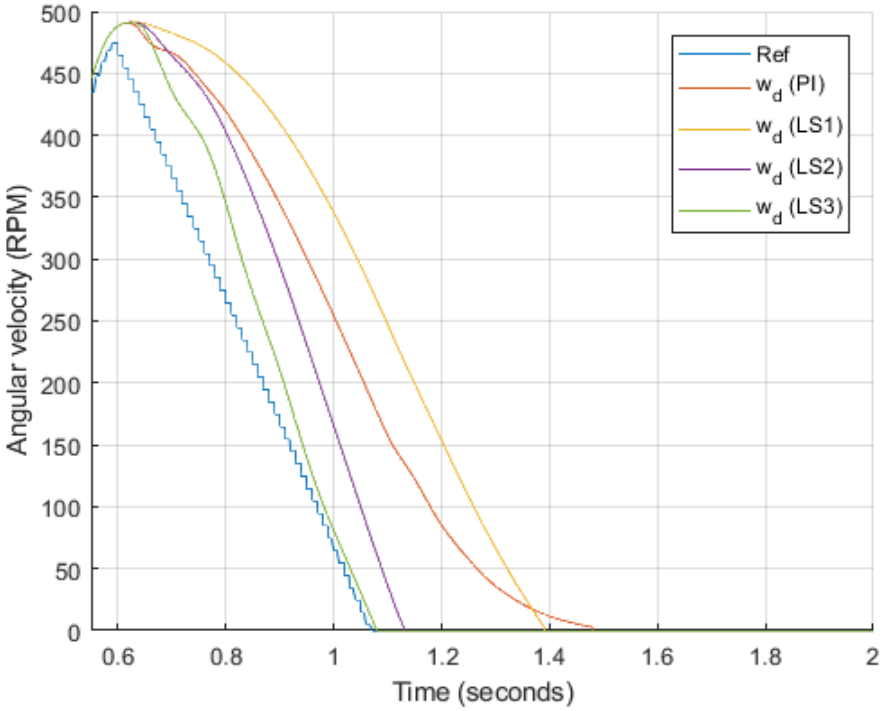

Figure 2.23: w_d responses with CRF's model (first three controllers and PI).

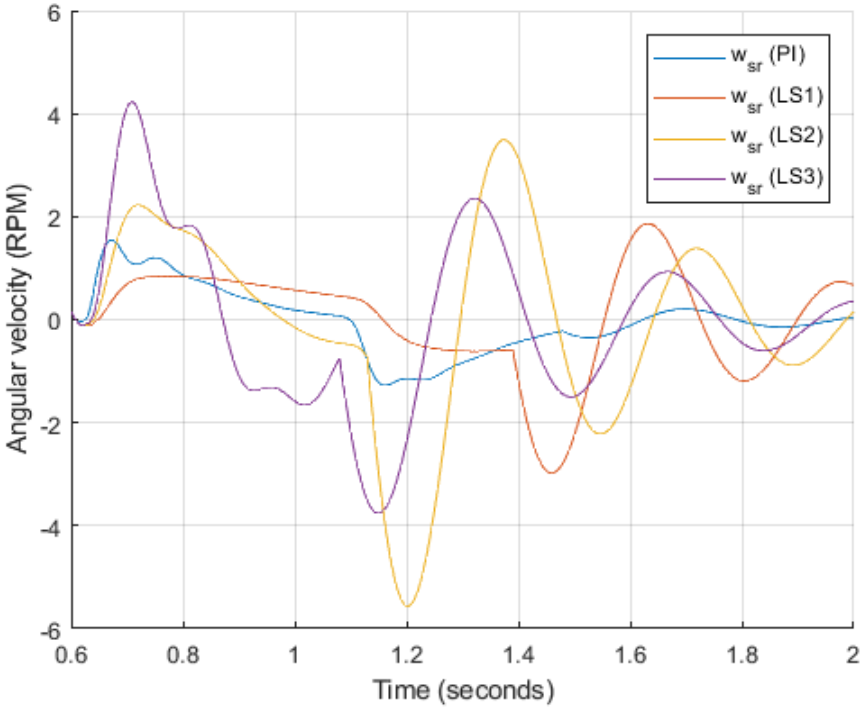

Figure 2.24: w_{sr} outputs with CRF's model (first three controllers and PI).

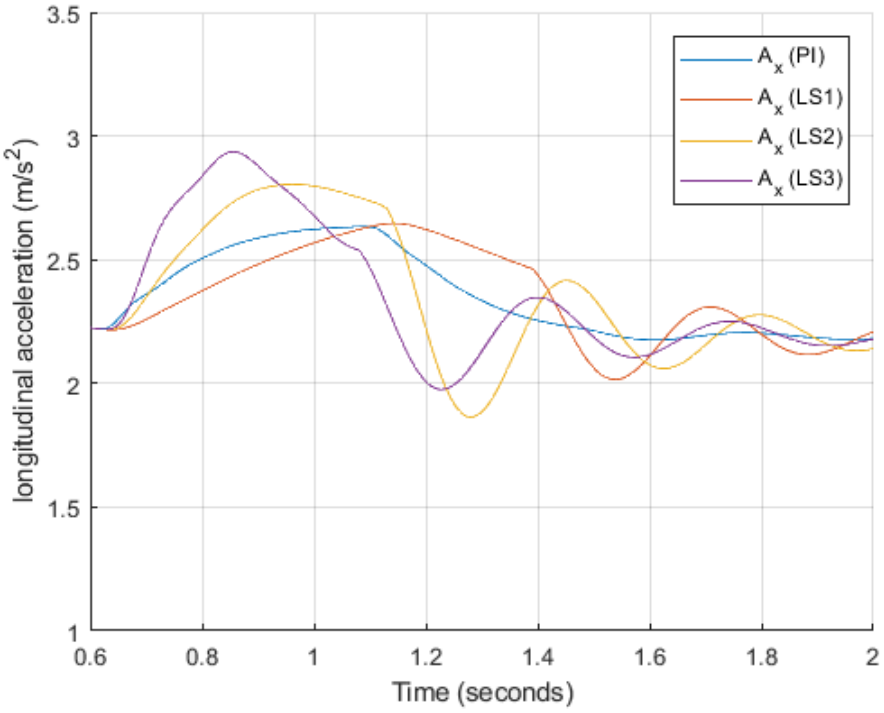

Figure 2.25: A_x values with CRF's model (first three controllers and PI).

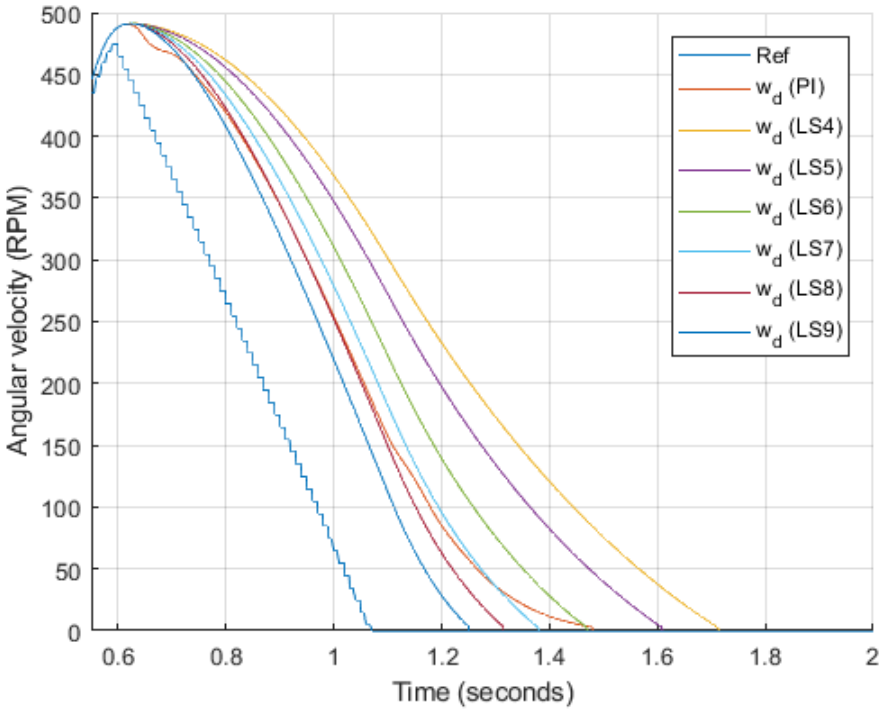

Figure 2.26: w_d responses with CRF's model (last six controllers and PI).

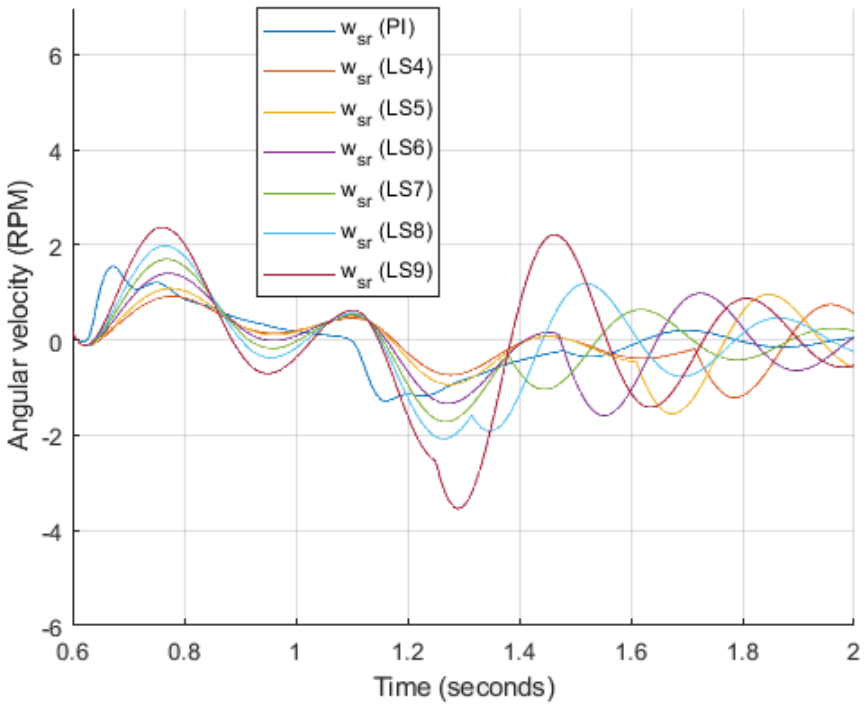

Figure 2.27: w_{sr} outputs with CRF's model (last six controllers and PI).

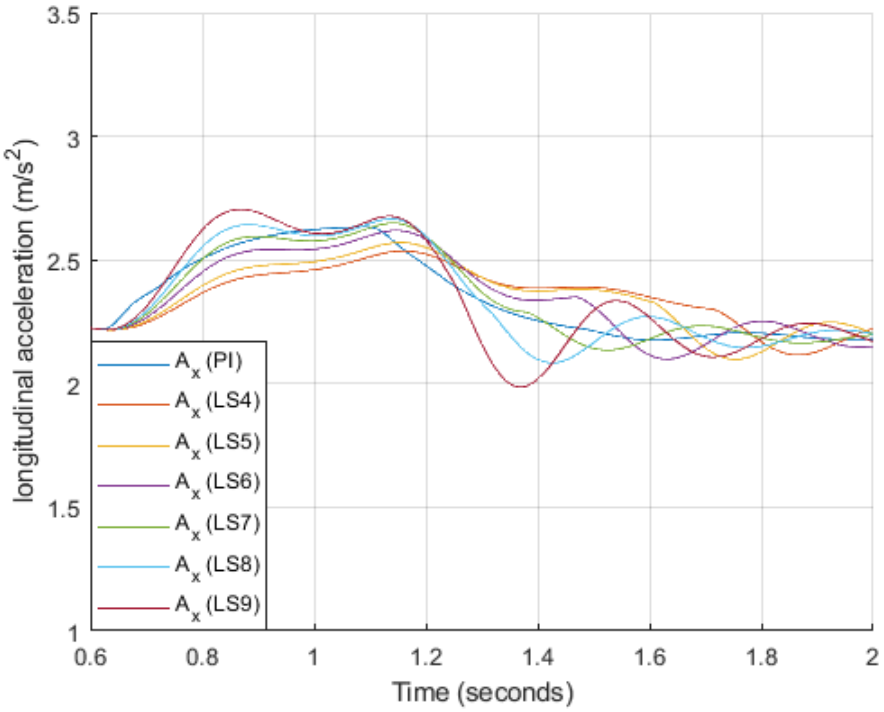

Figure 2.28: A_x values with CRF's model (last six controllers and PI).

Here are present few controllers with good results because as already seen in the first chapter this model uses two controllers contribution.

Chapter 3

H-infinity (H_∞) approach

3.1 Overview

H-infinity is another method to design a controller using frequency-approach. To use this method the control problem have to be expressed as a mathematical optimization problem. H-infinity comes from the name of the mathematical space over which the optimization takes place: Hardy space. The H_∞ norm is the maximum singular value of the function over that space. This can be interpreted as a maximum gain in any direction and at any frequency, for SISO systems, this is effectively the maximum magnitude of the frequency response. By referring to the general system control scheme (Figure 2.1), H-infinity is also used to design robust control when the plant or whatever block in the system is affected by an uncertainty. Hence the H-infinity norm minimization approach, called H_∞ control, refers to a general formulation of the control problem which is based on the following block diagram representation of a general feedback system:

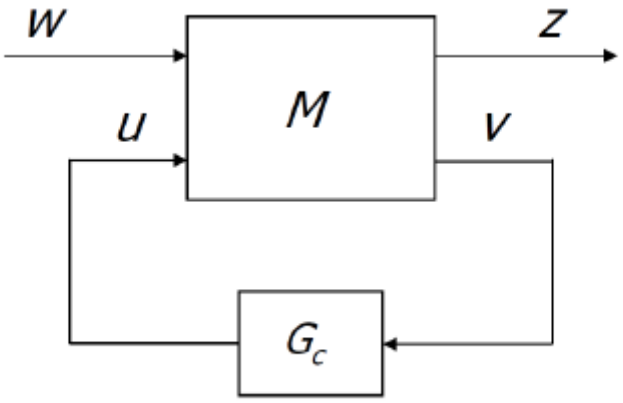

Figure 3.1: General feedback system.

Where M is the generalized plant which has two input, w are the external inputs and u the control inputs. There are also two outputs, z the external outputs and v the controller inputs. G_c is the controller. The external input and output signals of the generalized plant are not necessarily physical variables of the control system, they must be carefully selected in order to take into account the stability/performance requirements of the considered control problem. According to [7] the controller is obtained by solving the following optimization problem

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

where G_c^{stab} is the class of all the controllers which provide internal stability of the nominal feedback control system, and T_{wz} is the closed loop transfer function between the input w and the output z. Hence G_c is designed by minimizing the H_∞ norm of the function T_{wz}. Consider the problem of designing a controller G_c to satisfy the nominal performance conditions:

$$||W_1 S_n||_\infty < 1, ||W_2 T_n||_\infty < 1$$

where W_1 and W_2 are proper weighting functions (we will see how to compute them in the Design section), S_n and T_n are respectively the nominal sensitivity and the nominal inverse sensitivity function already seen in the Loop Shaping chapter. In order to achieve such an objective, the following result on the H_∞ norm of a stack of transfer functions:

$$\left\| \begin{array}{c} H_1 \\ H_2 \\ \dots \\ H_i \\ \dots \\ H_n \end{array} \right\|_\infty < 1 \Rightarrow \|H_i\|_\infty < 1 \ \forall i$$

According to this result, the minimization of the H_∞ norm of n transfer function can be performed by minimizing the H_∞ norm of the stack of such transfer functions, the so called "stacking procedure". For the conservativeness of this procedure we have

$$\|H_i\|_\infty = 1 \ \forall i \Rightarrow \left\| \begin{array}{c} H_1 \\ H_2 \\ \dots \\ H_i \\ \dots \\ H_n \end{array} \right\|_\infty = \sqrt{n}$$

which shows that the H_∞ norm of the stack of transfer functions is in the worst case $\sqrt{n}$ times the value of the H_∞ norm of each single transfer function. Hence, the idea is to design a controller G_c which minimizes both of these two weighted H-infinity norms. If the achieved minimum is less than 1, then the obtained controller satisfies the assigned nominal performance requirements. We have the following situation:

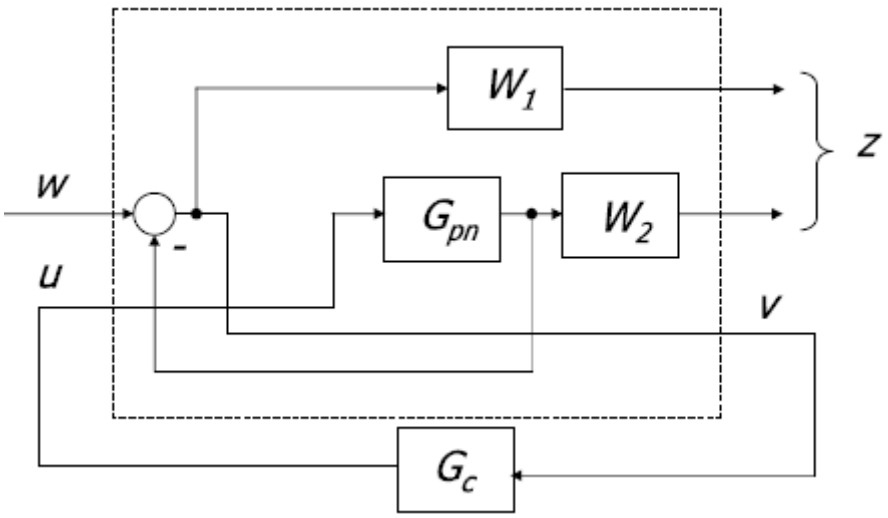

Figure 3.2: General feedback system with Weighting functions.

for simplicity in the above figure the actuator block $G_a = 1$, thus minimizing the H_∞ norm of the function T_{wz} means minimize these quantities: $W_1 S$ and $W_2 T$:

$$T_{wz}(s) = \begin{bmatrix} W_1 S_n \\ W_2 T_n \end{bmatrix} \tag{3.1}$$

3.1.1 Robust control

For what concern the robust control ([7]), we have to take into account that the plant is affected by uncertainty. The basic approach to take uncertainty into account is to describe the plant under investigation as a member of a set system, also called model set. There are several model sets such as additive, multiplicative, inverse additive and inverse multiplicative. In this thesis we will refer to the multiplicative model set. So the plant can be defined as follows:

$$G_p(s) = G_{pn}(s)[1 + W_u(s)\Delta(s)], ||\Delta(s)||_\infty \leq 1$$

where $W_u(s)$ is a weighting function which accounts for the size of the uncertainty and $\Delta(s)$ can be any possible transfer function whose H-infinity norm is less than 1. If the designed controller fulfills the following relation:

$$||W_u T_n||_\infty < 1$$

it ensures robust stability.

3.2 Design

In the Loop Shaping chapter, we have shown that performance objectives of a feedback control system are usually specified in terms of requirements on the loop function, in order to carry out the controller design through a loop-shaping approach. Here, we shall see how performance specifications can be specified in terms of requirements on the sensitivity function and complementary sensitivity function. The system must be modeled as a linear time-invariant (LTI) before proceeding with the design [8]. Let's assume to have the following requirements to achieve:

- Rise time: $t_r < 0.3\ s$
- Overshoot: $\hat{s} < 8\%$

- Steady-state output error in presence of C_m and C_{load}: $|e_{d_p}^{\infty}| = 0$

Translation of the specifications:

- $\hat{s} < 8\% \rightarrow \zeta \geq \frac{|\ln(\hat{s})|}{\sqrt{\pi^2+\ln^2(\hat{s})}} \rightarrow \zeta = 0.63$
- S_p : Maximum sensitivity $\rightarrow \underset{\omega\in[0,\infty]}{max}|S(j\omega)| \leq \frac{2\zeta\sqrt{2+4\zeta^2+2\sqrt{1+8\zeta^2}}}{\sqrt{1+8\zeta^2}+4\zeta^2-1} = 1.33$
- T_p : Maximum complementary sensitivity $\rightarrow \underset{\omega\in[0,\infty]}{max}|T(j\omega)| \leq \frac{1}{2\zeta\sqrt{1-\zeta^2}} = 1.02$
- $t_r < 0.3 \rightarrow \omega_n \geq \frac{(\pi-\arccos(\zeta))}{t_r\sqrt{1-\zeta^2}} = 9.6\ rad/s$ where ω_n is the natural frequency of 2^{nd} order prototype system
- $|e_{d_p}^{\infty}| = 0 \rightarrow$Applying the final-value theorem, just considering C_m as disturbance:

$$|e_{d_p}^{\infty}| = \lim_{t\to\infty}|e^{dp}(t)| = \lim_{s\to 0}s|e^{dp}(s)| = \lim_{s\to 0}s|y^{dp}(s)| = \lim_{s\to 0}s|S(s)d_p(s)| =$$

$$\lim_{s\to 0}\ s^{\mu+p+1}|P_1(s)S(s)^*\tfrac{D_{p0}}{s^{h_p+1}}| = 0$$

if $\mu \geq 1$ there is no constraint on $|S(0)^*|$, so we choose $\mu = 1$ and $|S(0)^*| = 1$. Where $S(s) = s^{\mu+p}S^*(s)$. Note that in this case the final-value theorem shows the sensitivity function to have the constraint on $|S(0)|$ instead on K_c like in the Loop Shaping chapter. The reason comes from the relation between frequency response of function $L(s)$ and $S(s)$, because the frequency value where $|S(j\omega)|$ crosses the 0 dB axis is a lower bound of the crossover frequency ω_c.

3.2.1 Weighting functions computation

Basically these functions can be computed through rational functions of the variable s, in order to approximate the frequency domain constraints on $S(s)$ and $T(s)$. Another possibility is the use of Butterworth polynomials either as denominator or numerator of the approximating rational function to effectively retain constraints on different frequency ranged.

3.2.1.1 $W_s(s)$ function

The W_s function have to satisfy the above derived constraints. We decided to use a second order function with this structure:

$$W_s^{-1}(s) = \frac{Ks^{\mu+p}}{1+1.414\frac{s}{w_1}+(\frac{s}{w_1})^2}$$

where the denominator is a Butterworth polynomial of second order and K is a constant value to tune. We want the following frequency characteristics:

- Low frequency: $\lim_{s\to 0} \frac{1}{s^{\mu+p}} W_s^{-1}(s) = |S(0)^*| = K$, plotting both $S_n(s)$ and W_s^{-1} we note that are not similar, hence we choose $|S(0)^*| = 0.15$.
- High frequency: $\lim_{s\to\infty} W_s^{-1}(s) = S_p$

From these relations we can compute the pole as $w_1 = \sqrt{\frac{S_p}{K}} = 2.98$

3.2.1.2 $W_t(s)$ function

The W_t function implies constraint on the complementary sensitivity function $T_n(s)$. Even in this case we choose a second order function:

$$W_t^{-1}(s) = \frac{T_p}{1+1.414\frac{s}{w_2}+(\frac{s}{w_2})^2}$$

The main characteristic of W_t is that the $\lim_{s\to 0} W_t^{-1}(s) = T_p$. In theory w_2 has to be chosen in order to increase the bandwidth remaining under a M_T constraint. This constraint derives from a possible source of error on the sensor. In this study we suppose to have a sensor error equal to 0. Hence for what concern the w_2 pole is just discretion of the designer who can increase or decrease the bandwidth of the final controller without constraints. So we put $w_2 = 28$.

3.2.1.3 $W_u(s)$ function

As already said in the 3.1.1 subsection, $W_u(s)$ accounts for the size of the uncertainty. The uncertainty under investigation is the K_{usura} value. This is a constant value inside the actuator block which takes into account the clutch wear. The range of values where this gain can change is:

$$0.7 \leq K_{usura} \leq 1.3 \qquad (3.2)$$

In this case we compute W_u through a graphical way. First we plot all possible errors computed as follows: $|\frac{G_p(j\omega)-G_{pn}(j\omega)}{G_{pn}(j\omega)}|$, then by the use of *ginput* command in Matlab we compute $W_u(s)$ in order to satisfy the relation:

$$\left|\frac{G_p(j\omega)-G_{pn}(j\omega)}{G_{pn}(j\omega)}\right| \leq |W_u(j\omega)| \qquad (3.3)$$

We have that:

$$W_u(s) = \frac{0.2987s+3913}{s+13030}$$

In the below figure we can see the error values when $K_{usura} = 0.7$ and $K_{usura} = 1.3$, the errors produced with the intermediate K_{usura}'s values are smaller so for simplicity are not shown.

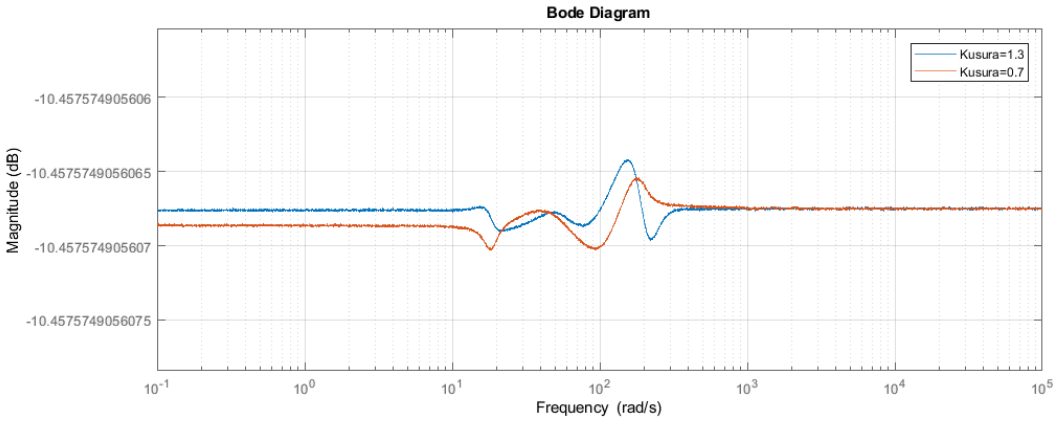

Figure 3.3: Plot of the relevant errors.

Note that G_{pn} is the nominal plant with $K_{usura} = 1$. In this design, the plant is composed by the product between the actuator transfer function $G_a(s)$ and the plant $G_p(s)$. Before continuing design we have to choose the weighting functions, in order to minimize the quantity (3.1). From [8] we have:

- $W_1(s) = W_s(s)$;
- $|W_2(j\omega)| = max(|W_t(j\omega)|, |W_u(j\omega)|) \Rightarrow W_2(s) = W_t(s)$.

3.2.2 LMI optimization approach

We already said that the controller is designed by solving the optimization problem (3.1). The literature proposes a large number of approaches which solve such an optimization problem. We exploited the one based on the solution of a suitable constrained optimization problem, where the constraints are in the form of linear matrix inequalities (LMI). As mentioned in [8], the LMI approach is based on a state-space description of the generalized plant M

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

where x_M is the state of the generalized plant given by the union of the state variables of the nominal model G_{pn} and those of the weighting function W_1 and W_2. The eigenvalues of matrix A_l are the union of the poles of the transfer function G_{pn}, W_1 and W_2. The LMI optimization problem can be solved under the assumptions that the matrix triplet (A_l, B_2, C_2) is stabilizable (i.e. if all unstable modes are controllable), detectable (i.e. if all unstable modes are observable) and $D_{22} = 0$. So we have a significant result: the generalized plant M can be internally stabilized by an LTI controller G_c if and only if W_1 and W_2 are stable transfer functions. This is a problem because in anyway the performance requirements lead to an unstable weighting function W_1 due to the presence of one or more poles at $s = 0$. Assuming that W_1 has $\mu+p$ poles at $s = 0$, we replace W_1 in the generalized plant with a new weighting function W_1^* obtained as follows:

$$W_1^*(s) = W_1 \frac{s^{\mu+p}}{(s+\varepsilon)^{\mu+p}}$$

where ε is a low frequency pole. The controller that we obtain with this new weighting function will have at most $\mu+p$ poles at $s = -\varepsilon$ which must be replaced with a pole at $s = 0$.

In our case W_1 has a pole at $s = 0$, moreover W_2 cannot be used in Simulink because is a non proper function so we have to modify these functions as follows:

- $W_1^*(s) = W1\frac{s^2}{(s+\varepsilon)^2}, \varepsilon = 0.01\omega_c$
- $W_2^* = \frac{1}{T_p}$

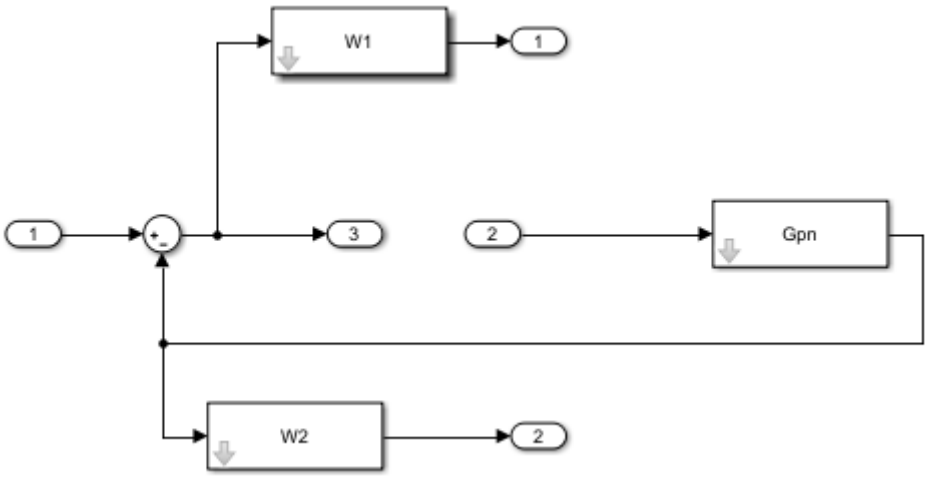

Figure 3.4: *'generalized_plant'* Simulink file.

In the follows we are going to present the Matlab code used to design the controller:

[Am,Bm,Cm,Dm]=linmod('generalized_plant')

Using *linmod* function in Matlab we get a state space description of the generalized plant M ('*generalized_plant*' file) (Figure 3.1);

M=ltisys(Am,Bm,Cm,Dm)

ltisys command computes the generalized model M in a specific form needed by the LMI control toolbox to solve the optimization problem;

M=sderiv(M,2,[1/wt 1])

M=sderiv(M,2,[1/wt 1])

sderiv command is used in order to avoid problem with Simulink as said before, we also indicate to Matlab where to add the zeros, i.e. channel 2 (Figure 3.4);

[gopt,Gc]=hinflmi(M,[1 1],0,0.01,[0 0 0])

hinflmi gives the best H-infinity performance and the controller G_c in a specific compact form;

[Ac,Bc,Cc,Dc]=ltiss(Gcmod);

The matrices of the state-space realization of G_c are obtained using *ltiss* command.

Gc=ss(Ac,Bc,Cc,Dc);

Finally to have the controller as transfer function the *ss* command is used:

$$G_c(s) = \frac{-434.85(s+0.7567)(s+0.009368)(s+0.004946)(s^2+82.23s+2563)(s^2+121.4s+7696)(s^2+134.1s+4.27e04)}{(s+1.418e05)(s+138)(s+89.52)(s+0.005905)(s^2+0.01896s+9.053e-05)(s^2+5.344s+339.4)(s^2+146.2s+2.914e04)}$$

This controller has two poles at very low frequency paired with two zeros around the same frequency, due to the use of W_1^*. We know that the controller must have a pole at $s = 0$ ($\mu = 1$), thus we have to delete one zero-pole pair and replace the remaining with a pole at $s = 0$, hence to obtain the final form we use to "clean" it, taking into account that to delete:

- the high frequency pole we have to use the *dc gain* form: $(1 + \frac{s}{\alpha})$ where α is the pole/zero to cancel;
- the low frequency pole we have to use the *zpk* mode: $(s+\alpha)$ with α pole/zero to cancel.

The final controller is:

$$G_c(s) = \frac{-0.012266(s+0.7567)(s+0.009368)(s^2+82.23s+2563)(s^2+121.4s+7696)(s^2+134.1s+4.27e04)}{s(s+138)(s+89.52)(s+0.005905)(s^2+5.344s+339.4)(s^2+146.2s+2.914e04)}$$

Looking at the norms we have:

$$||W_s S_n|| = 0.97 < 1,\ ||W_t T_n|| = 0.997 < 1, ||W_u T|| = 0.32 < 1$$

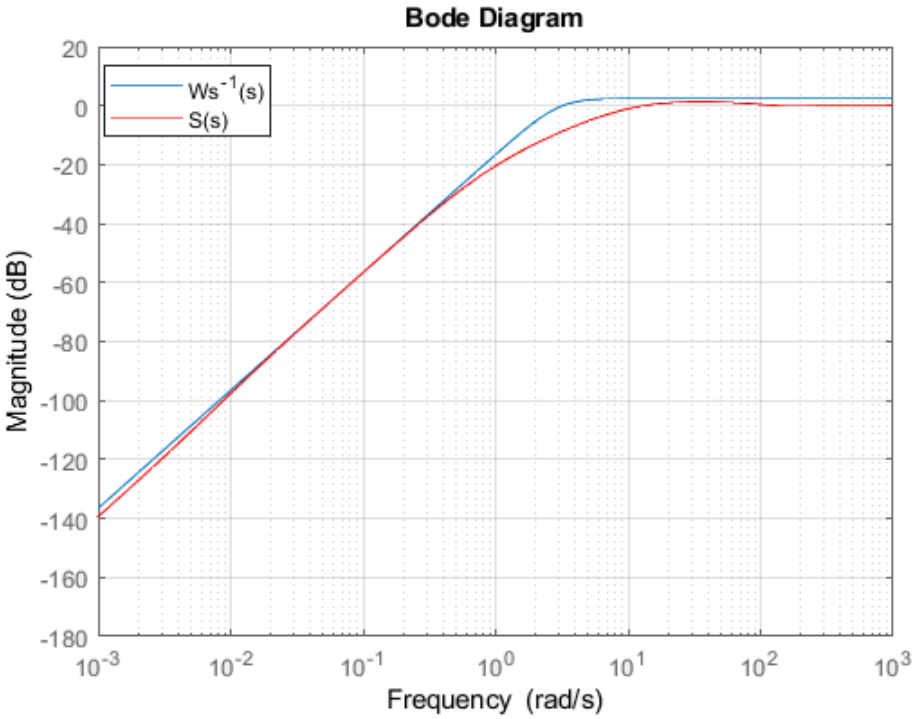

Figure 3.5: Sensitivity function $S_n(s)$ and weighting function $W_s^{-1}(s)$.

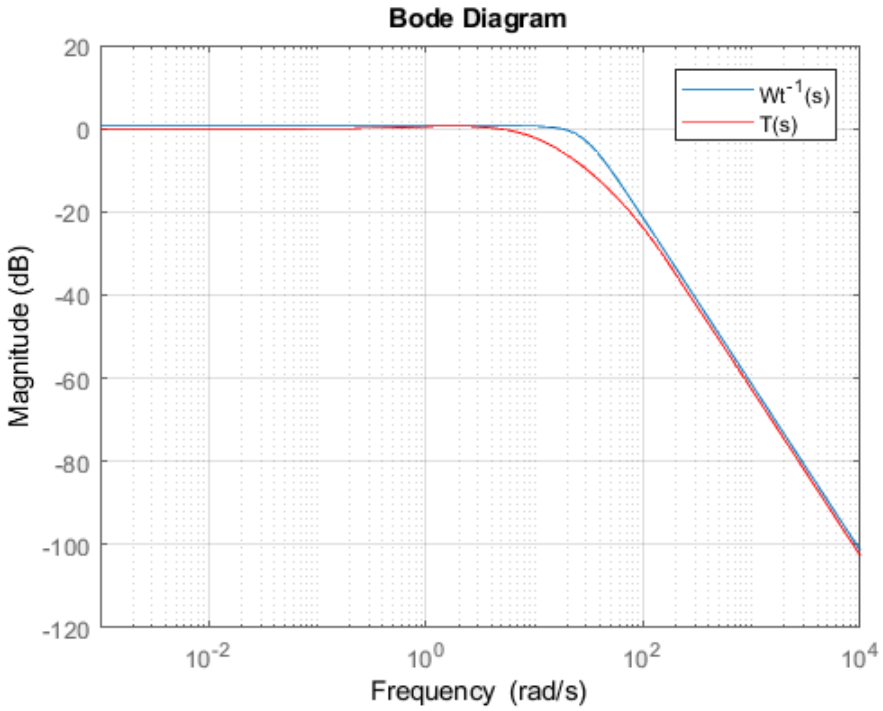

Figure 3.6: Complementary Sensitivity function $T_n(s)$ and weighting function $W_t^{-1}(s)$.

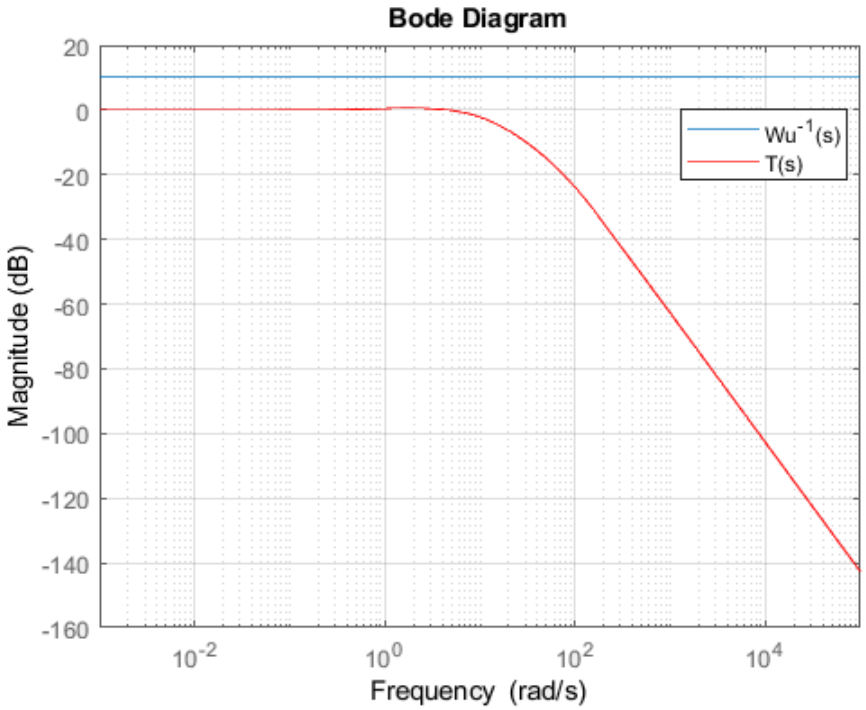

Figure 3.7: Complementary Sensitivity function $T_n(s)$ and weighting function $W_u^{-1}(s)$.

Nominal performance and robust stability are ensured. The step response and Nichols chart are plotted below:

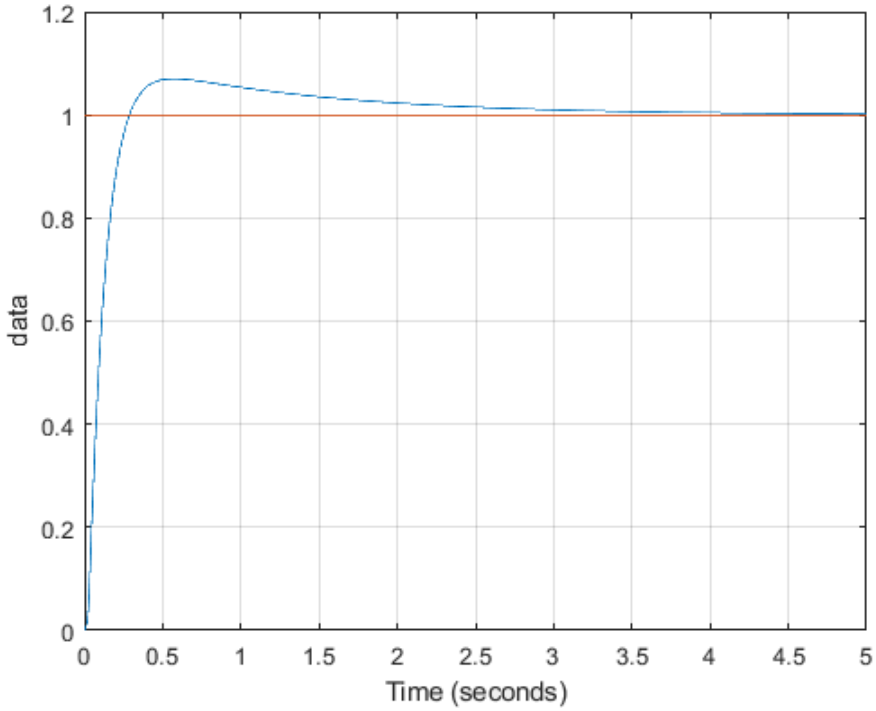

Figure 3.8: Step response of the system with $G_c(s)$.

The time step requirements fixed previously are fulfilled.

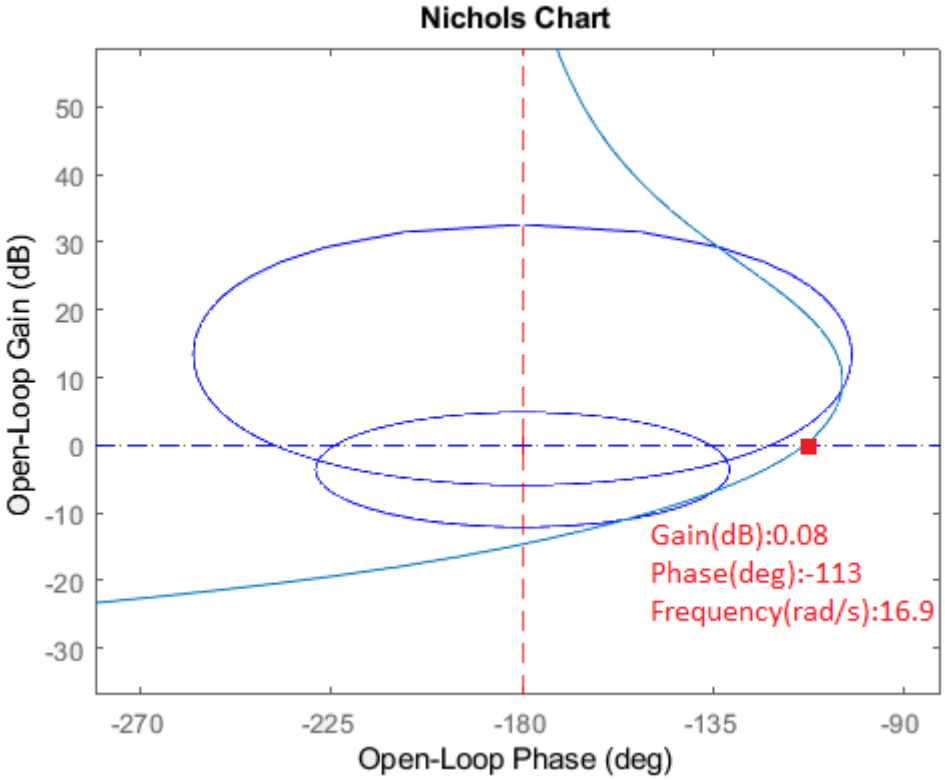

Figure 3.9: Nichols plot with $G_c(s)$.

Even in this case we have to discretize the controller @$T_s = 10\ ms$

$$G_c(z) = \frac{-0.012266(z-0.9925)(z-1)(z^2-1.208z+0.3832)(z^2-1.056z+0.5025)(z^2+0.1153z+0.4417)}{(z-0.4085)(z-0.2516)(z-1)(z-1)(z^2-1.915z+0.948)(z^2-0.02714z+0.2318)}$$

This is the workflow to obtain a controller using H_∞ approach, we made several attempts getting different controllers and results. The most useful controllers will be show in the following.

3.3 H_∞controller on Base architecture

Now we want to test the designed controllers through the Base architecture:

- $H_{\infty 1} = \frac{-0.012266(s+0.7567)(s+0.009368)(s^2+82.23s+2563)(s^2+121.4s+7696)(s^2+134.1s+4.27e04)}{s(s+138)(s+89.52)(s+0.005905)(s^2+5.344s+339.4)(s^2+146.2s+2.914e04)}$;
- $H_{\infty 2} = \frac{-0.48353(s+206.6)(s+7.48)(s+0.01008)(s^2+82.23s+2563)(s^2+121.4s+7696)}{s(s+303.1)(s+202)(s+138)(s+0.01729)(s^2+5.344s+339.4)}$;

- $H_{\infty 3} = \frac{-0.0050214(s+0.4894)(s+0.006739)(s+0.0002769)(s^2+82.23s+2563)(s^2+121.4s+7697)(s^2+134.1s+4.27e04)}{(s+138)(s+24.42)(s+0.01288)(s+0.01239)(s+0.005905)(s^2+5.344s+339.4)(s^2+167.4s+2.986e04)}$.

Plot name	$t_r[s]$	$\hat{s}$
$H_{\infty 1}$	0.3	8%
$H_{\infty 2}$	0.05	19%
$H_{\infty 3}$	0.4	7%

Table 3.1: H_∞ controllers overview.

As seen in the previous chapter this architecture does not provide good performance because of C_m, only the controller $H_{\infty 2}$ is able to track the reference but at the cost of huge w_{sr} oscillations.

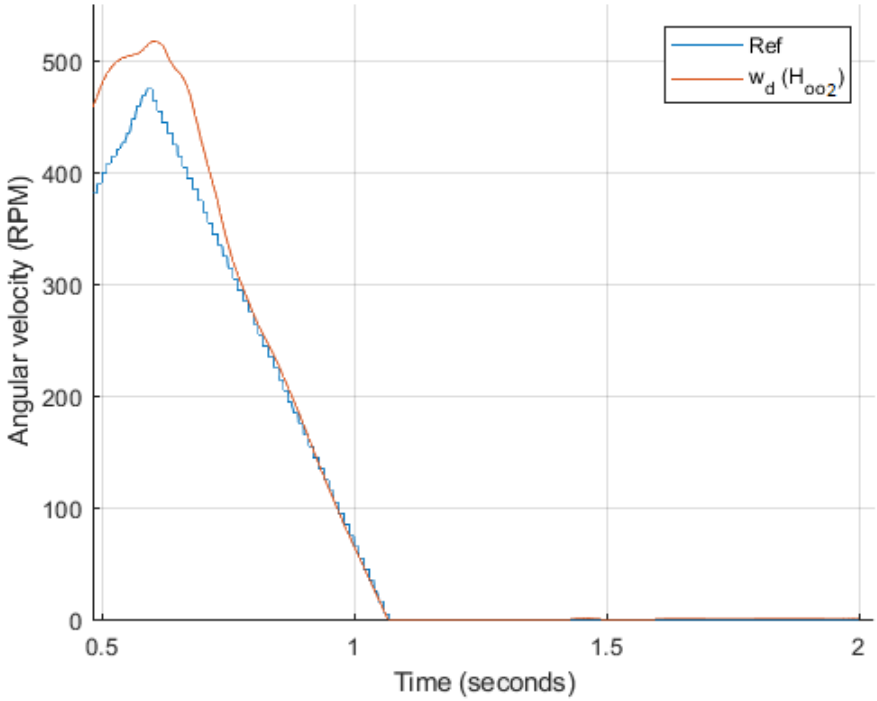

Figure 3.10: w_d response with $H_{\infty 2}$ in Base architecture.

3.4 H_∞controller on Feedforward architecture

As already done in the previous section we show the results obtained using the Feedforward architecture.

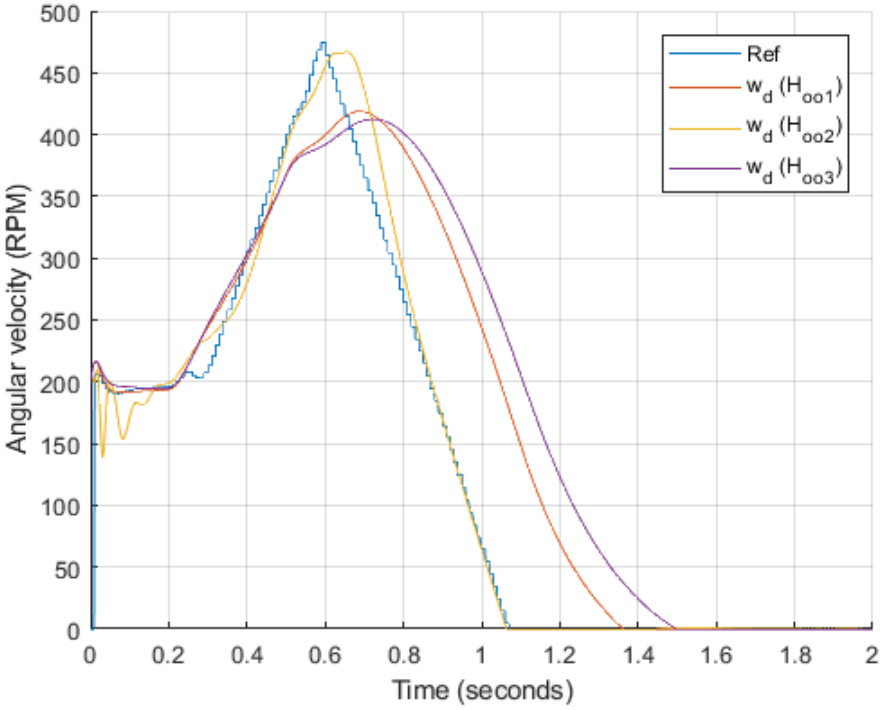

Figure 3.11: w_d responses using Feedforward architecture.

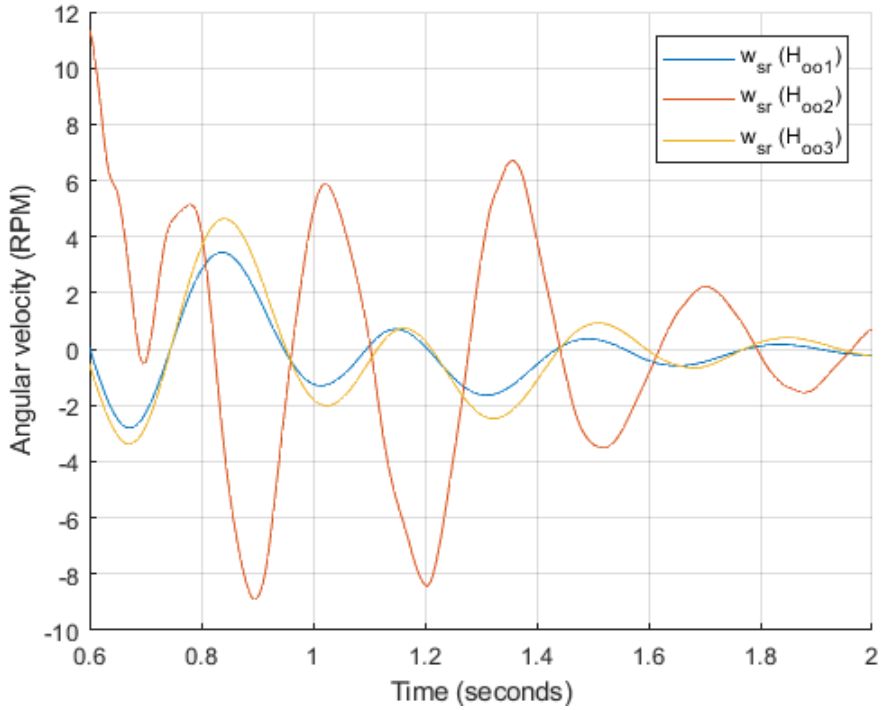

Figure 3.12: w_{sr} outputs using Feedforward architecture.

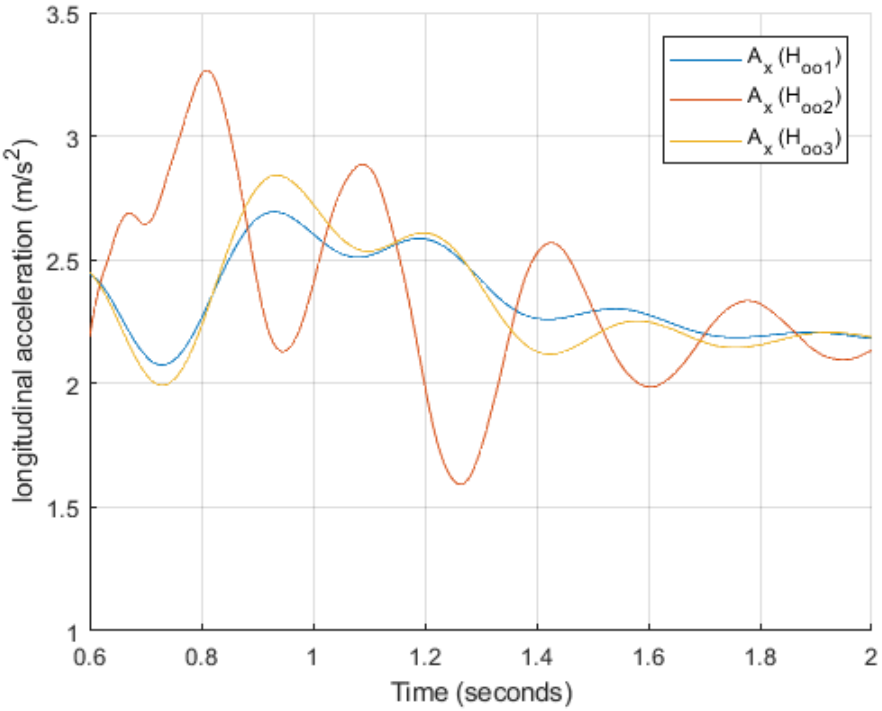

Figure 3.13: A_x values using Feedforward architecture.

We notice that faster convergence to zero leads to have huge oscillations on the driveline.

3.5 H_∞controller on Feedforward architecture with a new reference input

In this section there are the results obtained using this last architecture.

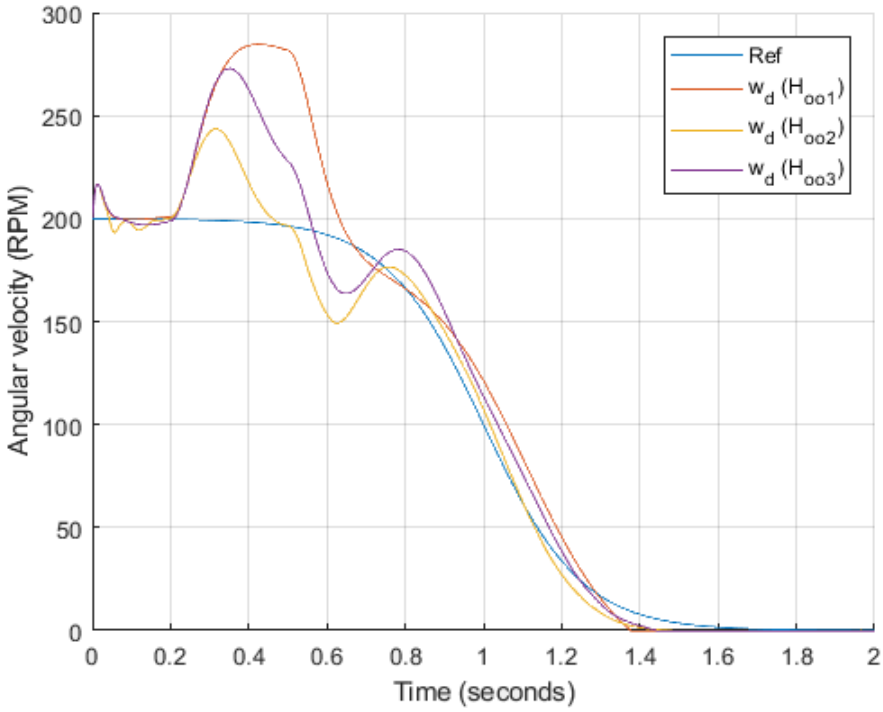

Figure 3.14: w_d responses using new reference architecture.

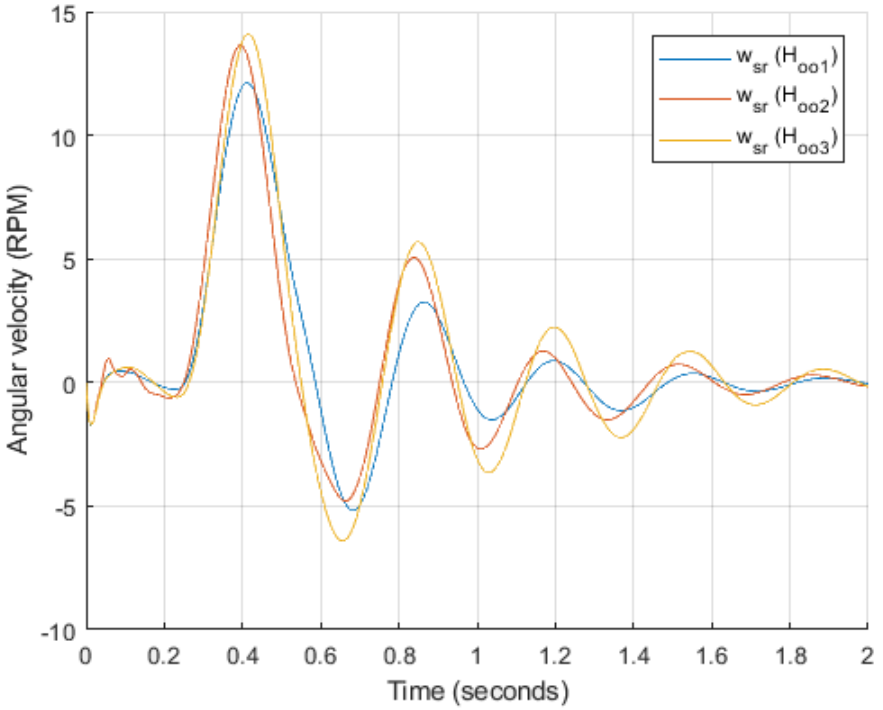

Figure 3.15: w_{sr} outputs using new reference architecture.

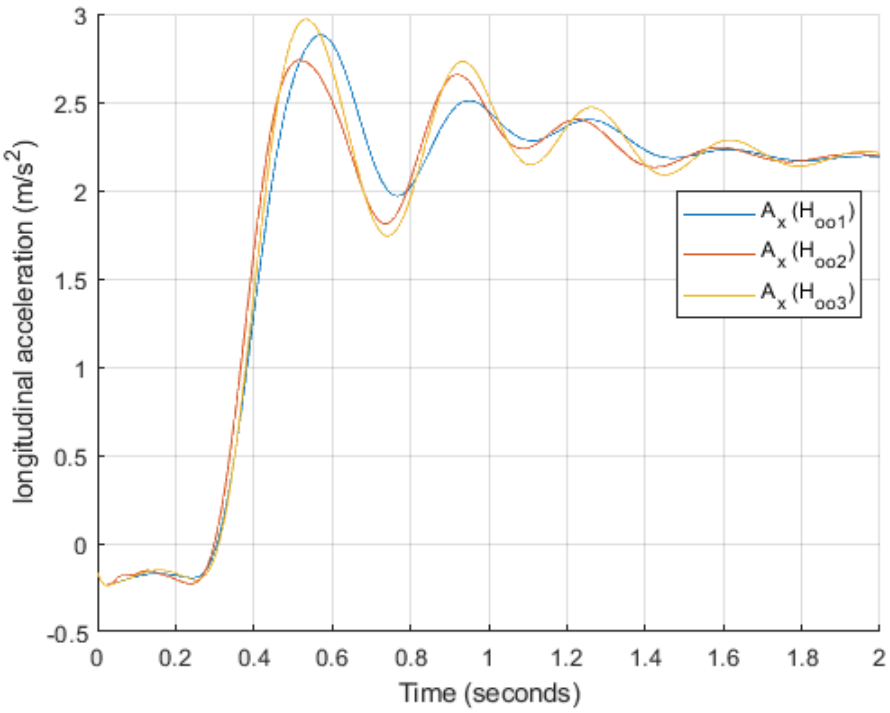

Figure 3.16: A_x values using new reference architecture.

Change the reference input does not cause better results.

3.6 H_∞controller on CRF's detailed model

As a final step we introduced the designed controllers in the detailed model taking also into account PI's values:

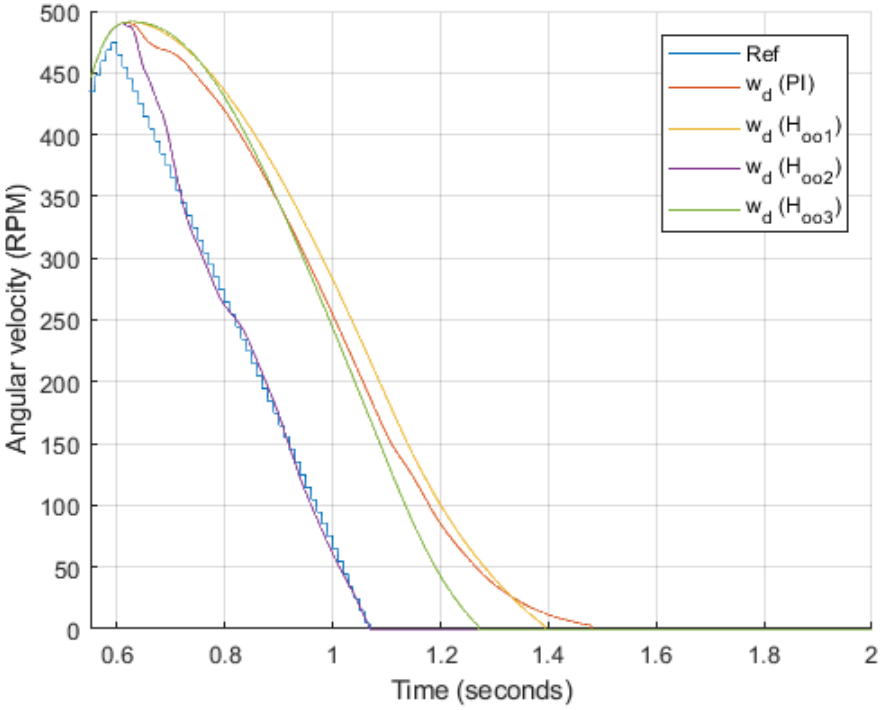

Figure 3.17: w_d responses with CRF's detailed model.

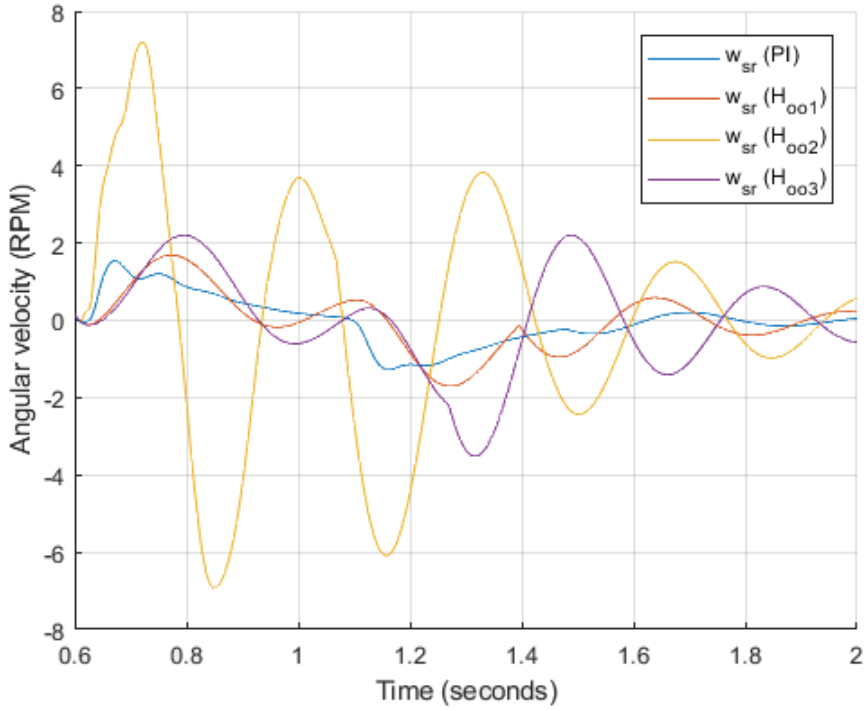

Figure 3.18: w_{sr} outputs with CRF's detailed model.

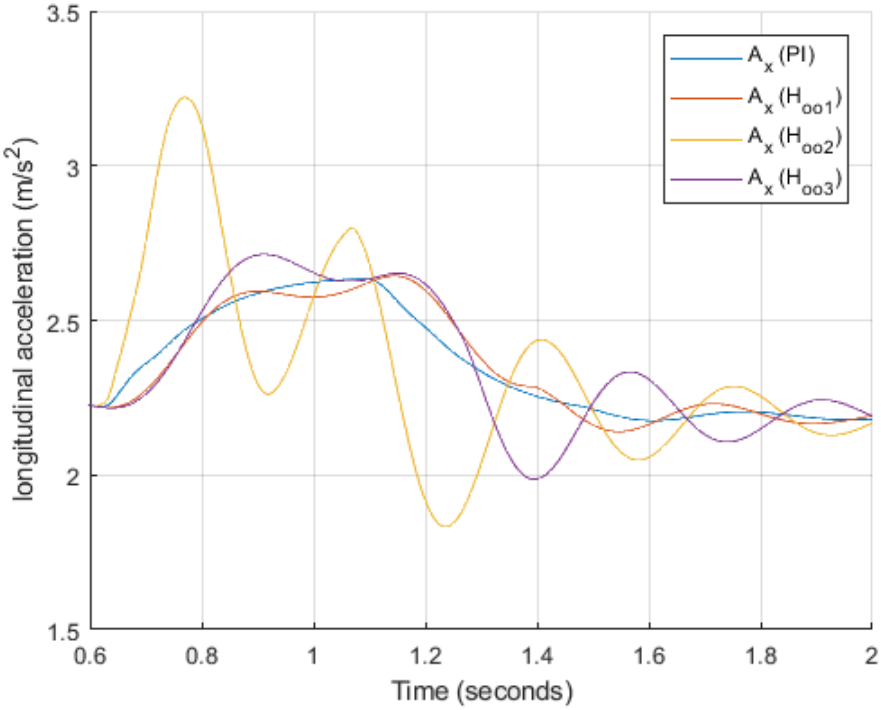

Figure 3.19: A_x values with CRF's detailed model.

Using the detailed model the oscillations are reduced, only $H_{\infty 2}$ has a very bad behavior both in w_{sr} and A_x.

Chapter 4

Model Predictive Control approach

4.1 Overview

Model Predictive Control (MPC) is an advanced and flexible control method. It is used to manage and control composite system, as matter of fact the complexity of the MPC control algorithm is not generally needed to provide adequate control of simple systems. It allows to deal with input/state/output constraints and to manage systematically the trade-off performance/command activity. The main advantage of MPC is the fact that it allows the current timeslot to be optimized, while keeping future timeslots in account. In particular, at each time step a prediction over a chosen horizon is performed, using a model of the plant, while the command input is chosen as the one yielding the best prediction (i.e. the prediction closest to the desired behavior) by means of some on-line optimization algorithm. MPC is used in several applications such as automotive system, aerospace systems, robotics, biomedical devices and chemical processes. As mentioned in [9], consider the generic linear system

$$\dot{x}(t) = Ax(t) + Bu(t)$$
$$y(t) = Cx(t)$$

where $x \in \mathbb{R}^n$ is the state, $u \in \mathbb{R}^m$ is the command input and the output is $y = x, y \in \mathbb{R}^p$. It is assumed that the state is measured, otherwise an observer has to be employed. Suppose that the state is measured in real-time, with a sampling time T_s so that the measurements are $x(t_k)$, with $t_k = T_s k$, $k = 0, 1,...,n$. To understand the MPC concept we can see the following picture.

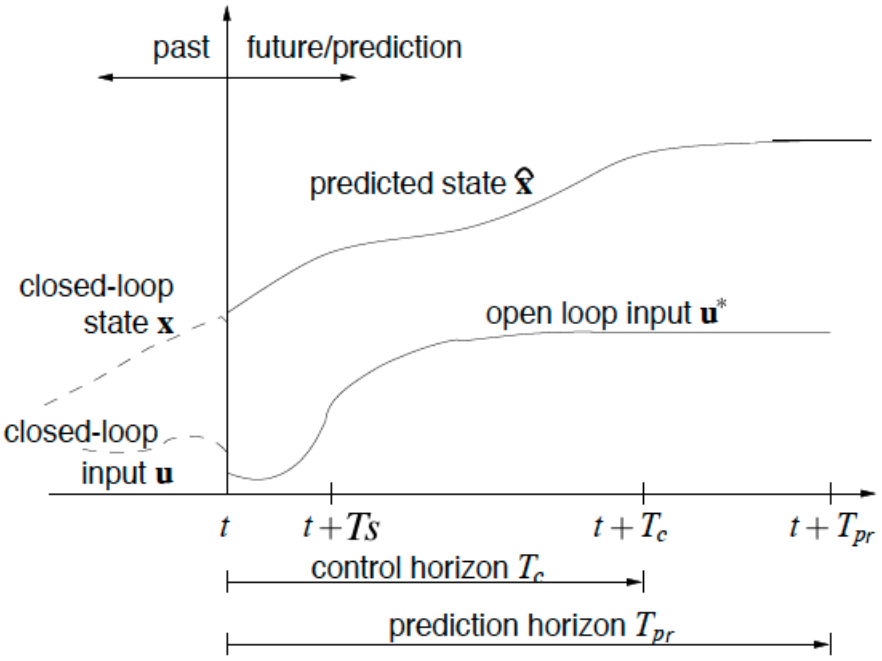

Figure 4.1: Control horizon T_c and prediction horizon T_{pr}.

A prediction of the state over an interval $[t, t + T_{pr}]$ is obtained by integration of the initial system. T_{pr} is called the *prediction horizon* ($T_{pr} \geq T_s$). At a time $\tau \in [t, t+T_{pr}]$ the predicted state is a function of the initial state $x(t)$ and the input signal $\hat{x}(\tau) \equiv \hat{x}(\tau, x(t), u(t : \tau))$, note that here the input signal u is in the interval $[t,\tau]$. Usually the input signal is assumed constant after a certain time T_c called the *control horizon*, $u(\tau) = u(t + T_c)$, where $0 \leq T_s \leq T_c \leq T_{pr}$. In this prediction u is a generic input which does not depend on x. At each time $t = t_k$, we look for an input signal $u^*(t : \tau)$ such that the predicted state $\hat{x}(\tau, x(t), u^*(t : \tau))$ has the desired behavior for $\tau \in [t, t + T_{pr}]$. The concept of desired behavior is formalized by defining the objective function:

$$J(u(t : t + T_{pr})) \doteq \int_t^{t+T_{pr}} [||\widetilde{x}_p(\tau)||_Q^2 + ||u(\tau)||_R^2] d\tau + ||\tilde{x}_p(t + T_{pr})||_P^2$$

where $\widetilde{x}_p(\tau) \doteq r(\tau) - \hat{x}(\tau)$ is the predicted tracking error, with $\hat{x}(\tau)$ obtained by integration of the initial system, $r(\tau)$ is a reference to track, and $\|.\|_X$ are weighted vector norms and their integrals are square signal norms. The input signal $u^*(t : t + T_{pr})$ is chosen as one minimizing the objective function above. Hence, the main goal of MPC is to minimize, at each time t_k, the tracking error over a finite time interval. It is important to highlight that the term $||u(\tau)||_R^2$ allows us to menage the trade-off between performance and activity, while the term

$||\tilde{x}_p(t+T_p)||_P^2$ gives further importance to the final tracking error. The weighted norm of a vector $x \in \mathbb{R}^n$ is defined:

$$||x||_Q^2 \doteq x^T Q x = \sum_{i=1}^{n} q_i x_i^2$$

where Q =diag$(q_{1,\dots,}q_n) \in \mathbb{R}^{n\times n}$. For additional mathematical information see the bibliography references.

4.2 Design

In this chapter the controller design using MPC technique is introduced. In Matlab is already present a particular toolbox which allows to design a controller in a simple way: *mpcDesigner*. Applying the above theory and considering that we are treating a SISO system, we have to change some parameters such as time horizon, control horizon and the weigths (we already have the sampling time $T_s = 10\ ms$). All these parameters can be chosen through the so called trial and error procedure in simulation, hence until the requirements are satisfied. We have to take into account that:

- if a polynomial parametrization is used, it is convenient to set $T_c = T_{pr}$, otherwise $T_c = T_s$;
- a large T_{pr} increases the closed-loop stability properties and a too large T_{pr} may reduce the tracking accuracy;
- Q, P and R are constants;
- the increase of Q and P leads to decrease the energy of x_i reducing the converging time and oscillations, while increasing R leads to decrease the energy of u_i reducing the command effort.

The MPC Simulink block used is:

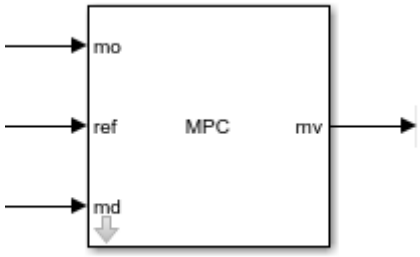

Figure 4.2: MPC Simulink block.

where:

- *mo* is the measured output (w_d);
- *ref* is the reference to follow;
- *md* is the measured disturbance (C_m);
- *mv* is the manipulated variable (C_f).

In the follows a table resuming the values used for the controller design:

Plot name	T_p	T_c	Q	P	R
MPC1	12	3	0.26	0	0.39
MPC2	16	6	0.26	0	0.39
MPC3	22	4	0.26	0	0.39

Table 4.1: MPC controller values.

The time results are illustrated below:

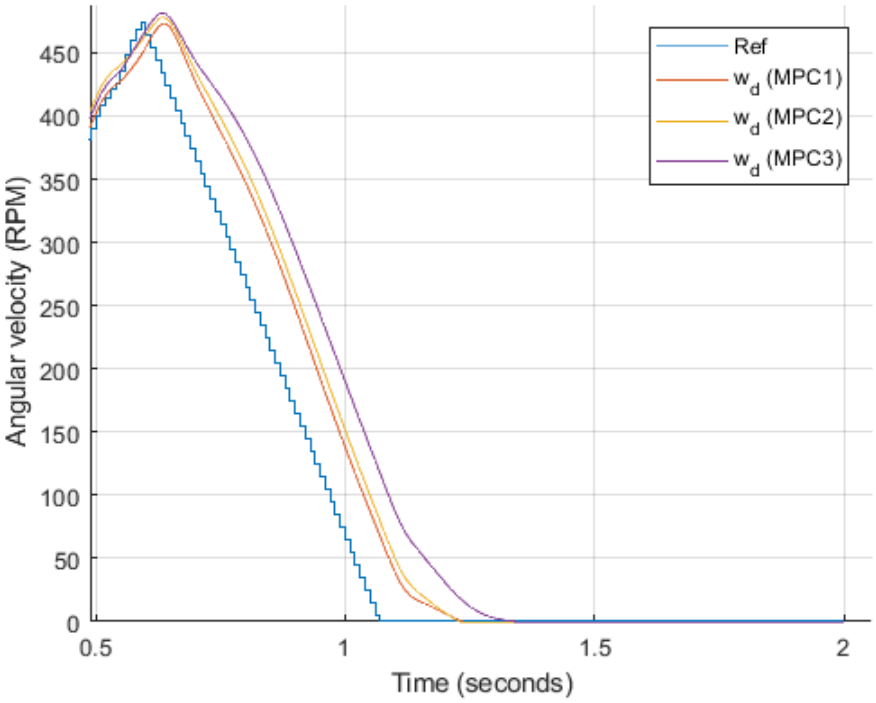

Figure 4.3: w_d responses with MPC.

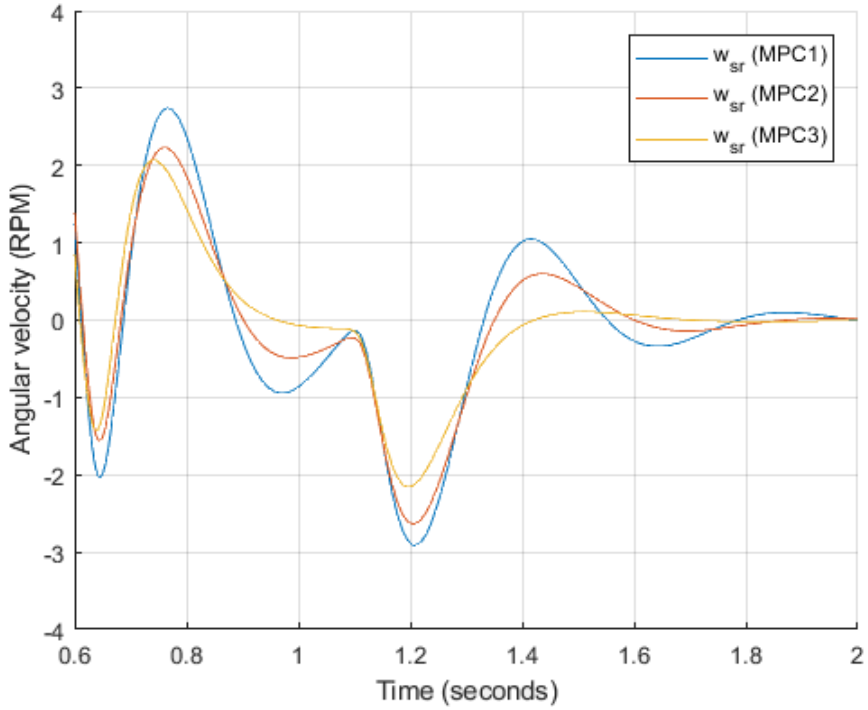

Figure 4.4: w_{sr} outputs with MPC.

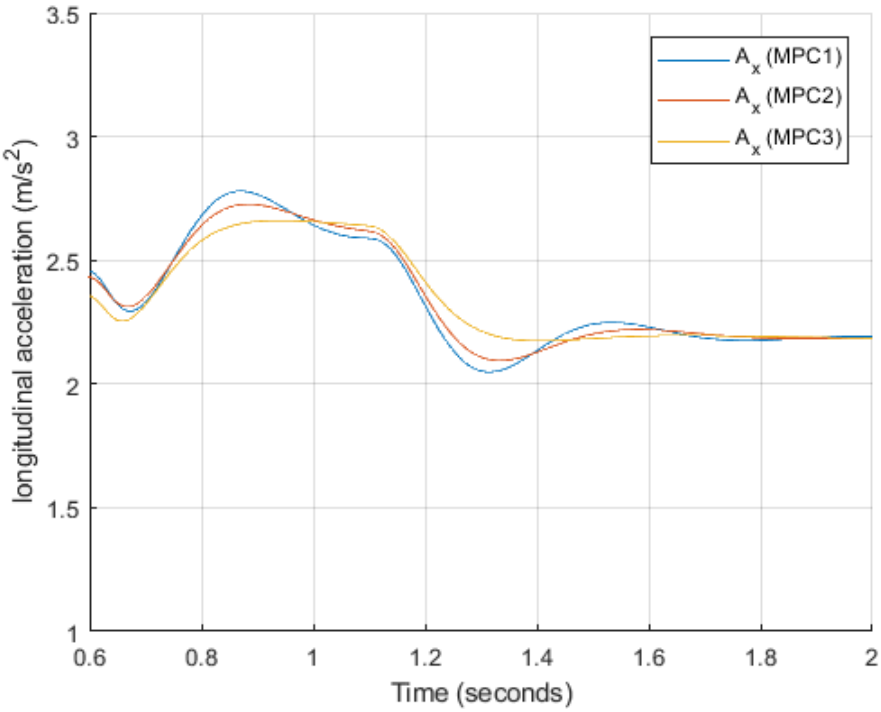

Figure 4.5: A_x values with MPC.

As we can see, the results in terms of oscillation on w_{sr} and A_x are better than the values obtained in the previous chapters.

4.3 MPC with two command input

We have seen so far that is difficult to manage the motor torque C_m. The idea is to use C_m as a command input instead as disturbance. In this way we have two command input (C_m and C_f), thus Q, P and R are 2×2 diagonal matrices. Furthermore it is possible to add a second reference on the w_d derivative ($\dot{w}_d$). The reason why we do this is to impose a smoother behavior on w_d. We obtained very good results using the following parameters:

- $T_p = 0.07$;
- $R = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}$;
- $Q = \begin{bmatrix} 1 & 0 \\ 0 & 0.05 \end{bmatrix}$;
- $P = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}$;

with constant input over the entire prediction time interval. In the follows, the results achieved with this values, which allow to obtain the MPC4 controller:

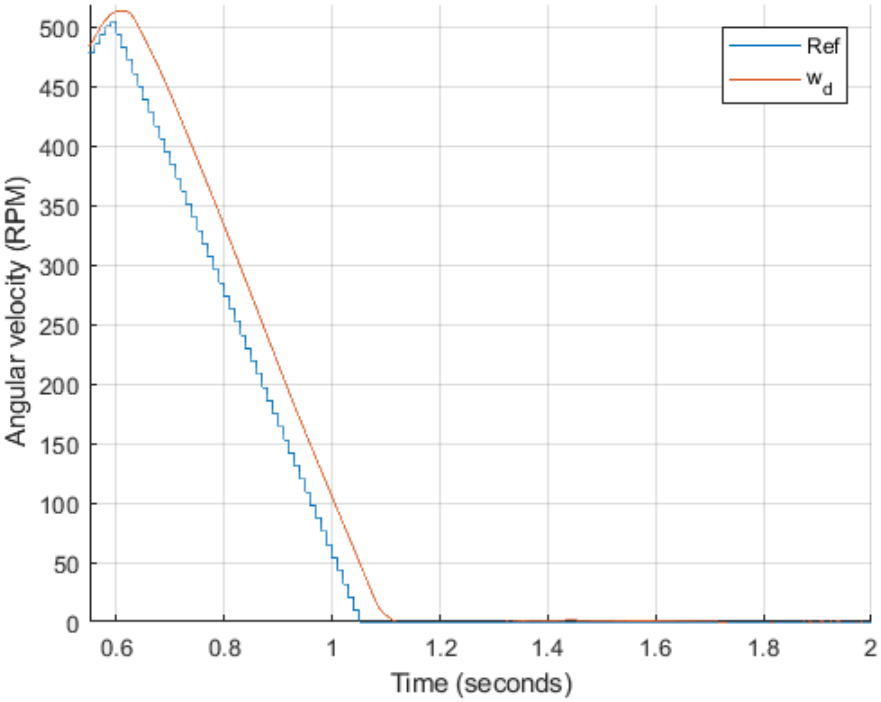

Figure 4.6: w_d response (MPC with two command input).

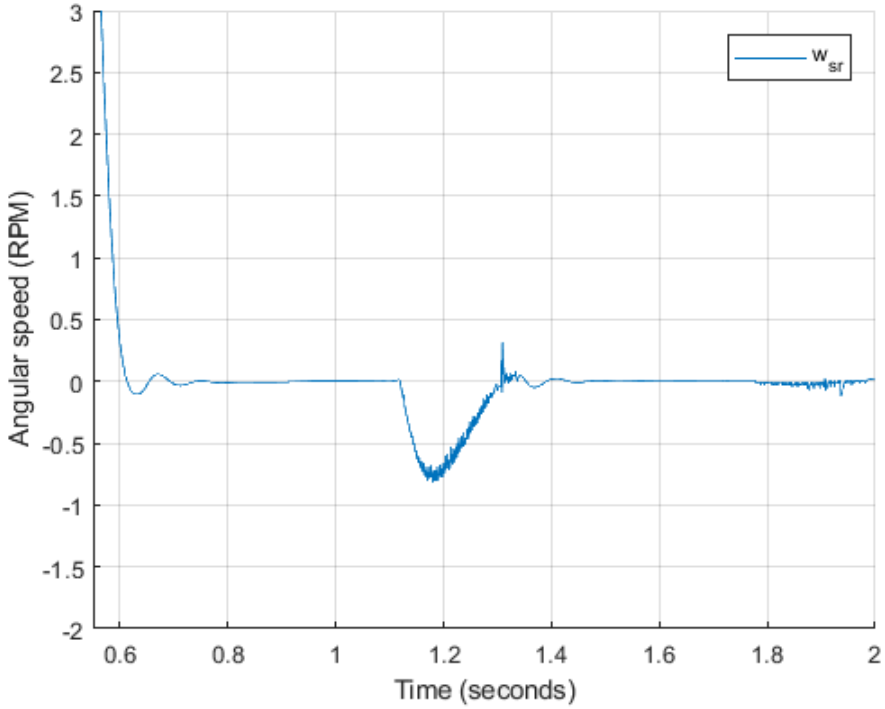

Figure 4.7: w_{sr} output (MPC with two command input).

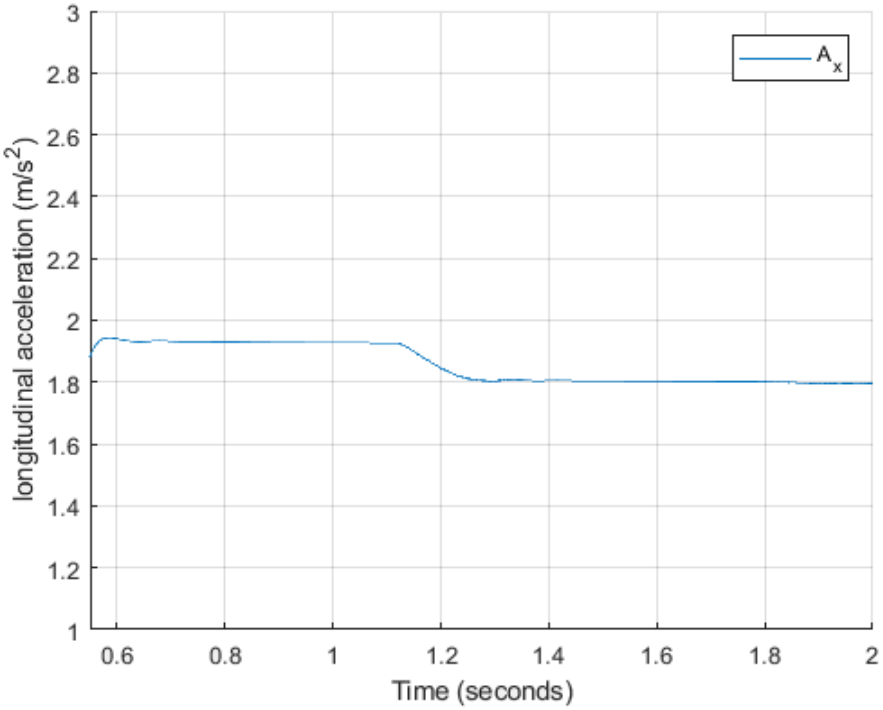

Figure 4.8: A_x value (MPC with two command input).

We can note that the results are excellent as matter of fact w_{sr} and A_x do not present oscillations.

Chapter 5

Comparison of used methods

In this chapter are shown the comparisons between the different methods utilized to design a controller for the DDCT. In particular, are compared only the best obtained performances of each single method. It is also present the preliminary PI controller. Note that we chose the best controllers by focusing on the simulation with CRF's detailed model. Finally a robust control analysis is presented.

It is important to clarify that the best performance is not the faster, but the one that ensure a good trade-off between speed and oscillations. We consider the time $t_a = t_f - t_s$, where t_s is the the starting control time (t_s =0.6 seconds) and t_f is the time in which w_d is equal to 0. With regard to the oscillations, we use the standard deviation σ, which is a measure used to quantify the amount of variation or dispersion of a set of data values. A low standard deviation indicates that the data points tend to be close to the mean of the set, while a high standard deviation indicates that the data points are spread out over a wider range of values.

Method	Controller	$t_a[s]$	$w_{sr}[\sigma]$	$A_x[\sigma]$
Loop Shaping	LS8	0.71	1.15	0.2
H_∞	$H_{\infty 1}$	0.79	0.87	0.17
MPC	MPC3	0.63	0.9	0.19
MPC	MPC4	0.52	0.2	0.06
PI	PI	0.88	0.7	0.18

Table 5.1: Recap of best controllers.

As reported in the table above, all controllers present a t_a smaller than PI. LS8 shows consistent values of standard deviation, while MPC3 and $H_{\infty 1}$ are able to damp oscillations. Further, these latter controllers have similar w_{sr} and A_x, but MPC3 goes to zero faster than $H_{\infty 1}$. MPC4 has no oscillations and its

convergence time is low with respect to the others. Hence, we can conclude that the best controller is designed through Model Predictive Control approach using C_m as command input.

5.1 Robust analysis

As final step we test if the designed controllers provide also robust performances with respect to (3.2). We used only the two critical values (i.e. $K_{usura} = 0.7$ and $K_{usura} = 1.3$). Resuming the analysis in the following table we have:

	LS8	$H_{\infty 2}$	MPC3	MPC4
$w_d(K_{usura} = 0.7)$	bad	good	bad	average
$w_d(K_{usura} = 1.3)$	bad	good	bad	average
$w_{sr}, A_x(K_{usura} = 0.7)$	high	low	average	average
$w_{sr}, A_x(K_{usura} = 1.3)$	average	average	average	average

Table 5.2: Robust analisys comparison.

From the table we can gather that only H-infinity controller, in particular $H_{\infty 2}$ provides good response when K_{usura} changes. Below a w_d plot obtained with that controller.

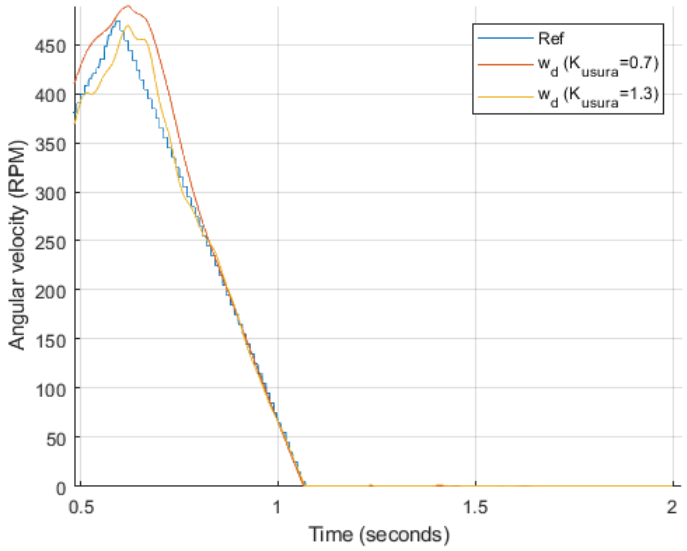

Figure 5.1: w_d responses with $H_{\infty 2}$ in Feedforward architecture with two different K_{usura} values.

It is important to underline that we did not use observers in order to estimate and then compensate the clutch wear.

Chapter 6

Conclusions

In this thesis we presented LS, H_∞ and MPC approaches to the slip control problem of a DDCT. The main purpose was to control the clutch torque (C_f) in order to ensure a smooth clutch engagement, improving drivability and driver comfort.

For what concern the LS approach we highlighted the importance of loop function $L(s)$ and how obtaining a good controller modifying its shape. We also saw how to translate time constraints into frequency constraints. LS designed controllers provided good track reference performance but bad oscillations on the driveline. The main problem of this method concerns the motor torque C_m. It is difficult to handle specially in the first 0.5 seconds, when has a sudden growth. The other problem lies in the non-linearities (delay block and saturation block inside actuator) which makes worse w_{sr} output. However, the simulation using the detailed model attested good results even in terms of oscillations on the driveline, due to the presence of static controller .

In the third chapter we introduced another frequency approach. We illustrated how to extract weighting function from time constraints and how to get a controller using the LMI optimization, taking also into account the uncertainty of the system. The problems concerning this method are more or less equal to the previous one. H_∞ is able to track the reference in a very small time but at the cost of high oscillations. It also proves to be a good robust method to design a controller. Even in this case the simulation on the detailed model gave good results.

In addition to the frequency design approaches, we described also a different method: MPC. We showed that the concept of this method is to make a future prediction considering a linear model of the plant. MPC allows also to regulate

two values such as x_i and the controller output u_i acting on three proper weights. In this way MPC is able to handle the disturbance motor torque C_m. The obtained results are good even for what concern oscillations. In this chapter we also illustrated the possibility to exploit C_m as command input instead as disturbance, the results were very good because oscillations disappear

In order to have an overview of the designed controllers we made a comparison based both on time convergence t_a and on standard deviation affecting oscillations. From the results appeared that frequency approaches guarantee a good convergence time to the detriment of huge oscillations on the driveline. We can conclude that the frequency approaches are not suitable to manage this kind of system, for which MPC with is more appropriate.

6.1 Future Works

Starting from the results obtained in this thesis project, other analysis can be conduct to improve the final results, such as:

- employment of w_{sr} as controlled output;
- MPC adding an integrator in the open-loop chain to take the steady-state reference error to zero;
- MPC control unit implementation;
- control algorithm experimental test using a real prototype;
- use of observer to provide better results in robust analysis.

Bibliography

[1] C.Vafidis, F.Cimmino, "FPT's High Torque Density Dual Clutch Transmission (HTDDDCT)", 8th CTI Symposium "*Innovative Automotive Transmissions*", Berlin, 2009.

[2] L. Glielmo, L. Iannelli, V. Vacca, and F. Vasca. Gearshift control for automated manual transmissions. *Mechatronics, IEEE/ASME Transactions on*, 11(1):17-26, 2006.

[3] T. Jin, P. Li, and G. Zhu. Optimal decoupled control for dry clutch engagement. *In American Control Conference (ACC), 2013*, pages 6740-6745. IEEE, 2013.

[4] P. J. Dolcini, C. Canudas-de Wit, and H. Bechart. *Dry clutch control for automotive applications*. Springer Science & Business Media, 2010.

[5] Astrom, Karl Johan, and Richard M. Murray. *"Feedback systems.", Princeton Univ (2008),* Chapter 9.

[6] Vito Cerone, Massimo Canale, and D Regruto. Loop-shaping design with constant magnitude loci in control education. *International Journal of Engineering Education*, 24(1):127, 2008.

[7] Kemin, Z. H. O. U., and JOHN C. Doyle. *"Essential of Robust Control,(1995).",* Chapter 6.

[8] V. Cerone, D. Regruto. Slide of Modern Design of Control Systems, 2018.

[9] Camacho, Eduardo F., & Alba, Carlos Bordons (2013). *Model predictive control*. Berlin: Springer.

[10] Di Cairano, Stefano, & Bemporad, Alberto (2010). Model predictive control tuning by controller matching. *IEEE Transactions on Automatic Control*, 55(1), 185–190.