POLITECNICO DI TORINO

Department of Control and Computer Engineering
(DAUIN)

Master of Science in Mechatronic Engineering

Master Degree Thesis

Actuator Position Control of Dual Clutch Transmission Systems using Model Predictive Control Techniques

---

Supervisors:
Prof. Vito Cerone
Prof. Massimo Canale
Prof. Diego Regruto

Author:
Stefano Coretti

December 2018

Contents

List of Figures

List of Tables

Introduction

Since the early nineties, interest in fuel efficency and driving comfort has increased dramatically in the automotive industry to enhance the vehicle commercial success. Improvements in these two aspects have been achieved thanks to several researches for the developement of advanced transmision and powertrain systems.

In this context, a remarkable result is represented by the **Automated Manual Transmission (AMT)** systems. They offer the Automated Transmission (AT) efficency preserving the Manual Transmission (MT) low fuel consumption. This is because in the AMT systems, the advantages of traditional torque mechanic transmission are combined with an automatic clutch actuation performed by a control unit during the gear shifting operations.

Moreover, the AMT systems are nowadays equipped with the **Dual Clutch Transmission (DCT)** technology yielding further improvements regarding the driving comfort. Indeed, a DCT systems between the engine and the gear transmission is able to alternate torque demands from one clutch to the other clutch without power interruption during the shift process. The result is a rapid gear shifting with an efficient fuel consumption and riding comfort. Furthermore, the use of dry clutches is less expensive since they are actuated by a low cost electromechanical system.

In such a Dry Dual Clutch Transmission (DDCT) system, the clutch engagement torque is proportional to the stroke of the actuator. As a consequence, the position control effectiveness of the clutch servo system determines the overall performance in relation to the driving behaviour. A mismatch between the command from the transmission control unit and the actual position signal results in an ineffective clutch torque. Thus, an accurate position control of the clutch actuator is needed for effortless driving without any torque interruption.

In this regard, the following thesis proposes a **position control strategy for the even gear actuator (K2 Actuator)** of a DDCT system. The aim of the project, developed in collaboration with **Centro Ricerche Fiat (CRF)**, is to design a controller to track different position trajectories guaranteeing smoothness during the gear shifting process with a continue torque transmission.

The thesis is organized as follows.

First of all, a brief overview of the DDCT system and its components, with a focus on the K2 Actuator structure, is introduced. The physical relationships between the involved variables are outlined and the control objectives are defined along with the performance requirements.

In the second Chapter, a **Model Predictive Control (MPC)** strategy is proposed for the position control of the K2 Actuator. This decision has been motivated by the wide versatility of the optimization problem formulation offered by the MPC framework and by the possibility of explicitly considering physical constraints on the input variables during the design procedure.
Once provided a compact overview of the MPC main theoretical aspects, a mathematical model of the K2 Actuator is achieved by means of a system identification procedure. The obtained results have highlighted a large variability of the K2 Actuator dynamics with respect to the operating region. For this reason, an Adaptive Model Predictive controller has been designed on the basis of a real time varying state space representation of the system. Moreover, it is shown how the MPC cost function and optimization problem can be suitably customized for the K2 Actuator position control problem. The tuning procedure of the controller design parameters has been performed by carrying out extensive simulations until satisfactory performances are obtained. These performances have been further improved by means of a real time adjustment of the controller design parameters on the basis of the working situation.
Finally, simulations results are presented in order to test the effectiveness of the proposed control strategy.

The third Chapter deals with the application of a **Linear Quadratic Regulator (LQR)** control technique to the K2 Actuator position control problem. Even if constraints on the involved variables are not handled, this approach allows to evaluate the optimal control action in a static state feedback form so that computational aspects can be enhanced. Indeed, a possible numerical implementation is proposed in the final section of this Chapter.

In this case, the identification methodology has been performed considering an Hammerstein model structure for the K2 Actuator plant. As for the MPC, an adaptive approach and a dynamic tuning of the controller design parameters has been developed to maintain the same level of control system performances despite the plant model dynamics are quickly changing on the basis of the working region.

In the fourth Chapter, the absence of a real position sensor in the described DDCT system is considered by exploiting a **Virtual Sensor based Control** architecture. Two types of virtual sensor are considered to provide the best position estimate on the basis of the K2 Actuator pressure actual value. Their inclusion in different control architectures is discussed by comparing several simulations results in order to decide the most suitable virtual sensor model structure and control technique.

The last Chapter reports some overall concluding considerations about the whole thesis work. Possible future developements are suggested in order to improve the obtained results.

Chapter 1

The Dry Dual Clutch Transmission System

This Chapter provides a brief outlook of the Dry Dual Clutch Transmission system with a particular address to the involved even gear actuator. In regard to the actuator position control of such a transmission system, the objectives and performance requirements of the whole thesis work are outlined in the last section.

1.1 Overview of Transmission Systems

The transmission system of an automotive vehicle represents the connection between the engine and the driver. Indeed, according to the driver's request, the internal engine power is converted and adapted to the wheels yielding the overall vehicle traction. Because of their outstanding importance, transmission systems have to be designed ensuring an efficient trade off between speed, climbing performance, acceleration and fuel consumption without never overlooking the driving comfort.

Nowadays, passengers vehicles are commonly equipped with **Manual Transmission (MT)** or **Automated Transmission (AT)** systems. Their geographic diffusion is different with respect to market demands. In particular, European consumers prefer the low cost and the full driving control offered by the manual transmission whereas in US and Japan, where comfort and ergonomics are promoted, the automatic transmission is more popular.

A good compromise between these two technologies is represented by the **Automated Manual Transmission (AMT)** systems. Thanks to the fusion between the

traditional torque mechanic transmission and an automatic clutch actuation, AMT systems are able to offer the AT efficiency still preserving the MT low fuel consumption. Nevertheless, one of the most important problem of the AMT systems during the gear shifting phase, is the presence of torque interruption that affects the driving comfort. For this reason, the Dual Clutch Transmission (DCT) has been developed to overcome such a drawback in the AMT systems.

1.1.1 The Dual Clutch Transmission for AMT systems

The Dual Clutch Transmission system was introduced in the first half of the twentieth century by the french military engineer Adolphe Kgresse. With the aim of providing a continue torque transmission, two input shafts are employed in the DCT system such that the torque demand is alternated from one clutch to the other without power interruption.

The DCT systems are characterized by the following features

- Gear pre-selection: the synchronization of the oncoming gear has been completed before the actual gear shifting procedure starts.
- Overlapping mechanism: the two clutches are wrinkled each other such that the needed torque is transferred from the engine to the driving wheels without any interruption during gear shifting.

The gear shifting process is usually handled in a fully automatic manner but also the driver manual selection is possible.
Specifically, the traditional torque converter is shelved in favour of a continue torque transmission with a low fuel consumption. This is because, unlike the disconnection between the engine and the wheels during the gear change, the DCT maintain a constant traction yielding excellent power transmission and efficiency.

As for the DCT types, two forms are employed with respect to the gears meshing method. **Wet Dual Clutch Transmissions (WDCT)** use oil bathed clutches for cooling and are able to provide torque values up to 350 Nm. On the contrary, friction between the clutches is used as meshing strategy in **Dry Dual Clutch Transmission (DDCT)** systems. The advantage of employing dry clutches with respect to the wet DCT is motivated by the capability of reducing pump losses and by the opportunity to use a low cost actuation system.

The general configuration of a DDCT system is showed in Figure 1.1.

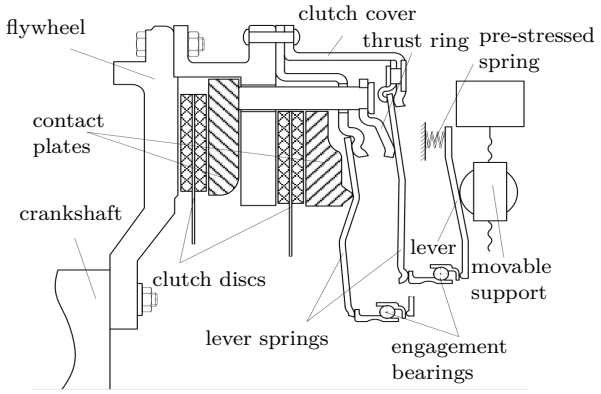

Figure 1.1: General configuration of Dry Dual Clutch Transmission System.

The showed common layout is composed of two independent electromechanical actuators that handle the twin clutches during a gear shift. The clutch engagement pressure force is related to the longitudinal position of the moveable support between the pre-stressed spring and the lever ratio.

However, unlike the described general configuration, modern Dry Dual Clutch Transmissions are actuated by means of fast electro-hydraulic systems as in the case of the Fiat Power-train C635 DDCT, presented in the following section.

1.2 The C635 Dry Dual Clutch Transmission System

In this thesis project, the Dry Dual Clutch Transmission system, developed and released in 2010 by **Fiat Power-train Technologies**, is considered. This kind of DDCT system is part of the the new C635 transmission family. They consist of a range of manual or transversal DDCT transmissions, characterized by a 6-speed and all wheel drive with a maximum input torque of 350 Nm and output torque of 4200 Nm.

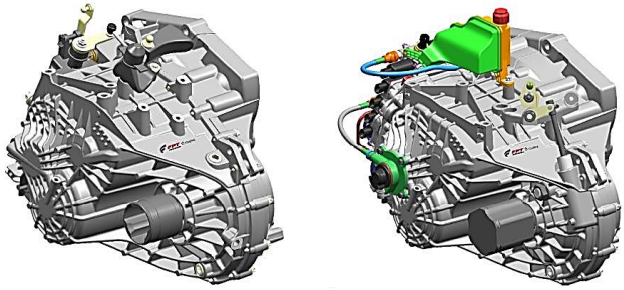

Figure 1.2: C635 MT and DDCT versions.

1.2.1 System Architecture and Features

The C635 DDCT system architecture, showed in Figure 1.3, is composed by three input shafts and contained in a two piece aluminium structure.
In particular, due to installation constraints, the gear set housing presents a reduced length of the upper secondary shaft yielding an efficient packaging even in the lower segment vehicles.

Moreover, it is worth to highlight the different actuation systems involved in the C635 DCCT. A coaxial pull-rod is adopted for the actuation of the **odd-gear Clutch (K1)**, while the **even-gear Clutch (K2)** is actuated by means of a rather conventional hydraulic Concentric Slave Cylinder (CSC).

In such a C635 DDCT, the torque transmission mechanism is related to the overlap between the engagement of the on-going clutch and the release of the off-going clutch. Both the twin clutches are installed on the proper housing by means of a single main

support bearing. This compact mounting solution is favoured by the low thickness of the chosen K1 actuation system.

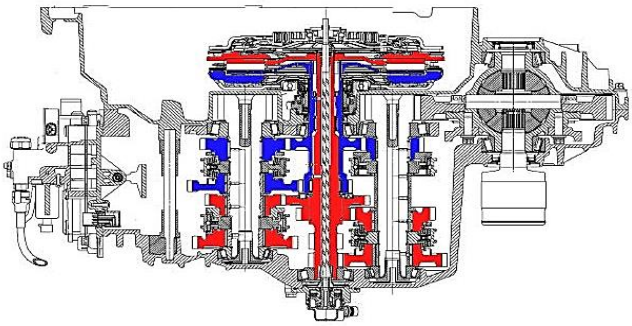

Figure 1.3: C635 DDCT cross section.

Another remarkable feature is the fact that a contact-less linear position sensor can be integrated in the odd gears actuation system, hence allowing the position control of the K1 clutch. On the contrary, the even gear clutch must be controlled in force by means of the hydraulic pressure provided by the CSC. The just outlined aspect is crucial for this thesis work since affects the K2 Actuator position control strategy.

1.2.2 Control Unit

The different algorithms, involved in the C635 DDCT control unit, run in a multitasking environment so that the Main Micro Controller resources are suitably managed. The following control problems are considered

- **Actuator Control:** the aim is to improve the electro-hydraulic clutch actuation.
  - Engagement Actuators Control: the desired trajectories are evaluated by commanding the relevant pressure valves one against the other.
  - Shifter Control: the command action is able to push the shifter piston against the proper spring in order to reach the desired position level.
  - Odd Gears Clutch Control: a position closed loop controls the first and the reverse gears clutch (K1).
  - Even Gears Clutch Control: the even gear clutch (K2) is controlled in force thanks to the pressure feedback provided by the associated sensor.

- **Self-Tuning Control:** with the aim of guaranteeing the same high-level calibrations to all vehicles, different self tuning control algorithms have been developed mainly concerning the transmitted torque conversion to the K1 position and K2 pressure.
- **Launch and Gear Shift strategies:** different shift patterns are considered and exploited by specific control and calibration strategies both in automatic and manual driving mode.

1.2.3 Electro-hydraulic Actuation System

A dedicated, sealed, hydraulic oil circuit is responsible for the actuation of the C635 DDCT system components. Such an electro-hydraulic actuation strategy guarantees good compactness improving, at the same time, the overall DDCT system performances. The actuation system main components are

- **Hydraulic Power Unit:** consist of a high pressure accumulator and an electrically-driven pump (Figure 1.4(a)).
- **Actuation Module:** includes the sensors, the gear shift actuators and the solenoid valves (Figure 1.4(b)). It can be divided in
  - four distinct double action pistons operating the gear engagement forks;
  - one shifter spool which selects the piston to be actuated;
  - five solenoid valves composed of four pressure proportional valve (PPV) and one flow proportional valve (QPV).

(a) (b)

Figure 1.4: C635 DDCT Hydraulic Power Unit (a) and Actuation Module (b).

Specifically, as far as the Actuation Module is concerned, two PPVs are related to the the gear engagement, while the third commands the spool valve that selects the associated piston. Finally, the clutches K2 and K1 are, respectively, controlled by the fourth PPV and the QPV.

Different contact-less position sensor are also included in the C635 Actuation Module, one for each engagement piston and one for the shifter pool. One pressure sensor is related to the K2 clutch control system and one exploits monitoring functions.

The overall hydraulic circuit of the C635 DDCT Actuation System is showed in Figure 1.5 below.

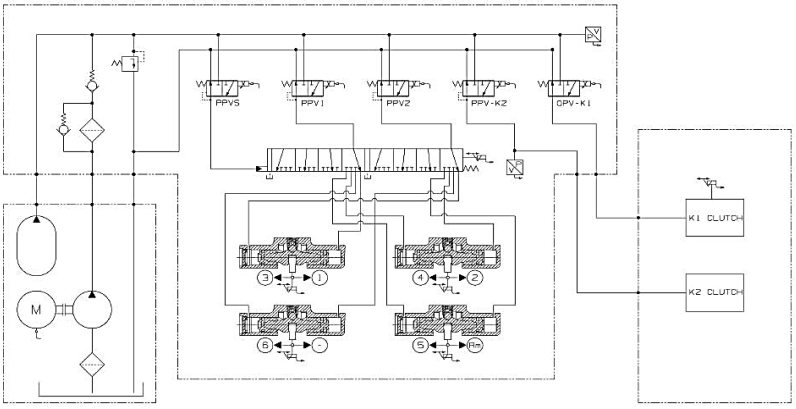

Figure 1.5: C635 DDCT complete actuation circuit.

1.3 The C635 even gear Actuator

Referring to the whole actuation module described in section 1.2.3, the fourth proportional pressure control valve (PPV), associated with the K2 Clutch, is specially considered here. The choice is motivated by the aim of this thesis project, that is to develop a control strategy for the even gear actuator of the C635 DDCT system.

In this regard, the basic physical properties are described along with the PPV general structure and components. Finally, control objectives and requirements of this thesis work are presented.

1.3.1 Essential elements of Proportional Pressure Control Valves

As already mentioned in section 1.2.3, the K2 Clutch, associated with the even gear engagement, is controlled by a proportional control valve. Specifically, these kind of electro-hydraulic valves can be considered as a trade-off between the cheaper solenoid valves and the outperforming servo valves in the sense that they preserve low manufacturing costs with the only drawback of a slight performance worsening. Anyway, PPVs provide an economical and satisfactory alternative for many applications and are therefore commonly employed in transmission systems.

The K2 Actuator working principle is essentially based on the mechanical forces deriving from magnetic interactions. That is, a proper solenoid is able to exert an output force, hence an output pressure, that is related to the current flowing through its wires. The reason behind the choice of a current based PPV, is to reduce temperature leakages associated with a voltage control system so that the overall efficiency is preserved.

The general configuration of a PPV is showed in Figure 1.6 below.

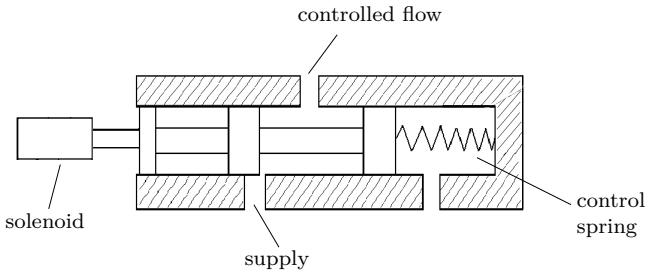

Figure 1.6: Schematic section of a PPV.

The output pressure is regulated by controlling the flow of the hydraulic fluid through the orifice of the PPV. More precisely, the valve spool is seated between a compression spring and a proportional solenoid. In this way, the orifice size is controlled via the spring deflection by modifying the mechanical force produced by the solenoid and related to the input current. This strategy leads to an almost proportional relationship between the valve output pressure and the input current flowing through the solenoid.

Concerning the possible drawbacks associated with these PPVs, overlapped spools have to be used because of the difficulties in producing a zero lap spool. This means that no output pressure is obtained until the input current exceeds a proper value that is required to overcome the spring pre-load and the spool overlap. Figure 1.7 graphically express this aspect by showing the dead zone that is present between the input current and the output pressure.

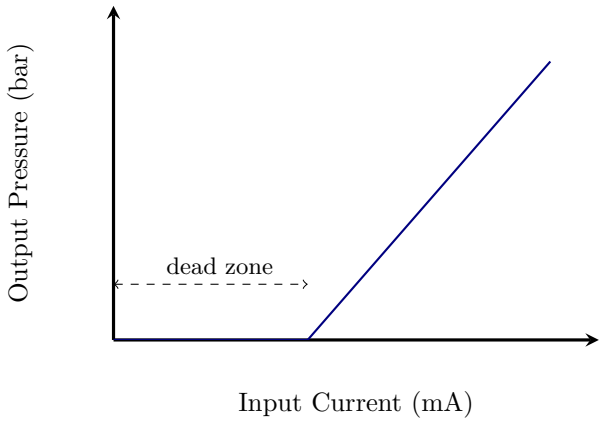

Figure 1.7: Pressure current characteristic of a PPV.

Another important feature of proportional pressure valves is the hysteresis effect. In fact, depending on whether the current is increasing or decreasing, a considerable difference in the valve output pressure takes place. This is because the valve relies on the force exerted by the solenoid acting against the spring to move the spool.

Both the dead zone and the hysteresis are crucial aspects in regard to the even gear actuator control strategy. The former is specifically accounted by one of the proposed control strategies (details in section 3.2.1), the latter is treated in [1].

1.3.2 K2 Actuator position control objectives

With the aim of improving driving comfort and guaranteeing a smooth gear shift process, a position control strategy for the even gear actuator is proposed in this thesis work. Different position reference trajectories have been considered and the tracking problem has been addressed by accounting the following performance requirements, provided by Centro Ricerche Fiat (CRF).

- Position Overshoot: $\hat{s} \leq 5\%$
- Rise Time: $50 \leq t_r \leq 120$ ms
- Steady State Error for step reference: $|e_r^{\infty}| \cong 0$
- Input Current: $0 \leq I_{cmd} \leq 1000$ mA
- Pressure: $0 \leq p \leq 40$ bar

In the following Chapters, different control architectures and model identification approaches are proposed to develop an efficient position controller for the K2 Actuator.

Chapter 2

Adaptive Model Predictive Control for K2 Actuator

The main purpose of this chapter is to provide a compact overview of the essential elements of model predictive control (MPC) along with its application to the K2 Actuator Control problem.

2.1 MPC Overview

Control systems based on the MPC concept have gained popularity in a wide range of applications in different engineering fields due to their ability to yield high performance control systems together with the facility of flexible constraints handling.
These peculiarities are conditioned by explicitly considering the model of the system to obtain the control action as a result of a constrained optimization problem. A numerical optimization problem has to be solved at each sampling time and the computational effort is consequently quite high. For this reason, originally, MPC was mainly employed for systems with slow dynamics and large computational resources such as chemical or aerospace processes. However, in the last years, significant improvements of microprocessors and computer power allowed Model predictive control to be applied in faster system like mechatronic and automotive applications.

Model predictive control has its roots in optimal control. The key idea of MPC is to use a mathematical model to forecast the system behavior, in order to determine the best control actions to apply over some period as a result of a constrained optimization problem [10], [8].
The MPC architecture is composed of (see Figure 2.1)

- the prediction model;
- the cost function and the constraints;
- the optimizer;
- the controlled system.

All these aspects will be discussed in the following sections of this chapter.

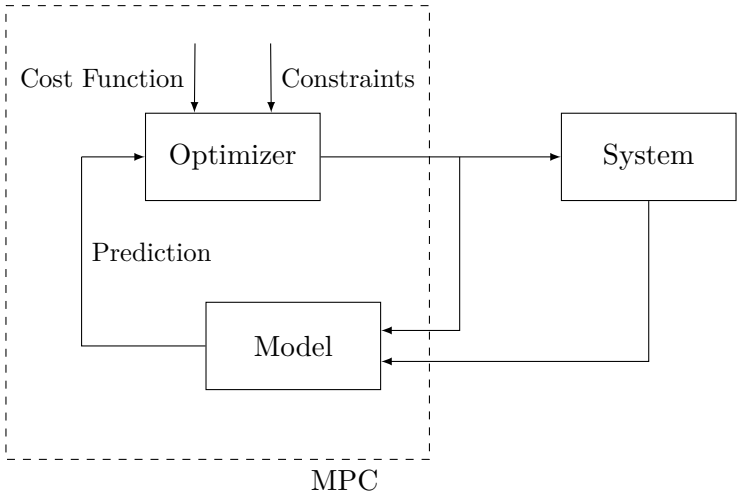

Figure 2.1: General Scheme of Model Predictive Control

The most relevant benefits of exploiting MPC strategy can be outlined in

- The opportunity to explicitly include both control input and state variables constraints;
- The capacity to manage multi-variable control problems;
- The possibility to trade off between different control objectives by tuning some critical parameters.

2.1.1 Prediction Model

Discrete time prediction models are often suitable if the considered system is sampled at discrete times. If the sampling rate is properly chosen, the behaviour between the samples can be safely ignored and the model describes exclusively the behaviour at the sample times.

The more general representation for the prediction model is a non linear, time invariant state space system of the following form

$$\begin{aligned} x(k+1) = &\ f\big(x(k), u(k)\big) \quad f \in \mathbb{C}^1 \\ y(k) = &\ g\big(x(k), u(k)\big) \quad g \in \mathbb{C}^1 \end{aligned} \tag{2.1}$$

where $x(k) \in \mathbb{R}^n$ is the state variable, $y(k) \in \mathbb{R}^p$ is the system output, $u(k) \in \mathbb{R}^m$ is the control input.

Assuming that all the system state variables $x(k)$ are measurable, the prediction of the model expressed in (2.1) consists in considering the states evolution from a time instant k over a certain number of time steps in the future.
The length H_p of the finite optimization horizon is referred as **prediction horizon**.

In order to easily approximate and analyse physical systems, the prediction model (2.1) often consists in a Linear Time Invariant (LTI) discrete time system, described by the following state space representation

$$\begin{aligned} x(k+1) = &\ Ax(k) + Bu(k) \\ y(k) = &\ Cx(k) \end{aligned} \tag{2.2}$$

where $A \in \mathbb{R}^{n,n}$, $B \in \mathbb{R}^{n,m}$, $C \in \mathbb{R}^{n,p}$.

For such a LTI System the i^{th} step ahead state prediction $x(k+i|k)$ can be expressed as

$$\begin{aligned} x(k+i|k) = &\ A^i x(k|k) + A^{i-1}Bu(k|k) + A^{i-2}Bu(k+1|k) + \cdots + Bu(k+i-1|k) \\ = &\ A^i x(k|k) + \sum_{j=0}^{i-1} A^{i-j-1}Bu(k+j|k). \end{aligned} \tag{2.3}$$

Therefore, the prediction model (2.2) depends only on the current state $x(k|k)$ and on

the control sequence $U(k) = [u(k|k)\ u(k+1|k)\ ...\ u(k+H_p-1|k)]$.

2.1.2 Cost Function and Optimization Problem

The MPC strategy is related to the minimization, over a specified finite prediction horizon H_p, of a cost function whose general expression is

$$J(x(k|k), U(k)) = \Phi\big(x(k+H_p|k)\big) + \sum_{i=0}^{Hp-1} L\big(x(k+i|k), u(k+i|k)\big) \tag{2.4}$$

where

- $x(k|k)$ is the state measurement at current time k.
- $x(k+i|k)$ is the i^{th} step ahead state prediction.
- $U(k) = [u(k|k)\ u(k+1|k)\ ...\ u(k+H_p-1|k)]$ is the command input sequence to be optimized.
- $L(\cdot)$ is the per-stage weighting function.
- $\Phi(\cdot)$ is the terminal state weighting function.

The **weighting functions** $L(\cdot)$ and $\Phi(\cdot)$ are assumed to be continuous in their arguments and are considered as design parameters to be suitably chosen according to the desired control performances.

Therefore, considering the generic non linear system expressed in (2.1), the **constrained finite time optimization problem** assumes the following form

$$\begin{aligned}
&U^* = \ \arg\min_{U}\ J(x(k|k), U(k)) \\
&\text{subject to} \\
&x(k+1) = \ f\big(x(k), u(k)\big) \\
&x(k+i|k) \in \mathcal{X}, \quad i = 1...H_p - 1 \\
&u(k+i|k) \in \mathcal{U}, \quad i = 1...H_p - 1 \\
&x(k+H_p|k) \in \mathcal{X}_f
\end{aligned} \tag{2.5}$$

where

- $\mathcal{X} \in \mathbb{R}^n$ and $\mathcal{U} \in \mathbb{R}^m$ are polyhedra representing respectively the states and the input constraints invariant sets (details in [10]).

- $\mathcal{X}_f$ is the terminal polyhedral region introduced in the optimization problem to ensure asymptotic stability (details in [19]).

- $U^*(k) = [u^*(k|k)\ u^*(k+1|k)\ ...\ u^*(k+H_p-1|k)]$ is the optimal input control sequence.

2.1.3 Control Input and State Constraints

The possibility to handle input and state constraints is the main quality that distinguishes MPC from the standard linear quadratic (LQ) control.

For systems subject to external inputs such as (2.1) and (2.2), the two regions $\mathcal{X} \in \mathbb{R}^n$ and $\mathcal{U} \in \mathbb{R}^m$ are referred as **control invariant sets** ([10], [20], [3]). These two convex regions are useful to answer questions such as: Find the set of initial states for which there exists a controller such that the system constraints are never violated. For the sake of simplicity, in the following it will be assumed that both $\mathcal{X}$ and $\mathcal{U}$ are reachable polytope containing the origin in their interior so they will be modeled as sets of inequalities.

Regarding control input constraints, the set $\mathcal{U}$ is usually chosen to take care of the **actuator devices physical limitations**. For this reason, input actuator saturation and slew rate constraints can be managed with the following formulation

$$\begin{aligned} U^{min} &\leq u(k+i|k) \leq U^{max},\ i = 1...H_p - 1 \\ \Delta U^{min} &\leq u(k+i|k) \leq \Delta U^{max},\ i = 1...H_p - 1 \end{aligned} \tag{2.6}$$

where

- U^{min} and U^{max} are respectively the constraint vector associated with the minimum and maximum control input value.

- ΔU^{min} and ΔU^{max} are respectively the constraint vector associated with the minimum and maximum control input variation rate.

In order to clarify this aspect, a prediction horizon $H_p = 2$ is considered and the control input vector becomes simply $U(k) = [u(k|k)\ u(k+1|k)]$. The set of constraints on the control input value can be rewritten as

$$\begin{matrix} u^{min} \leq u(k|k) \leq u^{max} \\ u^{min} \leq u(k+1|k) \leq u^{max} \end{matrix} \Rightarrow \begin{matrix} \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} u(k|k) \\ u(k+1|k) \end{bmatrix} \leq \begin{bmatrix} u^{max} \\ u^{max} \end{bmatrix} \\ \begin{bmatrix} -1 & 0 \\ 0 & -1 \end{bmatrix} \begin{bmatrix} u(k|k) \\ u(k+1|k) \end{bmatrix} \leq \begin{bmatrix} u^{min} \\ u^{min} \end{bmatrix} \end{matrix}$$

$$\Rightarrow \underbrace{\begin{bmatrix} I \\ -I \end{bmatrix}}_{L_{U_v}} \begin{bmatrix} u(k|k) \\ u(k+1|k) \end{bmatrix} \leq \underbrace{\begin{bmatrix} u^{max} \\ u^{max} \\ u^{min} \\ u^{min} \end{bmatrix}}_{W_{U_v}} \Rightarrow \quad L_{U_v} U(k) \leq W_{U_v} \tag{2.7}$$

In a similar way the matrix inequalities for the slew rate constraints are obtained in the following form

$$L_{U_{sr}} U(k) \leq W_{U_{sr}} \tag{2.8}$$

Therefore, combining (2.7) and (2.8) the overall input actuator constraints of the form (2.6) can be rearranged as

$$L_U U(k) \leq W_U \tag{2.9}$$

As to the state constraints, the set $\mathcal{X}$ is often related to the performance requirements. More precisely state constraints are useful to impose **limitations on output variables** e.g. to mitigate overshoots or inverse behaviour in the system response. State constraints can be managed with the same above formulation

$$\mathbf{x}^{min} \leq x(k+i|k) \leq \mathbf{x}^{max}, \; i = k...H_p \tag{2.10}$$

where $\mathbf{x}^{min}$ and $\mathbf{x}^{max}$ are respectively the constraint vector associated with the minimum and maximum value of each state.

Considering the LTI prediction model (2.2) the state constraints can be expressed as linear constraints in $U(k)$. In fact, choosing again a prediction horizon $H_p = 2$, the state boundaries are

$$\begin{matrix} L_{x_1} x(k+1|k) \leq W_{x_1} \\ L_{x_2} x(k+2|k) \leq W_{x_2} \end{matrix} \Rightarrow \begin{matrix} L_{x_1}\big(Ax(k|k) + Bu(k|k)\big) \leq W_{x_1} \\ L_{x_2}\big(A^2 x(k|k) + ABu(k|k) + Bu(k+1|k)\big) \leq W_{x_1} \end{matrix}$$

where the step ahead prediction state expression (2.3) has been applied. Further, the above expression of the state constraints can be written as a linear matrix inequality in $U(k)$

$$L_x U(k) \leq W_x \tag{2.11}$$

where

$$\begin{aligned} L_x &= \begin{bmatrix} L_{x_1} & 0 \\ 0 & L_{x_2} \end{bmatrix} \begin{bmatrix} B & 0 \\ AB & B \end{bmatrix} \\ W_x &= \begin{bmatrix} -L_{x_1} & 0 \\ 0 & -L_{x_2} \end{bmatrix} \begin{bmatrix} A \\ A^2 \end{bmatrix} x(k|k) + \begin{bmatrix} W_{x_1} \\ W_{x_2} \end{bmatrix} \end{aligned} \tag{2.12}$$

2.1.4 Quadratic Programming

As discussed in section 2.1 Model Predictive Control strategy is based on solving a constrained optimization problem related to the minimization of a cost function. Therefore, a suitable choice for the weighting functions $L(\cdot)$ and $\Phi(\cdot)$ expressed in (2.4) guarantee an efficient set up for the MPC control problem.
For instance, for output or states regulations, the following quadratic form is commonly chosen

$$\begin{aligned} L(\cdot) &= x(k+i|k)^T Q x(k+i|k) + u(k+i|k)^T R u(k+i|k), \quad i = 0...H_p - 1 \\ \Phi(\cdot) &= x(k+H_p|k)^T P x(k+H_p|k) \end{aligned}$$

where $Q \succeq 0$, $P \succeq 0$ and $R \succ 0$ are symmetric **weighting matrices** to consider as tunable design parameters to reach the control objectives.

By substituting the above weighting functions in (2.4), the cost function to be minimized in the MPC optimization problem can be rewritten as

$$\begin{aligned} J(x(k|k), U(k)) = x(k+H_p|k)^T P x(k+H_p|k) + \sum_{i=0}^{Hp-1} \big(& x(k+i|k)^T Q x(k+i|k) \\ & + u(k+i|k)^T R u(k+i|k) \big) \end{aligned} \tag{2.13}$$

An optimization problem is called quadratic program (QP) if the constraint functions

are affine and the cost function is a convex quadratic function ([10], [13]).

In order to show that the just presented cost function (2.13) is quadratic with respect to the optimization variable $U(k)$, the LTI System (2.2) is considered as prediction model. Recalling the equation (2.3), the vector of the predicted states $X(k) = [x(k|k)\ x(k+1|k)\ ...\ x(k+H_p-1|k)]$ can be written as

$$X(k) = \mathcal{A}X(k) + \mathcal{B}U(k) \tag{2.14}$$

where

$$\mathcal{A} = \begin{bmatrix} A \\ A^2 \\ \vdots \\ A^{H_p} \end{bmatrix} \in \mathbb{R}^{n\cdot H_p \times n}, \quad \mathcal{B} = \begin{bmatrix} B & 0 & \cdots & 0 \\ AB & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \vdots \\ A^{H_p-1}B & \cdots & \cdots & B \end{bmatrix} \in \mathbb{R}^{n\cdot H_p \times H_p}$$

and by defining also the weighting matrices

$$\mathcal{Q} = \begin{bmatrix} Q & 0 & \cdots & 0 \\ 0 & \ddots & \ddots & \vdots \\ \vdots & \ddots & Q & \vdots \\ 0 & \cdots & 0 & P \end{bmatrix} \in \mathbb{R}^{n\cdot H_p \times n\cdot H_p}, \quad \mathcal{R} = \begin{bmatrix} R & 0 & \cdots & 0 \\ 0 & \ddots & \ddots & \vdots \\ \vdots & \ddots & R & \vdots \\ 0 & \cdots & 0 & R \end{bmatrix} \in \mathbb{R}^{m\cdot H_p \times m\cdot H_p}$$

the cost function (2.13) assumes the following compact matrix form

$$J(x(k|k), U(k)) = \ X(k)^T \mathcal{Q} X(k) + U(k)^T \mathcal{R} U(k) \tag{2.15}$$

After some mathematical manipulations and substituting the (2.14) in (2.15), such a **quadratic form of the cost function** is obtained

$$J(x(k|k), U(k)) = \ \frac{1}{2} U(k)^T \mathcal{H} U(k) + x(k)^T \mathcal{F} U(k) + \overline{\mathcal{J}} \tag{2.16}$$

where

- $\mathcal{H} = 2(\mathcal{B}^T \mathcal{Q} \mathcal{B} + \mathcal{R}) \succ 0$ is the Hessian of the quadratic form.
- $\mathcal{F} = 2\mathcal{A}^T \mathcal{Q} \mathcal{B}$ is the mixed term of the quadratic form.

- $\overline{\mathcal{J}} = x(k)^T \mathcal{A}^T \mathcal{Q} \mathcal{A} x(k)$ is the vertical offset of the quadratic form.

Regarding the affinity of the constraint function, the expression (2.6) of the input constraints and (2.11) of the state constraints can be combined in the following **single set of linear constraints**

$$LU(k) \leq W \tag{2.17}$$

where $L = [L_x L_U]$ and $W = [W_x W_U]$.

In conclusion, the MPC constrained optimization problem (2.5) can be expressed as a QP.

$$\begin{aligned} U^* = \ & \arg\min_U \ \left(\frac{1}{2} U(k)^T \mathcal{H} U(k) + x(k)^T \mathcal{F} U(k) + \overline{\mathcal{J}}\right) \\ & \text{subject to} \ \ LU(k) \leq W \end{aligned} \tag{2.18}$$

The just expressed formulation ensure the convexity of the QP problem, therefore the unique optimal solution can be efficiently computed by different numerical algorithms such as

- "active" set algorithms [4], [11], [22].
- "primal-dual" interior point algorithms [4], [22].

Moreover, new algorithms have been recently introduced improving the MPC online computation. Some of these method are

- the partial enumerator methodology [4], [19].
- the modified active set method [11].
- the approximate primal barrier method [28].

2.1.5 Receding Horizon Control

The solution of the finite horizon QP (2.18) results in an optimal control move $U^*(k) = [u^*(k|k) ... \ u^*(k+H_p-1|k)]$ which starts at current time $t = k$ and ends at $t = k+H_p-1$. The application of this sequence over the time interval $[t, t+H_p]$ gives rise to an open loop control strategy. However, it is well known that modeling errors, parameter uncertainties or disturbances may lead to poor control performances with an open loop technique.

To overcome such a drawback, a feedback control action can be obtained through the Receding Horizon (RH) principle. An infinite horizon sub-optimal controller is designed by repeatedly solving finite time optimal control problems in a receding horizon fashion as described next.

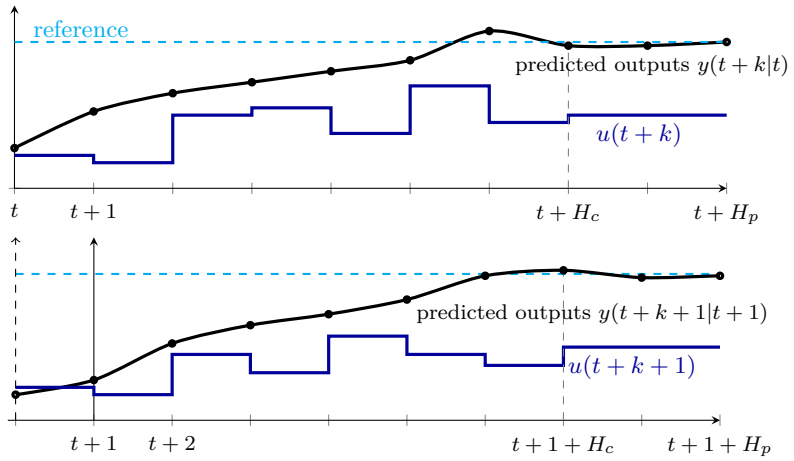

Figure 2.2: Receding Horizon Principle

Starting at current time $t = k$, the following open-loop optimal control problem is solved over a finite horizon (top graph in Figure 2.2)

$$\begin{aligned} &\min_{U_{t\to t+H_c|t}} \; J_t(x(t), U_{t\to t+H_c|t}) \\ &\text{subject to} \;\; LU_{t\to t+H_c|t} \leq W \end{aligned} \tag{2.19}$$

where $U_{t\to t+H_c|t} = [u_{t|t}, \cdots, u_{t+H_c-1|t}]$ is the reduced number of input sequence. The chosen $H_c \leq H_p$, referred as Control Horizon, is often used to reduce the number of variables involved in the optimization problem in order to tone down the computational effort. In case $H_c < H_p$, the remaining $H_p - H_c$ control input sequence $[u_{t+H_c|t}, \cdots, u_{t+H_p-1|t}]$, needed to evaluate the state prediction until the time $t = H_p$, can be chosen as

- $u(k+i|k) = 0$ with $H_c \leq i \leq H_p - 1$.
- $u(k+i|k) = u(k+H_c|k)$ with $H_c \leq i \leq H_p - 1$.

Let $U^*_{t\to t+H_c|t} = [u^*_{t|t}, \cdots, u^*_{t+H_c-1|t}]$ be the optimal solution of (2.19) at time t. Then, the Receding Horizon strategy can be explained by the following iterative procedure

I) only the first element $u^*_{t|t}$ is applied as control action to the system during the sampling interval $[t, t+1]$.

II) At the next time step $t+1$ a new optimal control problem (2.19) based on new measurements of the state $x(t+1)$ is solved over a shifted horizon (bottom graph in Figure 2.2).

The resulting Receding Horizon Controller (RHC) evaluates, at each sampling time, an optimal control input which depends only on the current state $x(k)$. The computed control input at time $t = k$ can be expressed as

$$u^*(k) = u^*(k|k) = u^*\big(x(k|k)\big) = u^*\big(x(k)\big) \tag{2.20}$$

Moreover, since in the considered optimization problem (2.19) either the system either the constraints and the cost function are time invariant, also the solution (2.20) is a time invariant function of the state. That is, the RHC implicitly defines a **non-linear time invariant static state feedback control law** of the form $u(k) = \mathcal{K}\big(x(k)\big)$.

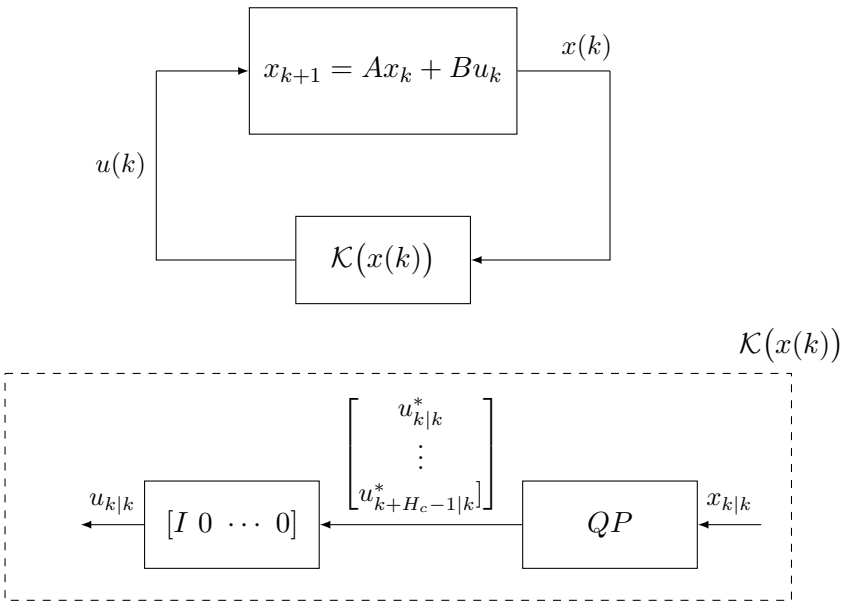

Figure 2.3: General Scheme of RH controller as a solution of the QP

In conclusion, a receding horizon controller where the finite time optimal control law is computed by solving a QP problem on-line is usually referred as Model Predictive Control (MPC). Figure 2.3 resumes the MPC strategy.

2.2 Adaptive MPC Principles

In the previous section 2.1 the general set up of the MPC strategy has been discussed. In particular, it has been highlighted how the optimal control law is strongly related to the model parameters as well as to the imposed constraints.

Adaptive Control covers a set of techniques based on a **real time adjustment of the controller parameters**, in order to maintain a desired level of control system performance despite the plant dynamic model parameters are changing in time.
At each sampling time, a suitable on-line estimator updates the model parameters on the basis of the collected data and the MPC control law is consequently adjusted in real time yielding a closed loop tuning procedure [17].

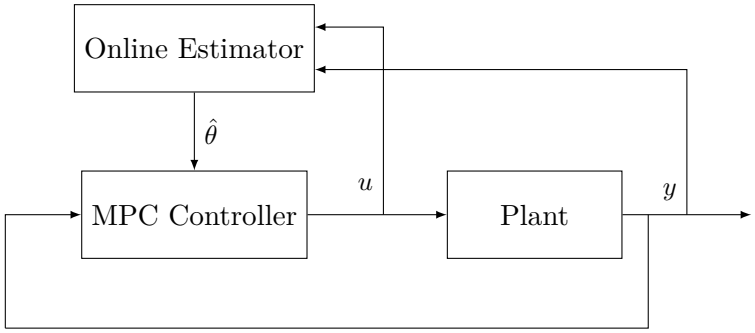

Figure 2.4: General Scheme of an Adaptive MPC Control System

The Adaptive MPC idea, as generally displayed in the Figure 2.4 above, is based on the following procedure

I) Collect the Plant input $u(k)$ and output $y(k)$ measurements.

II) Estimate the plant dynamic model parameters $\hat{\theta}(k)$ in real time using a suitable estimator algorithm.

III) Adjust the MPC optimal control law by updating, with the real time estimate $\hat{\theta}(k)$, the state space representation of the plant prediction model.

The effectiveness of the just presented Adaptive MPC control strategy occurs in a particular way when dealing with non-linear systems or with systems whose behaviour is strongly influenced by the working point.
Recursive least square (RLS) is a common estimator to be employed for estimating

the system parameters in real time. The RLS is fundamentally based on solving a Least Square estimation problem in a recursive fashion. For this reason, in the next paragraphs the main conceptual aspects of the Least Square criterion will be highlighted and the RLS estimation algorithms will be presented.

2.2.1 Least Square Method

The Least Square method has its roots in the linear regression problem i.e.the problem of finding the values of n real parameters $\theta_1, ..., \theta_n$ such that the sum of the squares of the differences between the output $y(k)$ of the plant and the output $\hat{y}(k)$ of the prediction model is minimized [27].

Considering for the Plant model the equation error or ARX model structure, the input-output relationship can be described by the following difference equation

$$\begin{aligned} y(k) + a_1 y(k-1) + ... + a_{n_a} y(k-n_a) =& b_0 u(k+n_a-n_b) + b_1 u(k+n_a-n_b-1) + \\ & ... + b_{n_b} u(k-n_b) + e(k) \end{aligned}$$

where

- $y(k)...y(k-n_a)$ are the collected output measurements for $k = 1, 2\ ...N$.
- $u(k+n_a-n_b)...u(k-n_b)$ are the collected input measurements for $k = 1, 2\ ...N$.
- $e(k)$ is a white noise term entering the process for $k = 1, 2\ ...N$.

In discrete-time domain

$$Y(z) = \frac{B(z)}{A(z)} + H(z)E(z) \tag{2.21}$$

where

$$\begin{aligned} B(z) &= b_0 + b_1 z^{-1} + b_2 z^{-2} + ... + b_{n_b} z^{-n_b} \\ A(z) &= 1 + a_1 z^{-1} + a_2 z^{-2} + ... + a_{n_a} z^{-n_a} \\ H(z) &= \frac{1}{A(z)} \end{aligned}$$

In order to evaluate the prediction error norm $\|y(k) - \hat{y}(k)\|_2$ the ARX model expression (2.21) need to be written in prediction form

$$\hat{Y}(z) = \left(1 - \frac{1}{H(z)}\right) Y(z) + \frac{G(z)}{H(z)} U(z) = \left(1 - A(z)\right) Y(z) + B(z) U(z) \tag{2.22}$$

which leads to the following difference equation for the predicted output

$$\hat{y}(k) = -a_1 y(k-1) - ... - a_{n_a} y(k-n_a) + b_0 u(k+n_a-n_b) + ... + b_{n_b} u(k-n_b) = \varphi^T(k)\theta \tag{2.23}$$

where

- $\varphi^T(k) = [-y(k-1)\ ...\ -y(k-n_a)\ u(k+n_a-n_b)\ ...\ u(k-n_b)]$ is the **regression vector**.
- $\theta = [a_1\ a_2\ ...\ a_{n_a}\ b_0\ b_1\ ...\ b_{n_b}]$ are the **model parameters to be identified**.

Finally, the quadratic optimality criterion related to the Least Square problem can be written as

$$J_N(\theta) = \frac{1}{N}\sum_{k=1}^{N}\left(y(k) - \varphi^T(k)\theta\right)^2 \tag{2.24}$$

and the unique optimal parameter vector $\hat{\theta}_{LS}$ is given by

$$\hat{\theta}_{LS} = \arg\min_{\theta\in\mathbb{R}^{n_a+n_b}}\left(J_N(\theta)\right) = \sum_{k=1}^{N}\left(\varphi(k)\varphi^T(k)\right)^{-1}\cdot\sum_{k=1}^{N}\varphi(k)y(k) \tag{2.25}$$

2.2.2 Recursive Least Square

The Recursive Least Square estimation method is based on minimizing the one-step prediction error in a recursive fashion on the basis of the previously acquired input-output data [25].

The least square estimate solution (2.25) at a generic time instant t is considered

$$\hat{\theta}_t = \underbrace{\sum_{k=1}^{t}\left(\varphi(k)\varphi^T(k)\right)^{-1}}_{S(t)^{-1}}\cdot\sum_{k=1}^{t}\varphi(k)y(k) = S(t)^{-1}\sum_{k=1}^{t}\varphi(k)y(k)$$

where

$$S(t) = \sum_{k=1}^{t}\varphi(k)\varphi^T(k) = S(t-1) + \varphi(t)\varphi^T(t)$$

After some manipulations the general structure of the RLS algorithm can be written as

$$\hat{\theta}_t \;=\; \hat{\theta}_{t-1} + \underbrace{S(t)^{-1}\varphi(t)}_{K(t)}\underbrace{\left(y(t) - \varphi^T(t)\hat{\theta}_{t-1}\right)}_{\epsilon(t)} \;=\; \hat{\theta}_{t-1} + K(t)\epsilon(t) \tag{2.26}$$

where

- $\hat{\theta}_{t-1}$ is the old parameter vector.
- $K(t)$ is a gain term related to the new parameters sensitivity with respect to the old parameters estimate.
- $\epsilon(t)$ is the prediction error.

As highlighted by the above equation (2.26), the RLS procedure is mainly performed by updating the old parameter vector estimate $\hat{\theta}_{t-1}$ with a correction term related to the actual prediction error $\epsilon(t)$. In this sense, the gain term $K(t)$ can be seen as a weighting function which decides how much the new parameters are influenced either by the previous parameters estimate or by the new measurements.

Different forms of the gain term $K(t)$ are associated with the following estimation algorithms

- Unnormalized and Normalized Gradient.
- Forgetting Factor.
- Kalman Filter.

Among the just exposed procedures the most commonly used is the **Normalized Gradient Algorithm**. This method is capable of guaranteeing good performances in terms of parameters estimation preserving the system stability thanks to the normalization term.

More precisely, the gain term $K(t)$ is evaluated through the ratio between an adaptation factor γ and the norm of the regression vector $\varphi(t)$, as expressed by the following equation

$$K(t) = \frac{\gamma}{\|\varphi(t)\|_2 + \varepsilon}\varphi(t) \tag{2.27}$$

where

- $0 \leq \gamma \leq 1$ is the adaptation factor related to the parameters sensitivity with respect to the variations of the plant dynamics.
- $\|\varphi(t)\|_2$ is the normalization term that maintains the system stability.
- ε is a small bias term that prevents sudden jumps in the gain term value when the normalization term is almost null.

In conclusion, collecting the expressions (2.26) and (2.27), the RLS iterative procedure can be resumed as

I) Collect the Plant input $u(k)$ and output $y(k)$ measurements.

II) Build the regression vector $\varphi(t)$.

III) Evaluate the gain term $K(t)$ according to (2.27).

IV) Evaluate the prediction error $\epsilon(t)$.

V) Update the old parameter vector $\hat{\theta}_{t-1}$ according to (2.26).

2.3 Adaptive MPC for K2 Actuator

In the previous sections 2.1 and 2.2 the main conceptual aspects of the Adaptive Model Predictive control technique have been discussed. In the following, the practical application to the K2 Actuator position control will be exploited.

The goal of the project is to control the proportional pressure valve (PPV) that actuates the K2 Clutch for the even gears. As long as the K2 Actuator is concerned, the output variables are the pressure valve p and the clutch position X_{csc} whereas the only control input variable is the actuation current I_{cmd} that should be provided by the controller in such a way that the position tracking error is minimized.

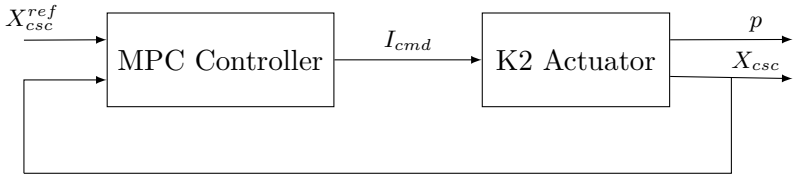

Figure 2.5: General Overview of the MPC position control for the K2 Actuator

Assuming that an ideal position measurement is available, a model predictive control architecture has been developed (general scheme in Figure 2.5).

MPC is a particularly suitable control strategy to handle such a problem because it is capable to manage different objectives even taking into account the physical limits of the K2 Actuator, including its saturation constraint in the control computation.

2.3.1 K2 Actuator linear model Identification

As discussed in section 2.1, the chosen MPC strategy needs a mathematical model of the plant to control, so the K2 Actuator dynamics has been identified on the basis of the measured data provided by Centro Ricerche Fiat (CRF).

The relationship between the input current I_{cmd} and the output position X_{csc} is represented by a first order model described by the following LTI discrete time transfer function

$$G(z) = \frac{\beta}{z+\alpha} = \frac{Y(z)}{U(z)} \tag{2.28}$$

where $Y = X_{csc}$ and $U = I_{cmd}$.

The identification of the model parameters α and β has been performed through MATLAB System Identification Toolbox. Different strategies have been adopted as described next.

Single linear model

The K2 Actuator dynamics has been preliminarily considered as a single linear model of the form (2.28).
A single data set, composed by eighteen different ascending position steps, has been used both for identifying the K2 Actuator and for validating the identified model.

From now on, the presented Figures, as well as the physical variables numerical data, will be normalized with respect to their maximum value.

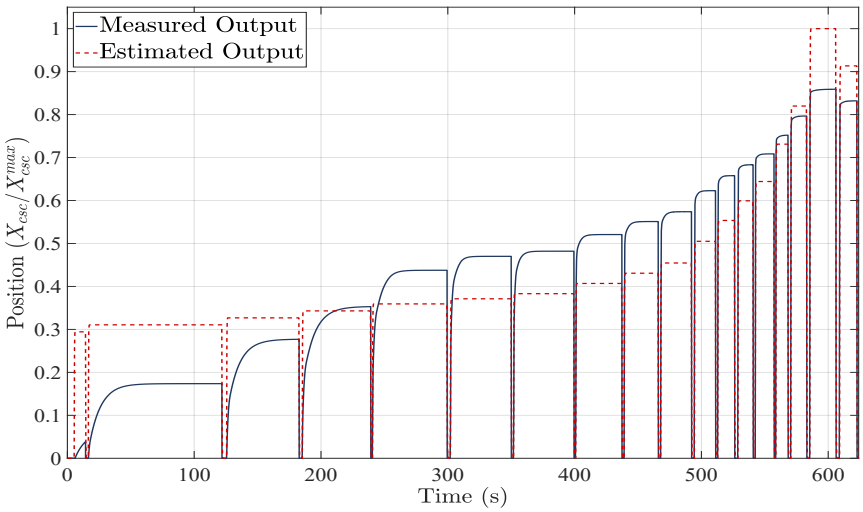

Figure 2.6: Validation data set for single single linear model identification strategy.

Such an identification strategy shows the following results for the model parameters

$$\alpha = -0.9828$$
$$\beta = 1.8208 \cdot 10^{-4}$$

As highlighted by Figure 2.6 above, it is quite evident that the estimated output is not able to track the measured output data for the whole working conditions. This is

because the Plant position responses are so variable in their static and dynamic properties, with respect to the different working regions, that a single identified model is not enough to catch these variations.

For this reason, a local multiple identification has been exploited as described next.

Multiple linear model

On the basis of the K2 Actuator static and dynamic properties, the whole provided data set, composed of eighteen ascending set points, has been previously divided in 4 different position ranges as highlighted by Figure 2.7 below.

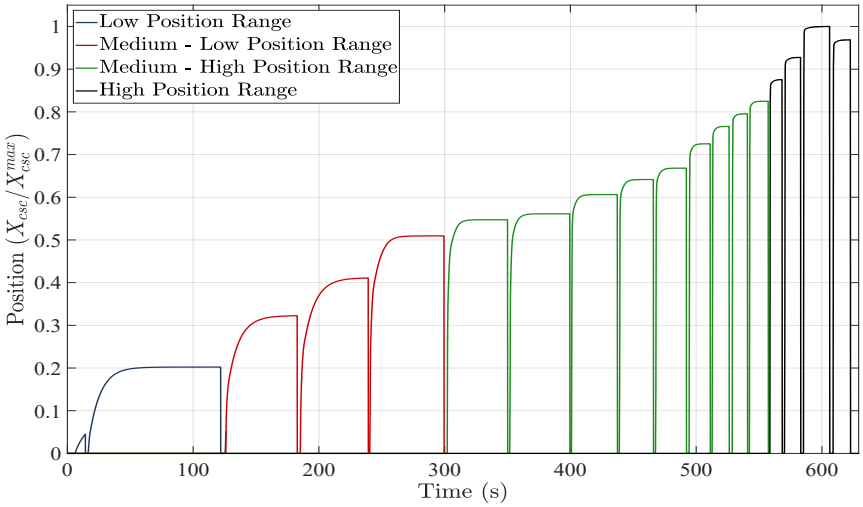

Figure 2.7: Different position ranges highlighted on the given data set.

- Low Position Range: [8 - 21] %
- Medium - Low Position Range: [21 - 57] %
- Medium - High Position Range: [57 - 85] %
- High Position Range: [85 - 100] %

For each of the chosen position region, a single linear model of the form (2.28) has been identified. More precisely, the local identification has been performed considering

as output data the corresponding position responses when the K2 Actuator Plant receives as input a suitable input current.

The static and dynamic properties in the different position ranges are resumed in the following table.

Range (%)	**DC Gain** $\left(\frac{\mathbf{mm}}{\mathbf{A}}\right)$	$\boldsymbol{\beta}$	$\boldsymbol{\alpha}$
8 to 21	$0.73 \leq K_{DC} \leq 5$	$2.6962 \cdot 10^{-6}$	-0.9997
21 to 57	$6.2 \leq K_{DC} \leq 7.96$	$4.6775 \cdot 10^{-5}$	-0.9964
57 to 85	$8.29 \leq K_{DC} \leq 8.64$	$1.1522 \cdot 10^{-4}$	-0.9865
85 to 100	$7.23 \leq K_{DC} \leq 8.05$	$2.3719 \cdot 10^{-4}$	-0.9740

Table 2.1: System parameters in the different working regions.

As highlighted in Table 2.1, high position ranges are related to fast position responses whereas small position ranges correspond to slow system dynamics. That is, the model behaviour is highly variable with respect to the operating range.

For the above considerations, a recursive estimation strategy has been exploited in order to have an adaptive Model Predictive Controller designed on the basis of a real time varying state representation of the system.

Recursive Estimation

The parameters α and β, related to the first order model describing the K2 Actuator dynamics, are estimated in real time through a recursive Least Square method on the basis of the collected current I_{cmd} and position X_{csc} measurements.

Recalling the expression (2.21) for the ARX model structure, the K2 Actuator input output relationship can be written in the following form

$$G(z^{-1}) = \frac{\beta z^{-1}}{1 + \alpha z^{-1}} = \frac{Y(z^{-1})}{U(z^{-1})} \tag{2.29}$$

where $Y = X_{csc}$ and $U = I_{cmd}$.

In time domain

$$X_{csc}(t) = \ -\alpha X_{csc}(t-1) + \beta I_{cmd}(t-1) = \varphi^T(t)\ \theta$$

where

- $\varphi^T(t) = [-X_{csc}(t-1)\ I_{cmd}(t-1)]$ is the regression vector.
- $\theta = [\alpha\ \beta]$ are the model parameters to be identified.

As previously discussed in section 2.2.2, the parameters recursive estimation is based on the normalized gradient algorithm. The overall recursive equation can be written as

$$\hat{\theta}_t \ = \ \hat{\theta}_{t-1} + \frac{\gamma}{\|\varphi(t)\|_2 + \varepsilon}\varphi(t)\epsilon(t) \tag{2.30}$$

Tuning and implementation details are discussed in section 2.4.

2.3.2 Observer for non measurable disturbance

The identification results, discussed in section 2.3.1, have brought out some important aspects related to the K2 Actuator system dynamics.

Especially in the low position ranges, when the system response is much slower, a static non-linear effect takes places due to the dead-zone relating the input current and the position of the clutch. This non-linearity, together with other internal physical effect inducing a mismatch from the nominal linear model (2.28), can be taken into account with a non measurable disturbance seen by the controller as an extra state variable.
In the MPC control architecture this can be exploited by means of a state Observer able of providing the controller the disturbance estimate needed for the prediction yielding an implicit feed-forward action in the control architecture.

The main idea of the state Observer is to use the actual position X_{csc} and current I_{cmd} measurements in order to provide the state estimate $\hat{x} = [\hat{X}_{csc}\ \hat{d}]$. The observer general structure is resumed in Figure 2.8 below.

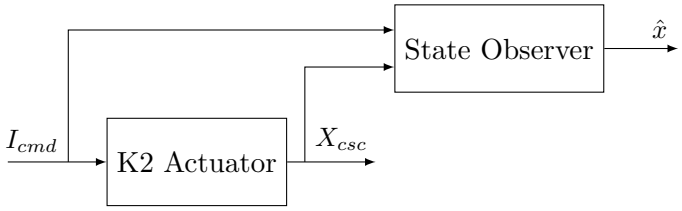

Figure 2.8: General Scheme of the State Observer.

The state observer dynamics can be associated to a MISO system according to the following state equations

$$\begin{aligned} \hat{x}(k+1) &= A_{obs}\hat{x}(k) + B_{obs}u(k) \\ \hat{y}(k) &= C_{obs}\hat{x}(k) \end{aligned} \tag{2.31}$$

where

- $u(k) = [X_{csc}(k)\ I_{cmd}(k)]$ is the is the Observer input vector.
- $B_{obs} = [B_{red}\ L]$, $A_{obs} = A_{red} - LC_{red}$ and $C_{obs} = [1\ 0]$ are the observer state matrices.

The matrix L, referred as the Observer Gain, has been suitably chosen in order to ensure the matrix A_{obs} asymptotic stability.

In particular, the **Luenberger Observer** has been adopted as state estimation strategy and a suitable tuning for the observer convergence speed has been exploited by means of a reasonable allocation of the gain matrix L eigenvalues.
According to the dominant dynamics of the model (2.28) associated to the K2 Actuator, the following eigenvalues have been assigned

$$\begin{aligned} \lambda_{obs_1} &= 0.5 \\ \lambda_{obs_2} &= 0.05 \end{aligned} \tag{2.32}$$

2.3.3 Augmented prediction model

As discussed in section 2.1.1, the Model Predictive Control architecture is related to the state-space representation of the system. In this case, the following LTI discrete time form is adopted

$$\begin{aligned} x(k+1) &= Ax(k) + Bu(k) \\ y(k) &= Cx(k) \end{aligned} \tag{2.33}$$

where $x = X_{csc}$, $u = I_{cmd}$, and $A = -\alpha$, $B = \beta$, $C = 1$.

Explicit integral Action

A standard approach to improve tracking performance for a constant reference and to reduce tracking errors even in presence of constant disturbances, is to use an explicit integral action in the control architecture.

For the considered MPC strategy, the explicit integral action can be taken into consideration by adding in the prediction model (2.33) the integral of the tracking error $q(k)$ as an extra state variable.
Therefore, the new state variable becomes $\tilde{x} = [X_{csc}\ X_{csc}^{ref}\ q]$ and the augmented prediction model can be described by the following state equations

$$\begin{aligned} x(k+1) &= Ax(k) + Bu(k) \\ X_{csc}^{ref}(k+1) &= X_{csc}^{ref}(k) \\ q(k+1) &= q(k) - T_s C x(k) + T_s X_{csc}^{ref}(k) \end{aligned} \tag{2.34}$$

where $T_s = 0.002\ s$.

In matrix form

$$\begin{aligned} \tilde{x}(k+1) &= \begin{bmatrix} A & 0 & 0 \\ 0 & 1 & 0 \\ -T_s C & T_s & 1 \end{bmatrix} \tilde{x}(k) + \begin{bmatrix} B \\ 0 \\ 0 \end{bmatrix} u(k) \\ y(k) &= \begin{bmatrix} C & 0 & 0 \end{bmatrix} \tilde{x}(k) \end{aligned} \tag{2.35}$$

Implicit integral Action

In the MPC architecture the control input variation Δu can be considered as optimization variable for the QP problem. This is another method to include an integral action in the control formulation and it can even give further improvements to the System response. In particular, by tightening the control input variations the transient oscillations can be reduced by means of a smooth control action.

In this context, implicit integral action is exploited by considering the previous control input value $u(k-1)$ as an extra state variable. Therefore, the new state variable becomes $\tilde{x} = [X_{csc}\ I_{cmd}(k-1)]$ and the augmented prediction model can be described by the following state equations

$$\begin{aligned} x(k+1) &= Ax(k) + Bu(k) \\ \Delta u(k) &= u(k) - u(k-1) \end{aligned} \tag{2.36}$$

In matrix form

$$\begin{aligned} \tilde{x}(k+1) &= \begin{bmatrix} A & B \\ 0 & 1 \end{bmatrix} \tilde{x}(k) + \begin{bmatrix} B \\ 1 \end{bmatrix} \Delta u(k) \\ y(k) &= \begin{bmatrix} C & 0 \end{bmatrix} \tilde{x}(k) \end{aligned} \tag{2.37}$$

In conclusion, taking into account the equations (2.34) and (2.36), and considering, for the sake of simplicity, a constant additive disturbance $d(k+1) = d(k)$, the overall augmented prediction model for the K2 Actuator is described by the following state equations

$$\begin{aligned} x(k+1) &= Ax(k) + Bu(k) + B_d d(k) \\ u(k) &= u(k-1) + \Delta u(k) \\ d(k+1) &= d(k) \\ X_{csc}^{ref}(k+1) &= X_{csc}^{ref}(k) \\ q(k+1) &= q(k) - T_s C x(k) + T_s X_{csc}^{ref}(k) \end{aligned} \tag{2.38}$$

where $x = X_{csc}$, $u = I_{cmd}$, and $A = -\alpha$, $B_d = B = \beta$, $C = 1$.

In matrix form

$$\begin{aligned} \begin{bmatrix} x(k+1) \\ u(k) \\ d(k+1) \\ X_{csc}^{ref}(k+1) \\ q(k+1) \end{bmatrix} &= \underbrace{\begin{bmatrix} A & B & B_d & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ -T_s C & 0 & 0 & T_s & 1 \end{bmatrix}}_{\tilde{A}} \underbrace{\begin{bmatrix} x(k) \\ u(k-1) \\ d(k) \\ X_{csc}^{ref}(k) \\ q(k) \end{bmatrix}}_{\tilde{x}} + \underbrace{\begin{bmatrix} B \\ 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}}_{\tilde{B}} \Delta u(k) \\ y(k) &= \underbrace{\begin{bmatrix} C & 0 & 0 & 0 & 0 \end{bmatrix}}_{\tilde{C}} \tilde{x}(k) \end{aligned} \tag{2.39}$$

where

- $\tilde{x}(k)$ is the augmented state variable.
- $\tilde{A}$, $\tilde{B}$ and $\tilde{C}$ are the augmented state matrices.

The general control architecture of the adaptive MPC position control for the K2 Actuator is resumed in Figure 2.9 below.

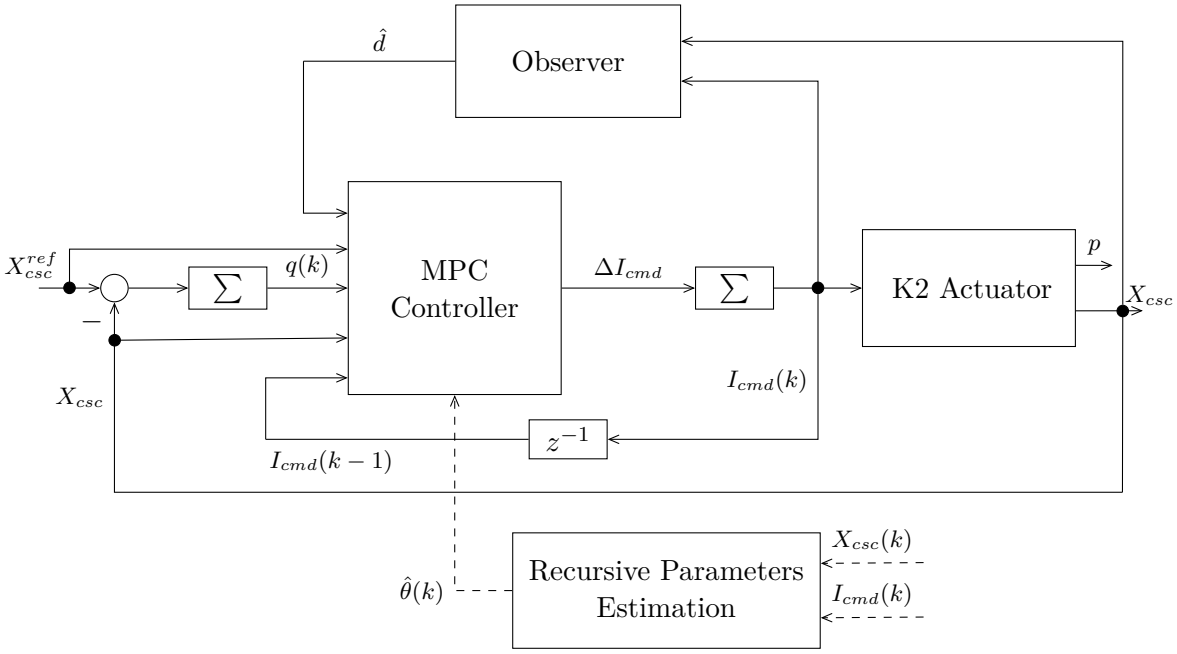

Figure 2.9: Adaptive MPC position control architecture for K2 Actuator.

2.3.4 Cost Function and Optimization Problem for K2 Actuator

As already mentioned in section 2.1.5, the MPC optimal control law is computed by solving the QP problem (2.18) on-line and by applying the Receding Horizon Principle. For this reason, the choice of a suitable cost function is crucial for the MPC set up.

In particular, the quadratic form (2.13) is selected and a proper weights and constraints design has been performed in order to guarantee the position reference X_{csc}^{ref} tracking without any steady state offset and the input current I_{cmd} oscillations attenuation in the K2 Actuator.
Therefore, considering the K2 Actuator augmented prediction model (2.39), the just mentioned control objectives can be fulfilled by expressing the cost function in the following tracking form

$$J(\tilde{x}(k|k), \Delta I_{cmd}(k)) = \sum_{i=0}^{Hp-1} Q_y(X_{csc}(k+i|k) - X_{csc}^{ref}(k+i|k))^2 + R\Delta I_{cmd}^2(k+i|k) + \\ + Q_q q^2(k+i|k) \tag{2.40}$$

where

- Q_y is the scalar weight for the position tracking.
- R is the scalar weight for the command input variation ΔI_{cmd}.
- Q_q is the scalar weight for the integral state $q(k)$.

Such a tracking problem has been treated as a regulation problem through the cost function below

$$J_{K2}(\tilde{x}(k|k), \Delta I_{cmd}(k)) = \sum_{i=0}^{Hp-1} \left(\tilde{x}(k+i|k)^T \, \tilde{Q} \, \tilde{x}(k+i|k) \; + \; R\Delta I_{cmd}^2(k+i|k)\right) \tag{2.41}$$

where

$$\tilde{Q} = \begin{bmatrix} Q_y & 0 & 0 & -Q_y & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ -Q_y & 0 & 0 & Q_y & 0 \\ 0 & 0 & 0 & 0 & Q_q \end{bmatrix}$$

Note that, in order to regulate the tracking error to zero and to add an explicit integral action in the control formulation, the weights Qy and Qq are introduced in the proper positions with respect to the augmented state vector $\tilde{x}(k) = [X_{csc}(k) \; I_{cmd}(k-1) \; d(k) \; X_{csc}^{ref}(k) \; q(k)]$.
The just mentioned weighting factors Q_y, Q_q and R are MPC project parameters that

have to be suitably chosen in order to perform the desired control action. Tuning and implementation details are discussed in section 2.4.

Once defined the proper cost function, the complete MPC optimization problem formulation can be obtained by considering the control input and state constraints. In this context, only the input constraint due to the K2 Actuator physical limitation has to be included.
More precisely, the actuation current must remain inside the following range

$$0 \leq I_{cmd} \leq 1000 \quad \text{mA} \tag{2.42}$$

Besides, the control input variation rate ΔI_{cmd} has been bounded as well in order to avoid sudden jumps in the control action.

$$-800 \leq \Delta I_{cmd} \leq 800 \quad \frac{\text{mA}}{T_s} \tag{2.43}$$

In conclusion, considering the just expressed linear inequalities, the augmented prediction model (2.39) and the cost function (2.41), the final expression of the QP problem for the K2 Actuator is given by

$$\begin{aligned}
&\Delta I^*_{cmd} = \ \arg\min_{\Delta I_{cmd}} \ J_{K2}(\tilde{x}(k|k), \Delta I_{cmd}(k)), \Delta I_{cmd}) \\
&\text{subject to} \\
&\tilde{x}(k+1) = \ \tilde{A}\tilde{x}(k) + \tilde{B}u(k) \\
&-I_{cmd}(k+i|k) \leq 0, \quad i = 1...H_p - 1 \\
&I_{cmd}(k+i|k) \leq 1000, \quad i = 1...H_p - 1 \\
&-\Delta I_{cmd}(k+i|k) \leq 800, \quad i = 1...H_p - 1 \\
&\Delta I_{cmd}(k+i|k) \leq 800, \quad i = 1...H_p - 1
\end{aligned} \tag{2.44}$$

2.4 MPC tuning and simulations results

The following section covers the MPC tuning procedure exploited to reach the best performance in the position control for the K2 Actuator.

Several simulation results using different position reference profiles will be presented in order to show the effectiveness of the proposed MPC control strategy. In particular, the following position profiles, provided by CRF, reasonably represent the K2 Actuator in different situations

- Slow Gear Change profile.
- Gear Change profile.
- Slow Ramp profile.
- Ramp profile.
- Stairs profile.

The tuning procedure of the Adaptive MPC controller is related to the choice of the prediction horizon H_p and the scalar weights Q_y, Q_q and R of the quadratic cost function (2.40) reminded below

$$J(\tilde{x}(k|k), \Delta I_{cmd}(k)) = \sum_{i=0}^{Hp-1} Q_y(X_{csc}(k+i|k) - X_{csc}^{ref}(k+i|k))^2 + R\Delta I_{cmd}^2(k+i|k) + \\ + Q_q q^2(k+i|k)$$

Moreover, since in the proposed control architecture a recursive estimator (section 2.2.2) provides a real time estimate of the model parameters, the adaptation gain γ can be considered as tuning parameters as well.

In a preliminary design procedure, the prediction horizon H_p is selected to be sufficiently long with respect to the K2 Actuator dominant dynamics, while the control horizon H_c is assumed to be equal to H_p. The scalar weights Q_y, Q_q and R are originally chosen in order to equally weight each term of the cost function. Afterwards, each coefficient has been properly tuned by means of a trial and error procedure until satisfactory control performances have been achieved.
Such a trial and error procedure involves extensive simulation tests developed through the Multi-Parametric Toolbox 3.0 (MPT) [15]. The just named Toolbox guarantees a

wide versatility in the MPC optimization problem formulation as well as a fully customizable way of handling input and state constraints.

The following design parameters values have been chosen to guarantee a satisfactory trade-off between the performance requirements

$$\begin{aligned} R &= 1 \\ Q_y &= 1.3 \cdot 10^3 \\ Q_q &= 1 \\ H_p &= 31 \\ \gamma &= 0.005 \\ \theta(0) &= [-0.9865 \quad 1.1522 \cdot 10^{-4}] \end{aligned} \tag{2.45}$$

where $\theta(0) = [\alpha_0 \ \beta_0]$ is the initial model parameters vector assumed to be inside the medium - high position range identified in section 2.3.1.

The proper values of the just expressed design parameters have been achieved through a suitable tuning procedure as described next.

2.4.1 Tuning of the MPC Prediction Horizon

From a computational point of view short prediction horizons are desirable, since the number of decision variables of the optimization problem is reduced. However, choosing a long prediction horizon induce intrinsic robustness in the control system and is required to achieve the desired closed-loop performances and to maintain the system stability.

Figure 2.10 clarifies the above considerations by showing the influence of the prediction horizon variation in the position response when the other design parameters are kept constant as in (2.45).
In the reported plot the value of the prediction horizon H_p is changed from $H_p = 25$ to $H_p = 40$ in one between the different reference profiles. It is evident that a too small prediction horizon is not able to handle the K2 Actuator dominant dynamics leading to position oscillations in the system response. On the contrary, a long prediction horizon guarantees the system stability but also affects the response speed and the computational effort as a side effect. Therefore, a prediction horizon $H_p = 31$ is selected.

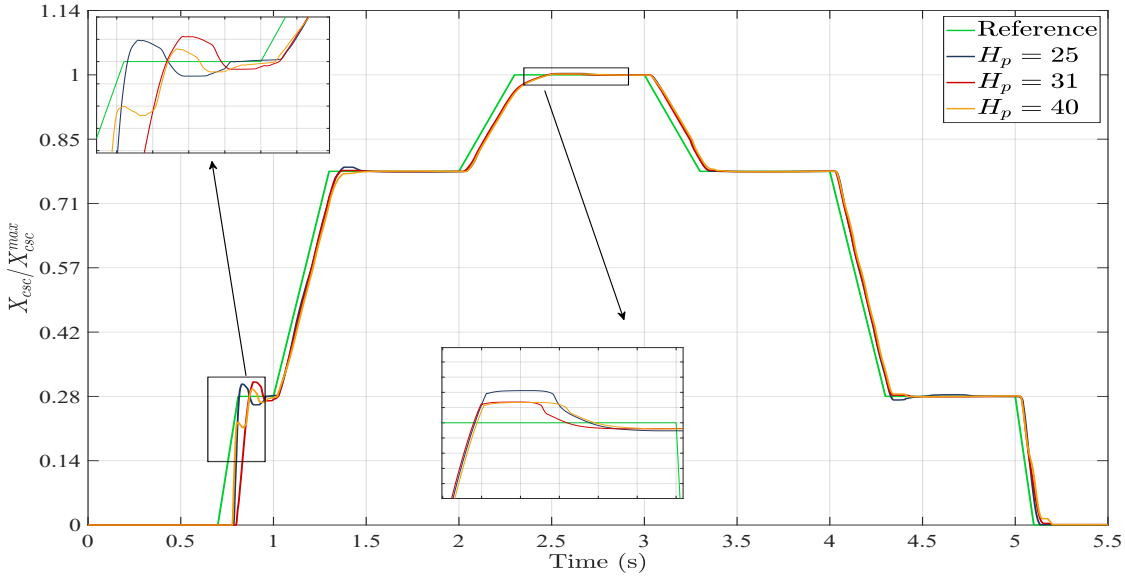

Figure 2.10: Position response with Gear shift reference profile adopting Adaptive MPC for different values of H_p.

2.4.2 Tuning of the MPC Cost Function Weights

The tuning procedure of the cost function weighting matrices is a crucial aspect in the MPC control design since it allows to find a suitable trade-off between performances and command activity.

Referring to the cost function (2.40), the scalar weights Q_y, Q_q and R have to be considered.

- The weight Q_y is related to the tracking error minimization.
- The weight Q_q gives penalty to the explicit integral action.
- The weight R is related to the command effort minimization.

Extensive simulations have been performed in which each of the three weights has been individually changed while the other two are kept fixed . These tests have brought out some important considerations about the MPC control law sensitiveness with respect to the single weight variation.

In practice, the MPC control action mainly depends on the ratio between the weights. More precisely, a scaling factor that affects all the weighting factor results in the same control action computed without scaling the weights.

For the above considerations, both the input weight R and the integral action weight Q_q are chosen to be equal as in (2.45). However, the tuning procedure for the tracking weight Q_y is reported in Figure 2.11 below.

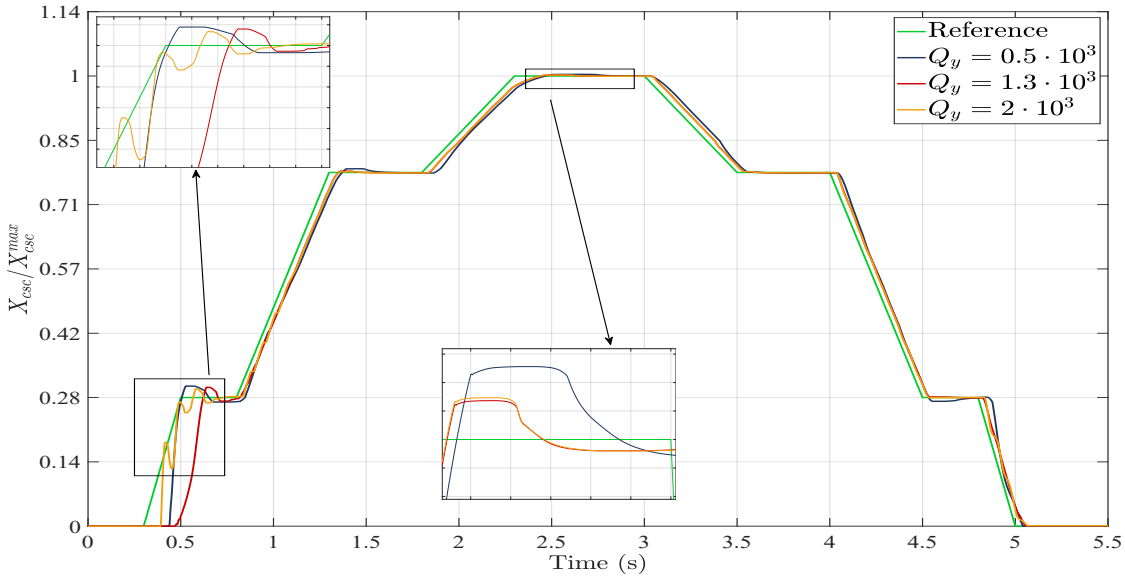

Figure 2.11: Position response with slow Gear shift reference profile adopting Adaptive MPC for different values of Q_y.

It is evident that small values of the output weight Q_y cause a slight worsening of the position tracking performances. On the other hand a too high value increases excessively the system reactivity inducing oscillations in the transient. The value $Q_y = 1.3 \cdot 10^3$ has been chosen as a reasonable trade off between the control objectives.

2.4.3 Tuning of the RLS adaptation gain

As discussed in section 2.2.2, for the chosen adaptive MPC control strategy the adaptation gain γ of the recursive estimator is an important parameter that has to be accounted in the design procedure.
The adaptation gain tuning allows to regulate the algorithm convergence speed but at the same time it affects the output behaviour.

The optimal suggested approach to estimate in real time the K2 Actuator model parameters is to use $\gamma = 1$ so that, whatever are the chosen initial conditions, the algorithm converges to the actual system dynamics as soon as possible.
However, several simulations results pointed out that the K2 Actuator system is really

sensitive with respect to the model parameters variation so a small adaptation gain $\gamma = 0.05,\ 0.005,\ 0.0005$ should be chosen.

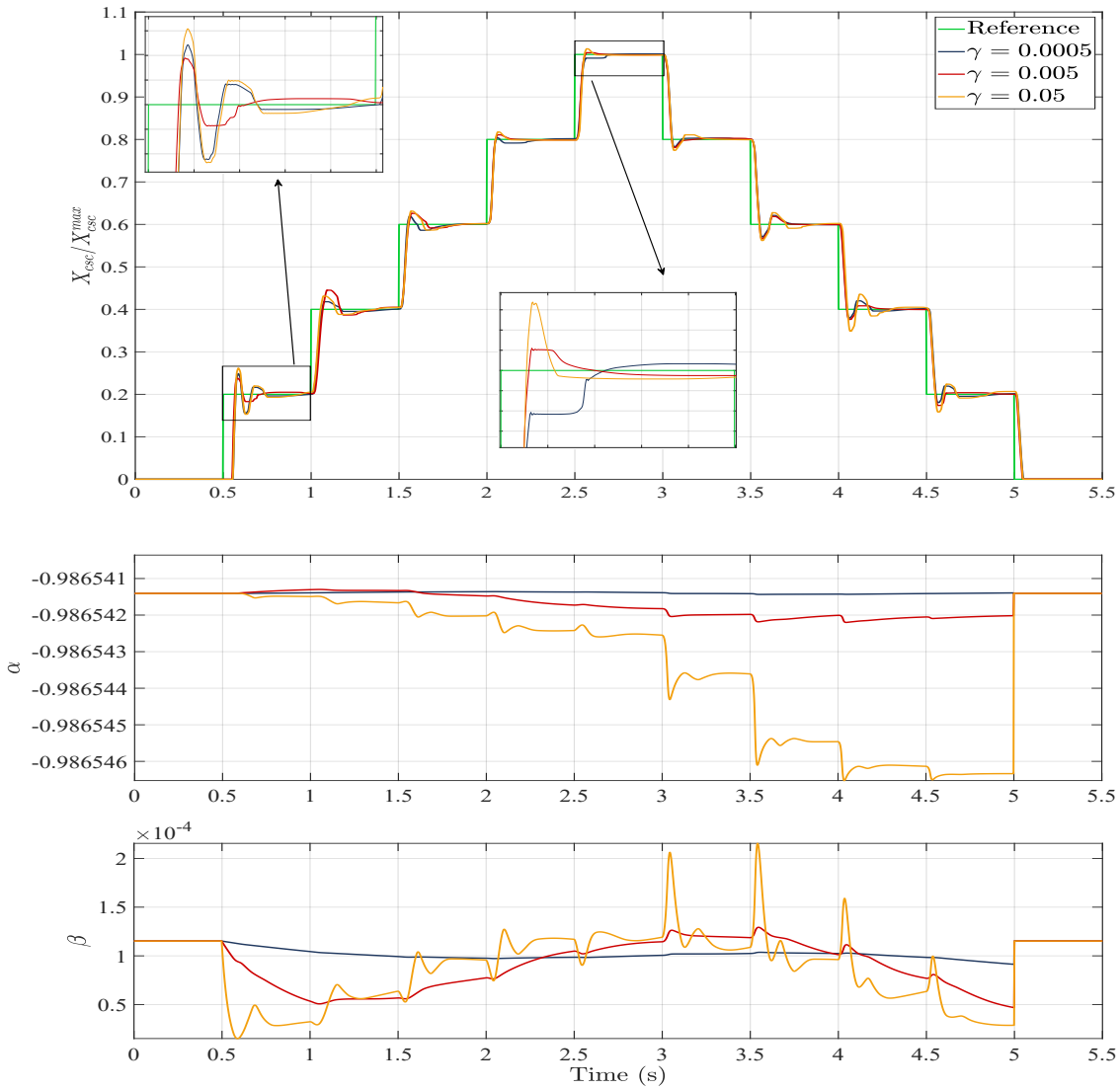

Figure 2.12: (Top) Position response and model parameters variation (Bottom) with Stairs reference profile adopting Adaptive MPC for different values of γ.

Figure 2.12 shows the effects of the adaptation gain γ variation on the position response and on the model parameters α and β. According to the above considerations, the value $\gamma = 0.005$ has been selected as adaptation gain of the recursive algorithm.

In conclusion, an overall review of the adaptive MPC tuning procedure can be made. The most important project parameter is the prediction horizon H_p. In fact, choosing a short prediction horizon leads to system instability whereas a too long horizon cause a significant performance worsening in terms of rise time. Instead, as long as the cost function weighting matrices are concerned, they have been changed in order to have a more precise and finer tuning. In this sense, the fact that an excessive increase of the tracking weight Q_y and of the integral action weight Q_q makes the system more aggressive, has been taken into account.

For the just discussed MPC controller (2.45) some transient and steady-state tracking indices have been evaluated in the presence of different position reference profiles. Table 2.2 below resumes the obtained results.

Range (%)	**Overshoot (%)**	**Rise Time (ms)**	**Steady State Error (mm)**
8 to 21	$\hat{s} \leq 17$	$t_r \leq 70$	$\cong 1 \cdot 10^{-2}$
21 to 57	$\hat{s} \leq 10$	$t_r \leq 60$	$\cong 1 \cdot 10^{-2}$
57 to 85	$\hat{s} \leq 4$	$t_r \leq 60$	$\cong 5 \cdot 10^{-3}$
85 to 100	$\hat{s} \leq 4$	$t_r \leq 80$	$\cong 2 \cdot 10^{-3}$

Table 2.2: Adaptive MPC controller performance resume with different reference profiles.

The chosen MPC parameters setting ensures quite satisfactory performance in terms of steady state tracking and rise time for all the identified position ranges. However, it is quite evident that the system response is characterized by a still too high overshoot, especially in low position ranges.

In order to improve the performance of the just described predictive controller, the design parameters have been dynamically tuned by means of a scheduling algorithm based on the actual value of the position reference and the tracking error.

2.4.4 Scheduling Algorithm

As discussed in section 2.3.1, the K2 Actuator dynamics is strongly variable with respect to the position range. In such a non-linear system it is a reasonable approach to design a specific controller on the basis of the working point in order to maintain a desired level of control system performance in all the tracking situations.

The Gain Scheduling control strategy consists in a real time adjustment of the controller design parameters on the basis of the measurements of some relevant system variables. In this context, the Gain scheduling approach is exploited by a suitable scheduling algorithm.

The position reference X_{csc}^{ref} and the tracking error $e_{csc} = X_{csc}^{ref} - X_{csc}$ have been considered as input measurements whereas the prediction horizon H_p, the output weight Q_y and the control input variation rate ΔI_{cmd} as the adjustable controller design parameters. In the following the operations computed by the scheduling algorithm are described (detailed MATLAB code in appendix)

1. MEASURE the reference $X_{csc}^{ref}(k)$ and the tracking error $e_{csc}(k)$ actual values.
2. CHECK the position range on the basis of $X_{csc}^{ref}(k)$.
3. IF$\big(e_{csc}(k) \leq \bar{e}_{csc}\big)$
   - 3.1 SET the prediction horizon H_p on the basis of the position range.
   - 3.2 IF$\big(e_{csc}(k) \leq e_{csc}^{min}\big)$

     INCREASE the output weight Q_y and DECREASE the control input variation rate ΔI_{cmd}.
   - 3.3 ELSE

     SET the output weight Q_y and the control input variation rate ΔI_{cmd} on the basis of the position range.
4. IF$\big(e_{csc}(k) > \bar{e}_{csc}\big)$
   - 4.1 SET the prediction horizon H_p on the basis of the position range.
   - 4.2 SET the output weight $Q_y = Q_y^{min}$.
   - 4.3 SET the control input variation rate $\Delta I_{cmd} = \Delta I_{cmd}^{max}$.

where $\bar{e}_{csc}$ is a suitable threshold adopted to handle sudden variations in the position references.

Referring to the above pseudo-code, the main aspects of the adopted scheduling algorithm can be clarified. The basic idea is to perform different tuning procedures on the controller design parameters with respect to the working region. More precisely, a larger prediction horizon H_p is needed when the system dynamics are slower, on the contrary, it can be decreased in the high position ranges where the system is characterized by

a faster dynamics. Furthermore, in the neighbourhood of tracking error minima e_{csc}^{min}, the output weight Q_y is increased and the constraint on ΔI_{cmd} is tightened in order to mitigate oscillations and guarantee good tracking performances without significantly decreasing the response speed. The effectiveness of the just presented control strategy has been tested by carrying out several simulations with different position reference profiles as illustrated in Figures 2.13, 2.14, 2.15, 2.16 and 2.17.

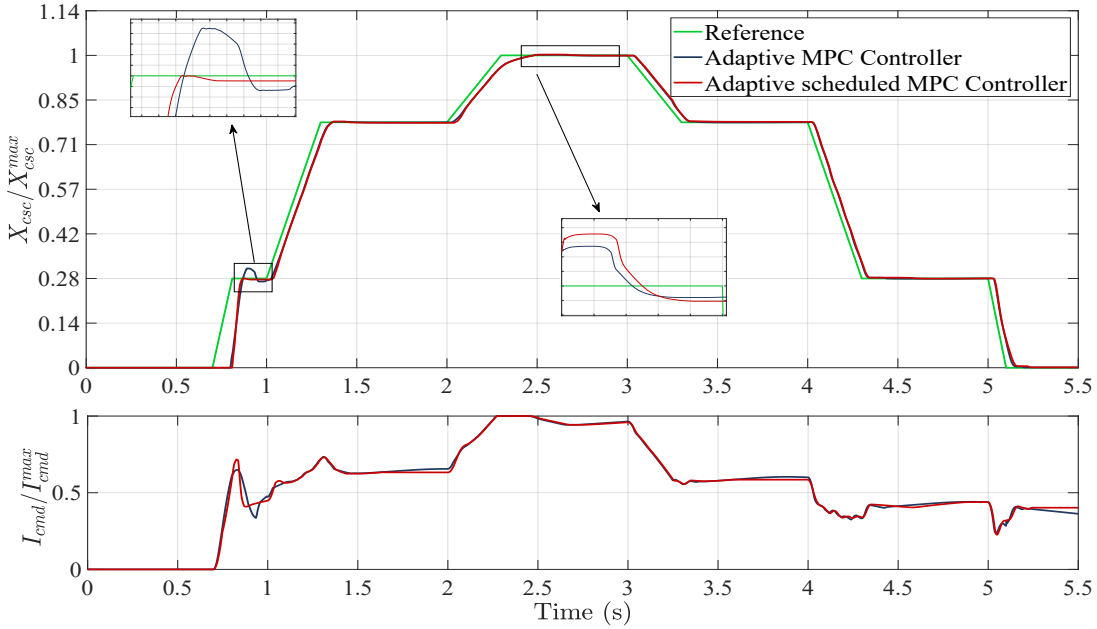

Figure 2.13: (Top) Position response and Input Current (Bottom) with Gear Change profile adopting Adaptive MPC and Adaptive scheduled MPC.

A final comparison between the Adaptive MPC and the Adaptive scheduled MPC control strategies can be made by evaluating both tracking and steady-state performances. Table 2.3 resumes the obtained results.

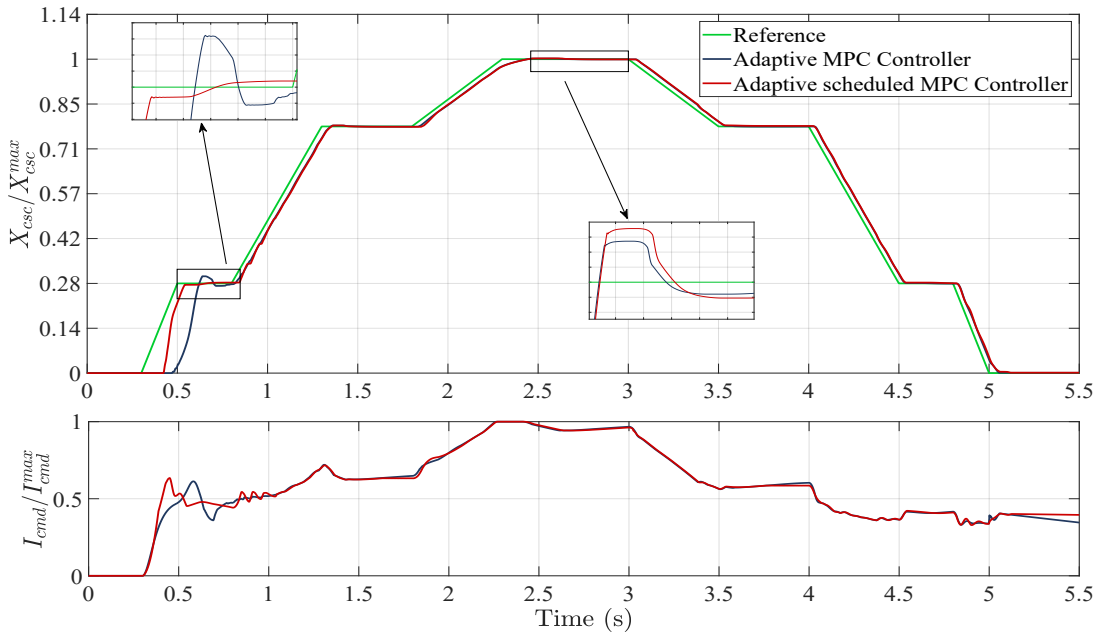

Figure 2.14: (Top) Position response and Input Current (Bottom) with slow Gear Change reference profile adopting Adaptive MPC and Adaptive scheduled MPC.

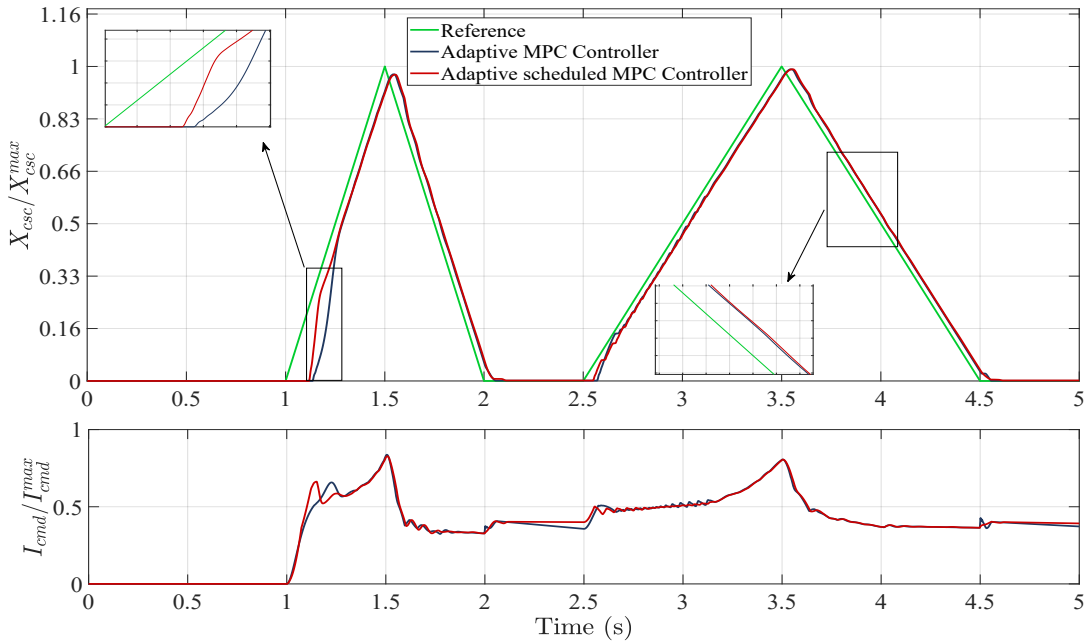

Figure 2.15: (Top) Position response and Input Current (Bottom) with ramp reference profile adopting Adaptive MPC and Adaptive scheduled MPC.

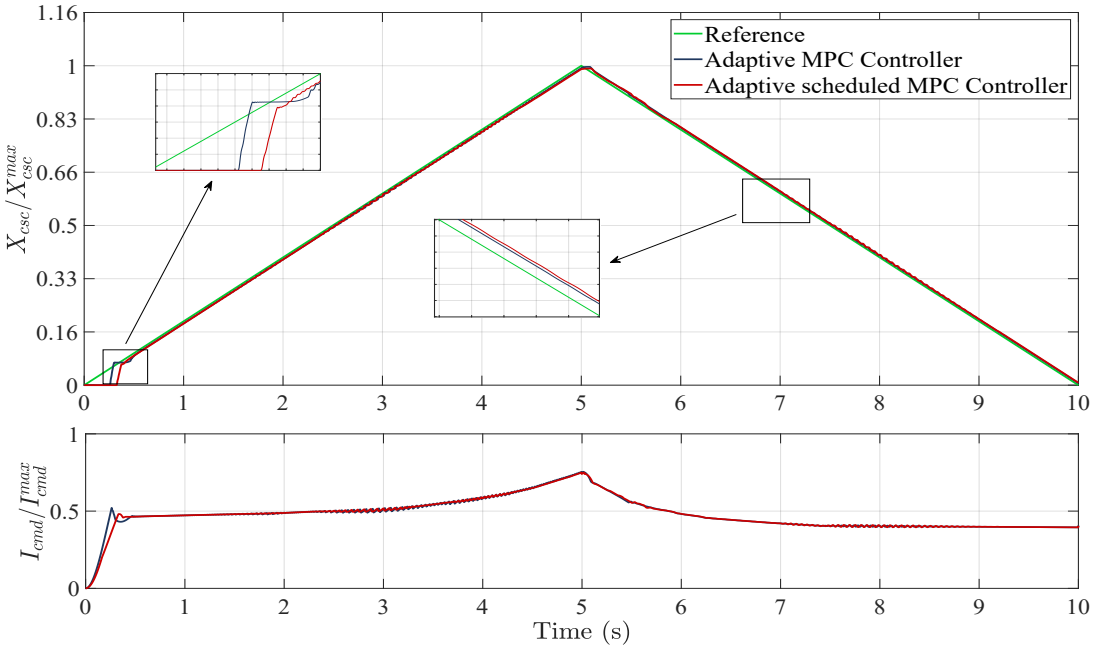

Figure 2.16: (Top) Position response and Input Current (Bottom) with slow ramp reference profile adopting Adaptive MPC and Adaptive scheduled MPC.

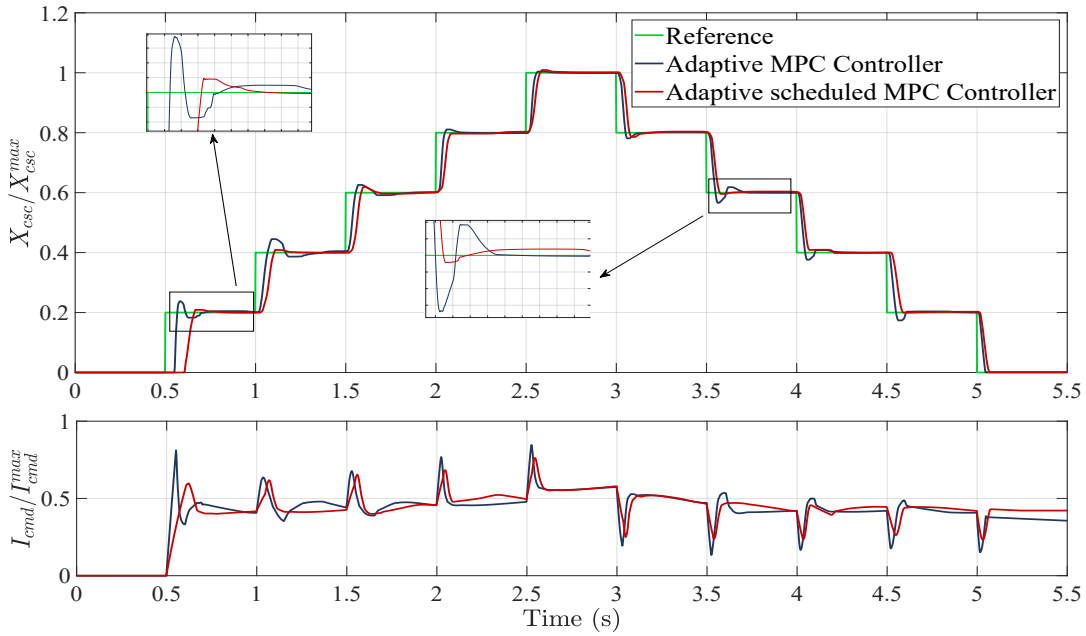

Figure 2.17: (Top) Position response and Input Current (Bottom) with stairs reference profile adopting Adaptive MPC and Adaptive scheduled MPC.

Range (%)	**Overshoot (%)**		**Rise Time (ms)**		**Steady State Error (mm)**	
	MPC 1^{st}	MPC 2^{nd}	MPC 1^{st}	MPC 2^{nd}	MPC 1^{st}	MPC 2^{nd}
8 to 21	$\hat{s} \leq 20$	$\hat{s} \leq 4$	$t_r \leq 70$	$t_r \leq 140$	$\cong 1 \cdot 10^{-2}$	$\cong 2 \cdot 10^{-3}$
21 to 57	$\hat{s} \leq 10$	$\hat{s} \leq 8$	$t_r \leq 60$	$t_r \leq 100$	$\cong 1 \cdot 10^{-2}$	$\cong 1 \cdot 10^{-2}$
57 to 85	$\hat{s} \leq 4$	$\hat{s} \leq 5$	$t_r \leq 60$	$t_r \leq 80$	$\cong 5 \cdot 10^{-3}$	$\cong 4 \cdot 10^{-3}$
85 to 100	$\hat{s} \leq 4$	$\hat{s} \leq 3$	$t_r \leq 80$	$t_r \leq 95$	$\cong 2 \cdot 10^{-3}$	$\cong 2 \cdot 10^{-3}$

Table 2.3: Adaptive MPC (MPC 1^{st}) and Adaptive scheduled MPC (MPC 2^{nd}) controller performance resume with different reference profiles.

Referring the above table, significant performance improvements, especially in low position ranges, are visible. The adopted Adaptive scheduled MPC control strategy is able to guarantee a smooth transient behaviour still maintaining the almost equal response speed. Besides, input current oscillations have been mitigated and a satisfactory steady state tracking has been assured thanks to the dynamic tuning of the controller design parameters.

As long as the computational effort is concerned, such design parameters (2.45) need quite high resources to compute in real time the optimal control action. This is mainly because a long prediction horizon is necessary to properly handle the K2 Actuator dominant dynamics.
For this reason, a LQR control strategy has been considered in order to enhance the computational aspects. This approach allows to evaluate the optimal control action in a static state feedback form as the solution of an unconstrained optimization problem. In the following Chapter the detailed implementation of the LQR control strategy is explained.

Chapter 3

Adaptive Linear Quadratic Control for K2 Actuator

This chapter will cover the main theoretical aspects of the Linear Quadratic Regulator (LQR) control pointing out some relevant differences and similarities with respect to the MPC technique discussed in Chapter 1. The LQR application to the K2 Actuator position control problem will be exploited together with several simulations results. Moreover, a final section is devoted to the numerical implementation of the proposed LQ control strategy.

3.1 LQR Overview

Linear optimal control is a particular branch of optimal control. The plant to control is assumed to be modelled by a LTI transfer function while the controller that generates the optimal control move is constrained to be linear. Moreover, in linear optimal control, the cost function adopted as performance index is quadratic either in the control action ether in the optimization variable. However, with this strategy, constraints on the involved variables are not accounted so an explicit saturation has to be exploited off-line.

Some advantages of the just presented linear optimal control can be outlined [2].

- In practice, all linear optimal control problems have readily computable solutions whereas, on the contrary, many generic optimal control problems require high computational effort to obtain a usable solution.
- Linear optimal control may be also applied to non-linear systems operating in a restricted working region.

- Linear optimal controller yield to possess some attractive properties such as good gain margin,good phase margins and good tolerance of non-linearities.
- Linear optimal control is related to the same framework of the control problems studied via classical methods.

As previously mentioned, the LQR control architecture is related to the mathematical description of the plant to control. For such a linear case, the following LTI discrete time model is considered

$$\begin{aligned} x(k+1) &= Ax(k) + Bu(k) \\ y(k) &= Cx(k) \end{aligned} \tag{3.1}$$

where $A \in \mathbb{R}^{n,n}$, $B \in \mathbb{R}^{n,m}$, $C \in \mathbb{R}^{n,p}$.

The associated performance index is the quadratic function defined over a finite horizon of H_p steps

$$\begin{aligned} J(x(k), U(k)) = x(k+H_p)^T P x(k+H_p) + \sum_{i=0}^{Hp-1} \big(& x(k+i)^T Q x(k+i) \\ & + u(k+i)^T R u(k+i) \big) \end{aligned} \tag{3.2}$$

where $Q \succeq 0$ and $P \succeq 0$ are respectively the state and terminal state penalties while $R \succ 0$ is the input penalty. The LQR optimal control action $U^*(k) = [u^*(k)\ u^*(k+1)\ ...\ u^*(k+H_p-1)]$ is obtained as the solution of the following optimization problem

$$\begin{aligned} U^* = & \arg\min_U \big(J(x(k), U(k)) \big) \\ \text{subject to } & x(k+1) = Ax(k) + Bu(k) \end{aligned} \tag{3.3}$$

Once described the general framework, two different LQR approaches are presented.

3.1.1 Finite Horizon LQR

The finite horizon LQR can be considered as a QP problem with the absence of constraints. Therefore, recalling section 2.1.4, the states vector $X(k) = [x(k)\ ...\ x(k+H_p-1)]$ are expressed in a compact matrix form

$$X(k) = \mathcal{A}X(k) + \mathcal{B}U(k)$$

where

$$\mathcal{A} = \begin{bmatrix} A \\ A^2 \\ \vdots \\ A^{H_p} \end{bmatrix} \in \mathbb{R}^{n \cdot H_p \times n}, \quad \mathcal{B} = \begin{bmatrix} B & 0 & \cdots & 0 \\ AB & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \vdots \\ A^{H_p-1}B & \cdots & \cdots & B \end{bmatrix} \in \mathbb{R}^{n \cdot H_p \times H_p}$$

and the cost function (3.2) can be written in a suitable quadratic form

$$J(x(k), U(k)) = \frac{1}{2} U(k)^T \mathcal{H} U(k) + x(k)^T \mathcal{F} U(k) + \overline{\mathcal{J}} \tag{3.4}$$

where

- $\mathcal{H} = 2(\mathcal{B}^T \mathcal{Q} \mathcal{B} + \mathcal{R}) \succ 0$ is the Hessian of the quadratic form.
- $\mathcal{F} = 2\mathcal{A}^T \mathcal{Q} \mathcal{B}$ is the mixed term of the quadratic form.
- $\overline{\mathcal{J}} = x(k)^T \mathcal{A}^T \mathcal{Q} \mathcal{A} x(k)$ is the vertical offset of the quadratic form.

with the following weighting matrices

$$\mathcal{Q} = \begin{bmatrix} Q & 0 & \cdots & 0 \\ 0 & \ddots & \ddots & \vdots \\ \vdots & \ddots & Q & \vdots \\ 0 & \cdots & 0 & P \end{bmatrix} \in \mathbb{R}^{n \cdot H_p \times n \cdot H_p}, \quad \mathcal{R} = \begin{bmatrix} R & 0 & \cdots & 0 \\ 0 & \ddots & \ddots & \vdots \\ \vdots & \ddots & R & \vdots \\ 0 & \cdots & 0 & R \end{bmatrix} \in \mathbb{R}^{m \cdot H_p \times m \cdot H_p}$$

Since the just presented cost function (3.4) is quadratic with respect to the optimization variable, the minimizer vector $U^*(k) = [u^*(k) ... \ u^*(k + H_p - 1)]$ can be found by computing its gradient and setting it to zero. This **Batch Approach** yields the following closed form for the optimal vector of control input

$$U^*(k) = -\left(\mathcal{B}^T \mathcal{Q} \mathcal{B} + \mathcal{R}\right)^{-1} \left(\mathcal{A}^T \mathcal{Q} \mathcal{B}\right)^T x(k) = -\mathcal{H}^{-1} \mathcal{F}^T x(k) \tag{3.5}$$

As this expression implies, the finite horizon LQR optimal control law is a linear function of the measured state $x(k)$. More precisely, the optimal control move at the generic time instant $t = k + i$ depends on the i^{th} step ahead predicted state $x(k + i)$ obtained starting from the measured state $x(k)$. Hence, as a matter of fact, it does not

depends on the actual state $x(k+i)$.

As a consequence, it can be said that the finite horizon LQR optimal control law (3.5) is provided in an open-loop fashion leading to a quite weak control action. This is because the model (3.1) for predicting the system states may be inaccurate and subject to not accounted disturbances [10], [2].

The LQR control strategy can be improved by considering the optimization problem (3.3) operating over a long time period as described next.

3.1.2 Infinite Horizon LQR

The LQR optimization problem (3.2) is carried out to infinity leading to the following cost function

$$J(x(k), U(k)) = \sum_{i=0}^{\infty} \big(x(k+i)^T Q x(k+i) + u(k+i)^T R u(k+i)\big) \tag{3.6}$$

In this case the Batch approach can not be applied and the optimal solution has to be found with a **Recursive Method** [10], [12].
Starting from the previously described finite horizon LQR, the Hamiltonian function can be defined as

$$H(k) = \frac{1}{2}\big(x(k)^T Q x(k) + u(k)^T R u(k)\big) \; + \; \lambda(k+1)^T \big(Ax(k) + Bu(k)\big) \tag{3.7}$$

where $\lambda(k)$ are the Lagrange multiplier vectors associated with the LQR performance index.

The optimality and stationary conditions can be written as

$$\begin{aligned} \frac{\partial H(k)}{\partial \lambda(k+1)} &= Ax(k) + Bu(k) \; = x(k+1) \\ \frac{\partial H(k)}{\partial x(k)} &= Qx(k) + A^T \lambda(k+1) \; = \lambda(k) \\ \frac{\partial H(k)}{\partial u(k)} &= Ru(k) + B^T \lambda(k+1) \; = 0 \end{aligned} \tag{3.8}$$

These lead to the following expression for the control input sequence

$$u(k) = -R^{-1}B^T\lambda(k+1) \tag{3.9}$$

As this expression implies, the optimal control input is determined only if the co-state sequence $\lambda(k+1)$ is known. Therefore, it is necessary to find a relationship between the co-states $\lambda(k)$ and the states $x(k)$ of the System.

The final state weighting function is $\Phi = \frac{1}{2}x(k+H_p)^T P_{H_p} x(k+H_p)$. Hence, the boundary condition implies that

$$\frac{\partial \Phi}{\partial x(k+H_p)} = P_{H_p}x(k+H_p) = \lambda(k+H_p)$$

Using the sweep method [12], the above linear relation is supposed to hold for all $k \leq H_p$ so that

$$\lambda(k) = P_k x(k) \tag{3.10}$$

Using (3.10) in (3.9) and then in (3.8) the dependencies from the co-states sequence can be eliminated. After some manipulations, the following time-variant expression for the optimal control sequence is obtained

$$u^*(k) = -(R + B^T P_{k+1} B)^{-1} \cdot B^T P_{k+1} A x(k), \quad \text{for } k = 0...H_p - 1 \tag{3.11}$$

where

$$P_k = A^T P_{k+1} A + Q - A^T P_{k+1} B \cdot (R + B^T P_{k+1} B)^{-1} \cdot B^T P_{k+1} A \tag{3.12}$$

referred as **Riccati Difference Equation (RDE)**, is a recursive equation initialized with $P_{H_p} = Q$ and solved backwards.
In case of the infinite horizon LQR strategy, the following conditions are sufficient for the RDE to converge [10], [12]

- the pair (A, B) is stabilizable.
- the pair $(Q^{1/2}, B)$ is observable.

Assuming that $P_{k\to\infty} = P$, the (3.12) becomes the **Discrete-time Algebraic Riccati Equation (DARE)**

$$P = A^T PA + Q - A^T PB \cdot (R + B^T PB)^{-1} \cdot B^T PA \tag{3.13}$$

As a consequence, the time-variant control law (3.11) assumes the following time-invariant form

$$u^*(k) = -\underbrace{(R + B^T PB)^{-1} \cdot B^T PA}_{K} \; x(k) = -Kx(k) \tag{3.14}$$

where the constant matrix K is referred as steady-state Kalman Gain [12].

The equation (3.14) is a **static state feedback** control law, which express the optimal control sequence $u^*(k)$ as a linear function of the measured state $x(k)$ at the current time $t = k$. Therefore, in the infinite horizon LQR, or asymptotic LQR, the optimal control move is applied in a closed loop manner resulting in a more robust control action.
Moreover, since the Kalman Gain K is expressed in terms of the DARE solution P and the system and weighting matrices, it can be computed off-line and stored in memory before the control action is ever applied to the system.

3.2 Adaptive LQ for K2 Actuator

The previous section 3.1.2 has highlighted the advantages, from the control action robustness point of view, of the infinite horizon LQR control law (3.14) with respect to the classical LQR approach.
The application of the just presented LQR design technique to the K2 Actuator position control problem will be exploited in the following section.

Assuming that the K2 Actuator position measurement is available, a LQR control architecture has been developed.

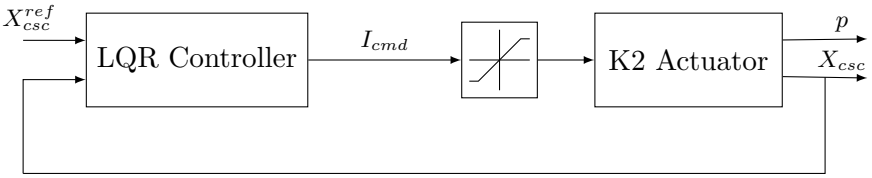

Figure 3.1: General Overview of the LQR position control for the K2 Actuator

As Figure 3.1 shows, an explicit saturation of the actuation current I_{cmd} has to be performed. This is to handle the K2 Actuator physical limitations that cannot be managed by the chosen LQR control strategy.

3.2.1 K2 Actuator Hammerstein model identification

As for the MPC, also in the LQR control architecture a mathematical model for the K2 Actuator is needed.
However, a preliminary design and simulation study showed that the disturbance compensation strategy (section 2.3.2) employed in MPC does not introduce significant performance improvements for LQ controller. As a consequence, plant non-linearity have been explicitly included in the model through an Hammerstein structure as described next.

The same four different position ranges presented in section 2.3.1 have been considered

- Low Position Range: [8 - 21] %
- Medium - Low Position Range: [21 - 57] %
- Medium - High Position Range: [57 - 85] %

- High Position Range: [85 - 100] %

For each position range, the relationship between the input current I_{cmd} and the output position X_{csc} has been represented by a static non-linear function $N(\cdot)$ followed by a discrete-time LTI System as shown in the below Figure 3.2.

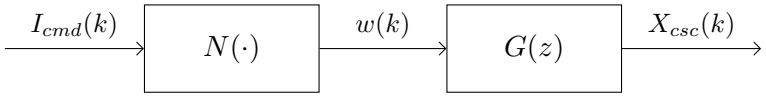

Figure 3.2: Hammerstein System block diagram representation for the K2 Actuator.

where the intermediate variable $w(k)$ is the output of the non-linear function such that $w(k) = N(I_{cmd}(k))$ has the same dimensions of $I_{cmd}(k)$.
More precisely, the static non linear part has been associated to a current dead zone D (see Figure 3.3), while the LTI part is assumed to be represented by the following first order discrete time transfer function

$$G(z) = \frac{\beta}{z + \alpha} = \frac{Y(z)}{U(z)} \tag{3.15}$$

where $Y = X_{csc}$ and $U = I_{cmd}$.

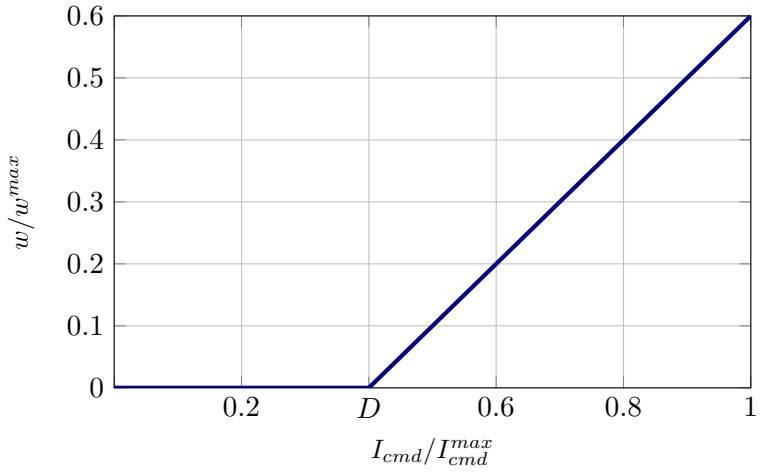

Figure 3.3: Static non-linear function $N(\cdot)$ represented as a dead zone of amplitude D.

MATLAB System Identification Toolbox has been employed to identify the model

(3.15) parameters α and β. Table 3.1 below resumes the model characteristics in the different position ranges

Range (%)	**Deadzone**	$\boldsymbol{\beta}$	$\boldsymbol{\alpha}$
8 to 21	$[0; 0.39]$	$9.1452 \cdot 10^{-6}$	-0.99998
21 to 57	$[0; 0.28]$	$5.8677 \cdot 10^{-5}$	-0.99996
57 to 85	$[0; 0.26]$	$7.0651 \cdot 10^{-5}$	-0.9969
85 to 100	$[0; 0.2]$	$4.7817 \cdot 10^{-4}$	-0.9837

Table 3.1: System parameters in the different working regions.

As confirmed by the above Table, the model dynamic properties as well as the dead zone amplitude are strongly variable with respect to the operating working region. For this reason, an adaptive LQR controller has been designed on the basis of a real time varying state representation of the system.
More precisely, similarly as for the MPC (section 2.3.1), the Normalized Gradient technique has been employed to estimate the model (3.15) parameters.

$$\hat{\theta}_t \;=\; \hat{\theta}_{t-1} + \frac{\gamma}{\|\varphi(t)\|_2 + \varepsilon}\varphi(t)\epsilon(t)$$

where

- $\varphi^T(t) = [-X_{csc}(t-1)\ I_{cmd}(t-1)]$ is the regression vector.
- $\theta = [\alpha\ \beta]$ are the model parameters to be identified.
- $\epsilon(t) = y(t) - \varphi^T(t)\hat{\theta}_{t-1}$ is the estimation error.

Tuning and implementation details are discussed in section 3.3.

3.2.2 Adaptive LQ control architecture

As already pointed out in section 3.1, the Linear Quadratic control architecture is based on the state space representation of the plant to control. In this context, the linear part (3.15) of the Hammerstein system, representing the K2 Actuator model, is represented by the following LTI discrete time form

$$\begin{aligned} x(k+1) &= Ax(k) + Bu(k) \\ y(k) &= Cx(k) \end{aligned} \tag{3.16}$$

where $x = X_{csc}$, $u = w = N(I_{cmd})$, and $A = -\alpha$, $B = \beta$, $C = 1$.

Integral Action

As for the MPC, in the LQ control architecture an explicit integral action is necessary to improve the steady state tracking performances in the presence of constant references. This can be accounted by considering the integral term $q(k)$ as an extra state variable.

The augmented model is described by the following state equations

$$\begin{aligned} x(k+1) &= Ax(k) + Bu(k) \\ q(k+1) &= q(k) - T_s C x(k) + T_s X_{csc}^{ref}(k) \end{aligned} \tag{3.17}$$

where $T_s = 0.002\ s$.

In matrix form

$$\begin{aligned} \tilde{x}(k+1) &= \begin{bmatrix} A & 0 \\ -T_s C & T_s \end{bmatrix} \tilde{x}(k) + \begin{bmatrix} B \\ 0 \end{bmatrix} u(k) \\ y(k) &= \begin{bmatrix} C & 0 \end{bmatrix} \tilde{x}(k) \end{aligned} \tag{3.18}$$

Dead zone compensation

Since the K2 Actuator has been represented by a Hammerstein system (section 3.2.1), a two stage approach is needed to provide the correct LQ control action. Due to its particular structure, the LQ controller is firstly designed taking only into account the linear model, and then the non-linearity is removed by inverting the non-linear static block $N(\cdot)$.

In this case, the static non-linear function $N(\cdot)$, represented by a dead zone D, can be described by he following piecewise linear relationship

$$N(I_{cmd}):\ w = \begin{cases} mI_{cmd} - q & I_{cmd} \geq D \\ 0 & I_{cmd} \leq D \end{cases} \tag{3.19}$$

where $m = 1$ and $q = D$ is the dead zone amplitude.

Therefore, the inversion of such a function has been performed by adopting the following piecewise linear expression

$$N^{-1}(w): \; I_{cmd} = \begin{cases} \frac{q}{m} + \frac{w}{m} & w \geq 0 \\ D & w \leq 0 \end{cases} \tag{3.20}$$

In practice, the dead zone compensation in the control architecture is performed, at each sampling time, by a static map that associates the command input $I_{cmd}(k)$ with the intermediate variable $w(k)$ produced by the controller. Figure 3.4 below graphically shows the just presented inverting function $N^{-1}(w)$.

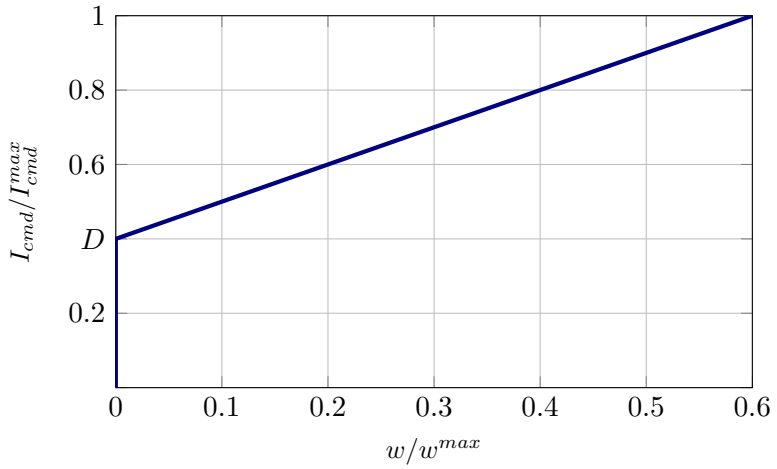

Figure 3.4: Static map $N^{-1}(\cdot)$ for dead zone compensation.

Cost function and LQ control Law

As discussed in section 3.1, the infinite horizon LQR control strategy is related to the minimization of a quadratic performance index. In this context, the K2 Actuator augmented model (3.18) is considered and the infinite horizon LQR unconstrained optimization problem can be written as

$$\begin{aligned} w^* = & \arg \min_{W^k,\, k\in[0,\,\infty)} \left(J(\tilde{x}(k), W^k)\right) \\ \text{subject to} \quad & \tilde{x}(k+1) = \tilde{A}\tilde{x}(k) + \tilde{B}u(k) \end{aligned} \tag{3.21}$$

with the associated quadratic cost function

$$J_{K2}(\tilde{x}(k), W^k) = \sum_{i=0}^{\infty} \left(\tilde{x}(k+i)^T \tilde{Q}\tilde{x}(k+i) + Rw^2(k+i)\right) \tag{3.22}$$

where

$$\tilde{Q} = \begin{bmatrix} Q_y & 0 \\ 0 & Q_q \end{bmatrix}$$

The scalar terms Q_y and Q_q are, respectively, weighting factors for position X_{csc} tracking and for the integral state $q(k)$. These have to be considered as LQR project parameters whose choice is related to the control action effectiveness. Tuning and implementation details are discussed in section 3.3.

Referring to section 3.1.2, such a tracking problem is exploited by the infinite horizon LQR control strategy through the following static state feedback control law

$$w^*(k) = N(I^*_{cmd}(k)) \; = -K_x X_{csc}(k) - K_q q(k) + K_r X^{ref}_{csc}(k) \tag{3.23}$$

where

$$\begin{aligned} \tilde{K} = & \; [K_x \; K_q] = \; (R + \tilde{B}^T P \tilde{B})^{-1} \cdot \tilde{B}^T P \tilde{A} \\ K_r = & \; \left(C \cdot \left(1 - (A - BK_x)\right)^{-1} \cdot B\right)^{-1} \end{aligned}$$

Note that the expression (3.23) is on the form of an **affine state feedback** control law whose gains depends on the solution P of the DARE (3.13). More precisely, the feedforward gain K_r has been included in the control formulation in order to guarantee the position reference X^{ref}_{csc} tracking without any steady state offset.

The general control architecture of the adaptive LQR position control for the K2 Actuator is resumed in Figure 3.5 below.

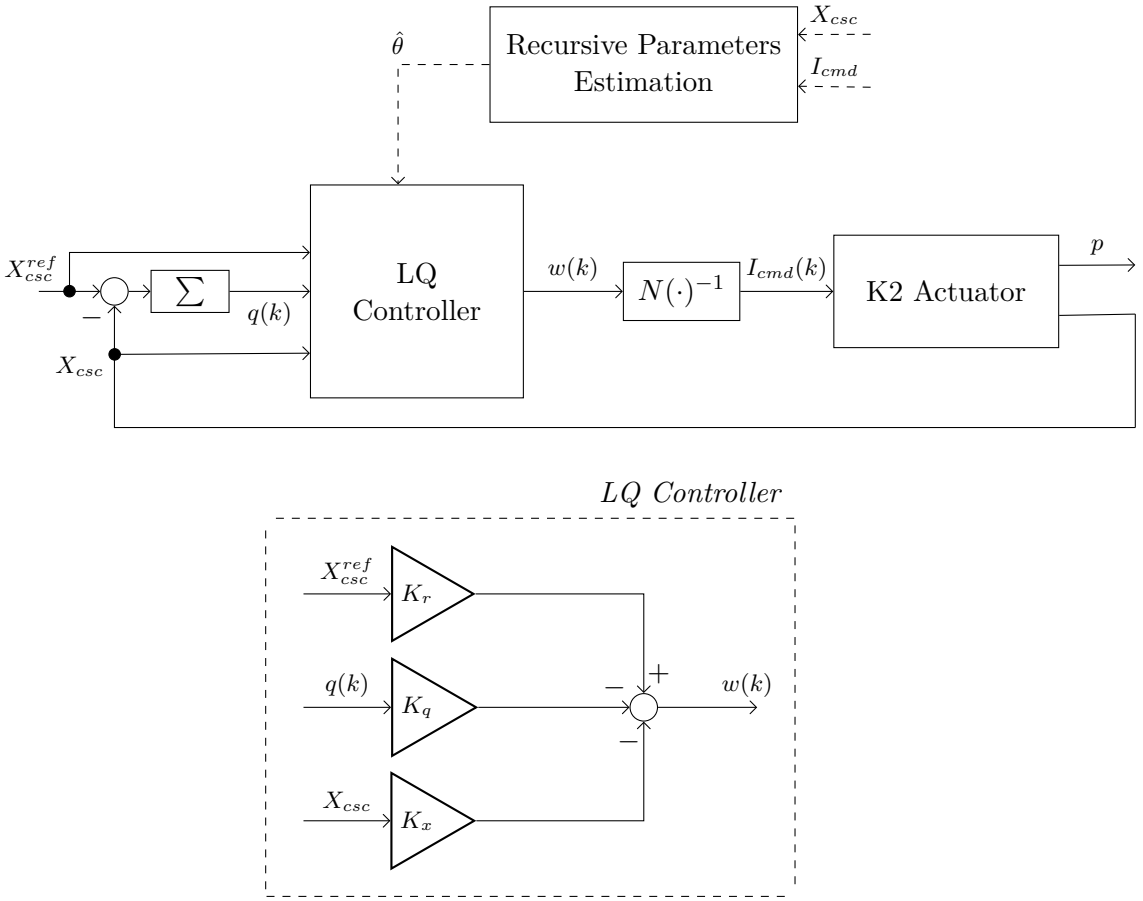

Figure 3.5: Adaptive LQ position control architecture for K2 Actuator.

3.3 Adaptive LQ tuning and simulations results

This section will describe the tuning procedure of the adaptive LQ controller design parameters. Numerical simulations results will be presented in order to evaluate the effectiveness of the adopted control strategy. In particular, a wider spectrum of working situations has been considered. That is, some other position trajectories have been accounted in addition to the position reference profiles previously expressed in section 2.4.

The infinite horizon LQ controller design parameters are the weighting scalar factors Q_y, Q_q and R of the quadratic performance index (3.22) reminded below in tracking form

$$J_{K2}(\tilde{x}(k), W^k) = \sum_{i=0}^{\infty} Q_y\big(X_{csc}(k+i) - X_{csc}^{ref}(k+i)\big)^2 + Rw^2(k+i) + Q_q q(k+i)^2$$

Note that, in the chosen LQ control architecture, an adaptive approach has been developed (section 3.2.1) and an explicit compensation of the static non-linearity has been performed (section 3.2.2) through the dead zone inverting function. As a consequence, the adaptation gain γ and the dead zone amplitude D can be considered as tuning parameters as well.
Similarly as for the MPC, a trial and error procedure has been necessary to ensure a satisfactory trade off between the control performances.

The following settings have been considered for the adaptive LQ controller design

$$\begin{aligned} R &= 1 \\ Q_y &= 2 \cdot 10^5 \\ Q_q &= 1 \\ \text{Dead zone} &= [0; 37]\ \% \\ \gamma &= 0.012 \\ \theta(0) &= [-0.9969 \quad 7.0651 \cdot 10^{-5}] \end{aligned} \tag{3.24}$$

where $\theta(0) = [\alpha_0\ \beta_0]$ is the initial model parameters vector assumed to be inside the medium - high position range identified in section 3.2.1.

The just expressed design parameters values have been achieved by means of a

suitable tuning procedure as described in the following paragraphs.

3.3.1 Tuning of the LQ Cost Function Weights

For the same considerations expressed in section 2.4.2,both the input weight R and the integral action weight Q_q are chosen to be equal as in (3.24). About the output weight Q_y, several simulations have been performed keeping fixed the other design parameters. These have pointed out that a high value of the output weight is responsible for a steady state tracking error reduction especially in high position ranges whereas it induces small oscillations in low position tracking situations. On the contrary, a too small value for Q_y causes a slight worsening of the position tracking performances.

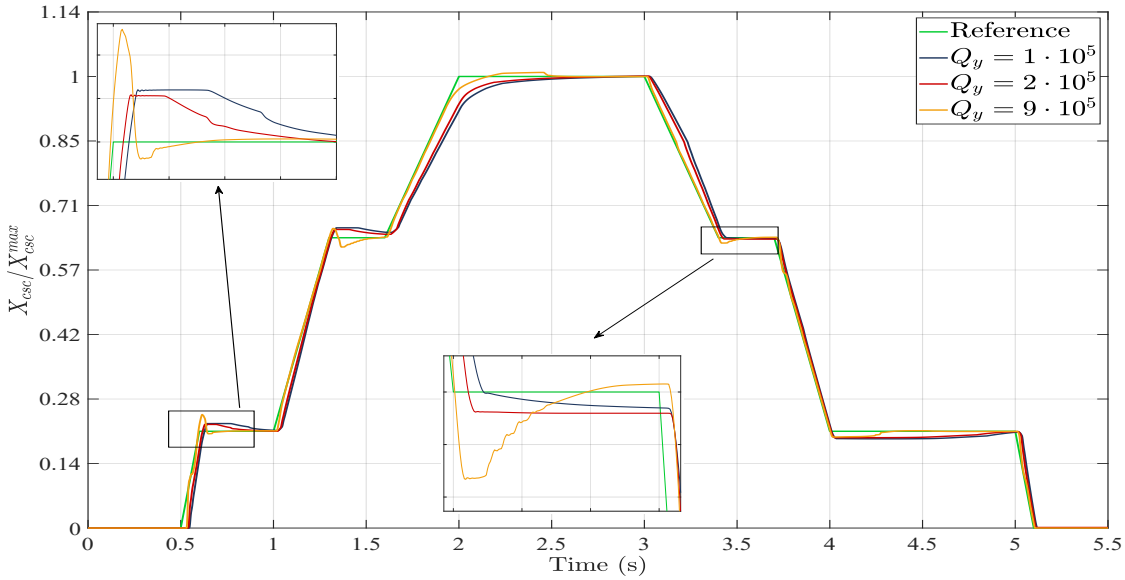

Figure 3.6: Position response with 3^{rd} Gear shift reference profile adopting Adaptive LQ for different values of Q_y.

In Figure 3.6 the output weight Q_y is changed from $Q_y = 1 \cdot 10^5$ to $Q_y = 9 \cdot 10^5$ when one of the difference position trajectories is adopted as reference profile. A suitable trade off between the control objectives has been assured selecting the value $Q_y = 2 \cdot 10^5$ for the output weight.

3.3.2 Tuning of the dead zone amplitude

In section 3.2.2, the Hammerstein model representation of the K2 Actuator has pointed out the need for a two stage control approach. That is, an explicit inversion of the

static non linear function, represented as a dead zone, has to be performed before the command input is applied to the plant.
For this reason, the proper choice of the dead zone amplitude D becomes crucial in the adopted control architecture.

On the basis of the local identification results discussed in section 3.2.1, it is evident that the more the system is demanded to track high values of position references X_{csc}^{ref}, the less is the dead zone between the command input I_{cmd} and the output position X_{csc}.

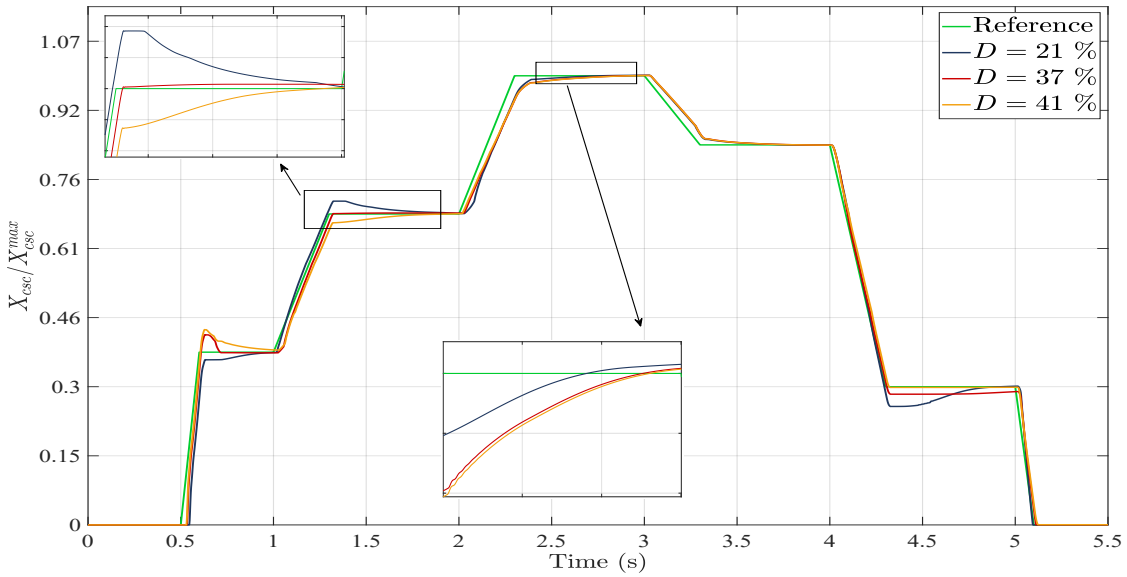

Figure 3.7: Position response with 2^{nd} Gear shift reference profile adopting Adaptive LQ for different values of dead zone amplitude D.

Figure 3.7 confirms the above discussion by showing the influence of the dead zone amplitude variation in the position response when the other design parameters are kept constant as in (3.24).
According to the obtained results, the value $D = 37$ % has been considered as dead zone amplitude in order to ensure satisfactory performances in all the position ranges.

3.3.3 Tuning of the RLS adaptation gain

Similarly as for the adaptive MPC, the recursive algorithm adaptation gain γ plays a key role in the adopted LQ control architecture. This is because the plant parameters estimation speed is related to the adaptive LQ control action, hence, it affects the system response, too.

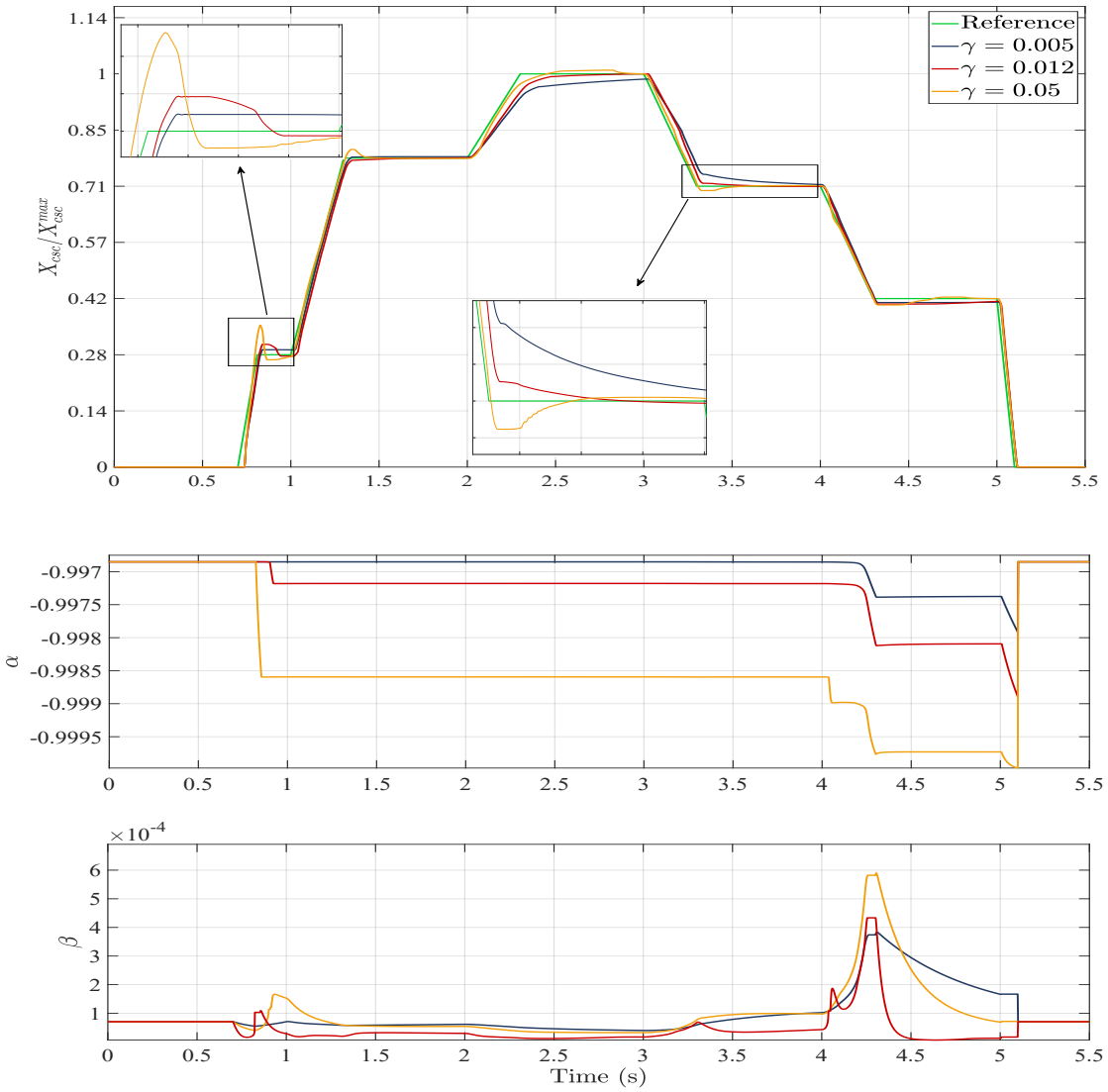

Figure 3.8: (Top) Position response and model parameters variation (Bottom) with 4^{th} Gear shift reference profile adopting Adaptive LQ for different values of γ.

Figure 3.8 above outlines how an high adaptation gain implies a too reactive position response, whereas a small adaptation gain leads to a significant steady state tracking error.

For the same considerations expressed in section 2.4.3, the K2 Actuator high sensitivity with respect to the model parameters variation has been properly handled with a small adaptation gain $\gamma = 0.012$.

The just presented tuning procedure can be resumed with a more general consideration. A crucial role is assumed by the proper choice of the dead zone amplitude D between the command input I_{cmd} and the position output X_{csc} of the K2 Actuator. This is because a biased compensation of the static non linear function leads to an ineffective control action.
The cost function weights and the RLS adaptation gain tuning has been necessary in order to refine the obtained control action.

On the basis of the obtained results, the LQ controller performances have been outlined in the following Table

Range (%)	**Overshoot (%)**	**Rise Time (ms)**	**Steady State Error (mm)**
8 to 21	$\hat{s} \leq 8$	$t_r \leq 32$	$\cong 5 \cdot 10^{-3}$
21 to 57	$\hat{s} \leq 15$	$t_r \leq 34$	$\cong 2 \cdot 10^{-2}$
57 to 85	$\hat{s} \leq 3$	$t_r \leq 38$	$\cong 8 \cdot 10^{-3}$
85 to 100	$\hat{s} \leq 1$	$t_r \leq 45$	$\cong 6 \cdot 10^{-3}$

Table 3.2: Adaptive LQ controller performance resume with different reference profiles.

Referring Table 3.2, the proposed LQ controller design (3.24) is able to guarantee a fast position response with a small steady state tracking error. However, as for the MPC, significant performance improvements, especially in terms of position overshoots, can be obtained by means of a suitable scheduling algorithm as described in the next paragraph.

3.3.4 Scheduling Algorithm

A suitable trade off between a non oscillating input current and a fast position response with small overshoots has been obtained by dynamically tuning the LQ controller parameters on the basis of the working situations.
More precisely, the developed scheduling algorithm exploits a real time adjustment of the dead zone amplitude D and the output weight Q_y considering as input measurement the position reference X_{csc}^{ref} and the tracking error $e_{csc} = X_{csc}^{ref} - X_{csc}$ actual values. The generic strategy is resumed in the following steps (detailed MATLAB code in appendix)

1. MEASURE the reference $X_{csc}^{ref}(k)$ and the tracking error $e_{csc}(k)$ actual values.

2. CHECK the position range on the basis of $X_{csc}^{ref}(k)$.
3. SET the dead zone amplitude D on the basis of the position range.
4. IF$\big(e_{csc}(k) \leq \bar{e}_{csc}\big)$
   4.1 INCREASE the output weight Q_y.
5. IF$\big(e_{csc}(k) > \bar{e}_{csc}\big)$
   5.1 SET the output weight $Q_y = Q_y^{min}$.

where $\bar{e}_{csc}$ is a suitable threshold adopted to handle position references characterized by fast variations in their set point amplitudes.

The just presented pseudo code can be further clarified. Different dead zone amplitudes are adopted in order to assure an efficient non linearity compensation in all the working regions guaranteeing a fast position response. Moreover, a significant steady state error reduction is achieved by increasing the output weight when the position is approaching the set point.

Several simulations have been carried out to test the effectiveness of the adopted scheduled LQ controller and to evaluate some transient and steady-state tracking indices in the presence of different position trajectories. Figures from 3.9 to 3.17 and Table 3.3 resume the obtained results.

Range (%)	**Overshoot (%)**		**Rise Time (ms)**		**Steady State Error (mm)**	
	LQ 1^{st}	LQ 2^{nd}	LQ 1^{st}	LQ 2^{nd}	LQ 1^{st}	LQ 2^{nd}
8 to 21	$\hat{s} \leq 8$	$\hat{s} \leq 5$	$t_r \leq 32$	$t_r \leq 38$	$\cong 5 \cdot 10^{-3}$	$\cong 3 \cdot 10^{-3}$
21 to 57	$\hat{s} \leq 15$	$\hat{s} \leq 7$	$t_r \leq 34$	$t_r \leq 40$	$\cong 2 \cdot 10^{-2}$	$\cong 2 \cdot 10^{-2}$
57 to 85	$\hat{s} \leq 3$	$\hat{s} \leq 2$	$t_r \leq 38$	$t_r \leq 42$	$\cong 8 \cdot 10^{-3}$	$\cong 5 \cdot 10^{-3}$
85 to 100	$\hat{s} \leq 1$	$\hat{s} \leq 1$	$t_r \leq 45$	$t_r \leq 51$	$\cong 6 \cdot 10^{-3}$	$\cong 5 \cdot 10^{-3}$

Table 3.3: Adaptive LQ (LQ 1^{st}) and Adaptive scheduled LQ (LQ 2^{nd}) controller performance resume with different reference profiles.

Referring the above Table, significant performance improvements have been obtained thanks to the adopted scheduling technique. In fact, the proposed adaptive LQ controller guarantees a slight reduction of the steady state tracking error and an evident position overshoots mitigation. Besides, the almost equal response speed has been preserved.

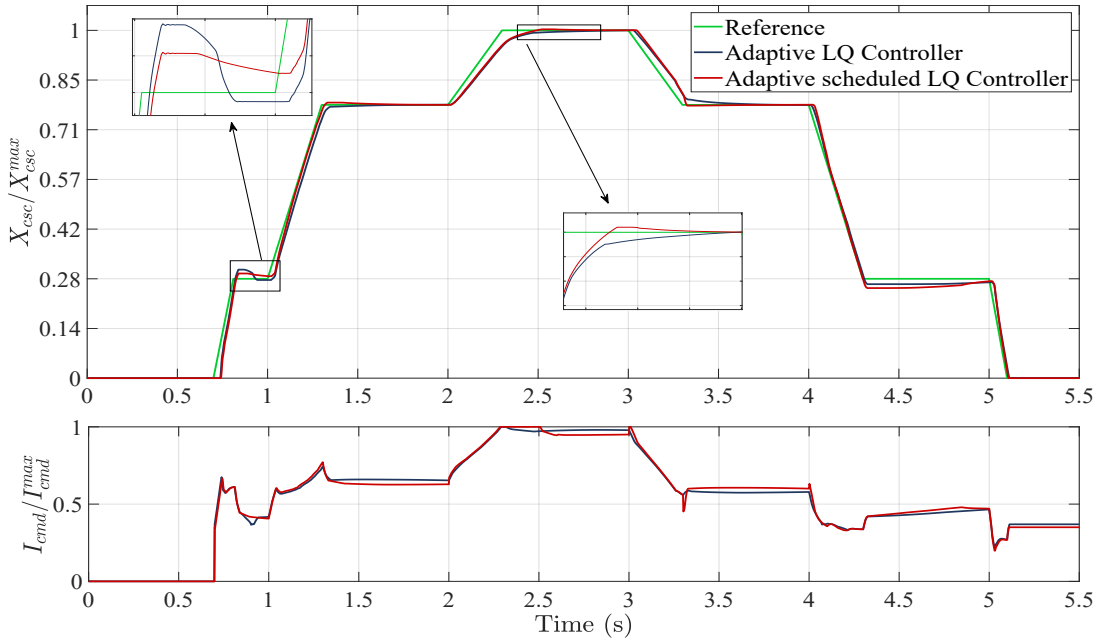

Figure 3.9: (Top) Position response and Input Current (Bottom) with Gear Change profile adopting Adaptive LQ and Adaptive scheduled LQ.

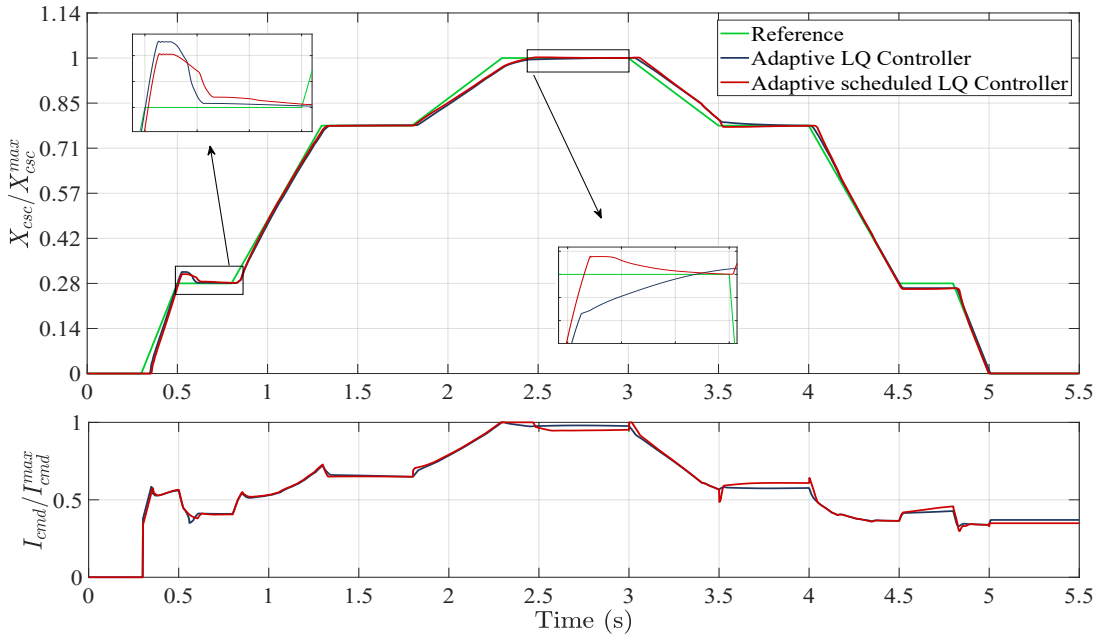

Figure 3.10: (Top) Position response and Input Current (Bottom) with slow Gear Change reference profile adopting Adaptive LQ and Adaptive scheduled LQ.

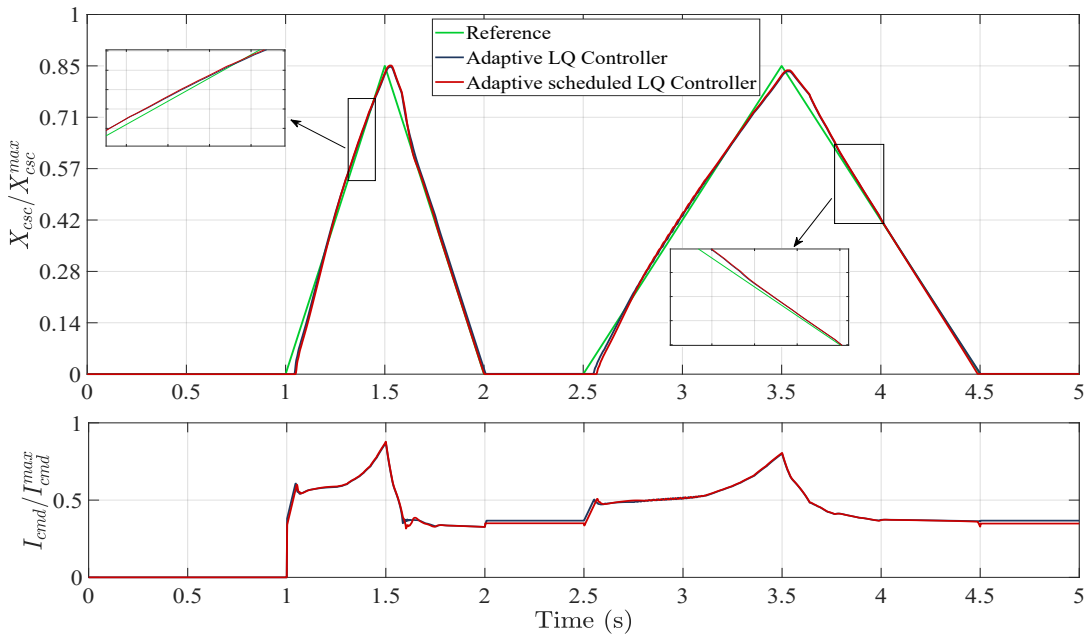

Figure 3.11: (Top) Position response and Input Current (Bottom) with ramp reference profile adopting Adaptive LQ and Adaptive scheduled LQ.

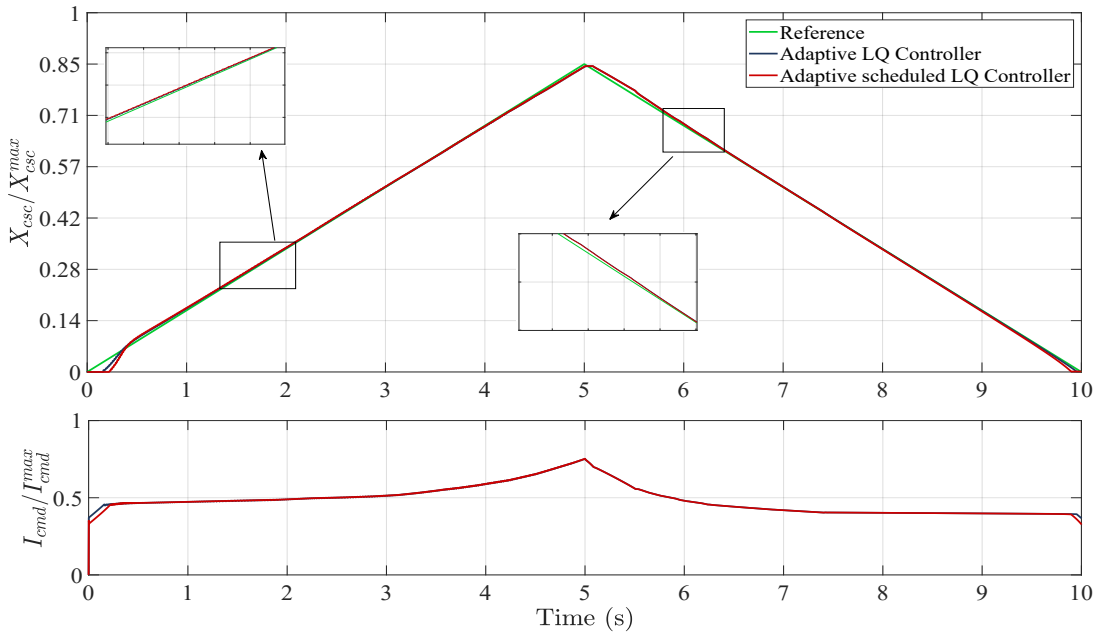

Figure 3.12: (Top) Position response and Input Current (Bottom) with slow ramp reference profile adopting Adaptive LQ and Adaptive scheduled LQ.

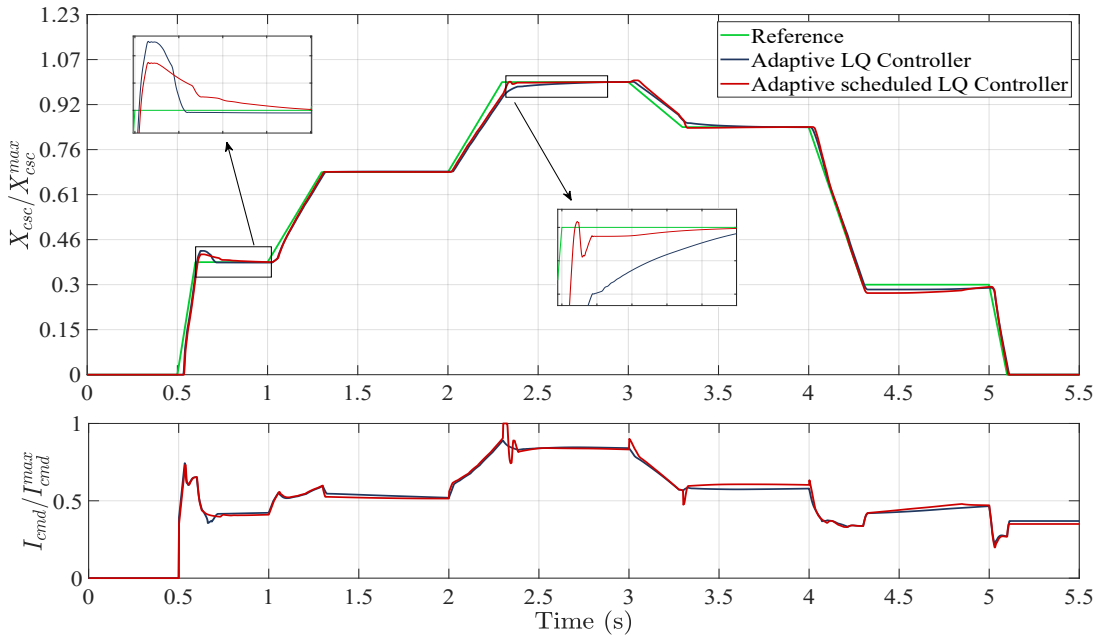

Figure 3.13: (Top) Position response and Input Current (Bottom) with 2^{nd} Gear Change reference profile adopting Adaptive LQ and Adaptive scheduled LQ.

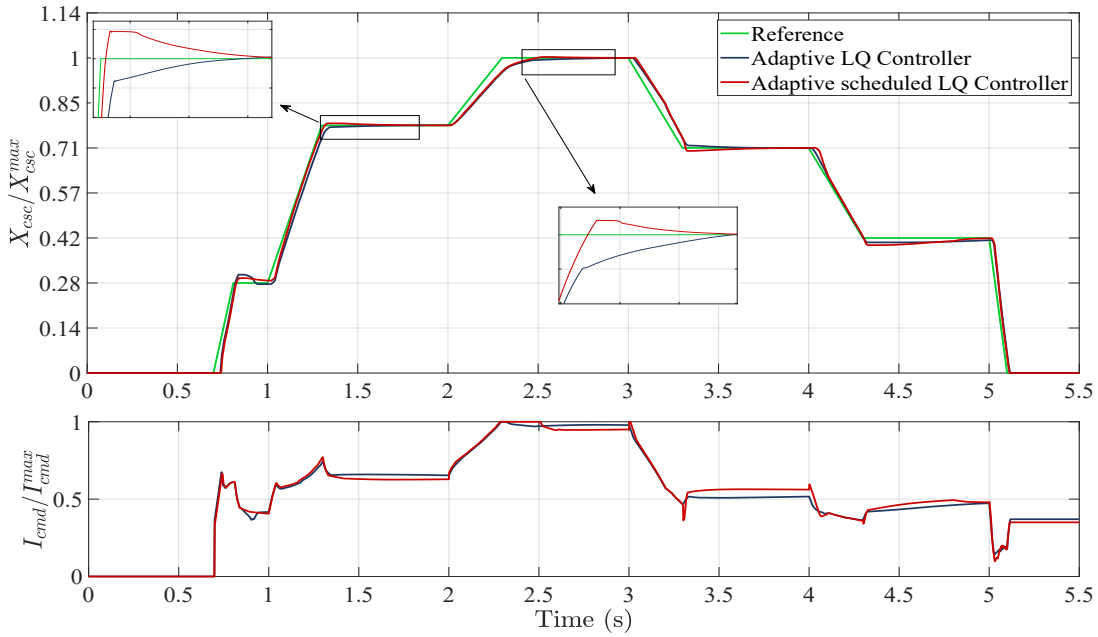

Figure 3.14: (Top) Position response and Input Current (Bottom) with 3^{rd} Gear Change reference profile adopting Adaptive LQ and Adaptive scheduled LQ.

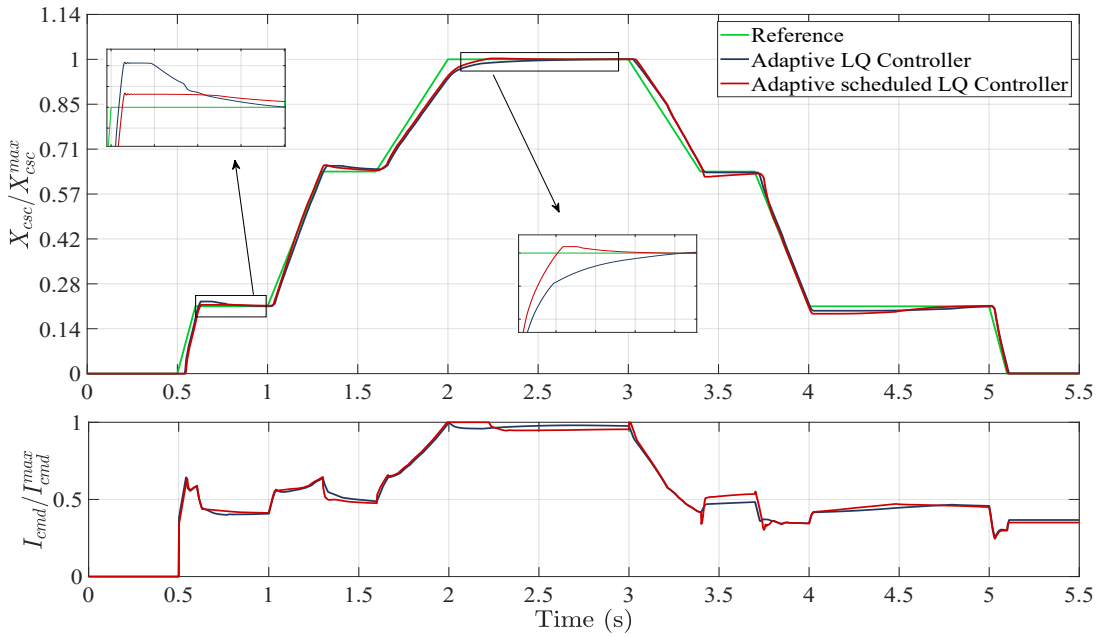

Figure 3.15: (Top) Position response and Input Current (Bottom) with 4^{th} Gear Change reference profile adopting Adaptive LQ and Adaptive scheduled LQ.

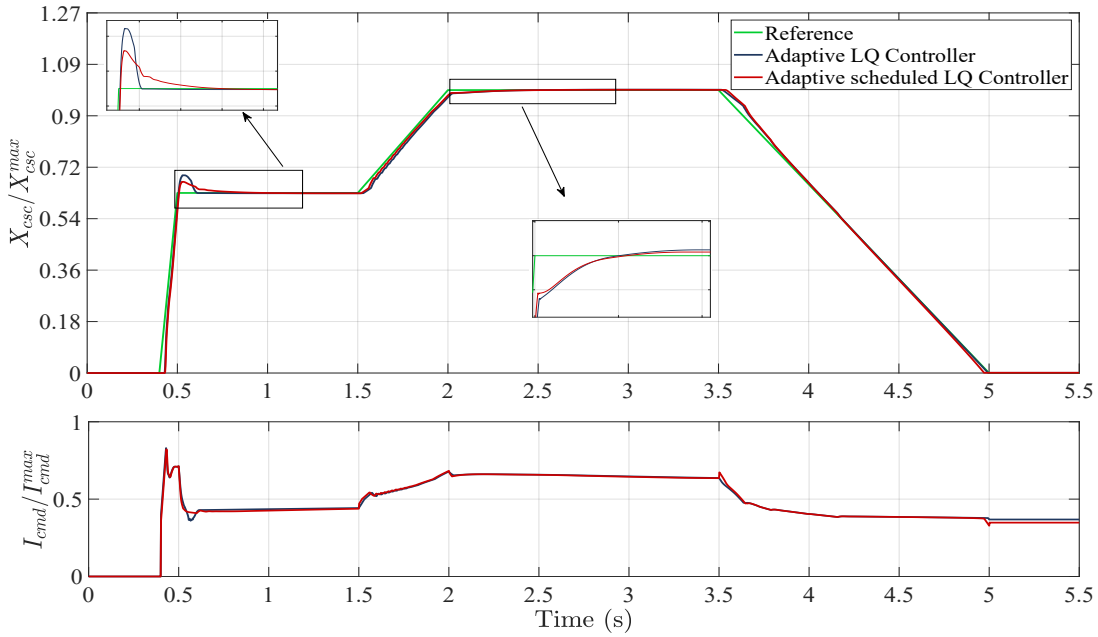

Figure 3.16: (Top) Position response and Input Current (Bottom) with 5^{th} Gear Change reference profile adopting Adaptive LQ and Adaptive scheduled LQ.

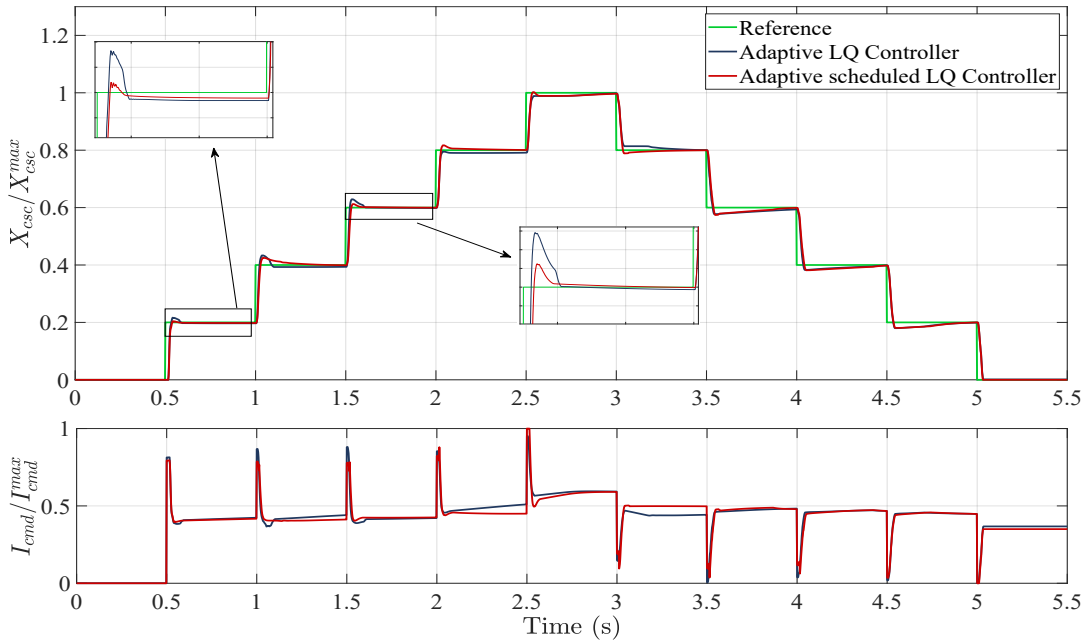

Figure 3.17: (Top) Position response and Input Current (Bottom) with stairs reference profile adopting Adaptive LQ and Adaptive scheduled LQ.

Considering the implementation of the proposed LQ control strategy in a real time platform, the problem of solving a infinite horizon DARE at each sampling time has been overcome by performing an analytic iterative method. That is, the DARE problem has been rearranged in a suitable matrix form so that a structured doubling algorithm (SDA) can be applied. The detailed numerical procedure is explained in the following section.

3.4 Numerical implementation of the infinite horizon DARE

Solving discrete time algebraic Riccati equations (DARE) is essential in many computational problems for different engineering applications. However, a detailed analysis of the DARE mathematical properties is needed in order to find more effective algorithms. The following section is focused on the DARE application to the infinite horizon LQ problem related to the position control of the K2 Actuator. The needed mathematical structures will be briefly presented together with the chosen structured doubling algorithm adopted to find the DARE analytical solution.

3.4.1 DARE and matrix pencils

As discussed in section 3.1.2, the infinite horizon LQ controller, adopted for the K2 Actuator position control problem, is related to the DARE solution X. The following general form is considered

$$\tilde{A}^T X \tilde{A} + \tilde{Q} - (N + \tilde{B}^T X \tilde{A})^T \cdot (R + \tilde{B}^T X \tilde{B})^{-1} \cdot (N + \tilde{B}^T X \tilde{A}) - E^T X E \;=\; 0$$

where $\tilde{A}$, $\tilde{B}$ are the system augmented state matrices, $\tilde{Q}$ is the cost function augmented weighting matrix, R is the control input weighting factor. In this context, it can be assumed that $E = I$ and $N = 0$ since no mixed term is present in the adopted cost function (3.6). Hence, the DARE can be written as

$$X = \; \tilde{A}^T X \tilde{A} + Q - \tilde{A}^T X \tilde{B} \cdot (R + \tilde{B}^T X \tilde{B})^{-1} \cdot \tilde{B}^T X \tilde{A} \tag{3.25}$$

According to [7] and [5], the above DARE solution P can be expressed in terms of deflating subspaces of a suitable matrix pencil (definitions in appendix) $L - zK$.

$$\begin{bmatrix} \tilde{A} & 0 & \tilde{B} \\ -\tilde{Q} & I & 0 \\ 0 & 0 & R \end{bmatrix} - z \begin{bmatrix} I & 0 & 0 \\ 0 & -\tilde{A}^T & 0 \\ 0 & -\tilde{B}^T & 0 \end{bmatrix} \tag{3.26}$$

Hence, the (almost) d-stabilizing solution P is such that

$$\begin{bmatrix} \tilde{A} & 0 & \tilde{B} \\ -\tilde{Q} & I & 0 \\ 0 & 0 & R \end{bmatrix} \begin{bmatrix} I \\ X \\ Z \end{bmatrix} = \begin{bmatrix} I & 0 & 0 \\ 0 & -\tilde{A}^T & 0 \\ 0 & -\tilde{B}^T & 0 \end{bmatrix} \begin{bmatrix} I \\ P \\ Z \end{bmatrix} \Phi \tag{3.27}$$

where $\rho(\Phi) \leq 1$ and Z is associated with the solution of the dual DARE.

According to [7] and [5], if W is a Hermitian matrix such that $\hat{R} = R + R^T W \tilde{B}$ is non singular, the above equation (3.27) can be rearranged as

$$\begin{bmatrix} E_0 & 0 & 0 \\ -P_0 & I & 0 \\ B^T W \tilde{A} & 0 & \hat{R} \end{bmatrix} \begin{bmatrix} I \\ X - W \\ Z \end{bmatrix} = \begin{bmatrix} I & -G_0 & 0 \\ 0 & E_0^T & 0 \\ 0 & -\tilde{B}^T & 0 \end{bmatrix} \begin{bmatrix} I \\ P - W \\ Z \end{bmatrix} \Phi \tag{3.28}$$

where

$$\begin{aligned} P_0 &= \tilde{Q} - W + \tilde{A}^T W E_0 \\ G_0 &= -\tilde{B} \hat{R}^{-1} \tilde{B}^T \\ E_0 &= (I + G_0 W) \tilde{A} \end{aligned} \tag{3.29}$$

The result (3.28) is useful to define the new matrix pencil $\hat{N} - z\hat{K}$

$$\begin{bmatrix} E_0 & 0 \\ -P_0 & I \end{bmatrix} - z \begin{bmatrix} I & -G_0 \\ 0 & E_0^T \end{bmatrix} \tag{3.30}$$

whose form is referred as to **standard structured form-I (SSF-I)** [7]. Therefore, the following equation holds

$$\begin{bmatrix} E_0 & 0 \\ -P_0 & I \end{bmatrix} \begin{bmatrix} I \\ X - W \end{bmatrix} = \begin{bmatrix} I & -G_0 \\ 0 & E_0^T \end{bmatrix} \begin{bmatrix} I \\ X - W \end{bmatrix} \Phi \tag{3.31}$$

and a special structured doubling algorithm can be applied in order to find an analytical expression for the DARE solution X.

3.4.2 DARE solution by means of a structured doubling algorithm

The structured doubling algorithm (SDA) is a recent and advanced technique whose framework is independent from the the DARE application. SDA can be seen as an iterative method for generating a sequence of matrix pencils belonging to the same deflating subspaces and such that their eigenvalues are squared at each step [16], [7].

The application of this class of algorithm to the DARE problem is based of the fact that, there is a one-to-one correlation between the graph invariant deflating subspaces of a matrix pencil and the solutions of the corresponding DARE.
More precisely, the (3.30) is assumed to be the starting matrix pencil with initial values

(3.29). Then, according to [7] and [5], the following sequence $\hat{N}_k - z\hat{K}_k$ can be generated

$$\begin{bmatrix} E_k & 0 \\ -P_k & I \end{bmatrix} - z \begin{bmatrix} I & -G_k \\ 0 & E_k^T \end{bmatrix} \tag{3.32}$$

with

$$\begin{aligned} P_{k+1} &= P_k + E_k^T (I - P_k G_k)^{-1} P_k E_k \\ G_{k+1} &= G_k + E_k (I - P_k G_k)^{-1} G_k E_k^T \\ E_{k+1} &= E_k (I - P_k G_k)^{-1} E_k \end{aligned} \tag{3.33}$$

Therefore, assuming that $\lim_{k\to\infty} P_k = P$, the analytical expression for the DARE solution X can be evaluated in the following way

$$X = P + W \tag{3.34}$$

where W is simply chosen as $W = \sigma I$, with σ a real small positive number such that $\det(R + \tilde{B}^T \sigma I \tilde{B}) \neq 0$.

For the sake of clarity, a simple pseudo code of the adopted SDA algorithm is reported below (detailed MATLAB code in appendix)

1. BUILT the matrices $W = \sigma I$ and $\hat{R} = R + R^T W \tilde{B}$.
2. SET the initial matrices P_0, E_0 and G_0 as in (3.29).
3. FIX the iterative algorithm exit conditions
   - 3.1 SET the minimum error tolerance e^{min} and the error initial value $e_{start} = 1$.
   - 3.2 SET the maximum number of iterations k^{max}.
4. WHILE$\big((e_k > e^{min})\&(k < k^{max})\big)$
   - 4.1 UPDATE the matrices P_k, E_k and G_k as in (3.33).
   - 4.2 EVALUATE the error actual value $e_k = \|E_k\|_1$.
   - 4.3 CHECK the exit conditions.
5. EVALUATE the DARE solution $X = P + W$.

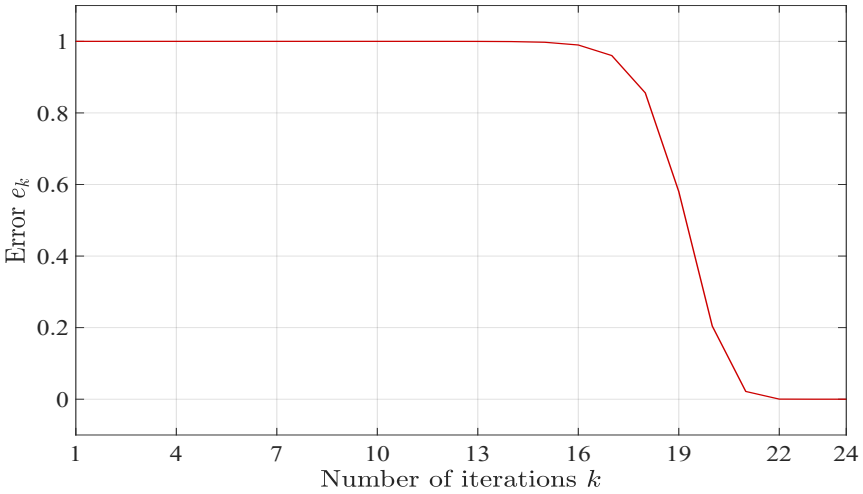

Figure 3.18: Converging trend of the SDA error related to the DARE solution analytical evaluation.

Considering $k^{max} = 30$ as maximum number of iterations and $e^{min} = 10^{-13}$ as error tolerance, the SDA algorithm takes approximately 24 iterations to converge. Figure 3.18 above shows one of the possible error trend when the sub-routine is stopped during the program execution.

Chapter 4

Virtual Sensor based Control for K2 Actuator

In this Chapter, the absence of a real position sensor in the described DDCT system is considered. A Virtual Sensor is employed such that the needed clutch position estimate can be obtained on the basis of available measurements.
The last section presents several simulation results in order to decide the most suitable virtual sensor model structure and control architecture.

4.1 Virtual sensor Overview

A virtual sensing system uses information available from other measurements and process parameters to calculate an estimate of the quantity of interest. Empirical methods base the calculus of the needed estimate on historical measurements of the same quantity, and on its correlation with other available measurements or parameters [21], [26].

In this context, the K2 Actuator virtual sensor to be modelled is considered as a SISO black box model that transforms, in real time, the actual measure of the output pressure p in the clutch position estimate $\hat{X}_{csc}$.

A preliminary analysis of the data sets, provided by Centro Ricerche Fiat, showed the presence of a dead zone between the pressure and the position. The relationship between the involved variables is showed in Figure 4.2.

As highlighted by the above curve shape, the K2 Actuator position values are related to the output pressure in a non linear manner. For this reason, in the following section

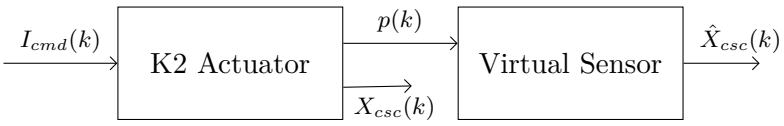

Figure 4.1: Black box model of the Virtual Sensor for the K2 Actuator.

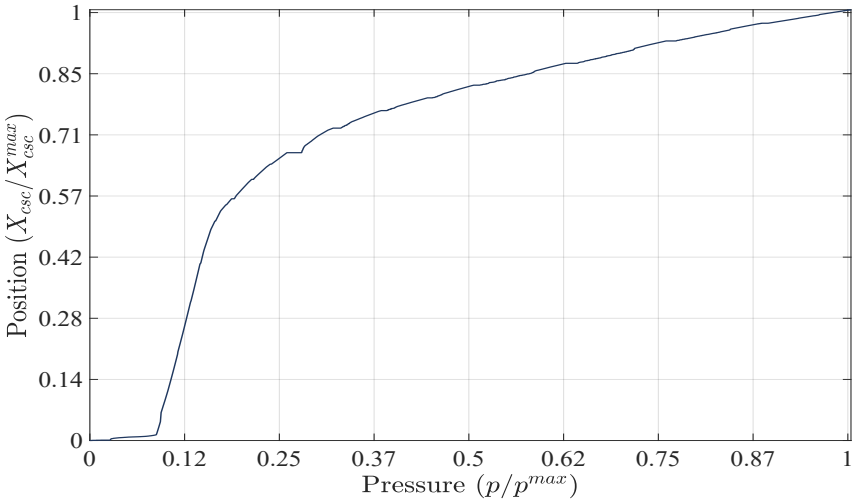

Figure 4.2: Pressure-Position curve shape for the given data set.

different non linear model structures are proposed to represent the Virtual Sensor.

4.2 Virtual Sensor Modelling

The non linear characteristic between the K2 Actuator output pressure and the clutch position value has been reproduced by considering two different strategies to develop a Virtual Sensor.
The first method, whose description can be found in section 4.2.1, involves the use of Multilayer Feedforward Neural Networks to describe the static relationship between the pressure $p(k)$ and the position $X_{csc}(k)$ [1].
The second approach, presented in section 4.2.2, tries to simplify such a Neural Network based Virtual Sensor by means of a block-structured model design. The Hammerstein-Wiener has been demonstrated to be the most suitable model and the Set-Membership identification methodology has been exploited to determine the proper parameters values [9].

4.2.1 Static feedforward Neural Network model structure

The chosen class of Neural Network is based on a layer-by-layer forward propagation of the pressure input signal. The training procedure, exploited through the MATLAB Neural Network Toolbox, is based on the error back-propagation algorithm and implemented in a supervised manner following the Levenberg-Marquardt optimization method (for more details see [1], [14], [24], [18]).

The following network anatomy, showed in Figure 4.3, has been selected

- Input Layer: acquires the input p_t at each sampling time $T_s = 2$ ms.
- Hidden Layer: in each neuron the input p_t is weighted by the term w_j and biased by b_j. Then, the Hyperbolic Tangent Function is chosen as activation function to obtain the output vector $y = [y_1 \ ... \ y_{10}]$ as reported in the equation below.

$$y_j = \frac{2}{1 + e^{-2\varphi_j}} - 1 \quad \text{for } j = 1...10$$

$$\text{where } \varphi_j = w_j \cdot p_t + b_j$$

- Output Layer: produces the Neural Network output by adjusting the Hidden Layer output vector $[y_1 \ ... \ y_{10}]$ with the weighting terms $[w_{x_1} \ ... \ w_{x_{10}}]$ and the bias term b_x. The resulting Virtual Sensor's position estimate $\hat{X}_{csc}$ is given by

$$\hat{X}_{csc}(t) = f(p_t) = \begin{bmatrix} w_{x_1} & \cdots & w_{x_{10}} \end{bmatrix} \begin{bmatrix} y_1 \\ \vdots \\ y_{10} \end{bmatrix} + b_x \tag{4.1}$$

Note that, the use of the hidden layer allows the network to learn more complex tasks or, in this context, to perform a more accurate identification of the K2 Actuator non linear dynamics. Indeed, this kind of Neural Network, referred as Multilayer Perceptrons (MLPs), is able to extract more meaningful features from the input patterns [14].

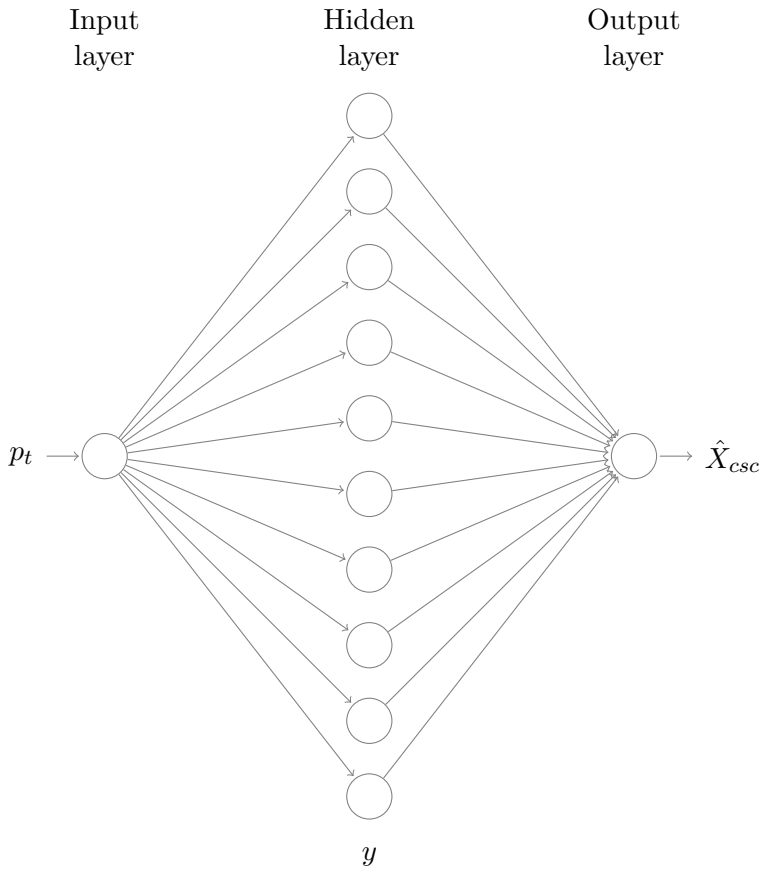

Figure 4.3: Neural Network Anatomy for the K2 Actuator Virtual Sensor.

The Neural Network parameters values are resumed in Table 4.1.

	Hidden Layer		**Output Layer**	
Neuron	Weights w_j	Bias b_j	Weights w_{x_j}	Bias b_x
1	10.833	-11.631	0.03	
2	11.826	-10.641	0.023	
3	10.47	-7.141	0.039	
4	-10.569	4.553	-0.05	
5	-223.276	2.895	-0.048	0.019
6	4.755	0.959	0.123	
7	37.33	14.2855	0.046	
8	22.656	12.156	0.054	
9	10.122	7.688	0.751	
10	237.94	198.521	-0.133	

Table 4.1: Neural Network parameters values.

The validation procedure, performed on the given data set and portrayed in Figure 4.4, shows a quite good tracking of the measured output position. Therefore, it can be stated that the developed Virtual Sensor, represented by a static Neural Network, is able to provide an accurate position estimate $\hat{X}_{csc}$ on the basis of the K2 Actuator output pressure actual value.

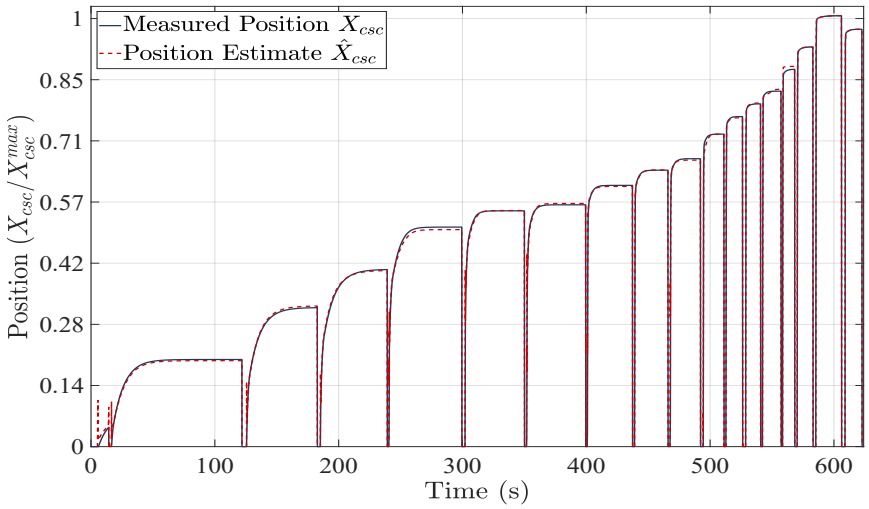

Figure 4.4: Neural Network based Virtual Sensor validation.

However, despite of the good tracking performances, such a developed Virtual Sensor is not easy to be deployed in a real time platform because of the complexity related to the Neural Networks inherent estimation approach. For this reason, in the following paragraph, a simplified Virtual Sensor has been developed to obtain the needed position estimate by means of manageable block structures.

4.2.2 Robust Hammerstein-Wiener model structure

The aim of this strategy is to build a Virtual Sensor by considering an uncertain system identified through a robust identification method. That is, the given experimental measurements are assumed to be affected by unknown but bounded (UBB) error so that the Set-membership identification technique can be applied to evaluate the Parameters Uncertainty Intervals (PUIs) of the pre-defined model structures.
Once computed the proper bounding sets, the central estimate has been considered as the most suitable parameter value for all the chosen model structures.

According to [9], the following Hammerstein-Wiener model has been chosen

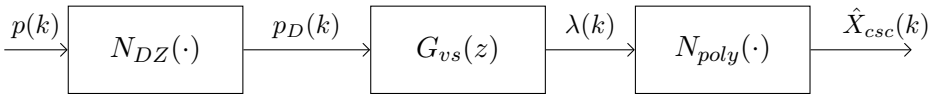

Figure 4.5: Hammerstein-Wiener block diagram representation for the K2 Actuator Virtual Sensor.

where $N_{DZ}(\cdot)$ and $N_{poly}(\cdot)$ are, respectively, the input non liner block represented by a dead zone and the output non linear block modelled by a finite polynomial. Moreover, the dynamic part between the dead zone processed pressure $p_D(k)$ and the inner signal $\lambda(k)$ is represented by the discrete time transfer function $G_{vs}(z)$.

The identification procedure has been formulated in terms of optimization problems as asserted by the Set-membership methodology. Linear programming methods as well as convex relaxation techniques have been applied in order to handle such a diverse optimization framework. More precisely, linear programming and convex optimization problems have been solved by MATLAB routines whereas the global optima of the non convex polynomial optimization problems have been evaluated through the SparsePOP MATLAB toolbox.

Referring to Figure 4.5, the adopted Hammerstein-Wiener Virtual Sensor settings can be resumed in the following way (details are explained in [9])

- Input non linear block $N_{DZ}(\cdot)$: as already mentioned, data analysis has confirmed the presence of a pressure dead zone according to which positive pressure values result in null position response. The identified range is found to be

$$\text{Dead Zone} = [0 - 3.89] \text{ bar} \tag{4.2}$$

- Output non linear block $N_{poly}(\cdot)$: under the assumption of invertible non linearity, the polynomial coefficients have been evaluated by considering the inverse identification problem. That is, the experimental steady state operating conditions p_D^{ss} and X_{csc}^{ss} have been exploited yielding the following fourth degree polynomial

$$p_D^{ss} = N^{-1}(X_{csc}^{ss}) = 0.5766 \cdot X_{csc}^{ss} - 0.6147 \cdot (X_{csc}^{ss})^3 + 0.02174 \cdot (X_{csc}^{ss})^4 \tag{4.3}$$

- Linear dynamic Model $G_{vs}(z)$: the identification procedure has been carried out in the output error model framework considering the known input samples $p_D(k)$ and the uncertain output signal $\lambda(k)$. Additional constraints have been included in the optimization problem in order to ensure the system BIBO stability and unitary steady-state gain. The identified transfer function is characterized by a first order dynamics expressed as

$$G_{vs}(z) = \frac{0.3132}{z - 0.6868} \tag{4.4}$$

Figure 4.6 illustrates the validation procedure performed on the given data set.

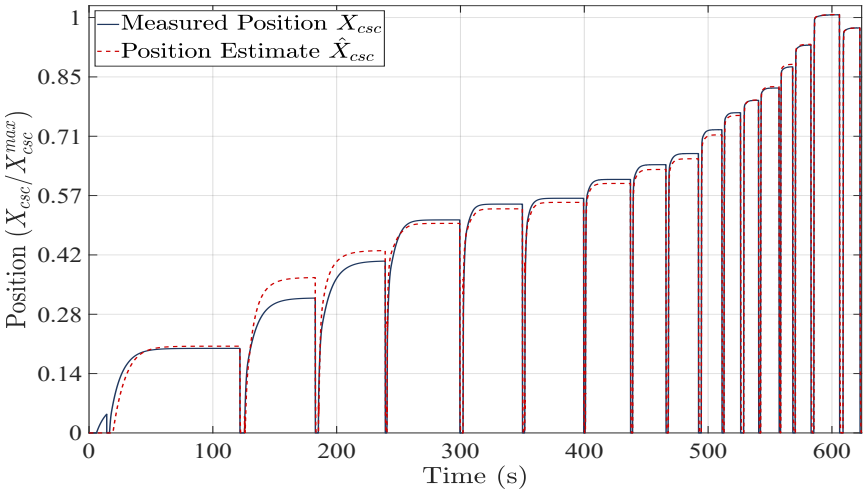

Figure 4.6: Hammerstein-Wiener model based Virtual Sensor validation.

As can be seen from the above image, this Virtual Sensor modelling strategy represents a trade off between the tracking accuracy and the ease of implementation. However, the reduction of the position estimate oscillations is a considerable achievement for our final control purpose since the estimated position is directly fed back to track a given reference trajectory.

4.3 Virtual sensor based Control Architectures

The previous section has dealt with two different modelling strategies to obtain the proper clutch position estimate by means of the available K2 Actuator output pressure. In the following, both the developed Virtual Sensors are included in different architectures that are based on single or nested feedback loops control techniques.
Several simulations results are presented with the aim of comparing and testing each control strategies in terms of different performance criteria such as the real position tracking behaviour, the command input oscillations reduction, the smoothness of the position estimate.

Specifically, an Up shift manoeuvre is considered as the typical working situation. This is to properly test the effectiveness of the control techniques when a congenial position reference, and hence pressure trajectory, is involved in the Virtual Sensor based architecture.

Moreover, such a Virtual Sensor based control strategy causes the need to adopt the following adjustments

- the plant model, needed for the controller design, has been re-identified by forcing the K2 Actuator with a suitable input sequence and collecting as output the corresponding position estimate $\hat{X}_{csc}$ provided by the Virtual Sensor.
- the command input I_{cmd} and output position X_{csc} closed loop data have been acquired and considered as training data set for the Neural Network based Virtual Sensor.

A more appropriate initial model parameters vector improves the recursive algorithm accuracy. Besides, a more consistent training data set enhances the accuracy of the Virtual Sensor position estimation thanks to the inherent auto-learning capability that distinguishes the Neural Networks.

4.3.1 Single feedback Loop Control

A Virtual Sensor based control technique using a single feedback loop is treated in the following paragraph. That is, a position control strategy , based on the position estimate $\hat{X}_{csc}$ provided by the Virtual Sensor, has been exploited such that the real K2 Actuator output position X_{csc} follows a suitable reference profile.

The general control architecture is showed in Figure 4.7.

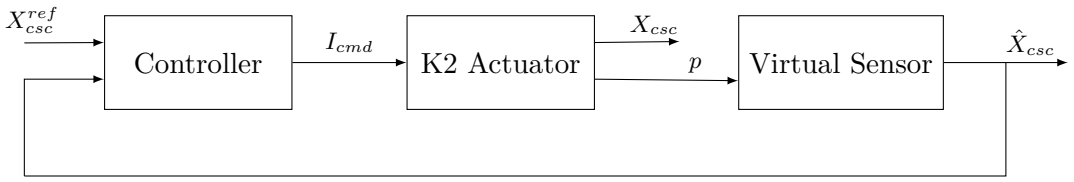

Figure 4.7: General block scheme of the single feedback loop Virtual Sensor based control architecture for the K2 Actuator.

As to the Controller, both the MPC and the LQR control methods have been employed while, as to the Virtual Sensor, either the Neural Network or the Robust Hammerstein-Wiener modelling procedures have been considered.

LQ position Control

The LQ controller design strategy is the same discussed in Chapter 3.
However, since the position estimate provided by the Virtual Sensor is characterized by sudden jumps in low position ranges, some modifications have been necessary. A fine tuning of the weighting matrices, as well as the other involved design parameters, has been exploited until satisfactory control performances are achieved. The tracking weight Q_y has been dynamically adjusted on the basis of the working region so that sudden jumps of the position estimate values are properly reduced.

The chosen parameters settings are

$$\begin{aligned} R &= 1 \\ Q_y &= [0.5 \div 9] \cdot 10^5 \\ Q_q &= 1 \\ \text{Dead zone} &= [0; 37] \ \% \\ \gamma &= 0.0065 \\ \theta(0) &= [-0.9872 \quad 1.1069 \cdot 10^{-4}] \end{aligned} \tag{4.5}$$

where $\theta(0) = [\alpha_0 \ \beta_0]$ is the initial model parameters vector re-identified considering the Virtual Sensor.

Figure 4.8 depicts the K2 Actuator position response, the Virtual sensor position estimate, the output pressure and the LQ controller command action when the position estimate $\hat{X}_{csc}$ is provided by three types of different modelled Virtual Sensors.

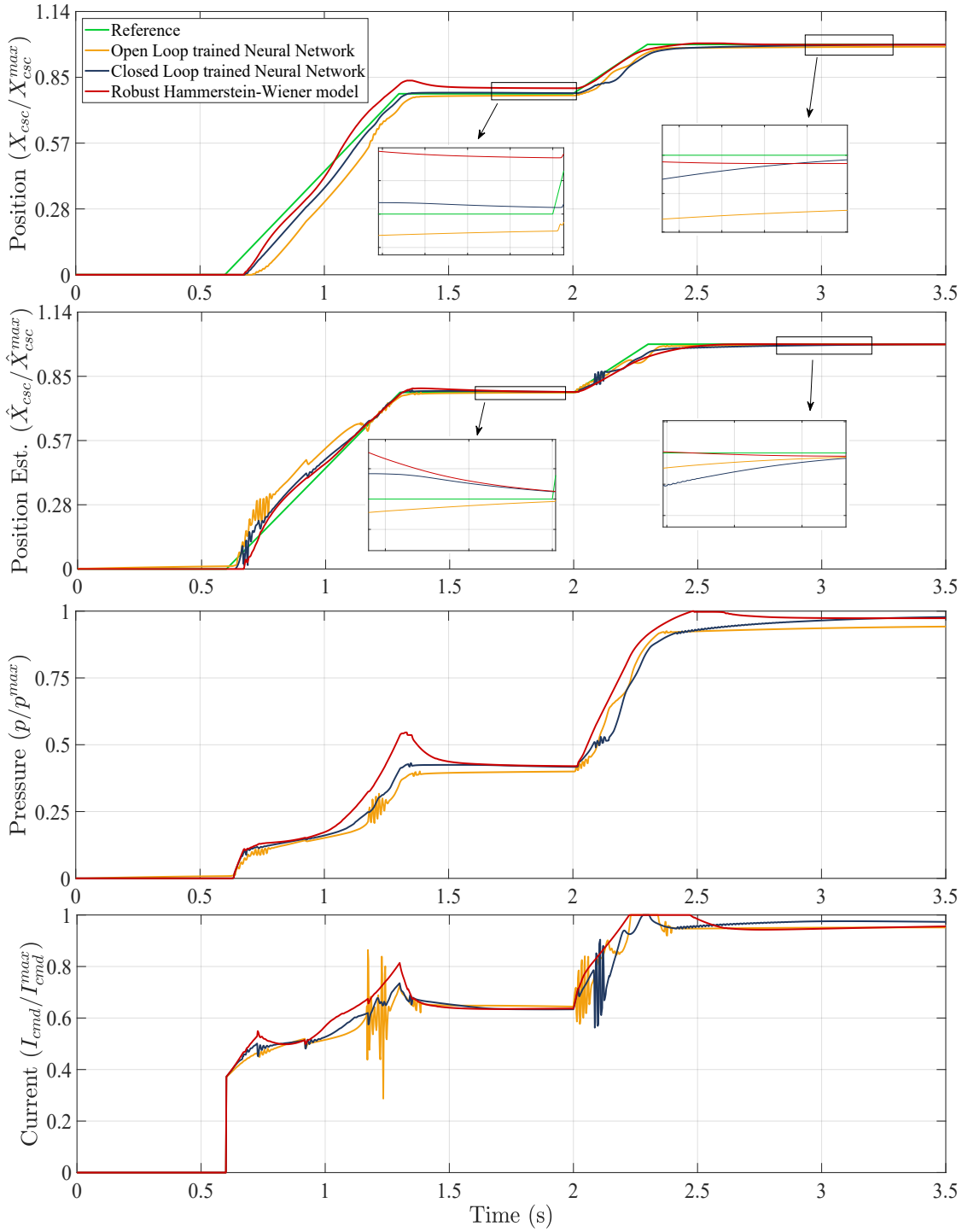

Figure 4.8: Position response, position estimate, output pressure, command input with Up shift reference profile adopting a Virtual Sensor based LQ control.

From the reported image it is evident that, adopting such a LQ position control strategy, satisfactory trade off results are obtained when the closed loop trained Neural Network is adopted as Virtual Sensor. A significant ripple attenuation is achieved together with an appreciable steady state tracking behaviour indeed.

However, it is worth to highlight that the best results in terms of smoothness are obtained by means of the robust Hammerstein-Wiener Virtual Sensor. This can be related to the dynamic part of the adopted model structure that, acting as a filter, is able to reduce the output oscillations.

MPC position Control

The same approach described in Chapter 2 is adopted to design the MPC position controller.
On the other hand, the fact that the Virtual Sensor position estimate is directly fed back to the controller has caused the need to adjust the involved design parameters, and to exploit a dynamic tuning of the tracking weight Q_y and the prediction horizon H_p yielding the following settings

$$\begin{aligned} R &= 1 \\ Q_y &= [1 \div 25] \cdot 10^2 \\ Q_q &= 1 \\ H_p &= [28 \div 36] \\ |\Delta I_{cmd}^{max}| &= 500 \ \tfrac{mA}{T_s} \\ \gamma &= 0.01 \\ \theta(0) &= [-0.9872 \quad 1.1069 \cdot 10^{-4}] \end{aligned} \tag{4.6}$$

where $\theta(0) = [\alpha_0 \ \beta_0]$ is the initial model parameters vector re-identified considering the Virtual Sensor.

Figure 4.9 confirms the considerations made in the previous paragraph. The filtering properties associated with the Hammerstein-Wiener Virtual Sensor lead to a remarkable ripple mitigation. Moreover, the inclusion of state and input constraints in the control problem formulation has given further benefits in terms of smoothness with respect to the previously presented Virtual Sensor based LQ control strategy.

However, the best compromise between the response speed and the steady state error minimization is still obtained by means of the closed loop trained Neural Network.

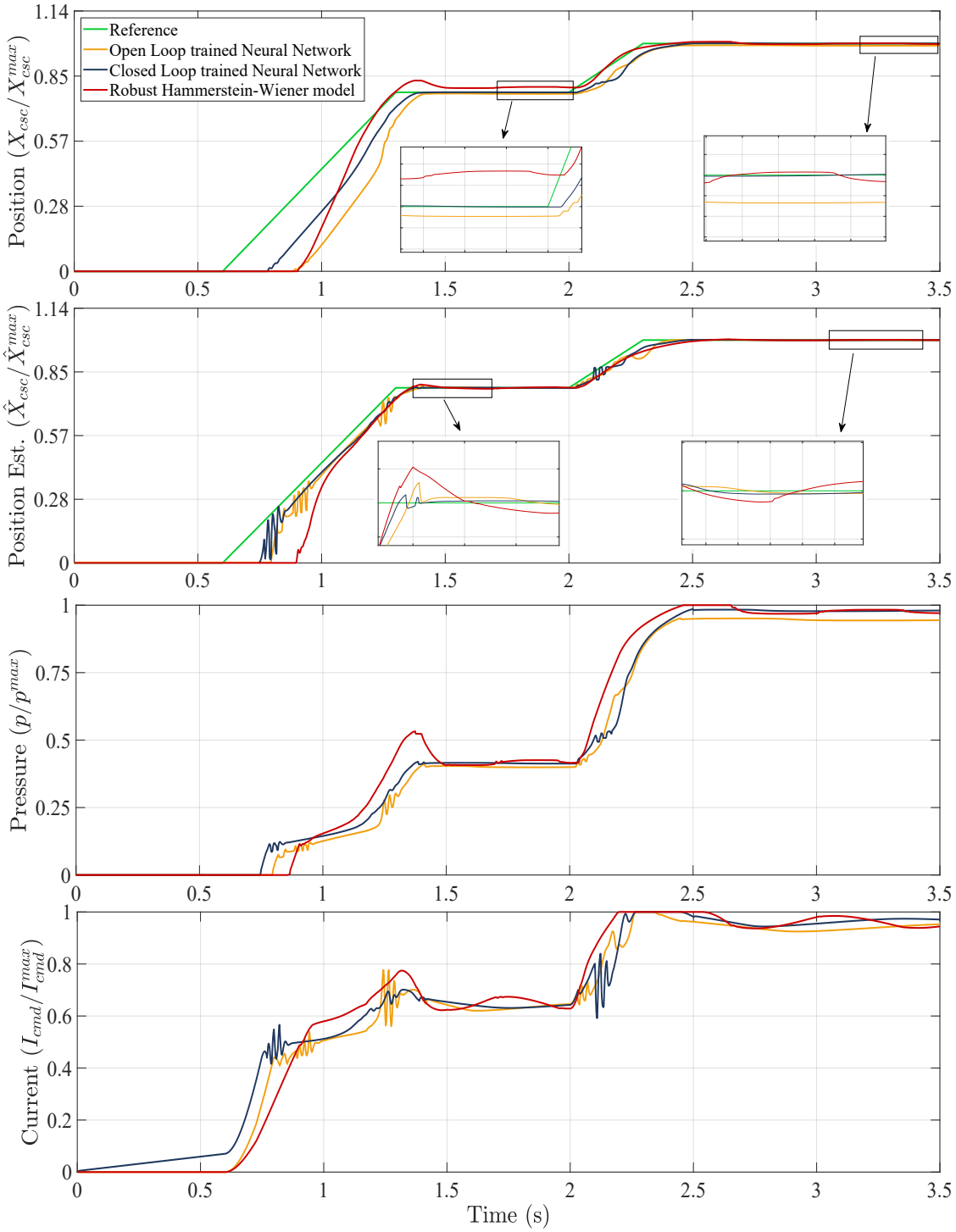

Figure 4.9: Position response, position estimate, output pressure, command input with Up shift reference profile adopting a Virtual Sensor based MPC control.

For the sake of completeness, in the next section a Virtual Sensor based nested loop control approach is described. The intent is to study if an inner loop that controls the K2 Actuator output pressure can give benefits in terms of the position estimation, hence improving the overall control performances.

4.3.2 Nested feedback Loops Control

For nested feedback loops control it is intended a control architecture in which two cascade controllers are implemented. More precisely, they are functionally wired in the sense that the outer loop literally commands the inner loop by adjusting its set point.

In this context, a secondary pressure control is nested inside a primary position control which is designed on the basis of the position estimate $\hat{X}_{csc}$ provided by the Virtual Sensor. The aim is to improve the control action effectiveness by refining the position estimate by means of a controlled pressure response.

Figure 4.10 shows the general control architecture.

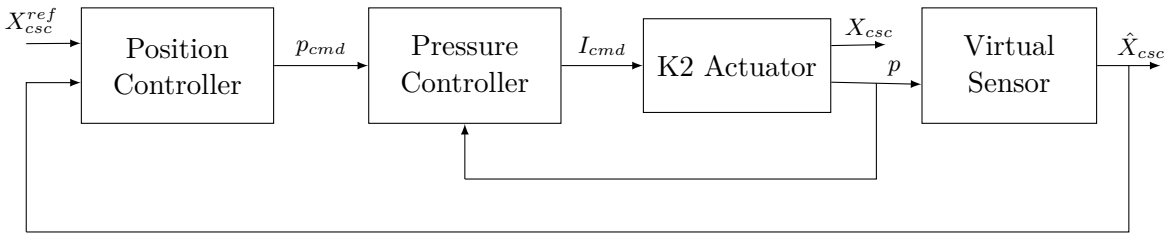

Figure 4.10: General block scheme of the nested feedback loops Virtual Sensor based control architecture for the K2 Actuator.

As to the outer position loop, only the Virtual Sensor based LQ control strategy has been implemented. This decision is motivated by the fact that, as presented in the previous paragraph, the explicit feed forward action, provided by the LQ control law, guarantees a fast position response without significantly increasing overshoots and steady state errors.
On the contrary, two different strategies are proposed for the inner pressure loop as described in the following paragraphs.

Outer LQ position Control

The LQ position control has been designed observing the same principles described in Chapter 3. However, since the relationship between the clutch position and the output pressure can be well represented by a linear first order model with gain β and pole α (details in [1]), an explicit dead zone compensation is not needed any more. Besides, the tracking weight has to be adjusted on the basis of the working point to guarantee a smooth behaviour.

A suitable trade off between a non oscillating command pressure and a satisfactory position reference tracking has been achieved by setting the following controller parameters values.

$$\begin{aligned} R &= 1 \\ Q_y &= [0.5 \div 9] \cdot 10^2 \\ Q_q &= 1 \\ \gamma &= 0.03 \\ \theta(0) &= [-0.7489 \quad 0.098] \end{aligned} \tag{4.7}$$

where $\theta(0) = [\alpha_0 \ \beta_0]$ is the initial model parameters vector as in [1].

Inner pressure Control using Pole Placement techniques

The key idea of this approach is to design a controller for the K2 Actuator such that the closed loop poles lays in assigned positions on the complex plane.
In this regard, since the linear model that relates the command input with the output pressure is represented by a first order transfer function G_{Ip}(details in [1]), a one degree of freedom (1dof) controller is needed. Figure 4.11 shows the adopted 1dof control architecture.

According to [6] and [23], the associated diophantine equation leads to the following structure for the 1dof controller with built-in integrator

$$C(z) = \frac{s_0 \ (z + \sigma)}{z - 1} = \frac{U(z)}{E(z)}$$

where $U = I_{cmd}$, $E = p_{cmd} - p$ is the tracking error, $\sigma = -9.83 \cdot 10^{-3}$ is the pole of the plant model G_{Ip}.

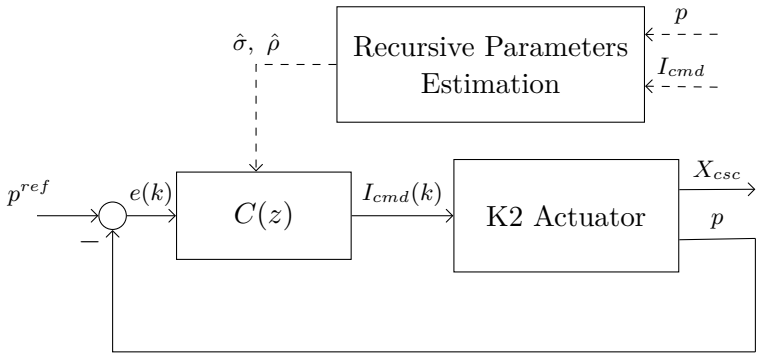

Figure 4.11: Adaptive 1dof inner pressure control architecture for K2 Actuator.

Specifically, on the basis of the plant model G_{Ip} dominant dynamics, the closed loop characteristic polynomial is chosen to be

$$A_m(z) = z + \lambda_{CL} = z - 0.96 \tag{4.8}$$

Hence, the controller design parameter is fixed according to

$$s_0 = \frac{1 + \lambda_{CL}}{\rho}$$

where $\rho = 5.3 \cdot 10^{-2}$ is the gain of the plant model G_{Ip}.

Moreover, an adaptive approach has been exploited so that the plant model parameters $\hat{\sigma}$, $\hat{\rho}$ and, consequently, the controller design parameter $\hat{s}_0$, are adjusted in real time on the basis of the command input I_{cmd} and output pressure p actual values.

The resulting control action that guarantees an adaptive pole cancellation is

$$I_{cmd}(k) = I_{cmd}(k-1) + \hat{s}_0 \left(e(k) + \hat{\sigma}\ e(k-1)\right) \tag{4.9}$$

Finally, employing the LQ position controller 4.7 and the Virtual Sensor in the outer loop, the just described nested control strategy has been adopted to track an Up shift position reference profile. Figure 4.12 shows the K2 Actuator position response X_{csc}, the position estimate $\hat{X}_{csc}$, the output pressure p and the command input I_{cmd} in the presence of three different Virtual Sensor modelling approaches.

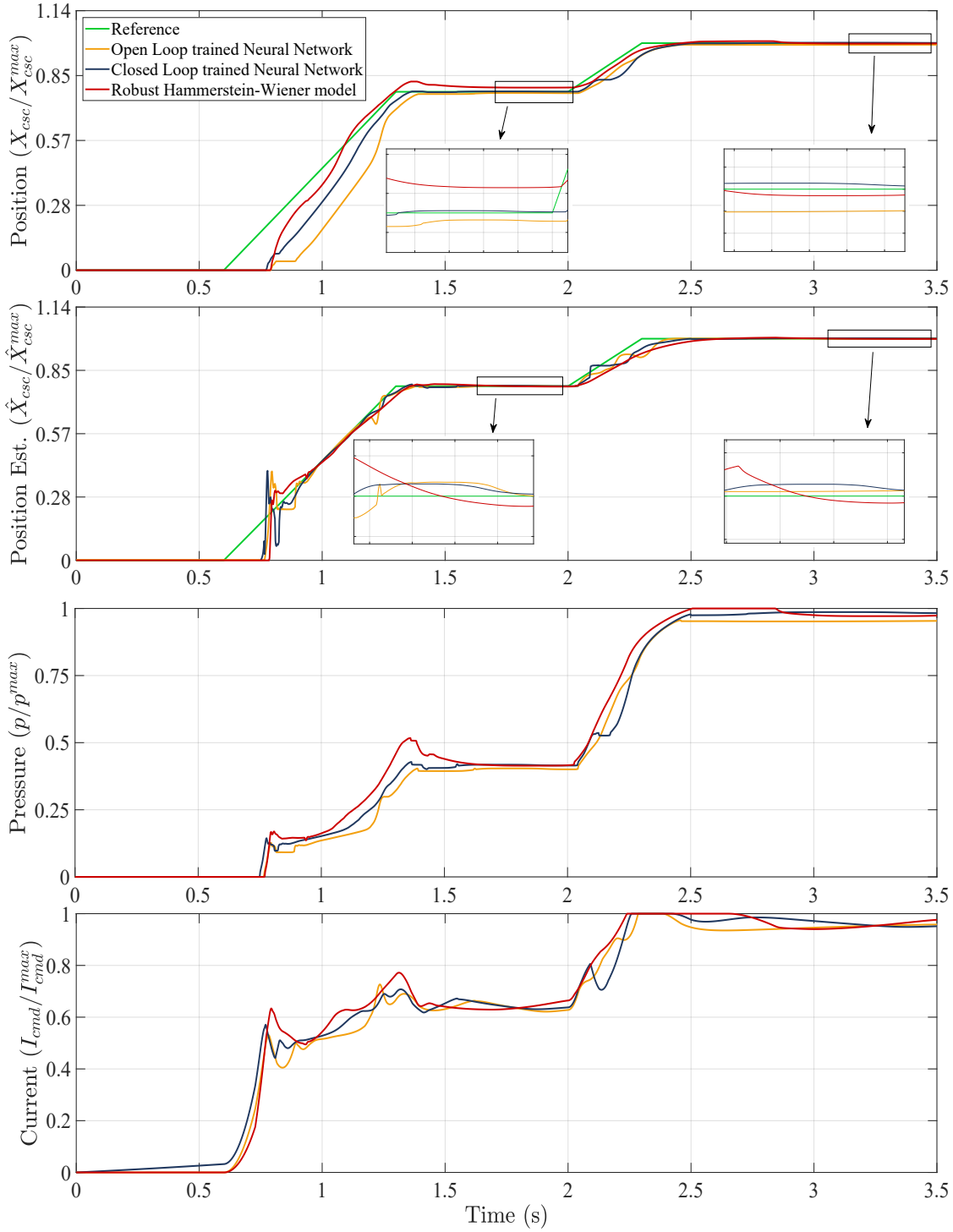

Figure 4.12: Position response, position estimate, output pressure, command input with Up shift reference profile adopting a Virtual Sensor based nested control (LQ outer loop - 1dof inner loop).

As confirmed by the reported simulation results, the inclusion of the 1dof inner pressure loop gives particular benefits in terms of position estimate smoothness for all the Virtual Sensor models employed in the outer position loop.
Secondly, in comparison with the feedback position control discussed in the previous section, a faster response is achieved without significantly increasing the position overshoot.

Improvements are proposed in the next paragraph by enhancing the internal pressure control loop with a MPC control architecture.

Inner pressure Control using MPC techniques

On the basis of the same procedures described in Chapter 2, an adaptive Model Predictive Control architecture has been exploited for the inner pressure loop of the K2 Actuator.

Figure 4.13 resumes the general MPC layout.

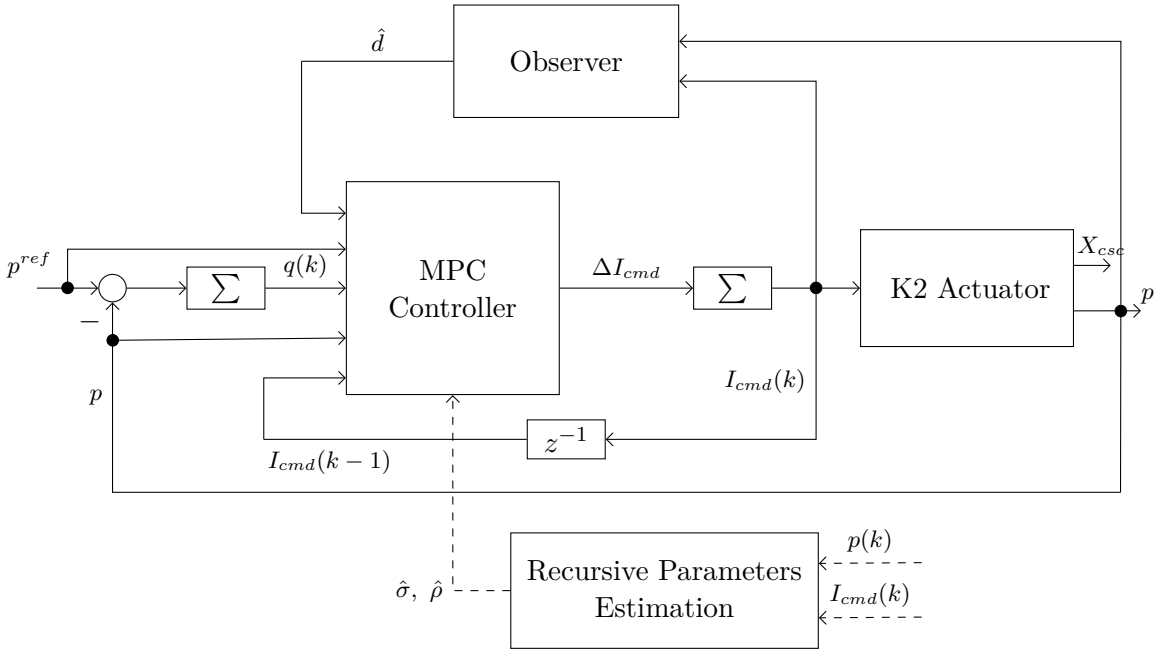

Figure 4.13: Adaptive MPC inner pressure control architecture for K2 Actuator.

In this different scenario and according to [1], the relationship between the command input and the output pressure is assumed to be modelled by a first order transfer

function G_{Ip} with pole σ and gain ρ. Again, a non linearity model compensation is exploited by performing a state observer based feed forward action. The current variation rate ΔI_{cmd} has been adopted as optimization variable and a suitable variation rate constraint has been imposed. The chosen controller parameters settings are listed in (4.10).

Once designed the pressure inner loop, the LQ controller 4.7 and the Virtual Sensor have been included in the outer loop of the overall nested control architecture.
Simulations results, in the presence of an Up shift position reference profile, are illustrated in Figure 4.14.

$$\begin{aligned} R &= 1 \\ Q_y &= 1.2 \cdot 10^2 \\ Q_q &= 1 \\ H_p &= 10 \\ |\Delta I_{cmd}^{max}| &= 500 \ \tfrac{mA}{T_s} \\ \gamma &= 0.01 \\ \theta(0) &= [-9.83 \cdot 10^{-3} \quad 5.3 \cdot 10^{-2}] \end{aligned} \tag{4.10}$$

where $\theta(0) = [\sigma_0 \ \rho_0]$ is the initial model parameters vector as in [1].

The reported plots highlight the effectiveness of the just presented nested control architecture. In comparison with the previous approaches, each of the three Virtual Sensor modelling strategies presents quite good results. More precisely, a smoother behaviour of the real output position along with significant improvements in terms of response speed have been obtained, especially when the Hammerstein-Wiener Virtual Sensor is included in the outer loop.

However, as long as the Virtual Sensor based control is concerned, and with the aim of providing a general consideration on the proposed control strategies, an overall performance resume is presented in the following section.

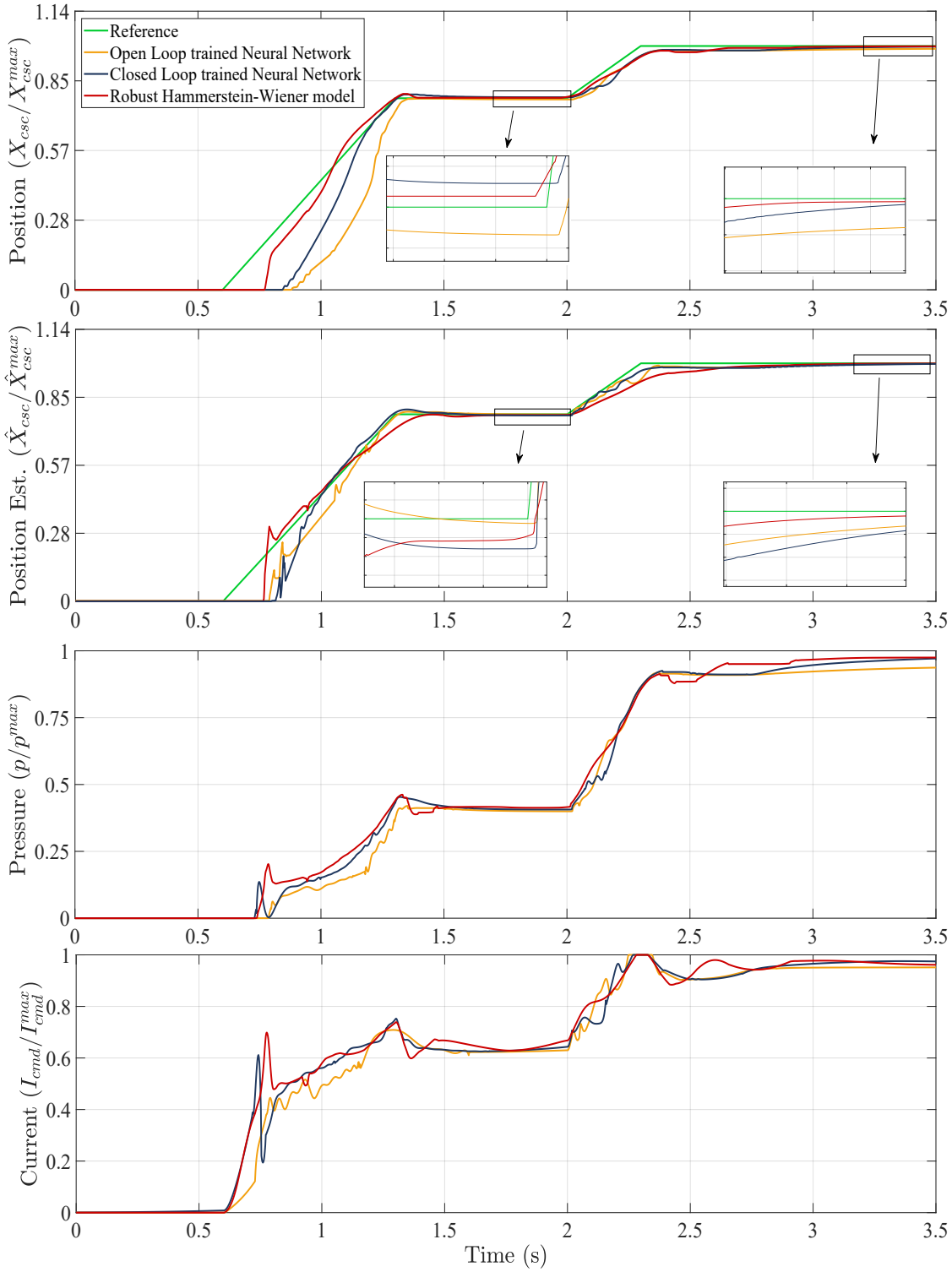

Figure 4.14: Position response, position estimate, output pressure, command input with Up shift reference profile adopting a Virtual Sensor based nested control (LQ outer loop - MPC inner loop).

4.4 Final Comparisons and Results

In the previous section two modelling strategies for the Virtual Sensor development have been described. Their inclusion in different control architectures has been analysed by means of simulations results. Besides, control performances have been discussed for each of the adopted methodologies by taking into account not only the real position response behaviour, but also the position estimate accuracy along with the output pressure and command input signals.

In the following, dealing with the Virtual Sensor based approaches, and considering both the MPC and LQR position controllers developed in Chapter 2 and Chapter 3, the K2 Actuator position response is studied. This is to investigate and compare the achievable control performances when a Virtual Sensor, and not a real position sensor, provides the controller the needed samples of the K2 Actuator output position.
Specifically, regarding the previous section results, only the control architectures, and the associated Virtual Sensor, that provide satisfactory performances in terms of real position output behaviour, are selected to be compared.

The first plot, reported in Figure 4.15, shows the command input and the position output behaviours when either a real position sensor or a Virtual Sensor is employed in both the single and nested control architectures. The LQ approach is here investigated and the closed loop trained Neural Network is assumed to be the best Virtual Sensor modelling strategy for this position control purpose.
These simulation results outline the fact that an inner 1dof pressure controller guarantees an effortless control action and a good tracking behaviour of the K2 Actuator output position. Nevertheless, these improvements in terms of smoothness are obtained at the cost of slowing down the system response.

In Figure 4.16 the same scenarios as before are considered but, in this case, MPC techniques are inspected and the Hammerstein-Wiener Virtual Sensor is chosen to be the more performing one with the nested control architecture.
As confirmed by the reported plots, if the K2 Actuator output pressure is controlled by a MPC internal loop, significant advancements of the response speed as well as of the input current smoothness are appreciable with the only drawback of a slightly larger position overshoot.

Generally speaking, and with reference to the just discussed simulation results, it is evident that the inclusion of the Virtual Sensor in the control architecture entails some oscillations in the command input. These are responsible, particularly with ramp

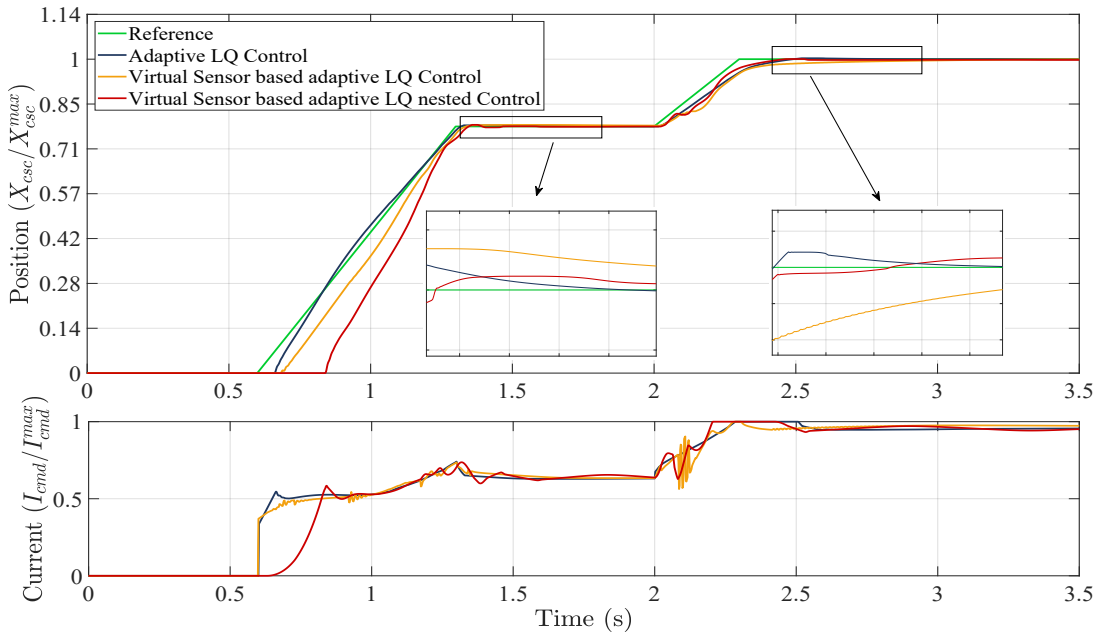

Figure 4.15: Position response and Input Current with an Up shift reference profile adopting Adaptive LQ techniques when either a real position sensor or a Virtual Sensor are employed.

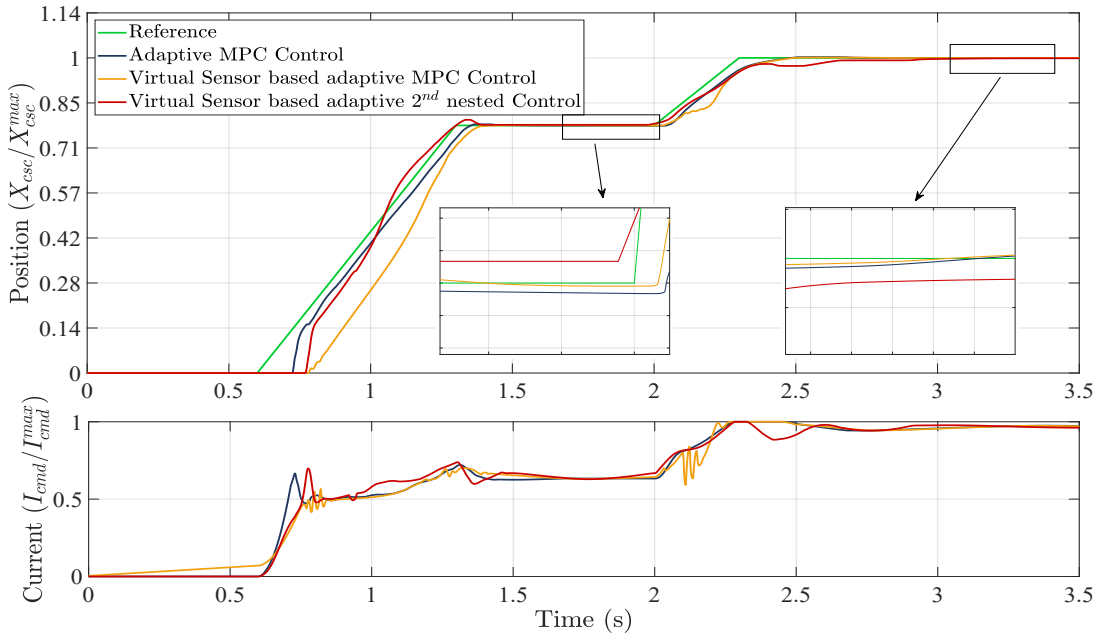

Figure 4.16: Position response and Input Current with an Up shift reference profile adopting Adaptive MPC techniques when either a real position sensor or a Virtual Sensor are employed.

position trajectories, of a delayed or slightly biased reference tracking.
Enhancements about these aspects are proposed by means of different nested approaches that, however, represent a trade-off between the overall control objectives.

However, as expected and confirmed by the above plots, when the DDCT system can be equipped with a real sensor, the position control of the even gear actuator is naturally far more effective.

Chapter 5

Conclusions

This thesis has dealt with the even gear actuator control of a dry dual clutch transmission system. More precisely, the aim of the project has been to design a controller that, using the clutch position measurement, is able to guarantee a smooth reference tracking and a continue torque transmission.

With the purpose of improving driving comfort, different control architectures, related to the optimal control methodologies, have been exploited. Moreover, due to the lack of an a priory model for the considered K2 Actuator, identification procedures, performed on the basis of the given experimental data, have been necessary to evaluate a mathematical relationship between the involved variables.

Specifically, identification results have shown that the system dynamic and steady state properties are strictly related to the working region. For this reason, a recursive least square estimation algorithm has been developed so that all the proposed control strategies are designed on the basis of a real time varying state space model representation.

First of all, an Adaptive Model Predictive Control approach has been considered. Physical constraints and multiple control objectives have been handled through a convenient customization of the cost function and the control problem formulation. A non linear effect has caused the need to perform an implicit feed forward action by including, among the state variables, an unmeasurable disturbance whose estimate is provided by a suitable state observer.

Different position reference profiles, that reasonably represent the K2 Actuator working situations, have been considered.

The Adaptive MPC controller parameters setting, whose choice has been reasoned by extensive tuning simulations results, ensures quite satisfactory performances in terms of steady state tracking and rise time. The proposed MPC control action is able to guarantee an accurate tracking even if the reference profile is characterized by sudden set point variations. More precisely, the minimum tracking error is obtained in the presence of constant references while, with ramp trajectories, a small offset occurs in the position response.
Motivated by the huge variability of the system behaviour with respect to the working point, a real time adjustment of the controller design parameters have been proposed by means of a scheduling algorithm. Thanks to this approach, significant improvements, especially in terms of position overshoots, have been achieved.
However, due to the long prediction horizon, such an Adaptive MPC method need quite high resources to compute in real time the optimal control action.

Therefore, with the aim of improving computational aspects, an Adaptive Linear Quadratic Control strategy has been exploited. An analytic iterative method has been developed to overcame the problem of solving, at each sampling time, the infinite horizon Discrete Time Riccati equation that is related to the proposed LQ approach.
As for the MPC, a wide spectrum of working situations has been considered to test the effectiveness of the developed control architecture and a suitable scheduling algorithm has been proposed to ensure the same level of control performances in all the position regions.

Simulations results have shown that also an adaptive LQ control action, resulting from an unconstrained optimization problem, is able to guarantee a fast position response without significantly increasing position overshoots and steady state errors. This is due to the affine static state feedback form of the LQ control action that, thanks to the explicit feed forward loop, ensures satisfactory performances in terms of rise time and reference tracking.

It is worth to highlight that, both the MPC and the LQ controllers have been developed assuming the availability of a real position sensor in the considered Dry Dual Clutch Transmission System.
In this regard, the last Chapter of this thesis work deals with the lack of a real sensor by including a Virtual Sensor into the control architecture. Extensive simulations and several comparisons have proved that the best trade-off between a smooth input current and a fast position response is obtained by means of a nested control architecture designed with an inner MPC pressure loop and an outer LQ position loop.

In conclusion, considering the overall DDCT system, the clutch torque effectiveness is strictly related to the gear actuator control accuracy, especially in high position ranges. Therefore, the position response smoothness is of paramount importance in regard to the driving comfort.
With this in mind, among the different proposed approaches for the position control of the K2 Actuator, the adaptive LQ showed outstanding results in terms of achieved performances and computational effort. Besides, it offers the possibility of an effective deployment in a real time platform thanks to the closed form of the LQ control action.

Future Work

Starting from the results presented in this thesis project, further investigations can be carried out.

The position estimate accuracy can be improved by designing the Hammerstein-Wiener Virtual Sensor on the basis of the proper closed loop data. Hence, the proposed LQ controller can be implemented on the transmission control unit in order to test the achievable performances on the real vehicle.

Another interesting aspect that deserves a more thorough study, is to develop a Virtual Sensor based MPC controller by accounting at first the inclusion of the Virtual Sensor into the control problem formulation.

Bibliography

[1] Talal Almutaz Almansi Abdalla. Modelling and identification of a dual clutch actuator. Master's thesis, Politecnico di Torino, April 2018.

[2] J.B. Moore B.D.O. Anderson. Optimal Control. Prentice-Hall, 1 edition, 1989.

[3] F. Blanchini. Set invariance in control a survey. In Automatica, number 35(11), page 1747 1768. November 1999.

[4] S. Boyd and L. Vandenberghe. Convex optimization. Cambridge University Press, 2004.

[5] H. Fan C. Chiang and W. Lin. A structured doubling algorithm for discretetime algebraic riccati equations with singular control weighting matrices. In Taiwanese Journal Math, number 14(3A), page 933954. 2010.

[6] R.Zanasi C.Bonivento, C.Melchiorri. Sistemi di Controllo digitale. Progetto Leonardo - Esculapio, 1995.

[7] B.Meini D.B.Bini, B.Iannazzo. Numerical Solution of Algebraic Riccati Equation. SIAM, 2012.

[8] C.Bordons E.F.Camacho. Model Predictive Control. Springer, 2 edition, 2007.

[9] Beccani Eva. Virtual sensor feedback control of a clutch actuator. Master's thesis, Politecnico di Torino, December 2018.

[10] M. Morari F. Borrelli, A. Bemporad. Predictive Control for linear and hybrid systems. Cambridge, 2011.

[11] H. J. Ferreau. An online active set strategy for fast solution of parametric quadratic programs with applications to predictive engine control. Technical report, University of Heidelberg, 2006.

[12] V.L. Syrmos F.L. Lewis, D.L. Vrabie. Optimal Control. Wiley, 3 edition, 2012.

[13] L. El Ghaoui G. C. Calafiore. Optimization Models. Cambridge University Press, 2014.

[14] S. Haykin. Neural Networks: a Comprehensive Foundation. Pearson Education, 2005.

[15] M. Herceg, M. Kvasnica, C.N. Jones, and M. Morari. Multi-Parametric Toolbox 3.0. In Proc. of the European Control Conference, pages 502–510, Zürich, Switzerland, July 17–19 2013. https://www.mpt3.org/pmwiki.php.

[16] H.K.Wimmer. Normal forms of symplectic pencils and the discrete-time algebraic riccati equation. Mathematisches Znstitut, D-8700 Würzburg, Germany, August 1990.

[17] M. MSaad A. Karimi I. D. Landau, R. Lozano. Adaptive Control Algorithms, Analysis and Applications. Springer, 2 edition, 2011.

[18] R.A. Flauzino L.H. Barocci S. dos Reis Alves I.N. da Silva, D.H. Spatti. Artificial Neural Networks: a Practical Course. Springer, 2017.

[19] David Q. Mayne James B. Rawlings. Model Predictive Control: Theory and Design. Nob Hill, 2013.

[20] E. C. Kerrigan. Robust Constraints Satisfaction: Invariant Sets and Predictive Control. PhD thesis, University of Cambridge,Cambridge, England, 2000.

[21] L. Ljung. Black-box models from input-output measurements. IEEE, 2001.

[22] R. Milman and E.J. Davison. A fast mpc algorithm using nonfeasible active set methods. In Journal of Optimization Theory and Applications, number 139(3), page 591 616. 2008.

[23] Kannan M.Moudgalya. Digital Control. Wiley, 2007.

[24] R.Rojas. Neural Networks: a Systematic Introduction. Springer, 1996.

[25] P. Colaneri S. Bittanti. Periodic Systems Filtering and Control. Springer, 2008.

[26] T. Söderström. Discrete-time Stochastic Systems Estimation and Control. Springer, 2002.

[27] B.Hassibi T.Kailath, A.H.Sayed. Linear Estimation. Prentice Hall, 2000.

[28] Y.Wang and S. Boyd. Fast model predictive control using online optimization. control systems technology. In Control Systems Technology, IEEE Transactions on, number 18(2), pages 267 – 278. 2010.