Politecnico di Torino

DEPARTMENT OF CONTROL AND COMPUTER ENGINEERING
(DAUIN)

MASTER OF SCIENCE IN MECHATRONIC ENGINEERING
Master Degree Thesis

MODEL PREDICTIVE CONTROL OF A GEAR ACTUATOR FOR DUAL CLUTCH TRANSMISSION SYSTEMS

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

Author:
Ahmed Hassan Abdelaziz Elhaddad

APRIL 2018

Acknowledgement

I would like to express my deep gratitude to Prof. Vito Cerone, Prof. Massimo Canale and Prof. Diego Regruto for their guidance, gracious support and encouragement through out the thesis work. Their comprehensive advice allowed me to achieve these results.

My grateful thanks are also extended to Valentino Razza for his generous support in resolving challenging problems.

My sincere thanks to Centro Ricerche Fiat team for providing me such an opportunity to join this thesis. I would like to express my strong appreciation to Engineers; Emanuel Corigliano, Simone Baliva and Sabrina Spagnolo for their extensive support. It was a very fruitful and an invaluable experience to work with this team.

Finally, I would like to thank my family and friends for supporting me spiritually and being by my side during this journey.

Abstract

Nowadays, Automotive manufacturers have devoted their researches to enhance the efficiency of the transmissions with a mind to the future taking into consideration ride quality and fuel conservation at the same time. Automated Manual Transmission (AMT) has been introduced before offering a good compromise between Manual Transmission (MT) and Automatic Transmission(AT) offering high transmission efficiency with less fuel consumption. The AMT consists of a clutch with added-on control unit that control the gear shifting operations. However, the main problem of AMT is the torque interruption that effects the ride quality.

A remarkable result of transmissions development is the Dual Dry Clutch Transmission (DDCT). The DDCT basically consists of two independent transmission lines one is devoted for the shifting of the odd gears while the other is responsible for the even gears. Indeed, the theory of using two independent gearboxes can result in elimination of the torque interruption improving traction without interruption.

According to the literature, the problems appeared with the invention of DDCT are the management of the dual clutch in the launch phase and the gearshift phase. The aim of this thesis, developed in collaboration with **Centro Ricerche Fiat (CRF)** in Torino, Italy, is to design a controller for the even clutch actuator (K2-actuator) to track different pressure trajectories during the gear shifting phase. The purpose of controlling the K2-actuator is to guarantee smoothness during the gear shifting process without any torque interruption that can lead to discomfort to the driver.

This Thesis consists of two main phases. The first phase is to identify the K2-actuator by using system identification methodology to build a mathematical model that represents the dynamics of the system since there is not any priory model available to represent the dynamics K2-actuator. The second phase is to use the identified model in controlling the pressure to guarantee a good compromise between tracking the predefined pressure trajectories. Thus, various control architectures are considered to control the K2-actuator starting from Multiple Model Predictive Controllers (MPC) based on Multi models, Adaptive Model Predictive Control (MPC) and finally Adaptive Linear Quadratic Regulator (LQR).

MATLAB and SIMULINK are exploited to identify and control the K2-actuator consequently. Moreover, extensive simulations are carried out to decide which control architecture is better to be deployed on the Electronic Control Unit (ECU) of the DDCT.

Contents

List of Figures

1 The Dual Dry Clutch Transmission (DDCT)

1.1 Introduction

In automotive vehicles, transmission system is a fundamental part that converts the power from the engine into vehicle traction with the highest efficiency possible that suits the vehicle, the surface, the driver and the environment. The transmission has to be designed guaranteeing a good balance between the number of speeds, climbing performance, acceleration and fuel consumption of the vehicle.

The most common widely used transmissions for the passenger cars are the Manual Transmission (MT) and the Automatic Transmission (AT). In the USA, the favorable design for Passengers cars is the conventional Automatic Transmission (AT) where 75 to 85 % of the passengers cars are equipped by the Automatic Transmission system. On the other hand, the Europeans prefer to use the Manual Transmission design where around 85 % of the cars are found to be equipped by the Manual Transmission (MT) design [4].

Each design has an advantage over the other. For instance, The Manual Transmission ensures high efficiency, low cost and full control of the drive, while on the other hand the Automatic Transmission guarantees smooth transition during shifting between the gears. This improvement comes over the price of the fuel consumption.

In this context, in order to make a good compromise between the Manual Transmission(MT) and the Automatic Transmission (AT), the Automated Manual Transmission(AMT) has been introduced combining the features of the Manual and the Automatic Transmissions together ensuring high transmission efficiency in terms of fuel consumption and the add-on costs. However, the drawback of the Automated Manual Transmission (AMT) that torque interruption exists during the gear shifting which causes bumpy feeling during changing the speed of the car. Thus, The Dual Clutch Transmission (DCT) has been developed to attenuate the torque interruption.

1.1.1 History of Dual Clutch Transmission (DCT)

The Dual Clutch Transmission (DCT) had a great impact on the Automotive industry over the last few decades in terms of enhancing the performance of traction of an Automobile car and minimizing waste energy. The Dual Clutch Transmission was invented by Adolphe Kgresse a french military engineer before starting the World war II, it was never deployed on an automobile though until Harry Webster a British automotive Engineer came and developed the design of the DCT in 1980 at Automotive product, Leamington Spa to be deployed on Ford Fiesta Mk1, Ford Ranger, and Peugeot 205.

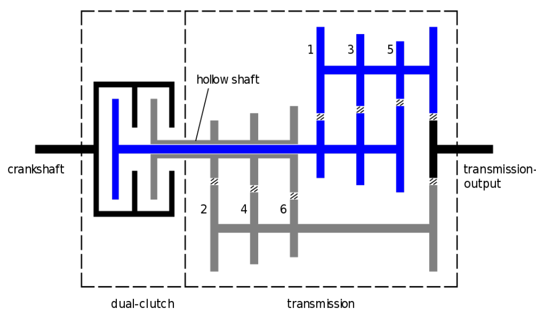


Figure 1: Dual Clutch schematic Diagram

1.1.2 DCT principal of operation

The DCT has two conventional clutches, one is dedicated for the switching of the odd gears while the other is dedicated for the even gears. The DCT consists essentially of two transmissions, each one with its own clutch working in Parallel. The gear shifting is performed automatically by gradual but very rapid switching between the two clutches. The power of the DDCT that it put an end to the usage of the Torque converter that is used in the traditional transmissions. Unlike Manual or Automatic Transmissions which momentary disconnect the Engine from the Wheels while changing gears which cause traction interruption , the DCT maintain traction constant during gear shifting which guarantee excellent power transmission smoothness and Efficiency. Clutches and Gears are actuated by a compact super fast electrohydraulic system. This electrohydraulic system is responsible to convey pressurized oil by means of solenoid valves that actuates the fork that selects the speeds. The Pressure required for actuation is generated by an electrical pump which is only operated when needed to minimize the waste energy.

1.1.3 Types of Dual Clutch Transmission

DCT can be found in two forms either . The first form is wet dual clutch transmission that uses oil as a coolant during the meshing of the gears. The Wet dual clutch transmission is quite common in Automotive cars that its engines produces high torque. The second type is the Dual Dry Clutch Transmission (DDCT) which basically uses the friction to mesh between the gears where the DDCT can be found with Automobile that has low torque engines. The advantage of using a dry clutch over wet clutch that it increases fuel efficiency by decreasing the pumping losses of the hydraulic fluid in the transmission housing, however the wet clutch is better in terms of torque efficiency. This thesis reports the Dual Dry Clutch Transmission (DDCT) invented by Fiat Power-Train Technologies that belongs to the C635 Transmission Family. The Fiat's C635 DDCT was released in 2010 with front wheel drive, all wheel drive and manual versions. It can supply torque up to 350 $N.m$ and weighs around 82 $K.g$ including the Electronic Control Unit and the oil of the electrohydraulic system.

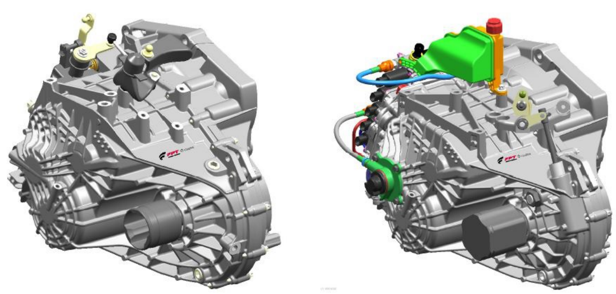
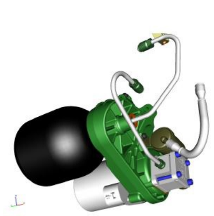


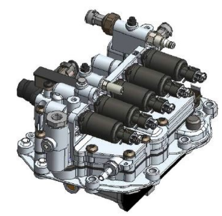
Figure 2: C635 MT and DDCT versions

1.1.4 Electro-hydraulic Actuation System

The C635 DDCT clutches and gear shifting mechanisms are electro-hydraulically actuated through a dedicated, sealed hydraulic oil circuit. The Actuation system consists of a hydraulic power unit (PU) that composed of a pump and an accumulator and the actuation module (CAM,figure 3). The CAM module contains the control solenoid valves, gear shift actuators and sensors. The hydraulic oil used (Cspeed) has the same characteristics of the current FPTs Automated Manual Transmissions (AMTs).



(a)



(b)

Figure 3: (a) Hydraulic Power Unit (PU), (b) Complete Actuation Model (CAM)

The clutch and gear actuation module (CAM) consists of:

- Four distinct double action cylinder operating the gear engagement forks.

- One shifter spool which selects the piston to be actuated.

- Five solenoid valves of which 4 are Proportional Pressure Valve (PPV) and one as Flow Pressure Valve (QPV).

The first PPV valve is responsible to control the spool that selects the needed doubling acting cylinder for gear engagement. The two other PPVs actuates the selected double action cylinder. The fourth PPV is used to control the K2-clutch CSC for the even gears while the QPV used to control the the distance of the K1-clutch devoted for the odd gears. All solenoid valves are direct derivatives of those currently used in FPTs AMT systems and, therefore, employ well proven technology and guarantee robustness. The Actuation Module also comprises 5 non-contact linear position sensors, one for each shifting piston and one for the shifter spool, as well as two speed sensors reading the speed of the two primary shafts. Finally, one pressure sensor is used for the control of the K2 clutch and one for the system pressure monitoring and control.

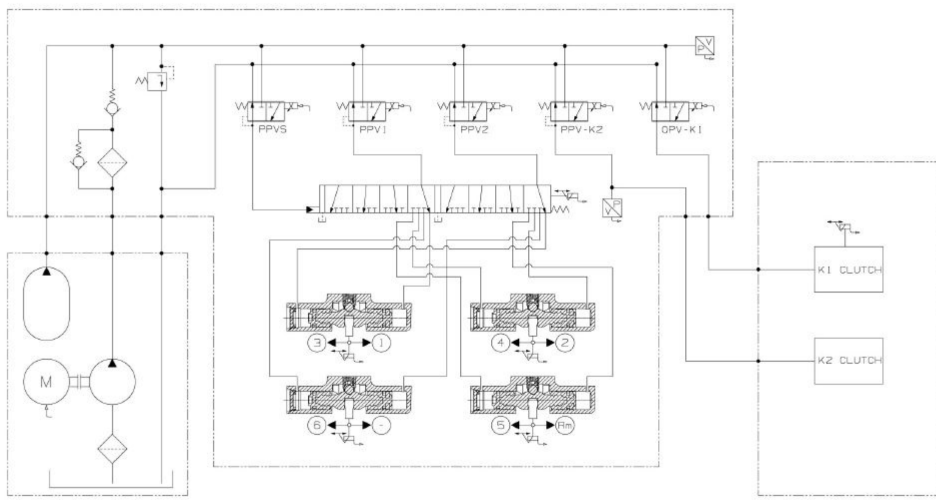


Figure 4: Complete Actuation System (CAS)

1.1.5 C635 DDCT control unit

The C635 DDCT control strategies have been developed by FPT and run in a multitasking environment in order to meet the frequency requirements of the control loops that they implement, preserving at the same time the Main Micro Controller resources. They can be grouped as described below: Actuator controls: Based on the experience of many AMT systems in production, the actuator control strategies exploit the high performance attainable with electro-hydraulic actuators. The main control strategies concern:

- **Engagement actuators control**: based on a force/speed control concept, the desired profiles are realized by commanding the two relevant PPVs one against the other.
- **Shifter (selector) control**: hydraulic power to the required engagement actuator is guaranteed by a fast and precise control of the shifter. The related PPV is commanded to push the shifter piston against a spring in order to reach one of the four desired positions. Continuous monitoring of the selector position guarantees the required safety level.
- **Odd gears clutch controls**: the normally closed clutch (K1) is controlled by a position closed loop. This is the clutch of the first and of

the reverse gear; therefore, this control strategy is essential also for the vehicle starting performance.

- **Even gears clutch**: the normally open clutch (K2) is controlled in force with a pressure feedback signal delivered by one of the CAM sensors.

- **Self-tuning controls**: FPTs DDCT control system has, clearly, many self-tuning controls in order to compensate for the various parameters drift and to adapt the same high-level calibrations to all vehicles. The main self-tuning control algorithms concern the conversion of the requested clutch transmitted torque to K1 position and K2 pressure.

- **Launch and gear shift strategies**: The C635 DDCT implements various driving modes, depending on the desired performance and Brand/ OEM requirements, both in manual and in automatic mode. Three different modes of shift patterns in automatic and two different ones in manual (tip) mode are contemplated and are accomplished also by specific control strategies and calibrations on the engine side. Vehicle creeping on brake release is also implemented, together with the braking systems hill holding functions.

1.2 K2-actuator

The aim of this thesis is to control the pressure applied to the even gears clutch by controlling the dual clutch K2-actuator through designing a Model predictive control to track a predefined pressure reference signals. Pressure is measured by a pressure sensor applying a feedback controller signal to the controller which is crucial in order to obtain the control objectives.

1.2.1 Proportional Pressure Control Valve

Proportional Pressure Control valves (PPV) are easily controlled using a solenoid coil. A valve is considered open when the solenoid is supplied with electric current and closed when the coil is de-energized. The proportional valve produces an output (direction, pressure, flow) that is proportional to the control input(electric current). The PPV can be utilized using different ways of control techniques depending on the type of the application. PPV can be controlled using open loop control system where no feedback from the output of the valve is taken into consideration in the control circuit. On the contrary, PPV can be controlled using an Electronic control unit that computes the control input providing high accuracy of control over the pressure. The feedback is received from a pressure transducer providing high accuracy. This allows the spool of the PPV to be stopped at intermediate positions according to the input electric current applied to the solenoid coil.

1.2.2 Pressure Control and Dead Zone

Most PPVs use varying electric current as a control variable instead of varying voltage. The reason behind that is to avoid the influence of the temperature variation on the electric current since the resistance of the solenoid coil changes within the fluctuation of the temperature. Thus, current control system is devoted to eliminate this problem.

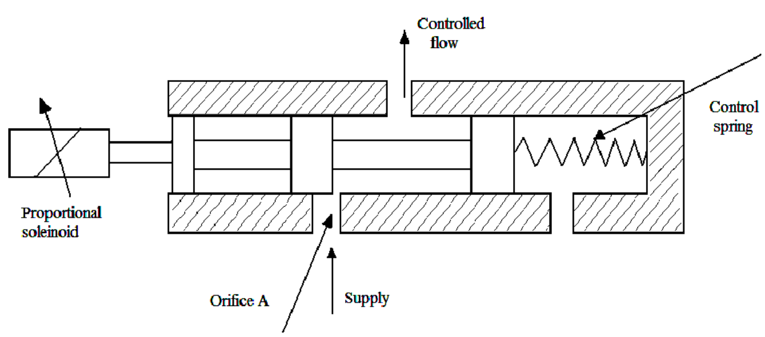


Figure 5: Schematic section of a PPV

It is possible to control the pressure by controlling the flow of the hydraulic fluid through the orifice of the PPV. By applying the force to a compression spring, its deflection can be controlled. If the spool in a valve (as shown in figure 5) is acted on by a spring at one end and a proportional solenoid on the other, the orifice size can be varied along with the control current. The flow from the valve is proportional to the current flowing through the solenoid. Because of the difficulties in producing a zero lap spool, overlapped spools are used in proportional spool valves. This means that the spool has to move a distance equal to the overlap before any flow occurs through the valve, giving rise to a dead zone (as shown in figure 6).

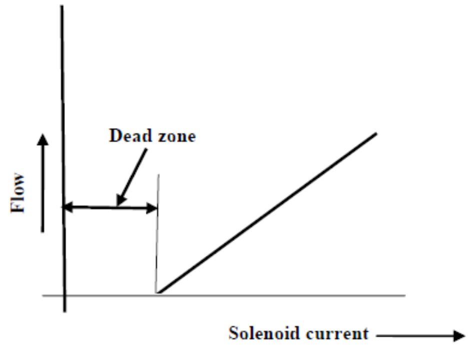


Figure 6: Flow current characteristics of a PPV

1.2.3 Control objectives and performance requirements

The design specifications requested by Centro Ricerche Fiat (CRF) is to track the pressure at different steps references with the following performance requirements:

- overshoot: $\hat{s} \leq 10\%$
- rising time: $T_r \leq 120\ ms$
- settling time: $T_{settling5\%} \leq 120\ ms$
- steady steady error for constant reference: $|e_r^\infty| = 0$
- Input current: 0% of $U^{max} \leq u \leq 100\%$ of U^{max}
- pressure: 0% of $P^{max} \leq P \leq 100\%$ of P^{max}

Where P^{max} and U^{max} in the following, will denote the maximum values of respectively the current and the pressure operating ranges.

Yet, the greatest challenge to reach the control objective is to identify first the K2-actuator since there is no any prior model available to represent the system. In control system a good mathematical model that represents the dynamics system is crucial to be able to meet the performance requirements. Hence, the K2-actuator will be considered as black box and system identification methodology is going to be exploited to build a mathematical model that properly represents the dynamics of the system.

2 Model Predictive Control for DDCT

This chapter is devoted to illustrate the main conceptual theory behind Model Predictive Control (MPC) as well as the formulation to track a desired reference signal with constraints applied on the system to achieve the performance requirements defined in (1.2.3).

Nowadays, Model-Based Predictive Control (MPC) is the most popular efficient controller that has a great impact over a wide range of applications in different engineering fields especially for chemical industries. The power of MPC arises due the efficiency in handling the constraints with achieving high performance in the mean time. These features are conditioned by explicitly considering the model of the system to obtain the control action as a result of of a constrained optimization problem.

Fundamentally, The MPC architecture uses the system's model to predict its dynamic behavior over a finite prediction steps and to solve a constrained optimization problem based on the predicted behavior to compute the optimal control action at the current time. The computation for the optimal control action can be quite intensive for large prediction horizon. Nevertheless, the great impact of technology on computers and microprocessors allowed Model predictive control to be applied in faster system like automotive applications as well as chemical applications. The MPC architecture is composed of (refer to figure 7) :

- Optimizer
- Cost function and Constraints
- Prediction model
- The controlled System

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

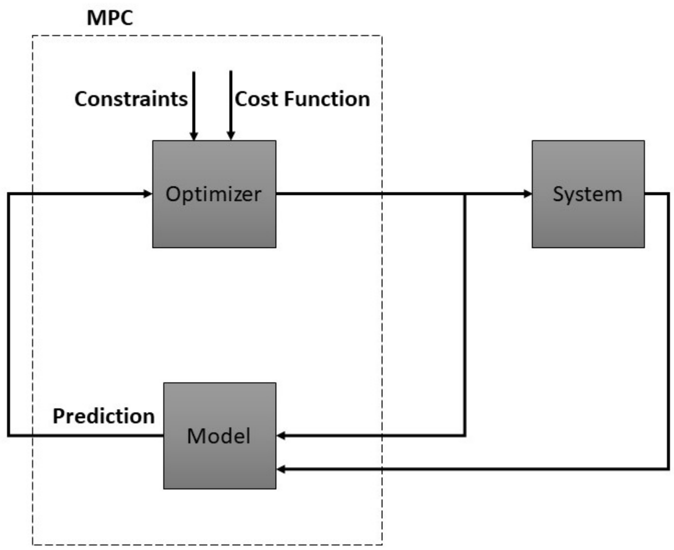


Figure 7: Generic Scheme of MPC

The merit of exploiting MPC architecture can summarized as follows:

- The possibility to explicitly include constraints on the control input and the states variables.
- The capability to handle multi-variable control problem.
- The formulation of the optimization problem allows to trade off between different objectives by tuning some critical parameters.

2.1 Non-Linear time invariant system

2.1.1 Prediction for Non-Linear Time invariant System

The prediction model is represented by a discrete time, non linear, time invariant state space representation of the following form:

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

where:

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

The prediction of the model takes place by considering the evaluation of the states formulated in (2.1) starting from a time instant k over a finite prediction horizon H_p. In this context, all state variables $x(k)$ are assumed to measurable.

2.1.2 Cost function and optimization problem for Non-Linear Time Invariant system

The MPC is based on solving a constrained optimization problem that minimizes a cost function over a defined finite prediction horizon H_p. The cost function can be expressed as:

$$J(x(k|k), U(k)) = \sum_{i=0}^{H_p-1} L(x(k+i|k), u(k+i|k)) + \Phi(x(k+H_p|k)). \quad (2.2)$$

where:

- H_p is the prediction horizon interval
- $x(k|k)$= States measurements at current time k.
- $x(k+i|k)$ is the i^{th} step ahead state prediction, obtained using model(2.1)
- $U(k) = \begin{bmatrix} U(k|k) & U(k+1|k) & \dots & U(k+H_p-1|k) \end{bmatrix}^T$ is the command input sequence to be optimized.
- $L(.)$= Per-stage weighting function.
- $\Phi(.)$= Terminal state weighting function.

$L(.)$ and $\Phi(.)$ are considered as design parameters to achieve the required performance defined in section.

The general formulation of the optimal control can be expressed as :

$$
\begin{aligned}
U^o = arg \quad \underset{U}{min} \quad J(U, x(k)) \\
s.t \\
x(k+1) = f(x(k), u(k)) \\
x(k+i|k) \in \mathcal{X}, i = 1, \cdots, H_p - 1 \\
u(k+i|k) \in \mathcal{U}, i = 0, \cdots, H_p - 1 \\
x(k+i|k) \in \mathcal{X}_f
\end{aligned}
\tag{2.3}
$$

where:

- $\mathcal{X}$ and $\mathcal{U}$ are the input and state constraints sets respectively that are assumed to be convex.

- $\mathcal{X}_f$ is the terminal constrain set introduced in the optimization problem to ensure asymptotic stability (more details in [12])

- $U^o(k) = \begin{bmatrix} U^o(k|k) & U^o(k+1|k) & \ldots & U^o(k+H_p-1|k) \end{bmatrix}^T$

2.2 Linear time invariant system

2.2.1 Prediction for linear time invariant system

similarly, the prediction model (2.1) can be formulated as well as for a Linear Time Invariant (LTI) discrete time system, described by the following state space representation:

$$
\begin{aligned}
x(k+1) = Ax(k) + Bu(k) \\
y(k) = Cx(k)
\end{aligned}
\tag{2.4}
$$

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

Hence, the i^{th} step ahead prediction state $x(k+i|k)$ for a LTI system is given by:

$$
\begin{aligned}
x(k+i|k) &= A^i x(k|k) + A^{i-1}Bu(k|k) + A^{i-2}Bu(k+1|k) + \cdots + Bu(k+i-1|k). \\
&= A^i x(k|k) + \sum_{j=0}^{i-1} A^{i-j-1}Bu(k+j|k)
\end{aligned}
\tag{2.5}
$$

Hence, the prediction model (2.5) depends only on the current state $x(k|k)$ and on the control sequence $U(k) = \begin{bmatrix} U(k|k) & U(k+1|k) & \dots & U(k+H_p-1|k) \end{bmatrix}^T$

2.2.2 Input and states Constraints

Control Input and state constraints $\mathcal{U}$ and $\mathcal{X}$ are usually modeled as sets of inequalities.
$\mathcal{U}$ is often chosen to introduce input actuator saturation and slew rate constraints. These constrains can be formulated as:

$$\begin{aligned} U^{min} \leq u(k+i|k) \leq U^{max}, i = 1, 2, \cdots, H_p - 1 \\ \Delta U^{min} \leq \Delta u(k+i|k) \leq \Delta U^{max}, i = 1, 2, \cdots, H_p - 1 \end{aligned} \tag{2.6}$$

where:

- U^{min} is the constraint vector associated with the minimum control input.
- U^{max} is the constraint vector associated with maximum control input.
- ΔU^{min} is the constraint vector associated with the minimum control input increment.
- ΔU^{max} is the constraint vector associated with the maximum control input increment.

In order to clarify more this context, for simplicity $H_p = 2$ is considered. Hence, the control command input $U(k) = \begin{bmatrix} u(k|k) & u(k+1|k) \end{bmatrix}^T$ is subjected to the following control input constraints:

$$\begin{aligned} U^{min} \leq u(k+i|k) \leq U^{max} \\ U^{min} \leq u(k+1|k) \leq U^{max} \end{aligned} \tag{2.7}$$

After some rearrangements , Control input constraints (2.7) is formulated as:

$$
\begin{aligned}
\begin{bmatrix}1 & 0\\0 & 1\end{bmatrix} & \begin{bmatrix}u(k|k)\\u(k+1|k)\end{bmatrix} & \leq \begin{bmatrix}u^{max}\\u^{max}\end{bmatrix} \\
\begin{bmatrix}-1 & 0\\0 & -1\end{bmatrix} & \begin{bmatrix}u(k|k)\\u(k+1|k)\end{bmatrix} & \leq \begin{bmatrix}u^{min}\\u^{min}\end{bmatrix} \\
\Rightarrow \underbrace{\begin{bmatrix}I\\-I\end{bmatrix}}_{L_U} & \begin{bmatrix}u(k|k)\\u(k+1|k)\end{bmatrix} & \leq \underbrace{\begin{bmatrix}u^{max}\\u^{max}\\u^{min}\\u^{min}\end{bmatrix}}_{W_u} \Rightarrow L_u U(k) \quad \leq \quad W_u
\end{aligned}
\tag{2.8}
$$

Moreover, through some manipulations, the constrains associated with the control input increment can be expressed as:

$$L_{\Delta u} U(k) \quad \leq \quad W_{\Delta u} \tag{2.9}$$

By combing constrains 2.7 and 2.9, the linear inequality constraints of the control input can be formulated as:

$$L_U U(k) \quad \leq \quad W_U \tag{2.10}$$

where L_U and W_U are matrices that represents the linear constrain on the control input.

Further, the state constraints $\mathcal{X}$ can be taken into consideration during the prediction of the states. This is crucial to limit the response of state variables and/or output variables to not exceed a predefined value. For instance, to limit the overshoot during the transient phase to satisfy the performance requirements.

likewise, the state variables that subjected to state constraints can be formulated as :

$$x^{min} \leq x(k+i|k) \leq x^{max}, i = k, \cdots, H_p \tag{2.11}$$

By considering the same previous example (2.7) such that $H_p = 2$, the predicted state that subjected to constraints can be written as :

$$
\begin{aligned}
L_{x1} x(k+1|k) \leq W_{x1} \\
L_{x2} x(k+2|k) \leq W_{x2}
\end{aligned}
\tag{2.12}
$$

Further, by substituting the step ahead prediction state (2.5) in both, we have:

$$\begin{aligned} L_{x1}(Ax(k|k) + Bu(k|k)) &\leq W_{x1} \\ L_{x2}(A^2x(k|k) + ABu(k|k) + Bu(k+1|k)) &\leq W_{x2} \end{aligned} \tag{2.13}$$

By applying some rearrangements , (2.13) can be written as:

$$L_x U(k) \leq W_x \tag{2.14}$$

where:

$$\begin{aligned} Lx &= \begin{bmatrix} L_{x1} & 0 \\ 0 & L_{x2} \end{bmatrix} \begin{bmatrix} B & 0 \\ AB & B \end{bmatrix} \\ Wx &= \begin{bmatrix} -L_{x1} & 0 \\ 0 & -L_{x2} \end{bmatrix} \begin{bmatrix} A \\ A^2 \end{bmatrix} x(k|k) + \begin{bmatrix} W_{x1} \\ W_{x2} \end{bmatrix} \end{aligned} \tag{2.15}$$

2.2.3 Quadratic Programme(QP) optimization Problem

As has been discussed in section 2.1.2, MPC is based on solving an optimization problem that subjected to linear constraints. Recalling the cost function 2.2 associated with the optimization problem:

$$J(x(k|k), U(k)) = \sum_{i=0}^{H_p-1} L(x(k+i|k), u(k+i|k)) + \Phi(x(k+H_p|k)). \tag{2.16}$$

It is crucial to choose the form of the cost function 2.16 the allows the MPC to reach the control objectives. For instance, for output or states regulations, a common choice is to formulate the weighting functions $L(.)$ and $\Phi(.)$ in a quadratic form as below.

$$\begin{aligned} &L(.) = x(k+i|k)^T Qx(k+i|k) + u(k+i|k)^T Ru(k+i|k), i = 0, \cdots, H_p - 1 \\ &\Phi(.) = x(k+H_p|k)Px(k+H_p|k) \end{aligned} \tag{2.17}$$

where $Q \geq 0, R > 0$ and $P \geq 0$ are suitable matrices to be defined to reach the control objectives.
By substituting (2.17) in (2.16), the cost function be rewritten as :

$$\begin{aligned} J(x(k|k), U(k)) = &\sum_{i=0}^{H_p-1} x(k+i|k)^T Qx(k+i|k) + u(k+i|k)^T Ru(k+i|k) \\ &+ x(k+H_p|k)^T Px(k+H_p|k) \end{aligned} \tag{2.18}$$

This is to say, a Quadratic Programme optimization problem arises when the following characteristics take place [17]:

- a quadratic cost function as in (2.18)

- a LTI prediction model as in (2.5)

- linear input and slew rate constraints as in (2.8) and (2.14)

In this context, in the following it will be shown that :

1. the cost function (2.18) is quadratic with respect to optimization variable $U(k)$

2. input and state constraints (2.8) and (2.14) can be rearranged in a unique linear constraint with respect to $U(k)$

By recalling equation (2.5),

$$
\begin{aligned}
x(k+i|k) &= A^i x(k|k) + A^{i-1}Bu(k|k) + A^{i-2}Bu(k+1|k) + \cdots + Bu(k+i-1|k). \\
&= A^i x(k|k) + \sum_{j=0}^{i-1} A^{i-j-1}Bu(k+j|k)
\end{aligned}
\tag{2.19}
$$

the sequence of the predicted states can be expressed as:

$$
\begin{aligned}
X(k) &= [x(k|k), x(k+1|k), \cdots, x(k+H_p-1|k)]^T \\
X(k) &= \mathcal{A}x(k|k) + \mathcal{B}U(k)
\end{aligned}
\tag{2.20}
$$

where:

$$
\mathcal{A} = \begin{bmatrix} A \\ A^2 \\ \vdots \\ A^{H_p} \end{bmatrix}, \mathcal{B} = \begin{bmatrix} B & 0 & 0 & \cdots & 0 \\ AB & B & 0 & \cdots & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ A^{H_p-2}B & A^{H_p-3}B & A^{H_p-4}B & \cdots & \hat{B} \\ A^{H_p-1}B & A^{H_p-2}B & A^{H_p-3}B & \cdots & AB \end{bmatrix}
$$

and by defining matrices $\mathcal{Q}$ and $\mathcal{R}$

$$\mathcal{Q} = \begin{bmatrix} Q & 0 & \cdots & 0 \\ 0 & \ddots & \ddots & \vdots \\ \vdots & \ddots & Q & 0 \\ 0 & \cdots & 0 & P \end{bmatrix} \in \mathbb{R}^{nH_p \times nH_p}, \quad \mathcal{R} = \begin{bmatrix} R & 0 & \cdots & 0 \\ 0 & R & \ddots & \vdots \\ \vdots & \ddots & R & 0 \\ 0 & \cdots & 0 & R \end{bmatrix} \in \mathbb{R}^{mH_p \times mH_p}$$

the cost function (2.18) can be rewritten as :

$$J(x(k|k), U(k)) = X(k)^T \mathcal{Q} X(k) + U^T \mathcal{R} U(k) \tag{2.21}$$

by substituting expression $X(k)$ (2.18) in equation (2.21) and after some manipulations, we get such quadratic form:

$$J(x(k|k), U(k)) = \frac{1}{2} U^T(k) H U(k) + x^T(k|k) f U(k) + \overline{J} \tag{2.22}$$

where:

$$\begin{aligned} H = & \quad 2(\mathcal{B}^T \mathcal{Q} \mathcal{B} + \mathcal{R}) \\ f = & \quad 2\mathcal{A}^T \mathcal{Q} \mathcal{B} \\ \overline{J} = & \quad x^T(k|k) \mathcal{A}^T Q \mathcal{A} x(k|k) \end{aligned} \tag{2.23}$$

Principally, H is the Hessian matrix which is positive definite such that $H > 0$.

As has been discussed in section 2.2.2, the control input and linear state constraints can be expressed as:

$$\begin{aligned} L_u U(k) \leq W_u \\ L_x U(k) \leq W_x \end{aligned}$$

By combining both input and state constraints, we get:

$$LU(k) \leq W \tag{2.24}$$

where $L = [L_u, L_x]^T$ and $W = [W_u, W_x]^T$ represent a set linear constrain on $U(k)$.

To sum up, the optimization problem associated with the MPC can be formulated as the following Quadratic Programming problem :

$$\begin{aligned} U^o = arg \quad & \min_{U(k)} \frac{1}{2} U^T(k) H U(k) + x^T(k|k) f U(k) \\ s.t \\ LU(k) < W \end{aligned} \tag{2.25}$$

According to the literature, such QP formulations ensure that the optimization problem is convex and the optimal solution can be satisfactorily computed through different numerical algorithms such as:

- "active" set algorithm [13],

- "Prime-dual" interior point algorithms [13],

Furthermore, new algorithms have been introduced recently ensuring efficiency in terms of online computation of the MPC. Some of these algorithms are:

- The Partial enumerator methodology [14],

- The modified active set method [15],

- The approximate barial solution [16],

Moving to another context, in case of the absence of the constraints, an explicit solution for the optimal control input sequence U^o can be found by minimizing the cost function (2.22) such that:

$$U^o = -H^{-1} f x(k|k) \tag{2.26}$$

The optimal control input sequence U^o has to be computed for all the finite prediction steps H_p. In this context, for High values of H_P, the computation of the optimal control sequence becomes significantly intensive. In order to solve this problem a possible solution is to optimize the the cost function (2.18) with a reduced number of finite steps such that $U(k) = [u(k|k), u(k+1|k), \cdots, u(k+H_c-1|k)]^T$ where $H_c \leq H_P$ is referred as the Control Horizon. In case $H_c < H_P$ the remaining $H_p - H_c$ control input command $U(k) = [u(H_c|k), u(H_c+1|k), \cdots, u(k+H_p-1|k)]^T$ can be chosen :

1. $u(k+i|k) = 0 \quad H_c \leq i \leq H_p - 1$,
2. $u(k+i|k) = u(k+H_c|k) \quad H_c \leq i \leq H_p - 1$,
3. $u(k+i|k) = -Fx(k+i|k) \quad H_c \leq i \leq H_p - 1$ where $F \in \mathbb{R}^n$ is the gain of stabilizing feedback gain.

2.2.4 Receding Horizon (RH) principle

Finite horizon(FH) optimization results in optimal control sequence obtained from equations (2.25) and (2.26) which starts from at current time i and ends at $H_C - 1$. The optimized control sequence corresponds to an open loop control technique. However, open loop control strategy has potential drawbacks , it is well known that such control technique is highly non robust due to modeling errors, parameter uncertainties or disturbances. Thus, closed loop is applied by considering the Receding Horizon (RH) Principle is going to be exploited to overcome such limitations. The RH principle can be summarized in the following:

1. get the state $x(k) = x(k|k)$
2. get the optimal control sequence by solving the optimization problem over the interval $[k, k + H_P - 1]$.
3. apply the first step of the optimal control sequence.
4. Repeat the finite optimization problem by moving to time $k + 1$ and predict over the interval $[k + 1, k + H_P]$ considering the current state $x(k + 1)$.

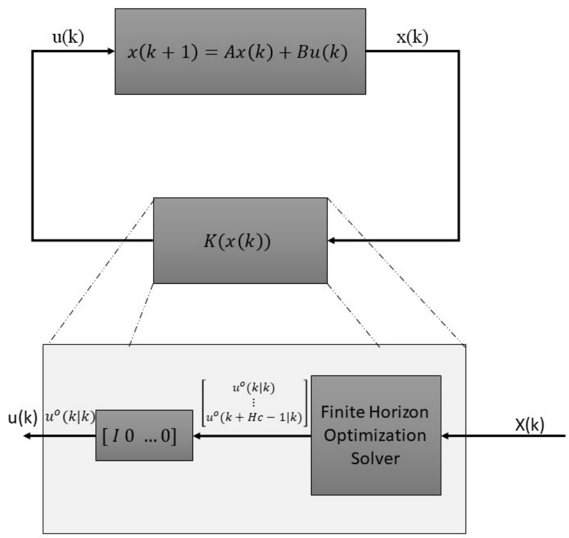


Figure 8: Scheme of static feedback control law

from equation (2.22), it is obvious the cost function $J(.)$ depends only on the current state $x(k|k) = x(k)$. Thus the computed control input $u(k)$ at time k is equal to :

$$u(k) = u^o(k|k) = u^o(k|k) = u^o(x(k|k)) = u^o(x(k)) \tag{2.27}$$

Therefore, RH principle implicitly introduces a state feedback control law in the form $U(k) = \mathcal{K}(x(k))$. Figure 8 shows the principle of applying a state feedback control law.

2.2.5 Tracking Problem

In order to track a desired pressure reference, the cost function can be modified taking into account the reference $P_{ref}(k)$ into the formulation:

$$\begin{aligned} J(P(k|k), U(k)) = \sum_{i=0}^{H_p-1} &(P_{ref}(k+i|k) - Cx(k+i|p))^T Q_p (P_{ref}(k+i|k) - Cx(k+i|p)) \\ &+ u(k+i|k)^T R u(k+i|k) \end{aligned} \tag{2.28}$$

The optimization problem (2.25) tries to eliminate the tracking error of the states variables $\|P_{ref}(k+i|k) - Cx(k+i|k)\|$. However, due to the presence of uncertainties in the system, it is common to find a steady state error, even for tracking constant references. A possible solution to overcome this problem, is to include integral action to eliminate the steady state error. This is will be discussed in the next sections.

2.2.6 Explicit integral action

In control theory, in order to achieve a good response in terms of tracking a constant reference, an integral action should be taken into consideration. Integral action is a common approach to eliminate steady state error such that the integral term of the tracking error is taken in consideration as an extra state variable. The tracking error in the discrete time form is:

$$e(k) = P_{ref}(k) - P(k) = P_{ref}(k) - Cx(k) = \frac{q(k+1) - q(k)}{T_s} \tag{2.29}$$

Where $P(k)$, T_s are the output pressure and sampling time respectively.

Hence, the integral term of the tracking error is:

$$q(k+1) = q(k) + T_s(P_{ref}(k) - Cx(k)) \tag{2.30}$$

Consequently, the state equation is augmented in following form:

$$\begin{aligned}
\underbrace{\begin{bmatrix} x(k+1) \\ q(k+1) \end{bmatrix}}_{\hat{x}(k+1)} &= \underbrace{\begin{bmatrix} A & 0 \\ -T_s C & 1 \end{bmatrix}}_{\hat{A}} \underbrace{\begin{bmatrix} x(k) \\ q(k) \end{bmatrix}}_{\hat{x}} + \underbrace{\begin{bmatrix} B \\ 0 \end{bmatrix}}_{\hat{B}} u(k) + \begin{bmatrix} 0 \\ T_s \end{bmatrix} r(k) \\
\hat{y}(k) &= \underbrace{\begin{bmatrix} C & 0 \end{bmatrix}}_{\hat{C}} \begin{bmatrix} x(k) \\ e(k) \end{bmatrix}
\end{aligned} \tag{2.31}$$

where:

- $\hat{A}$, $\hat{B}$ and $\hat{C}$ are the augmented state matrices.
- $\hat{x}$ is the augmented state variables.

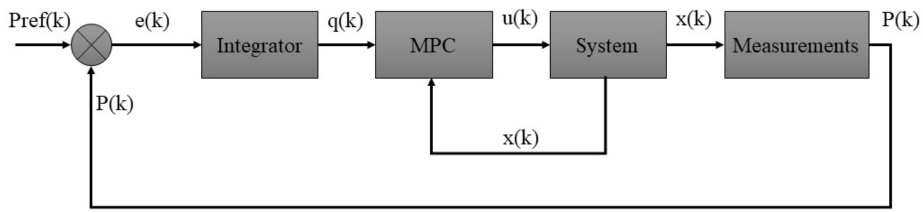


Figure 9: MPC schematic diagram with integral action

Accordingly, the cost function (2.28) is modified to the following :

$$
\begin{aligned}
J(P(k|k), U(k)) = & \sum_{i=0}^{H_p-1} (P_{ref}(k+i|k) - P(k+i|p))^T Q_p (P_{ref} - P(k+i|p))) \\
& + q(k+i|k)^T Q_{int} q(k+i|k) + u(k+i|k)^T R u(k+i|k) \\
& s.t \\
Pmin(k+i|k) \leq & \, P(k+i|k) \leq Pmax(k+i|k), i = 1, 2, \cdots, H_p - 1 \\
U^{min} \leq & \, u(k+i|k) \leq U^{max}, i = 1, 2, \cdots, H_c - 1 \\
\Delta U^{min} \leq & \, \Delta u(k+i|k) \leq \Delta U^{max}, i = 1, 2, \cdots, H_c - 1
\end{aligned}
\tag{2.32}
$$

where:

- $Pmin$ and $Pmax$ are the minimum and maximum pressure constraints associated with the overshoot $\hat{s}$ defined in (1.2.3).

- Q_{int} is a tuning parameter adopted to regulate the tracking error. High values of Q_{int} ensures that the tracking error converges to zero faster with a strong command action. However, the higher the Q_{int}, the more aggressive is the system response.

2.2.7 Implicit integral action

Another method to include an integral action in the formulation of the MPC is to optimize the cost function based on the increment of the Control input Δu by considering the control input Δu. By considering the previous control input value$u(k-1)$ as an extra state variable, an implicit integral action is imposed to derive a control input $u(k)$ to the system as shown in figure 10.

This results in eliminating the steady state error.

In this context the state space matrices are augmented in the following form:

$$x(k+1) = Ax(k) + Bu(k) \tag{2.33a}$$
$$y(k) = Cx(k) \tag{2.33b}$$
$$x(0) = x0 \tag{2.33c}$$
$$\Delta u(k) = u(k) - u(k-1) \tag{2.33d}$$

By substituting equation (2.33d) in (2.33b) ,we get:

$$\begin{gathered}\underbrace{\begin{bmatrix} x(k+1) \\ u(k) \end{bmatrix}}_{\hat{x}(k+1)} = \underbrace{\begin{bmatrix} A & B \\ 0 & I \end{bmatrix}}_{\hat{A}} \underbrace{\begin{bmatrix} x(k) \\ u(k-1) \end{bmatrix}}_{\hat{x}} + \underbrace{\begin{bmatrix} B \\ 1 \end{bmatrix}}_{\hat{B}} \Delta u(k) \\ \hat{y}(k) = \underbrace{\begin{bmatrix} C & 0 \end{bmatrix}}_{\hat{C}} \begin{bmatrix} x(k) \\ u(k-1) \end{bmatrix}\end{gathered} \tag{2.34}$$

Where:

- $\hat{A}$, $\hat{B}$ and $\hat{C}$ are the augmented state matrices.
- $\hat{x}$ is the augmented state variables.

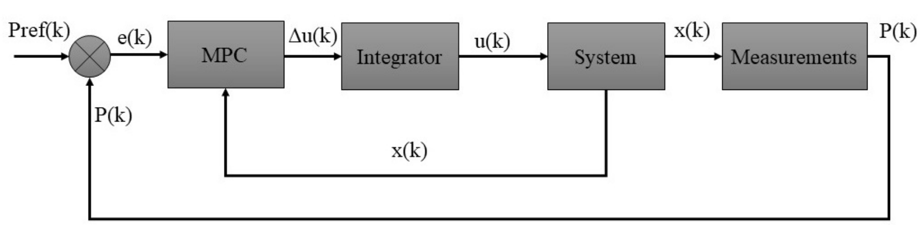


Figure 10: MPC schematic diagram with integral action based on the input increment

Hence, the objective function can be formulated as follows:

$$
\begin{aligned}
J(P(k|k), \Delta U(k)) &= \sum_{i=0}^{H_p-1} (P_{ref}(k+i|k) - P(k+i|k))^T Q_p (P_{ref}(k+i|k) - P(k+i|k)) \\
&+ \Delta u(k+i|k)^T R \Delta u(k+i|k) \\
\Delta U(k) &= \begin{bmatrix} \Delta u(k|k) & \Delta u(k+1|k) & \dots & \Delta u(k+H_c-1|k) \end{bmatrix}^T \\
& s.t \\
Pmin(k+i|k) &\leq P(k+i|k) \leq Pmax(k+i|k), i = 1, 2, \cdots, H_p - 1 \\
U^{min} &\leq u(k+i|k) \leq U^{max}, i = 1, 2, \cdots, H_c - 1 \\
\Delta U^{min} &\leq \Delta u(k+i|k) \leq \Delta U^{max}, i = 1, 2, \cdots, H_c - 1
\end{aligned}
\tag{2.35}
$$

3 Multi-MPC based on multiple Hammerstein models

As has been discussed in chapter 1, it is crucial to represent the K2-actuator with a mathematical model that represents the dynamics of the system. The identification of K2-actuator is based on measured data provided by Centro Ricerche Fiat(CRF). MATLAB system identification tool box is used to identify the system.

3.1 Hammerstein system

The presence of the dead zone as discussed in section 1.2.2, non linearity plays a big role in our system. The K2-actuator can be considered as a Hammerstein system. A Hammerstein model consists of a static non-linear function followed by a Linear Time Invariant (LTI) system as shown in the below figure.

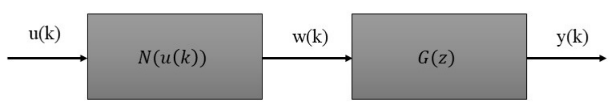


Figure 11: Hammerstein system block diagram

Where:

- N: is a static nonlinear function.
- $w(k)$: is the intermediate variable which is the output of the nonlinear function such that $w(k) = N(u(k))$. $w(k)$ has the same dimensions of $u(k)$.
- $G(z)$: Discrete Linear Time Invariant system where $G(z)$ is the ratio between of the the output of the system $y(k)$ and the intermediate variable $w(k)$ such that $G(z) = \frac{y(z)}{w(z)}$.

3.1.1 Identification

Two different data sets have been provided by CRF during different working conditions. One data set can be used for identifying the K2-actuator while the other can be used for validating the identified model or vice versa.
As has been discussed, MATLAB System Identification Toolbox is exploited to identify the K2-actuator. In MATLAB System Identification Toolbox, the feature of estimating nonlinear models has been chosen considering the system as a Hammerstein model such that the nonlinear part behaves as dead zone. After running several simulations and by trial and error approach, we have identified the model based on the on the following structure:

- Nonlinear Part is described as a dead zone until [0%, 38.99676%] of U^{max}. After then, the intermediate variable w increases linearly with respect to the input current.

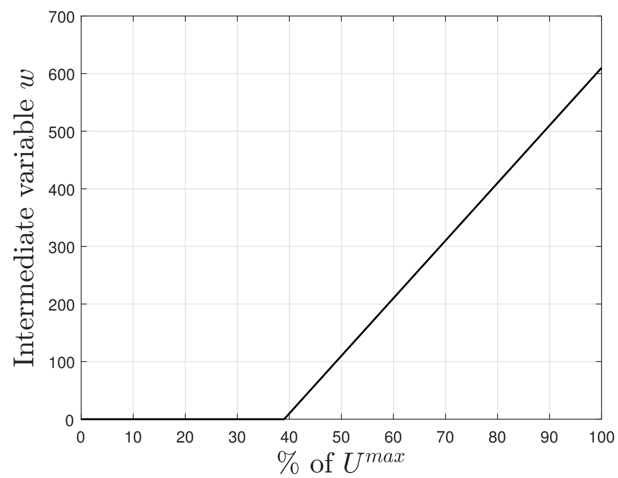


Figure 12: Static non-linearity of the dead zone

- Linear Part:

$$G(z) = \frac{y(z)}{w(z)} = \frac{\beta}{z+\alpha} \tag{3.1}$$

where $\alpha = -0.9939$ and $\beta = 4.1746e-04$

3.1.2 Validation

The identified model has been validated using the same the input-output data used for the identification procedure. As shown below in figure 13, the estimated output is able to track the measured output data. Correspondingly, the absolute error of the estimated output is about 5.6250% of P^{max} as shown in figure 14.

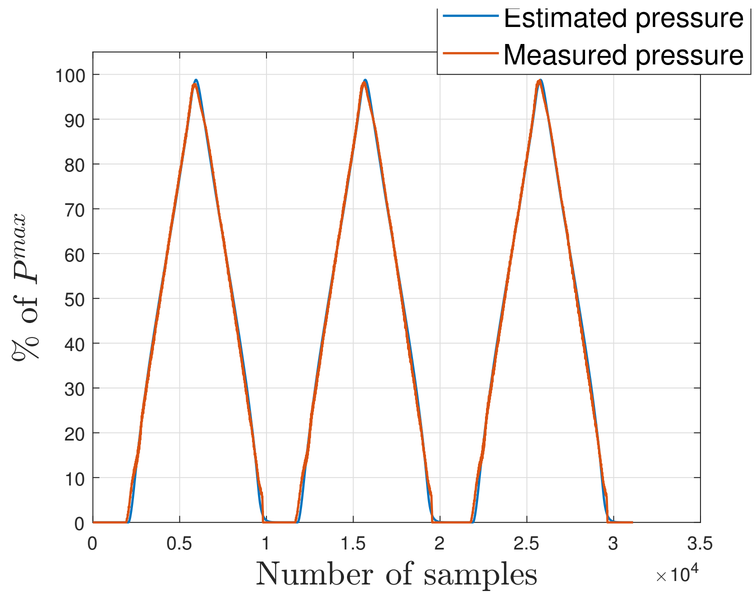


Figure 13: Estimated output vs measured output - ramp reference

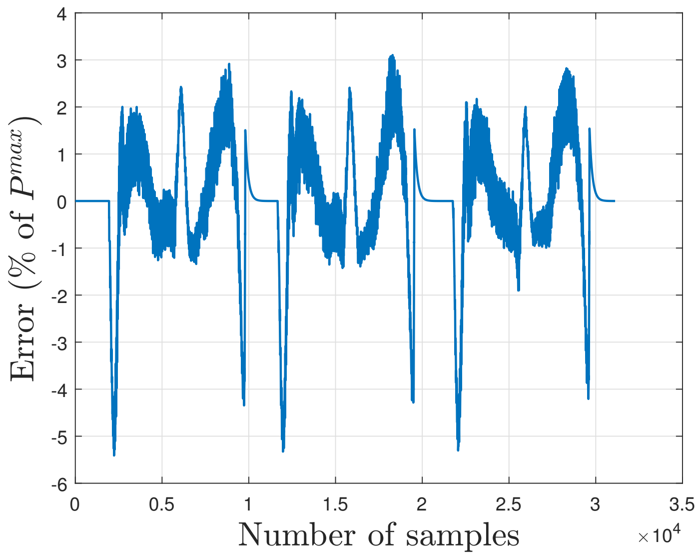


Figure 14: Error of the estimated output- ramp reference

The second data set; the steps profile is used to validate the identified model as well, however unlike the previous data set, the estimated output is not able to catch the measured output data for the whole working conditions as shown in figure 15. Consequently, the absolute error of the estimated output increased dramatically recording 80.85% of P^{max} as shown in figure 16.

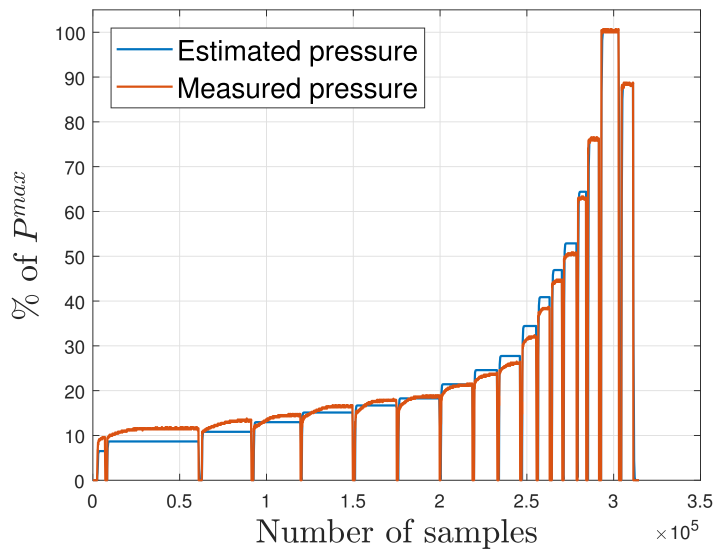


Figure 15: Estimated output vs measured output - steps reference

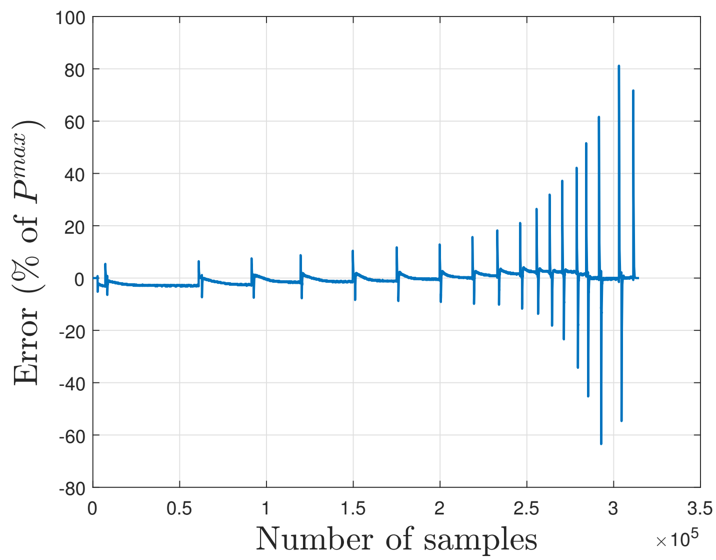


Figure 16: Error of the estimated output- steps reference

3.2 Multi-Hammerstein models

Based on the validation results of the identified Hammerstein model in the previous section 3.1.2 that the estimated output was not able to catch the measured output in different operating conditions. Multiple Models approach will be considered in the next section to represent the K2-actuator in varying working conditions.

3.2.1 Identification of multiple Hammerstein models

The second input-output data set that represents the steps profiles will be subdivided into 3 data sets to represent the K2-actuator in the following working conditions:

- Low Pressure [0%,12.5%] of P^{max}.
- Medium Pressure [12.5%,50%] of P^{max}.
- High Pressure [50%,100%] of P^{max}.

Hence, 3 different Hammerstein models will be identified to represent the K2-actuator for the entire pressure range. Accordingly, the K2-actuator will be represented as the structure illustrated in the below figure 17.

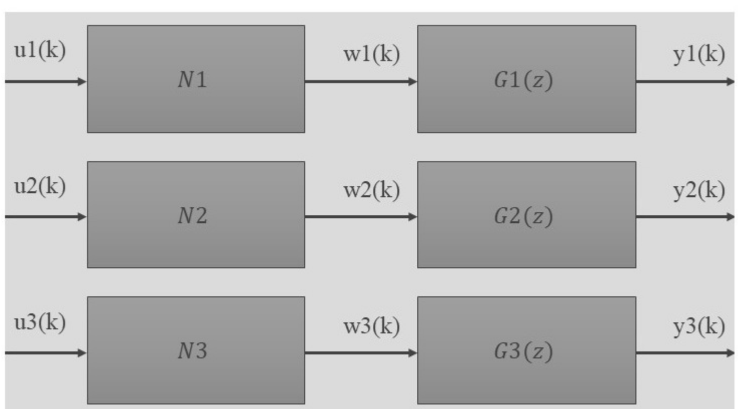


Figure 17: K2-actuator represented in Multi-Models approach

Likewise, the identification of the multiple models follow the same procedure of identifying a single model as has been discussed in section 3.1.1 by utilizing the nonlinear models features provided by MATLAB System Identification Toolbox.

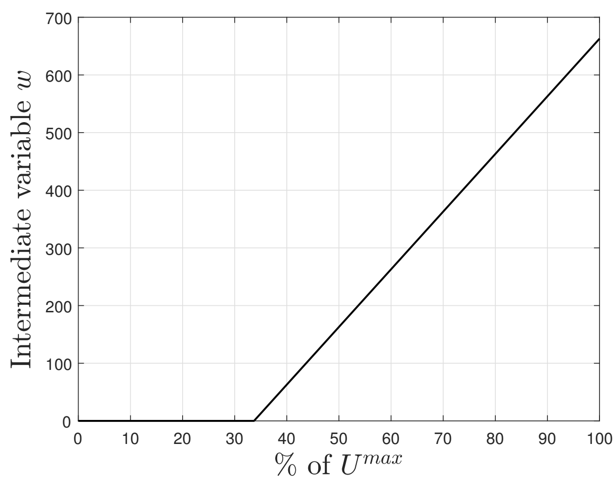
Each data set is exploited separately to identify a unique Model that corresponds to a particular working conditions. By considering the nonlinear part as a dead zone and the linear model made up of 1st order transfer such that:

$$G_n(z) = \frac{y(z)}{w(z)} = \frac{\beta_n}{z + \alpha_n} \qquad n = 1, 2, 3 \tag{3.2}$$

We are able to identify 3 different models that satisfy the whole working conditions.

- First Hammerstein Model

  * Non-Linear Model
    Dead zone interval :[0%,33.70901%] of P^{max}

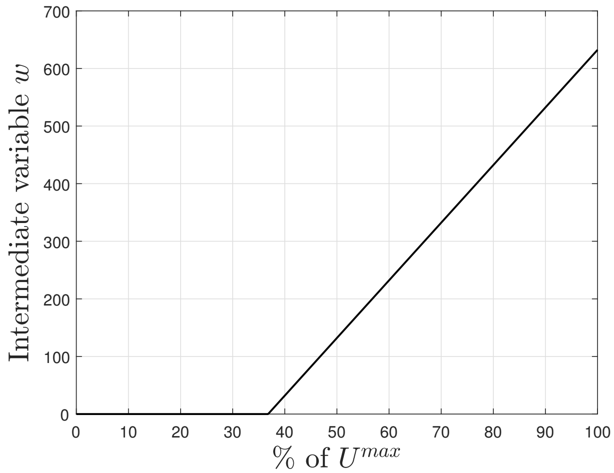


  * Linear Model

$$G_1(z) = \frac{0.001225}{z - 0.9717} \tag{3.3}$$

- Second Hammerstein Model

* Non-Linear Model
  Dead zone interval :[0%,36.7.92%] of P^{max}

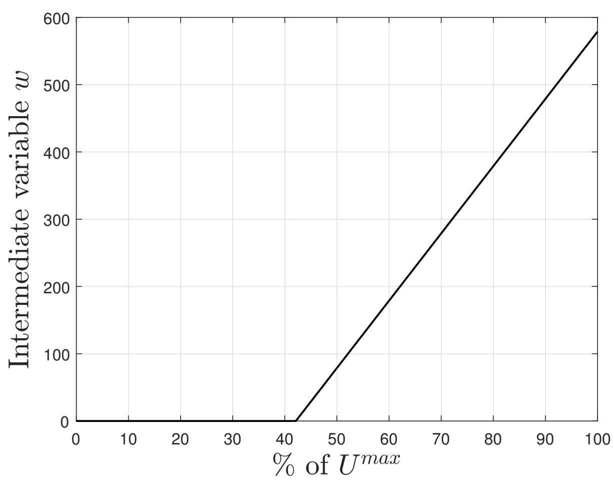


* Linear Model

$$G_2(z) = \frac{0.001635}{z - 0.9711} \tag{3.4}$$

- Third Hammerstein Model

  * Non-Linear Model
    Dead zone interval : [0%,42.1179%] of P^{max}



* Linear Model

$$G_3(z) = \frac{0.003533}{z - 0.9513} \tag{3.5}$$

3.2.2 Multi-Model MPC Controller

Forthwith identifying the Multi-Models that satisfy the wide range of operating conditions, multiple MPC controllers have been tuned in order to meet the performance requirements of the system. Basically, Multiple MPC is fundamentally utilizing Gain Scheduling technique to control a nonlinear plant that operates over a wide range of operating conditions. Multiple MPC switches between the pre-tuned controllers according to a predefined logic that detect the suitable controller that fit that particular operating condition range noting that each of these controller is tuned based on one of the three identified models that have been discussed before with different weighting functions.

11 controllers have been designed to satisfy the whole range of operating conditions by exploiting the 3 identified model such that:

- $1^{st}, 2^{nd}$ and 3^{rd} controllers are dedicated to control the 1^{st} model that represents the K2-actuator at low pressure.

- $4^{th}, 5^{th}, 6^{th}, 7^{th}, 8^{th}$ and 9^{th} controllers are dedicated to control the 2^{nd} model that represents the K2-actuator at medium pressure.

- 10^{th} controller is dedicated to control the 3^{rd} model that represents the K2-actuator at high pressure.

- 11^{th} controller is designed to track 0% of P^{max} as a reference which has been modeled based on the 1^{st} model.

In fact, only one controller from the 11 provides the optimal control input. In the meanwhile, the other controllers become inactive during run time. To enhance the performance of the multiple MPC and to ensure a smooth response during the switching between the controllers, the inactive controllers keep on predicting the states. This prevents any sudden change in the manipulated variable when the controller switching exists.

As has been discussed in section 2.2.3, MPC is based on solving a QP optimization optimization problem. However, a Hammerstein Model is composed of a static nonlinear function followed by a linear dynamic model. This static non linearity leads the optimization problem to be non convex. In order to solve this problem, the linear dynamic model is only taken into consideration for designing each MPC controller where the intermediate function is considered as the input to the linear model. Therefore, the optimized manipulated variable is the intermediate variable $w(k)$. Hence, an inverse of the static nonlinear function is applied directly to the manipulated variable$w(k)$ in order to obtain the actual input $u(k)$ as in the below figure 18.

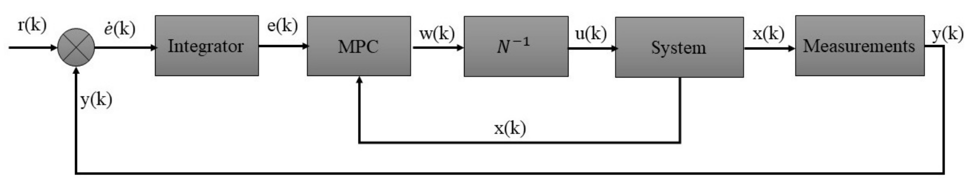


Figure 18: MPC scheme for Hammerstein system

3.3 Multiple MPC tuning and simulation results

MATLAB MPC Toolbox is used to simulate the tracking of the Pressure for different steps references. The advantage of exploiting MATLAB MPC Toolbox is to decrease the running time of the simulation since there are 11 controllers running in parallel that make the computation of the optimal solution extremely intensive.

MATLAB MPC Toolbox formulates the MPC problem by optimizing the objective function based on the increment of the manipulated variable adding an explicit integral action as has been discussed in equation (2.35).

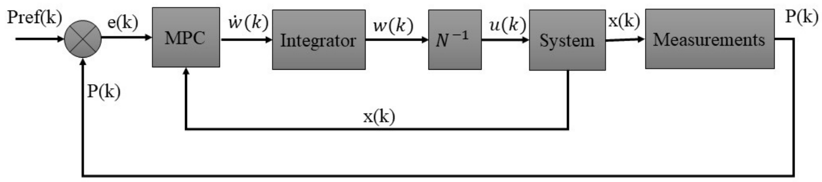


Figure 19: MPC scheme for Hammerstein system

As has been discussed in 3.2.2 that the optimized variable for the MPC to control a Hammerstein model is the intermediate variable $w(k)$, the cost function (2.35) utilized by MATLAB is modified as the following:

$$\begin{aligned}
J(x(k|k), \Delta W(k)) = &\sum_{i=0}^{H_p-1} (P_{ref}(k+i|k) - P(k+i|k))^T Q_p (P_{ref}(k+i|k) - P(k+i|k)) \\
&+ \Delta w(k+i|k)^T R \Delta w(k+i|k) \\
\Delta W(k) = &\begin{bmatrix} \Delta w(k|k) & \Delta w(k+1|k) & \dots & \Delta w(k+H_c-1|k) \end{bmatrix}^T \\
&s.t \\
Pmin(k+i|k) \leq P(k+i|k) &\leq Pmax(k+i|k), i = 1, 2, \cdots, H_p - 1 \\
w^{min}(k+i|k) \leq W(k+i|k) &\leq W^{max}(k+i|k), i = 1, 2, \cdots, H_c - 1 \\
\Delta w^{min}(k+i|k) \leq \Delta w(k+i|k) &\leq \Delta w^{max}(k+i|k), i = 1, 2, \cdots, H_c - 1
\end{aligned} \tag{3.6}$$

After several simulations by trial and error technique, the following design parameters of the 11 controllers have been obtained to control the whole range of the operating region of the K2-actuator.

1^{st} **Controller**

$$
\begin{aligned}
Q_p &= 0.55 \\
R &= 1 \\
w^{min}(k+i|k) &= 0 \\
w^{max}(k+i|k) &= 662.91 \\
\Delta w^{min}(k+i|k) &= -0.04 \\
H_p &= 150 \\
H_c &= 150
\end{aligned}
\tag{3.7}
$$

2^{nd} **Controller**

$$
\begin{aligned}
Q_p &= 10 \\
R &= 12 \\
w^{min}(k+i|k) &= 0 \\
w^{max}(k+i|k) &= 662.91 \\
\Delta w^{min}(k+i|k) &= -0.075 \\
H_p &= 150 \\
H_c &= 150
\end{aligned}
\tag{3.8}
$$

3^{rd} **Controller**

$$
\begin{aligned}
Q_p &= 10 \\
R &= 12 \\
w^{min}(k+i|k) &= 0 \\
w^{max}(k+i|k) &= 662.91 \\
\Delta w^{min}(k+i|k) &= -0.075 \\
H_p &= 150 \\
H_c &= 150
\end{aligned}
\tag{3.9}
$$

4^{th} **Controller**

$$\begin{aligned} Q_p &= 10 \\ R &= 12 \\ w^{min}(k+i|k) &= 0 \\ w^{max}(k+i|k) &= 632.31 \\ \Delta w^{min}(k+i|k) &= -0.087 \\ H_p &= 150 \\ H_c &= 150 \end{aligned} \tag{3.10}$$

5^{th} **Controller**

$$\begin{aligned} Q_p &= 10 \\ R &= 11.8 \\ w^{min}(k+i|k) &= 0 \\ w^{max}(k+i|k) &= 632.31 \\ \Delta w^{min}(k+i|k) &= -0.1 \\ H_p &= 150 \\ H_c &= 150 \end{aligned} \tag{3.11}$$

6^{th} **Controller**

$$\begin{aligned} Q_p &= 10 \\ R &= 12.5 \\ w^{min}(k+i|k) &= 0 \\ w^{max}(k+i|k) &= 632.31 \\ \Delta w^{min}(k+i|k) &= -0.115 \\ H_p &= 150 \\ H_c &= 150 \end{aligned} \tag{3.12}$$

7^{th} **Controller**

$$\begin{aligned} Q_p &= 10 \\ R &= 12.8 \\ w^{min}(k+i|k) &= 0 \\ w^{max}(k+i|k) &= 632.31 \\ \Delta w^{min}(k+i|k) &= -0.14 \\ H_p &= 150 \\ H_c &= 150 \end{aligned} \tag{3.13}$$

8^{th} **Controller**

$$\begin{aligned} Q_p &= 10 \\ R &= 15 \\ w^{min}(k+i|k) &= 0 \\ w^{max}(k+i|k) &= 632.31 \\ \Delta w^{min}(k+i|k) &= -0.14 \\ H_p &= 150 \\ H_c &= 150 \end{aligned} \tag{3.14}$$

9^{th} **Controller**

$$\begin{aligned} Q_p &= 10 \\ R &= 15 \\ w^{min}(k+i|k) &= 0 \\ w^{max}(k+i|k) &= 632.31 \\ \Delta w^{min}(k+i|k) &= -0.17 \\ H_p &= 150 \\ H_c &= 150 \end{aligned} \tag{3.15}$$

10^{th} **Controller**

$$\begin{aligned}
Q_p &= 0.0415 \\
R &= 6.7281 \\
w^{min}(k+i|k) &= 0 \\
w^{max}(k+i|k) &= 578.83 \\
\Delta w^{min}(k+i|k) &= -2.8 \\
H_p &= 20 \\
H_c &= 20
\end{aligned} \tag{3.16}$$

11^{th} **Controller**

$$\begin{aligned}
Q_p &= 0.55 \\
R &= 1 \\
w^{min}(k+i|k) &= 0 \\
w^{max}(k+i|k) &= -0.087 \\
H_p &= 150 \\
H_c &= 150
\end{aligned} \tag{3.17}$$

Figure 20 shows the response of the pressure to track different steps references besides the switching of the pre-tuned controllers according to a predefined logic that detect the right controller for the current working condition. Moreover, figures 21,22 and 23 and table 1 reports the performance of the transient performance at different step references.

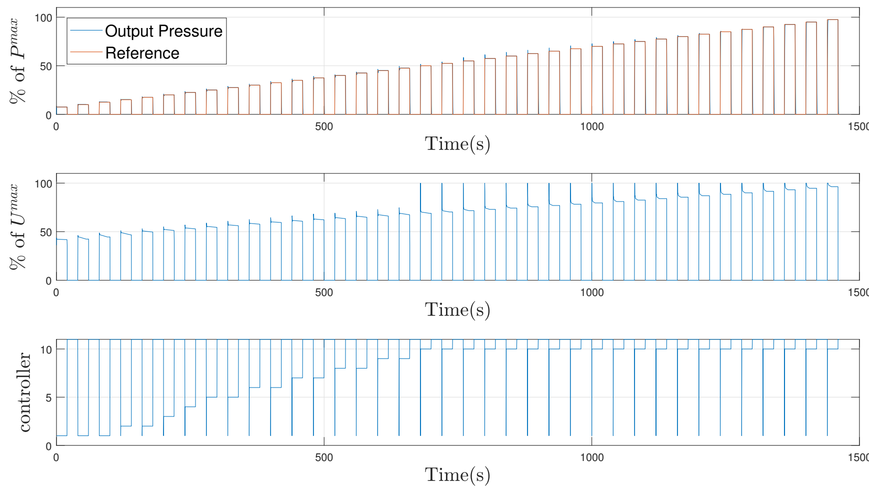


Figure 20: Pressure response for step reference using Multi-model Multi-MPC approach

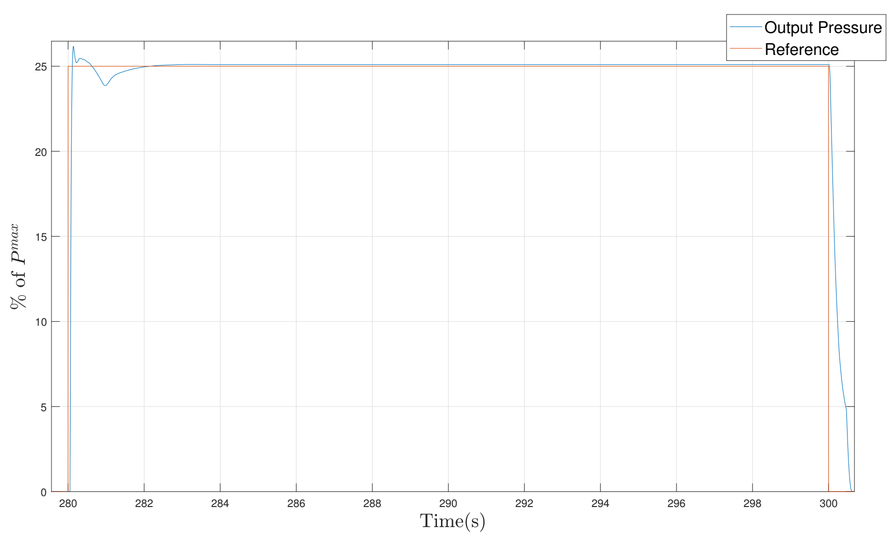


Figure 21: Pressure response for step reference of 25% of P^{max}

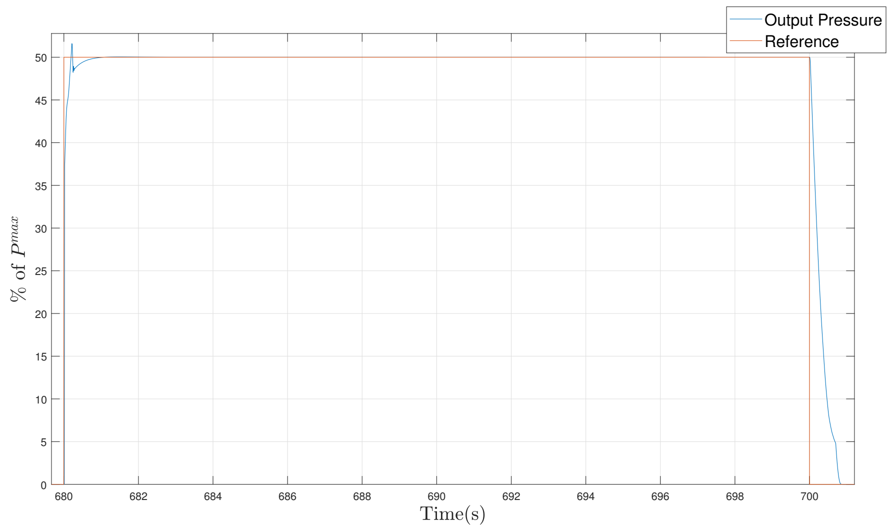


Figure 22: Pressure response for step reference of 50% of P^{max}

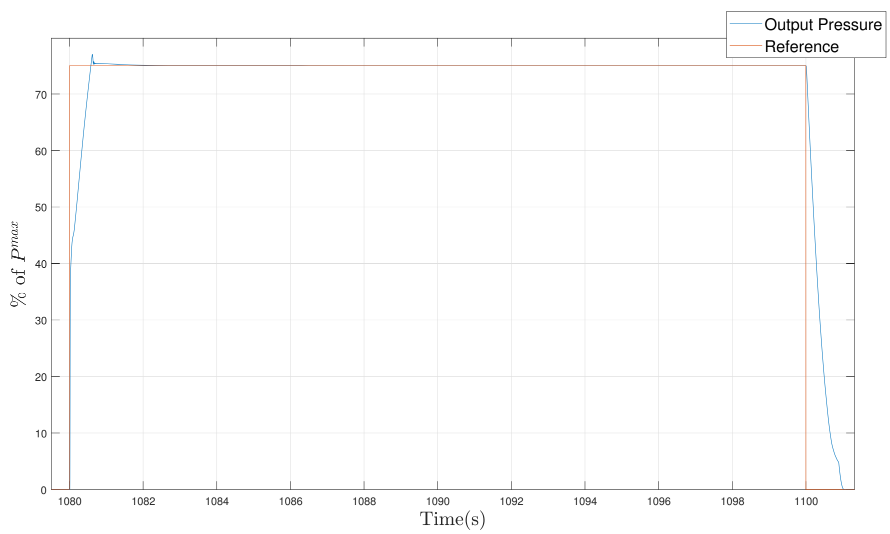


Figure 23: Pressure response for step reference of 75% of P^{max}

Reference % of P^{max}	Overshoot (%) $\hat{s}$	rise time (ms) T_r	settling time (ms) $T_{settling}$
25	4.7	111	111
50	3.19	192	192
75	2.7	581	581

Table 1: Transient response at different step references for Multi-MPC approach

To conclude, by exploiting Multiple Model Predictive Controlled based on Multiple models , we have achieved a good performance particularly for the overshoot requirement as shown in table 1. However, due to the complexity of Multiple MPC structure and due to the long prediction horizon used by each MPC controller, the computation of the optimal solution is quite intensive which lasts for almost 24 hours that can not be implemented on a real Vehicle's hardware. Therefore, different control architectures will be considered and described in the following chapters to achieve a good trade-off between pressure trajectories.

4 Adaptive Model Predictive Control

Adaptive control is a common approach to control a nonlinear system which its parameters vary within time. The control procedure is based on adjusting the control law at each sampling time by adapting the system's parameters using a suitable online estimator algorithm. In fact, The effectiveness of the online estimation occurs particularly for controlling nonlinear system that its dynamic behavior changes significantly according to the operating condition. Figure 24 shows a schematic scheme of Adaptive Model Predictive control which is based on two procedures:

1. Estimate the system parameters $\hat{\theta}(k)$ online using a suitable online estimator algorithm. The estimation of the parameters employ on the basis of the current input $u(k)$ and output $y(k)$ measurements.

2. Adjust the control law of the MPC by updating the state space model associated with the new estimated parameters $\hat{\theta}(k)$.

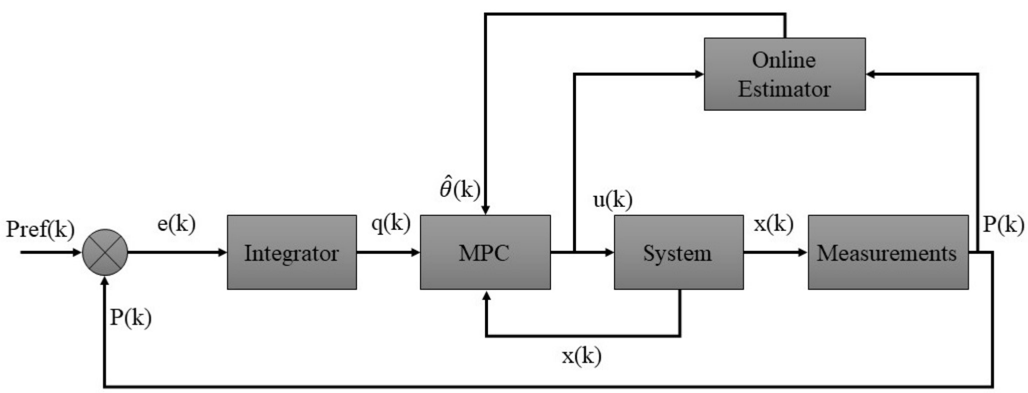


Figure 24: Schematic diagram of Adaptive MPC

Recursive least square (RLS) is a common estimator to be employed for estimating the system online. In the next section we will discuss the main conceptual aspects of the RLS and the different RLS estimation algorithms that can be exploited to reach a good performance in terms of estimation error and stability.

4.1 Least square

Fundamentally, RLS is based on solving a Least square problem recursively, where least square is common linear regression approach to find an approximate solution for an over-determined system by minimizing the least square error that satisfy a finite measured input-output data N. In the following we will show the derivation of the optimal solution for estimating the parameters $\hat{\theta}$. First of all, by assuming the system has the following structure:

$$\begin{aligned} G(z) = \frac{Y(z)}{U(z)} &= \frac{b_0 z^m + b_1 z^{m-1} + \cdots + b_m}{z^n + a_1 z^{n-1} + \cdots + a_n}, m \leq n \\ \frac{Y(z)}{U(z)} &= \frac{b_0 z^{m-n} + b_1 z^{m-n-1} + \cdots + b_m z^{-n}}{1 + a_1 z^{-1} + \cdots + a_n z^{-n}} \end{aligned} \tag{4.1}$$

where n is the system order type. By applying some rearrangements , $Y(z)$ can be written as:

$$\begin{aligned} &Y(z)[1 + a_1 z^{-1} + \cdots + a_n z^{-n}] = U(z)[b_0 z^{m-n} + b_1 z^{m-n-1} + \cdots + b_m z^{-n}] \\ &Y(z) = -a_1 Y(z) z^{-1} - \cdots - a_n Y(z) z^{-n} + b_0 U(z) z^{m-n} + b_1 U(z) z^{m-n-1} + \cdots + b_m U(z) z^{-n} \end{aligned} \tag{4.2}$$

By transferring to the time domain

$$\begin{aligned} y(k) &= -a_1 y(k-1) - \cdots - a_n y(k-n) \\ &+ b_0 u(k+m-n) + b_1 u(k+m-n-1) + \cdots + b_m u(k-n) \end{aligned} \tag{4.3}$$

$$Y = \Psi \hat{\theta} \tag{4.4}$$

$$Y = \begin{bmatrix} Y(k+n+1) \\ Y(k+n+2) \\ \vdots \\ Y(k+N) \end{bmatrix} \tag{4.5}$$

$$\Psi = \begin{bmatrix} -y(k+n) & -y(k+n-1) & \cdots & -y(k+1) & u(k+m+1) & U(k+m) & \cdots & u(k+1) \\ -y(k+n+1) & -y(k+n) & \cdots & -y(k+2) & u(k+m+2) & U(k+m+1) & \cdots & u(k+2) \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ -y(k+N-1) & -y(k+N-2) & \cdots & -y(k+N-n) & u(k+N+m-n) & u(k+N+m-n-1) & \cdots & u(k+N-n) \end{bmatrix} \tag{4.6}$$

$$\theta = \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_n \\ b_0 \\ b_1 \\ \vdots \\ b_m \end{bmatrix} \tag{4.7}$$

where:
θ: are the parameters of the system to be identified.
Ψ: is the regressor matrix that composed of previous values of output and input.
Y: The output samples have been collected up to the current time.

If the number of the output samples N equals to the number of parameters to be identified $(n + m + 1)$, a unique solution is found. On the other hand, if the number of the output samples N is greater than the number of parameters to be estimated which is the common case, an approximate solution has to be found to satisfy the whole data set. The approximate solution is found my minimizing the least square error such that:

$$\begin{aligned} \hat{\theta} =& arg \quad \underset{\theta}{min} \|Y - \Psi\theta\|_2 \\ Y =& \Psi\hat{\theta} \\ \Psi^T Y =& \Psi^T \Psi\hat{\theta} \\ (\Psi^T \Psi)^{-1} \Psi^T Y =& (\Psi^T \Psi)^{-1} \Psi^T \Psi\hat{\theta} \\ \hat{\theta} =& (\Psi^T \Psi)^{-1} \Psi^T Y \end{aligned} \tag{4.8}$$

Where P^{-1} can be considered as $P^{-1} = (\Psi^T \Psi)$, Therefore, the equation can be rewritten as

$$\hat{\theta} = P\Psi^T Y \tag{4.9}$$

An approximate solution is found for the whole data set if and only if the P matrix is invertible.

4.2 Recursive least square

In case of receiving a new measurements of input and output data, the least square matrices have to be augmented increasing in dimensions and solve the least square optimization problem again.

$$
\begin{aligned}
Y_{k+N+1} &= \begin{bmatrix} Y \\ y(k+N+1) \end{bmatrix} \\
\Psi_{k+N+1} &= \begin{bmatrix} \Psi \\ \psi(k+N+1) \end{bmatrix} \\
P_{k+1}^{-1} &= P_k^{-1} + \psi(k+1)\psi^T(k+1)
\end{aligned}
\tag{4.10}
$$

$$
\hat{\theta}_{k+1} = (\Psi_{k+1}^T \Psi_{k+1})^{-1} \Psi_{k+1}^T Y_{k+1} \tag{4.11}
$$

At each instant of receiving a new measurement , the size of the matrices get bigger and bigger and solving the least square optimization problem becomes computationally intensive. In order to solve this problem, recursive least square method is exploited to estimate the parameters of the system online such the last regressor vector $\psi(k+N+1)$ is only taken into consideration for the computation of the new parameters based on the last measurements measurement $y(k+N+1)$. Any recursive estimation is based on the following structure:

$$
\hat{\theta}(k) = \hat{\theta}(k-1) + Q(k)\psi(k)(y(k) - \hat{y}(k)) \tag{4.12}
$$

such that the new estimated parameters $\hat{\theta}(k)$ is equal to the old parameters $\hat{\theta}(k-1)$ plus a correction term $Q(k)\psi(k)(y(k) - \hat{y}(k))$.
Where:

- $y(k)$: the measured output at the current time.
- $\hat{y}(k)$: the predicted output based on the measurements at $k-1$.
- $\psi(k)$: the regression vector of the last input-out data.
- $Q(k)$: a gain that has different forms and based on choosing the form of $Q(k)$, the algorithm of the recursive estimation differs. $Q(k)$ implies a weighting function which decides how much the new parameters are influenced either by the previous parameters estimate or by the new measurements.

By changing the form of $Q(k)$, the following estimations algorithms associated with RLS estimator can be adopted:

1. Unnormalized and Normalized Gradient
2. Forgetting Factor
3. Kalman Filter

In the next sections, each estimation algorithm will be explained to decide the most suitable algorithm assuming that the system is of type order one as follows:

$$G_k(z) = \frac{\alpha_k}{z + \beta_k} \tag{4.13}$$

where α and β vary within time.

4.2.1 Unnormalized and Normalized Gradient algorithm

Regarding the Unnormalized Gradient method, $Q(k)$ is considered as an adaptation gain γ where γ determines how sensitive the estimated parameters is sensitive to the variations of the dynamics of the plant. The value of γ ranges between $0 \leq \gamma \leq 1$.

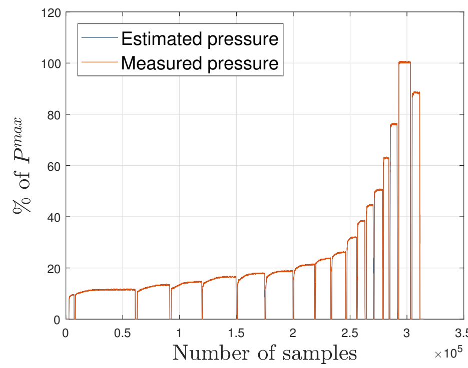
However , the disadvantage of using the unnormalized gradient algorithm that as a matter of fact it is difficult to find a range of γ that maintains the stability of the system.

In order to solve this problem , the normalized gradient method is induced such that the Adaptation gain γ is scaled by the square norm of the two-norm of the gradient vector , where the regressor matrix Ψ is considered to be the gradient vector. Hence, $Q(k)$ has the following form:

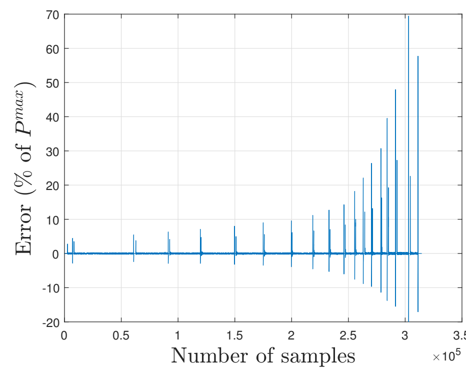
$$Q(k) = \frac{\gamma}{\|\Psi\|_2 + \epsilon} \tag{4.14}$$

Where the bias term is ϵ a very small number has to be taken into consideration in case the norm of the vector reaches to zero. This prevents any sudden jumps in the estimation of the parameters.
Different values of γ have been taken into consideration to determine the most suitable value of γ to be used in estimating the system parameters. Figures 25, 26 and 27 report the effect of variation of γ on the efficiency of the estimation.

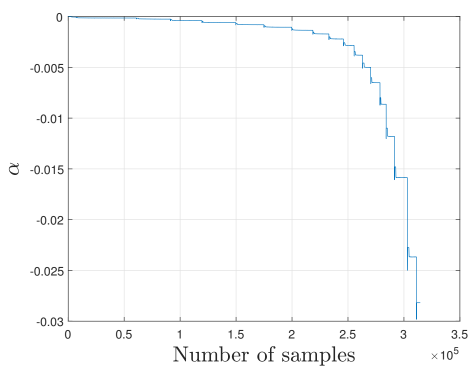
Adaptation gain γ=0.1



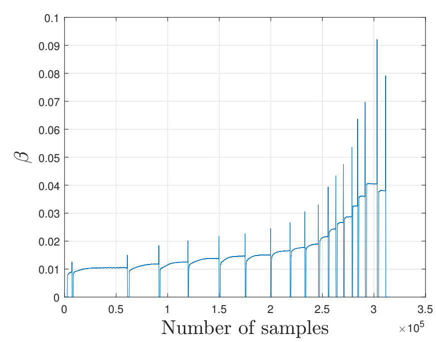
(a): Estimated pressure VS measured pressure



(b): Estimation error



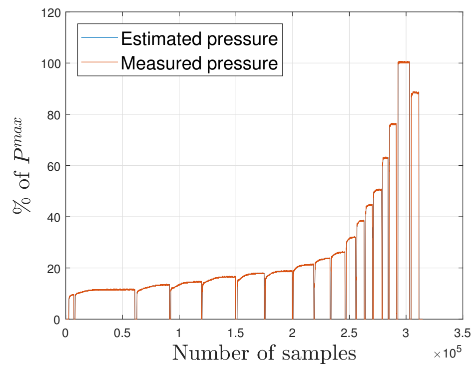
(c): Variation of α over time



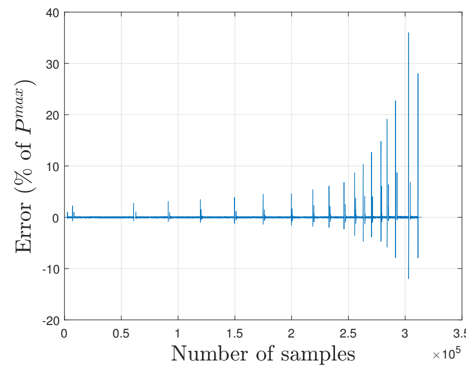
(d): Variation of β over time

Figure 25: Identification results by adopting normalized gradient algorithm such that $\gamma = 0.1$

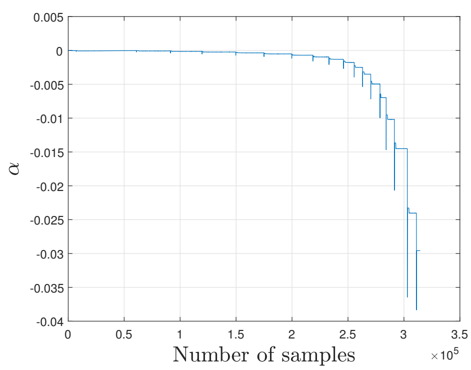
Adaptation gain γ=0.5



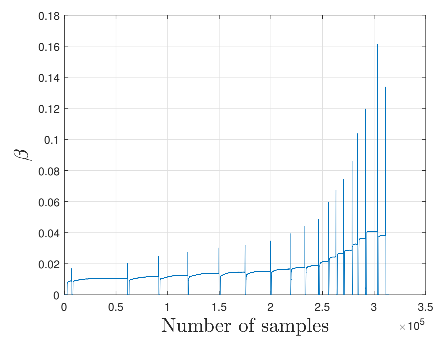
(a): Estimated pressure VS measured pressure



(b): Estimation error



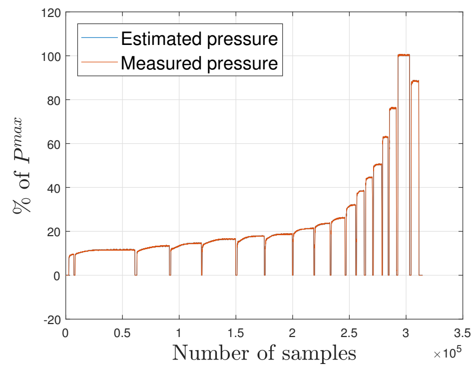
(c): Variation of α over time



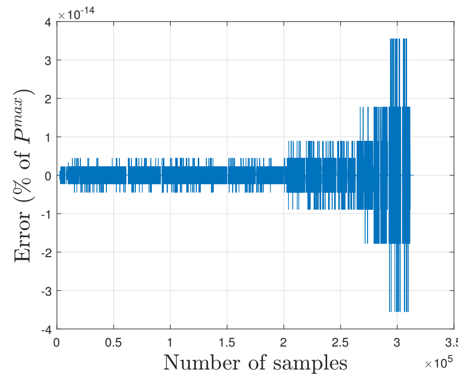
(d): Variation of β over time

Figure 26: Identification results by adopting normalized gradient algorithm such that $\gamma = 0.5$

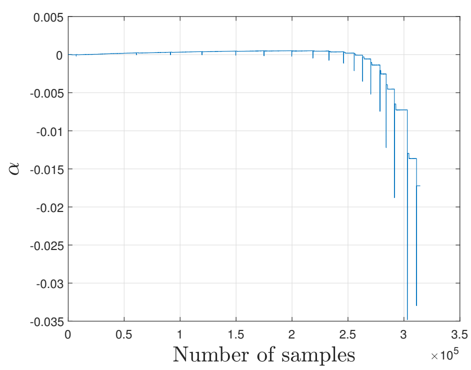
Adaptation gain γ=1



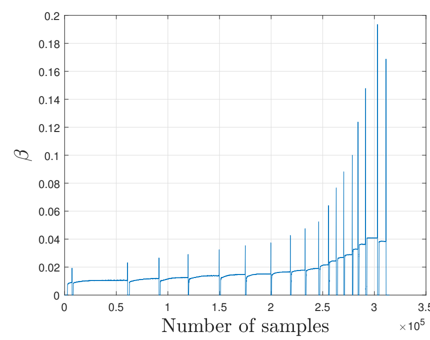
(a): Estimated pressure VS measured pressure



(b): Estimation error



(c): Variation of α over time



(d): Variation of β over time

Figure 27: Identification results by adopting normalized gradient algorithm such that $\gamma = 1$

4.2.2 Kalman Filter algorithm

Kalman filter is a widely used as a state estimator and observers. The Kalman filter is based on a group of mathematical formulations that minimizes the mean of the square error where $Q(k)$ is considered to have the following form:

$$Q(k) = \frac{P(k-1)}{R_2 + \Psi(k)^T P(k-1)\Psi(k)} \tag{4.15}$$

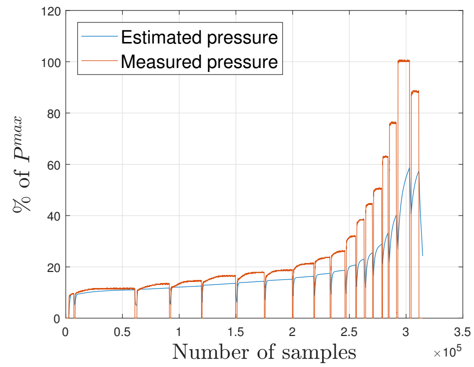
Where $P(k)$ is considered to be the covariance matrix such that:

$$P(k) = P(k-1) + R_1 - \frac{P(k-1)\Psi(k)\Psi(k)^T P(k-1)}{R_2 + \Psi(k)^T P(k-1)\Psi(k)} \tag{4.16}$$

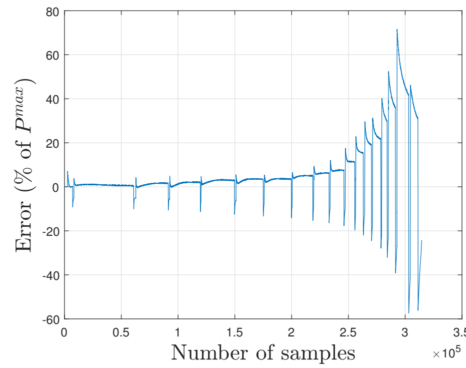
Where R_1 is considered to be the process noise covariance matrix, while R_2 is considered to be the measurement covariance matrix. Based on the selection of R_1 , $P(k-1)$ is scaled such that to remain the value of $R_2 = I$. This ensures that the state covariance matrix $P(k)$ doesn't converge to zero or to become too small.

Figures 28, 29 and 30 show the effect of selection of the noise covarinace matrix R_1 on the parameter estimation.

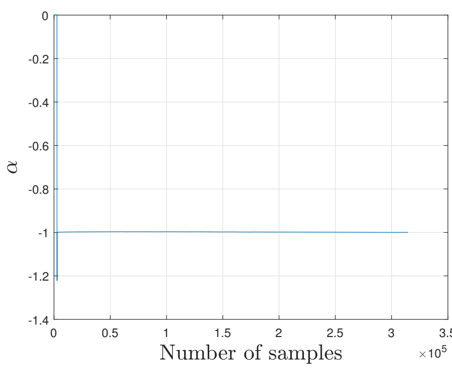
Process Noise Covariance matrix $R_1 = 0$



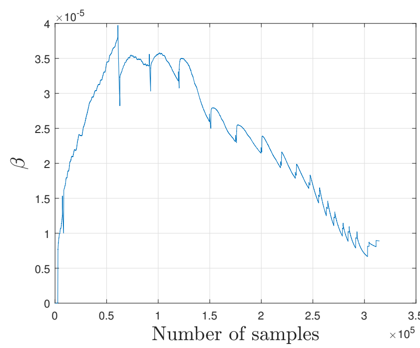
(a): Estimated pressure VS measured pressure



(b): Estimation error



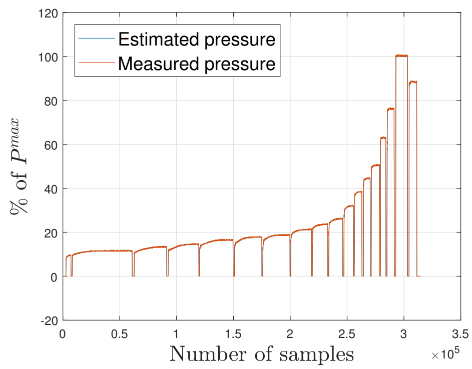
(c): Variation of α over time



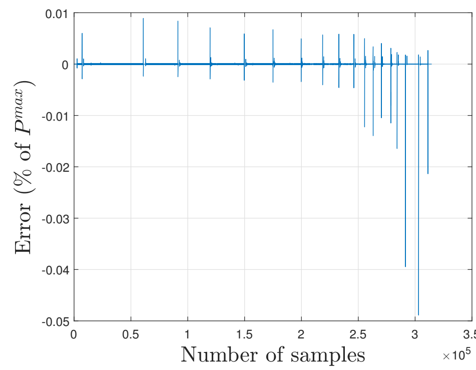
(d): Variation of β over time

Figure 28: Identification results by adopting Kalman Filter algorithm such that $R_1 = 0$

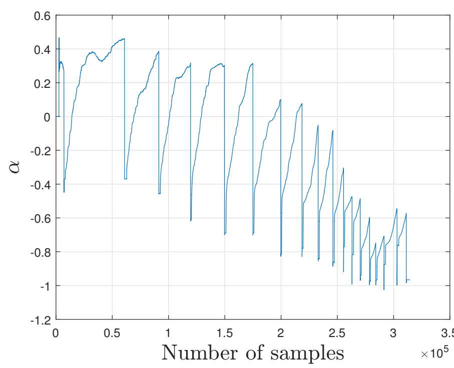
Process Noise Covariance matrix $R_1 = 0.01$



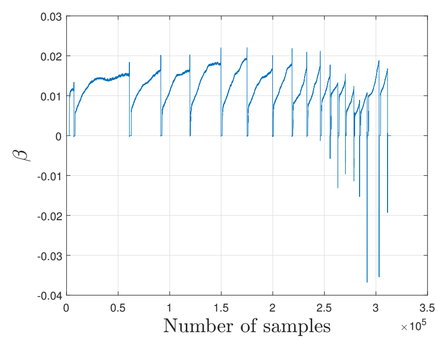
(a): Estimated pressure VS measured pressure



(b): Estimation Error



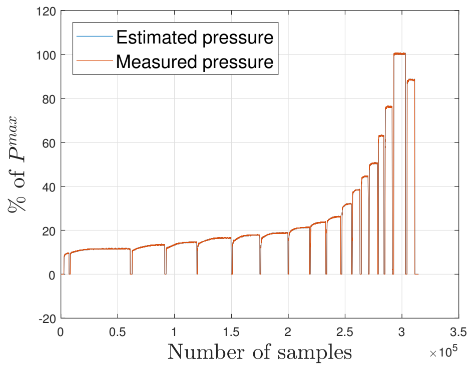
(c): Variation of α over time



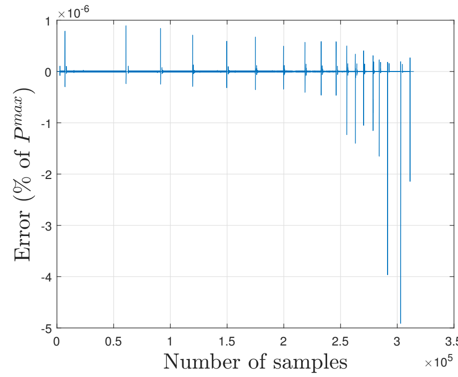
(d): Variation of β over time

Figure 29: Identification results by adopting Kalman Filter algorithm such that $R_1 = 0.01$

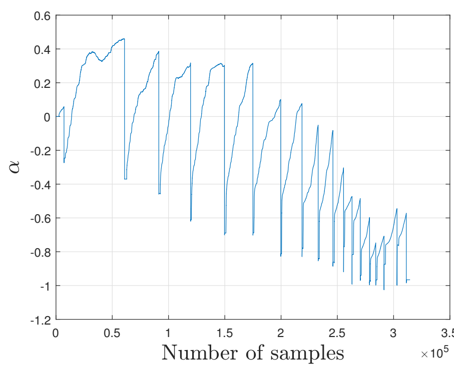
Process Noise Covariance matrix $R_1 = 100$



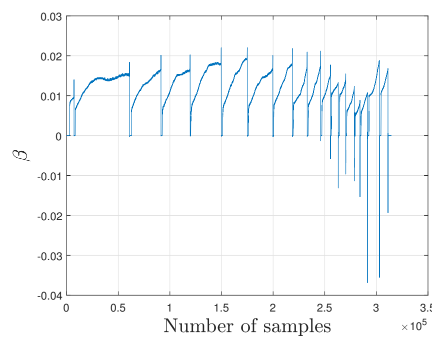
(a): Estimated pressure VS measured pressure



(b): Estimation error



(c): Variation of α over time



(d): Variation of β over time

Figure 30: Identification results by adopting Kalman Filter algorithm such that $R_1 = 100$

4.2.3 Forgetting Factor algorithm

The forgetting factor algorithm applies an exponential weighting function to the previous estimated parameters to be taken into consideration in the following estimation procedure. In other words , the forgetting factor decides the influence of the previous estimated parameters on the estimation of the new parameters in an exponential mean. Where $0 < \lambda \leq 1$ is the forgetting factor which usually ranges between 0.98 and 0.995 for efficient parameter estimation. By applying some manipulations using the matrix inversion lemma rule, $P(k)$ is formulated as follows:

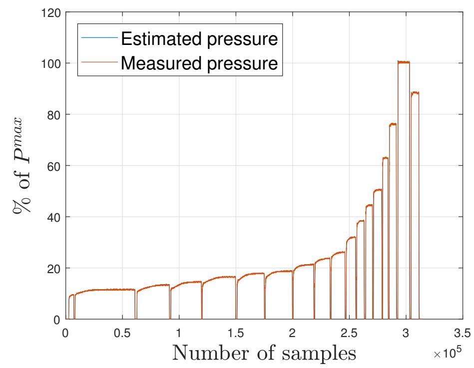
$$P(k) = \frac{1}{\lambda}(P(k-1) - \frac{P(k-1)\psi(k)\psi(k)^T P(k-1)}{\lambda + \psi(k)^T P(k-1)\psi(k)}) \tag{4.17}$$

Accordingly, $Q(k)$ can be considered as follows:

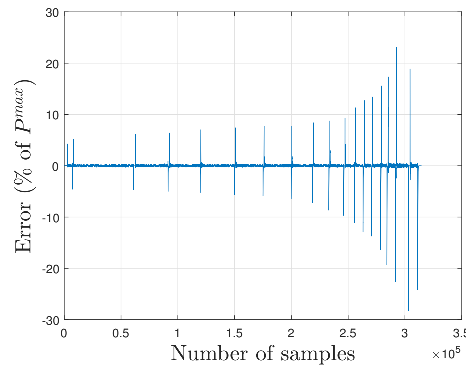
$$Q(k) = P(k) = \frac{1}{\lambda}(P(k-1) - \frac{P(k-1)\psi(k)\psi(k)^T P(k-1)}{\lambda + \psi(k)^T P(k-1)\psi(k)}) \tag{4.18}$$

Figures 31, 32 and 33 report how varying the value of the forgetting factor λ influences the efficiency of the parameter estimation.

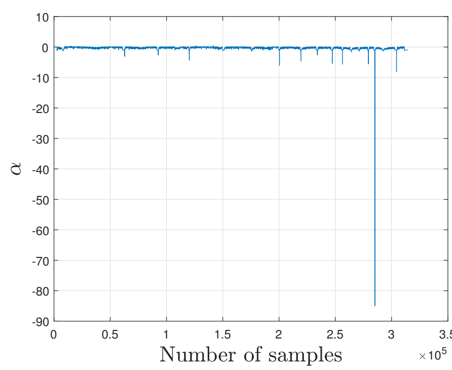
Forgetting Factor $\lambda = 0.98$



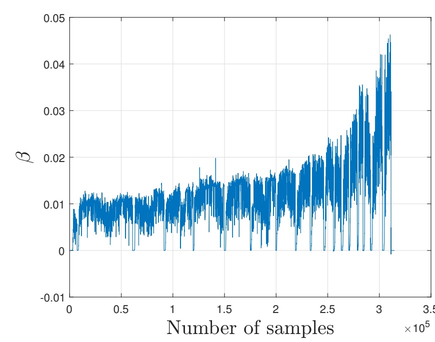
(a): Estimated pressure VS measured pressure



(b): Estimation error



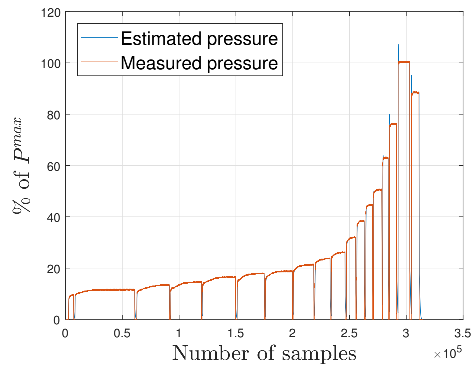
(c): Variation of α over time



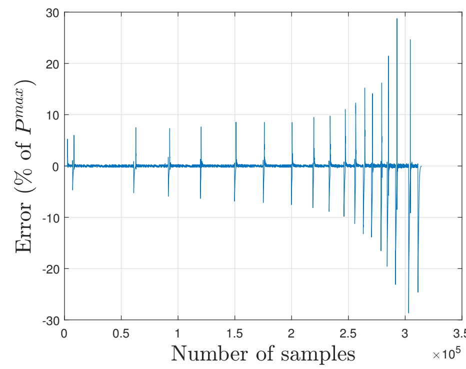
(d): Variation of β over time

Figure 31: Identification results by adopting Forgetting Factor algorithm such that $\lambda = 0.98$

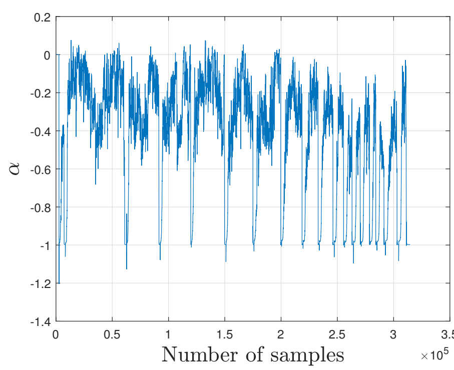
Forgetting Factor $\lambda = 0.995$



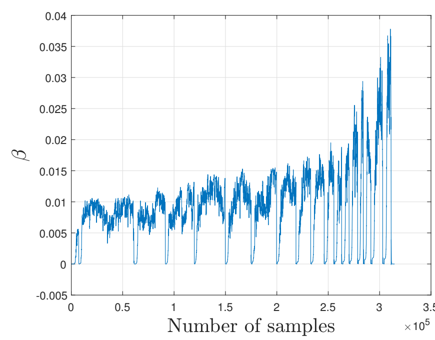
(a): Estimated Pressure VS measured pressure



(b): Estimation Error



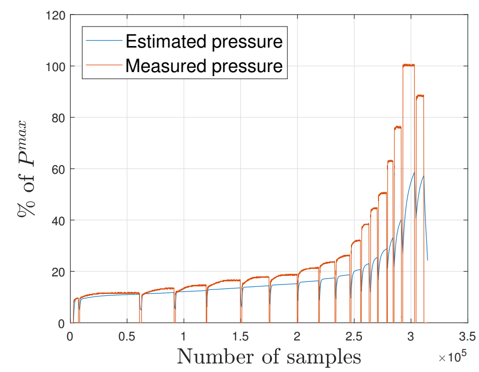
(c): Variation of α over time



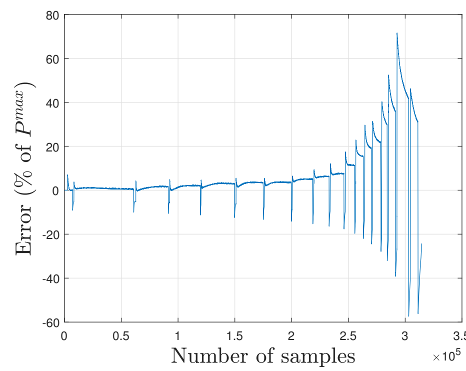
(d): Variation of β over time

Figure 32: Identification results by adopting Forgetting Factor algorithm such that $\lambda = 0.995$

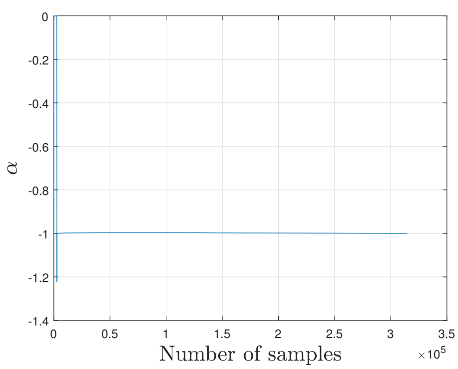
Forgetting Factor $\lambda = 1$



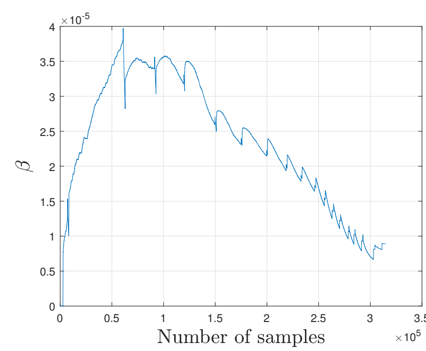
(a):Estimated pressure VS measured pressure



(b): Estimation error



(c): Variation of α over time



(d): Variation of β over time

Figure 33: Identification results by adopting Forgetting Factor algorithm such that $\lambda = 1$

As has been discussed before that the K2-actuator is of system order type one such that :

$$G(z) = \frac{\beta}{z + \alpha} \tag{4.19}$$

The value of α must vary within the unitary circle in order to guarantee stability (refer to figure 34) such that:

$$|\alpha| < 1 \tag{4.20}$$

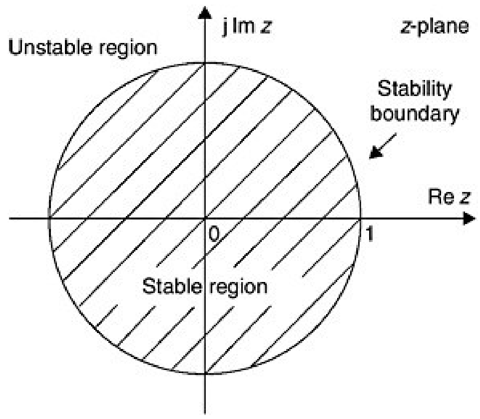


Figure 34: Stability in Discrete LTI system

By referring to figures 28, 29, 30, 31, 32 and 33, we will find that by adopting recursive estimation using Kalman filter or Forgetting Factor, the value of α exceeds the unitary circle such that $|\alpha| > 1$ in the z domain which can lead to instability during the control phase. While on the other hand, by adopting Normalized Gradient algorithm as shown in figures 25, 26 and 27 the value of α varies without exceeding the bounds of unitary circle such that $|\alpha| < 1$.
Under those circumstances, Normalized Gradient algorithm is the optimal algorithm to adopt for the recursive estimation in order to order guarantee stability and efficiency in the mean time.

4.3 MPC tuning and simulation results

In the following part, tuning of the MPC will be covered to reach the best performance. Moreover, extensive simulations have been carried out using different profiles that represent the K2-actuator in different situations.

Recalling the formulation of the cost function stated in equation (2.32), we will find that the tuning of the cost function depends on the selection of the output weighting matrix Q_p, Input weighting matrix R, Prediction Horizon H_p and Control Horizon H_c. Moreover, the adaption gain γ of the Normalized gradient adopted for the online recursive estimator has influence on the tuning of the MPC as well.

$$
\begin{aligned}
J(P(k|k), U(k)) = & \sum_{i=0}^{H_p-1} (P(k+i|k) - P_{ref}(k+i|k))^T Q_p (P(k+i|k) - P_{ref}(k+i|k)) \\
& + q(k+i|k)^T Q_{int} q(k+i|k) + u(k+i|k)^T R u(k+i|k) \\
U(k) = & \begin{bmatrix} u(k|k) & u(k+1|k) & \dots & uk + H_c - 1|k) \end{bmatrix}^T \\
& s.t \\
Pmin(k+i|k) \leq & P(k+i|k) \leq Pmax(k+i|k), \quad i = 1, 2, \cdots, H_p - 1 \\
U^{min} \leq & u(k+i|k) \leq U^{max} \quad mA, \quad i = 1, 2, \cdots, H_c - 1
\end{aligned} \tag{4.21}
$$

At the beginning, initial values for the K2-actuator parameters α and β have been guessed as an initial condition. Zero initial conditions will lead to instability of our system since $\beta \neq 0$. In addition, H_p and H_c are considered to be equal. Later on with trial and error technique, extensive simulations have carried out by tuning each design parameter until an adequate performance has been achieved. The following design parameters values have been obtained that ensures a satisfactory trade-off between the performance requirements.

$$
\begin{aligned}
\alpha(0) &= -8.4937 \times 10^{-3} \\
\beta(0) &= 0.040574 \\
Q_p &= 1 \times 10^8 \\
Q_{int} &= 0 \\
R &= 0.0001 \\
H_p &= 1 \\
H_c &= 1 \\
\gamma &= 0.01
\end{aligned} \tag{4.22}
$$

In the following we will discuss the reason behind the selection of such design parameters, in particular the choice of selecting $H_p = 1$ and $H_c = 1$.

4.3.1 Selecting prediction horizon H_p

In the literature , choosing a large prediction horizon is crucial in order to induce robustness and maintain stability of the system.

However and unusually, there is no significant difference in the response by varying the Prediction Horizon. Hence, Prediction Horizon has been considered $H_p = 1$, since large Prediction Horizon makes the computation of the optimal solution quite intensive. Accordingly, the control Horizon is set to $H_c = 1$ as well.

4.3.2 Selecting output pressure weight Q_p

Figures 35, 36, 37 and 38 report the impact of the variation of the weighting value Q_p on the tracking of the pressure for different profiles. The value Q_p changes from 10 to 1×10^8 maintaining the weights of the other design parameters as in (4.22).

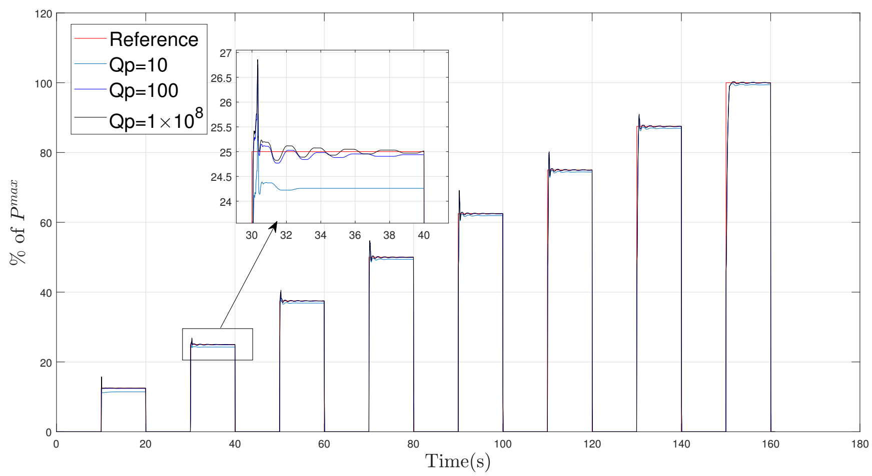


Figure 35: Pressure response for different steps references adopting Adaptive MPC for different values of Q_p

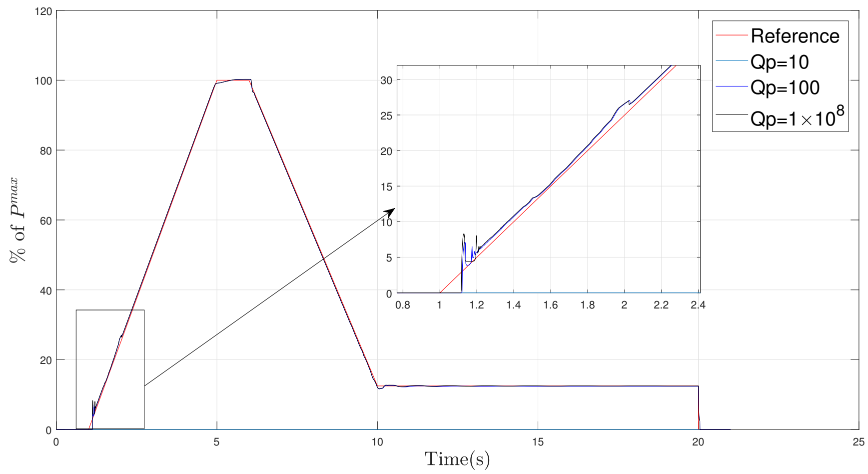


Figure 36: Pressure response for ramp trajectory adopting Adaptive MPC for different values of Q_p

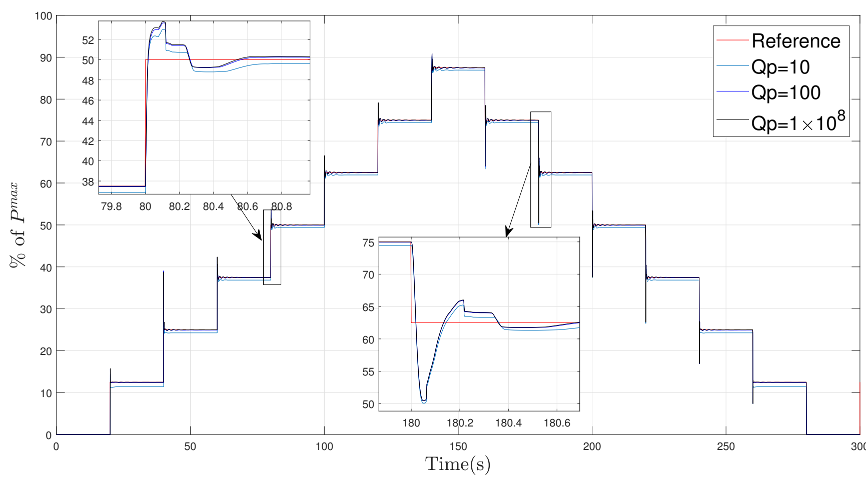


Figure 37: Pressure response for stairs trajectory adopting Adaptive MPC for different values of Q_p

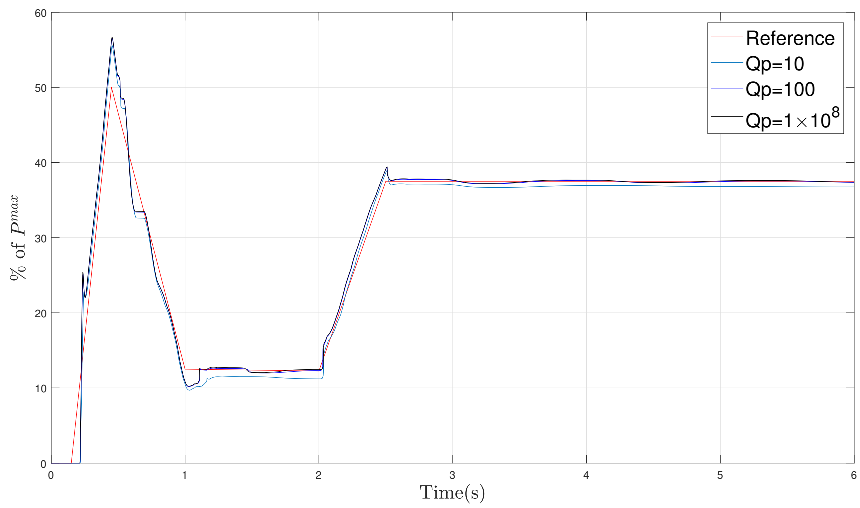


Figure 38: Pressure response for fast ramp trajectory adopting Adaptive MPC for different values of Q_p

Referring to figure 35, it is obvious that by increasing the weight of Q_p with respect to the other design parameters, the performance of tracking the pressure has been improved in terms of transient and tracking error. Moreover, with referring to figure 36, the was no response for selecting $Q_p = 10$. The reason for this behavior that manipulated variable was not able to overcome the deadzone. Hence, the weight Q_p has been taken as 1×10^8.

4.3.3 Selecting input weight matrix R

Figures 39, 40, 41 and 42 demonstrate the effect of changing the Weighting matrix R on the tracking of the pressure to the same profile applied as discussed in keeping the others design parameters constants.

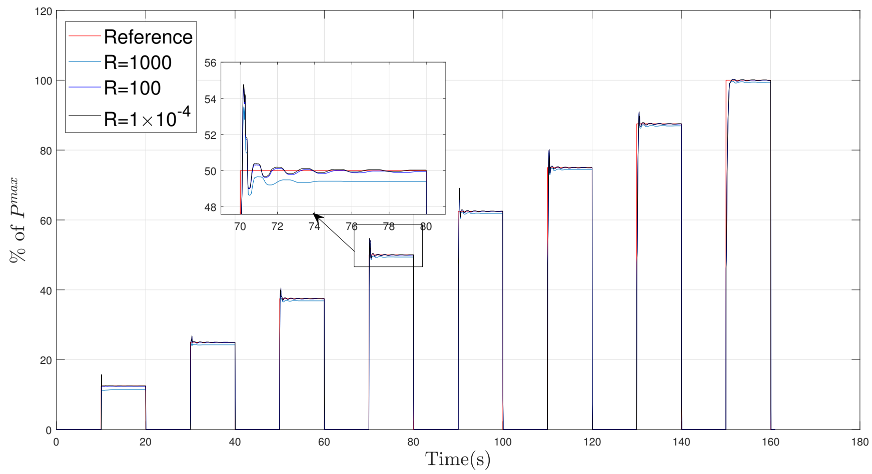


Figure 39: Pressure response for different steps references adopting Adaptive MPC for different values of R

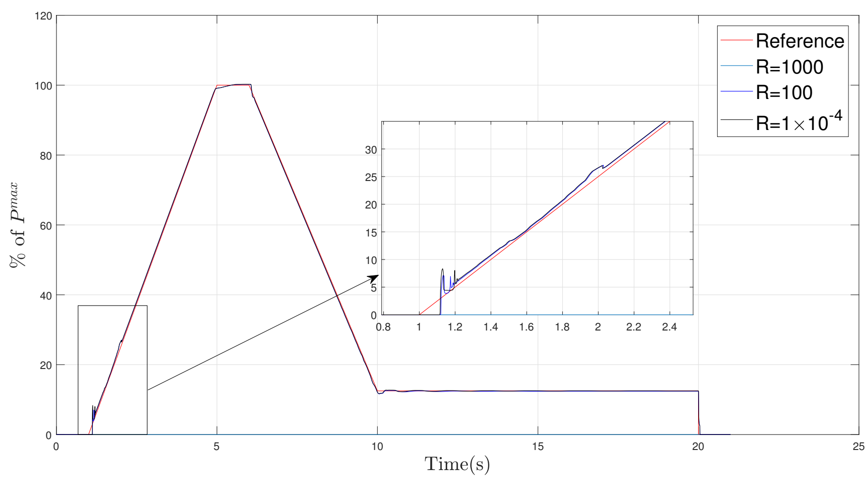


Figure 40: Pressure response for ramp trajectory adopting Adaptive MPC for different values of R

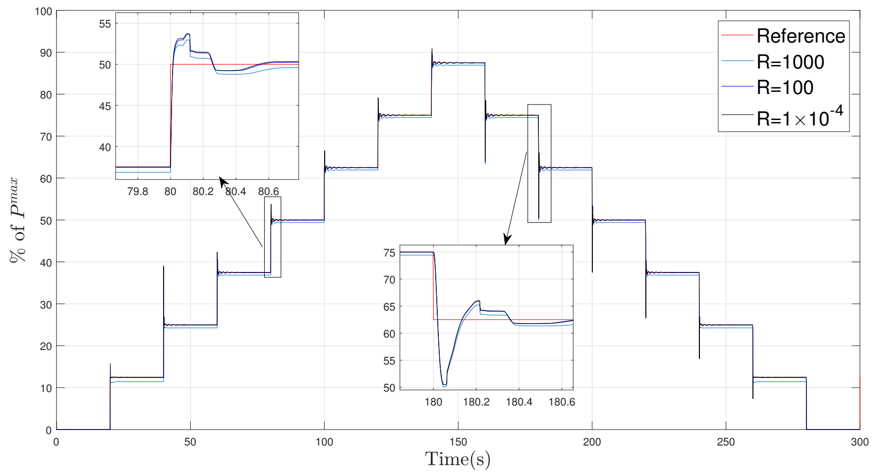


Figure 41: Pressure response for stairs trajectory adopting Adaptive MPC for different values of R

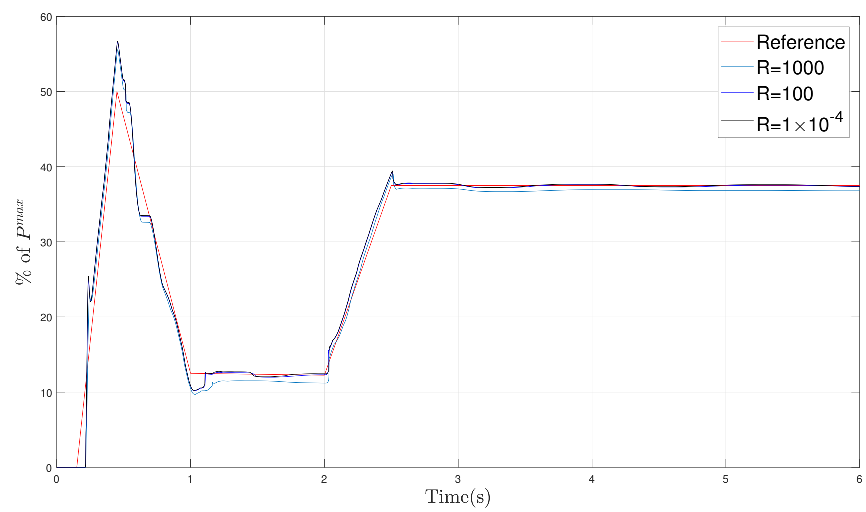


Figure 42: Pressure response for fast ramp trajectory adopting Adaptive MPC for different values of R

Conversely, with regarding to choosing the setting of Q_p , increasing the value R implies degradation on the performance in terms of transient and

Tracking error. To emphasize, high values of R increase the offset size resulting in high tracking error. Moreover, the response becomes slower increasing rising T_r and setting time $T_{settling}$.

4.3.4 Selecting adaption gain γ of the recursive estimation

As has been discussed in , Normalized Gradient algorithm using adaptation gain $\gamma = 1$ is the optimal approach to identify the K2-actuator. However after running plenty of simulations, we have discovered that $\gamma = 1$ is not the suitable value for the adaption gain to control the K2-actuator. Figure 43 represents the effect of the variation of the adaptation gain $\gamma = 0.01$, 0.05 and 0.1 on the response of the pressure of the K2-actuator to track different steps references. It is quite evident that for selecting $\gamma = 0.05$, we have maintained the overshoot of the system response for the whole working conditions.

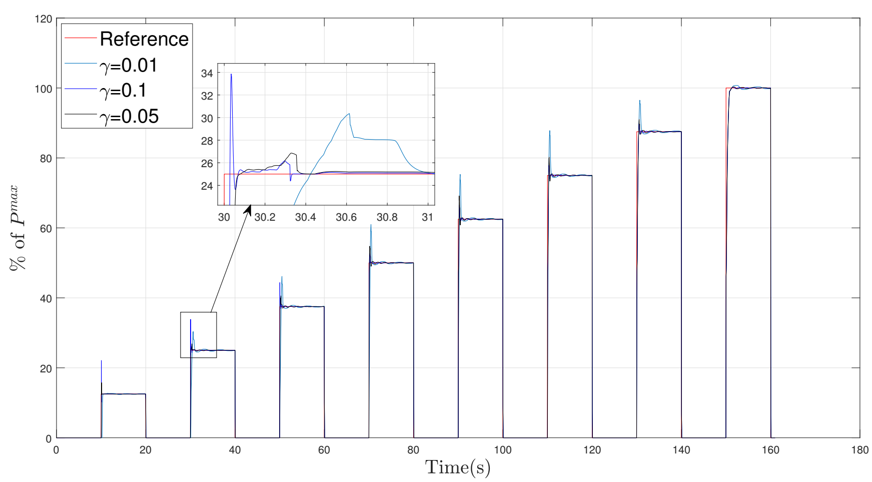


Figure 43: Pressure response for steps references based on different values of γ

To conclude, the settings of design parameters discussed in (4.22) ensure a good performance of pressure tracking for different trajectories that represent the K2-actuator in different applications. Moreover, with reference to table 2, it is quite evident that Adaptive MPC is able to satisfy the control objective explicitly for $\hat{s}$ and T_r for most of the working conditions, where tracking a step reference over 50% of P^{max} is not a common practice for K2-actuator.

Besides, regarding the $T_{settling}$ requirement, in control theory it is possible to trade-off between the control objective to reach the best performance.

Reference % of P^{max}	Overshoot (%) $\hat{s}$	rise time (ms) T_r	settling time (ms) $T_{settling}$
25	7.4	77.49	350
50	9.5	108.38	273.4
75	6.95	262.2	385.74

Table 2: Transient response at different step references for Adaptive MPC approach

Concerning the computational effort required to compute the optimal control input adopted by this architecture, such design parameters (4.22) ensure quite fast computation since we have considered prediction horizon $H_p = 1$. Moreover, the computation has been enhanced by considering an explicit saturated control input. This is achieved by solving a non constrained problem as in equation (2.26) instead of using a QP optimizer to solve a constrained optimization problem (2.25). The advantage of applying such explicit solution that the optimal control input becomes on the basis of matrices multiplication.

5 Adaptive Linear Quadratic Regulator control

In this chapter , we will discuss the architecture of Adaptive Linear Quadratic Regulator (LQR) to control the K2-actuator to track the same pre-defined trajectories that have been discussed in the previous chapter. The Estimation of the K2-actuator online is done using the same the algorithm used in the Adaptive MPC which is recursive least square based on Normalized gradient algorithm. Moreover, simulation results are provided to compare the performance between Adaptive MPC and Adaptive LQR.

5.1 Infinite horizon Linear Quadratic Control for Discrete Systems

The fundamental aspect of Linear Quadratic Regulator Control (LQR) is based on Optimal Control as well as Model Predictive Control (MPC) which its objective is to find an optimal solution by minimizing a cost function which has a quadratic form.

$$J(u) = \sum_{k=0}^{\infty} (x(k)^T Q x(k) + u(k)^T R u(k) + 2x(k)^T N u(k)) \tag{5.1}$$

The optimal solution of LQR is based on solving a Riccati differential equation associated with the infinite horizon problem such that:

$$S = A^T S A + (A^T S B + N)(B^T S B + R)^{-1}(B^T S A + N^T) + Q \tag{5.2}$$

where:

- S is the solution of the infinite horizon Riccati equation.
- A and B are the state matrices of the system
- Q and R are the design parameters to minimize the cost function

5.2 Integral action for LQR

As has been discussed in section 2.2.6, an integral action is crucial to be included in the formulation of the cost function in order to eliminate the steady state error. This can be done by adding the integral term of the tracking error as an extra state. Hence, the state space matrices can be augmented as follows:

$$\begin{aligned}
\underbrace{\begin{bmatrix} x(k+1) \\ q(k+1) \end{bmatrix}}_{\hat{x}(k+1)} &= \underbrace{\begin{bmatrix} A & 0 \\ -T_s C & 1 \end{bmatrix}}_{\hat{A}} \begin{bmatrix} x(k) \\ q(k) \end{bmatrix} + \underbrace{\begin{bmatrix} B \\ 0 \end{bmatrix}}_{\hat{B}} u(k) + \begin{bmatrix} 0 \\ Ts \end{bmatrix} r(k) \\
\hat{y}(k) &= \underbrace{\begin{bmatrix} C & 0 \end{bmatrix}}_{\hat{C}} \begin{bmatrix} x(k) \\ q(k) \end{bmatrix}
\end{aligned} \tag{5.3}$$

Consequently, the cost function and the solution of the Riccati equation can be modified using the augmented state matrices as follows:

$$J(u) = \sum_{k=0}^{\infty} (\hat{x}(k)^T \hat{Q} \hat{x}(k) + u(k)^T R u(k) + 2\hat{x}(k)^T N u(k)) \tag{5.4}$$

The optimal solution of LQR is based on solving a Riccati differential equation associated with the infinite horizon problem such that:

$$S = \hat{A}^T S \hat{A} + (\hat{A}^T S \hat{B} + N)(\hat{B}^T S \hat{B} + R)^{-1} (\hat{B}^T S \hat{A} + N^T) + \hat{Q} \tag{5.5}$$

where $\hat{Q} = \begin{bmatrix} Q_x & 0 \\ 0 & Q_{int} \end{bmatrix}$

5.3 Tracking a reference

After solving the Riccati equation ,the optimal solution U^0 can be found by applying a feedback contribution such that:

$$\begin{aligned}
\hat{K}(k) &= (\hat{B}^T(k) S(k) \hat{B}(k) + R)^{-1} (B^T(k) S(k) \hat{A}(k) + N^T) \\
\hat{K}(k) &= \begin{bmatrix} K_x & K_q \end{bmatrix}
\end{aligned} \tag{5.6}$$

$$U^0 = -\hat{K}\hat{x}(k) \tag{5.7}$$

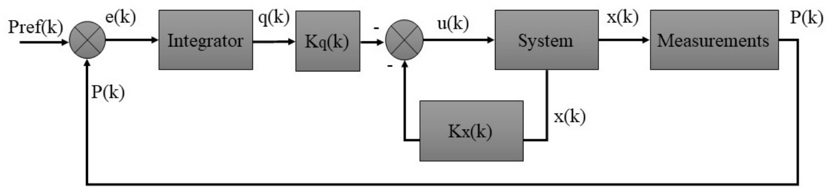


Figure 44: Adaptive LQR scheme

Since we are adopting an adaptive approach to estimate the system online, at each new sampling time Ts a new Riccati equation is solved to compute the feedback matrix $\hat{K}$ that is contributed for the computation of the optimal solution. The computation of $\hat{K}$ matrix has been carried out by exploiting the dlqr MATLAB function.

5.4 Simulation results for Adaptive LQR

After carrying out several simulations and following trail and error technique , the following design parameters have been obtained that guarantee a good trade-off to track the predefined trajectories of the K2-actuator.

$$\begin{aligned} Q_x &= 10 \\ Q_{int} &= 5 \\ R &= 1 \\ \gamma &= 0.1 \end{aligned} \tag{5.8}$$

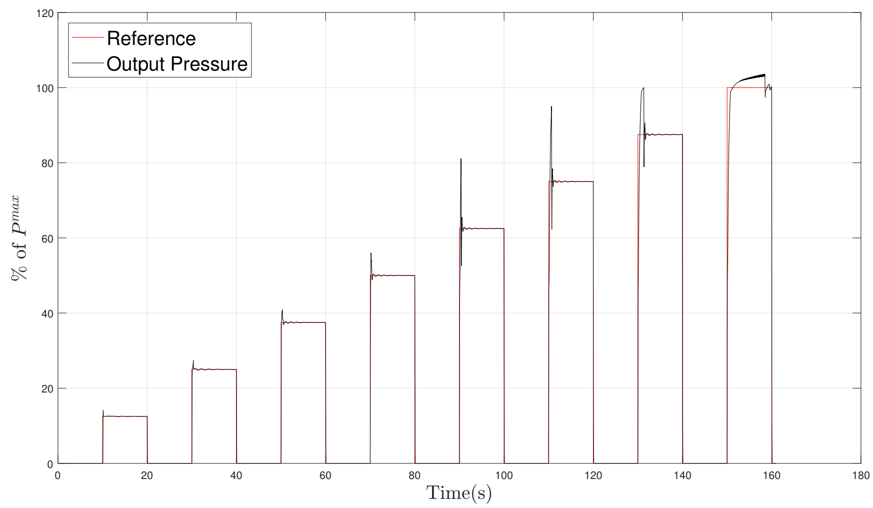


Figure 45: Pressure response for different steps references adopting Adaptive LQR architecture

Reference % of P^{max}	Overshoot (%) $\hat{s}$	rise time (ms) T_r	settling time (ms) $T_{settling}$
25	9.6	117.63	373.84
50	12	125.24	181.86
75	26.66	282.5	743.87

Table 3: Transient response at different step references for Adaptive LQR approach

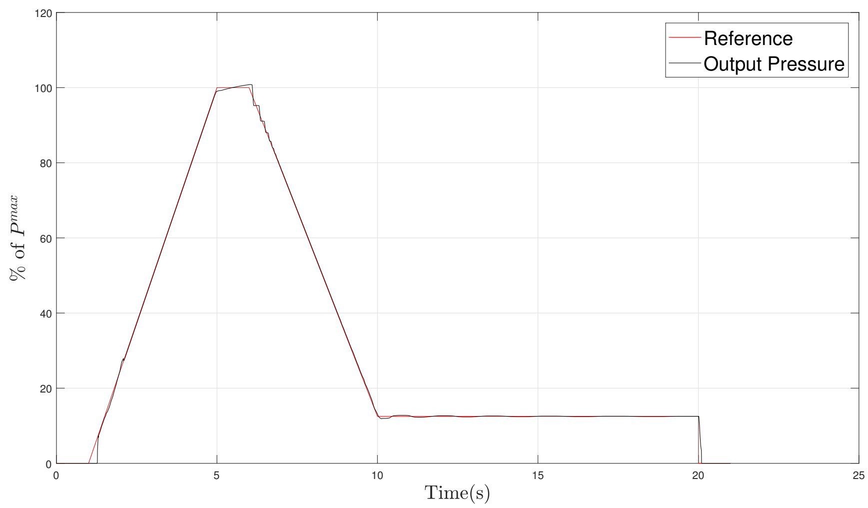


Figure 46: Pressure response for ramp trajectory adopting Adaptive LQR architecture

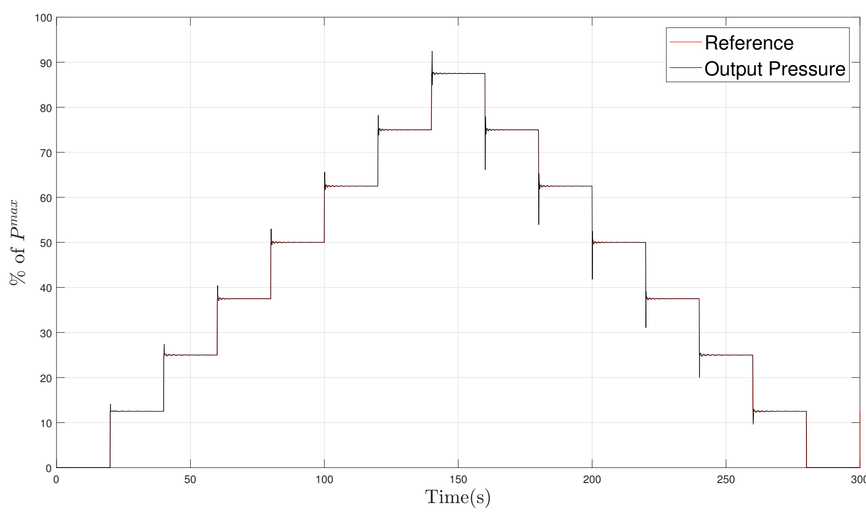


Figure 47: Pressure response for stairs trajectory adopting Adaptive LQR architecture

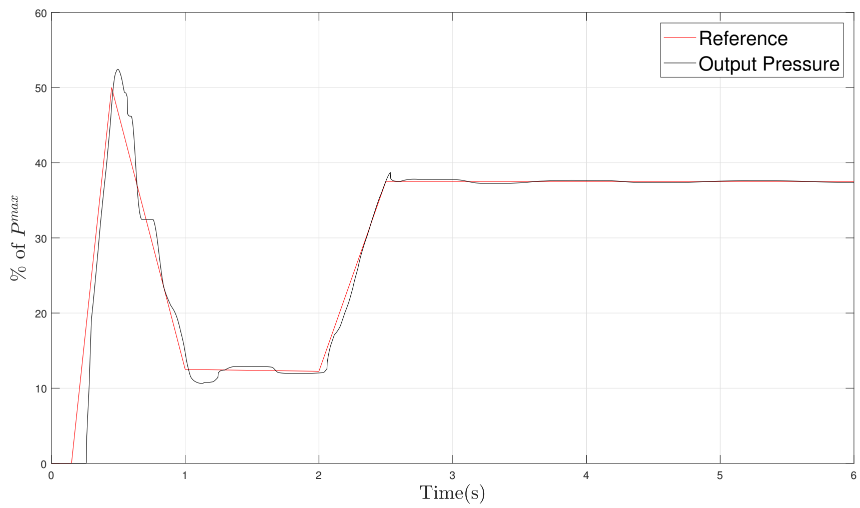


Figure 48: Pressure response for fast ramp trajectory adopting Adaptive LQR architecture

With Reference to figures 45, 46, 47 and 48, the approach for adopting Adaptive Linear Quadratic Regulator (LQR) shows an great performance to track the pressure trajectories of the K2-acutator especially for the stairs trajectory as shown in figure 47. However, regarding the steps references as shown in figure 45, there is an apparent overshoot that exists for tracking steps over 50% of P^{max} (refer to table 3). Nevertheless, as has been discussed in 4.3.4, it is not a common situation to track step reference over 50% of P^{max}.

Considering the deployment of Adaptive LQR on the vehicle's hardware, such architecture is not ideal to be equipped for controlling the K2-actuator since we need to solve an infinite horizon Riccati equation at each sampling time which is not affordable by the vehicle's hardware.

In conclusion and with reference to table 4, it is quite evident that Adaptive MPC is the most adequate approach to control the K2-actuator. Adaptive MPC showed a good performance allowing trading off between control objectives. Moreover, the advantage of deploying Adaptive MPC on the vehicle's hardware over the other control methods is that Adaptive MPC consumes less energy for computing the optimal control input since an explicit solution has been considered instead of exploiting a QP solver which has reduced the computational effort and accelerated the simulation results.

Steps (% of P^{max})	Multi-MPC ,Multi-Model			Adaptive MPC			Adaptive LQR		
	($\hat{s}$%)	($T_r ms$)	($T_{set} ms$)	($\hat{s}$%)	($T_r ms$)	($T_{set} ms$)	($\hat{s}$%)	($T_r ms$)	($T_{set} ms$)
25	4.7	111	111	7.4	77.49	350	9.6	117.63	373.84
50	3.19	192	192	9.5	108.38	273.4	12	125.24	281.86
75	2.7	581	581	6.95	262.2	385.74	26.66	282.5	743.87

Table 4: Comparison of the performance achieved by each control method

6 Conclusion and future work

6.1 Conclusion

This thesis emphasized on controlling the pressure applied by the K2-actuator on the even gear clutch to attenuate the torque interruption and guaranteeing smoothness during the gear shift phase. Different control architectures have been exploited to conclude which one is ideal in terms of satisfying the performance requirements to be devoted for controlling the pressure of the K2-actuator.

In fact, the biggest challenge to control the K2-actuator was to be able first to build up a model that represents the dynamics of the system. A possible representation of the K2-actuator is to model it as a Hammerstein system since non-linearity plays a big role in the system due to the presence of a dead zone in our system.

In chapter 3, multiple MPC based on multiple Hammerstein models approach has been introduced. The reason behind considering this approach is to control the K2-actuator in different working conditions. This architecture showed a good performance, particularly to achieve the overshoot requirement. However, due to the structure complexity of the Multiple MPC which has employed 11 controller to control the whole working conditions. Moreover, because of the long prediction horizon consumed by each controller, the computation of the optimal control input was quite intensive which is not recommended for real time automotive applications. Hence, adaptive control was proposed to control the K2-actuator instead of Multiple MPC approach.

In chapter 4, Adaptive MPC has been employed to control the pressure of the K2-acutator using recursive least square algorithm to estimate the system online. Moreover, the selection of the estimation algorithm associated with RLS have been discussed concluding that Normalized gradient was the optimum algorithm to employ such that it provides very low estimation error and guarantees system stability. Further, Extensive simulations have been presented to demonstrate the selection of the design parameters of the Adaptive MPC. Moreover, an explicit solution has been considered to compute the optimal control input which has improved the computational effort. Indeed, Adaptive MPC showed outstanding results for tracking the pressure satisfying the overshoot and rising time requirements.

Finally, In chapter 5 Adaptive LQR approach has been considered using the same estimation method of Adaptive MPC as well. Adaptive LQR is ideal for tracking stairs references and it showed good performance for tracking steps reference up to 50% of the maximum pressure operating range. However, due to hardware limitation of the electronic control unit, Adaptive LQR is not the ideal controller to deploy since a riccati equation needed to be solved online equation at each sampling time which can not be implemented in our case.

To conclude, different control architectures have been introduced to control the K2-actuator to determine which control method is superior for deployment, taking into consideration the performance achieved and the computational effort of each control method. Adaptive MPC showed outstanding performance allowing trading-off between control objectives. Moreover, the power of deploying Adaptive MPC in comparison to other control methods is the low computation effort needed by Adaptive MPC since an explicit solution has been employed to compute the optimal control input which has reduced the computational effort and accelerated the simulation results..

6.2 Future Work

A possible way to control the K2-actuator is to track the position of the clutch. However, due to design limitations of the DDCT, position sensor is not able to be deployed on the system to provide position feedback of the clutch to the controller. A possible solution to overcome this problem is to provide a position feedback to the controller by employing a virtual sensor through exploiting Neural Network. The Neural Network will provide an estimate of the position using finite measured input-output data which can be used then as a feedback signal to the controller.

Further after implementing the virtual sensor, an appealing way to control the K2-actuator is to consider a nested feedback loops approach such that both pressure and position feedback are taken into consideration in the control law. Nested feedback loops may enhance the performance of tracking the position and pressure satisfying the performance requirements ensuring smoothness during the gear shifting phase.

Appendix A Simulink implementations

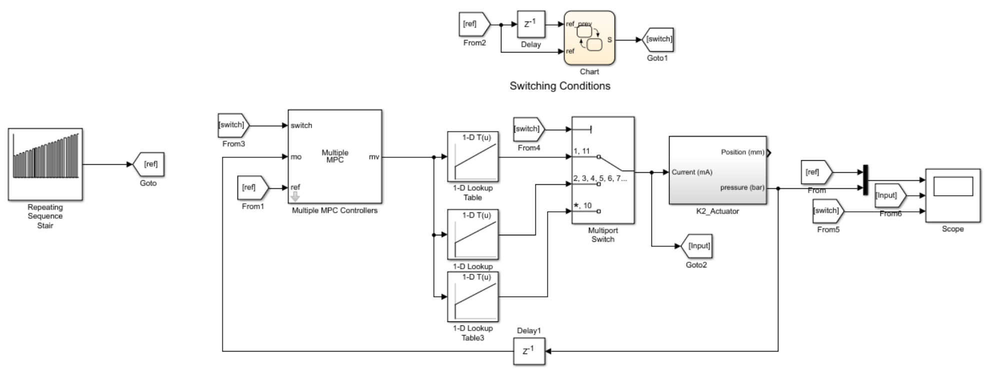


Figure 49: Simulink scheme for Multi-MPC based on Multi-Models approach

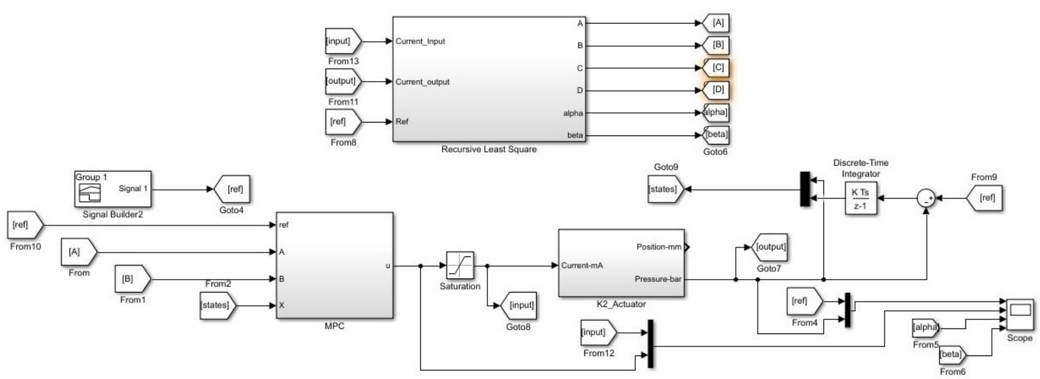


Figure 50: Simulink scheme for Adaptive MPC approach

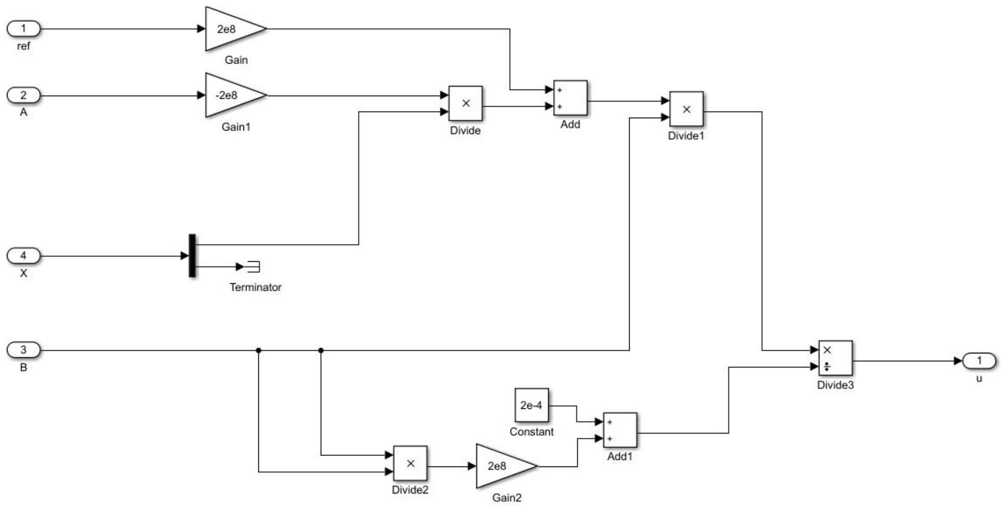


Figure 51: Simulink scheme for Adaptive MPC explicit solution

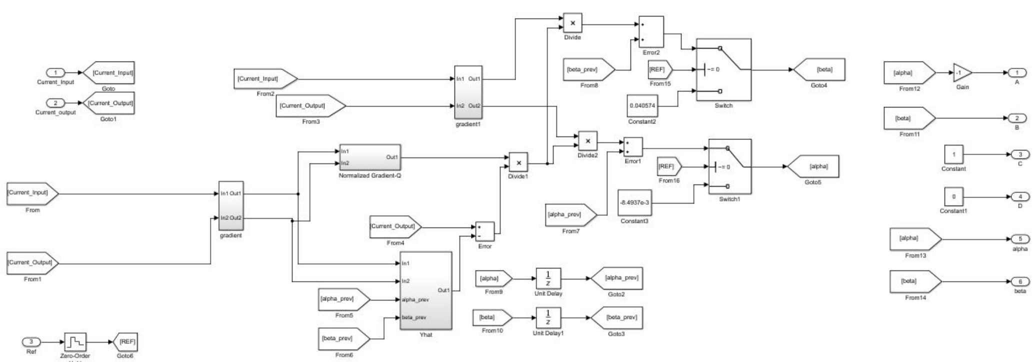


Figure 52: Simulink scheme for recursive least square (RLS) estimator

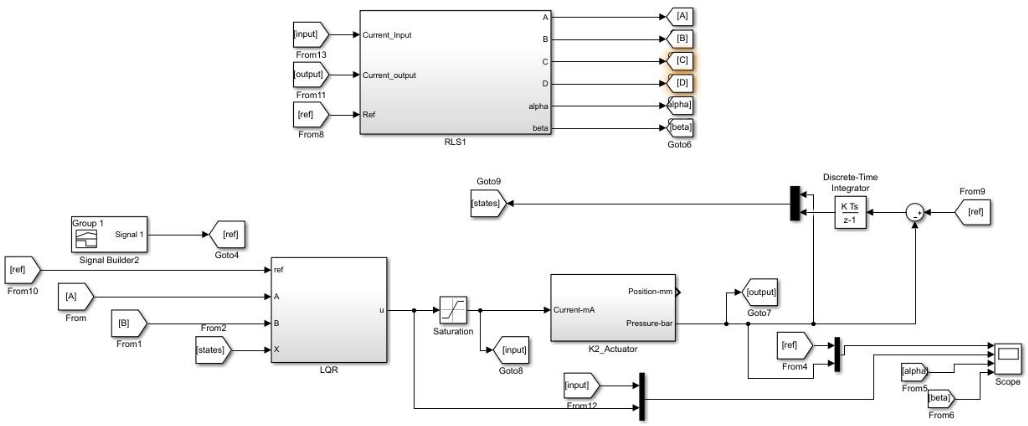


Figure 53: Simulink scheme for Adaptive LQR approach

References

[1] Cimmino Vafidis, Constantinos and Francesco. Fpt's high torque density dual dry clutch transmission (htd-ddct), Fiat Powertrain Research and Technology.

[2] Maurizio Zoppi, Claudio Cervone, Gerardo Tiso, and Francesco Vasca. Software in the loop model and decoupling control for dual clutch automotive transmissions. In Systems and Control (ICSC), 2013 3rd International Conference on, pages 349354. IEEE, 2013.

[3] Baliva S. (2015). Micro-slip control of a Dry Dual Clutch Transmission system using Model Predictive Control techniques.

[4] Gisbert Lechner, Harald Naunheimer (1999), Automotive Transmissions: Fundamentals, Selection, Design and Application .

[5] Ljung, L. (1999). System identification theory for the user (2nd ed., Prentice Hall information and system sciences series). Upper Saddle River: Prentice Hall PTR.

[6] Sderstrm, T., & Stoica, P. (1989). System identification (Prentice- Hall International series in systems and control engineering). New York: Prentice- Hall.

[7] Ljung, L. (2001). Black-box Models from Input-output Measurements.

[8] Young, P. (1984). Recursive estimation and time- series analysis an introduction (Communications and Control Engineering Series). Berlin: Springer.

[9] Vichugov, V.N., Tsapko, G.P., & Tsapko, S.G. (2005). Application of reinforcement learning in control system development. Science and Technology, 2005. KORUS 2005. Proceedings. The 9th Russian-Korean International Symposium on, 732-733.

[10] Now A., Vrancx P., De Hauwere YM. (2012) Game Theory and Multiagent Reinforcement Learning. In: Wiering M., van Otterlo M. (eds) Reinforcement Learning. Adaptation, Learning, and Optimization, vol 12. Springer, Berlin, Heidelberg.

[11] Gill, P. R.; Murray, W.; and Wright, M. H. "The Levenberg-Marquardt Method." 4.7.3 in Practical Optimization. London: Academic Press, pp. 136-137, 1981.

[12] D. Q. Mayne and J. B. Rawlings. Model Predictive Control: Theory and Design, volume 1. Nob Hill Publishing, 2012.

[13] S. Boyd and L. Vandenberghe. Convex optimization. Cambridge university press, 2004.

[14] G. Pannocchia, J. B. Rawlings, and S. J. Wright. *Fast, large-scale model predictive control by partial enumeration. Automatica*, 43(5):852-860, 2007.

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

[16] Y.Wang and S. Boyd. Fast model predictive control using online optimization. Control Systems Technology, IEEE Transactions on, 18(2):267-278, 2010.

[17] A. Bemporad, M. Morari, V. Dua, and E. N. Pistikopoulos. The explicit linear quadratic regulator for constrained systems. Automatica, 38(1):3-20, 2002.