POLITECNICO DI TORINO

MASTER OF SCIENCE IN MECHATRONIC ENGINEERING

Master Degree Thesis

Dry Dual Clutch Transmission Systems: a Model Predictive Approach to Clutch Slipping Control

Supervisors:
Prof. Carlo Novara
Ing. Emanuel Corigliano

Author:
Danilo Voria

July 2019

Ringraziamenti

La scrittura della tesi rappresenta la parte finale di un percorso universitario. Nel mio percorso molte sono state le persone che mi hanno supportato e a loro voglio dedicare qualche parola.

Ringrazio la mia famiglia, in modo particolare i miei genitori che mi hanno permesso di intraprendere questo percorso e non mi hanno mai fatto mancare il loro aiuto, che si è espresso nelle forme più varie ed è stato indispensabile per il raggiungimento di questo traguardo.

Ringrazio Giulia, la persona con cui più da vicino ho condiviso tutti i momenti, felici e non, di questi anni.

Un grazie va anche a tutte le persone lontane che in questi anni hanno avuto un pensiero per me facendomi capire di essermi vicine nonostante la distanza.

Rivolgo un ringraziamento particolare al mio relatore Prof. Carlo Novara e al correlatore Ing. Emanuel Corigliano, che mi hanno dato la possibilità di prendere parte ad un progetto che ha da subito suscitato in me un forte interesse.

Abstract

In this thesis project, developed in collaboration with Centro Ricerche Fiat (CRF), a Model Predictive Control approach is proposed to control the Dry Dual Clutch of an Automated Manual Transmission (AMT) system during its slipping phase. In particular, the analysis is focused on the Dry Dual Clutch Transmission (DDCT) developed by Fiat Power-train Technologies.

During gear shifting operations, the clutch disks that are about to close do not immediately take the same speed: the controller's goal is to ensure that the speed difference between the disks (slipping speed) follows a given profile and reaches the null value in a smooth way.
The results taken as a reference (and therefore to be improved) are those obtained with a PID controller designed by CRF. This controller was the one that guaranteed best performances among other controllers developed using different techniques (like loop shaping or H_∞ control).

Firstly, the Model Predictive Controller is simulated using the same configuration used with the PID: only the clutch torque is an input computed by the controller, with the engine torque considered as a known, external non-modifiable input, and only the slipping speed is controlled as output.
Then, with the aim of improving the obtained results, some changes are made: the engine torque is considered as another manipulated input rather than a disturbance, and other two outputs are controlled: the integral error of the slipping speed and the drive shaft torsion speed. A successive tuning of the controller is also performed in order to compensate for a disturbance acting on the engine torque: this disturbance in the real world can be identified with the driver action.

The simulated results show that in the configuration with only one manipulated variable the slipping speed reaches the null value in a faster way with respect to what was possible to achieve using a PID controller, while the second type of configurations ensures a slight decrease of the oscillations.

In conclusion, the choice to use the Model Predictive approach for the clutch slipping control turns out to be advantageous in terms of performances, but also for the flexibility proper of the MPC, that can be easily extended to MIMO systems, allowing to control multiple outputs and to compensate for external disturbances.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The efforts of the automotive industry are directed towards many different fields. One of them which is gaining more and more importance is that related to the improvements regarding fuel consumption and driving comfort. These two aspects are becoming essential for the commercial success of a vehicle, and the major companies continuously try to achieve better results by developing advanced transmission and powertrain systems.

A transmission is a machine in a power transmission system, which provides controlled application of the power. As far as a motor vehicle is concerned, transmission can be seen as the interface through which the driver communicates with the engine and the vehicle.

There are essentially two types of transmission: Manual Transmission (MT) and Automatic Transmission (AT).
The Manual Transmission is widespread in the European automotive market, since it guarantees high efficiency, low costs and full control to the driver. In the Japanese and US market, the Automatic Transmission is preferred, since comfort and ergonomics are privileged. In order to exploit the best of both technologies, the Automated Manual Transmissions (AMT) have been introduced.

There are two types of AMTs: single-clutch and dual-clutch. Single-clutch AMTs are older, lurch-prone and generally unpleasant. Dual-clutch AMTs, on the other hand, are designed to eliminate lurching, and the best units provide incredibly quick yet perfectly smooth shifts. Most AMT-equipped cars use dual-clutch technology.
AMTs tend to yield better fuel economy and acceleration than regular automatics. The reason is that AMTs are more efficient, that is, they allow more

of the engine's energy to flow directly to the wheels. The big advantage of the AMT lies in its ability to combine the fuel economy and performance of a true manual with the everyday convenience of an automatic.
On the other hand, there are several issues concerning AMT, such as the low-speed behaviour. The driver comfort is highly affected by the way the clutch engagement occurs: it is important that the engagement occurs smoothly. A possible strategy to achieve this objective is to reduce the driveline oscillations by applying a proper control strategy.

This thesis deals with the clutch engagement problem, in particular the analysis is focused on the Dry Dual Clutch Transmission (DDCT) system and the problem of controlling the dual clutch during its slipping phase. The technique used here to control the system is Model Predictive Control (MPC).

1.1 Thesis outline

In the second chapter, an overview of the DDCT system and its components is presented.

In the third chapter a linear model of the system is illustrated along with the mathematical description of the system in terms of state equations and state-space representation.

In the fourth chapter the Model Predictive Control technique is described from a theoretical point of view. The main reasons why MPC is used are illustrated together with the advantages that this technique can bring by being applied to the DDCT system.

In the fifth chapter the controller design is illustrated and the simulation results are shown. Different models of the system are used for control purposes and for each of them a detailed analysis is performed.

The sixth chapter introduces the problem of controlling the system by means of the explicit MPC.

The final chapter exhibits conclusions about the obtained results and additional considerations.

Chapter 2

Overview of the Dry Dual Clutch Transmission System

2.1 Introduction

Dual Clutch Transmission is essentially implemented in two ways: either two wet multi-disc clutches which are bathed in oil for cooling (WDCT), or two dry single-disc clutches (DDCT). Wet Dual Clutch Transmissions (WDCT) are able to provide torque values up to 350 Nm, so they are used in high torque applications where there is more energy to handle and more heat to dissipate, while the dry-clutch design is generally suitable for smaller vehicles with lower torque outputs.
The advantage of employing dry clutches with respect to the WDCT is that the dry-clutch versions offer an increase in fuel efficiency, due to the reduction of pumping losses of the transmission fluid in the clutch housing. This thesis takes into account the Dry Dual Clutch Transmission (DDCT) developed by Fiat Power-train Technologies.

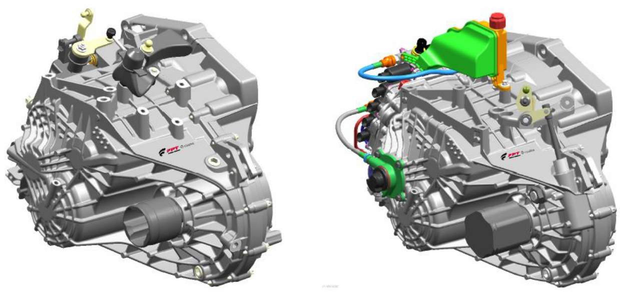


Figure 2.1: C635 MT and DDCT versions.

2.2 The C635 Dry Dual Clutch Transmission System

The Dry Dual Clutch Transmission system developed by Fiat Powertrain Technologies is part of the new C635 transmission family. This new transmission family consists of a range of Manual, All Wheel Drive and DCT transversal, 6-speed transmissions with a maximum input torque of 350 Nm and output torque of 4200 Nm. The DCT version of this transmission is the highest rated amongst the double dry dual clutch transmissions in the market.

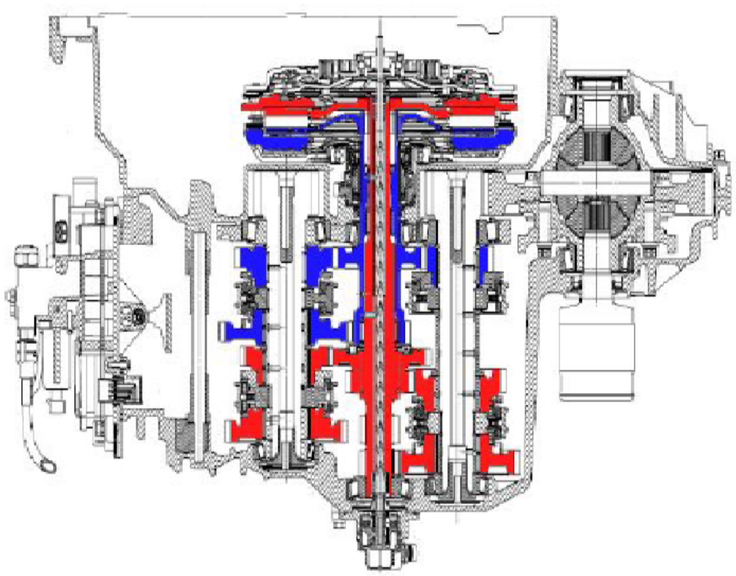


Figure 2.2: Cross section of the C635 DDCT.

As shown in Figure 2.2, the C635 DDCT system architecture is composed by three input shafts and contained in a two piece aluminium structure.

2.2.1 Dry Dual Clutch Unit

During the shifting process the transmitted torque is obtained by the overlap of the engagement of the closing clutch and the release of the opening clutch. The clutch K1 is normally closed as in conventional manual transmissions and it is position-controlled by means of a contactless linear position sensor integrated in the rear hydraulic piston actuator. The even gear clutch K2 is normally open and is controlled in force through hydraulic pressure given by the Concentric Slave Cylinder (CSC). The two clutches act on a centre plate which, together with the two pressure plates, has been dimensioned according to the thermal dissipation characteristics required for the most critical vehicle/engine applications foreseen.

2.2.2 The Electro-Hydraulic Actuation System

A dedicated, sealed, hydraulic oil circuit is in charge of actuating electro-hydraulically the C635 DDCT clutches and gear shifting mechanisms. This choice was mainly based on system efficiency and compactness considerations. The main components of the actuation system are:

- **Hydraulic Power Unit:** consists of an electrically driven high pressure pump and accumulator.
- **Actuation Module:** includes the control solenoid valves, gear shift actuators and sensors. It can be divided in 4 distinct double action pistons operating the gear engagement forks, one "shifter" spool which selects the piston to be actuated and 5 solenoid valves of which 4 are pressure proportional (PPV) and one is flow proportional (QPV).

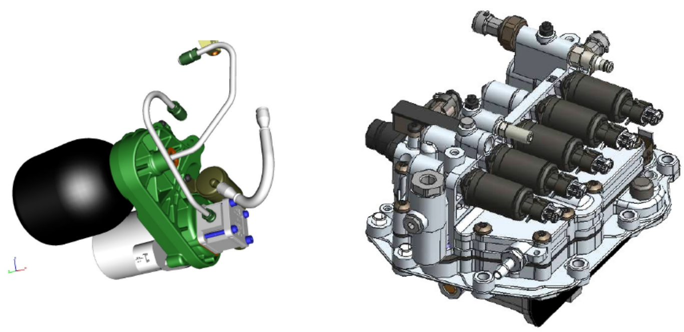


Figure 2.3: Hydraulic Power Unit and Complete Actuation Module.

In particular, in the actuation module two PPVs actuate the gear engagement piston which is selected by the spool valve actuated by the third PPV. The fourth PPV and the QPV control respectively the clutches K2 and K1.
In the Actuation Module are also included 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 below represents the overall hydraulic circuit of the C635 DDCT Actuation System:

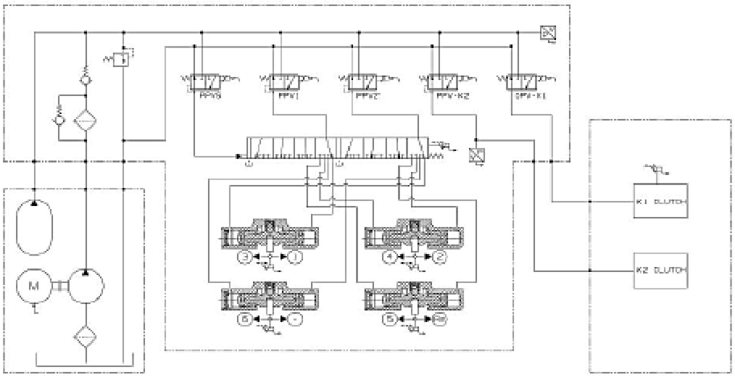


Figure 2.4: C635 DDCT complete actuation circuit.

2.2.3 Control Unit

The C635 DDCT control strategies run in a multitasking environment so that the Main Micro Controller resources are optimally managed. These strategies can be grouped as:

- **Actuator Control:** exploits the high performance achievable with electro-hydraulic actuators. The main control strategies deal with:
  – Engagement Actuators Control: the desired trajectories are realized by commanding the two relevant PPVs one against the other.
  – Shifter Control: hydraulic power to the required engagement actuator is guaranteed by a fast and precise control of the shifter, obtained by commanding the related PPV to push the shifter piston against the spring in order to reach the desired position.
  – Odd Gears Clutch Control: the normally closed clutch (K1), which is the clutch of the first and of the reverse gear, is controlled by a position closed loop.
  – Even Gears Clutch Control: the normally open clutch (K2) is controlled in force with a pressure feedback signal delivered by one of the CAM sensors.

- **Self-Tuning Control:** many self-tuning controls are needed in order to compensate for the various parameters' drift and to guarantee 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:** different modes of shift patterns in automatic and in manual mode are contemplated and are accomplished by specific control strategies and calibrations on the engine side.

2.3 Micro-slip problem

In the following figure, a schematic representation of the dry-clutch system is shown:

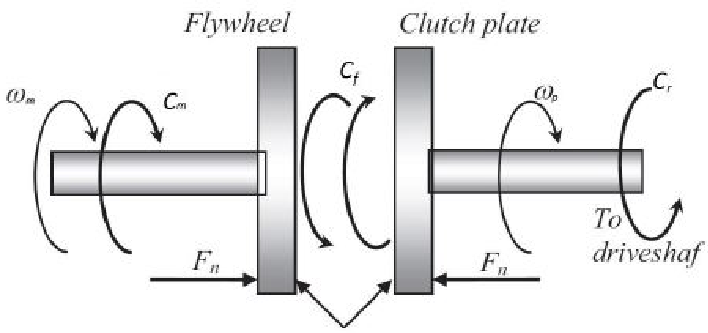


Figure 2.5: Schematic model of a dry-clutch.

It consists in practice of two rotating friction surfaces (flywheel-clutch and pressure plate-clutch) that can be pressed together.
A gearshift, independently of its direction, is composed of three phases:

- a complete disengagement of the clutch coupled with a sudden reduction of the engine torque to avoid the engine to revving up ($C_f = 0$);
- gradual closure of the clutch in which the speed difference between the engine speed and the primary shaft speed, i.e. the slipping speed $\omega_d = \omega_m - \omega_p$, is not zero; during this phase, the clutch torque C_f is controlled by the normal force F_n produced by the clutch actuator;
- closed clutch phase, in which the synchronization of the engine shaft and the primary shaft is completed and the clutch allows the full transmission of the engine torque to the driveline ($C_f = C_m$, the clutch behaves like a simple connecting rod).

The micro-slip problem is related to the second phase.
In Figure 2.5 the engine and the clutch angular speed (ω_m; ω_p), the engine torque C_m and the torque reacting by the driveline C_r are also represented.

The objective of the control design in this thesis is to provide a controlled smooth slip between the two clutch disks in order to reduce driveline oscillations, improving comfort and drivability.

2.4 State of art

In literature the dry clutch engagement problem has been widely analysed and different solutions have been proposed. In [2] is presented a control strategy where only the clutch torque is used as a control variable, while the engine torque is considered as a known non-controllable input. The resulting control scheme is illustrated in the following figure:

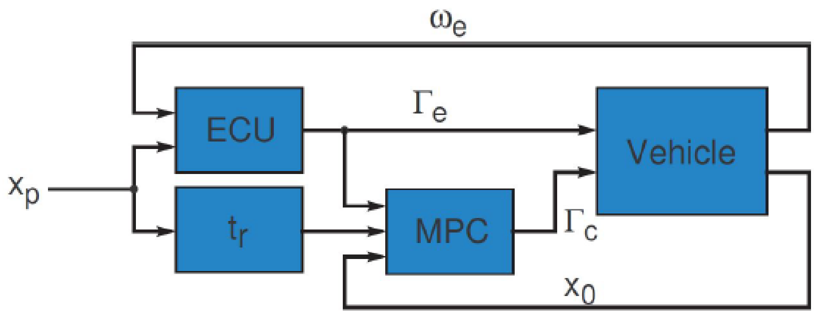


Figure 2.6: MPC control scheme presented in [2].

The engine control unit (ECU) provides as output the engine torque Γ_e based on the throttle pedal position x_p. The time control horizon t_r is given as a function of the total engagement time that is computed for a certain throttle pedal position. The clutch torque C_f is obtained by the controller solving the optimal control problem with a suitable cost function.

In [3] a decoupling controller is derived that independently tracks the optimized engine speed and the clutch slipping speed. The control scheme is represented in Figure 2.7:

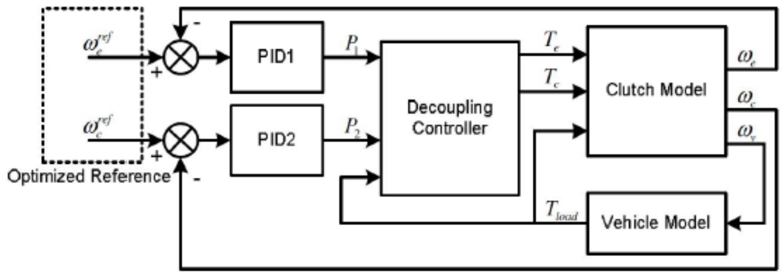


Figure 2.7: Scheme of decoupling control implemented in [3].

Engine and slipping speed can be controlled independently through the usage of the two PID controllers P1 and P2. Figure 2.7 also shows that both the vehicle and the clutch model provide a speed feedback in order to guarantee a smooth clutch engagement and a better reference tracking.

Another type of control concerning dual clutch transmission is presented in [4]. The gearshift control is implemented using a torque rate limitation strategy. The key idea here, is that the controlled clutch is only actuated in one direction in order to prevent undesirable effects such as dead zone non-linearity and saturation. For example, the oncoming clutch is only controlled such that it has the normal force to be ramped up or held. On the other hand, the off-going clutch is only actuated such that it has the normal force to be decreased or held.

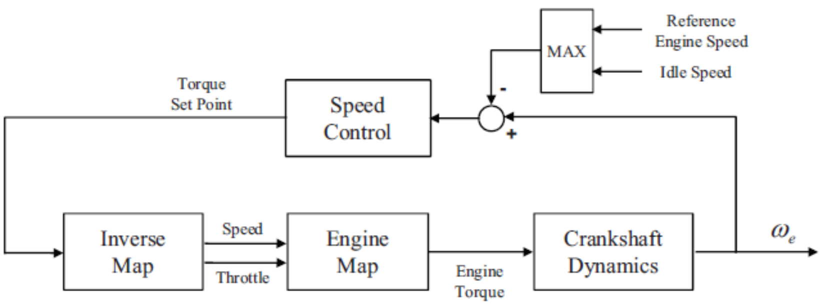


Figure 2.8: Block diagram of the engine speed/torque control analysed in [4].

The target engine speed is determined by two clutch speed measurements and the desired gear shifting duration and the engine torque controller is implemented as a simple proportional-integrative (PI) controller. Once it is defined appropriately, the engine torque is also controlled indirectly by feedback of a speed tracking error.

2.5 Model configuration details

In this section the devices that are connected to the Dry Dual Clutch are analysed and the detailed model of the entire system used by Centro Ricerche Fiat is presented. This model will be later slightly modified, and the "new" model will be the starting point for the design of the controller implemented in this thesis.
In the following figure the general scheme of the DDCT system is illustrated:

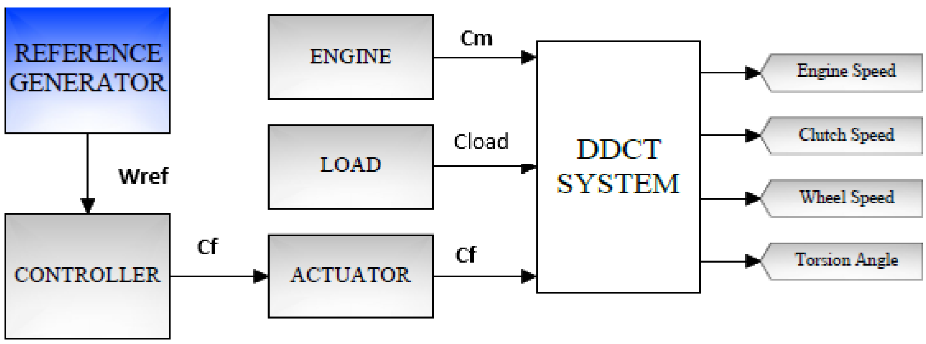


Figure 2.9: Block diagram of the DDCT system.

The model of Figure 2.9 is composed of:

- **Reference generator:** provides the reference signal ω_{Ref} for the clutch slipping speed ω_d;
- **Actuator:** provides to the system the clutch torque C_f computed by the controller;
- **Load:** includes the effects of the air, roll and slope resistances;
- **Engine:** provides the engine torque C_m;
- **DDCT system:** includes the clutch and the transmission model.

In the following all these elements are individually described in detail and for each one the corresponding Matlab/Simulink implementation is introduced.

2.5.1 Reference generator

The reference generator is that part of the system that provides the desired profile of the variable under control, i.e., the clutch slipping speed. The corresponding Simulink model is shown in Figure 2.10:

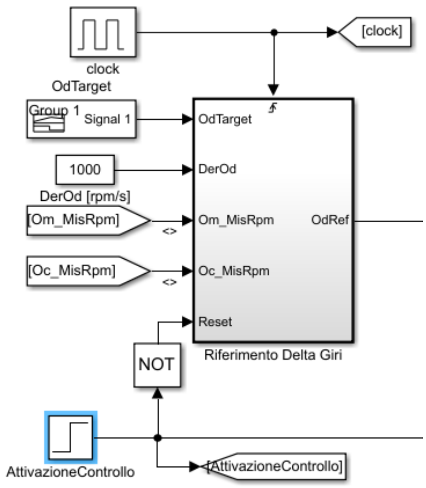


Figure 2.10: Block scheme of the reference generator.

The inputs to the block are: *Signal 1* which is equal to zero, $OmMisRpm = \omega_m$ is the engine speed and $OcMisRpm = \omega_p$ is the primary shaft speed. The *Reset* input has the task to activate the control after 0.6 seconds. To understand how this block works, it is useful to look inside it:

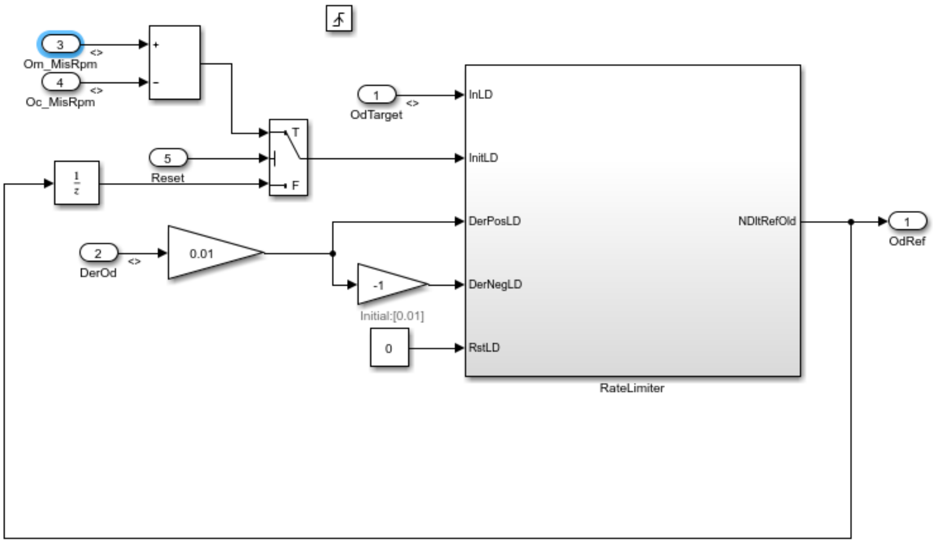


Figure 2.11: Inner part of the reference generator.

It is possible to see that the output depends on the difference $\omega_m - \omega_p$ only for 0.6 seconds: at this time the control becomes active and the output depends on the last value computed by the controller. The reference signal for the clutch slipping speed is shown in Figure 2.12:

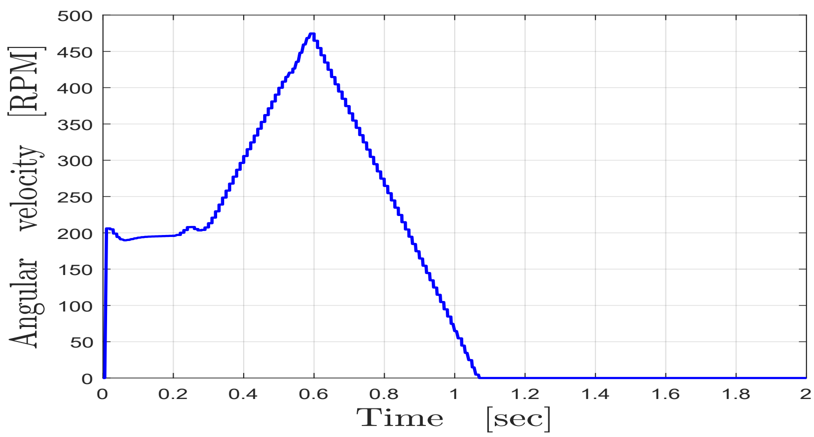


Figure 2.12: Reference signal for clutch slipping speed.

2.5.2 Actuator

The actuator is an essential part of the system since it allows torque transmission. It is composed by the blocks visible in the following figure:

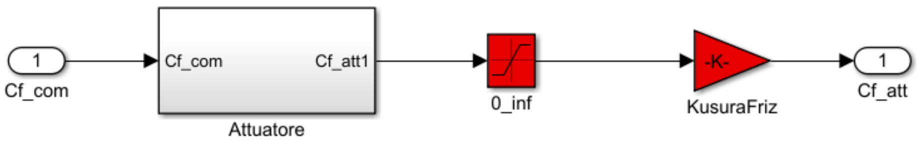


Figure 2.13: Inner part of the actuator.

The block "attuatore" contains a transfer function that describes the actuator dynamics:

$$G_a = \frac{25.305(s+138)(s^2-161.3s+3.134\cdot10^4)}{(s^2+82.23s+2563)(s^2+134.4s+4.27\cdot10^4)}$$

and a delay block that simulates a delay equal to 10 ms. There are also a saturation block, which imposes a minimum value for the clutch torque equal to zero, and a gain that identifies the transmissibility K_t, i.e., the ratio between the input and the output torque of DCT.

2.5.3 Load

The contribution of this block is essentially given by three terms:

- **aerodynamic resistance force:** it can be written as

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

  The terms in this equation are:
  – ρ_a : air density;
  – A_f : frontal area of the vehicle;
  – C_a : aerodynamic drag coefficient;
  – v_a : wind speed;
  – v_v : vehicle speed.

- **rolling resistance force:** it is modelled as

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

Here we have:
– m_v : vehicle mass;
– g : gravity acceleration;
– μ_r : rolling friction coefficient;
– β : slope angle of the road.

- **uphill driving force:** it is the resistance force arising when driving the car on a non-horizontal road. It is expressed as:

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

If the car is driven on a horizontal road, then $\alpha = 0$ and $F_g = 0$.

Considering the wheel radius r_w, the total load torque final expression is:

$$C_{Load} = (F_a + F_r + F_g) r_w$$

2.5.4 Engine

In the first approach to the control problem addressed in this thesis, the engine torque is modelled as an external input that cannot be modified; the signal related to engine torque C_m is illustrated in the following figure:

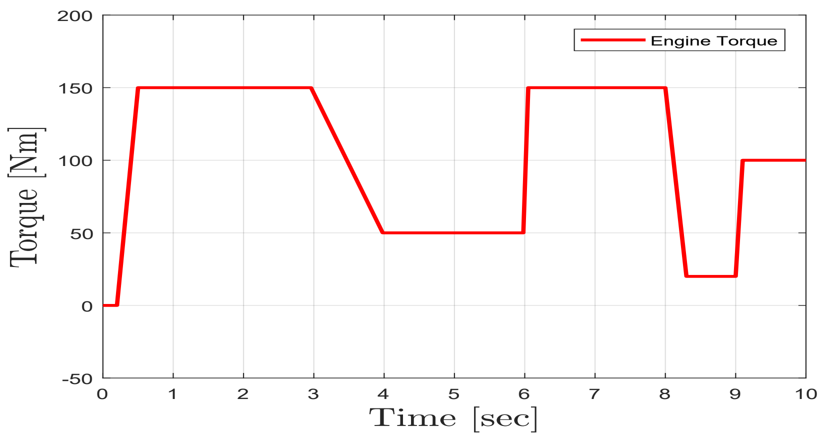


Figure 2.14: Engine torque external signal.

In a successive part of the analysis, the engine torque will not be considered as an external input, but it will be, together with the clutch torque, a manipulated variable depending on the controller computations.

2.5.5 Complete DDCT system model

The complete model including all the components previously examined is shown in figure 2.15.
Initially in the "Control" block there is a PI controller, that later on will be substituted by the Model Predictive Controller designed in this thesis.

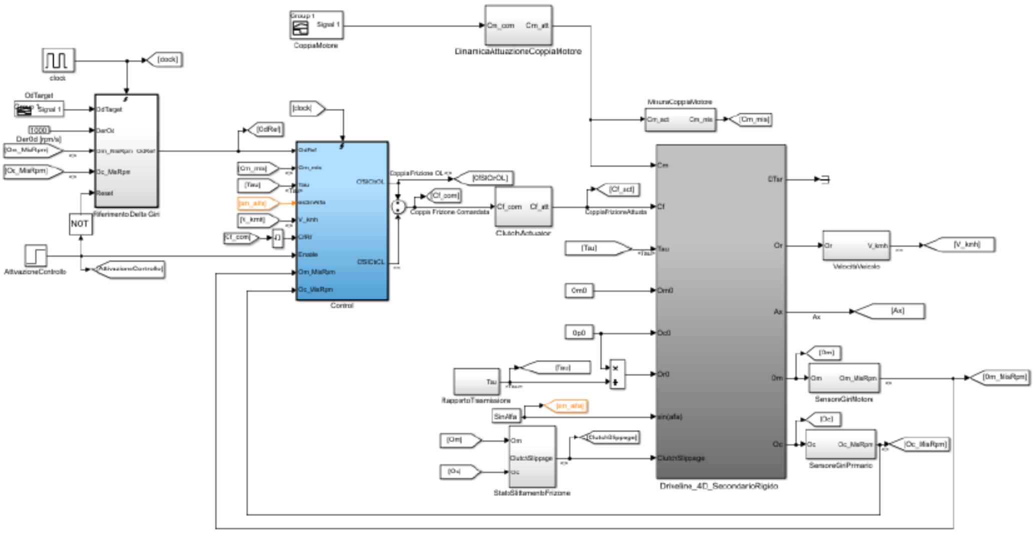


Figure 2.15: Complete model of the DDCT system.

Chapter 3

Mathematical Description of Clutch Model

3.1 Preliminary considerations

In order to design a Model Predictive Controller using the official toolbox provided by MathWorks, a state-space representation of the system is needed. In this regard a linear model is taken into account which, however, provides an accurate representation of the behaviour of the system.
In the following figure all the main components of the driveline are shown:

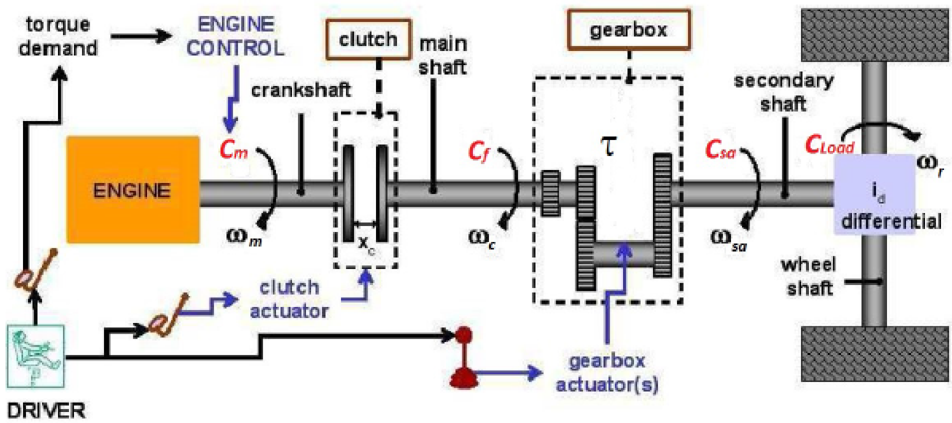


Figure 3.1: General driveline scheme.

As a consequence of considering a linear model, three assumptions have to be made:

1. The wheels motion is pure rolling;

2. The main shaft is perfectly rigid;
3. The two branches of the driveline are perfectly symmetric.

Starting from these considerations, the following linear scheme of the driveline is built:

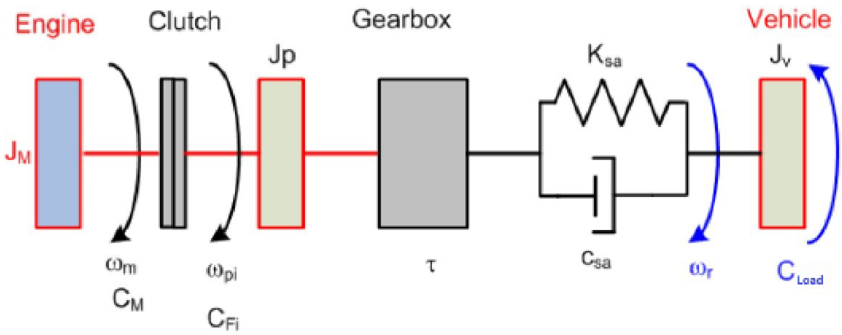


Figure 3.2: Linear driveline scheme.

3.2 Initial system equations and state-space representation

The meaning of the parameters shown in Figure 3.2 is explained in the following table:

Component	Symbol
Torsional damping coefficient	c_{sa}
Torsional stiffness coefficient	k_{sa}
Axle gear ratio	τ
Wheel radius	r_w
Wheel inertia	J_r
Vehicle mass	M
Vehicle inertia	J_v
Motor damping coefficient	c_m
Motor inertia	J_m

Table 3.1: DDCT system main parameters.

Imposing the rotational equilibrium on the system shown in Figure 3.2 the following system of differential equations is obtained:

$$
\begin{aligned}
J_m\dot{\omega}_m &= C_m - c_m\omega_m - C_f \\
J_p\dot{\omega}_p &= C_f + \frac{1}{\tau}(-c_{sa}\omega_{sr} - k_{sa}\theta_{sr}) \\
J_v\dot{\omega}_r &= c_{sa}\omega_{sr} + k_{sa}\theta_{sr} - C_{Load} \\
\dot{\theta}_{sr} &= \omega_{sr}
\end{aligned}
\tag{3.1}
$$

In particular:

$$
\begin{aligned}
\theta_{sr} &= \theta_{sa} - \theta_r \\
\omega_{sr} &= \frac{\omega_p}{\tau} - \omega_r
\end{aligned}
$$

Rearranging the various terms we obtain the final form of the equations:

$$
\begin{aligned}
\dot{\omega}_m &= \frac{1}{J_m}(C_m - C_f) - \frac{c_m}{J_m}\omega_m \\
\dot{\omega}_p &= \frac{C_f}{J_p} - (\frac{c_{sa}}{J_p\tau^2} + \frac{c_{prim}}{J_p})\omega_p + \frac{c_{sa}}{J_p\tau}\omega_r - \frac{k_{sa}}{J_p\tau}\theta_{sr} \\
\dot{\omega}_r &= \frac{c_{sa}}{J_p\tau}\omega_p - \frac{c_{sa}}{J_v}\omega_r + \frac{k_{sa}}{J_v}\theta_{sr} - \frac{C_{Load}}{J_v} \\
\dot{\theta}_{sr} &= \frac{\omega_p}{\tau} - \omega_r
\end{aligned}
\tag{3.2}
$$

The objective now is to express these equations using the state-space representation. In order to do this, some assumptions have to be made:

- The clutch torque C_f is the manipulated variable, depending on the controller computations;
- The load torque C_{Load} and the engine torque C_m are assumed to be known external disturbances;
- As a consequence of the two previous statements, the input vector is $u(t) = [C_f(t), C_m(t), C_{Load}(t)]^T$;
- The state vector is $x(t) = [\omega_m(t), \omega_p(t), \omega_r(t), \theta_{sr}(t)]^T$
- The output vector is $y(t) = [\omega_d(t), \omega_{sr}(t)]^T$, where $\omega_d(t) = \omega_m(t) - \omega_p(t)$ is the clutch slipping speed and $\omega_{sr}(t) = \frac{\omega_p}{\tau} - \omega_r$ is the drive shaft torsion speed.

Taking into account these considerations and remembering that the general form of a state-space representation is

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

the four matrices describing our system are:

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

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

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

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

The four matrices A, B, C and D represent the starting point for the design of the controller developed in this thesis. In a successive part of the analysis of the system, some changes will be done in the state equations and consequently the matrices will change. This variations are due to the introduction of new elements in the Simulink model of the system and to investigate the possible improvements caused by different control strategies.

3.3 Complete model

In this section the modifications made with respect to the "basic" model are illustrated, together with the final form of the state-space representation and of the Simulink model used for controller tuning and simulations.

3.3.1 Integral action

A standard approach to reduce (ideally eliminate) steady state errors is to use the integral action. Considering our primary output, that is the clutch slipping speed ω_d, the tracking error of interest is

$$\tilde{y} = \omega_{Ref} - \omega_d = \dot{z}$$

A new output is considered:

$$z = \int_0^t \tilde{y}(\tau)\, d\tau$$

The aim of control is to make z converge to 0 as $t \to \infty$.
Due to errors/approximations, z may not converge to 0 but to a finite value:

$$\lim_{t \to \infty} z(t) = z_{ss} = const < \infty$$

Since $\tilde{y} = \dot{z}$, it follows that

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

The variable z is considered as an additional state of the system.

3.3.2 Engine actuator

In order to make the Simulink model closer to reality, the engine actuator is taken into account, whose transfer function is:

$$AC_m = \frac{14.05}{s + 14.15}$$

This actuator is then compensated: a lead network is introduced. The general form of a lead network is:

$$R_D = \frac{1 + \frac{s}{\omega_d}}{1 + \frac{s}{m_d \omega_d}}, \quad m_d > 1$$

In order to not have a significant impact on the system dynamics, the following lead network has been chosen:

$$R_D = \frac{1}{AC_m} \cdot \frac{100}{s + 100}$$

After some mathematical manipulations, the final form of the lead network results:

$$R_D = \frac{100.7}{100} \cdot \frac{1 + \frac{s}{14.05}}{1 + \frac{s}{100}}$$

Since a compensation is performed, the only thing to be taken into account for the changes in the state equations is that the engine torque C_m has to be multiplied by the transfer function $\frac{100}{s+100}$. As a consequence an additional state variable has to be introduced.

3.3.3 Disturbances on manipulated variables

In the first version of the system model, the only manipulated variable is the clutch torque; a second implementation considers the engine torque to be another manipulated variable. In addition, disturbances acting both on engine and clutch torque are considered: in particular, the disturbance on the engine torque can be represented by the driver action (disturbance equal to zero means that the driver is not pressing the accelerator pedal). This situation in Simulink is illustrated in the following figure:

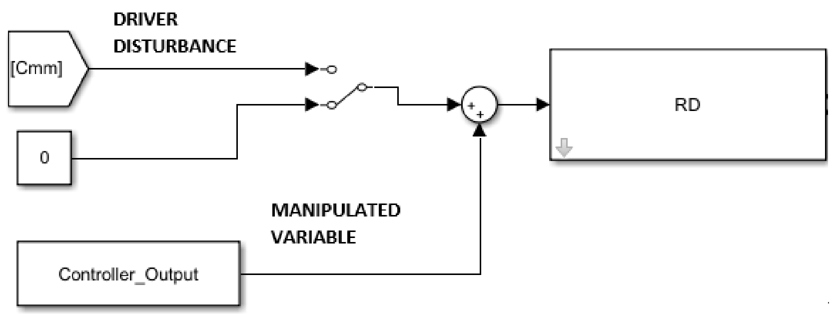


Figure 3.3: Engine torque scheme block.

The output of this scheme block is $C'_m = (C_m + D_m) \cdot \frac{100}{s+100}$, where D_m represents the disturbance due to the driver.
A similar scheme can be used for the clutch torque.

3.3.4 Final representation

The introduced modifications lead to the following changes:

- state vector

$$x = \begin{bmatrix} \omega_m \\ \omega_p \\ \omega_r \\ \theta_{sr} \\ C'_m \\ z \end{bmatrix}$$

- input vector

$$u = \begin{bmatrix} C_m \\ C_f \\ C_{Load} \\ D_m \\ D_f \\ \omega_{Ref} \end{bmatrix}$$

where D_f represents the disturbance acting on the clutch torque.

- output vector

$$y = \begin{bmatrix} \omega_m - \omega_p \\ \frac{\omega_p}{\tau} - \omega_r \\ z \end{bmatrix}$$

The final complete system equations are:

$$\begin{aligned}
\dot{\omega}_m &= \frac{1}{J_m}[\frac{(C_m + D_m) \cdot 100}{s + 100} - C_f - D_f] - \frac{c_m}{J_m}\omega_m \\
\dot{\omega}_p &= \frac{C_f + D_f}{J_p} - (\frac{c_{sa}}{J_p \tau^2} + \frac{c_{prim}}{J_p})\omega_p + \frac{c_{sa}}{J_p \tau}\omega_r - \frac{k_{sa}}{J_p \tau}\theta_{sr} \\
\dot{\omega}_r &= \frac{c_{sa}}{J_v \tau}\omega_p - \frac{c_{sa}}{J_v}\omega_r + \frac{k_{sa}}{J_v}\theta_{sr} - \frac{C_{Load}}{J_v} \\
\dot{\theta}_{sr} &= \frac{\omega_p}{\tau} - \omega_r \\
\dot{C'}_m &= -100 \cdot C'_m + 100 \cdot C_m + 100 \cdot D_m \\
\dot{z} &= \omega_{Ref} - (\omega_m - \omega_p)
\end{aligned} \tag{3.3}$$

As a consequence the following matrices are obtained:

$$A = \begin{bmatrix} -\frac{c_m}{J_m} & 0 & 0 & 0 & \frac{1}{J_m} & 0 \\ 0 & -(\frac{c_{sa}}{J_p \tau^2} + \frac{c_{prim}}{J_p}) & \frac{c_{sa}}{J_p \tau} & -\frac{k_{sa}}{J_p \tau} & & \\ 0 & \frac{c_{sa}}{J_p \tau} & -\frac{c_{sa}}{J_v} & \frac{k_{sa}}{J_v} & 0 & 0 \\ 0 & \frac{1}{\tau} & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -100 & 0 \\ -1 & 1 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$B = \begin{bmatrix} 0 & -\frac{1}{J_m} & 0 & 0 & -\frac{1}{J_m} & 0 \\ 0 & \frac{1}{J_p} & 0 & 0 & \frac{1}{J_p} & 0 \\ 0 & 0 & -\frac{1}{J_v} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 100 & 0 & 0 & 100 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

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

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

In Figure 3.4 the complete Simulink model of the system is shown:

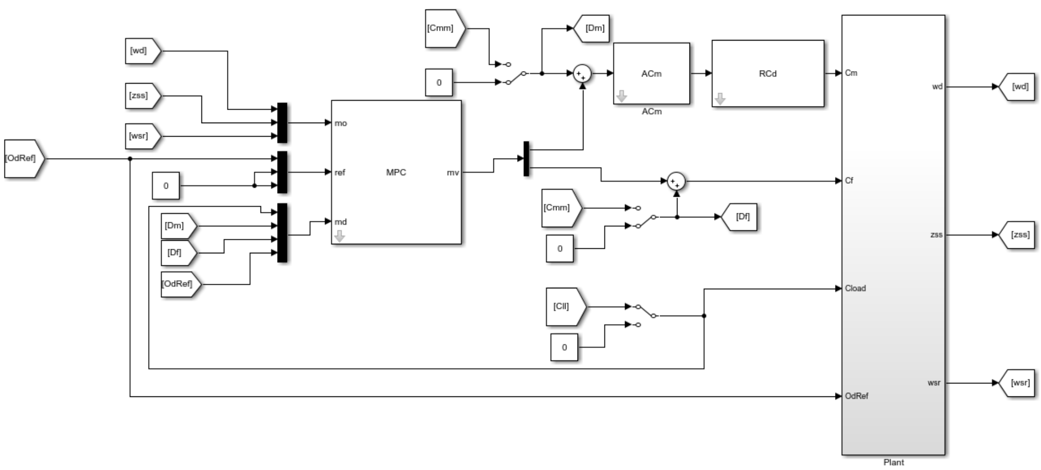


Figure 3.4: Complete Simulink scheme.

In Figure 3.5 the internal structure of the plant can be seen:

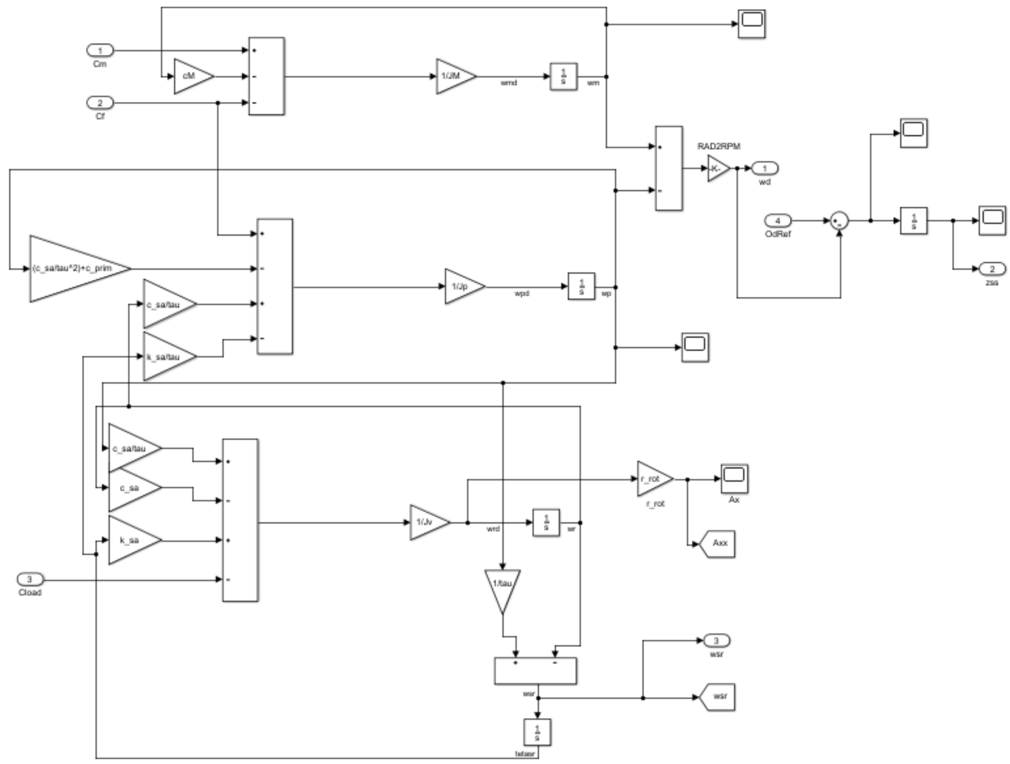


Figure 3.5: Plant internal structure.

The clutch actuator is used in the Simulink model, but it is not considered in the state-space equations because it does not have an important effect on the considered signals and by considering it, the equations would be much more complicated since its transfer function is a fourth order one.

Chapter 4

Model Predictive Control Technique

4.1 Introduction

Model Predictive Control (MPC) originated in the late seventies and since then it has been widely developed and improved, both from a theoretical and a practical point of view. This technique is based on two key operations: prediction and optimization. In fact, the term Model Predictive Control indicates a group of control methods which make explicit use of a dynamic model of the system to be controlled in order to predict the system future outputs and to produce a control signal that minimizes an objective function. Therefore, models have an important role in MPC.

The growing application of this technique to automotive systems, aerospace systems, chemical processes, robotics and biomedical devices, resides in the numerous advantages it gives with respect to more traditional techniques:

- It can be used to control a great variety of processes, from those with slow dynamics and long delays (chemical processes) to the faster ones (like mechatronic and automotive applications)
- It is able to handle state/input/output constraints
- Its formulation can be easily extended to MIMO systems
- It is able to compensate for measurable disturbances

On the other hand, there are obviously some drawbacks. One of them is the amount of computational effort required to derive the control law: this problem has been partially solved thanks to the remarkable improvements of the last decade in micro-processors and computers computational capability. Anyway, the greatest drawback is represented by the need for an appropriate model of the process to be available. The design algorithm is based on the prior knowledge of the model, and the efficiency of the control action will be affected by the discrepancies existing between the real process and the used model.

4.2 MPC strategy

In Figure 4.1 it is possible to see the main components of an MPC:

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

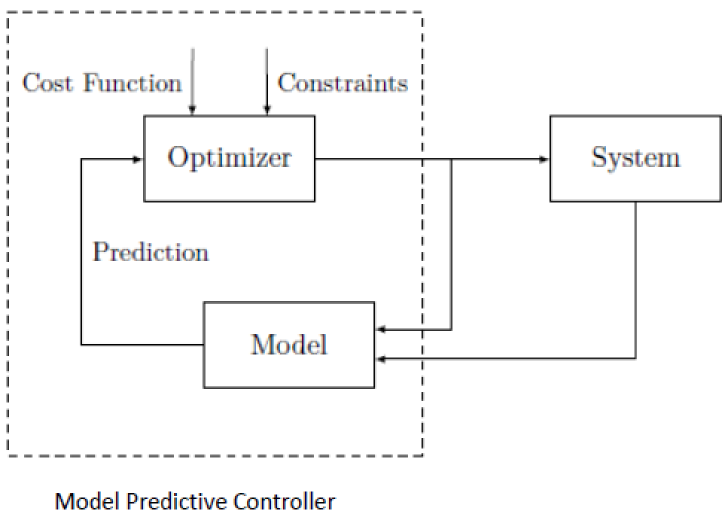


Figure 4.1: Block scheme of an MPC.

The approach is the following: at each time step

- a prediction over a given time horizon is performed, using a model of the plant
- the command input is chosen as the one yielding the "best" prediction (i.e., the prediction closest to the desired behaviour) by means of some optimization algorithm.

4.2.1 Prediction Model

The model is an element of paramount importance in model predictive control. An ideal model should be complete enough to fully capture the process dynamics and allow the predictions to be calculated, and at the same time to be intuitive and permit theoretic analysis. The system model is needed in order to calculate the predicted outputs at future time instants.

The predictive control problem is often dealt with using state space models: this is principally due to the fact that the main theoretical results of MPC related to stability come from a state space formulation, which can be used for both monovariable and multivariable processes and can easily be extended to nonlinear processes.
Discrete time models are convenient if the system of interest is sampled at discrete times: if the sampling rate is appropriately chosen, the behaviour between the samples can be safely ignored and the model describes exclusively the behaviour at the sample times.
The general representation of the prediction model is a nonlinear, time-invariant state-space system:

$$\begin{aligned} x(k+1) &= f(x(k), u(k)) \qquad f \in \mathbb{C}^{\mathbb{1}} \\ y(k) &= g(x(k), u(k)) \qquad g \in \mathbb{C}^{\mathbb{1}} \end{aligned} \tag{4.1}$$

where $x(k) \in \mathbb{R}^n$ is the state variable, $y(k) \in \mathbb{R}^p$ is the system output, $u(k) \in \mathbb{R}^m$ is the control input. In any case, as mentioned in chapter 3, in this thesis a Linear Time Invariant (LTI) discrete model is used, which is represented in the following way:

$$\begin{aligned} x(k+1) &= Ax(k) + Bu(k) \\ y(k) &= Cx(k) + Du(k) \\ x(0) &= x_0 \end{aligned} \tag{4.2}$$

Here $A \in \mathbb{R}^{n \times n}$ is the state matrix, $B \in \mathbb{R}^{n \times m}$ is the input matrix, $C \in \mathbb{R}^{p \times n}$ is the output matrix and $D \in \mathbb{R}^{m \times p}$ is the feedforward matrix. x_0 represents

the initial state of the system that is usually given. The parameter k is a non-negative integer denoting the sample number: it is related to the time t by the equation $t = kT_s$, where T_s is the sample time.
The solution to the system 4.2 is:

$$x(k) = A^k x_0 + \sum_{j=0}^{n-1} A^{k-j-1} B u(j) \tag{4.3}$$

4.2.2 Constraints

Usually the manipulated inputs of a system are bounded due to physical and/or technological constraints. These limitations are taken into account by model predictive technique adding some linear inequalities:

$$Eu(k) \leq e, \qquad k \geq 0 \tag{4.4}$$

In particular

$$E = \begin{bmatrix} I \\ -I \end{bmatrix}$$

and

$$e = \begin{bmatrix} \overline{u} \\ \underline{u} \end{bmatrix}$$

where $\overline{u}$ and $\underline{u}$ represent respectively the maximum and the minimum control input value.

Sometimes for reasons of safety, product quality etc., constraints on states and outputs are also imposed:

$$Fx(k) \leq f, \qquad k \geq 0 \tag{4.5}$$

Another type of possible constraint is the one on the rate of change of the input, $u(k) - u(k-1)$. In order to maintain the state-space form of the model, we may augment the state as:

$$\tilde{x}(k) = \begin{bmatrix} x(k) \\ u(k-1) \end{bmatrix}$$

The augmented system model becomes:

$$\begin{aligned} \dot{\tilde{x}} &= \tilde{A}\tilde{x} + \tilde{B}u \\ y &= \tilde{C}\tilde{x} \end{aligned} \tag{4.6}$$

where

$$\tilde{A} = \begin{bmatrix} A & 0 \\ 0 & 0 \end{bmatrix}$$

$$\tilde{B} = \begin{bmatrix} B \\ I \end{bmatrix}$$

$$\tilde{C} = \begin{bmatrix} C & 0 \end{bmatrix}$$

Constraints on rate change are taken into account in the following way:

$$\Delta_{min} \leq u(k) - u(k-1) \leq \Delta_{max}$$

$$F\tilde{x}(k) + Eu(k) \leq e$$

$$F = \begin{bmatrix} 0 & -I \\ 0 & I \end{bmatrix}$$

$$E = \begin{bmatrix} I \\ -I \end{bmatrix}$$

$$e = \begin{bmatrix} \Delta_{max} \\ \Delta_{min} \end{bmatrix}$$

Δ_{max} and Δ_{min} represent respectively the maximum and the minimum control input variation rate.

The conceptual distinction between input constraints, and output or state constraints is that the input constraints often represent *physical limits*. In these cases, if the controller does not respect the input constraints, the physical system enforces them. The output or state constraints are instead parameters imposed by the controller designer. They may not be achievable depending on the disturbances affecting the system. It is often the function of an MPC controller to determine in real time that the output or state constraints are not achievable, and relax them in some satisfactory manner.

4.3 Regulation problem

The aim of this problem is to design a controller to take the state of a system to the origin. If the setpoint is not the origin, or the objective is to track a time-varying setpoint trajectory, proper modifications of the zero setpoint problem will be made to account for that.

To make the analysis simpler, in this paragraph it is assumed that the state is measured. The state estimation problem will be later analysed for the situations where the state cannot be measured. Using the system model described in (3.2), we can predict how the state evolves given any set of inputs we are considering.

The choice of the cost function is really important, since the achievement of the controller objectives is strongly influenced by it. Taking into account H_p steps in the future, where H_p is known as *prediction horizon*, a general formulation of the cost function to be minimized can be the following:

$$V(x(0), \mathbf{u}) = \frac{1}{2} \sum_{k=0}^{H_p-1} L(x(k), u(k)) + \frac{1}{2} L_n(x(H_p)) \tag{4.7}$$

Considering the general nonlinear model of the system (4.1), the mathematical formulation of the optimization problem is:

$$\begin{gathered} U^* = arg \ \min_u V(x(0), \mathbf{u}) \\ subject \quad to \\ x(k+1) = f(x(k), u(k)) \\ x(k+i|k) \in \mathbb{X} \\ u(k+i|k) \in \mathbb{U} \\ x(k+H_p|k) \in \mathbb{X}_f \end{gathered} \tag{4.8}$$

where:

- $\mathbb{X}$ and $\mathbb{U}$ are the polyhedral regions representing respectively state and input constraints (details in [8])
- $\mathbb{X}_f$ is the terminal constraint set introduced in the optimization problem in order to ensure asymptotic stability (see [7])
- $U^*(k) = [u^*(k|k), u^*(k+1|k), ..., u^*(k+H_p-1|k)]^T$ is the optimal sequence of manipulated variables (controlled inputs)

Looking at the expression of $U^*(k)$, it is possible to recognize that the higher is H_p, the higher will be the computational time/effort required to calculate the optimal control sequence. In order to reduce such effort, the cost function (4.7) can be minimized with respect to the reduced sequence

$U(k) = [u(k|k), u(k+1|k), ..., u(k+H_c-1|k)]^T$, where H_c is referred to as *control horizon* and the inequality $H_c < H_p$ holds. For the remaining $H_p - H_c$ control inputs, needed to calculate the state predictions until the time step H_p, three solutions are possible:

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

- $u(k+i|k) = u(k+H_c|k), \quad H_c \leq i \leq H_p - 1$

- $u(k+i|k) = -Fx(k+i|k), \quad H_c \leq i \leq H_p - 1$, where $F \in R^n$ is the gain needed to stabilize state feedback

Generally, for the regulation problem, the weighting functions $L(\cdot)$ and $L_n(\cdot)$ are expressed as quadratic forms: $L(x,u) = \frac{1}{2}(x'Qx + u'Ru)$ and $L_n(x) = \frac{1}{2}(x'P_f x)$. The functions $L(\cdot)$ and $L_n(\cdot)$ are assumed to be continuous in their arguments and are considered as design parameters to be suitably chosen in order to reach the desired control performances.

The tuning parameters of the controller are the matrices Q and R. Having a different weighting matrix for the final state penalty reflects the most general case.

Large values of Q with respect to R reflect the designer's intent to drive the state to the origin quickly accepting to have a large control action. Penalizing the control action through large values of R relative to Q is the way to reduce the control action and slow down the rate at which the state approaches the origin.

Usually the matrices Q, P_f and R are chosen to be diagonal, real and symmetric. In particular, Q and P_f are *positive semidefinite* ($Q \succeq 0$ and $P_f \succeq 0$), while R is *positive definite* ($R \succ 0$). These assumptions guarantee that the solution to the optimal control problem exists and is unique.

4.4 Tracking problem

The tracking problem is the case where the controller goal is to to move the measured outputs of a system to a specified and constant setpoint. In order to treat this problem, the desired output is indicated with y_{sp}, while a steady state of the system model is indicated with (x_s, u_s). From the model (4.2), the steady state condition satisfies the following:

$$\begin{bmatrix} I - A & -B \end{bmatrix} \begin{bmatrix} x_s \\ u_s \end{bmatrix} = 0 \tag{4.9}$$

For unconstrained systems, we also impose the requirement that the steady state satisfies $Cx_s = y_{sp}$ for the tracking problem, giving the set of equations:

$$\begin{bmatrix} I - A & -B \\ C & 0 \end{bmatrix} \begin{bmatrix} x_s \\ u_s \end{bmatrix} = \begin{bmatrix} 0 \\ y_{sp} \end{bmatrix} \tag{4.10}$$

If it is possible to find a solution to this system, the deviation variables are defined

$$\tilde{x}_k = x(k) - x_s$$
$$\tilde{u}_k = u(k) - u_s$$

that satisfy the following model

$$\tilde{x}(k+1) = A\tilde{x} + B\tilde{u}(k) \tag{4.11}$$

The solution to the regulation problem gives the input to be applied to the system that is $u(k) = \tilde{u}(k) + u_s$.

In order to have a solution for the system (4.10), it is sufficient that all the rows of the matrix are linearly independent; this means that are required at least as many inputs as outputs with setpoints. However in many applications it is possible to have more measured outputs than manipulated inputs: to handle this situation a new matrix H is introduced, such that $r = Hy$ where r represents a selection of linear combinations of the measured outputs. The variable $r \in \mathbb{R}^{n_c}$ is known as *controlled variable*, while r_{sp} indicates the setpoints for the controlled variables.

4.4.1 Steady-state target problem

The function to be minimized can be expressed as follows:

$$\min_{x_s, u_s} \frac{1}{2}(|u_s - u_{sp}|^2_{R_s} + |Cx_s - y_{sp}|^2_{Q_s}) \tag{4.12}$$

subject to:

$$\begin{bmatrix} I-A & -B \\ HC & 0 \end{bmatrix} \begin{bmatrix} x_s \\ u_s \end{bmatrix} = \begin{bmatrix} 0 \\ r_{sp} \end{bmatrix}$$
$$Eu_s \leq e$$
$$FCx_s \leq f$$

In order to guarantee that the solution exists and is unique, two assumptions are made:

- The target problem is feasible for the controlled variable setpoints of interest r_{sp}
- The steady-state input penalty R_s is positive definite.

4.4.2 Dynamic regulation problem

The following objective function is defined:

$$V(\tilde{x}(0), \tilde{\mathbf{u}}) = \frac{1}{2} \sum_{k=0}^{N-1} |\tilde{x}(k)|_Q^2 + |\tilde{u}(k)|_R^2 \tag{4.13}$$

subject to:

$$\dot{\tilde{x}} = A\tilde{x} + B\tilde{u}$$

Here $\tilde{x}(0) = \hat{x}(k) - x_s$: this means that the initial condition for the regulation problem is given by the state estimate minus the steady state x_s. The minimization problem is stated in this way:

$$\min_{\tilde{\mathbf{u}}} V(\tilde{x}(0), \tilde{\mathbf{u}}) \tag{4.14}$$

subject to

$$E\tilde{u} \leq e - Eu_s$$
$$FC\tilde{x} \leq f - FCx_s$$

As it is possible to see in the previous inequalities, also the constraints are shifted by the steady state (x_s, u_s). With $V^0(\tilde{x}(0))$ and $\tilde{\mathbf{u}}^0(\tilde{x}(0))$ are indicated respectively the optimal cost and the solution. Only the first move of the optimal sequence, which is $\tilde{u}^0(\tilde{x}(0)) = \tilde{\mathbf{u}}^0(0; \tilde{x}(0))$, is applied to the system at each time step, so the controller output is $u(k) = \tilde{u}^0(\tilde{x}(0)) + u_s$.

4.5 Receding horizon control

The optimal control move $U^*(k) = [u^*(k|k), u^*(k+1|k), ..., u^*(k+H_p-1|k)]^T$ applied over the interval $[t, t+H_p]$ leads to an open loop control strategy. Performances can therefore deteriorate, since open loop techniques are highly affected by parameter uncertainties and disturbances. To overcome such a drawback, a feedback control action can be obtained through the Receding Horizon (RH) principle.

Receding horizon control (RHC) is sometimes called receding-horizon predictive control (RHPC) and is better known as model predictive control (MPC) for state-space models. Its origins reside in conventional optimal control that is based on the minimization of some performance criterion, where both finite or infinite horizon can be considered.

RHC basic concept is the following: at the current time, the optimal control is obtained on a finite fixed horizon from the current time k (for example the interval $[k, k+H_p]$ is considered). Among all the control inputs computed on the entire horizon $[k, k+H_p]$, only the first is applied to the system. The procedure is then repeated at the next time, in this case $[k+1, k+1+H_p]$. This situation is represented in the following figure:

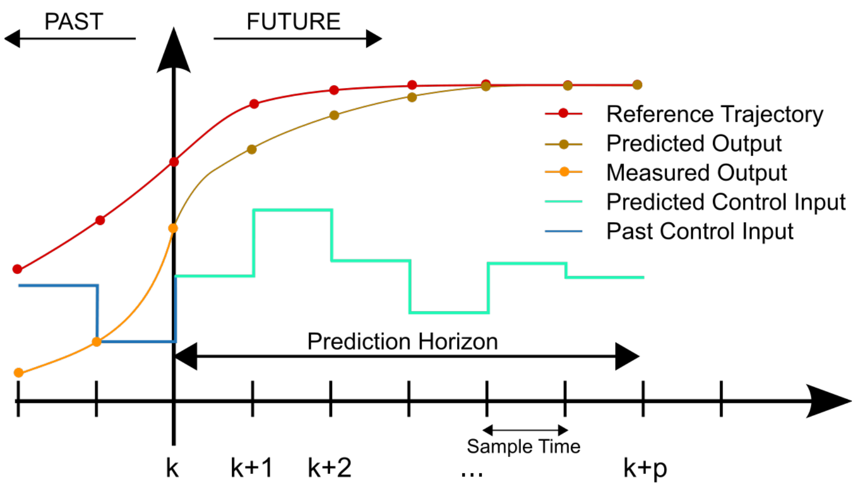


Figure 4.2: Receding Horizon principle.

Recalling that the cost function depends on the initial state, if the model and the cost function are time invariant, then the same input $u(k)$ will be applied to the system whenever the state takes the same value.

The term "receding horizon" indicates the horizon recedes as time proceeds. RHC is usually represented by a state feedback control if states are available. However, full states may not be available, since measurement of all states may be expensive or impossible. From measured inputs and outputs, we can construct or estimate all states: this is often called a filter for stochastic systems or a state observer for deterministic systems.

4.6 State estimation

In the theoretical analysis so far, the state is always been supposed to be measured. In some cases this could not be possible and a state observer (SO) is needed in order to provide an estimate $\hat{x}(k)$ of the system states $x(k)$. The estimation is generally based on the input/output measurements of the system to be controlled. The general structure of an observer is reported in Figure 4.3:

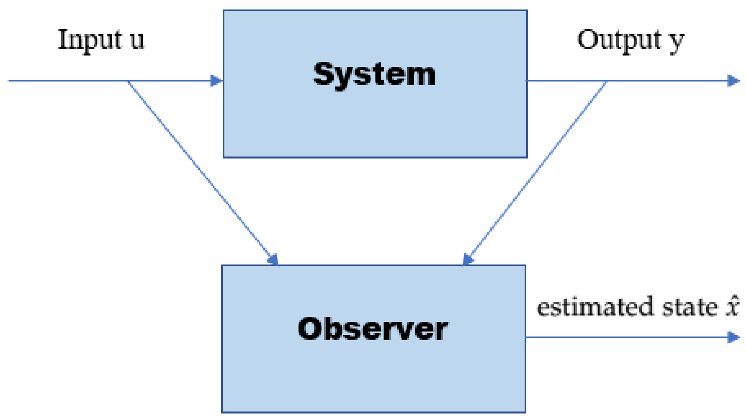


Figure 4.3: General observer scheme.

The observer estimation error $e_e(k|k-1)$ is defined as:

$$e_e(k|k-1) = x(k) - \hat{x}(k|k-1) \tag{4.15}$$

A state observer is defined *asymptotic* if

$$\lim_{k\to\infty} ||e_e(k|k-1)|| = \lim_{k\to\infty} ||x(k) - \hat{x}(k|k-1)|| = 0$$

Taking into account the state equation of the system and of the estimate

$$x(k+1) = Ax(k) + Bu(k)$$
$$\hat{x}(k+1) = A\hat{x}(k) + Bu(k)$$

and through some manipulations of the terms

$$\begin{aligned} e_e(k+1|k) &= x(k+1) - \hat{x}(k+1|k) \\ &= Ax(k) + Bu(k) - A\hat{x}(k|k-1) - Bu(k) \\ &= Ae_e(k|k-1) \end{aligned} \tag{4.16}$$

it is possible to make the following considerations:

- if the matrix A is not asymptotically stable ⇒

$$\lim_{k\to\infty} ||e_e(k+1|k)|| \neq 0$$

- if the matrix A is asymptotically stable ⇒

$$\lim_{k\to\infty} ||e_e(k+1|k)|| = 0$$

This means that if the observer is implemented using state equations, the asymptotic estimation is not always guaranteed, and even if the matrix A is asymptotically stable, the state estimation error may have a slow and oscillating convergence. To overcome this drawback, a term depending on the output estimation error $y(k) - y(k|k-1)$ is introduced:

$$\begin{aligned} \hat{x}(k+1|k) &= A\hat{x}(k|k-1) + Bu(k) + L(y(k) - \hat{y}(k|k-1)) = \\ &= A\hat{x}(k|k-1) + Bu(k) + L(y(k) - C\hat{x}(k|k-1)) \end{aligned} \tag{4.17}$$

Repeating the same kind of manipulations done for (4.15), the following equation related to the estimation error is obtained:

$$e_e(k+1|k) = [A - LC]e(k|k-1) \tag{4.18}$$

The matrix L is called *observer gain* and if the couple (A, C) is observable, L can be chosen in order to make the difference $A - LC$ asymptotically stable.

A more complicated situation is the one where disturbances are considered. Denoting with $w(k)$ the process disturbance and with $v(k)$ the measurement error, the state equations become:

$$\begin{aligned} x(k+1) &= Ax(k) + Bu(k) + Gw(k) \\ y(k) &= Cx(k) + Dv(k) \end{aligned} \tag{4.19}$$

The estimation error is now

$$e_e(k+1) = [A - LC]e_e(k|k-1) + Gw(k) - LDv(k) \tag{4.20}$$

This last equation highlights that, even if the matrix L is properly chosen, in the presence of disturbances the estimation error can not be made asymptotically null. In order to minimize the estimation error suitable assumptions have to be made on $v(k)$ and $w(k)$: the Kalman filter is introduced.

4.6.1 The Kalman filter

In statistics and control theory, Kalman filtering, also known as linear quadratic estimation (LQE), is an algorithm that uses a series of measurements observed over time, containing statistical noise and other inaccuracies, and produces estimates of unknown variables that tend to be more accurate than those based on a single measurement alone, by estimating a joint probability distribution over the variables for each time frame.

In order to design a Kalman filter, the following hypotheses are introduced:

- the initial state x_0 is a Gaussian variable such that:

$$\begin{aligned} E[x_0] &= 0 \\ E[x_0 x_0^T] &= X_0 \\ X_0 &\geq 0 \end{aligned} \tag{4.21}$$

  where $E[\cdot]$ indicates the *expected value*.

- the process disturbance $w(k)$ and the measurement error $v(k)$ are white noises with zero mean value and known variance which are uncorrelated with each other:

$$\begin{aligned} E[w(k)] = 0, &\quad E[v(k)] = 0 \\ E[w(k)w(k)^T] = w\tilde{Q}, &\quad E[v(k)v(k)^T] = v\tilde{R} \\ E[v(k)w(k)^T] = E[x_0 v(k)^T] &= E[x_0 w(k)^T] = 0 \end{aligned} \tag{4.22}$$

where $\tilde{Q} \succ 0$ and $\tilde{R} \succ 0$, i.e. there is a not null noise component for each output.

Considering equation (4.18) and since $E[w(k)] = 0$ and $E[v(k)] = 0$:

$$E[e_e(0|-1)] = E[X_0 - \tilde{X}_0] = 0 \tag{4.23}$$

This means that

$$E[e_e(k+1|k)] = 0, \quad \forall k \tag{4.24}$$

The covariance matrix of the error is defined as:

$$\tilde{P}(k|k-1) = E[e_e(k|k-1)e_e(k|k-1)^T] \tag{4.25}$$

It results

$$\tilde{P}(0|-1) = \tilde{P}_0$$

The gain observer L is determined by minimizing a quadratic expression of the error covariance:

$$\min_L (\psi^T \tilde{P}(k+1|k)\psi) \tag{4.26}$$

where ψ is a generic vector.
The expression of the gain L deriving from the solution to the previous minimization problem is the following:

$$L = A\tilde{P}(k|k-1)C^T + [C\tilde{P}(k|k-1)C^T + \tilde{R}]^{-1} \tag{4.27}$$

The expression $\tilde{P}(k|k-1)$ comes from the resolution of the *Difference Riccati Equation* (DRE):

$$\tilde{P}(k+1|k) = A\tilde{P}(k|k-1)A^T + \tilde{Q} - A\tilde{P}C^T[C\tilde{P}(k|k-1)C^T + \tilde{R}]^{-1}C\tilde{P}A^T \tag{4.28}$$

$\tilde{Q}$ and $\tilde{R}$ are weight matrices that can be conveniently made.

The matrices $\tilde{Q}$ and $\tilde{R}$ are related to the information regarding the noise. The $\tilde{R}$ diagonal elements are given by the variance of the output errors. The definition of $\tilde{Q}$ is more complicated because the process disturbances are often unknown. Therefore, more frequently the matrices $\tilde{Q}$ and $\tilde{R}$ are considered as two project parameters.

4.7 Model Predictive Control Toolbox in Matlab

Matlab provides a specific toolbox (Model Predictive Control Toolbox) for the design of model predictive controllers and provides Simulink blocks for simulations. The toolbox allows to specify plant and disturbance models, prediction and control horizons, constraints and weights. It is possible to adjust the behaviour of the controller by varying its weights and constraints at run time. In this thesis the controller is designed at the command line, and then loaded in the Simulink block in Figure 4.4, which has to be properly connected to the rest of the signals considered in the complete model.

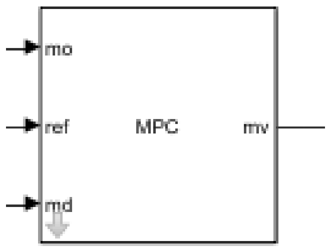


Figure 4.4: MPC Simulink block.

In Figure 4.4 different input and output ports can be recognized:

- **mo** is the port where *measured outputs* have to be connected
- the port **ref** is connected to the reference signals for the measured outputs
- **md** is the port where *measured disturbances* are connected
- from the port **mv** *manipulated variables* are delivered to the system

Chapter 5

MPC Tuning and Simulations Results

This chapter covers the MPC tuning procedure exploited to reach the best possible performance in the clutch slipping control.
Different configurations have been used, and they are listed below:

- **Single manipulated variable:** in this configuration the engine torque C_m is treated as a known external input that cannot be modified, while the controller has the task to compute the clutch torque C_f;

- **Double manipulated variable:** both the engine and the clutch torque are manipulated variables depending on the controller computation;

- **Double manipulated variable with integrator:** with respect to the previous model, the integral of the tracking error is controlled as a second output of the system;

- **Triple output control:** in addition to the clutch slipping speed ω_d and to the integral of the tracking error, the drive shaft torsion speed ω_{sr} is also considered as a controlled output.

For each of these configurations the Simulink model and the simulation results are shown. The sampling time for the MPC controller has been chosen as $T_s = 0.01\,s$.

5.1 Single manipulated variable

The used Simulink model is the following:

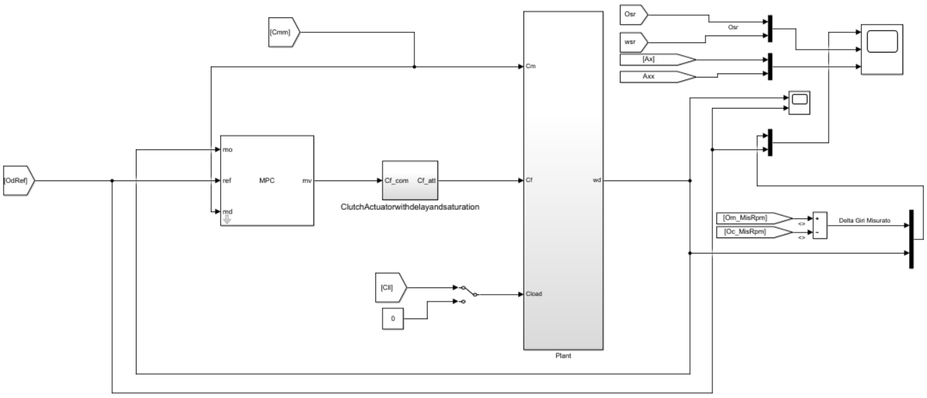


Figure 5.1: Single Manipulated Variable Simulink Scheme.

In this case disturbances on manipulated variables are not considered and the engine torque is treated a known fixed external signal.

Different simulations are performed varying from time to time the values of the tuning parameters. The used values are summarized in the following table:

Plot	H_p	H_c	Output weight	MV weight	MV Rate weight
SMV 1	5	1	1	1	1
SMV 2	7	1	1	1	1
SMV 3	9	1	1	1	1

Table 5.1: Single Manipulated Variable Tuning Parameters.

Where 'SMV' stands for 'Single Manipulated Variable'. The time results of the simulations related to each of the quantities of interest are shown in the following figures:

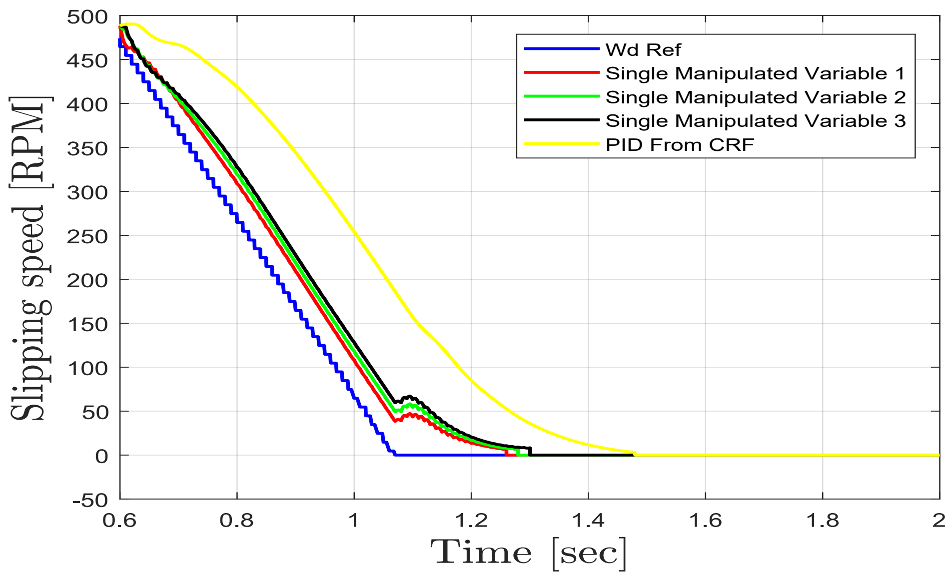


Figure 5.2: Slipping speed simulation results.

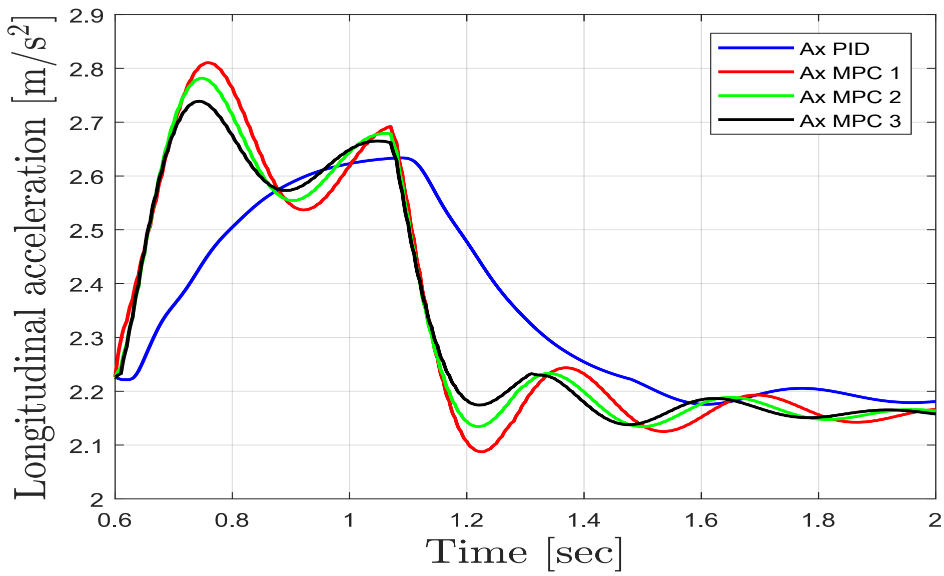


Figure 5.3: Longitudinal acceleration simulation results.

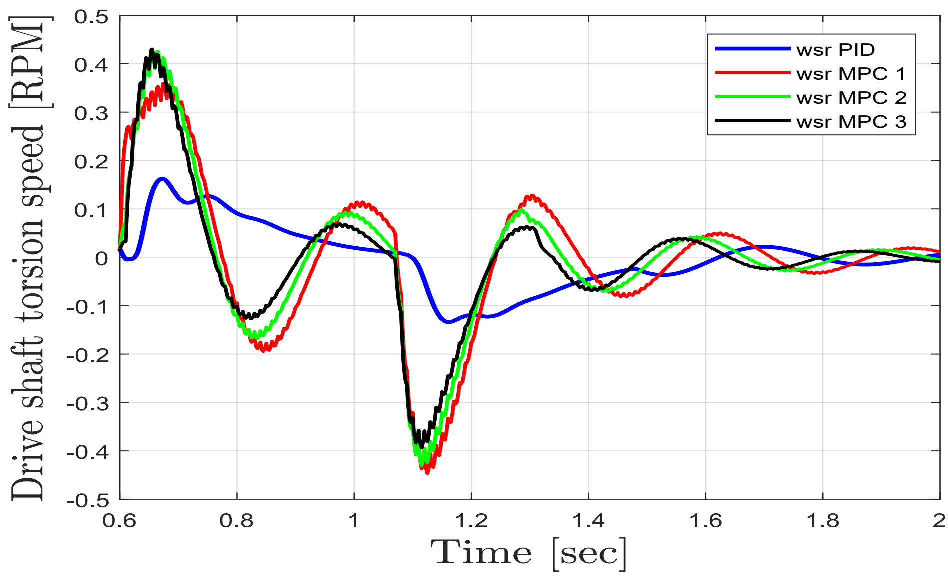


Figure 5.4: Drive shaft torsion speed simulation results.

The command activity of the clutch torque is also shown:

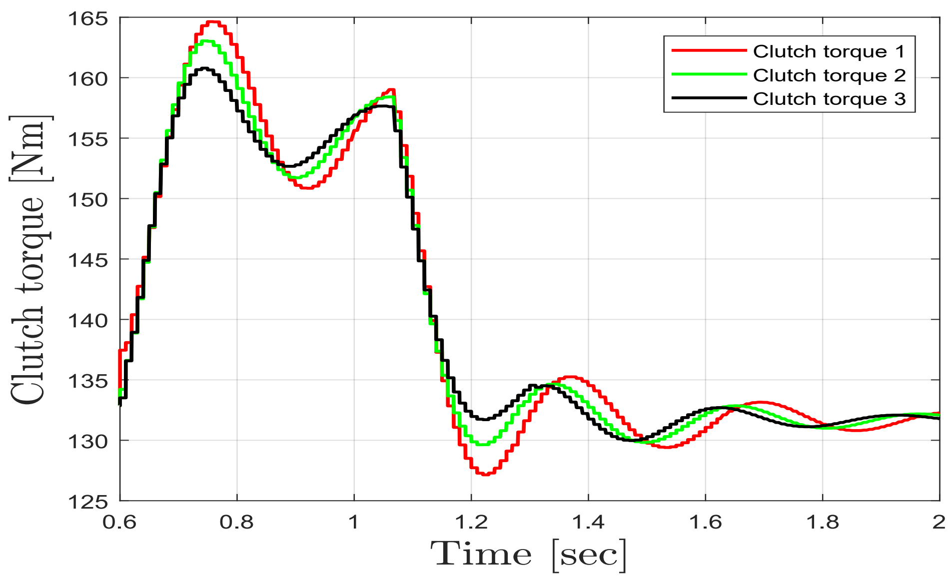


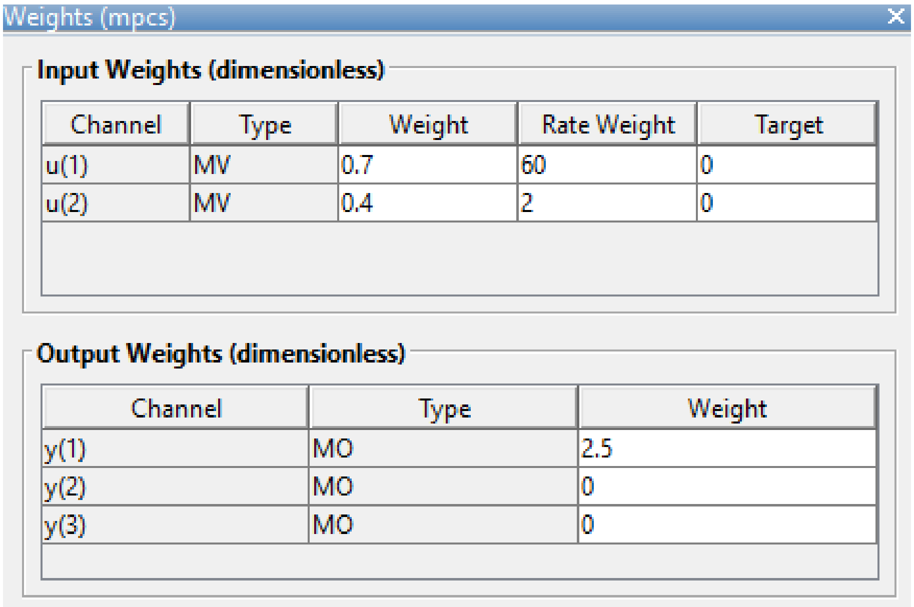
Figure 5.5: Clutch torque.

For these simulations the following constraint has been set on the manipulated variable: $C_f \geq 0$. Constraints on the output ω_d are not necessary.

From the obtained results it can be said that MPC is able to make the slipping speed converging to zero faster than what the PID controller allows. The aim of the next configurations used for simulations will be to obtain better performances in terms of convergence speed and reduction of the drive shaft oscillations, as well as being able to manage the command activity in a more flexible manner.

5.2 Double manipulated variable

For this configuration and for the following ones the Simulink model in Figure 3.4 is considered. Only the slipping speed is considered as a controlled output, so the weights on the integral error and on the drive shaft torsion speed are set equal to zero:


Weights (mpcs)

Input Weights (dimensionless)

Channel	Type	Weight	Rate Weight	Target
u(1)	MV	0.7	60	0
u(2)	MV	0.4	2	0

Output Weights (dimensionless)

Channel	Type	Weight
y(1)	MO	2.5
y(2)	MO	0
y(3)	MO	0

Figure 5.6: Weights in the Model Predictive Control Toolbox.

Constraints on engine torque and clutch torque are imposed:

$$C_m \geq -40\,Nm \text{ and } C_f \geq -40\,Nm$$

The tuning parameters for the various simulations without disturbances on the engine torque are summarized here:

Plot	H_p	H_c
DMV 1	12	1
DMV 2	13	2
DMV 3	15	3

Table 5.2: Double Manipulated Variable model parameters.

Simulation results show the signals behaviour only from 0.6 seconds: this is the instant when the control action starts.

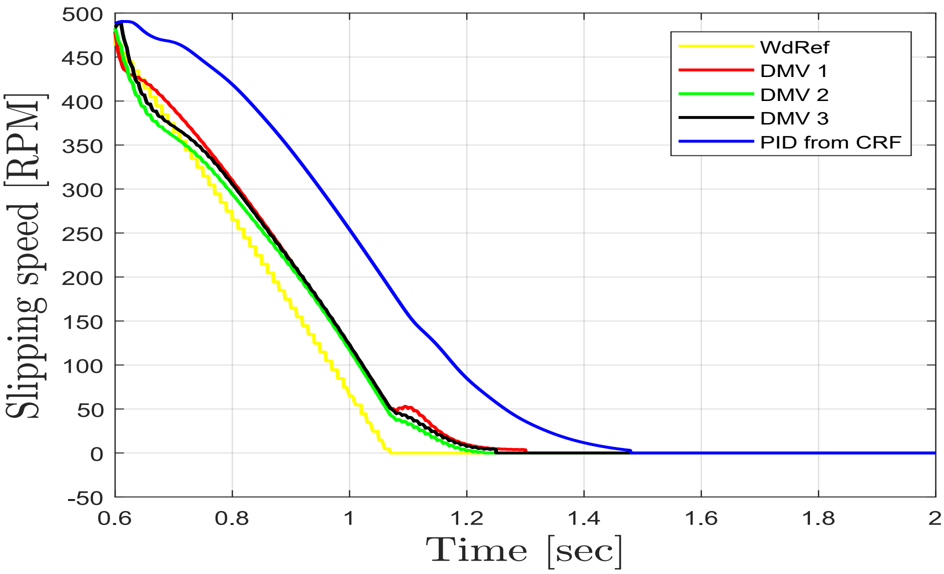


Figure 5.7: Slipping speed Double Manipulated Variable.

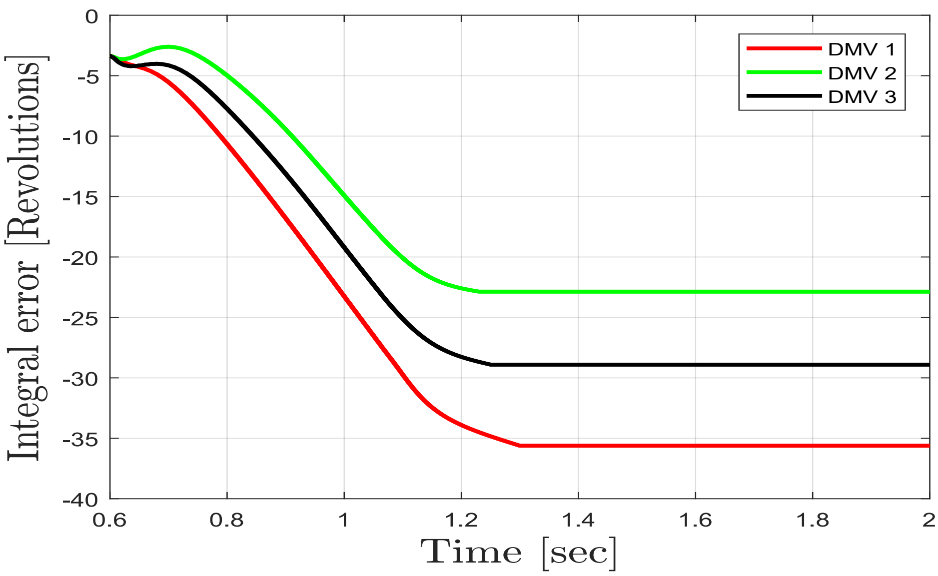


Figure 5.8: Integral error Double Manipulated Variable.

From the previous figure it is possible to see that the integral error of the slipping speed does not reach a null value, but it is going to stabilize around finite one: as explained in section (3.3.1), this is sufficient to bring the error to zero as time tends to infinity.

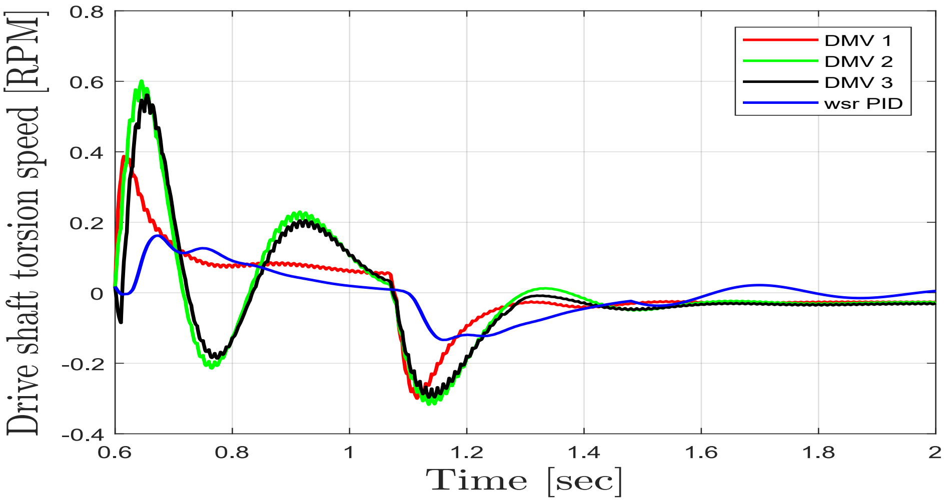


Figure 5.9: Drive shaft torsion speed Double Manipulated Variable.

The time behaviour of the drive shaft torsion speed seems to give better results with the first set of tuning parameters.

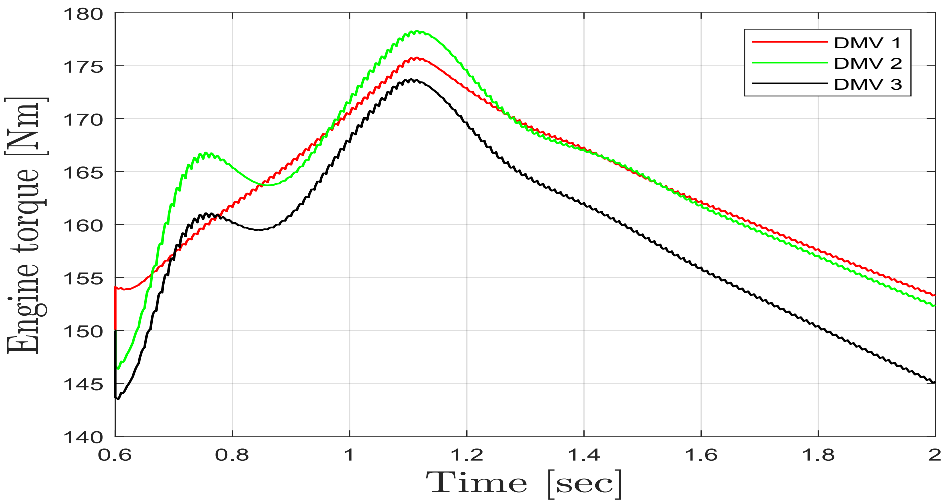


Figure 5.10: Engine torque Double Manipulated Variable.

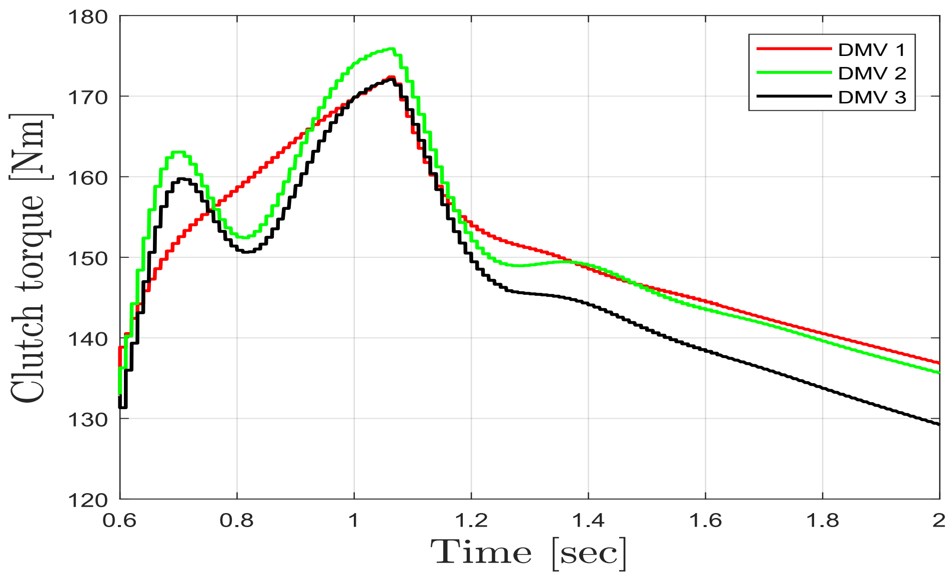


Figure 5.11: Clutch torque Double Manipulated Variable.

Figure 5.10 and 5.11 show that with the third set of parameters a smaller control action is needed.

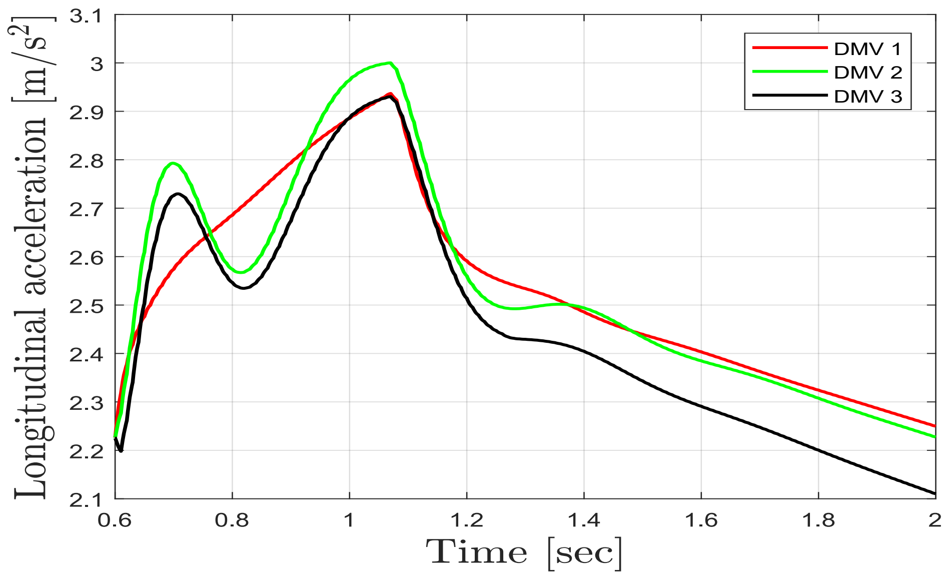
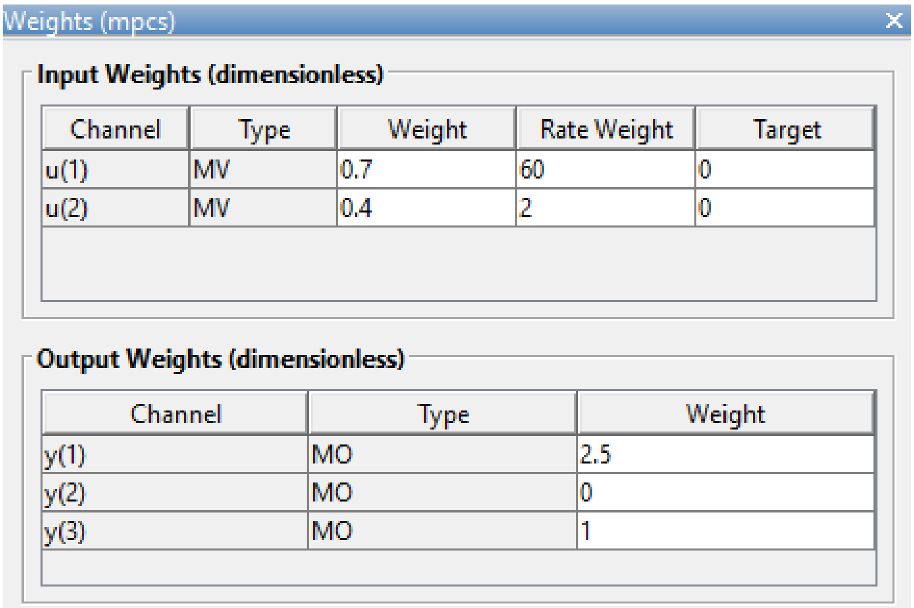


Figure 5.12: Longitudinal acceleration Double Manipulated Variable.

5.3 Double manipulated variable with integrator

In this configuration the integral of the slipping speed error is also taken in consideration as a second controlled output. The reference signal for this output is a constant signal equal to zero. The primary output is always the slipping speed, so the new weight for the primary output is set to 2.5, while the weight for the integral of the slipping speed error is set equal to 1.


Weights (mpcs)

Input Weights (dimensionless)

Channel	Type	Weight	Rate Weight	Target
u(1)	MV	0.7	60	0
u(2)	MV	0.4	2	0

Output Weights (dimensionless)

Channel	Type	Weight
y(1)	MO	2.5
y(2)	MO	0
y(3)	MO	1

Figure 5.13: Double Manipulated Variable with integrator tuning parameters.

Constraints on engine torque and clutch torque are the same as the previous model:

$$C_m \geq -40\,Nm \text{ and } C_f \geq -40\,Nm$$

The parameters used for simulations are shown:

Plot	H_p	H_c
DVI 1	12	1
DVI 2	13	2
DVI 3	15	3

Table 5.3: Double Manipulated Variable with integrator model parameters.

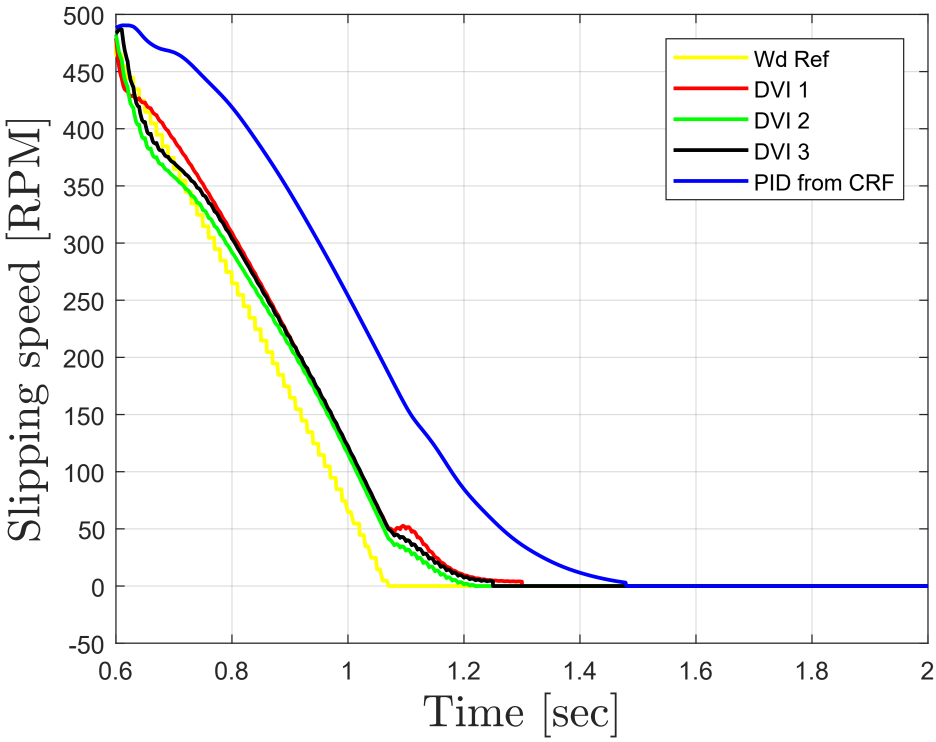


Figure 5.14: Slipping speed Double Manipulated Variable with Integrator.

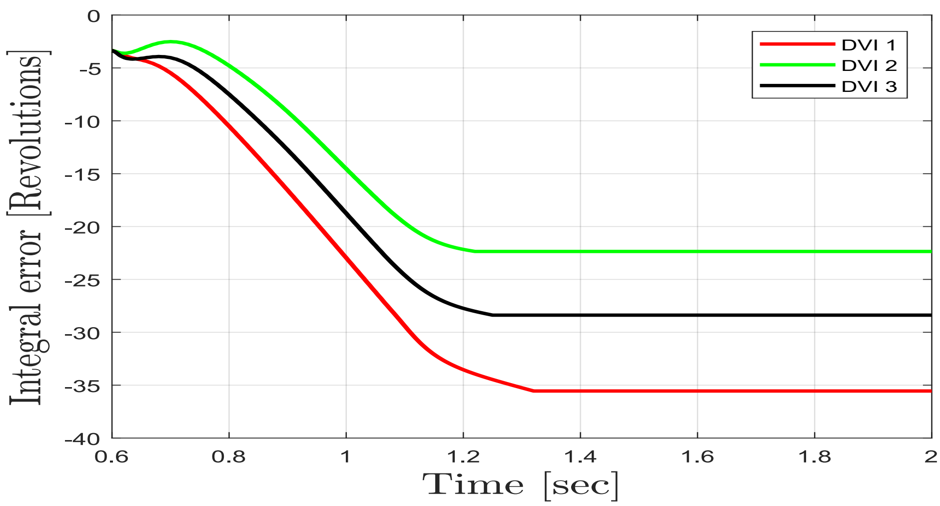


Figure 5.15: Integral error Double Manipulated Variable with Integrator.

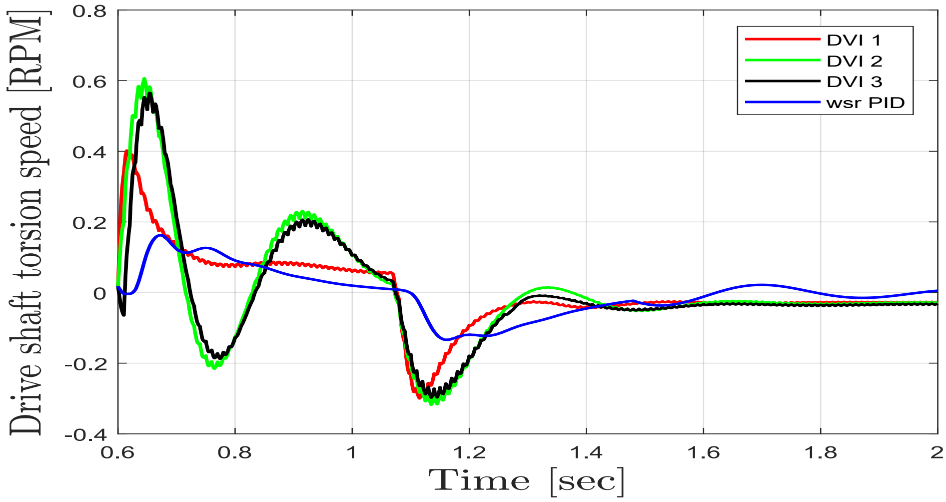


Figure 5.16: Drive shaft torsion speed Double Manipulated Variable with Integrator.

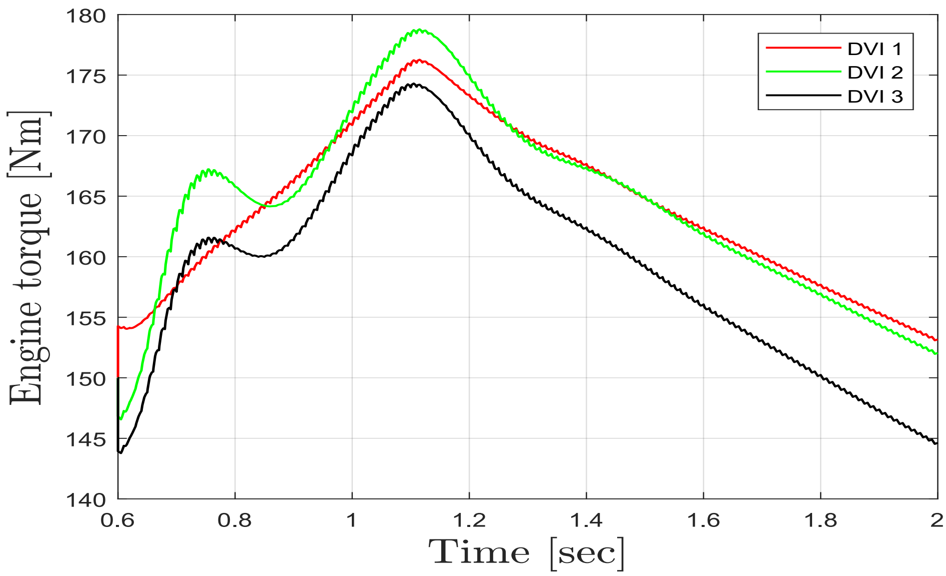


Figure 5.17: Engine torque Double Manipulated Variable with Integrator.

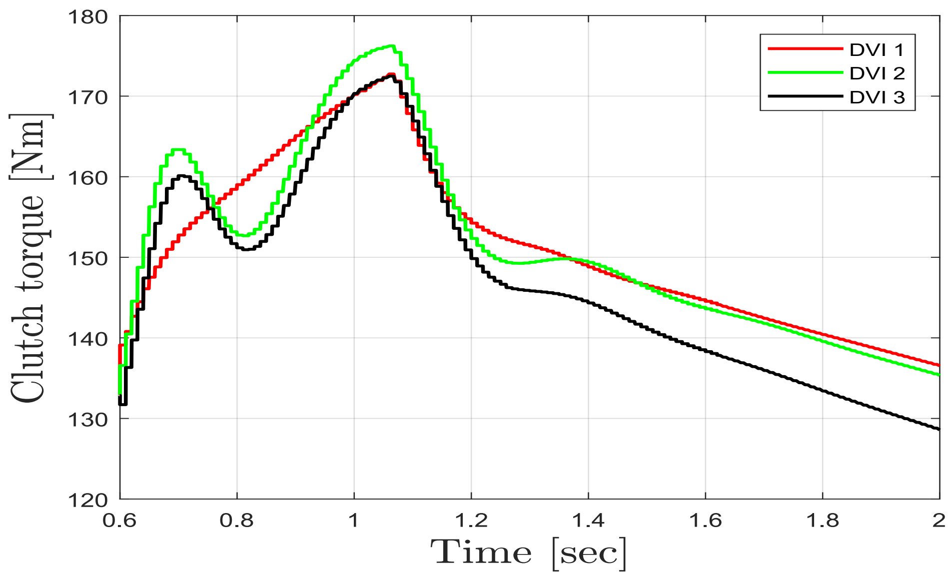


Figure 5.18: Clutch torque Double Manipulated Variable with Integrator.

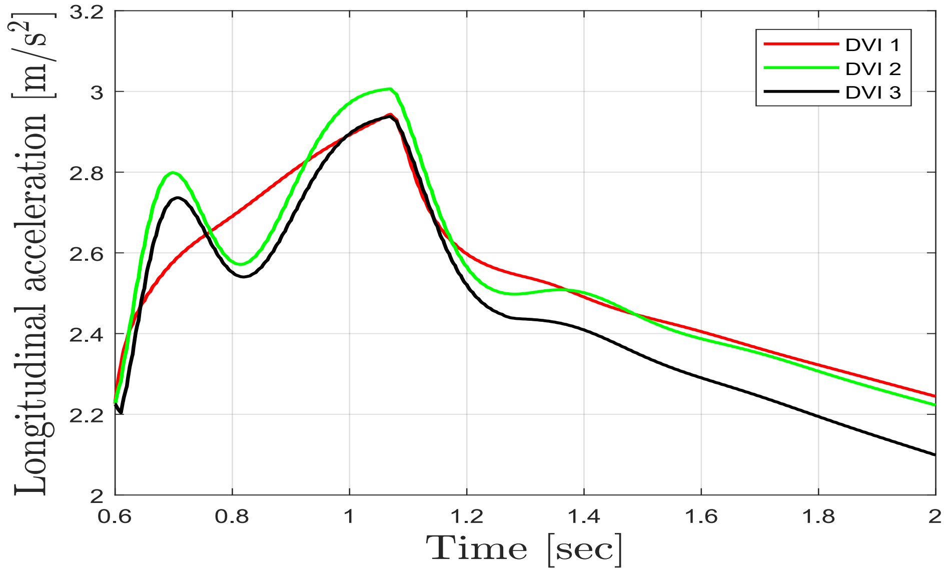


Figure 5.19: Longitudinal acceleration Double Manipulated Variable with Integrator.

5.4 Triple output control

In this configuration, the drive shaft torsion speed ω_{sr} is considered as a third output to control: the same weight is assigned to z_{ss} and ω_{sr}:

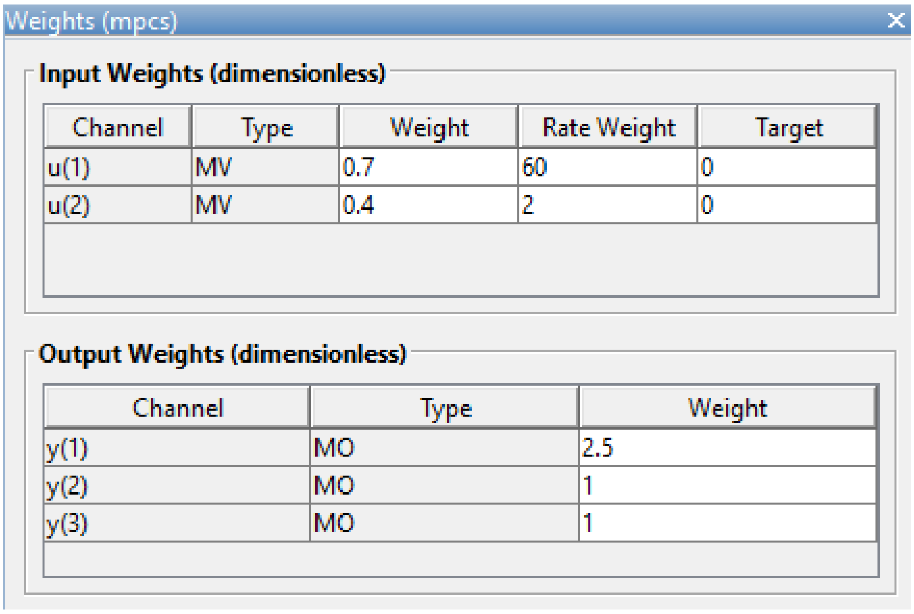


Figure 5.20: Triple output control configuration weights.

The parameters used for the simulations are listed here:

Plot	H_p	H_c
DVT 1	12	1
DVT 2	13	2
DVT 3	15	3

Table 5.4: Double Manipulated Variable with triple output model parameters.

In the following figures simulation results are shown:

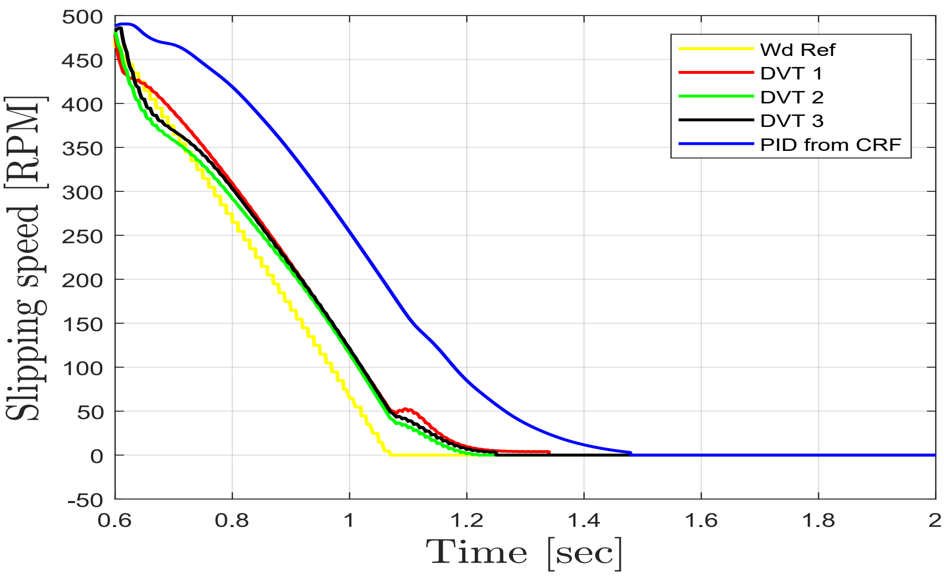


Figure 5.21: Slipping speed Triple output control.

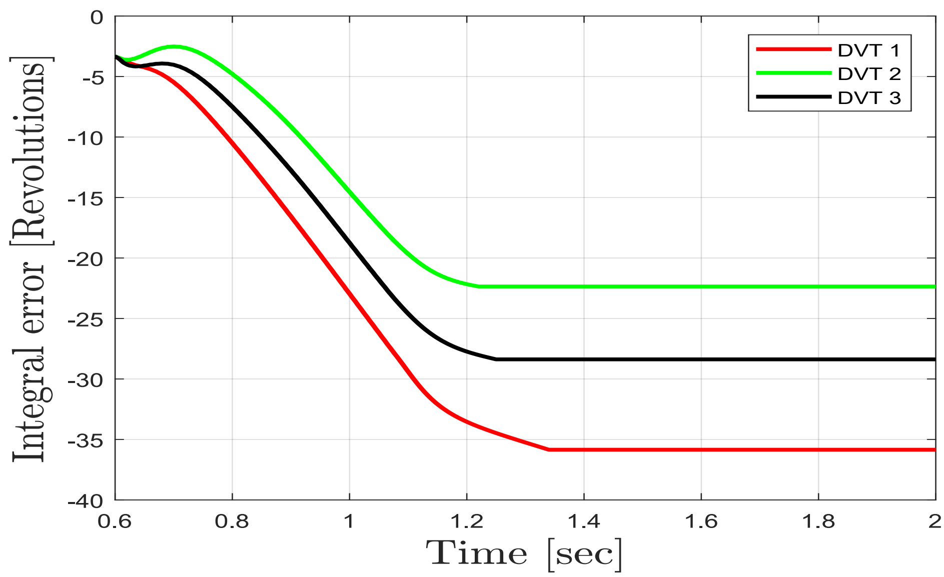


Figure 5.22: Integral error Triple output control.

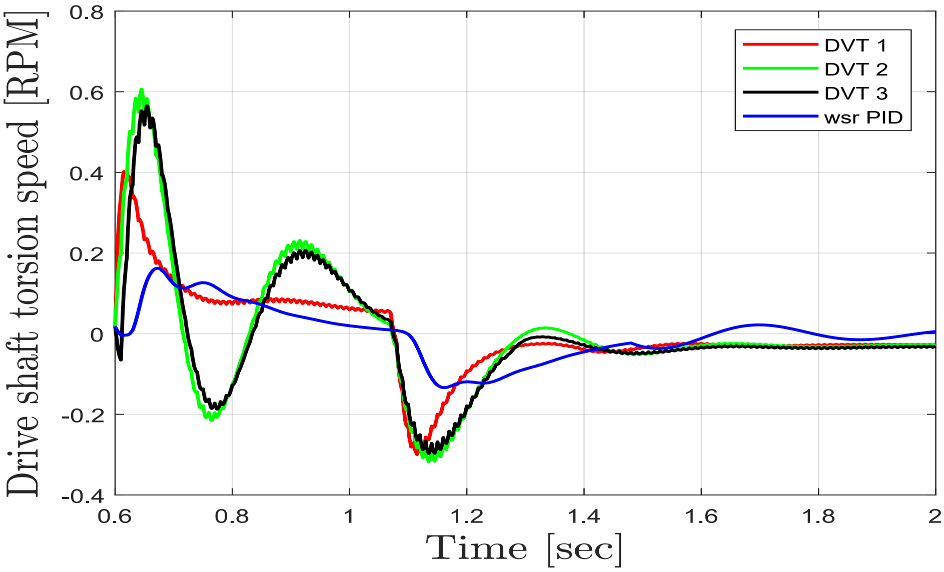


Figure 5.23: Drive shaft torsion speed Triple output control.

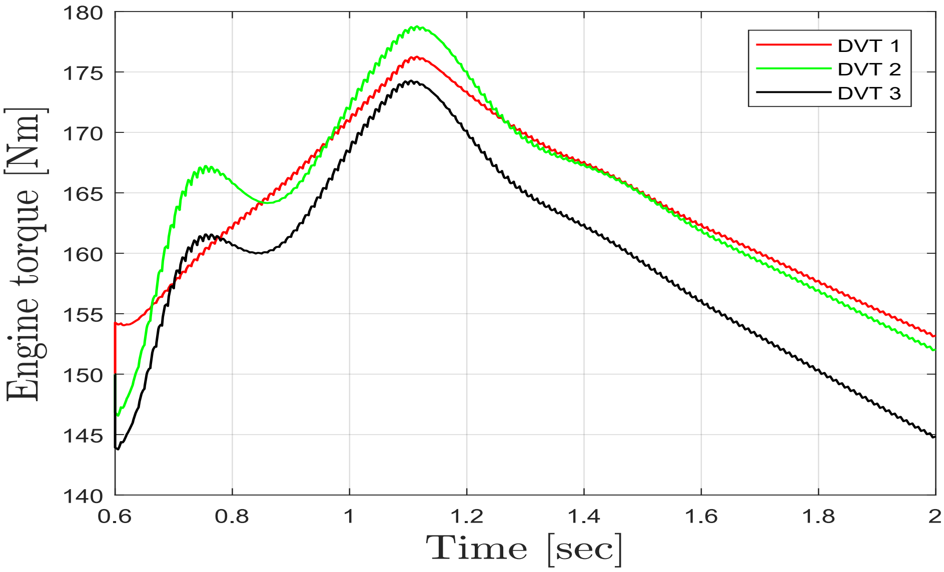


Figure 5.24: Engine torque Triple output control.

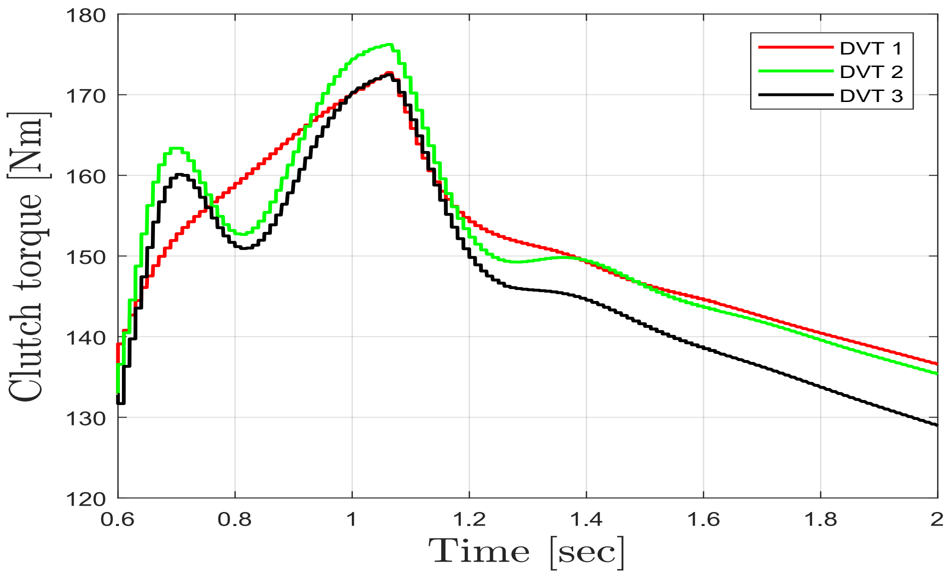


Figure 5.25: Clutch torque Triple output control.

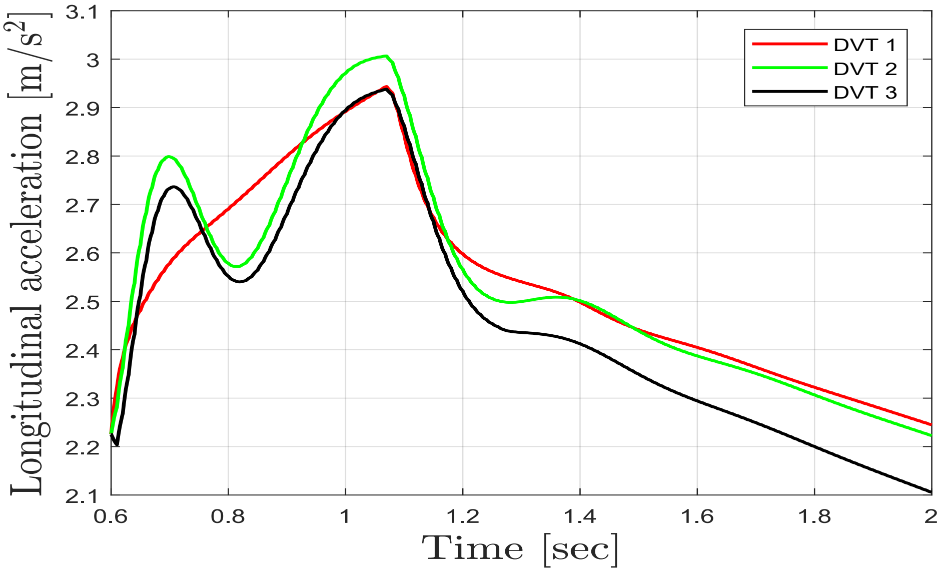
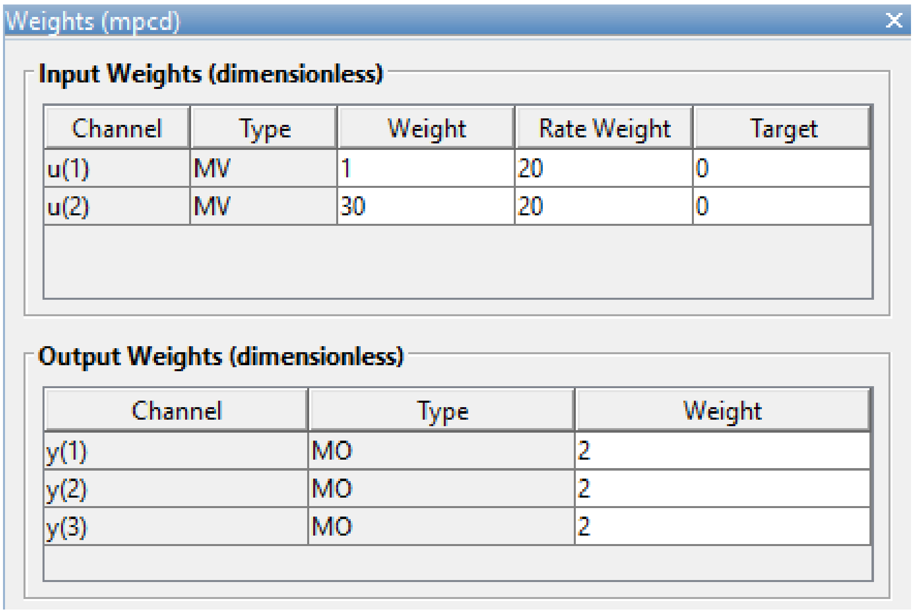


Figure 5.26: Longitudinal acceleration Triple output control.

All the plots put in evidence the fact that the so called "Double Manipulated Variable" configurations guarantee a relatively quick convergence of the slipping speed (as it was already able to do the configuration with only one manipulated variable) and moreover ensure, with some of the tuning parameters sets, a decrease of the drive shaft oscillations.

5.5 Behaviour with disturbance on engine torque

To make the controller handling the presence of a signal adding to the engine torque, a new set of tuning parameters has been chosen: it is shown in the following figure.


Weights (mpcd)

Input Weights (dimensionless)

Channel	Type	Weight	Rate Weight	Target
u(1)	MV	1	20	0
u(2)	MV	30	20	0

Output Weights (dimensionless)

Channel	Type	Weight
y(1)	MO	2
y(2)	MO	2
y(3)	MO	2

Figure 5.27: MPC parameters.

The prediction and control horizon used for simulations are listed here:

Plot name	H_p	H_c
Dist 1	75	2
Dist 2	80	2
Dist 3	85	2

Table 5.5: Prediction and Control horizon values.

Simulation results are illustrated by the following plots:

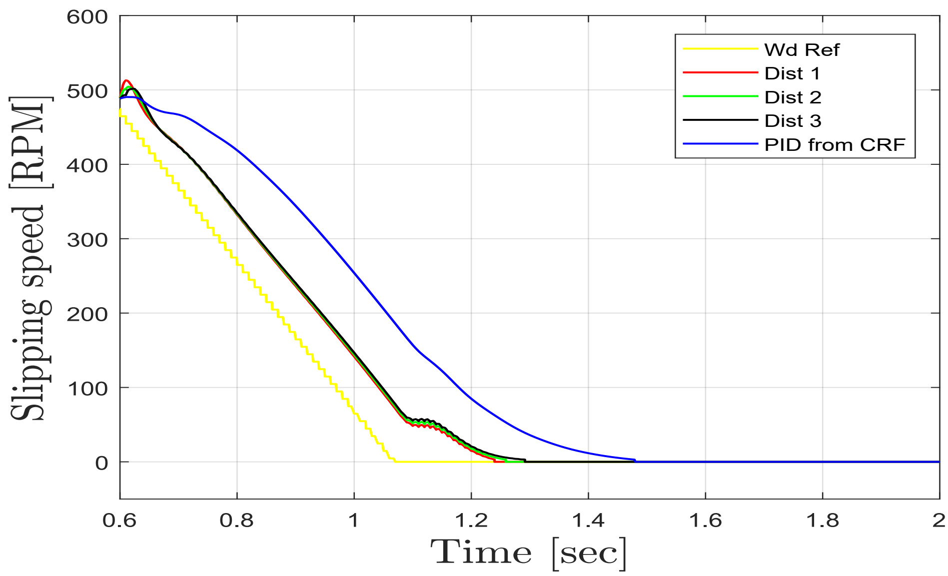


Figure 5.28: Slipping speed with driver disturbance.

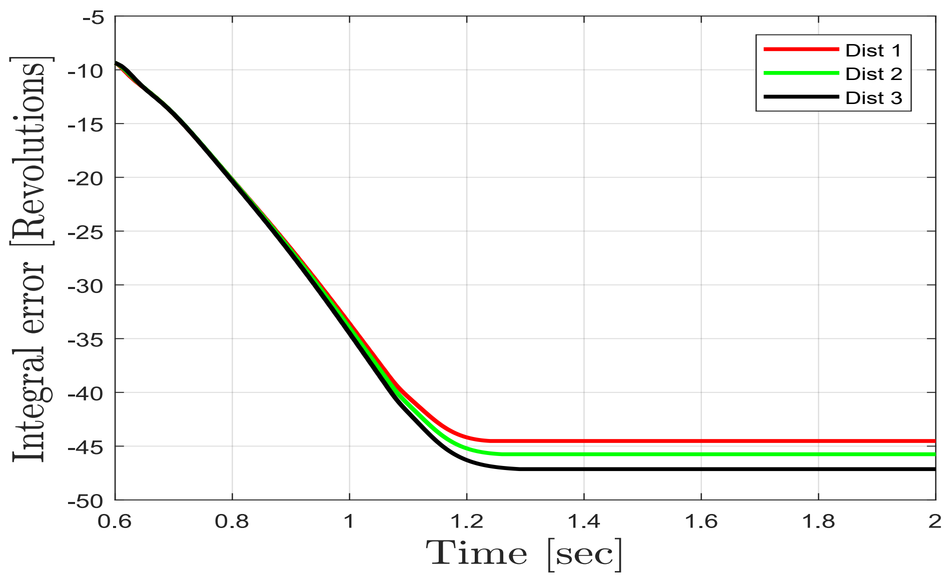


Figure 5.29: Integral error with driver disturbance.

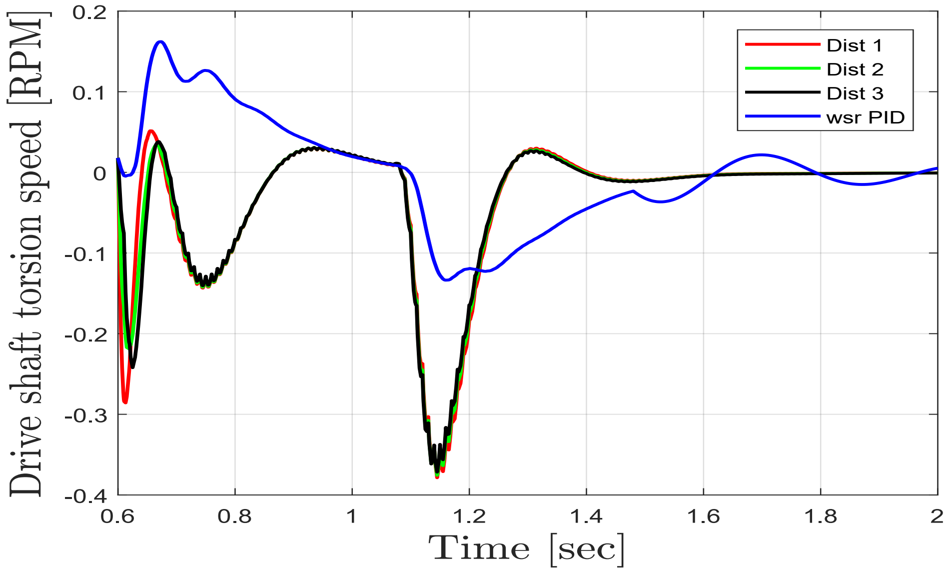


Figure 5.30: Drive shaft torsion speed with driver disturbance.

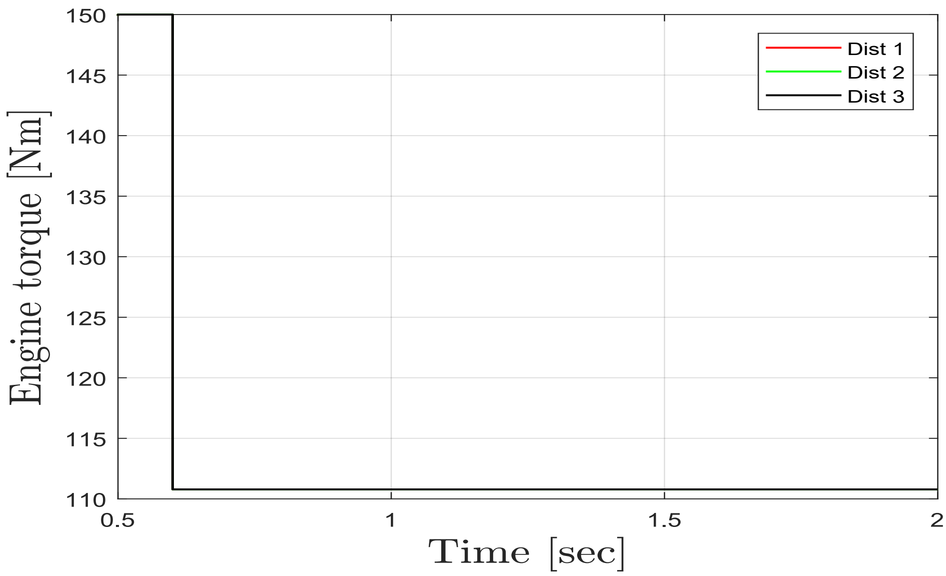


Figure 5.31: Engine torque with driver disturbance.

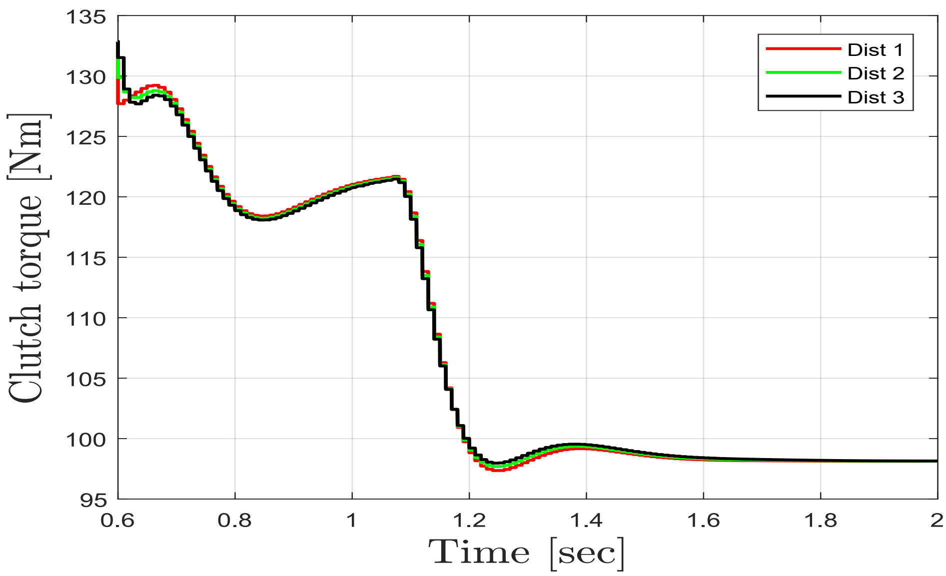


Figure 5.32: Clutch torque with driver disturbance.

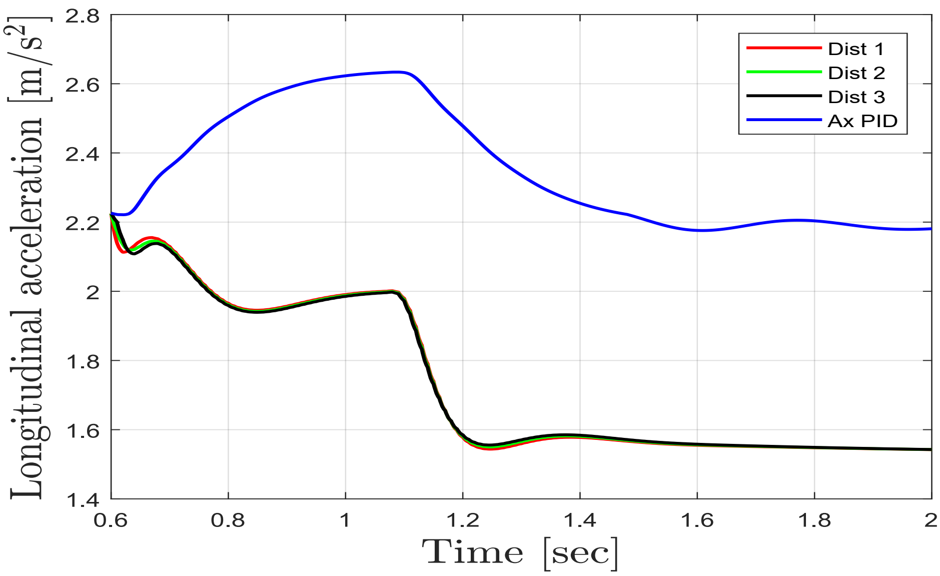


Figure 5.33: Longitudinal acceleration with driver disturbance.

5.6 Comparison between simulation results

In this section comparisons between the obtained results are shown, with the aim of understanding which is the model and the control strategy that ensure best performances.

In order to evaluate the goodness of a method, the convergence speed of the primary output is not the only parameter to take into account: the amplitude of the oscillations is an important parameter too. In particular, the time $\Delta t = t_f - t_s$ is considered: it represents the time interval between the starting control time ($t_s = 0.6s$) and the time t_f when the slipping speed ω_d reaches the null value.

With respect to the oscillations, the standard deviation σ is considered: it is a measure of the variation/dispersion of a set of data with respect to the mean value. A low standard deviation indicates that the data points tend to be close to the mean of the set, while a high standard deviation indicates that the data points are spread out over a wider range of values. The standard deviation formula is the following:

$$\sigma = \sqrt{\frac{\sum_{i=1}^{N} (x_i - \mu)}{N}} \tag{5.1}$$

where x_i represents a particular value of the distribution of the quantity of interest and μ is the mean value.

In calculating these parameters the mean values of the longitudinal acceleration and of the drive shaft torsion speed have been considered as the mean values computed in the entire time interval from $0s$ to $2s$.

In Table 5.6 the time interval Δt and the standard deviation for the drive shaft torsion speed $\sigma_{\omega_{sr}}$ and for the longitudinal acceleration σ_{A_x} of each model are listed.

In Table 5.7 the results for the configuration with additional signal on engine torque are reported.

It is important to specify that in the Δt calculation the digits up to the hundredth of a second have been taken into account.

Model	Δt	$\sigma_{\omega_{sr}}$	σ_{A_x}
PID	0.88	0.36	1.33
SMV 1	0.66	0.36	1.33
SMV 2	0.68	0.37	1.33
SMV 3	0.7	0.37	1.33
DMV 1	0.7	0.36	1.32
DMV 2	0.63	0.35	1.33
DMV 3	0.65	0.35	1.31
DVI 1	0.72	0.36	1.33
DVI 2	0.62	0.35	1.33
DVI 3	0.65	0.35	1.31
DVT 1	0.74	0.36	1.33
DVT 2	0.62	0.35	1.33
DVT 3	0.65	0.35	1.31

Table 5.6: Comparison of performances for models without additional signal on manipulated inputs.

Model	Δt	$\sigma_{\omega_{sr}}$	σ_{A_x}
Dist 1	0.64	0.32	1.17
Dist 2	0.66	0.32	1.17
Dist 3	0.69	0.32	1.17

Table 5.7: Comparison of performances for models with additional signal on engine torque.

5.6.1 Observations

As it is possible to notice by observing Table 5.6, the main improvement in performances between single manipulated variable and double manipulated variable is represented by the time the slipping speed takes to reach the desired value. In this regard the tuning parameters providing best results are those used in the plots called "DVI 2" and "DVT 2".
The oscillations of the drive shaft torsion speed ω_{sr} and of the longitudinal acceleration A_x decrease significantly in the configuration where an external signal is directly added to the manipulated variable C_m and is considered by the controller as a measured disturbance.

Table 5.7 shows that the last analysed configuration gives equivalent results in terms of oscillations using the three different proposed sets of tuning parameters. The unique difference is represented by the Δt time interval: the first model is the one that guarantees faster tracking of the slipping speed reference signal.

Chapter 6

Introduction to Explicit MPC

The main reason why Explicit MPC is introduced in this thesis is that it is able to remove one of the main drawbacks of MPC: the need of solving a mathematical program on-line to compute the control action. Explicit MPC is based on the multiparametric programming techniques, and allows to solve off-line the optimization problem, so that on-line operations reduce to a simple function evaluation. Such a function is piecewise affine in most cases.

The optimization problem is solved for all $x(t)$ within a given set X, that here is assumed to be polytopic:

$$X = \{x \in \mathbb{R}^n : S_1 x \leq S_2\} \subset \mathbb{R}^n \tag{6.1}$$

The idea is to make $u(t)$ depending *explicitly* on $x(t)$ rather than *implicitly*. Most of the time such a dependence is piecewise affine and the MPC controller can be represented in the following way:

$$u(x) = \begin{cases} F_1 x + g_1 & if \quad H_1 x \leq k_1 \\ \vdots & \vdots \quad \vdots \quad \vdots \\ F_M x + g_M & if \quad H_M x \leq k_M \end{cases}$$

On-line computations are reduced to the simple evaluation of the previous system of equations. This makes Explicit MPC useful for applications that require small sample times.

Details about multiparametric programming formulation can be found in the survey from Alessandro Alessio and Alberto Bemporad [12].

6.1 Explicit MPC in Matlab

A Simulink block is available in order to use the explicit MPC. To make it working properly it is necessary to set in the right way the ranges of parameters such as state values and manipulated variables.

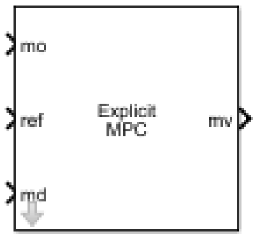


Figure 6.1: Explicit MPC block in Simulink.

In this thesis project the problem of writing analytically the explicit control law is not addressed: the aim is only to show that using the explicit MPC block in the Simulink model of the system are obtained exactly the same results as those produced by the implicit MPC, as it is possible to see in Figure 6.2 where the system is simulated in the configuration with the additional signal on engine torque.

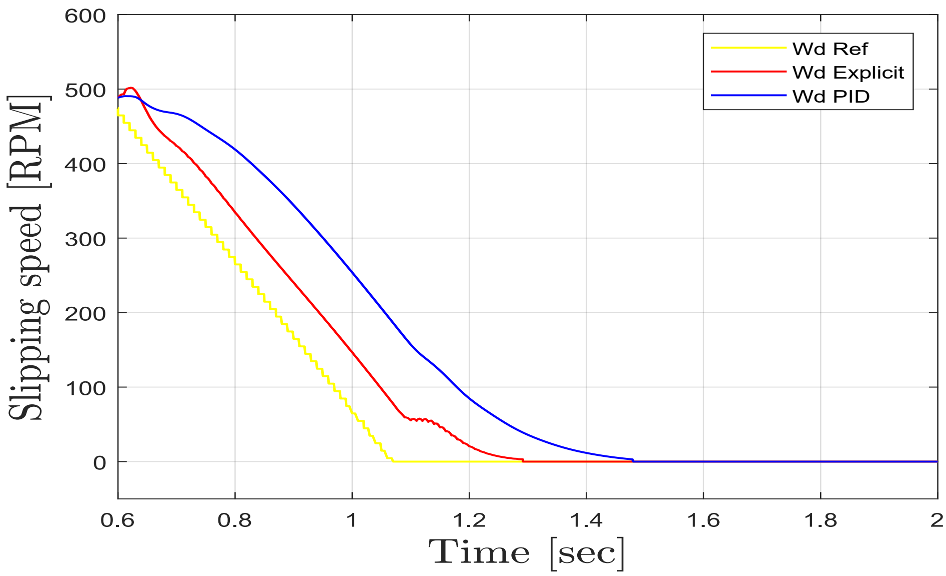


Figure 6.2: Slipping speed obtained with explicit MPC.

Chapter 7

Conclusions

This thesis has dealt with the control of the dry dual clutch during its slipping phase. In particular, the main purpose of the project has been to design a controller that, computing properly the clutch and the engine torque, is able to ensure a smooth tracking of the reference clutch slipping speed in order to improve driver comfort and drivability.

The Model Predictive Control technique has been used: it has been described from a theoretical point of view, then the advantages of using this technique have been explained; in particular the main reason is represented by the fact that this method can be easily extended to MIMO systems and is more flexible with respect to more traditional techniques.

In this regard, different configurations have been used in order to exploit the possible advantages caused by different control strategies.

As a starting point, a model where the unique controlled output is the clutch slipping speed and the only manipulated variable is the clutch torque has been considered.

Then a new control strategy has been adopted in order to try to achieve better results: the engine torque has been considered as a new manipulated variable, and two more quantities have been considered as controlled outputs: the integral error, that is the integral of the difference between the reference clutch slipping speed and the measured output, and the drive shaft torsion speed.

In addition to the basic model, the engine torque actuator has been inserted in Simulink and subsequently it has been compensated by the introduction of a lead network.

For both configurations simulation results corresponding to different controller parameters settings have been presented.

The best set of parameters used for simulations have been chosen after performing an extensive tuning procedure: both configurations ensure satisfactory performances. In particular, it is shown that the Model Predictive Controller ensures better performances with respect to the PID controller previously designed by Centro Ricerche Fiat.

In changing the configurations of the model, the MPC Toolbox available in Matlab turns out to be very flexible since it can be used to control a great variety of systems through a general approach.

In the end an introduction to Explicit MPC has been presented: the main purpose was to show that it guarantees exactly the same results obtained with "classical" MPC.

By observing simulation results it is clear that the configuration with two manipulated variables ensures best results in terms of slipping speed reference tracking and oscillations reduction: this justify the choice to adopt the engine torque as a manipulated input instead of considering it as an external non modifiable disturbance. The strategy of controlling other two outputs does not bring significant improvements in terms of performances: the most important elements in this regard are instead represented by the prediction and control horizons.

Future works

Starting from the results presented in this thesis project, further studies can be conducted and different applications can be explored. In particular it is possible to:

- deeply investigate the explicit version of the model predictive controller trying to extract the analytical expression of the control law
- export the explicit control law on a suitable microcontroller in order to test it experimentally on a real prototype

Bibliography

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

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

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

[4] Jinsung Kim, Kwanghyun Cho, and Seibum B. Choi. *Gear Shift Control of Dual Clutch Transmissions with a Torque Rate Limitation Trajectory*, 2011 American Control Conference on O'Farrell Street, San Francisco, CA, USA. June 29 - July 01, 2011

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

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

[7] J. B. Rawlings and D. Q. Mayne. *Model Predictive Control: Theory and Design*, Nob Hill publishing, Madison, Wisconsin

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

[9] W.H. Kwon and S. Han. *Receding Horizon Control, Model Predictive Control for State Models*, Springer

[10] L. Magni and R. Scattolini. *Complementi di controlli automatici*, Pitagora, 2006

[11] Carlo Novara. *Slides of the course "Nonlinear control and aerospace applications"*

[12] Alessandro Alessio and Alberto Bemporad. *A Survey on Explicit Model Predictive Control*