Takagi-Sugeno Fault Tolerant Control of an Autonomous Vehicle

Author: **Konstantinos Spanos**

Director: **Dr. Vicenç Puig Cayuela**

A thesis submitted for the

Master's Degree in Automatic Control and Robotics

Escola Tecnica Superior d'Enginyeria Industrial de Barcelona

(ETSEIB)

Universitat Politècnica de Catalunya

Barcelona, Spain

April 2018

Abstract

This thesis is devoted to design a Takagi-Sugeno Fault Tolerant model for the longitudinal and lateral control of autonomous vehicles. Employing the kinematic and dynamic non-linear models of the vehicle, a gain-scheduling state-feedback controller for each model, as well as a state observer for the dynamic model, have been designed to control both vehicle behaviors. The design of the controllers and the observer has been implemented by solving LQR and Kalman filter formulated as linear matrix inequalities problems. Moreover, so as to achieve the desired performance for tracking the references of position, orientation, linear and angular velocities, the combination of both controllers is presented. Additionally, a trajectory planner that provides the references to the kinematic model is used. The ultimate goal of this project, is to compensate the faults of the actuators, which is achieved by employing various techniques, as the Unknown Input Observer, Augmented State Observer and Least Square Parameter Estimation. The performance of the specific control algorithm has presented remarkable simulation results, both in the decoupled and the cascade manner. Taking into account these results, we could verify that the complete control model can be used in real applications of autonomous vehicles.

Acknowledgments

First, I would like to thank my advisor Prof. Vicenç Puig Cayuela, for his endless patience, tutoring and support throughout the implementation of this thesis. He gave me the opportunity to work, learn, and believe to my abilities in an exceptional way.

The very special thanks go to my grandparents Anastasia and Nikolaos, for supporting me during all the years of my studies, in all the ways they could.

I would also like to say a big "thank you" to my aunt Jenny, for what she did for me.

Last but not least, special thanks to Adriana, for her tolerance, patience and understanding, during this period.

Contents

ETSEIB

List of Figures

List of Tables

List of Acronyms

T-S	Takagi-Sugeno
LMI	Linear Matrix Inequalities
LTI	Linear Time Invariant
GEVP	Generalized Eigenvalue Problem
PDC	Parallel Distributed Compensation
PDI	Polytopic Differential Inclusion
BIBO	Bounded-Input, Bounded-Output
LQC	Linear Quadratic Controller
LQR	Linear Quadratic Regulation
FTC	Fault Tolerant Control
FD	Fault Diagnosis
UIO	Unknown Input Observer
ASO	Augmented State Observer
LSPE	Least Squares Parameter Estimation
CoG	Center of Gravity
RMSE	Root Mean Square Error

Chapter 1

Introduction

Nowadays, we are witnessing the technologies related to unmanned/autonomous vehicles to become more and more conventional in our lives. A rapid increase in production of this kind of vehicles like drones, autonomous mobile robots and autonomous cars, have been demonstrated over the last decade. It can be observed that big companies, and mainly automotive producers have increased their interest in such a way that it becomes their main research and developing goal. The evolution of this kind of technology had been launched at first because of military purposes, which pushed the research community to deal with that like never before. Recently, the needs of the society have changed, and people require new services. These kind of needs like safer and more relaxed transportation, as well as safety in dangerous working environments or even better delivery services, have motivated the researchers to improve the autonomous vehicle technologies.

An autonomous vehicle consists of four fundamental technologies: environment perception and modeling, localization and map building, path planning and decision-making, and motion control. The environment perception and modeling module is responsible for sensing environment structures in a multi-sensor way and providing a model of the surrounding environment. Here, the environment model includes a list of moving objects, such that static obstacles, vehicle position relative to the current road, the road shape, etc. Finally, this module provides the environment model and the local map to the localization and map building module by processing the original data, vision, LIDAR, and radar. The goal of the second module, vehicle localization and map building is to generate a global map by combining the environment model, a local map and global information so as to determine the vehicle's position, and to estimate the locations of geometric features. The third module, path planning and decision-making is in

charge of ensuring that the vehicle is moving according to the rules of the ground and environment frame, safety, comfortability and the vehicle dynamics. Therefore, this module improves the efficiency and the desired path generation. Finally, the motion control provides specific commands, by translating the decision made by the previous modules, to the actuators of the vehicle.

In this thesis the analysis of the motion control module is done, as well as the implementation of the control algorithm in order to control the movements of a bicycle-like vehicle. Vehicle control can be broadly divided into two categories: lateral control and longitudinal control. The longitudinal control is related to distance–velocity control between vehicles for safety and comfort purposes. Here some assumptions are made about the state of vehicles and the parameters of models. The lateral control maintains the vehicle's position in the lane center, and it can be used for vehicle guidance assistance. Moreover, it is well known that the lateral and longitudinal dynamics of a vehicle are coupled in a combined lateral and longitudinal control, where the coupling degree is a function of the tire and vehicle parameters. The construction of the controller algorithm is done under some specific requirements, like the accuracy of following the path receiving from the path planing module, as well as the smoothness of the movement and the desired time of reaction to the new orders. In order to control the vehicle movement it is needed to know the next position that the vehicle has to move, as well as the linear velocity and steering angle of the wheels. Consequently, in order to achieve the desired performance of the vehicle, two different controllers are required, a kinematic and a dynamic one. The kinematic controller is in charge of controlling the position, orientation and linear velocity by means of actuating over the linear and angular velocities of the vehicle and the dynamic one addresses the tracking of the linear and angular velocity references of the vehicle by applying force to the rear wheels and an angle to the front wheels.

In order to control the vehicle behavior, a model which is constructed from the physical characteristics and parameters is used. In general, this model is a non-linear model and normally is quite complex to control it. In order to reduce the level of complexity it can be simplified with the aim of easing the computations.

The design of the controller is a process which consists of the vehicle mathematical model and the control method which is chosen. The controller can be designed using linearization of the original model by applying linear techniques. In this method, the very well-studied classical control techniques can be applied although the non-linearities of the original model are not taken into account which implies to non-desired behavior of the controller system out

of the linearization point. Several nonlinear methods that solve these specific problems exist. An alternative control approach is to consider the non-linearities of the system and apply some of the existing non-linear methods. In this case the accuracy of the algorithm is the advantage comparing to the forementioned method, although the complexity of these methods make them not so appealing. The above methods required non-systematic tuning to guarantee stability and robustness of a complex system. An alternative approach is the use of linear like techniques which can address the nonlinear control problem in an efficient way. This thesis employs one of these methods, the Takagi-Sugeno (T-S) fuzzy logic control approach, in order to control the vehicle kinematic and dynamic models in a robust manner. Using the Takagi-Sugeno method a controller can be designed, capable of automatically varying its gains, and working in different regions of operating points and taking into account the nonlinear model. However, since this method follows a parametric approach, the complexity of the problem is increased with the number of varying parameters.

In order to solve the linear like models obtained from T-S method a linear optimization algorithm is needed. This algorithm is the Linear Matrix Inequalities (LMI) technique which can provide the desired controller and observer gains. The LMI method can be adjusted and in order to obtain a stable controller, as well as under the desired performance.

The process of designing the controller consists of some steps. First, it is needed to obtain the kinematic and dynamic models of the vehicle by its physical modelling. Second, the transformation of these models in a T-S representation. The third step is the construction of the controller and/or observer algorithms by means of LMI techniques. Specifically, in this thesis there is a last step. The combination of the Kinematic and Dynamic controllers and Dynamic observer in order to obtain the global cascade control algorithm.

Finally, another very important aspect of controlling an autonomous vehicle, is the fault tolerant control. In general, technological systems are vulnerable to faults. Actuator faults reduce the performance of control systems and may even cause a complete break-down of the system. Erroneous sensor readings are the reason for operating points that are far from the optimal ones. Wear reduces the efficiency and quality of a system. In many faulty situations, the system operation has to be stopped to avoid damage to machinery and humans.

As a consequence, the detection and handling of faults play an increasing role in modern technology, where many highly automated components interact in a complex way such that a fault in a single component may cause the malfunction of the whole system.

The classic way of fault diagnosis boils down to controlling the limits of single variables

and then using the resulting knowledge for fault alarm purposes. Apart from the simplicity of such an approach, the observed increasing complexity of modern systems necessitates the development of new fault diagnosis techniques. On the other hand, the resulting fault diagnosis system should be suitably integrated with the existing control system in order to prevent the development of faults into failures, perceived as a complete breakdown of the system being controlled and diagnosed.

Such a development can only be realized by taking into account the information hidden in all measurements. One way to tackle such a challenging problem is to use the so-called model-based approach. Indeed, the application of an adequate model of the system being supervised is very profitable with respect to gaining the knowledge regarding its behavior. A further and deeper understanding of the current system behavior can be achieved by implementing parameter and state estimation strategies. Then, the obtained estimates can be used for supporting diagnostic decisions and increasing the control quality, while the resulting models (along with the knowledge about their uncertainty) can be used for designing suitable control strategies.

In this thesis, to estimate the faults/disturbances that affect the actuators (longitudinal, lateral) and compensate them to the system, various of the estimation methods are employed. Firstly, the Unknown Input Observer technique is used to estimate these faults and some disturbances of the actuators, as well as the dynamic states. Subsequently, the Augmented State Observer method is employed, in order to estimate the faults as augmented states to the system and the dynamic states as well. Ultimately, the Least Squares Parameter Estimation technique is applied for estimating the faults. After investigating which of these methods provide reliable performance/operation, the best ones are employed. Specifically, the combination of the Unknown Input Observer with the Least Squares Parameter Estimation is used.

Chapter 2

Background Theory

Fuzzy control systems is an important tool which allows representing and implementing human heuristics to control systems. Fuzzy models, are designed for describing non-linear systems as a collection of Linear Time Invariant (LTI) models blended together with system non-linearities. These fuzzy logic functions, called weighting functions, depend on measurable premise variables (inputs, outputs of the system or external variables). The Takagi–Sugeno (T–S) fuzzy structure, offers an efficient representation of non-linear systems while remaining relatively simple compared to general nonlinear models. Another advantage of this system representation is that some results developed in the linear concept can be extended to T–S fuzzy models. Using the T–S fuzzy models has caused research on fuzzy controller design to gain great interest in recent years. These include stability analysis, incorporation of the performance index and others such as robustness and numerical implementations. In order to design fuzzy control systems, the process is based on the concept of Parallel Distributed Compensation (PDC). The main idea of controller design is to derive each control rule in order to compensate each rule of the fuzzy system. The stability of T–S fuzzy models and the design of T–S fuzzy control laws are, in most cases, addressed using the direct Lyapunov approach leading to a set of Linear Matrix Inequalities (LMIs), which can be solved efficiently by using the existing optimization techniques. To find a solution to the stabilization problem in T–S fuzzy control systems, the standard approach is based on looking for a common quadratic function that satisfies sufficient conditions to guarantee stability in the Lyapunov sense. Most of these conditions can be converted into LMI constraints, solvable through convex optimization techniques. The inherent flexibility of the LMI approach allows to obtain fuzzy controllers that guarantee both stability and performance of closed-loop systems.

Bibliography [4].

2.1 Takagi-Sugeno Fuzzy Model

A fuzzy control system is a control system based on fuzzy logic - a mathematical system that analyzes analog input values in terms of logical variables that take on continuous values between 0 and 1, in contrast to classical or digital logic, which operates on discrete values of either 1 or 0 (true or false, respectively).

The fuzzy model proposed by Takagi and Sugeno [19] is described by fuzzy IF-THEN rules which represent local linear input-output relations of a nonlinear system. The main feature of a Takagi-Sugeno fuzzy model is to express the local dynamics of each fuzzy implication (rule) by a linear system model. The overall fuzzy model of the system is achieved by fuzzy "blending" of the linear system models. In fact, it is proved that Takagi-Sugeno fuzzy models are universal approximators.

2.1.1 Fuzzy Model design

Figure 2.1 illustrates the model-based fuzzy control design approach. To design a fuzzy controller, we need a Takagi-Sugeno fuzzy model for a nonlinear system. Therefore the construction of a fuzzy model represents an important and basic procedure in this approach. In this section, it is discussed the procedure of how to construct such a fuzzy model.

In general there are two approaches for constructing fuzzy models:

1. Identification (fuzzy modeling) using input-output data and
2. Derivation from given nonlinear system equations.

There has been an extensive literature on fuzzy modeling using input-output data following Takagi's, Sugeno's, and Kang's excellent work [20, 21]. The procedure mainly consists of two parts: structure identification and parameter identification. The identification approach to fuzzy modeling is suitable for plants that are unable or too difficult to be represented by analytical and/or physical models. On the other hand, nonlinear dynamic models for mechanical systems can be readily obtained by, for example, the Lagrange method and the Newton-Euler method. In such cases, the second approach, which derives a fuzzy model from given nonlinear dynamical models, is more appropriate. This thesis focuses on the second approach which utilizes the idea of "sector non-linearity" to construct fuzzy models.

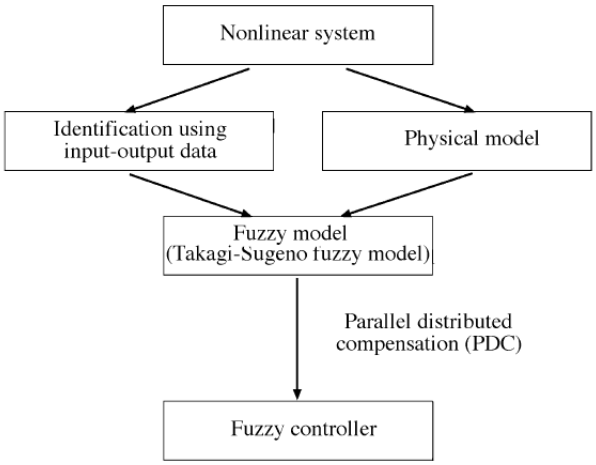

Figure 2.1: Model-based fuzzy control design.

Sector Nonlinearity

Sector nonlinearity is based on the following idea. Consider a simple nonlinear system $\dot{x}(t) = f(x(t))$, where $f(0) = 0$. The aim is to find the global sector such that $\dot{x}(t) = f(x(t)) \in [a_1\ a_2]x(t)$. Figure 2.2 illustrates the sector nonlinearity approach. This approach guarantees an exact fuzzy model construction. However, it is sometimes difficult to find global sectors for general nonlinear systems. In this case, we can consider local sector nonlinearity. This is reasonable as variables of physical systems are always bounded. Figure 2.3 shows the local sector nonlinearity, where two lines become the local sectors under $-d < x(t) < d$. The fuzzy model exactly represents the nonlinear system in the "local" region, that is, $-d < x(t) < d$.

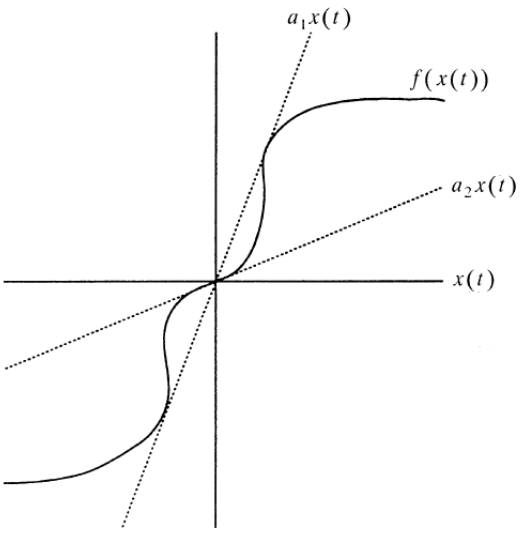

Figure 2.2: Global sector nonlinearity.

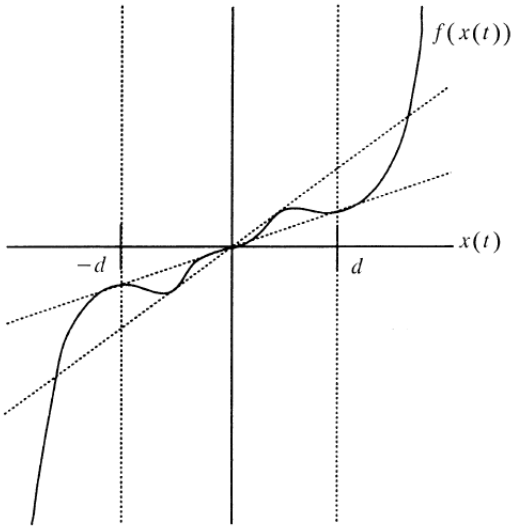

Figure 2.3: Local sector nonlinearity.

Remark 1 *Prior to applying the sector nonlinearity approach, it is often a good practice to simplify the original nonlinear model as much as possible. This step is important for practical applications because it always leads to the reduction of the number of model rules, which reduces the effort for analysis and design of control systems.*

2.1.2 T-S Fuzzy Model

The controller design procedure is based on the representation of a given nonlinear plant in terms of the fuzzy model given by (eq.2.1). The antecedent part of each rule $\mathbf{R}_i$ contains fuzzy linguistic descriptions M_{ji} of the scheduling variables $\delta_j(t)$ and the consequent part contains a local linear model of the nonlinear system

$\mathbf{R}_i$: **IF** δ_1 is $M_{1\mathrm{i}}$ **and**...**and** $\delta_j(t)$ is M_{ji} **then**

$$\begin{cases} \dot{x}_i(t) = A_i x(t) + B_i u(t) \\ y_i(t) = C_i x(t) + D_i u(t) \end{cases} \tag{2.1}$$

The entire fuzzy model of the plant (eq.2.1) is obtained by fuzzy blending of the consequent submodels. For a given pair of vectors $x(t)$ and $u(t)$, the final output of the fuzzy system is inferred as a weighted sum of the contributing submodels

$$\dot{x}(t) = \frac{\sum_{i=1}^{r} w_i(\delta(t))[A_i x(t) + B_i u(t)]}{\sum_{i=1}^{r} w_i(\delta(t))} \tag{2.2}$$

$$y(t) = \frac{\sum_{i=1}^{r} w_i(\delta(t))[C_i x(t) + D\text{`}_i u(t)]}{\sum_{i=1}^{r} w_i(\delta(t))} \tag{2.3}$$

with $w_i(\delta(t)) = \mathrm{aggop}[M_{1\mathrm{i}}\delta_1(t)... \ M_{\mathrm{ji}}\delta_j(t)]$ where $w_i(\delta(t)) \geq 0$ is the degree of fulfillment of rule i, aggop$(\cdot)$ is the aggregation operator (for instance, the product or the minimum), $\sum_{i=1}^{r} w_i(\delta(t)) \geq 0$ for all $i = 1, 2, ..., r$. With $0 \leq h_i(\delta(t)) = \frac{w_i(\delta(t))}{\sum_{i=1}^{r} w_i(\delta(t))} \leq 1$, (eq.2.2) and (eq.2.3) can be written as

$$\dot{x}(t) = \sum_{i=1}^{r} h_i(\delta(t))[A_i x(t) + B_i u(t)] \tag{2.4}$$

$$y(t) = \sum_{i=1}^{r} h_i(\delta(t))[C_i x(t) + D_i u(t)] \tag{2.5}$$

The TS fuzzy model can also be regarded as a *quasilinear system*, i.e., a system linear in both $x(t)$ and $u(t)$ whose matrices $A(\cdot), ..., D(\cdot)$ are not constant, but varying:

$$\dot{x}(t) = A(\delta(t))x(t) + B(\delta(t))u(t) \tag{2.6}$$

$$y(t) = C(\delta(t))x(t) + D(\delta(t))u(t) \tag{2.7}$$

From (eq.2.4) and (eq.2.5), one can see that for all possible values of $\delta(t)$, which are assumed to be known online, these matrices are bounded within a polytope whose vertices are the matrices of the individual rules:

$$\begin{bmatrix} A(\delta(t)) & B(\delta(t)) \\ C(\delta(t)) & D(\delta(t)) \end{bmatrix} \in \mathbf{Co}\left\{ \begin{bmatrix} A_i & B_i \\ C_i & D_i \end{bmatrix} : i = 1, 2, ..., , r \right\} \tag{2.8}$$

where:

$$\mathbf{Co}\ \{\mathbf{S}_i : i = 1, 2, ..., r\} = \left\{ \sum_{i=1}^{r} h_i(t)\mathbf{S}_i : \sum_{i=1}^{r} h_i(t) = 1, h_i(t) \geq 0 \right\}$$

and

$$\mathbf{S}_i = \begin{bmatrix} A_i & B_i \\ C_i & D_i \end{bmatrix}$$

For the sake of simplicity, the direct transmission matrices D_i are considered to be zero here. This can be assumed without any restrictions to real systems because they have dynamic parts between their inputs and outputs. Note, the presented design method is particularly intended for control of nonlinear systems. Hence, the scheduling variables are usually a function of the state; i.e., $\delta(t) = \delta(x(t))$ and (eq.2.6) yields $\dot{x}(t) = A(x)x + B(x)u$, etc.

2.1.3 Fuzzy Controller

PDC

The Parallel Distributed Compensation (PDC) offers a procedure to design a fuzzy controller from a given T-S fuzzy model. To realize the PDC, a controlled object (nonlinear system) is first represented by a T-S fuzzy model. It is emphasized that many real systems, for example, mechanical systems and chaotic systems, can be and have been represented by T-S fuzzy models. In the PDC design, each control rule is designed from the corresponding rule of a T-S fuzzy model. The designed fuzzy controller shares the same fuzzy sets with the fuzzy model in the premise parts.

Controller design

The controller design by means of the described method begins with the determination of the linear submodels in some operating regions of interest of the nonlinear system to be controlled. Then, convex optimization techniques are used to design local controllers within a fuzzy gain-scheduling scheme with the desired overall behavior. In this way, a wide-range stabilization and control problems can be solved. In the continuous-time case, the simplest TS fuzzy control rule being considered here, has the form:

$\mathbf{R}_i$: IF δ_1 is $M_{1\mathrm{i}}$ and...and $\delta_j(t)$ is M_{ji} then

$$u_i(t) = -F_i x(t) + V_i r(t) \tag{2.9}$$

where $r(t)$ is a stepwise reference signal. The controller's output is inferred as the weighted mean

$$u(t) = \frac{\sum_{i=1}^{r} w_i(\delta(t))[-F_i x(t) + V_i r(t)]}{\sum_{i=1}^{r} w_i(\delta(t))} \tag{2.10}$$

which yields

$$u(t) = \sum_{i=1}^{r} h_i(\delta(t))[-F_i x(t) + V_i r(t)] = -F(\delta(t))x(t) + V(\delta(t))r(t) \tag{2.11}$$

If the scheduling vector $\delta(t)$ is a function of the state vector $x(t)$, $u(t)$ represents a nonlinear gain-scheduled control law.

The goal of the controller design is to determine the constant matrices F_i and V_i such that the desired dynamics of the closed-loop system and some desired steady-state input–output behavior are obtained. Designing the state-feedback gains F_i requires dealing with the system dynamics and hence ensuring stability. This problem is solved by means of LMIs. For the TS fuzzy controller (eq.2.9), the best values for the static feed-forward gains are given by

$$V_i = (C_i(-A_i + B_i F_i)^{-1} B_i)^{-1} \tag{2.12}$$

This ensures for each closed-loop subsystem a unit steady-state gain. However, a reasonable requirement for the controller (eq.2.9) based on (eq.2.12) is rather to satisfy $x(t) \to 0$ when $t \to \infty$. This implies the stabilization problem of the control system where $r(t) = 0$. In

other words, this TS fuzzy controller cannot usually be used satisfactorily in tracking control problems where a given reference trajectory $r(t) \neq 0$ is to be followed. The reasons for this are the ever-present mismatch between the fuzzy model and the real plant and also the dynamic of the reference signal resulting in steady-state errors.

Remark 2 *Although the fuzzy controller (eq.2.11). is constructed using the local design structure, the feedback gains F_i should be determined using global design conditions. The global design conditions are needed to guarantee the global stability and control performance.*

Bibliography: [1,3–5,11]

2.2 LMI Techniques for Analysis and Synthesis

Recently a class of numerical optimization problems called linear matrix inequality (LMI) problems has received significant attention. The origin of Linear Matrix Inequalities (LMIs) goes back as far as 1890, although they were not called this way at that time, when Lyapunov showed that the stability of a linear system $\dot{x} = Ax$ is equivalent to the existence of a positive definite matrix P, which satisfies the matrix inequality $A^T P + PA < 0$, expression which will be clarified below. These optimization problems can be solved in polynomial time and hence are tractable, at least in a theoretical sense. The recently developed interior-point methods for these problems have been found to be extremely efficient in practice. For systems and control, the importance of LMI optimization stems from the fact that a wide variety of system and control problems can be recast as LMI problems. Except for a few special cases these problems do not have analytical solutions. However, the main point is that through the LMI framework they can be efficiently solved numerically in all cases. Therefore recasting a control problem as an LMI problem is equivalent to finding a "solution" to the original problem.

2.2.1 Definition of an LMI

An LMI is a matrix inequality of the form

$$F(x) = F_0 + \sum_{i=1}^{m} x_i F_i x(t) > 0$$

where

- $x^T = (x_1, x_2, ..., x_m) \in \mathbb{R}^m$ is the vector of the m variables,

- $F_i = F_i^T \in \mathbb{R}^{nxn}$ are given symmetric matrices,
- The inequality symbol > 0 means that $F(x)$ is positive definite

There are also nonstrict LMIs, of the form $F(x) \geq 0$, where $\geq$ means that the matrix $F(x)$ is positive semidefinite, and LMIs of the form $F(x) < 0$ which are obviously equivalent to $-F(x) > 0$.

2.2.2 Standard Problems Involving LMIs

There are two main classes of optimization problems with constraints, expressed as LMIs.

LMI Feasibility Problems

This problem consists in finding an $x \in \mathbb{R}^m$ solution to the LMI $F(x) > 0$, or to determine that this LMI is infeasible, i.e. that no such x exists.

Eigenvalue Problems

The eigenvalue problem consists in minimizing the eigenvalue of a matrix, which depends affinely on some variable, subject to an LMI constraint:

$$\begin{aligned} &\text{minimize} \quad \lambda, \\ &\text{subject to} \quad \lambda I - A(x) > 0, \quad B(x) > 0 \end{aligned}$$

Such problems appear often in the equivalent form of minimizing a linear function of x subject to an LMI:

$$\begin{aligned} &\text{minimize} \quad c^T x, \\ &\text{subject to} \quad F(x) > 0 \end{aligned}$$

2.2.3 Closed-Loop System

The closed-loop system consisting of the fuzzy model and the fuzzy controller is obtained by substituting the controller (eq.2.11) to the state equation of the fuzzy model (eq.2.4). The closed-loop system is given by

$$\dot{x}(t) = \sum_{i=1}^{r} \sum_{j=1}^{r} h_i(\delta(t)) h_j(\delta(t)) \times [[A_i - B_i F_j] x(t) + B_i V_j r(t)] \tag{2.13}$$

It is assumed that the weight of each rule in the fuzzy controller is equal to that of the corresponding rule in the fuzzy model - we call this the shared rules principle. This assumption is easy to satisfy since all weighting factors of the controller can be simply taken over from the known fuzzy model. Then, (eq.2.13) can be rewritten as

$$\begin{aligned} \dot{x}(t) = \sum_{i=1}^{r} h_i(\delta(t)) h_i(\delta(t)) G_{ii} + 2 \sum_{i}^{r} \sum_{j>i}^{r} h_i(\delta(t)) h_j(\delta(t)) \frac{G_{ij}+G_{ji}}{2} x(t) \\ + \sum_{i=1}^{r} \sum_{j=1}^{r} h_i(\delta(t)) h_j(\delta(t)) B_i V_j r(t) \end{aligned} \tag{2.14}$$

with

$$G_{ij} = A_i - B_i F_j \tag{2.15}$$

For the particular case of common matrices B_i, i.e., $B_i = B$ for all submodels $i = 1, 2, ..., r$, and for the shared rules, the following simplified description of the entire closed-loop system can be derived:

$$\dot{x}(t) = \sum_{i=1}^{r} h_i(\delta(t)) [(A_i - B F_i) x(t) + B V_i r(t)] \tag{2.16}$$

The terms known from standard PDC controllers are given by (eq.2.15). They are responsible for the stability of the control system-matrices F_j are calculated via LMIs such that an appropriate quadratic Lyapunov function can be found. The remaining terms given by the products $B_i V_j$ do not affect the dynamics; they are in the feedforward channel. They represent the steady-state gain of the control loop with V_j simple calculated as shown in (eq.2.12) so that the unity steady-state gain is ensured for the dynamic fuzzy system (eq.2.3) to follow the reference signal $r(t)$ as closely as possible.

2.2.4 Stability Conditions for Closed-Loop TS Fuzzy System

The considered systems are characterized by matrices $A(\cdot), ..., D(\cdot)$, $F(\cdot)$ and $V(\cdot)$ bounded in polytopes like (eq.2.8). There is no need to distinguish between external and internal nature of the time-varying parameters $\delta(\cdot)$ (internal stands for $\delta(x)$ where δ depends on some state variables x making the problem nonlinear) used for scheduling if the control system exhibits

its trajectories within the considered polytopic differential inclusion (PDI). The main idea is based on the fact that every trajectory of the considered system is also a trajectory of the PDI for which some properties can be guaranteed if a common Lyapunov function is found. Then, every trajectory of our (possibly nonlinear) system has these properties as well. Furthermore, starting from the formalism of Lyapunov function stability, it is quite straightforward to prove that using the presented design method for the considered class of control systems, BIBO stability is also guaranteed.

The common problem for fuzzy controller design is solved numerically, i.e., the stability conditions of the theorems are expressed in LMIs. The LMIs can then be solved to find a P or to determine that no such P exists. This is either the convex feasibility problem or the general eigenvalue problem. Numerically, these problems can be solved in polynomial time by means of powerful tools that became available lately. Introducing additional constraints on the locations of the eigenvalues of the underlying subsystems, some other useful performance criteria can be satisfied, e.g., the elimination of overshoots.

Theorem 1 *The equilibrium of the continuous-time closed-loop fuzzy control system described by (eq.2.16) is asymptotically stabilizable, if there are a common positive definite matrix Y and a set of matrices W_i for $i = 1, 2, .., r$ such that*

$$\mathcal{L}_{Y,W_i}(A_i, B) < 0, \tag{2.17}$$

$$\mathcal{L}_{Y,\gamma} > 0, \tag{2.18}$$

The linear operators $\mathcal{L}$ are defined for any matrix variables $Y \in \mathbb{R}^{n\times n}$ and $W_i \in \mathbb{R}^{m\times n}$ as shown in (eq.2.40),(2.41),(2.42). Then, the desired fuzzy state-feedback gain matrices F_i are given by $F_i = W_i Y^{-1}$, $i = 1, 2, ..., r$. The common matrix P can be obtained as $P = Y^{-1}$

It is easy to find $Y > 0$ and the corresponding W_i or to determine that no such Y, W_i exist. LMI-based techniques can be used for systematic analysis and also for the design of TS fuzzy control systems.

Bibliography: [1,8,9]

2.3 Enhancements

2.3.1 Fuzzy State Estimator

In Sections 2.1 and 2.2, where the state-feedback controller is implemented, all states of the plant have been implicitly assumed to be online available. However, in real processes, this is not always the case. To overcome this problem, a fuzzy observer can be used. Based on the plant's inputs and outputs, the observer estimates the states. The augmented fuzzy system that contains the observer and the controller is regarded as a *dynamic output-feedback fuzzy controller*. Thus, the demand for fuzzy observers is well motivated. Observers are known to satisfy the requirement $e(t) \to 0$ when $t \to \infty$, where $e(t) = x(t) - \hat{x}(t)$ means the deviation between the plant's state vector $x(t)$ and the state vector $\hat{x}(t)$ estimated by the observer. This requirement can be satisfied by a fuzzy observer based on the same model of the plant as the controller with an additional time-varying state injection matrix $L(\cdot)$

$\mathbb{R}_i$: **IF** δ_1 is $M_{1\mathrm{i}}$ **and**...**and** $\delta_j(t)$ is M_{ji} **then**

$$\begin{cases} \dot{\hat{x}}_i(t) = A_i x(t) + B_i u(t) - L_i[y(t) - \hat{y}(t)] \\ \hat{y}_i(t) = C_i \hat{x}(t) \end{cases} \tag{2.19}$$

where $i = 1, 2, ..., r$. For further considerations, the aforementioned fuzzy system can be expressed as

$$\begin{aligned} \dot{\hat{x}}_i(t) &= \frac{\sum_{i=1}^{r} w_i(\delta(t))[A_i x(t) + B_i u(t) - L_i(y(t) - \hat{y}(t))]}{\sum_{i=1}^{r} w_i(\delta(t))} \\ &= \sum_{i=1}^{r} h_i(\delta(t))[A_i x(t) + B_i u(t) - L_i(y(t) - \hat{y}(t))] \end{aligned} \tag{2.20}$$

$$\hat{y}_i(t) = \frac{\sum_{i=1}^{r} w_i(\delta(t)) C_i \hat{x}(t)}{\sum_{i=1}^{r} w_i(\delta(t))} = \sum_{i=1}^{r} h_i(\delta(t)) C_i \hat{x}(t) \tag{2.21}$$

The weights w_i generally depend either on the measured scheduling vector δ only, or on the scheduling vector $\hat{\delta}$ estimated by the observer itself or on some of its components. However, the weights of the contributing local observers are assumed to be the same as the weights used for the fuzzy model (the shared-rules principle). Note that the analysis of the augmented fuzzy system is straightforward only if the real states and the estimated ones can be assumed to

reside in the same fuzzy region. If they reside in different regions, the problem is much more difficult — the discrepancy becomes unstructured. The separation principle holds only if the scheduling variables do not depend on the estimated state. The above fact is a difficult problem and there is no clear solution yet. For the sake of simplicity, it is assumed that $\hat{\delta}(t) = \delta(t)$ for $\forall t$. In other words, the state-estimation is required to converge fast enough such that x can be replaced by $\hat{x}$ in the control loop. This fast convergence can be achieved by a suitable choice of the state-injection matrix $L(\cdot)$, which is responsible, similarly to the controller design, not only for a convergence, but rather for a convergence with some minimal decay rate. This decay rate should be slightly faster than the desired performance of the control loop. Bearing in mind the previous assumptions, the stability analysis of the augmented fuzzy system containing the fuzzy observer (eq.2.19) and an estimated-state based extended fuzzy scheduler (eq.2.22) becomes straightforward

$$\begin{aligned} u &= -\frac{\sum_{i=1}^{r} w_i(\delta(t))F_i\tilde{x}(t)}{\sum_{i=1}^{r} w_i(\delta(t))} \\ &= -\sum_{i=1}^{r} h_i(\delta(t))F_i\tilde{x} \end{aligned} \tag{2.22}$$

Combining the fuzzy controller (eq.2.22) and the fuzzy observer (eq.2.20)(eq.2.21), we obtain the following system representations:

$$\begin{aligned} \dot{x}(t) = &\sum_{i=1}^{r}\sum_{j=1}^{r} h_i(\delta(t))h_j(\delta(t)) \\ &\times[(A_i + B_iF_i)x(t) + B_iF_je(t)] \end{aligned} \tag{2.23}$$

$$\dot{e}(t) = \sum_{i=1}^{r}\sum_{j=1}^{r} h_i(\delta(t))h_j(\delta(t))[A_i + L_iC_j]e(t) \tag{2.24}$$

By combining these equations into one, we get

$$\begin{aligned} \dot{x}_a(t) &= \sum_{i=1}^{r}\sum_{j=1}^{r} h_i(\delta(t))h_j(\delta(t))G_{ij}x_a(t) \\ &= \sum_{j=1}^{r} h_i(\delta(t))h_j(\delta(t)))G_{ij}x_a(t) \\ &+2\sum_{i=1}^{r}\sum_{j>i}^{r} h_i(\delta(t))h_j(\delta(t))\frac{G_{ij}+Gji}{2}x_\alpha(t) \end{aligned} \tag{2.25}$$

with

$$x_a(t) = \begin{bmatrix} x(t) \\ e(t) \end{bmatrix} \tag{2.26}$$

and

$$G_{ij} = \left[\begin{array}{c|c} A_i + B_i F_j & B_i F_j \\ \hline 0 & A_i + L_i C_j \end{array}\right] \tag{2.27}$$

Note the form of the G_{ij} matrix in (eq.2.27) showing that under the considered assumptions the separation property holds. In other words, the controller given by F_i and the observer given by L_i can be designed separately. The stability theorems for the augmented system and for the convergence of the observer can be derived by means of the Lyapunov direct method and a quadratic function that can be solved by an LMI tool in a way similar to the FS fuzzy controller.

Theorem 2 *The equilibrium of the continuous-time fuzzy observer system described by (eq.2.20) is asymptotically stabilizable, if there are a common positive definite matrix Y and a set of matrices W_i for $i = 1, 2, .., r$ such that*

$$\mathcal{L}_{Y,W_i}(A_i, C) < 0, \tag{2.28}$$

$$\mathcal{L}_{Y,\gamma} > 0, \tag{2.29}$$

The linear operators $\mathcal{L}$ are defined for any matrix variables $Y \in \mathbb{R}^{n\times n}$ and $W_i \in \mathbb{R}^{m\times n}$ as shown in eq.(2.40),(2.41),(2.42). The desired fuzzy observer gain matrices L_i are then given by $L_i = (W_i Y^{-1})^T$, $i = 1, 2, ..., r$.

It is easy to find $Y > 0$ and the corresponding W_i or to determine that no such Y, W_i exist.

2.3.2 Performance

In the synthesis of controllers and observers, in addition to the stability requirements some performance of the closed-loop system is to be considered. The synthesis based on a quadratic Lyapunov function enables representing certain performance specifications, such as decay rates or constraints on the control input, in the form of LMIs. The performance specifications are introduced via exponential stability of the control system. Another useful requirement such as suppressing overshoots (damping) can be derived via so-called LMI regions. LMI regions,

although based on the definition of eigenvalues defined for LTI systems, can also find some practical use for fuzzy systems. Similarities have been found between an LMI region and a performance criterion based on the exponential stability combined with a quadratic Lyapunov function. Such a multiobjective approach has proven to be useful in practice when coping with some implementation constraints and desired performance specifications for the closed-loop dynamics. In this respect, this approach is superior to other known synthesis techniques where the desired control performance is achieved by a trial and error tuning which not only involves a great deal of time, but eventually neither the stability nor the performance of the entire closed-loop fuzzy system are guaranteed.

Exponential Stability—Decay Rates:

Proposition 1: The condition that

$$\dot{V}(x(t)) \leq 2\alpha V(x(t)) \tag{2.30}$$

for all trajectories of $x(t)$ of a continuous-time closed-loop TS fuzzy system is equivalent to

$$A^T P + PA + 2\alpha P \leq 0 \tag{2.31}$$

Therefore, the largest lower bound on the decay rate that we can find using a quadratic Lyapunov function can be found by solving the following generalized eigenvalue problem (GEVP) in P and α:

$$\begin{aligned} &maximize \quad \alpha \\ &subject\ to \quad P > 0 \end{aligned} \tag{2.32}$$

Decay rates in the synthesis of T-S fuzzy controllers: The approach based on the decay rates of the exponential stability and LMI techniques can be used for the synthesis of TS fuzzy controllers with prespecified closed-loop damping. As in the case of simple stability analysis shown in Section 2.2, the conditions that guarantee the desired decay rate α must be based on linear operators with respect to all their variables. Then, the GEVP can be solved by existing LMI solvers with respect to the minimization of α subject to those LMIs.

Theorem 3 *The equilibrium of the continuous-time fuzzy control systems described by (eq.2.14) is asymptotically stabilizable with closed-loop damping α, performance index γ, if there exist common positive–definite matrices X, Y and a set of matrices W_i for $i = 1, 2, ...r$ such that*

$$\mathcal{L}_{X,W_i,\alpha}(A_i, B) < 0 \tag{2.33}$$

$$\mathcal{L}_{X,Y,\gamma} < 0 \tag{2.34}$$

$$\mathcal{L}_{X,Y,W_i} < 0 \tag{2.35}$$

The linear operators $\mathcal{L}$ are defined for any matrix variables $X \in S^n$, $Y \in S^r$ and $W \in \mathbb{R}^{r\times n}$ and the scalar variable α as shown in (eq.2.55). The desired fuzzy state-feedback gain matrices K_i are then given by $K_i = W_i X^{-1}$, $i = 1, 2, ..., r$. The common matrix P can be obtained as $P = X^{-1}$.

Bibliography: [1,2,5]

2.4 LMI Selection

There are numerous LMI approaches for solving diverse problems, and finding controller and observer gains. In our case, the Linear Quadratic Controller problem and the Linear Quadratic Regulation via H_2 Control are solved with specific LMIs.

Thereinafter the two methods, and the used LMIs, are described analytically.

2.4.1 LQC Problem

Given a linear system, defined by

$$\dot{x}(t) = Ax(t) + Bu(t) \qquad A \in \mathbb{R}^{n\times n} \ \ and \ \ B \in \mathbb{R}^{n\times p}$$

with $x(t_0) = x_0$ specified, find a state feedback $u(t) = -Lx(t)$ which minimizes the following quadratic criterion:

$$J = \int_0^\infty \left[x^T(t)Qx(t) + u^T Ru(t)\right] dt$$

where $Q = H^T H \geq 0$ and $R = R^T > 0$, with $H \in \mathbb{R}^{q\times n}$, $q = rank(Q)$, and $R \in \mathbb{R}^{p\times p}$. In the following, we will be looking for a state feedback $u(t) = -Lx(t)$ which guarantees that the criterion J is inferior to some given number γ. Since $u^T Ru = x^T L^T RLx$, let us introduce the function $V(x) = x^T Px$, with $P = P^T > 0$, satisfying the two following conditions:

$$\begin{cases} V(x_0) < 0 \\ \dot{V}(x) + x^T Q x + x^T L^T R L x \end{cases} \tag{2.36}$$

Such a function $V(x)$ is a Lyapunov function, since it satisfies all three conditions:

1. $V(0) = 0$,
2. $V(x) > 0, \forall x \neq 0$,
3. $V(x)$ is decreasing, for any $x \neq 0$

Furthermore,

$$\int_0^\infty \dot{V}(x) + \int_0^\infty (x^T Q x + x^T L^T R L x) dt < 0 \tag{2.37}$$

which we can rewrite as

$$\underbrace{\int_0^\infty (x^T Q x + x^T L^T R L x) dt}_{\mathrm{J}} < V(x_0) = x_0^T P x_0 < \gamma \tag{2.38}$$

If there exist a matrix L and a function $V(x)$ satisfying (eq.2.36), then L solves this problem. By recalling that $\dot{V}(x) = \dot{x}^T P x + x T P \dot{x}$ and taking into account the closed-loop state equation, these two inequalities are equivalent to

$$\begin{cases} x_0^T P x_0 < \gamma \\ (A - BL)^T P + P(A - BL) + Q + L^T R L < 0 \end{cases} \tag{2.39}$$

By left and right multiplying the second inequality by $Y = P^{-1}$ and introducing $W = LY$, we obtain successively the following inequalities:

$$\begin{gathered} Y(A - BL)T + (A - BL)Y + YQY + YL^T RLY < 0, \\ YA^T + AY - BW - W^T B^T + YH^T HY + W^T RW < 0, \\ YA^T + AY - BW - W^T B^T + \begin{pmatrix} YH^T & W^T \end{pmatrix} + \begin{pmatrix} I_n & 0 \\ 0 & R \end{pmatrix} + \begin{pmatrix} HY \\ W \end{pmatrix} < 0 \end{gathered}$$

By applying the Schur Lemma, this inequality becomes the following LMI:

$$\begin{bmatrix} YA_i^T + A_iY - BW_i - W_i^TB^T & YH^T & W_i^T \\ HY & -I_n & 0 \\ W_i & 0 & -R^{-1} \end{bmatrix} < 0 \tag{2.40}$$

with $i = 1, 2, ..., r$, where r is the number of fuzzy rules.

The first of the inequalities (eq.2.39) becomes, successively:

$$\begin{gathered} \gamma - x_0^T P x_0 > 0 \\ \gamma - x_0^T Y^{-1} x_0 > 0 \\ \begin{pmatrix} \gamma & x_0^T \\ x_0 & Y \end{pmatrix} > 0 \end{gathered} \tag{2.41}$$

with the use of the Schur Lemma again. Note that the initial constraint $P > 0$, i.e. $Y > 0$, is contained in (2.41). A state feedback matrix L, solution of the problem, is thus obtained by solving the LMIs (eq.2.40) and (eq.2.41), and letting $L = WY^{-1}$.

In order to get rid of the knowledge of x_0 , the condition (eq.2.41) can be replaced by the condition $P - \gamma I_n < 0$. This guarantees that, for any x_0, $J < x_0^T P x_0 < \gamma x_0^T x_0$. The previous condition becomes then $\gamma I_n - Y^{-1} > 0$, i.e.,

$$\begin{pmatrix} \gamma I_n & I_n \\ I_n & Y \end{pmatrix} > 0 \tag{2.42}$$

If it is desired to minimize the value of the performance index γ, the following optimization problem, which is a typical eigenvalue problem, should be solved:

$$\begin{aligned} &\min_{\gamma, Y=Y^T, W} \quad \gamma \\ &\text{subject to} \quad (2.40), (2.42) \end{aligned} \tag{2.43}$$

Remark 3 *In the case of the observer, the LMIs are shown in eq.(2.40),(2.41),(2.42) are changed, and the matrix B is substituted by C^T.*

2.4.2 LQ Regulation via H_2 Control

Consider the constant linear multivariable system

$$\dot{x} = Ax + Bu, \qquad x(0) = x_0 \tag{2.44}$$

where $x \in \mathbb{R}^n$ and $u \in \mathbb{R}^r$ are the state vector and input vector, respectively A and B are the system coefficient matrices of appropriate dimensions.

The well-known linear quadratic optimal regulation problem for the aforementioned system is stated as follows

$$J(x,u) = \int_0^\infty \left(x^T Q x + u^T R u`\right) dt \tag{2.45}$$

is minimized, where

$$Q = Q^T \geq 0, \qquad R = R^T > 0 \tag{2.46}$$

Normally, the Q and R matrices are chosen as diagonal matrices such that the quadratic performance index is a weighted integral of squared error. The sizes of Q and R matrices depend on the number of state variables and input variables, respectively.

For a traditional solution to this problem we have a basic result which is based on the following two typical assumptions:

A1. (A, B) is stabilizable.
A2. (A, L) is observable, with $L = Q^{1/2}$.

Theorem 4 *Let assumptions A1 and A2 hold, then the following algebraic Riccati equation*

$$A^T P + PA - PBR^{-1}B^T P + Q = 0$$

has a unique symmetric positive definite solution P, and in this case the optimal solution to the aforementioned LQR problem is given, in terms of this positive definite matrix P, as follows:

$$u(t) = -R^{-1}B^T P x(t) \tag{2.47}$$

The corresponding closed-loop system is given by

$$\dot{x}(t) = (A - BR^{-1}B^T P)x(t) \tag{2.48}$$

and the minimum value of the performance index is

$$\gamma = \min_{u} J(x,u) = x_0^T P x_0 \tag{2.49}$$

Relation to H_2 Performance

Introduce the following auxiliary system

$$\begin{cases} \dot{x} = Ax + Bu + x_0\omega \\ y = Cx + Du \end{cases} \tag{2.50}$$

where

$$C = \begin{bmatrix} Q^{\frac{1}{2}} \\ 0 \end{bmatrix}, \quad D = \begin{bmatrix} 0 \\ R^{\frac{1}{2}} \end{bmatrix} \tag{2.51}$$

and ω represents an impulse disturbance. With the state feedback controller $u = Kx$ applied to the auxiliary system (eq.2.50), the closed-loop system is obtained as

$$\begin{cases} \dot{x} = (A + BK)x + x_0\omega \\ y = (C + DK)x \end{cases} \tag{2.52}$$

Thus, the transfer function of the aforementioned system (eq.2.52) from the disturbance ω to the output y is

$$G_{y\omega}(s) = (C + DK) - [sI - (A + BK)]^{-1}x_0 \tag{2.53}$$

The following theorem tells us that the LQR performance for system (eq.2.44) can be reformulated into an H_2 performance for the auxiliary system (eq.2.50) and (eq.2.51).

Theorem 5 *Given the linear system (eq.2.44) and the quadratic performance index (eq.2.45) satisfying (eq.2.46) and assumptions A1 and A2, then*

$$J(x,u) = \left\| G_{y\omega}(s) \right\|_2^2 \tag{2.54}$$

where $G_{y\omega}(s)$ is the transfer function given by (eq.2.53).

Let assumptions A1 and A2 hold, then a state feedback control in the form of $u = Kx$ exists such that $J(x,u) < \gamma$ if and only if there exist $X \in S^n$, $Y \in S^r$ and $W \in \mathbb{R}^{r\times n}$ satisfying

$$\begin{gathered}(AX+BW)+(AX+BW)^T+2\alpha X<0\\ trace(Q^{\frac{1}{2}}X(Q^{\frac{1}{2}})^T)+trace(Y)<\gamma\\ \begin{bmatrix}-Y & R^{\frac{1}{2}}W\\ (R^{\frac{1}{2}}W)^T & -X\end{bmatrix}<0,\end{gathered} \tag{2.55}$$

where α is the closed-loop damping parameter.

In this case, a feedback gain is given by

$$K=WX^{-1} \tag{2.56}$$

It clearly follows from the earlier problem can be solved via the following optimization problem:

$$\begin{aligned}\min \quad & \gamma\\ \text{subject to} \quad & (2.55)\end{aligned} \tag{2.57}$$

the produced optimal γ is then the minimum value of the performance index.

Bibliography: [2,3]

2.5 Fault Tolerant Control

A permanent increase in the complexity, efficiency, and reliability of modern controlled systems necessitates a continuous development in Fault Diagnosis (FD) theory and practice. A moderate combination of these two paradigms is intensively studied under the name Fault-Tolerant Control (FTC). FTC is one of the most important research directions underlying contemporary automatic control. It can also be perceived as an optimized integration of advanced fault diagnosis and control techniques. It has been investigated the occurrence and impact of faults for a long time, due to their potential to cause substantial damage to machinery and risk for human health or life. Early detection and maintenance of faults can help avoid system shutdown, breakdowns and even catastrophes involving human fatalities and material damage. A rough scheme of the modern control system that is able to tackle such a challenging problem is presented in Fig.2.4. As can be observed, the controlled system is the main part of the scheme, and it is composed of actuators, process dynamics and sensors.

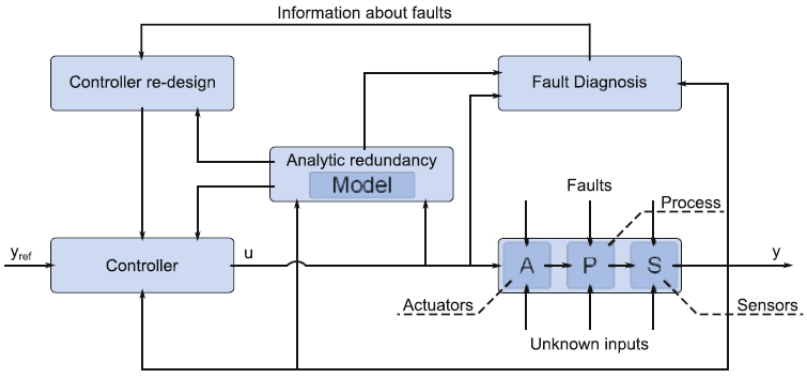

Figure 2.4: Fault Tolerant Control System.

Each of these parts is affected by the so-called unknown inputs, which can be perceived as process and measurement noise as well as external disturbances acting on the system. When model-based control and analytical redundancy-based fault diagnosis are utilised, then the unknown input can also be extended by model uncertainty, i.e., the mismatch between the model and the system being considered. The system may also be affected by faults. A fault can generally be defined as an unpermitted deviation of at least one characteristic property or parameter of the system from the normal condition, e.g. a sensor malfunction. All the unexpected variations that tend to degrade the overall performance of a system can also be interpreted as faults. Contrary to the term failure, which suggests a complete breakdown of the system, the term fault is used to denote a malfunction rather than a catastrophe. Indeed, failure can be defined as a permanent interruption in the system ability to perform a required function under specified operating conditions. Since a system can be split into three parts (fig. 2.4), i.e., actuators, the process, and sensors, such a decomposition leads directly to three classes of faults:

- *Actuator faults*, which can be viewed as any malfunction of the equipment that actuates the system.
- *Process faults* (or component faults), which occur when some changes in the system make the dynamic relation invalid.
- *Sensor faults*, which can be viewed as serious measurements variations.

The role of the fault diagnosis part is to monitor the behaviour of the system and to provide

all possible information regarding the abnormal functioning of its components. As a result, the overall task of fault diagnosis consists of three subtasks:

- *Fault detection*: to make a decision regarding the system stage–either that something is wrong or that everything works under the normal conditions;
- *Fault isolation*: to determine the location of the fault, e.g., which sensor or actuator is faulty;
- *Fault identification*: to determine the size and type or nature of the fault.

However, from the practical viewpoint, to pursue a complete fault diagnosis the following three steps have to be realized:

- *Residual generation*: generation of the signals that reflect the fault. Typically, the residual is defined as a difference between the outputs of the system and its estimate obtained with the mathematical model;
- *Residual evaluation*: logical decision making on the time of occurrence and the location of faults;
- *Fault identification*: determination of the type of fault, its size and cause.

The knowledge resulting from these steps is then provided to the controller re-design part, which is responsible for changing the control law in such a way as to maintain the required system performance.

2.5.1 Unknown Input Observer

One of the re-design strategies is the so-called Unknown Input Observer (UIO) for both deterministic and stochastic non-linear systems. As has already been mentioned, the unknown input may represent noise, disturbances as well as model uncertainty. This means that one way to achieve robustness is to decouple the unknown input from the residual. This can be realized with the UIO or filter.

As can be observed in the literature, observers (or filters in a stochastic framework) are commonly used in both control and fault diagnosis schemes of non-linear systems. Undoubtedly, the most common approach is to use robust observers, such as the UIO, which can tolerate a degree of model uncertainty and hence increase the reliability of fault diagnosis. One of the standard methods of observer design consists in using a non-linear change of coordinates to

turn the original system into a linear one (or a pseudo linear one). Such approaches can be applied for FD and FTC.

Unknown Input Decoupling

Let us consider a non-linear stochastic system given by the following equations

$$\dot{x} = g(x) + h(u) + Ed + Lf + w \tag{2.58}$$

$$\dot{y} = C\dot{x} + \dot{v} \tag{2.59}$$

Note that the unknown input and fault distribution matrices, denoted by E and L, are assumed (for the sake of simplicity) constant in this section. It should be mentioned that this section focuses on faults that can influence the state equation (2.58), such as actuator faults. The main problem is to design a filter which is insensitive to the influence of the unknown input (external disturbances and modeling errors) while being sensitive to faults. The necessary condition for the existence of a solution to the unknown input decoupling problem is as follows:

$$rank(CE) = rank(E) = q \tag{2.60}$$

If the condition (2.60) is satisfied, then it is possible to calculate

$$H = (CE)^{+} = [(CE)^{T}CE]^{-1}(CE)^{T} \tag{2.61}$$

Thus, by inserting (eq.2.58) into (eq.2.59) and then multiplying (eq.2.59) by H it is straightforward to show that

$$d = H\left[\dot{y} - C[g(x) + h(u) + Lf + w] - \dot{v}\right] \tag{2.62}$$

Substituting (eq.2.62) into (eq.2.58) for d_k gives

$$\dot{x} = \bar{g}(x) + \bar{h}(u) + \bar{E}\dot{y} + \bar{L}f + \bar{w} \tag{2.63}$$

where

$$\bar{g}(\cdot) = Gg(\cdot), \quad \bar{h}(\cdot) = Gh(\cdot)$$
$$\bar{E} = EH, \quad \bar{w} = Gw - EH\dot{v}$$

and

$$G = I - EHC$$

Consequently, the general observer structure is

$$\dot{\hat{x}} = \bar{g}(\hat{x}) + \bar{h}(\cdot) + \bar{E}\dot{y} + K(\cdot) \tag{2.64}$$

where $K(\cdot)$ is the state correction term. In order to make further deliberations more general, no particular form of $K(\cdot)$ is assumed. Let us define a residual as a difference between the output of the system and its estimate:

$$\begin{aligned} \dot{z} &= \dot{y} - C\dot{\hat{x}} \\ &= C(\bar{g}(x) - \bar{g}(\hat{x}) - K(\cdot)) + \bar{f} + C\bar{w} + \dot{v} \end{aligned} \tag{2.65}$$

where the faults/disturbances can be obtained as

$$\bar{f} = C\bar{L}f = C\Big[I_n - E\big[(CE)^T CE\big]^{-1}(CE)^T\Big]Lf \tag{2.66}$$

Bibliography: [14, 15]

2.5.2 Augmented State Observer

The Augmented State Observer (ASO) is a method which augments the system states with the faults/disturbances considering the latter ones as states, and it is used to estimate them as being a regular state observer. Let us consider a linear form state-space system given by the following equations:

$$\dot{x}_{aug} = A_{aug}x_{aug} + B_{aug}u \tag{2.67}$$

$$y_{aug} = C_{aug}x_{aug} \tag{2.68}$$

where

$$A_{aug} = \begin{bmatrix} A & F_{actuator} & O \\ O & O & O \end{bmatrix}, \quad B_{aug} = \begin{bmatrix} B \\ O \end{bmatrix}$$

$$C_{aug} = \begin{bmatrix} C & O & F_{sensor} \end{bmatrix}, \quad x_{aug} = \begin{bmatrix} x \\ f_{actuator} \\ f_{sensor} \end{bmatrix}$$

F is the fault distribution matrices and f the faults.

Consequently, the states and faults/disturbances can be estimated via an augmented state observer which has the following form:

$$\begin{cases} \dot{\hat{x}}_{aug} = A_{aug}x_{aug} + B_{aug}u - L[y_{aug} - \hat{y}_{aug}] \\ \hat{y}_{aug} = C_{aug}\hat{x}_{aug} \end{cases} \tag{2.69}$$

L is the state injection matrix.

Bibliography: [14, 16]

2.5.3 Least Squares Parameter Estimation

Under the condition that the faults affect the system parameters, the fault estimation can be formulated as a parameter estimation problem in such a way that any parameter estimation algorithms (such as least squares, generalized/extended least squares, instrumental variables, maximum likelihood, extended Kalman filter and others) could be used. In general, least-square algorithms can be formulated either in block or recursive online forms. Once the equation is put in regression form, the recursive formulation and the block formulation are interchangeable.

In the following, the fault estimation procedure is explained for the case of actuator faults (affecting matrix B) for clarity of the presentation. Similar approach can be applied for estimating faults affecting the plant dynamics (matrix A) and sensor dynamics (matrix C). Let us consider that the model for fault estimation of the system in state-space form including the actuator faults:

$$\dot{x}_f = Ax_f + B_f(\phi)u_f + \Phi f_u \tag{2.70}$$

$$y = Cx_f \tag{2.71}$$

where $x_f \in \mathbb{R}^{n_x}$ represents the state vector, $u_f \in \mathbb{R}^{n_u}$ denotes the control inputs. $f_u \in \mathbb{R}^{n_u}$ denotes the additive actuator faults and $\Phi \in \mathbb{R}^{n_x \times n_u}$ represents the actuator fault distribution

matrix. The multiplicative actuator faults are embedded in the input matrix $B_f(\phi)$ as follows:

$$B_f(\phi) = Bdiag(\phi_1, \phi_2, ..., \phi_{n_u}, \quad 0 \leq \phi_i \leq 1) \tag{2.72}$$

where B denotes the faultless input matrix. ϕ_i represents the effectiveness of the i_{th} actuator, such that the extreme values $\phi_i = 0$ and $\phi_i = 1$ represent a total failure of the i_{th} actuator and the healthy i_{th} actuator, respectively.

Typically, the assumption that $\Phi = B_f$ is performed. Cases different from this could be handled by adding more complexity to the mathematical formulation.

Actuator fault estimation: The procedure starts from the hypothesis that it is possible to find a state of the system that is directly influenced by the faulty actuator. For example, consider that the i_{th} actuator influences the j_{th} state of the system and that the model for the j_{th} state is given by the discrete-time equation:

$$\dot{x}_j = \sum_{i=1}^{n_x} a_{jl} x_l + b_{ji} u_i \tag{2.73}$$

Equation (2.73) can be written including the multiplicative and additive faults

$$\dot{x}_j = \sum_{i=1}^{n_x} a_{jl} x_l + \phi_i b_{ji} (u_i + f_{ui}) \tag{2.74}$$

where ϕ_i is the multiplicative fault and f_{ui} is the additive fault. This equation can be brought to the regression form $z = \theta v$ by considering the following:

$$\begin{array}{c} z = \sum_{i=1}^{n_x} a_{jl} x_l \\ \theta = \left[\begin{array}{cc} \phi_i & \nu_i \end{array} \right] \\ v = \left[\begin{array}{cc} b_{ji} u_i & b_{ji} \end{array} \right] \\ \nu_i = \phi_i f_{ui} \end{array} \tag{2.75}$$

and solving the least squares problem for θ:

$$\hat{\theta} = \left[\begin{array}{c} v(N)^T \cdot v(N) \end{array} \right]^{-1} v(N)^T z(N) \tag{2.76}$$

the additive fault estimation can be obtained as follows:

$$\hat{f_{ui}} = \frac{\hat{\nu_i}}{\hat{\phi_i}} \tag{2.77}$$

Note that N is the size of the moving horizon window of data.

Bibliography: [17, 18]

Chapter 3

Kinematic and Dynamic Vehicle Models

The movement of a vehicle can be described by using equations which represent the kinematic and dynamic behaviors. In contrast with common mobile robots, urban autonomous vehicles are systems with larger mass and higher operational velocity. Consequently, the use of dynamic models is necessary. In dynamic models, the sum of forces over the vehicle are taken into account so as to compute the vehicle acceleration. The motion is generated by applying forces over the wheels and mass, as well as inertia and tire parameters are taken into consideration. The kinematic models are based on the velocity vector movement, aiming to compute longitudinal and lateral velocities referenced to a global inertial frame. External forces are not considered in this case. For both the kinematic and dynamic models, the bicycle model has been employed (Fig.3.1). At this point, it is needed to specify that the roll, pitch and z motion are not considered, and only yaw, x and y movements are taken into account. In this thesis, at first the two models are built in a separated way. It means that, both model behaviors will be controlled in a decoupled way by using two control loops. Subsequently, the two models will be combined in order to track the desired behavior of the vehicle in a suitable way. In Table 3.1, the characteristic vehicle parameters which are used in the models are depicted.

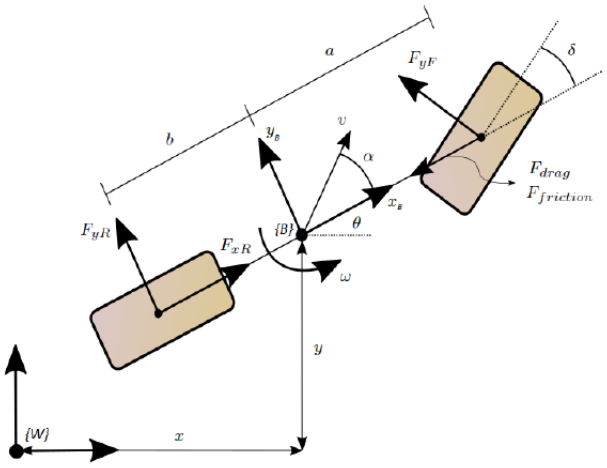

Figure 3.1: Bicycle model for vehicle dynamics.
{W} global inertial frame {B} body frame located in the centre of gravity (CoG) of the vehicle.

Parameter	Description	Value
a	Distance from CoG to front axle	0.758 m
b	Distance from CoG to rear axle	1.036 m
M	Vehicle mass	683 kg
g	Gravity of Earth	9.80665 $\frac{m}{s^2}$
I	Vehicle yaw inertia	560.94 kg m^2
C_d	Drag coefficient	0.36
A_r	Vehicle frontal area	4 m^2
ρ	Air density at 25°C	1.2 $\frac{kg}{m^3}$
μ	Nominal friction coefficient	0.5
C_x	Tire stiffness coefficient	25000 $\frac{N}{rad}$
F_{xF}	Longitudinal front force	0

Table 3.1: Kinematic and Dynamic Model Parameters

3.1 Kinematic non-liner model

The kinematic model of the vehicle is widely used due to its low parameter dependency. It is assumed to have zero skidding and is considered that lateral force is so small that it can be neglected. Basically, it is a geometric mode to compute vehicle position and orientation using the linear and angular velocities. The kinematic equations are introduced below:

$$\begin{cases} \dot{x} = v\cos(\theta) \\ \dot{y} = v\sin(\theta) \\ \dot{\theta} = \omega \end{cases} \tag{3.1}$$

where:

- x, y and θ represent the current position and orientation of the vehicle in meters (m) and radians (rad), respectively, with respect to the inertial frame (W).
- v is the linear velocity in $\frac{m}{s}$.
- ω is the angular velocity in $\frac{rad}{s}$.

3.2 Dynamic non-liner model

The dynamical behavior of a vehicle is generally complicated to represent it in a detailed manner. In real applications, usually simplified models are used and in our case as well. The dynamic model we use is based on the second Newton's law. Hence, the dynamic model of the vehicle can be written as:

$$\dot{u} = \frac{F_{xR}\cos(\alpha) + F_{yF}\sin(\alpha - \delta) + F_{yR}\sin(\alpha) - F_{drag} - F_{friction}}{m} \tag{3.2a}$$

$$\dot{\alpha} = \frac{-F_{xF}\sin(\alpha - \delta)) - F_{xR}\sin(\alpha) + F_{yF}\cos(\alpha - \delta) + F_{yR}\cos(\alpha) - mv\omega}{mv} \tag{3.2b}$$

$$\dot{\omega} = \frac{F_{xF}\alpha\sin(\delta) + F_{yF}\alpha\cos(\delta) + F_{yR}b}{I} \tag{3.2c}$$

$$F_{drag} = \frac{1}{2}C_d A_r v^2 \tag{3.3}$$

$$F_{friction} = \mu M g \tag{3.4}$$

$$F_{yF} = C_x(\delta - \alpha - \frac{\alpha\omega}{v}) \tag{3.5}$$

$$F_{yR} = C_x(-\alpha - \frac{b\omega}{v}) \tag{3.6}$$

where:

- v represents the vehicle linear velocity (m/s).
- ω represents the vehicle angular velocity (rad/sec).
- α represents the vehicle slip angle (rad).
- δ is the steering angle and one of the inputs of the system (rad).
- F_{xR} is the longitudinal rear force and the other input of the system (N).
- F_{yR} is the lateral rear force (N).
- F_{yF} is the lateral frontal force (N).
- F_{drag} represents drag force that opposes to the forward movement (N).
- F_{friction} is the friction force that also opposes to the longitudinal vehicle movement (N).

It is needed to note that the polar representation has been adopted, by considering the variables v and α, instead of using the states x and y. Note that the dynamic model variables refer to the vehicle body frame B while the kinematic set of variables refer to the global fixed coordinate system W in order to represent the trajectory from a relative point of view (fig.3.1).

Bibliography: [6]

Chapter 4

Takagi-Sugeno Modelling

The Takagi-Sugeno control technique in order to control the non-linear model employs a linear-like representation. This gain-scheduling method consists of incorporating the model non-linearities in the parameters of the model which depend on some scheduling variables, which have some predefined bounds. The kinematic and dynamic nonlinear models are adjusted, using the gain-scheduling method. For the kinematic state-space modelling, a model has been developed beforehand (eq.3.1), while for the dynamic, the model will be developed in this chapter. Two separated T-S models will be built in order to control the kinematic and dynamic parts of the vehicle, as well as a T-S model for a dynamic observer will be obtained.

4.1 Kinematic Gain-Scheduling Modelling

The model which is described in previous chapter (eq.3.1), depends on the measurements (x, y and θ) from the real system and the desired values (x_d, y_d and θ_d), and it is defined as the difference (errors) between them. These errors are expressed with respect to the global frame $\{W\}$. Although, in order to control the vehicle it is essential to express the errors with respect to the frame $\{B\}$, such that the lateral error is always measured in the lateral axis of the vehicle. For this reason, a rotation over the global orthogonal axis is considered to represent the errors in the body vehicle frame $\{B\}$ (Fig.3.1):

$$\begin{bmatrix} x_\mathrm{e} \\ y_\mathrm{e} \\ \theta_\mathrm{e} \end{bmatrix} = \begin{bmatrix} \cos(\theta) & \sin(\theta) & 0 \\ -\sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_\mathrm{d} - x \\ y_\mathrm{d} - y \\ \theta_\mathrm{d} - \theta \end{bmatrix} \tag{4.1}$$

where x_d, y_d, θ_d are the desired values of the positions and orientation, and x_e, y_e, θ_e are the errors with respect to the real values. In order to develop the model of the error we have to take into account the rear wheels non-holonomic constraint:

$$\dot{x}\sin(\theta) = \dot{y}\cos(\theta) \tag{4.2}$$

Therefore, by derivating (eq.4.1) and using (eq.3.1), (eq.4.2) and some trigonometric identities, the following open-loop error system is obtained:

$$\dot{x}_\mathrm{e} = \omega y_\mathrm{e} + v_\mathrm{d}\cos(\theta_\mathrm{e}) - v \tag{4.3a}$$

$$\dot{y}_\mathrm{e} = -\omega x_\mathrm{e} + v_\mathrm{d}\sin(\theta_\mathrm{e}) \tag{4.3b}$$

$$\dot{\theta}_\mathrm{e} = \omega_\mathrm{d} - \omega \tag{4.3c}$$

The state, input and output vectors are the following:

$$x = \begin{bmatrix} x_\mathrm{e} \\ y_\mathrm{e} \\ \theta_\mathrm{e} \end{bmatrix}, u = \begin{bmatrix} v \\ \omega \end{bmatrix}, y = \begin{bmatrix} x_\mathrm{e} \\ y_\mathrm{e} \\ \theta_\mathrm{e} \end{bmatrix} \tag{4.4}$$

Bearing in mind that $\omega, v_\mathrm{d}, \theta_\mathrm{e} \in \mathbb{R}$ are the scheduling variables, the gain-scheduling state-space representation for the kinematic model (eq.4.3) is the following:

$$\begin{cases} \dot{x} = A(\omega, v_\mathrm{d}, \theta_e)x + Bu_K + r \\ y = Cx \end{cases} \tag{4.5}$$

where r is the reference vector and matrices A, B, C have the form:

$$A(\omega, v_\mathrm{d}) = \begin{bmatrix} 0 & \omega & 0 \\ -\omega & 0 & v_\mathrm{d}\frac{\sin(\theta_\mathrm{e})}{\theta_\mathrm{e}} \\ 0 & 0 & 0 \end{bmatrix} \tag{4.6a}$$

$$B = \begin{bmatrix} -1 & 0 \\ 0 & 0 \\ 0 & -1 \end{bmatrix} \tag{4.6b}$$

$$C = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{4.6c}$$

$$r = \begin{bmatrix} v_{\mathrm{d}} \cos(\theta_{\mathrm{e}}) \\ \omega_{\mathrm{d}} \end{bmatrix} \tag{4.6d}$$

Remark 4.1: The vector r for the reference will be added directly to the control law. Hence, the resulting global kinematic controller has the following form:

$$u = -Fx + r \tag{4.7}$$

where F is the controller gain.

Trajectory Planner

The reference r is provided by a trajectory planner which is in charge on generating a feasible trajectory by means of using a polynomial curve generation method [10]. The trajectory planner is an offline mode, that arranges the linear and angular velocity references v_d, ω_d and calculates the desired positions and orientations x_d, y_d, θ_d. This algorithm uses the following equations:

$$x_d(t) = v_d(t)\cos(\theta_d(t))T_s + x_d(t-1) \tag{4.8a}$$

$$y_d(t) = v_d(t)\cos(\theta_d(t))T_s + y_d(t-1) \tag{4.8b}$$

$$\theta_d(t) = \theta_d(t-1) + \omega_d(t)T_s \tag{4.8c}$$

where T_s is the control sample time.

4.2 Dynamic Gain-Scheduling Modelling

4.2.1 Controller

As it is presented in previous chapter the dynamic non-linear model is rather complicated comparing to the kinematic one. For this reason its construction in a gain-scheduling form will be presented progressively. Denoting the state, input, and output vectors, respectively as

$$x = \begin{bmatrix} v \\ \alpha \\ \omega \end{bmatrix}, u = \begin{bmatrix} F_{\mathrm{xR}} \\ \delta \end{bmatrix}, y = \begin{bmatrix} v \\ \omega \end{bmatrix} \tag{4.9}$$

the dynamic model has the following state space representation:

$$\begin{cases} \dot{x} = Ax + Bu \\ y = Cx \end{cases} \tag{4.10}$$

where

$$A = \begin{bmatrix} A_{11} & A_{12} & A_{13} \\ 0 & A_{22} & A_{23} \\ 0 & A_{32} & A_{33} \end{bmatrix} \tag{4.11a}$$

$$B = \begin{bmatrix} B_{11} & B_{12} \\ B_{21} & B_{22} \\ 0 & B_{32} \end{bmatrix} \tag{4.11b}$$

$$C = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{4.11c}$$

and

$$A_{11} = \frac{-(F_{drag} + F_{friction})}{Mv}, A_{12} = \frac{C_x(\sin(\delta)\cos(\alpha) - \sin(\alpha)\cos(\delta) - \sin(\alpha))}{M} \tag{4.12a}$$

$$A_{13} = \frac{C_x(a(\sin(\delta)\cos(\alpha) - \sin(\alpha)\cos(\alpha)) - b\sin(\alpha))}{Mv} \tag{4.12b}$$

$$A_{22} = \frac{-C_x(\cos(\alpha)\cos(\delta) + \sin(\alpha)\sin(\delta) + \cos(\alpha))}{Mv} \tag{4.12c}$$

$$A_{23} = \frac{-C_x a(\cos(\delta)cos(\alpha) + \sin(\alpha)\sin(\delta)) + C_f b\cos(\alpha)}{Mv^2} - 1 \tag{4.12d}$$

$$A_{32} = \frac{C_x(b - a\cos(\delta))}{I}, A_{33} = \frac{-C_x(a^2\cos(\delta) + b^2)}{Iv} \tag{4.12e}$$

$$B_{11} = \frac{\cos(\alpha)}{M}, B_{12} = \frac{C_x(\sin(\delta)\cos(\alpha) - \sin(\alpha)\cos(\delta))}{M} \tag{4.12f}$$

$$B_{21} = \frac{-\sin(\alpha)}{Mv}, B_{22} = \frac{C_x(\cos(\alpha)\cos(\delta) + \sin(\alpha)\sin(\delta))}{Mv} \tag{4.12g}$$

$$B_{32} = \frac{C_x a \cos(\delta)}{I} \tag{4.12h}$$

A and B are time varying matrices. However, in order to avoid the dependency on a varying matrix B, the system has been adjusted by adding a fast dynamic filter [7] in the form:

$$\dot{x}_f = A_f x_f + B_f u_f \tag{4.13}$$

$$\begin{bmatrix} \dot{F}_{xR} \\ \dot{\delta} \end{bmatrix} = \begin{bmatrix} -\gamma & 0 \\ 0 & -\gamma \end{bmatrix} \begin{bmatrix} F_{xR} \\ \delta \end{bmatrix} + \begin{bmatrix} \gamma & 0 \\ 0 & \gamma \end{bmatrix} \begin{bmatrix} u_F \\ u_\delta \end{bmatrix}$$

where γ represents the filter gain, u_F is the new longitudinal behaviour input and u_δ is the new lateral behaviour input. This filter characterized by fast dynamics, so it will not disturb the dynamic model.

Hence, the system (eq.4.11) becomes a fifth order system with state and input vectors in the form:

$$x = \begin{bmatrix} v \\ \alpha \\ \omega \\ F_{xR} \\ \delta \end{bmatrix}, \quad u_f = \begin{bmatrix} u_F \\ u_\delta \end{bmatrix} \tag{4.14a}$$

with A and B matrices as

$$A = \begin{bmatrix} A_{11} & A_{12} & A_{13} & B_{11} & B_{12} \\ 0 & A_{22} & A_{23} & B_{21} & B_{22} \\ 0 & A_{32} & A_{33} & 0 & B_{32} \\ 0 & 0 & 0 & -\gamma & 0 \\ 0 & 0 & 0 & 0 & -\gamma \end{bmatrix} \tag{4.14b}$$

$$B = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ \gamma & 0 \\ 0 & \gamma \end{bmatrix} \tag{4.14c}$$

Nevertheless, the model still presents some features that will difficult the control design. One of them is that the input $\delta = 0$ has been identified as a singular point. Therefore, in order to avoid it, a change of variable has been done by shifting the δ interval:

$$\delta \in \left[\underline{\delta}, \overline{\delta}\right] \rightarrow \sigma \in \left[\underline{\delta} + \epsilon, \overline{\delta} + \epsilon\right] \tag{4.15}$$

where σ is the new scheduling variable and ϵ is a constant value greater than $\overline{\delta}$.

Another issue that makes difficult the control design is the steady state error that appears in linear and angular velocities channels. In order to solve this problem, it is proposed the addition of integral actions through the controller. Then, two new states (i_ω) ans (i_v) have been added as the integral of the states ω and v:

$$i_\omega = -\omega \qquad i_v = -v \tag{4.16}$$

Hence, taking into account the previous arrangements and denoting the scheduling variables as σ, v and $\alpha \in \mathbb{R}$, the dynamic gain-scheduling model is presented as:

$$\begin{cases} \dot{x} = A(\sigma, v, \alpha)x_D + Bu_f \\ y = Cx_D \end{cases} \tag{4.17}$$

with state and input vectors:

$$x_D = \begin{bmatrix} v \\ \alpha \\ \omega \\ F_{xR} \\ \sigma \\ i_\omega \\ i_v \end{bmatrix}, \quad u_f = \begin{bmatrix} u_F \\ u_\delta \end{bmatrix} \tag{4.18a}$$

and:

$$A = \begin{bmatrix} A_{11} & A_{12} & A_{13} & B_{11} & B_{12} & 0 & 0 \\ 0 & A_{22} & A_{23} & B_{21} & B_{22} & 0 & 0 \\ 0 & A_{32} & A_{33} & 0 & B_{32} & 0 & 0 \\ 0 & 0 & 0 & -\gamma & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -\gamma & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{4.18b}$$

$$B = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ \gamma & 0 \\ 0 & \gamma \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \tag{4.18c}$$

The above model (eq.4.18) will be used for designing the dynamic state feedback controller. Hence, the chosen control scheme for the dynamic loop has the following expression

$$u_f = K_D x_D + N_{ff} r_D \tag{4.19}$$

where K_D is the dynamic controller gain, N_{ff} is a feed-forward matrix (reference scale factor) [3], x_D is the dynamic state vector, r_D represents the reference vector which corresponds to the kinematic control signal u_K (eq.4.7), and u_f is control input to the added filter (eq.4.13). At this point, the dynamic control action u_D that is applied over the vehicle will be the output of applying u_f to the filter. The feedforward matrix has been computed using the following

expression:

$$N_{ff} = [C - (BK - A)^{-1}B]^{-1} \tag{4.20}$$

where matrices A and B are the ones presented in (eq.4.14) (i.e., without considering the added integrators), K is a sub-block of K_D in which the last two columns have been omitted and matrix C is of the form

$$C = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{bmatrix} \tag{4.21}$$

4.2.2 Observer

Due to the fact that the measurable states of the dynamic model are the linear and angular velocity v, ω and there is a state that is difficult to be measured, the slip angle α, an observer is recommended to be used for estimating the dynamic states.

At this point, denoting the state, estimated state, input, output and estimated output vectors, respectively, as

$$x = \begin{bmatrix} v \\ \alpha \\ \omega \end{bmatrix}, \hat{x} = \begin{bmatrix} \hat{v} \\ \hat{\alpha} \\ \hat{\omega} \end{bmatrix}, u = \begin{bmatrix} F_{\mathrm{xR}} \\ \delta \end{bmatrix}, y = \begin{bmatrix} v \\ \omega \end{bmatrix} \hat{y} = \begin{bmatrix} \hat{v} \\ \hat{\omega} \end{bmatrix} \tag{4.22}$$

the state space model for the dynamic representation can be obtained as:

$$\begin{cases} \dot{\hat{x}}(t) = A_{obs}x(t) + B_{obs}u(t) - L[y(t) - \hat{y}(t)] \\ \hat{y}(t) = C_{obs}\hat{x}(t) \end{cases} \tag{4.23}$$

where

$$A_{obs} = \begin{bmatrix} A_{11} & A_{12} & A_{13} \\ 0 & A_{22} & A_{23} \\ 0 & A_{32} & A_{33} \end{bmatrix} \tag{4.24a}$$

$$B_{obs} = \begin{bmatrix} B_{11} & B_{12} \\ B_{21} & B_{22} \\ 0 & B_{32} \end{bmatrix} \tag{4.24b}$$

where A_{obs} and B_{obs} elements come from (eq.4.12)

$$C_{obs} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{4.24c}$$

The above model (eq.4.24) will be used for designing the dynamic state observer.

Therefore, the complete control law is obtained by substituting in (eq.4.19) the state estimation $\hat{x}$ from (eq.4.23), instead of the real state x:

$$u_f = K_D \begin{bmatrix} \hat{x} \\ F_{xR} \\ \sigma \\ i_\omega \\ i_v \end{bmatrix} + N_{ff} r_D \tag{4.25}$$

4.3 Fault Tolerance Gain-Scheduling Modelling

In the case of the actuator faults (longitudinal, lateral) appearance to the system, it is needed to detect and estimate the fault, in order to compensate it. The most common methods so as to accomplish this challenge are analyzed and the adjustment of the specific system is presented.

4.3.1 Unknown Input Observer

State Space Model

In the case of the fault/disturbance existence, some modifications to the system will be needed. In order to compensate the faults of the actuators by using the UIO method, it is needed to adjust the dynamic state space model (eq.4.17) by adding the extra term due to the fault:

$$\begin{cases} \dot{x} = A(\sigma, v, \alpha)x_D + Bu_f + EFault \\ y = Cx_D \end{cases} \tag{4.26}$$

where E is the fault distribution matrix.

In the case of longitudinal actuator fault $F_{xR_{fault}}$, the matrix E has the form:

$$E_{longitudinal} = \begin{bmatrix} -B_{11} \\ -B_{21} \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \tag{4.27}$$

In the case of lateral actuator fault δ_{fault}, the matrix E has the form:

$$E_{lateral} = \begin{bmatrix} -B_{12} \\ -B_{22} \\ -B_{32} \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \tag{4.28}$$

Fault and State Estimation

In order to compensate the fault/disturbance in the system the estimation of this is needed. The proposed UIO estimation scheme is developed for Takagi-Sugeno gain scheduling systems affected by faults. Such a procedure is based on computing the difference between the real system and the model used for observation

$$C_{obs}EFault = \dot{y} - C_{obs}(A_{obs}\hat{x}_{uio} + B_{obs}u) \tag{4.29}$$

where

$$A_{obs}, B_{obs}, C_{obs} \quad \text{come from (eq.4.24)},$$

$$\dot{y} = C_{obs}\dot{x} + \dot{v},$$

$$\hat{x}_{uio} = \begin{bmatrix} \hat{v} \\ \hat{\alpha} \\ \hat{\omega} \end{bmatrix}$$

$$E \text{ is: } \quad E_{longitudinal} = \begin{bmatrix} -B_{11} \\ -B_{21} \\ 0 \end{bmatrix} \quad \text{or} \quad E_{lateral} = \begin{bmatrix} -B_{12} \\ -B_{22} \\ -B_{32} \end{bmatrix}$$

Thus, considering $H = (CE)^{+}$, the fault can be obtained as

$$Fault = H\big(\dot{y} - C_{obs}(A_{obs}\hat{x}_{uio} + B_{obs}u)\big) \tag{4.30}$$

Consequently, the state estimation can be obtained as follows:

$$\dot{\hat{x}}_{uio} = A_{uio}\hat{x}_{uio} + B_{uio}u - EH\dot{y} \tag{4.31}$$

where

$$A_{uio} = (I - EHC_{obs})A$$
$$B_{uio} = (I - EHC_{obs})B$$

and A, B come from (eq.4.11).

Then, the state estimation will depend on the unknown input observer gain L_{uio} and presents the form

$$\dot{\hat{x}}_{uio} = (A_{uio} - L_{uio}C_{obs})\hat{x}_{uio} + B_{uio}u - EH\dot{y} + L_{uio}y \tag{4.32}$$

Therefore, the complete control law is obtained by substituting in (eq.4.19) the state estimation $\hat{x}$ from (eq.4.32), instead of the real state x_{uio}:

$$u_f = K_D \begin{bmatrix} \hat{x}_{uio} \\ F_{xR} \\ \sigma \\ i_\omega \\ i_v \end{bmatrix} + N_{ff} r_D \tag{4.33}$$

At this point, the dynamic control action u_D that is applied over the vehicle will be the output of applying u_f to the filter (eq.4.13) and adding the vector generated by the fault compensator in (eq.4.30).

$$\text{- with longitudinal actuator fault:} \quad u_D = \begin{bmatrix} F_{\text{xR}} \\ \delta \end{bmatrix} + \begin{bmatrix} F_{xRfault} \\ 0 \end{bmatrix} \tag{4.34}$$

$$\text{- with lateral actuator fault:} \quad u_D = \begin{bmatrix} F_{\text{xR}} \\ \delta \end{bmatrix} + \begin{bmatrix} 0 \\ \delta_{fault} \end{bmatrix} \tag{4.35}$$

4.3.2 Augmented State Observer

State Space Model

In order to compensate the faults of the actuators by using the ASO, it is needed to adjust the dynamic state space model (eq.4.10) by augmenting the faults as states to the system:

$$\begin{cases} \dot{x}_{aug} = A_{aug}x_{aug} + B_{aug}u \\ y_{aug} = C_{aug}x_{aug} \end{cases} \tag{4.36}$$

where, for the longitudinal actuator fault ($F_{xRfault}$) the states and inputs vectors from (eq.4.9) take the form:

$$x_{aug} = \begin{bmatrix} v \\ \alpha \\ \omega \\ F_{xRfault} \end{bmatrix}, \quad u = \begin{bmatrix} F_{\text{xR}} \\ \delta \end{bmatrix} \tag{4.37a}$$

and matrices A, B and C from (eq.4.11) take the form:

$$A_{aug} = \begin{bmatrix} A_{11} & A_{12} & A_{13} & -B_{11} \\ 0 & A_{22} & A_{23} & -B_{21} \\ 0 & A_{32} & A_{33} & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \quad B_{aug} = \begin{bmatrix} B_{11} & B_{12} \\ B_{21} & B_{22} \\ 0 & B_{32} \\ 0 & 0 \end{bmatrix}, \quad C_{aug} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \tag{4.37b}$$

Respectively, for the lateral actuator fault (δ_{fault}) we have:

the states and inputs vectors

$$x_{aug} = \begin{bmatrix} v \\ \alpha \\ \omega \\ \delta_{fault} \end{bmatrix}, \quad u = \begin{bmatrix} F_{\mathrm{xR}} \\ \delta \end{bmatrix} \tag{4.38a}$$

and matrices A, B and C

$$A_{aug} = \begin{bmatrix} A_{11} & A_{12} & A_{13} & -B_{12} \\ 0 & A_{22} & A_{23} & -B_{22} \\ 0 & A_{32} & A_{33} & -B_{32} \\ 0 & 0 & 0 & 0 \end{bmatrix}, \quad B_{aug} = \begin{bmatrix} B_{11} & B_{12} \\ B_{21} & B_{22} \\ 0 & B_{32} \\ 0 & 0 \end{bmatrix}, \quad C_{aug} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \tag{4.38b}$$

Consequently, for the equations (4.14, 4.18 and 4.21) have to be adjusted adequately.

Fault and State Estimation

In order to estimate the states of the system, and consequently the faults we have to use the representation from Section 4.2.2. Adjusting the observer parts from the above augmented state space model, the ASO takes the form:

$$\begin{cases} \dot{\hat{x}}_{aug}(t) = A_{aug}x_{aug}(t) + B_{aug}u(t) - L[y_{aug}(t) - \hat{y}_{aug}(t)] \\ \hat{y}_{aug}(t) = C_{aug}\hat{x}_{aug}(t) \end{cases} \tag{4.39}$$

Therefore, the complete control law is obtained by substituting in (eq.4.19) the state estimation $\hat{x}_{aug}$ from (eq.4.39), instead of the real state x:

$$u_f = K_D \begin{bmatrix} \hat{x}_{aug} \\ F_{xR} \\ \sigma \\ i_\omega \\ i_v \end{bmatrix} + N_{ff} r_D \tag{4.40}$$

At this point, the dynamic control action u_D that is applied over the vehicle will be the output of applying u_f to the filter (eq.4.13).

4.3.3 Least Squares Parameter Estimation

Fault Estimation

As the faults affect the system parameters, the fault estimation can be formulated as a parameter estimation problem. The least-square algorithm can be formulated in a recursive online form.

In the following, the fault estimation procedure is explained for the case of actuator faults (affecting matrix B). The model for fault estimation of the system in state-space form (eq.4.10) including the additive actuator faults is given:

$$\begin{cases} \dot{x} = Ax + Bu + \Phi f_u \\ y = Cx \end{cases} \tag{4.41}$$

where $f_u \in \mathbb{R}^{n_u}$ denotes the additive actuator faults and $\Phi \in \mathbb{R}^{n_x \times n_u}$ represents the actuator fault distribution matrix.

Taking into account Section 2.5.3, the states of the system that are directly influenced by the faulty actuators can be shown.

Consider that the lateral actuator δ influences the ω state of the system, the model for the ω state is given by the following equation:

$$\dot{x}_\omega = \sum_{i=1}^{n_x} a_{\omega l} x_l + b_{\omega\delta} u_\delta + b_{\omega\delta} f_{u\delta} \tag{4.42}$$

This equation can be brought to the discrete-time regression form $z = \theta v$ by considering the following expressions:

$$\begin{gathered} x_\omega(k) = \omega \\ \dot{x}_\omega(k) = \frac{x_\omega(k) - x_\omega(k-1)}{T_s} \\ \\ z = \dot{x}_\omega(k) - (A_{32} x_\omega(k) + A_{33} x_\omega(k)) + B_{32}\delta \\ \theta = f_{u\delta} \\ v = B_{32} \end{gathered} \tag{4.43}$$

and solving the least squares problem for θ:

$$\hat{\theta} = \left[\; v(N)^T \cdot v(N) \; \right]^{-1} v(N)^T z(N) \tag{4.44}$$

$\delta_{fault} = \hat{\theta}$ can be estimated.

Note that N is the size of the moving horizon window of data and for the specific case for the lateral actuator fault δ_{fault} affecting state ω is arranged equal to 10.

At this point, the dynamic control action u_D (eq.4.19) that is applied over the vehicle, adding the vector generated by using the fault estimation in (eq.4.44) has the form.

$$u_D = \begin{bmatrix} F_{\mathrm{xR}} \\ \delta \end{bmatrix} + \begin{bmatrix} 0 \\ \delta_{fault} \end{bmatrix} \tag{4.45}$$

In order to avoid the redundant information, the rest of the cases where the actuators faults affect the other states are not presented, as the estimation of these faults is not realizable. The problem arises because in all the other cases the states depend on the state α (slip angle) which is not measurable. Hence, some numerical issues affect the estimation.

Bibliography: [3, 6, 7, 12–18]

Chapter 5

Control Design using Takagi-Sugeno Fuzzy Model

5.1 Description of the Design Method

The Takagi-Sugeno technique allows to use a family of systems for designing the controller. Specifically at each operating point, the system to be controlled is directly defined by the scheduling variables z and thus the system is denoted by:

$$\Gamma(z) = h(z) \tag{5.1}$$

In order to stabilize the system at each operating point for a set of arbitrary values of $\delta \in [\underline{z}, \overline{z}]$ it is essential to stabilize $h(z)$ at the extremes of the scheduling variables. Consequently, being n the number of scheduling variables, 2^n subsystems have to be stabilized so as to stabilize the global system in all operating points

$$\Gamma = h(z_i), \;\; i = 1, 2, ..., 2^n \tag{5.2}$$

which for simplicity it can be written as

$$\Gamma = h_i, \;\; i = 1, ..., 2^n \tag{5.3}$$

where h_i denotes the i_{th} vertices of the corresponding polytope. It has to be noted that the system does not depend on matrix B due to the fact that it is constant and common for all subsystems inside the family. Therefore, the controller design problem, where only the state

feedback control is taken into account, is formulated as:

Let h_i, $i = 1, ..., 2^n$, be defined by (eq.5.3), find a set of controllers K_i that stabilizes and provides satisfactory level of performance for the family of subsystems defined by (eq.5.3) using the state feedback control law $u = Kx$ or $u = -Kx$ (depending on the LMIs are used), where K is the interpolated matrix depending on K_i.

Remark 4 *After careful consideration of the LMIs that have been used in Takagi-Sugeno's implementations, and after applying them to the specific autonomous vehicle model, we have come up with two techniques which match better than the others to our problem. Specifically, for the kinematic controller and the dynamic observer optimization problems, the LQC(sec.2.4.1) technique is used. While this method works quite well regarding the stability of the system, the drawback is that the performance of the system cannot be adjusted in a satisfactory way. Apart from the case that the controller performance cannot be adjusted, if the system is combined with an observer, it cannot perform reliable, as the observer needs to be faster than the controller. Hence, an alternative to the LQC is the LQR(sec.2.4.2) technique via LMI for the H_2 problem is used, in order to adjust the performance of the controller via the decay rate parameter.*

5.2 Kinematic Controller Takagi-Sugeno Design

The kinematic controller is in charge of controlling the position, orientation and linear velocity by means of actuating over the linear and angular velocities of the vehicle. The chosen scheduling variables are v_{d}, ω and θ_{e} bounded in the following intervals:

$$v_{\mathrm{d}} \in [1, 18]\frac{m}{s}$$

$$\omega \in [-1.417, 1.417]\frac{rad}{s}$$

$$\theta_{\mathrm{e}} \in [-0.139, 0.139]rad$$

Remark 5 *In order to control the vehicle in the interval $v_d \in [0, 1]$, a translation has been applied. This means that when computing the controller at $v_d = 0\frac{m}{s}$, we are actually computing the controller at $v_d = 1\frac{m}{s}$ and using it as we are in $v_d = 0\frac{m}{s}$. In this way, we avoid to develop a hybrid control for this reduced velocity interval.*

Consequently we approximate the nonlinear plant by eight Takagi-Sugeno fuzzy rules.

- *Plant rule (1):* if $v_{\mathrm{d}} = 1,\ \omega = -1.417$ and $\theta_{\mathrm{e}} = -0.139$ then $\dot{x} = A_1 x + Bu$
- *Plant rule (2):* if $v_{\mathrm{d}} = 1,\ \omega = -1.417$ and $\theta_{\mathrm{e}} = 0.139$ then $\dot{x} = A_2 x + Bu$
- *Plant rule (3):* if $v_{\mathrm{d}} = 1,\ \omega = 1.417$ and $\theta_{\mathrm{e}} = -0.139$ then $\dot{x} = A_3 x + Bu$
- *Plant rule (4):* if $v_{\mathrm{d}} = 1,\ \omega = 1.417$ and $\theta_{\mathrm{e}} = 0.139$ then $\dot{x} = A_4 x + Bu$
- *Plant rule (5):* if $v_{\mathrm{d}} = 18,\ \omega = -1.417$ and $\theta_{\mathrm{e}} = -0.139$ then $\dot{x} = A_5 x + Bu$
- *Plant rule (6):* if $v_{\mathrm{d}} = 18,\ \omega = -1.417$ and $\theta_{\mathrm{e}} = 0.139$ then $\dot{x} = A_6 x + Bu$
- *Plant rule (7):* if $v_{\mathrm{d}} = 18,\ \omega = 1.417$ and $\theta_{\mathrm{e}} = -0.139$ then $\dot{x} = A_7 x + Bu$
- *Plant rule (8):* if $v_{\mathrm{d}} = 18,\ \omega = 1.417$ and $\theta_{\mathrm{e}} = 0.139$ then $\dot{x} = A_8 x + Bu$

The controller rules are defined by:

- *Controller rule (1):* if $v_{\mathrm{d}} = 1,\ \omega = -1.417$ and $\theta_{\mathrm{e}} = -0.139$ then $u = -F_1 x + r$
- *Controller rule (2):* if $v_{\mathrm{d}} = 1,\ \omega = -1.417$ and $\theta_{\mathrm{e}} = 0.139$ then $u = -F_2 x + r$
- *Controller rule (3):* if $v_{\mathrm{d}} = 1,\ \omega = 1.417$ and $\theta_{\mathrm{e}} = -0.139$ then $u = -F_3 x + r$
- *Controller rule (4):* if $v_{\mathrm{d}} = 1,\ \omega = 1.417$ and $\theta_{\mathrm{e}} = 0.139$ then $u = -F_4 x + r$
- *Controller rule (5):* if $v_{\mathrm{d}} = 18,\ \omega = -1.417$ and $\theta_{\mathrm{e}} = -0.139$ then $u = -F_5 x + r$
- *Controller rule (6):* if $v_{\mathrm{d}} = 18,\ \omega = -1.417$ and $\theta_{\mathrm{e}} = 0.139$ then $u = -F_6 x + r$
- *Controller rule (7):* if $v_{\mathrm{d}} = 18,\ \omega = 1.417$ and $\theta_{\mathrm{e}} = -0.139$ then $u = -F_7 x + r$
- *Controller rule (8):* if $v_{\mathrm{d}} = 18,\ \omega = 1.417$ and $\theta_{\mathrm{e}} = 0.139$ then $u = -F_8 x + r$

The controller gains F can be obtained by solving the following LMI (eq.5.4) optimization problem, applying the linear-quadratic control ($LQC - sec.2.4.1$) technique.

$$\begin{cases} \begin{pmatrix} YA_i^T + A_iY - BW_i - W_i^T B^T & YH^T & W_i^T \\ HY & -I_n & 0 \\ W_i & 0 & -R^{-1} \end{pmatrix} < 0 \\ \begin{pmatrix} \gamma I_n & I_n \\ I_n & Y \end{pmatrix} > 0 \end{cases} \tag{5.4}$$

where $H = Q^{1/2} \geq 0$ and $R = R^T > 0$, with $H \in \mathbb{R}^{q \times n}$, $q = rank(Q)$, $R \in \mathbb{R}^{p \times p}$ and $\gamma > 0$ are the LQC tuning variables.

The desired fuzzy state-feedback gain matrices F_i are then given by:

$$F_\mathrm{i} = W_\mathrm{i} Y^{-1} \tag{5.5}$$

and the interpolated control matrix F is obtained by:

$$F = \sum_{i=1}^{n=8} \mu_\mathrm{i} F_\mathrm{i} \tag{5.6}$$

where μ_i provides the weighting variables in function of the scheduling variables as follows

$$\mu_1 = M_{\underline{v_\mathrm{d}}} M_{\underline{\omega}} M_{\underline{\theta_\mathrm{e}}} \tag{5.7a}$$

$$\mu_2 = M_{\underline{v_\mathrm{d}}} M_{\overline{\omega}} M_{\underline{\theta_\mathrm{e}}} \tag{5.7b}$$

$$\mu_3 = M_{\overline{v_\mathrm{d}}} M_{\underline{\omega}} M_{\underline{\theta_\mathrm{e}}} \tag{5.7c}$$

$$\mu_4 = M_{\overline{v_\mathrm{d}}} M_{\overline{\omega}} M_{\underline{\theta_\mathrm{e}}} \tag{5.7d}$$

$$\mu_5 = M_{\underline{v_\mathrm{d}}} M_{\underline{\omega}} M_{\overline{\theta_\mathrm{e}}} \tag{5.7e}$$

$$\mu_6 = M_{\underline{v_\mathrm{d}}} M_{\overline{\omega}} M_{\overline{\theta_\mathrm{e}}} \tag{5.7f}$$

$$\mu_7 = M_{\overline{v_\mathrm{d}}} M_{\underline{\omega}} M_{\overline{\theta_\mathrm{e}}} \tag{5.7g}$$

$$\mu_8 = M_{\overline{v_\mathrm{d}}} M_{\overline{\omega}} M_{\overline{\theta_\mathrm{e}}} \tag{5.7h}$$

and

$$M_{\underline{v_\mathrm{d}}} = \frac{v_\mathrm{d} - \underline{v_\mathrm{d}}}{\overline{v_\mathrm{d}} - \underline{v_\mathrm{d}}} \tag{5.8a}$$

$$M_{\underline{\omega}} = \frac{\omega - \underline{\omega}}{\overline{\omega} - \underline{\omega}} \tag{5.8b}$$

$$M_{\underline{\theta_\mathrm{e}}} = \frac{\theta_\mathrm{e} - \underline{\theta_\mathrm{e}}}{\overline{\theta_\mathrm{e}} - \underline{\theta_\mathrm{e}}} \tag{5.8c}$$

$$M_{\overline{v_\mathrm{d}}} = 1 - M_{\underline{v_\mathrm{d}}} \tag{5.8d}$$

$$M_{\overline{\omega}} = 1 - M_{\underline{\omega}} \tag{5.8e}$$

$$M_{\overline{\theta_e}} = 1 - M_{\underline{\theta_e}} \tag{5.8f}$$

Finally, the kinematic controller gains F_1 to F_8 for every case of the plants are respectively:

$$F_1 = \begin{bmatrix} -0.3275 & -0.0096 & -0.1953 \\ -0.0668 & -0.4210 & -5.6997 \end{bmatrix} \quad F_1 = \begin{bmatrix} -0.3275 & -0.0096 & -0.1953 \\ -0.0668 & -0.4210 & -5.6988 \end{bmatrix}$$

$$F_3 = \begin{bmatrix} -0.3276 & -0.0227 & -0.2074 \\ -0.0672 & -1.5822 & -6.6147 \end{bmatrix} \quad F_4 = \begin{bmatrix} -0.3276 & -0.0227 & -0.2074 \\ -0.0672 & -1.5822 & -6.6147 \end{bmatrix}$$

$$F_5 = \begin{bmatrix} -0.3276 & 0.0100 & 0.2001 \\ 0.0667 & -0.4210 & -5.6982 \end{bmatrix} \quad F_6 = \begin{bmatrix} -0.3276 & 0.0100 & 0.2001 \\ 0.0667 & -0.4210 & -5.6982 \end{bmatrix}$$

$$F_7 = \begin{bmatrix} -0.3274 & 0.0233 & 0.2100 \\ 0.0667 & -1.5820 & -6.6229 \end{bmatrix} \quad F_8 = \begin{bmatrix} -0.3274 & 0.0233 & 0.2100 \\ 0.0667 & -1.5820 & -6.6229 \end{bmatrix}$$

5.3 Dynamic Controller Takagi-Sugeno Design

The dynamic controller addresses the tracking of the linear and angular velocity references of the vehicle by applying force to the wheels and an angle to the front wheels. The chosen scheduling variables are v, δ, and α bounded in the following intervals:

$$v \in [1, 18]\frac{m}{s}$$

$$\delta \in [-0.4363, 0.4363]rad \rightarrow \sigma \in [0.0873, 0.9599]rad$$

$$\alpha \in [-0.1, 0.1]rad$$

Consequently we approximate the nonlinear plant by eight Takagi-Sugeno fuzzy rules.

- *Plant rule (1):* if $v = 1$ and $\delta = -0.4363$ $\alpha = -0.1$ then $\dot{x} = A_1 x + Bu$
- *Plant rule (2):* if $v = 1$ and $\delta = 0.4363$ $\alpha = -0.1$ then $\dot{x} = A_2 x + Bu$
- *Plant rule (3):* if $v = 18$ and $\delta = -0.4363$ $\alpha = -0.1$ then $\dot{x} = A_3 x + B_3 u$
- *Plant rule (4):* if $v = 18$ and $\delta = 0.4363$ $\alpha = -0.1$ then $\dot{x} = A_4 x + Bu$
- *Plant rule (5):* if $v = 1$ and $\delta = -0.4363$ $\alpha = 0.1$ then $\dot{x} = A_5 x + Bu$
- *Plant rule (6):* if $v = 1$ and $\delta = 0.4363$ $\alpha = 0.1$ then $\dot{x} = A_6 x + Bu$
- *Plant rule (7):* if $v = 18$ and $\delta = -0.4363$ $\alpha = 0.1$ then $\dot{x} = A_7 x + Bu$

- *Plant rule (8):* if $v = 18$ and $\delta = 0.4363$ $\alpha = 0.1$ then $\dot{x} = A_8 x + Bu$

The controller rules are defined by:

- *Controller rule (1):* if $v = 1$ and $\delta = -0.4363$ $\alpha = -0.1$ then $u = K_1 x$
- *Controller rule (2):* if $v = 1$ and $\delta = 0.4363$ $\alpha = -0.1$ then $u = K_2 x$
- *Controller rule (3):* if $v = 18$ and $\delta = -0.4363$ $\alpha = -0.1$ then $u = K_3 x$
- *Controller rule (4):* if $v = 18$ and $\delta = 0.4363$ $\alpha = -0.1$ then $u = K_4 x$
- *Controller rule (5):* if $v = 1$ and $\delta = -0.4363$ $\alpha = 0.1$ then $u = K_5 x$
- *Controller rule (6):* if $v = 1$ and $\delta = 0.4363$ $\alpha = 0.1$ then $u = K_6 x$
- *Controller rule (7):* if $v = 18$ and $\delta = -0.4363$ $\alpha = 0.1$ then $u = K_7 x$
- *Controller rule (8):* if $v = 18$ and $\delta = 0.4363$ $\alpha = 0.1$ then $u = K_8 x$

The controller gains K can be obtained, by solving the following LMI (eq.5.9) optimization problem, where the linear-quadratic regulation ($LQR - sec.2.4.2$) technique for the H_2 problem, is used.

$$\begin{cases} (A_\mathrm{i}P + BW_\mathrm{i}) + (A_\mathrm{i}P + BW_\mathrm{i})^T + 2nP < 0 \\ \begin{bmatrix} -Y & R^{\frac{1}{2}}W_\mathrm{i} \\ (R^{\frac{1}{2}}W_\mathrm{i})^T & -P \end{bmatrix} < 0, \qquad i = 1, ..., 8 \\ trace(Q^{\frac{1}{2}}P(Q^{\frac{1}{2}})^T) + trace(Y) < \gamma \end{cases} \tag{5.9}$$

where $Q = Q^T > 0$, $R = R^T > 0$, and $\gamma > 0$ are the LQR tuning variables, and n is the decay rate (performance) parameter.

The desired fuzzy state-feedback gain matrices K_i are then given by:

$$K_\mathrm{i} = W_\mathrm{i} P^{-1} \tag{5.10}$$

and the interpolated control matrix K is obtained by:

$$K = \sum_{i=1}^{n=8} \mu_\mathrm{i} K_\mathrm{i} \tag{5.11}$$

where μ_{i} provides the weighting variables in function of the scheduling variables as follows

$$\mu_1 = M_{\underline{v}} M_{\underline{\delta}} M_{\underline{\alpha}} \tag{5.12a}$$

$$\mu_2 = M_{\underline{v}} M_{\underline{\delta}} M_{\overline{\alpha}} \tag{5.12b}$$

$$\mu_3 = M_{\underline{v}} M_{\overline{\delta}} M_{\underline{\alpha}} \tag{5.12c}$$

$$\mu_4 = M_{\underline{v}} M_{\overline{\delta}} M_{\overline{\alpha}} \tag{5.12d}$$

$$\mu_5 = M_{\overline{v}} M_{\underline{\delta}} M_{\underline{\alpha}} \tag{5.12e}$$

$$\mu_6 = M_{\overline{v}} M_{\underline{\delta}} M_{\overline{\alpha}} \tag{5.12f}$$

$$\mu_7 = M_{\overline{v}} M_{\overline{\delta}} M_{\underline{\alpha}} \tag{5.12g}$$

$$\mu_8 = M_{\overline{v}} M_{\overline{\delta}} M_{\overline{\alpha}} \tag{5.12h}$$

and

$$M_{\underline{v}} = \frac{v - \underline{v}}{\overline{v} - \underline{v}} \tag{5.13a}$$

$$M_{\underline{\delta}} = \frac{\delta - \underline{\delta}}{\overline{\delta} - \underline{\delta}} \tag{5.13b}$$

$$M_{\underline{\alpha}} = \frac{\alpha - \underline{\alpha}}{\overline{\alpha} - \underline{\alpha}} \tag{5.13c}$$

$$M_{\overline{v}} = 1 - M_{\underline{v}} \tag{5.13d}$$

$$M_{\overline{\delta}} = 1 - M_{\underline{\delta}} \tag{5.13e}$$

$$M_{\overline{\alpha}} = 1 - M_{\underline{\alpha}} \tag{5.13f}$$

Finally, the dynamic controller gains K_1 to K_8 for every case of the plants are the following:

$$K_1 = 10000 \begin{bmatrix} -0.8702 & -0.1147 & -0.0126 & -0.0001 & -0.0052 & 1.3751 & 2.7299 \\ 0.0003 & -0.0027 & -0.0021 & 0.0000 & -0.0018 & 0.0430 & -0.0010 \end{bmatrix}$$

$$K_2 = 10000 \begin{bmatrix} -0.8706 & 0.1223 & -0.0867 & -0.0001 & -0.0106 & 2.6050 & 2.7324 \\ 0.0004 & -0.0026 & -0.0021 & 0.0000 & -0.0018 & 0.0425 & -0.0015 \end{bmatrix}$$

$$K_3 = 10000 \begin{bmatrix} -0.6076 & -0.0242 & -0.0327 & -0.0000 & -0.0054 & 0.6706 & 1.7203 \\ 0.0004 & -0.0011 & -0.0029 & 0.0000 & -0.0018 & 0.0391 & -0.0012 \end{bmatrix}$$

$$K_4 = 10000 \begin{bmatrix} -0.6079 & 0.0336 & -0.0613 & -0.0000 & -0.0078 & 0.9535 & 1.7211 \\ 0.0005 & -0.0011 & -0.0028 & 0.0000 & -0.0018 & 0.0386 & -0.0017 \end{bmatrix}$$

$$K_5 = 10000 \begin{bmatrix} -0.8695 & -0.1242 & 0.0879 & -0.0001 & 0.0107 & -2.6224 & 2.7300 \\ -0.0004 & -0.0026 & -0.0021 & -0.0000 & -0.0018 & 0.0425 & 0.0015 \end{bmatrix}$$

$$K_6 = 10000 \begin{bmatrix} -0.8729 & 0.1126 & 0.0133 & -0.0001 & 0.0053 & -1.3607 & 2.7380 \\ -0.0003 & -0.0027 & -0.0021 & -0.0000 & -0.0018 & 0.0430 & 0.0010 \end{bmatrix}$$

$$K_7 = 10000 \begin{bmatrix} -0.6094 & -0.0334 & 0.0594 & -0.0000 & 0.0080 & -0.9182 & 1.7263 \\ -0.0005 & -0.0011 & -0.0028 & -0.0000 & -0.0018 & 0.0386 & 0.0017 \end{bmatrix}$$

$$K_8 = 10000 \begin{bmatrix} -0.6066 & 0.0246 & 0.0329 & -0.0000 & 0.0054 & -0.6752 & 1.7163 \\ -0.0003 & -0.0011 & -0.0029 & -0.0000 & -0.0018 & 0.0391 & 0.0012 \end{bmatrix}$$

5.4 Dynamic Observer Takagi-Sugeno Design

The dynamic observer is responsible for the estimation of the linear, angular velocity and slip angle states of the vehicle. The chosen scheduling variables are v, δ, and α bounded in the following intervals:

$$v \in [1, 18] \frac{m}{s}$$

$$\delta \in [-0.4363, 0.4363] rad \rightarrow \sigma \in [0.0873, 0.9599] rad$$

$$\alpha \in [-0.1, 0.1] rad$$

Consequently we approximate the nonlinear plant by eight Takagi-Sugeno fuzzy rules, which give the state estimation for each case.

- *Observer rule (1):* if $v = 1$ and $\delta = -0.4363$ $\alpha = -0.1$ then
  $\dot{\hat{x}}(t) = A_{obs1}x(t) + B_{obs1}u(t) - L_1[y(t) - \hat{y}(t)]$
- *Observer rule (2):* if $v = 1$ and $\delta = 0.4363$ $\alpha = -0.1$ then
  $\dot{\hat{x}}(t) = A_{obs2}x(t) + B_{obs2}u(t) - L_2[y(t) - \hat{y}(t)]$

- *Observer rule (3):* if $v = 18$ and $\delta = -0.4363$ $\alpha = -0.1$ then
  $\dot{\hat{x}}(t) = A_{obs3}x(t) + B_{obs3}u(t) - L_3[y(t) - \hat{y}(t)]$
- *Observer rule (4):* if $v = 18$ and $\delta = 0.4363$ $\alpha = -0.1$ then
  $\dot{\hat{x}}(t) = A_{obs4}x(t) + B_{obs4}u(t) - L_4[y(t) - \hat{y}(t)]$
- *Observer rule (5):* if $v = 1$ and $\delta = -0.4363$ $\alpha = 0.1$ then
  $\dot{\hat{x}}(t) = A_{obs5}x(t) + B_{obs5}u(t) - L_5[y(t) - \hat{y}(t)]$
- *Observer rule (6):* if $v = 1$ and $\delta = 0.4363$ $\alpha = 0.1$ then
  $\dot{\hat{x}}(t) = A_{obs6}x(t) + B_{obs6}u(t) - L_6[y(t) - \hat{y}(t)]$
- *Observer rule (7):* if $v = 18$ and $\delta = -0.4363$ $\alpha = 0.1$ then
  $\dot{\hat{x}}(t) = A_{obs7}x(t) + B_{obs7}u(t) - L_7[y(t) - \hat{y}(t)]$
- *Observer rule (8):* if $v = 18$ and $\delta = 0.4363$ $\alpha = 0.1$ then
  $\dot{\hat{x}}(t) = A_{obs8}x(t) + B_{obs8}u(t) - L_8[y(t) - \hat{y}(t)]$

The observer gains L can be obtained by solving the following LMI (eq.5.14) optimization problem, applying duality to the linear-quadratic control ($LQC - sec.2.4.1$) technique.

$$\begin{cases} \begin{pmatrix} YA_i^T + A_iY - C_{obs}^T W_i - W_i^T C_{obs} & YH^T & W_i^T \\ HY & -I_n & 0 \\ W_i & 0 & -R^{-1} \end{pmatrix} < 0 \\ \begin{pmatrix} \gamma I_n & I_n \\ I_n & Y \end{pmatrix} > 0 \end{cases} \qquad i = 1, 2, ..., r \qquad (5.14)$$

where $H = Q^{1/2} \geq 0$ and $R = R^T > 0$, with $H \in \mathbb{R}^{q \times n}$, $q = rank(Q)$, $R \in \mathbb{R}^{p \times p}$ and $\gamma > 0$ are the LQC tuning variables.

The desired fuzzy state-feedback gain matrices L_i are then given by:

$$L_\mathrm{i} = (W_\mathrm{i} Y^{-1})^T \qquad (5.15)$$

and the interpolated observer gain matrix L is obtained by:

$$L = \sum_{i=1}^{n=8} \mu_\mathrm{i} L_\mathrm{i} \qquad (5.16)$$

where μ_{i} provides the weighting variables of the observer in function of the scheduling variables as follows

$$\mu_1 = M_{\underline{v}} M_{\underline{\delta}} M_{\underline{\alpha}} \tag{5.17a}$$

$$\mu_2 = M_{\underline{v}} M_{\underline{\delta}} M_{\overline{\alpha}} \tag{5.17b}$$

$$\mu_3 = M_{\underline{v}} M_{\overline{\delta}} M_{\underline{\alpha}} \tag{5.17c}$$

$$\mu_4 = M_{\underline{v}} M_{\overline{\delta}} M_{\overline{\alpha}} \tag{5.17d}$$

$$\mu_5 = M_{\overline{v}} M_{\underline{\delta}} M_{\underline{\alpha}} \tag{5.17e}$$

$$\mu_6 = M_{\overline{v}} M_{\underline{\delta}} M_{\overline{\alpha}} \tag{5.17f}$$

$$\mu_7 = M_{\overline{v}} M_{\overline{\delta}} M_{\underline{\alpha}} \tag{5.17g}$$

$$\mu_8 = M_{\overline{v}} M_{\overline{\delta}} M_{\overline{\alpha}} \tag{5.17h}$$

and

$$M_{\underline{v}} = \frac{v - \underline{v}}{\overline{v} - \underline{v}} \tag{5.18a}$$

$$M_{\underline{\delta}} = \frac{\delta - \underline{\delta}}{\overline{\delta} - \underline{\delta}} \tag{5.18b}$$

$$M_{\underline{\alpha}} = \frac{\alpha - \underline{\alpha}}{\overline{\alpha} - \underline{\alpha}} \tag{5.18c}$$

$$M_{\overline{v}} = 1 - M_{\underline{v}} \tag{5.18d}$$

$$M_{\overline{\delta}} = 1 - M_{\underline{\delta}} \tag{5.18e}$$

$$M_{\overline{\alpha}} = 1 - M_{\underline{\alpha}} \tag{5.18f}$$

Finally, the dynamic observer gains L_1 to L_8 for every case of the plants are respectively:

$$L_1 = 10000\begin{bmatrix} 0.0170 & -0.3451 \\ -0.0013 & 1.0828 \\ -0.0002 & 0.1473 \end{bmatrix} \quad L_2 = 10000\begin{bmatrix} 0.1750 & -2.2448 \\ -0.0153 & 9.0538 \\ -0.0018 & 1.2836 \end{bmatrix}$$

$$L_3 = 10000\begin{bmatrix} 0.1757 & 1.2634 \\ 0.0028 & 7.8525 \\ 0.0005 & 1.2024 \end{bmatrix} \quad L_4 = 10000\begin{bmatrix} 0.1695 & 2.5326 \\ -0.0052 & 9.1159 \\ -0.0005 & 1.2871 \end{bmatrix}$$

$$L_5 = 10000\begin{bmatrix} 0.0169 & 0.2119 \\ -0.0012 & 1.2320 \\ -0.0002 & 0.1497 \end{bmatrix} \quad L_6 = 10000\begin{bmatrix} 0.0179 & 0.1250 \\ -0.0022 & 1.1581 \\ -0.0004 & 0.1465 \end{bmatrix}$$

$$L_7 = 10000\begin{bmatrix} 0.0179 & -0.1779 \\ -0.0006 & 1.0485 \\ -0.0000 & 0.1310 \end{bmatrix} \quad L_8 = 10000\begin{bmatrix} 0.0169 & -0.2736 \\ -0.0018 & 1.2058 \\ -0.0002 & 0.1533 \end{bmatrix}$$

5.5 Unknown Input Observer Takagi-Sugeno Design

The dynamic UIO is responsible for the estimation of the linear, angular velocity and slip angle states, as well as the faults of actuators of the vehicle. The chosen scheduling variables v, δ, and α are bounded in the following intervals:

$$v \in [1, 18]\frac{m}{s}$$

$$\delta \in [-0.4363, 0.4363]rad \rightarrow \sigma \in [0.0873, 0.9599]rad$$

$$\alpha \in [-0.1, 0.1]rad$$

Consequently we approximate the nonlinear plant by eight Takagi-Sugeno fuzzy rules, which give the state estimation for each case.

- *Observer rule (1):* if $v = 1$ and $\delta = -0.4363$ $\alpha = -0.1$ then
  $\dot{\hat{x}}_{uio} = (A_{uio1} - L_{uio1}C_{obs})\hat{x}_{uio} + B_{uio1}u - E_1H\dot{y} + L_{uio1}y$
- *Observer rule (2):* if $v = 1$ and $\delta = 0.4363$ $\alpha = -0.1$ then
  $\dot{\hat{x}}_{uio} = (A_{uio2} - L_{uio2}C_{obs})\hat{x}_{uio} + B_{uio2}u - E_2H\dot{y} + L_{uio2}y$
- *Observer rule (3):* if $v = 18$ and $\delta = -0.4363$ $\alpha = -0.1$ then
  $\dot{\hat{x}}_{uio} = (A_{uio3} - L_{uio3}C_{obs})\hat{x}_{uio} + B_{uio3}u - E_3H\dot{y} + L_{uio3}y$

- *Observer rule (4):* if $v = 18$ and $\delta = 0.4363$ $\alpha = -0.1$ then
  $\dot{\hat{x}}_{uio} = (A_{uio4} - L_{uio4}C_{obs})\hat{x}_{uio} + B_{uio4}u - E_4H\dot{y} + L_{uio4}y$

- *Observer rule (5):* if $v = 1$ and $\delta = -0.4363$ $\alpha = 0.1$ then
  $\dot{\hat{x}}_{uio} = (A_{uio5} - L_{uio5}C_{obs})\hat{x}_{uio} + B_{uio5}u - E_5H\dot{y} + L_{uio5}y$

- *Observer rule (6):* if $v = 1$ and $\delta = 0.4363$ $\alpha = 0.1$ then
  $\dot{\hat{x}}_{uio} = (A_{uio6} - L_{uio6}C_{obs})\hat{x}_{uio} + B_{uio6}u - E_6H\dot{y} + L_{uio6}y$

- *Observer rule (7):* if $v = 18$ and $\delta = -0.4363$ $\alpha = 0.1$ then
  $\dot{\hat{x}}_{uio} = (A_{uio7} - L_{uio7}C_{obs})\hat{x}_{uio} + B_{uio7}u - E_7H\dot{y} + L_{uio7}y$

- *Observer rule (8):* if $v = 18$ and $\delta = 0.4363$ $\alpha = 0.1$ then
  $\dot{\hat{x}}_{uio} = (A_{uio8} - L_{uio8}C_{obs})\hat{x}_{uio} + B_{uio8}u - E_8H\dot{y} + L_{uio8}y$

The observer gains L_{uio} can be obtained by solving the following LMI (5.19) optimization problem, applying duality to the linear-quadratic control ($LQC - 2.4.1$) technique in the same way as it is used in the previous section (eq.5.14). Although, because of some stability and performance problems, it is needed the addition of some additional LMIs (5.20) that was taken from ([1] chapter 4).

$$\begin{cases} \begin{pmatrix} YA_i^T + A_iY - C_{obs}^TW_i - W_i^TC_{obs} & YH^T & W_i^T \\ HY & -I_n & 0 \\ W_i & 0 & -R^{-1} \end{pmatrix} < 0 \\ \begin{pmatrix} \gamma I_n & I_n \\ I_n & Y \end{pmatrix} > 0 \end{cases} \qquad i = 1, 2, ..., r \qquad (5.19)$$

where $H = Q^{1/2} \geq 0$ and $R = R^T > 0$, with $H \in \mathbb{R}^{q\times n}$, $q = rank(Q)$, $R \in \mathbb{R}^{p\times p}$ and $\gamma > 0$ are the LQC tuning variables.

$$\begin{cases} A_i^TY - C_{obs}^TW_i + YA_i - W_i^TC_{obs} + (s+1)X + 2\alpha Y < 0 \\ A_i^TY - C_{obs}^TW_i + YA_i - W_i^TC_{obs} - 2X + 4\alpha Y + A_j^TY - C_{obs}^TW_j + YA_j - W_j^TC_{obs} < 0 \end{cases} \qquad (5.20)$$

where α is the decay rate parameter, $1 < s \leq r$ where r is the number of rules, and $i = 1, 2, ..., r$ and $i < j$ s.t.$h_i \cap h_j \neq \Phi$

The desired fuzzy state-feedback gain matrices L_i are then given by:

$$L_{uioi} = (W_\mathrm{i} Y^{-1})^T \tag{5.21}$$

and the interpolated observer gain matrix L_{uio} is obtained by:

$$L_{uio} = \sum_{i=1}^{n=8} \mu_\mathrm{i} L_{uioi} \tag{5.22}$$

where μ_i provides the weighting variables of the observer in function of the scheduling variables as follows

$$\mu_1 = M_{\underline{v}} M_{\underline{\delta}} M_{\underline{\alpha}} \tag{5.23a}$$

$$\mu_2 = M_{\underline{v}} M_{\underline{\delta}} M_{\overline{\alpha}} \tag{5.23b}$$

$$\mu_3 = M_{\underline{v}} M_{\overline{\delta}} M_{\underline{\alpha}} \tag{5.23c}$$

$$\mu_4 = M_{\underline{v}} M_{\overline{\delta}} M_{\overline{\alpha}} \tag{5.23d}$$

$$\mu_5 = M_{\overline{v}} M_{\underline{\delta}} M_{\underline{\alpha}} \tag{5.23e}$$

$$\mu_6 = M_{\overline{v}} M_{\underline{\delta}} M_{\overline{\alpha}} \tag{5.23f}$$

$$\mu_7 = M_{\overline{v}} M_{\overline{\delta}} M_{\underline{\alpha}} \tag{5.23g}$$

$$\mu_8 = M_{\overline{v}} M_{\overline{\delta}} M_{\overline{\alpha}} \tag{5.23h}$$

where

$$M_{\underline{v}} = \frac{v - \underline{v}}{\overline{v} - \underline{v}} \tag{5.24a}$$

$$M_{\underline{\delta}} = \frac{\delta - \underline{\delta}}{\overline{\delta} - \underline{\delta}} \tag{5.24b}$$

$$M_{\underline{\alpha}} = \frac{\alpha - \underline{\alpha}}{\overline{\alpha} - \underline{\alpha}} \tag{5.24c}$$

$$M_{\overline{v}} = 1 - M_{\underline{v}} \tag{5.24d}$$

$$M_{\overline{\delta}} = 1 - M_{\underline{\delta}} \tag{5.24e}$$

$$M_{\overline{\alpha}} = 1 - M_{\underline{\alpha}} \tag{5.24f}$$

Finally, the UIO gains L_{uio1} to L_{uio8} for every case of the plants are respectively:

with longitudinal actuator fault $F_{xRfault}$:

$$L_{uio1} = 1000 \begin{bmatrix} 0.2324 & -0.0040 \\ -0.2388 & 7.4585 \\ -0.0330 & 1.3179 \end{bmatrix} \quad L_{uio2} = 1000 \begin{bmatrix} 0.2324 & 0.0040 \\ 0.2253 & 7.4155 \\ 0.0320 & 1.3150 \end{bmatrix}$$

$$L_{uio3} = 1000 \begin{bmatrix} 0.2321 & -0.0010 \\ -0.0997 & 9.5243 \\ -0.0176 & 1.7195 \end{bmatrix} \quad L_{uio4} = 1000 \begin{bmatrix} 0.2321 & 0.0010 \\ 0.1000 & 9.4884 \\ 0.0177 & 1.7131 \end{bmatrix}$$

$$L_{uio5} = 1000 \begin{bmatrix} 0.2325 & -0.0035 \\ -0.2182 & 8.6456 \\ -0.0303 & 1.5088 \end{bmatrix} \quad L_{uio6} = 1000 \begin{bmatrix} 0.2325 & 0.0035 \\ 0.2302 & 8.6492 \\ 0.0311 & 1.5074 \end{bmatrix}$$

$$L_{uio7} = 10000 \begin{bmatrix} 0.0232 & -0.0003 \\ -0.0217 & 1.0724 \\ -0.0028 & 0.1883 \end{bmatrix} \quad L_{uio8} = 10000 \begin{bmatrix} 0.0232 & 0.0003 \\ 0.0216 & 1.0745 \\ 0.0028 & 0.1889 \end{bmatrix}$$

5.6 Augmented Controller and Observer Takagi-Sugeno Design

The dynamic Augmented Controller design is obtained following the same procedure with Section 5.3 and only adjusting the state space model as:

$$\dot{x}_{aug} = A_{aug}x_{aug} + B_{aug}u \tag{5.25}$$

The dynamic Augmented Observer design is obtained following the same procedure with Section 5.5 and only adjusting the observer state space model as:

$$\dot{\hat{x}}_{aug}(t) = A_{aug}x_{aug}(t) + B_{aug}u(t) - L[y_{aug}(t) - \hat{y}_{aug}(t)] \tag{5.26}$$

Chapter 6

Results

6.1 Introduction

The simulation results of the Kinematic and Dynamic Controllers and Dynamic Observer are presented in this Chapter. In Section 6.2 the simulation of each model is performed separated from one another. As the experiments progress the models are combined. First the Dynamic Controller and Observer are tested together. In Section 6.3 the Kinematic and Dynamic controllers are combined and subsequently the Cascade control is implemented with all the three models combined. Finally, in Section 6.4, the Fault Tolerant Control experiments take place employing the Cascade control scheme, and using Unknown Input Observer, Augmented Observer and Least Squares Parameter Estimation methods so as to compensate the faults. In order to check the Kinematic and Dynamic behavior of the vehicle, variable driving situations are covered. The simulation is arranged so as to have different values of the desired linear and angular velocities, as well as different values of the desired longitudinal and lateral position and orientation. Also it is arranged, the vehicle to accelerate, and reduce velocity on curves and slow down at the end of the route.

Remark 6 *The simulation is carried out by using MATLAB software, YALMIP optimization toolbox and SEDUMI solver.*

6.2 Decoupled Models Simulation

6.2.1 Kinematic Controller Simulation

In order to perform the Kinematic Controller simulation, it is need the suitable selection of the LMI parameters Q and R. The selection is done by using the root mean square error (RMSE) (Table 6.1). This index allows us to choose suitable Q and R parameters. The lower the values of RMSE is, the better the controller behavior. Longitudinal, lateral position and orientation errors are chosen for performing the comparison of the experiments. The results of the simulation are presented in Fig.6.1 and Fig.6.2.

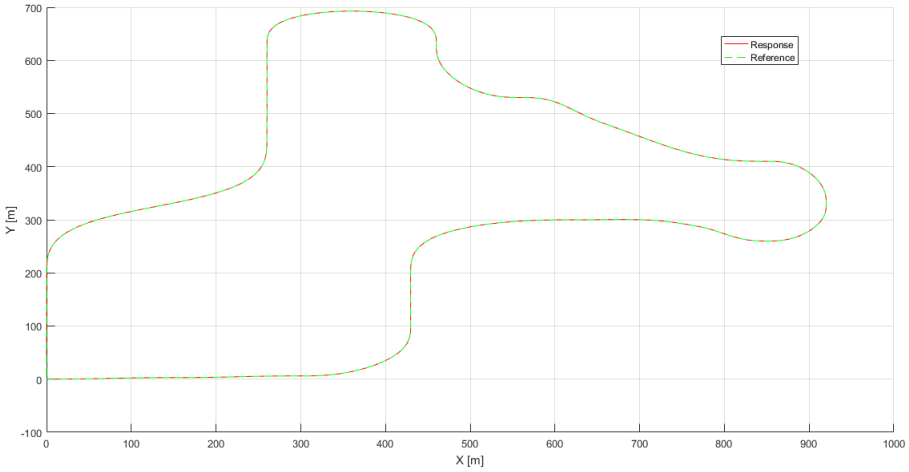

Figure 6.1: Kinematic Controler - Trajectory of the Vehicle

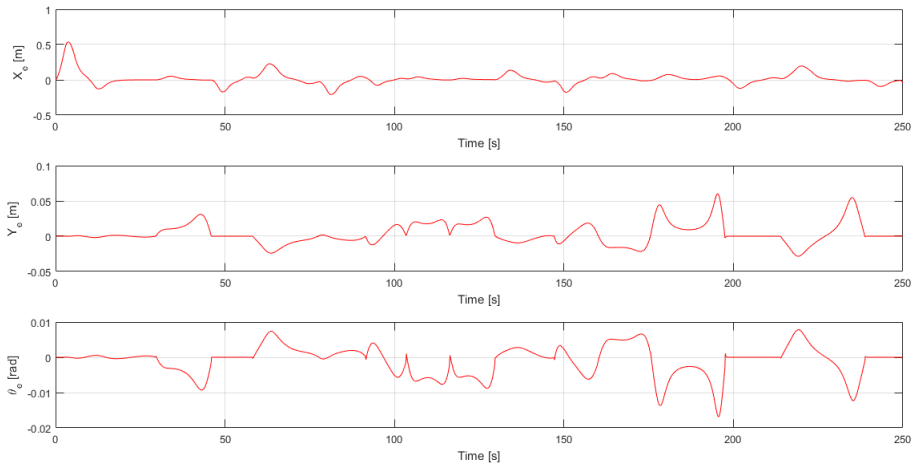

Figure 6.2: Kinematic Controller Position and Orientation Errors

We can observe that the controller works quite efficiently, as it follows the planned trajectory without deviating. Although, checking the Figure 6.2 and Table 6.1 there are some errors ralated with longitudinal, lateral position and orientation. It can be justified, as during the experiment the desired velocities and positions obtain miscellaneous values in an abrupt manner and this is why the vehicle reacts in this way. The most significant of these errors, is the lateral one Y_e as it is the one that shows if the vehicle follows the straight trajectory. As we can see the error Y_e is small, almost 5cm the biggest value. On the other hand, X_e the longitudinal error has about 0.5m as biggest value but it is only during the initial state where the vehicle accelerates.

Q	R	X_{RMSE}	Y_{RMSE}	θ_{RMSE}
$\begin{bmatrix} 0.1 & 0 & 0 \\ 0 & 0.001 & 0 \\ 0 & 0 & 1 \end{bmatrix}$	$\begin{bmatrix} 3 & 0 \\ 0 & 0.1 \end{bmatrix}$	0.1038	0.0149	0.0043
$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$	$\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$	0.0924	0.0536	0.0687
$\begin{bmatrix} 0.1 & 0 & 0 \\ 0 & 0.1 & 0 \\ 0 & 0 & 0.1 \end{bmatrix}$	$\begin{bmatrix} 0.1 & 0 \\ 0 & 0.1 \end{bmatrix}$	**0.0316**	**0.0131**	**0.0045**

Table 6.1: Kinematic Controller RMSE

In Table 6.1 above, some different values of Q and R parameters are presented among many choices during the performed experiments. The chosen parameters with the lowest RMSE, are presented in bolt.

Remark 7 *At this point it is needed to note some information for the references of the Kinematic Controller. These desired values of the states are first processed offline in the trajectory planner, which is in charge of generating the feasible position and orientation references and provide them to the Kinematic Controller.*

6.2.2 Dynamic Controller Simulation

Similar to the previous experiment, for Dynamic Controller tuning, it is needed the suitable selection of the LMI parameters Q and R. Additionally, the decay rate parameter, which is presented in the (eq.5.9), is arranged to be equal to 1.8. Performing the experiments it was realized that the lateral behavior of the vehicle is hard to be controlled. For this reason the values of Q matrix correspond to angular velocity as well as angular velocity integral is needed to have larger than the others. The selection is done similarly to the Section 6.2.1, by using the RMSE from Table 6.2, as much lower its value is, the better the controller behavior. Linear and angular velocity errors are chosen for performing the comparison of the experiments . The results of the simulation are presented in Fig.6.3.

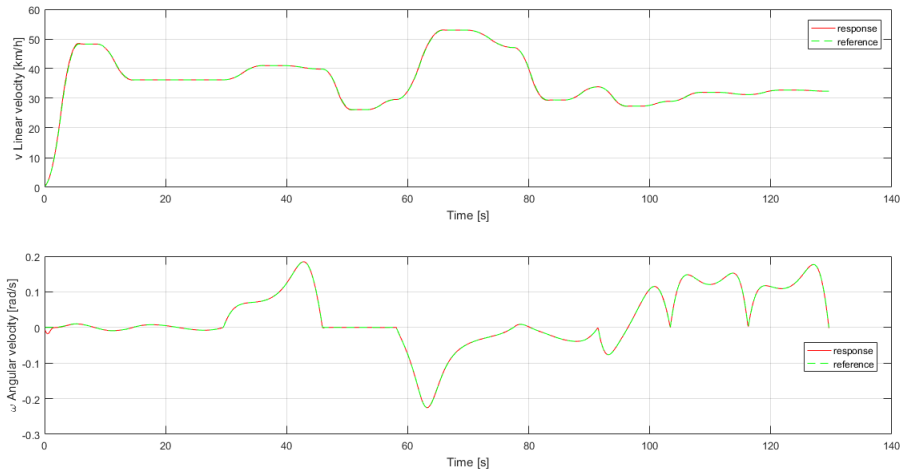

Figure 6.3: Dynamic Controller States

It can be seen that the dynamic controller works almost perfectly, as it follows the references of linear and angular velocities without deviating, throughout the experiment. The only deviation exists for the angular velocity at about 0.02 rad/s during the first second of the movement. This error can be justified as the vehicle has to change direction in order to

follow the planned trajectory. Additionally we can observe the RMSEs in Table 6.2 where the ω error is equal to 0.0028 rad/s, and we could say that it is an accepted value. Regarding the linear velocity error, the RMSE is equal to 0.0406 km/h and we can say again that this error is negligible.

Q	R	v_{RMSE}	ω_{RMSE}
$\begin{bmatrix} 0.05 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.01 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.01 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.01 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$	$\begin{bmatrix} 0.01 & 0 \\ 0 & 10 \end{bmatrix}$	0.2404	0.4968
$\begin{bmatrix} 0.05 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.01 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.01 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.01 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1000 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1000 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$	$\begin{bmatrix} 0.01 & 0 \\ 0 & 1 \end{bmatrix}$	0.0636	0.4329
$\begin{bmatrix} 0.05 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.01 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.01 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.01 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 100000 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 100000 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$	$\begin{bmatrix} 0.01 & 0 \\ 0 & 100 \end{bmatrix}$	**0.0406**	**0.0028**

Table 6.2: Dynamic Controller RMSE

In Table 6.2 above, some different values of Q and R parameters are presented among many choices during the performed experiments. The chosen parameters with the lowest RMSE, are presented in bolt.

6.2.3 Dynamic Observer Simulation

As it was discussed in previous chapters, the observer is used for estimating the states in the case that we cannot measure them. In our case this state is the slip angle α. In addition we estimate the linear and angular velocity states. The selection of the LMI parameters Q and R is done in the same way as in previous sections, by using the RMSE from Table 6.3. The results of the dynamic observer simulation are presented in Fig.6.4.

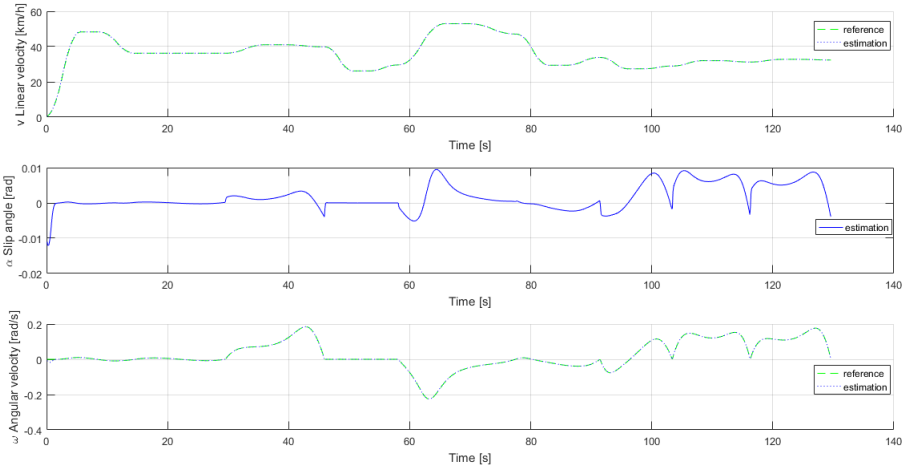

Figure 6.4: Dynamic Observer State Estimation

As we can see, the estimation of the states of linear and angular velocities follows the references in an almost absolute way. Nevertheless, the estimation of the slip angle can give some additional information. We can observe, that the estimated value of the slip angle, is no more than 0.01 rad. According to the bibliography and the range of the slip angle arranged in the construction of the control system (see Section5.4), it is in the permitted range so as vehicle not to slip away. Using this estimation and providing the values to the dynamic controller (see Section6.2.4), we can prevent the vehicle to have undesired behavior.

Q	R	v_{RMSE}	ω_{RMSE}
$\begin{bmatrix} 0.01 & 0 & 0 \\ 0 & 0.01 & 0 \\ 0 & 0 & 0.01 \end{bmatrix}$	$\begin{bmatrix} 0.01 & 0 \\ 0 & 0.1 \end{bmatrix}$	0.0041	6.5410e-05
$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$	$\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$	0.0034	4.2187e-05
$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$	$\begin{bmatrix} 1 & 0 \\ 0 & 0.005 \end{bmatrix}$	**0.0034**	**4.2035e-05**

Table 6.3: Dynamic Observer RMSE

In Table 6.3 above, some various values of Q and R parameters among many choices during the performed experiments, are presented. The chosen parameters with the lowest RMSE, are indicated in bold.

6.2.4 Dynamic Controller and Observer Combination

The estimated dynamic observer states are provided to the dynamic controller so as to control the vehicle properly. The Q an R parameters are taken from Table 6.2 for the controller and from Table 6.3 for the observer.

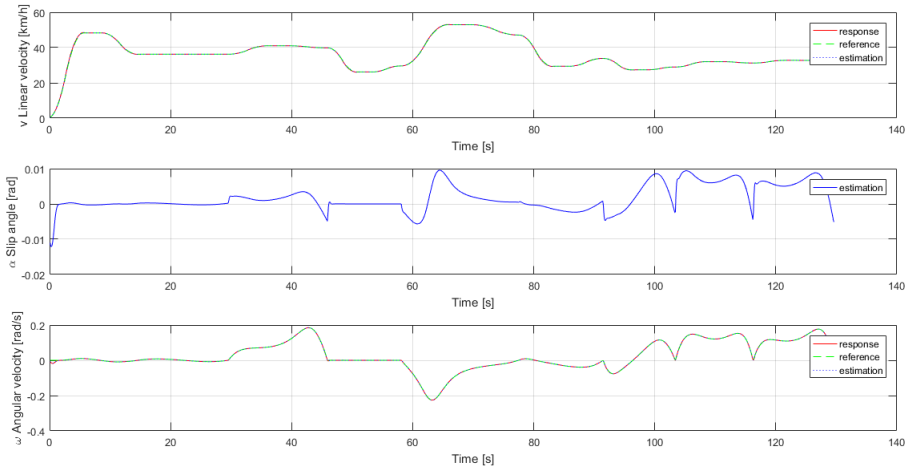

Figure 6.5: Dynamic Controller and Observer States

We can see that the behavior of the system has the same response as the in previous experiments. The real values of the linear and angular velocities follow the references, as well as the estimation of the states. Additionally, the estimation of slip angle is the similar with the decoupled observer simulation, Fig.6.4, and it does not take large values.

$Q_{controllerD}$	$R_{controllerD}$	v_{RMSE}	ω_{RMSE}
$\begin{bmatrix} 0.05 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.01 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.01 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.01 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 100000 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 100000 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$	$\begin{bmatrix} 0.01 & 0 \\ 0 & 100 \end{bmatrix}$	**0.0407**	**0.0028**
$Q_{observerD}$	$R_{observerD}$		
$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$	$\begin{bmatrix} 1 & 0 \\ 0 & 0.005 \end{bmatrix}$		

Table 6.4: Dynamic Controller and Observer RMSE

In the above Table 6.4, the choice of Q and R parameters is done by taking the best values of them, obtained in previous experiments. As it can be seen, the RMSEs have the same values as in the Dynamic Controller experiment (Table 6.2), which are insignificant to affect

the behavior of the vehicle. It means that the observer usage, has not affected the behavior of the control scheme.

6.3 Cascade Control Simulation

The simulation of the combined Kinematic and Dynamic Controllers and Dynamic Observer, is shown in this section. In the first experiment, the Observer is not used. Subsequently, the Observer is added and the Complete Control scheme as it is shown in Fig.6.6 is simulated. At this point, it is needed to note that the Dynamic Controller has to be faster than the Kinematic Controller, and as it is well known, the Dynamic Observer has to be faster than the Dynamic Controller. It can be verified by checking the poles of the system in Fig.6.7.

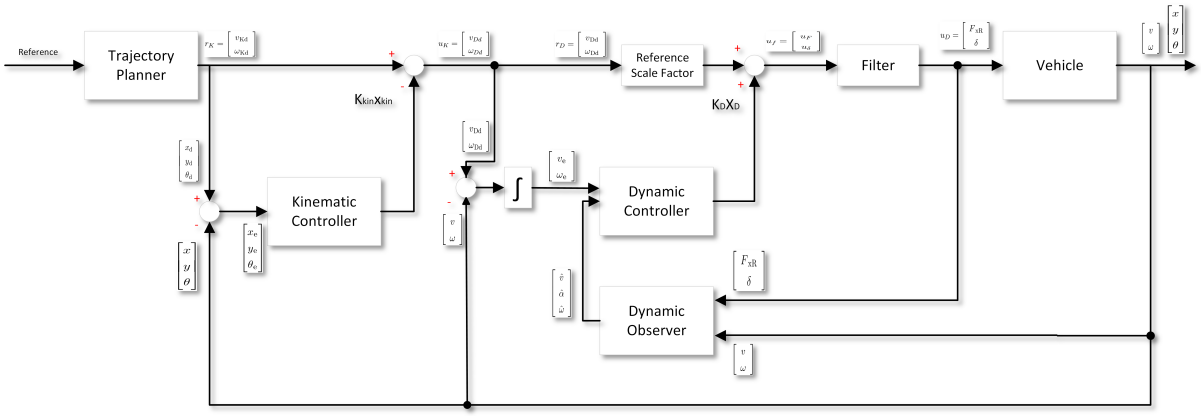

Figure 6.6: Cascade Control Block Diagram

In order to simulate the Cascade Control system the desired values have to be arranged. These values (references) are processed in a trajectory planner, which is in charge of generating the feasible trajectory and velocities references. The process takes place in an offline mode, before the control algorithm starts. In the next step, the planner provides the references of position and orientation, from which the real values are subtracted, and the result (error) is the input to the kinematic controller. The kinematic controller in combination with the reference of the velocities from the trajectory planner provides the references (linear and angular velocity) to the dynamic controller and to the reference scale factor. The combination of the dynamic controller and the reference scale factor gives the input (force and steering angle) to a filter, from which it is obtained the input (force and steering angle filtered) to control the vehicle.

The output of the vehicle is its current position, orientation and velocities. As it is noted in previous chapters, there is a state that cannot be measured, the slip angle α. For this reason the measured velocities v and ω are provided to the observer, alongside with the vehicle input, so as all the states to be estimated. The observer provides the estimated states to the dynamic controller, in parallel with the integration of the error between the desired velocities and real velocities of the vehicle.

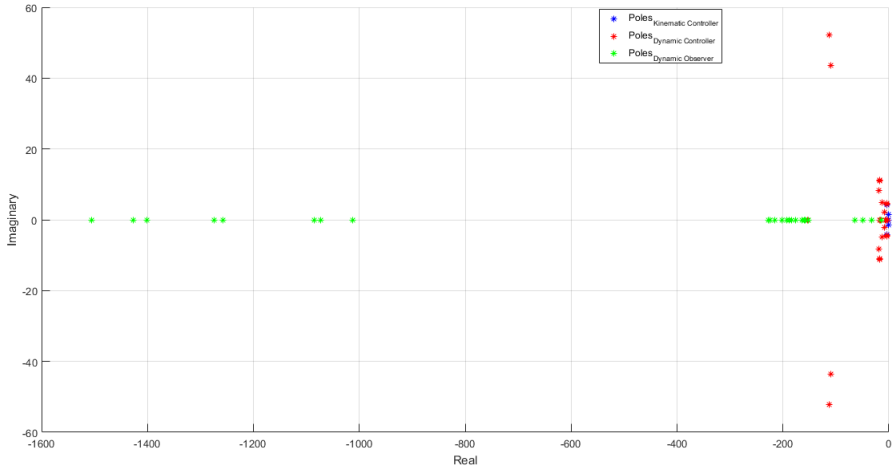

Figure 6.7: Cascade Control Poles

Remark 8 *The references for the Dynamic Controller are proceeded online in the Kinematic Controller.*

6.3.1 Kinematic and Dynamic Control Combination

In order to control the vehicle Kinematic and Dynamic behavior the combination of Kinematic and Dynamic Control loops is needed. The Q and R LMI parameters are taken from the previous simulations of each control scheme, selecting the best results as presented in Table 6.5. The simulation results are presented in Figures 6.8, 6.9, 6.10 and 6.11.

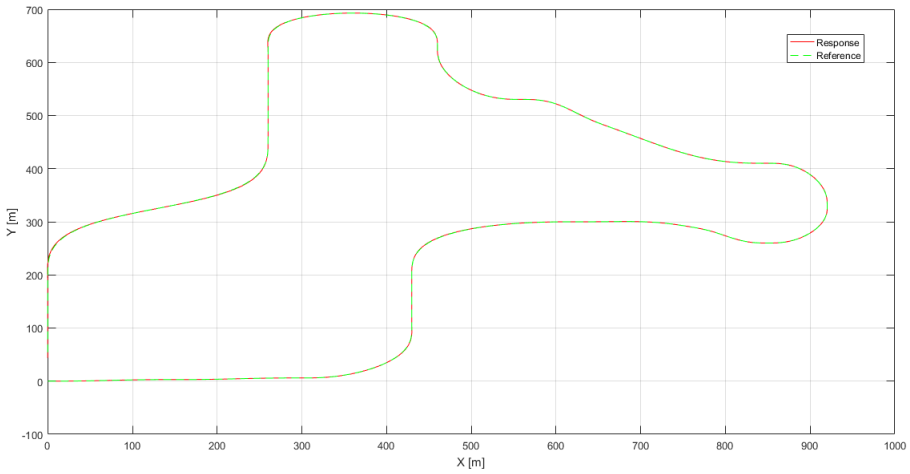

Figure 6.8: Kinematic and Dynamic Control - Trajectory of the Vehicle

As we can see in the above Figure 6.8, the trajectory of the the vehicle has miscellaneous features. However, the vehicle behavior is satisfactory, as it follows the planned trajectory without deviation. Although, in order to check more carefully its behavior, we have to check the errors of the positions and orientation in the next Figure 6.9.

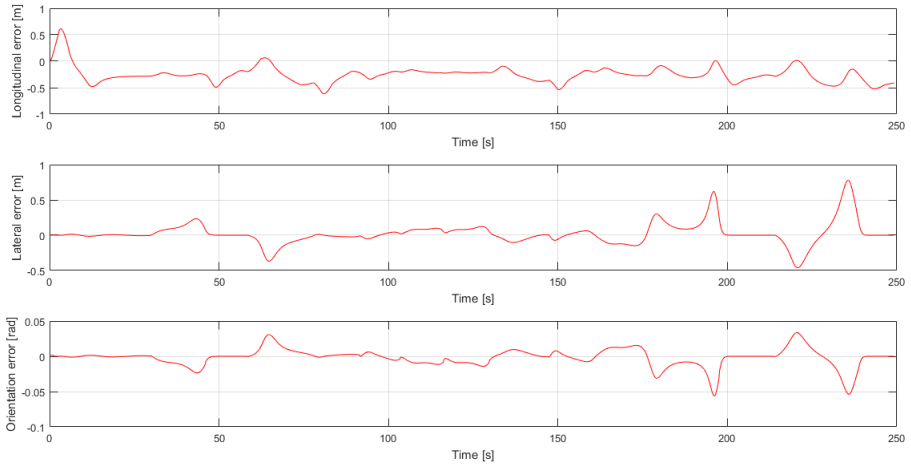

Figure 6.9: Kinematic and Dynamic Control - Position and Orientation Errors

We can observe from the above Figure 6.9, that we have some small errors in the longitudinal

and lateral position, and orientation of the vehicle. The error regarding the longitudinal position is small and it does not affect significantly the vehicle behavior. On the other hand, as we already know the lateral error is more significant as it keeps the vehicle to a straight line. It is observed that the lateral error presents some raise at particular times. These time instants correspond to the case when the vehicle increases its angular velocity suddenly. This fact can be checked in next Figure 6.10. The orientation error depends on angular velocity changes as well, since as one can observe, has similar shape with the lateral error plot.

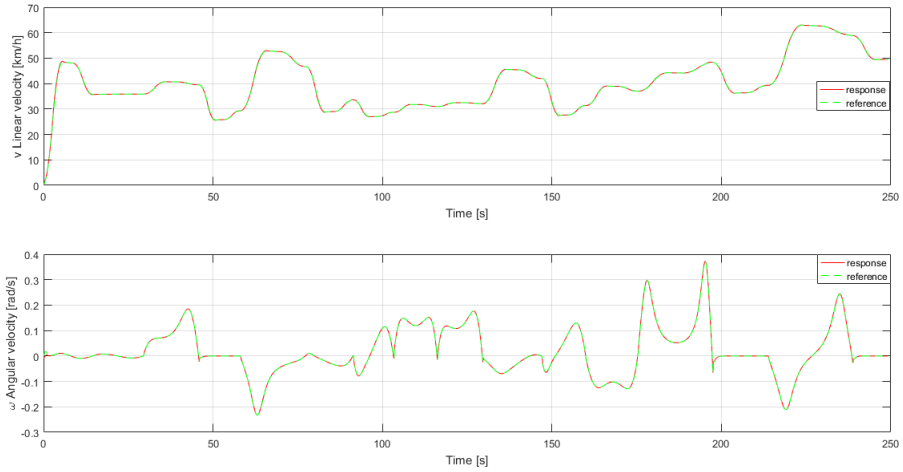

Figure 6.10: Kinematic and Dynamic Control - Linear and Angular Velocities States

The evolution of the linear and angular velocities states is presented in the above Figure 6.10. We can realize that both velocities follow the references in a very good manner. As we can see there is no deviation from the desired values except from the first moment, where the angular velocity has a very small insignificant deviation from the reference. It is normal and it can be explained as in this simulation the vehicle is arranged to start its movement from zero state of the velocities, and it needs more force in order to start the movement. This conclusion can be confirmed in next Figure 6.11.

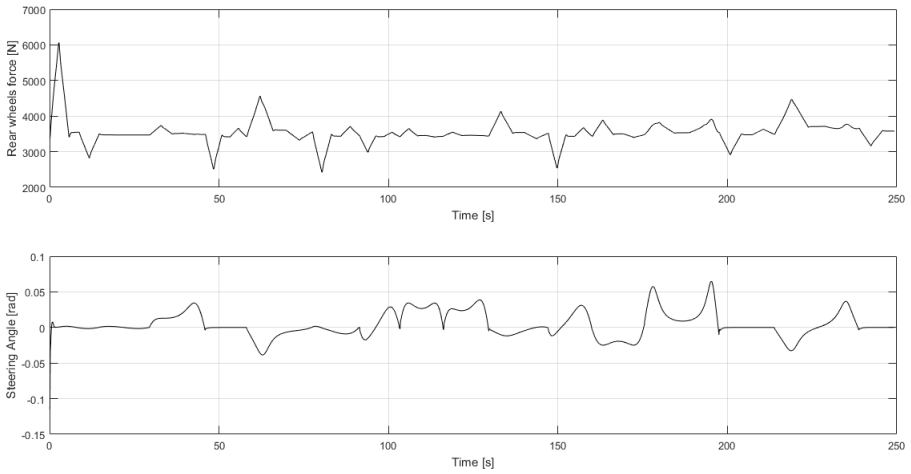

Figure 6.11: Kinematic and Dynamic Control - Inputs of the Vehicle

In order to control the vehicle behavior two input variables are needed, the force for the rear wheels and the steering angle of the front wheels. We can observe the evolution of these inputs in the above Figure 6.11. As we can see the force depends on the desired values of linear velocity. When we need to accelerate the vehicle, it is needed to apply force to the rear wheels abruptly, while when we have to decelerate it, the force has to be reduced suddenly. As it is logical, if the vehicle has to follow some specific values of velocity, we have to apply some force or reduce this force.

$Q_{controllerK}$	$R_{controllerK}$	v_{RMSE}	ω_{RMSE}	X_e	Y_e	θ_e
$\begin{bmatrix} 0.1 & 0 & 0 \\ 0 & 0.1 & 0 \\ 0 & 0 & 0.1 \end{bmatrix}$	$\begin{bmatrix} 0.1 & 0 \\ 0 & 0.1 \end{bmatrix}$					
$Q_{controllerD}$	$R_{controllerD}$	**0.0336**	**0.0011**	**0.2988**	**0.1569**	**0.0129**
$\begin{bmatrix} 0.05 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.01 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.01 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.01 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 100000 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 100000 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$	$\begin{bmatrix} 0.01 & 0 \\ 0 & 100 \end{bmatrix}$					

Table 6.5: Kinematic and Dynamic Control RMSE

Generally, the behavior of the vehicle can be judged by the RMSEs which depicted in the above Table 6.5. As we can see the velocities, positions and orientation errors dot not have big values which can affect the vehicle behavior. Although, following a more complex trajectory, the vehicle behavior could be affected.

6.3.2 Complete Control Scheme

As it is analyzed in this thesis, the employment of a dynamic observer is necessary, as there is a state that it cannot be measured. This state, the slip angle, can affect the behavior of a vehicle significantly, if it has large values. For this reason, its value is bounded in the controller and observer loop, in order not to obtain undesired values. Providing the observer estimation of the states to the controller, and using the implementation of the previous Section 6.3.1, the complete cascade control scheme is built. The Q and R LMI parameters, for the controllers are the same as in Section 6.3.1, and for the observer are taken from the simulation in Section 6.2.3, selecting the best results as presented in Table 6.6. The results of this experiment are presented in Figures 6.12, 6.13, 6.14, 6.15.

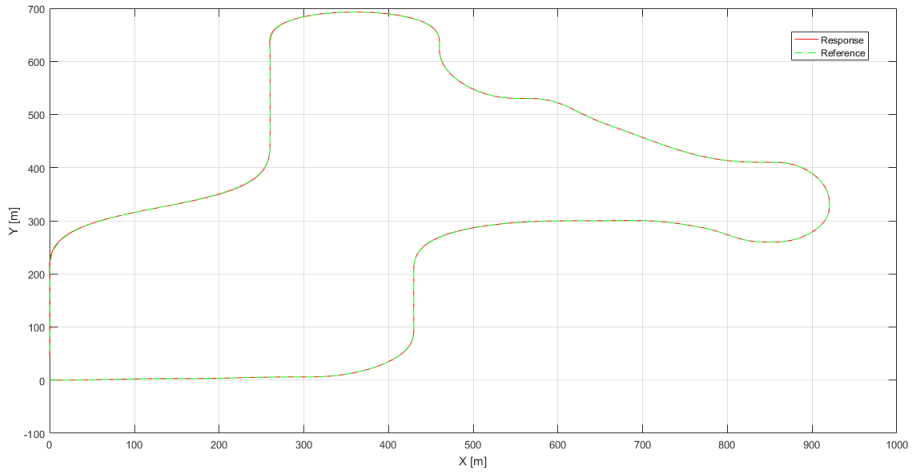

Figure 6.12: Complete Control Scheme - Trajectory of the Vehicle

As it is observed in the above Figure 6.12, the vehicle follows the desired trajectory in a satisfactory way, without deviating from the references. The results seem to be the same as in Section 6.3.1.

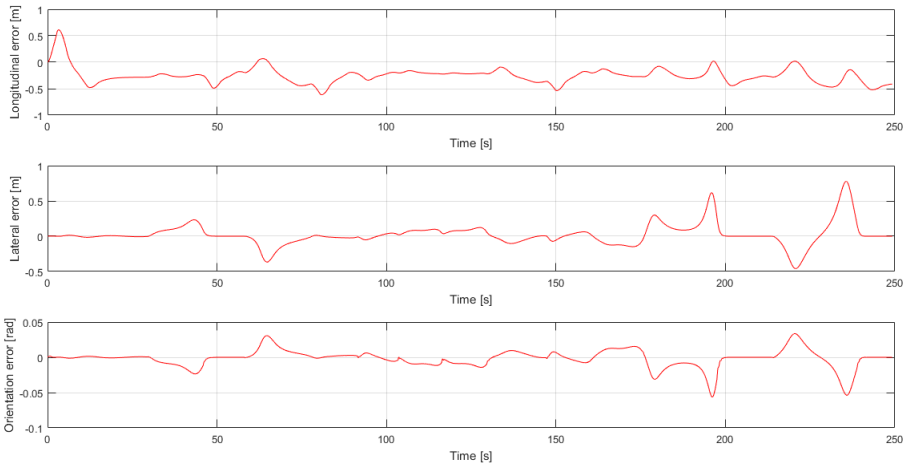

Figure 6.13: Complete Control Scheme - Position and Orientation Errors

The behavior of the vehicle can be judged in a better way by checking the position and orientation errors in the above Figure 6.13. We can see that the vehicle operates in a same way as before we add the observer. Specifically, we can see that the lateral and orientation errors have some increases when the vehicle steers, despite the good path tracking observed in Figure 6.12.

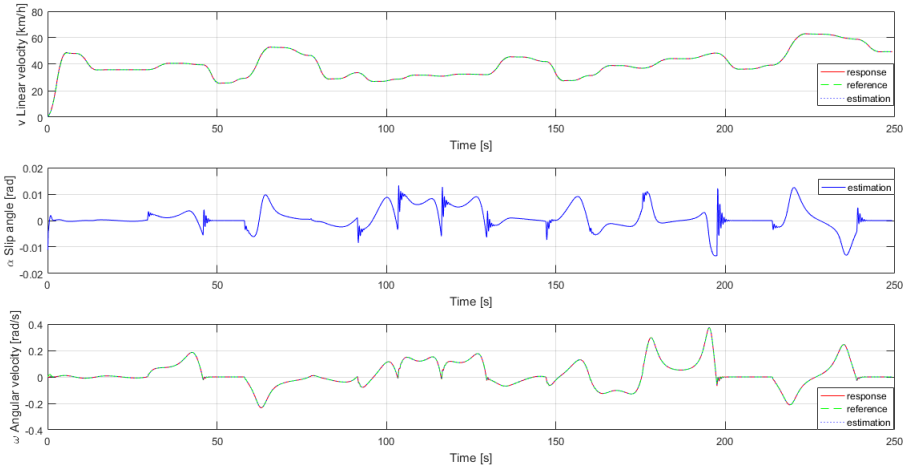

Figure 6.14: Complete Control Scheme - Dynamic States

As in this experiment we take into account the slip angle state, the vehicle has a slightly different behavior comparing with the experiment without the observer. We can observe it in the above Figure 6.14, and specifically in the points that the slip angle is estimated with some oscillations. This behavior appears when the velocities change abruptly. As we can see, the response of angular velocity at these time instants, presents some very small oscillations, as the controller tries to compensate the slip angle intense occurrence, and to prevent the vehicle from slipping away. On the other hand, we can see that the estimation of linear and angular velocity states is almost perfect, as they follow the references without deviating.

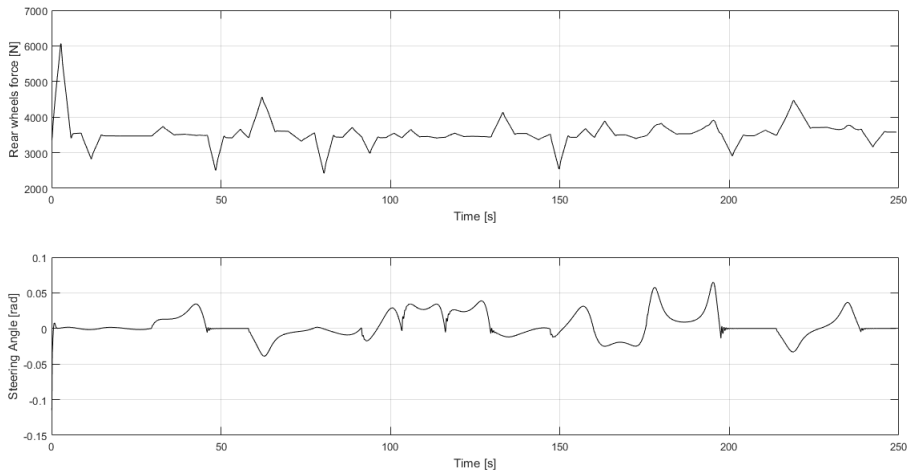

Figure 6.15: Complete Control Scheme - Inputs of the Vehicle

The previous consideration, that the controller tries to compensate the undesirable values of slip angle, can be observed in the above Figure 6.15. The compensation is achieved by the steering angle input of the vehicle. The oscillation of the steering angle, in the points that the angular velocity changes abruptly can confirm the previous ascertainment.

$Q_{controllerK}$	$R_{controllerK}$	v_{RMSE}	ω_{RMSE}	X_e	Y_e	θ_e
$\begin{bmatrix} 0.1 & 0 & 0 \\ 0 & 0.1 & 0 \\ 0 & 0 & 0.1 \end{bmatrix}$	$\begin{bmatrix} 0.1 & 0 \\ 0 & 0.1 \end{bmatrix}$	**0.0338**	**0.0014**	**0.2981**	**0.1313**	**0.0128**
$Q_{controllerD}$	$R_{controllerD}$					
$\begin{bmatrix} 0.05 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.01 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.01 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.01 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 100000 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 100000 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$	$\begin{bmatrix} 0.01 & 0 \\ 0 & 100 \end{bmatrix}$					
$Q_{observerD}$	$R_{observerD}$					
$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$	$\begin{bmatrix} 1 & 0 \\ 0 & 0.005 \end{bmatrix}$					

Table 6.6: Complete Control Scheme - RMSE

The RMSEs of the vehicle states are shown in the above Table 6.6. As we can see the errors are almost the same as before the state estimation was added (Table 6.5). Consequently, one could say that the vehicle operates in the same way with or without the observer. Although it is not true, as the use of the observer can prevent undesirable values of the unmeasured states and in the real world it is very important.

6.4 Fault Tolerant Control Simulation

The simulation of the FTC, for longitudinal and lateral actuator faults, takes place in this section. The experiments are carried out by using miscellaneous fault estimation techniques. In the first experiment the Unknown Input Observer method is used to estimate the longitudinal and lateral actuator faults, in a separate manner. Subsequently, the Augmented State Observer technique is employed, in order to estimate the same faults. Afterwards, the Parameter Estimation method is used by employing the Least Squares technique, so as to estimate the faults. Thereafter, the combination of the UIO and LSPE is employed so as to compensate the disturbances and faults. Eventually, the combination of UIO and LSPE methods is used, so as to estimate both faults concurrently.

6.4.1 Fault Estimation with Unknown Input Observer

In this section, the combined Kinematic and Dynamic Controllers are employed, along with the Unknown Input Observer. The implementation of the Cascade Control scheme from previous section (Fig. 6.6) is used, and with some minor alterations the Fault Tolerant Control scheme is implemented.

Longitudinal Actuator Fault

Subsequently, the Fault Tolerant Control scheme, taking into account the longitudinal actuator fault ($F_{xRfault}$) is simulated, as it is shown in Fig.6.16

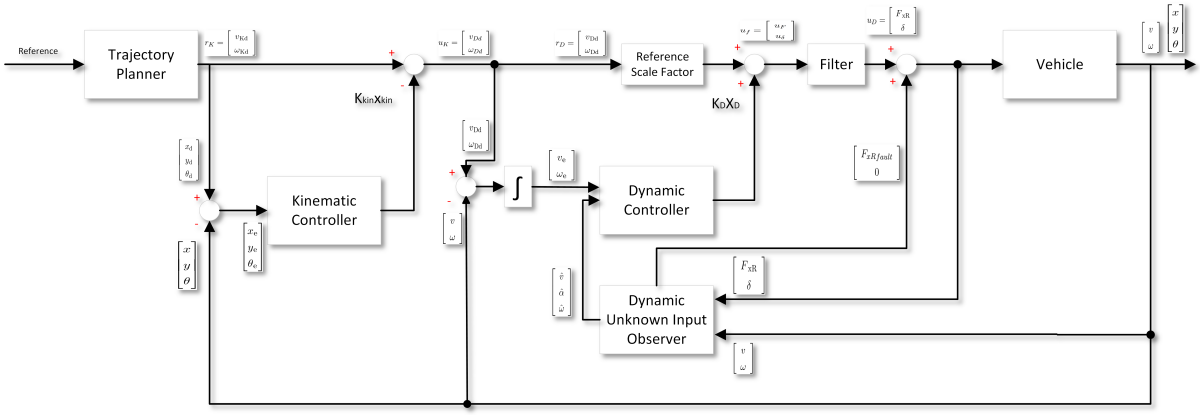

Figure 6.16: FTC with UIO - Block Diagram

Similarly to the previous experiments, Dynamic Controller has to be faster than the Kinematic Controller, and the Unknown Input Observer has to be faster than the Dynamic Controller. It can be verified by checking the poles of the system in the next Figure 6.17.

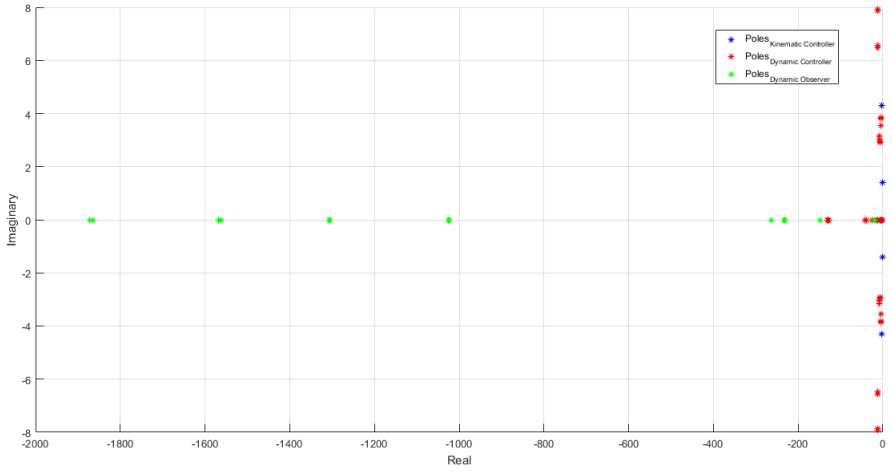

Figure 6.17: Longitudinal FTC with UIO - Poles

As it has previously analyzed, the employment of an UIO is necessary, so as to estimate the faults and disturbances, however it is used at the same time to estimate the states of the system. Apart from providing the observer estimation of the states, to the controller, the actuator fault estimation is used in order to compensate the fault. The best chosen LMI parameters Q and R for the controllers and the observer are depicted in Table 6.7. The results of this experiment are presented in Figures 6.18, 6.19, 6.20 , 6.21 and 6.22.

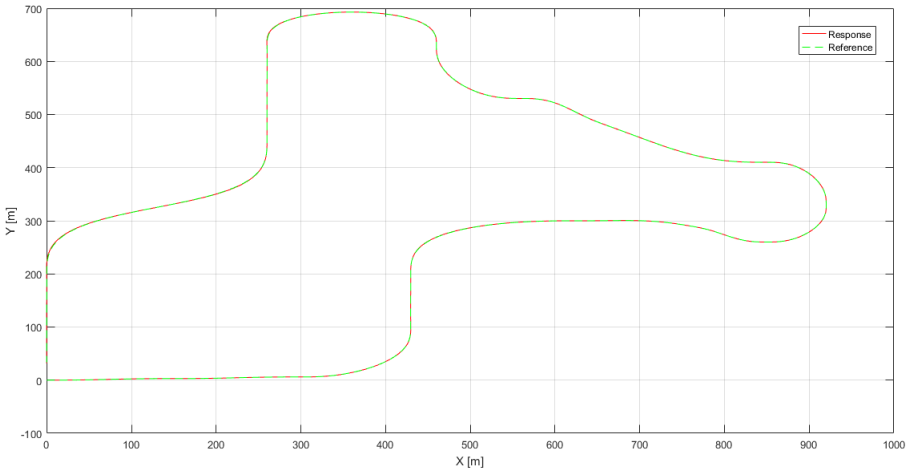

Figure 6.18: Longitudinal FTC with UIO - Trajectory of the Vehicle

As it is observed in the above Figure 6.18, the vehicle behaves in an ideal manner, as it follows the position and orientation references, despite the existence of actuator faults.

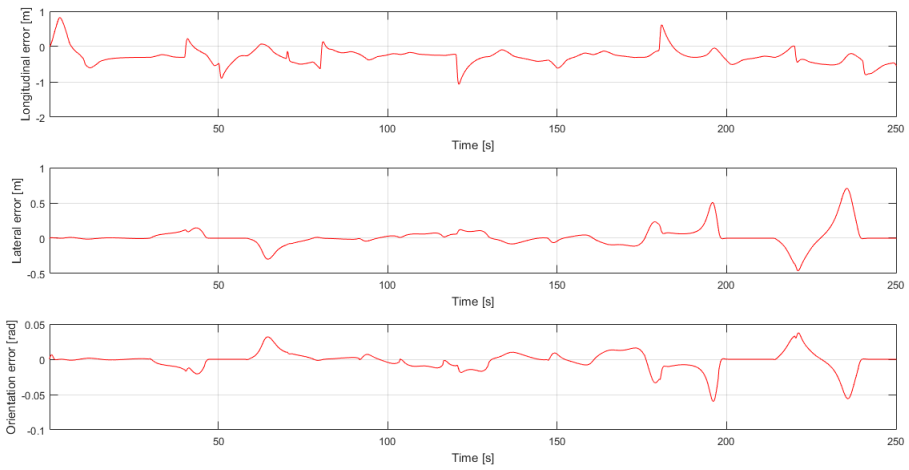

Figure 6.19: Longitudinal FTC with UIO - Position and Orientation Errors

It can be seen in the above Figure 6.19, that the lateral and orientation errors have some increases when the vehicle steers. This observation is the same as in previous experiment (eq. 6.13), having no actuator faults. Although, we can see that the longitudinal error, has some significant raise, comparing it with the experiment without the actuator faults, despite the good path tracking observed in Figure 6.18.

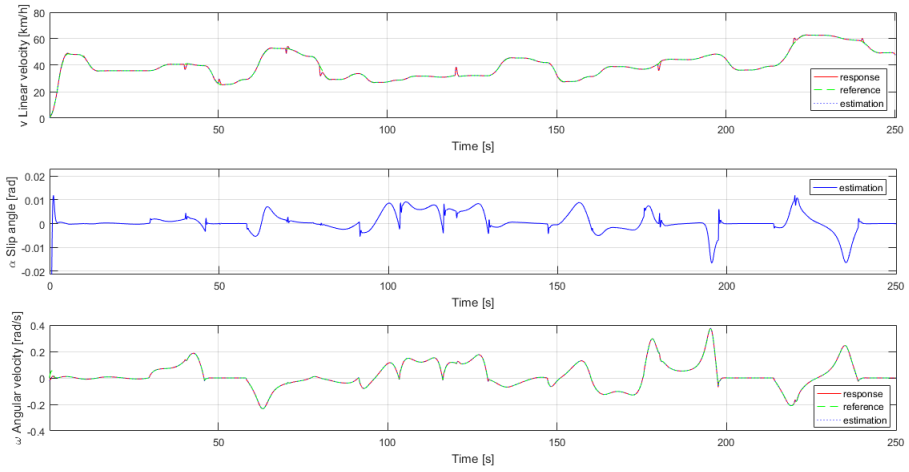

Figure 6.20: Longitudinal FTC with UIO - States of the Vehicle

In the previous Figure 6.20, we can realize the incidents when the actuator faults affect the vehicle behavior. Specifically, the faults affect both the linear and angular velocities of the vehicle, as well as the slip angle. The linear velocity is affected whenever the condition of the fault changes abruptly, and makes its response to depart from the reference. On the other hand, the angular velocity is affected at those incidents but only if the angular velocity at this time is different from zero, although it does not depart from its reference. Additionally, we can see that the slip angle is affected at these moments. However, the overall behavior of the vehicle is quite satisfactory, and only in the linear velocity plot we observe the departure from the reference, but it returns to the desired values very fast. The incidents, when the faults appear, can be seen more clearly in the following Figures 6.21 and 6.22.

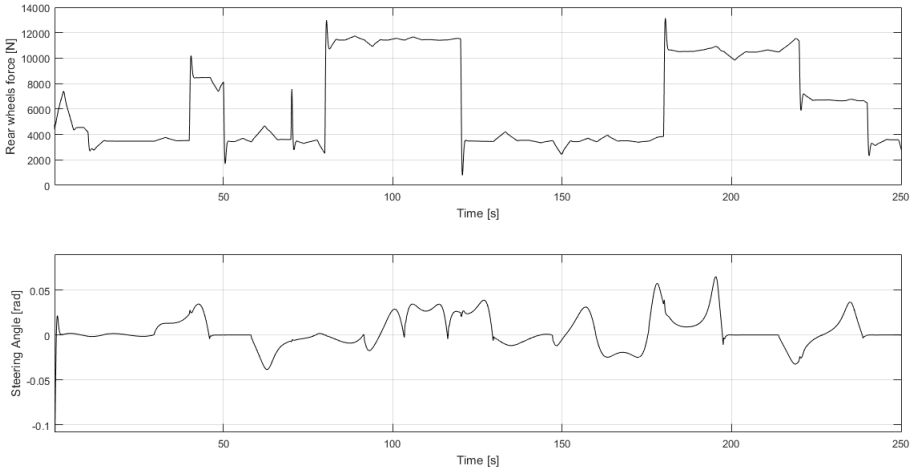

Figure 6.21: Longitudinal FTC with UIO - Inputs of the Vehicle

In the previous Figure 6.21, we can observe the behavior of the longitudinal and lateral actuators. Comparing the longitudinal force with the experiment where there is no actuator fault in Figure 6.15, we can realize that the force fault is compensated. It means that the estimation of the fault and consequently its compensation is done in an quite effective way. The latter consideration can be confirmed, observing the next Figure 6.22.

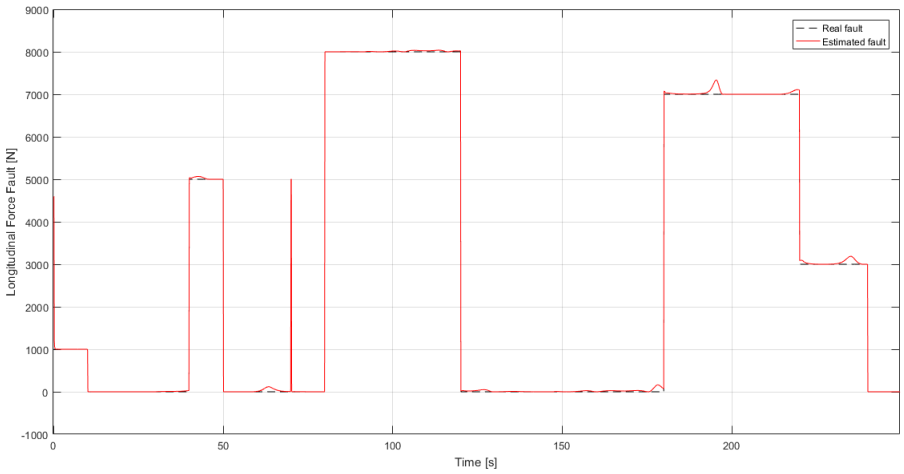

Figure 6.22: Longitudinal Actuator Fault Estimation with UIO

In the above Figure 6.22, the estimation of the longitudinal actuator fault is depicted. As we can see, except for some minor incidents, the estimation of the fault is done in a satisfactory way, following the value of the real fault.

$Q_{controllerK}$	$R_{controllerK}$	v_{RMSE}	ω_{RMSE}	X_e	Y_e	θ_e
$\begin{bmatrix} 0.1 & 0 & 0 \\ 0 & 0.1 & 0 \\ 0 & 0 & 0.1 \end{bmatrix}$	$\begin{bmatrix} 0.1 & 0 \\ 0 & 0.1 \end{bmatrix}$	**0.1555**	**0.0031**	**0.3584**	**0.1320**	**0.0132**
$Q_{controllerD}$	$R_{controllerD}$					
$\begin{bmatrix} 0.05 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.01 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.01 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.01 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 100000 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 100000 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$	$\begin{bmatrix} 0.01 & 0 \\ 0 & 100 \end{bmatrix}$					
Q_{UIO}	R_{UIO}					
$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$	$\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$					

Table 6.7: Longitudinal FTC with UIO - RMSE

The RMSEs of the vehicle states are shown in the above Table 6.7. We can confirm the previous considerations by comparing these results with the experiment without the presence of actuator faults (Table 6.6). As it can be seen, the only state of the vehicle which is affected significantly, is the linear velocity. Specifically, in the case without the occurrence of fault the linear velocity RMSE is equal to 0.0338m, while with the existence of fault the RMSE is equal to 0.155m. Eventually, one could say that the vehicle operates reliably, despite the existence

of the actuator fault and its effects.

Lateral Actuator Fault

Similarly to the previous experiment (see Section 6.4.1), the UIO is used for the fault estimation. The lateral actuator fault (δ_{fault}) is considered in this experiment. In order to compensate the fault to the controlled system, it is needed to have satisfactory estimation of it. Although, in this specific case, where the fault (δ_{fault}) is estimated by the UIO, some issues have arisen. Specifically, the system presents stability problems, since the UIO has two positive poles. It can be verified in the following Figure 6.24, where the poles of the controllers and the observer are depicted. It is needed to be mentioned that this result is the best obtained, as in the rest of the experiments the system had greater number of positive poles with much faster dynamics.

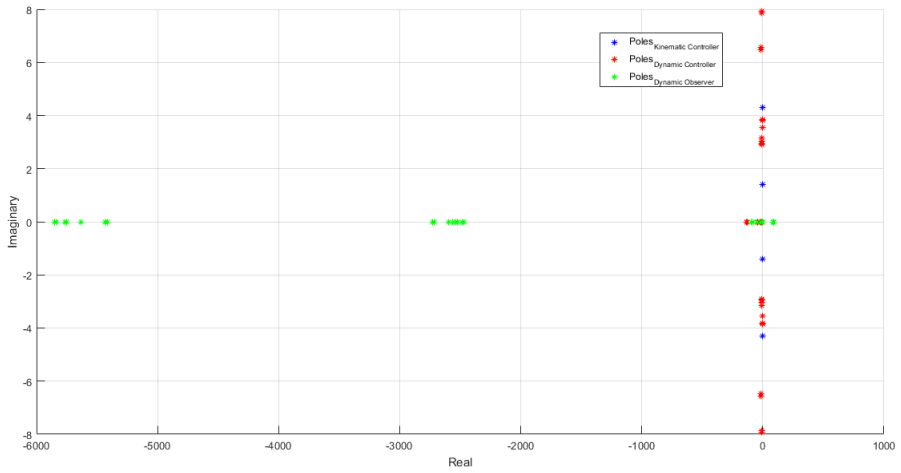

Figure 6.23: Lateral FTC with UIO - Poles

Obviously, the main reason for which a stable system cannot be obtained in this experiment, is that there is a state (α) of the vehicle, that is not measurable. In order to confirm this assumption, an experiment in which the state α is arranged to be measured is carried out, despite the fact that in the real world it cannot be measured. The results from this experiment are depicted in the next Figures 6.24, 6.25 and 6.26.

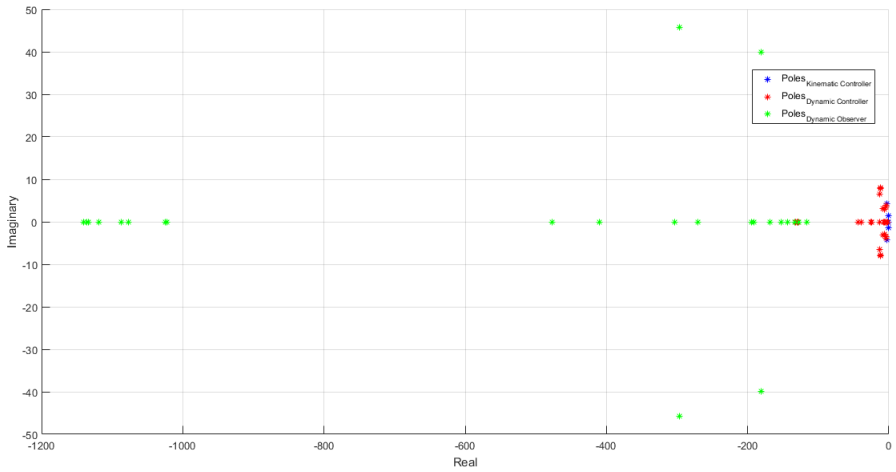

Figure 6.24: Lateral FTC with UIO (State α Measured) - Poles

As it can be seen in Figure 6.24, the poles of the system are negative so it is stable, and the control system dynamics are properly arranged as it is discussed in Section 6.3.

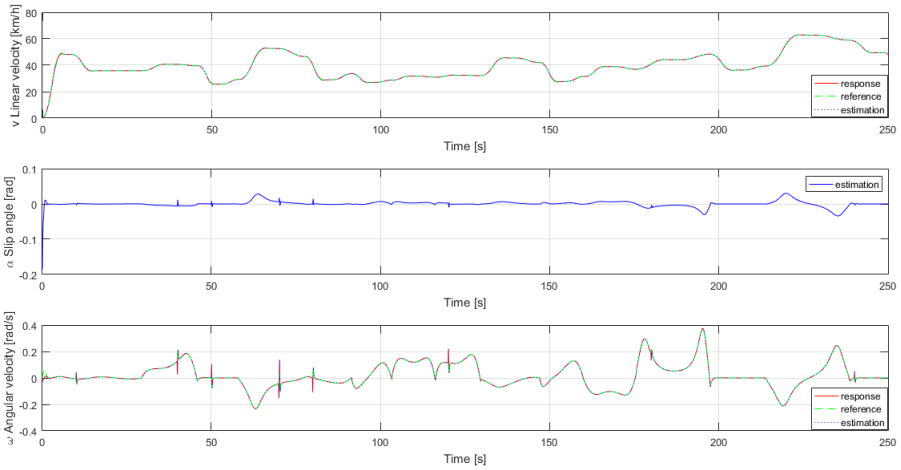

Figure 6.25: Lateral FTC with UIO (State α Measured) - States of the Vehicle

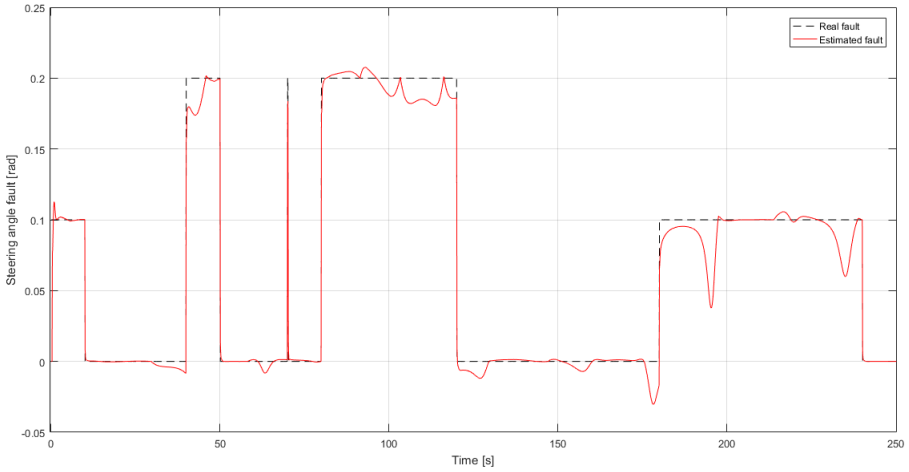

Figure 6.26: Lateral Actuator Fault Estimation with UIO (State α Measured)

As we can see in Figures 6.25 and 6.26, the compensation of the lateral fault, as well as the states estimation, present satisfactory results. The states of the system depart from the references, only in the cases the lateral error appears. The fault is compensated quickly even if we take into account the fact, that the lateral actuator dynamics are very fast. The effect of the fast actuator dynamics can be confirmed by checking the Figure 6.26. As we can see the fault estimation has satisfactory performance. However there are some incidents where it departs from the real fault value. It can been verified, if we observe the Figure 6.25 and specifically the angular velocity plot. We can ascertain that when there are abrupt or large value changes of the angular velocity, the estimation of the lateral actuator fault is affected significantly without compensating very fast. This is the confirmation of the previous assumption that the fault estimation is affected by the fast dynamics of the lateral actuator.

6.4.2 Fault Estimation with Augmented State Observer

Despite the good results, from the experiment where the state α is measurable, it is required the usage of another method which takes into account the real characteristics of the controlled system, without the α being measured. Hence in this section, the ASO fault estimation technique is employed. Specifically, the actuator fault is arranged to be a state of the system and it is estimated via the ASO. It has been tried to obtain stable controller scheme, with the proper dynamics. Although, there were some issues to obtain such a control scheme and basically the

issue arose with the observer dynamics. In the following figures 6.27 and 6.28 the poles of the system with longitudinal and lateral faults are depicted.

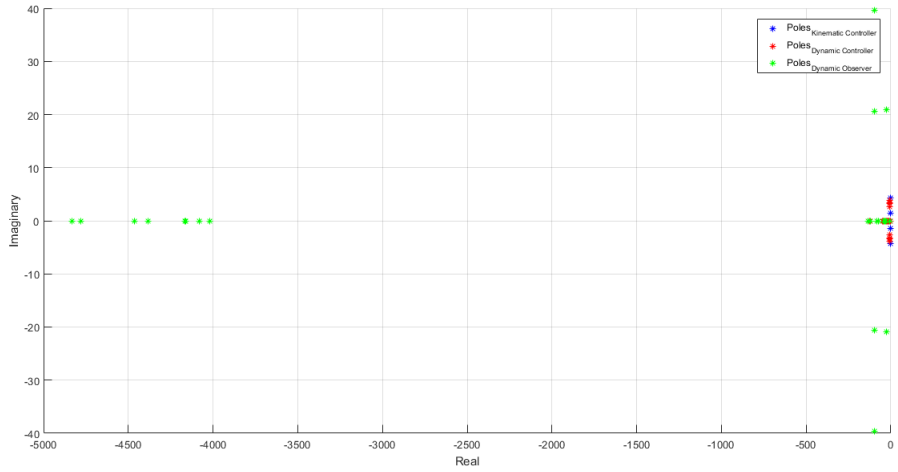

Figure 6.27: Longitudinal FTC with Augmented State Observer - Poles

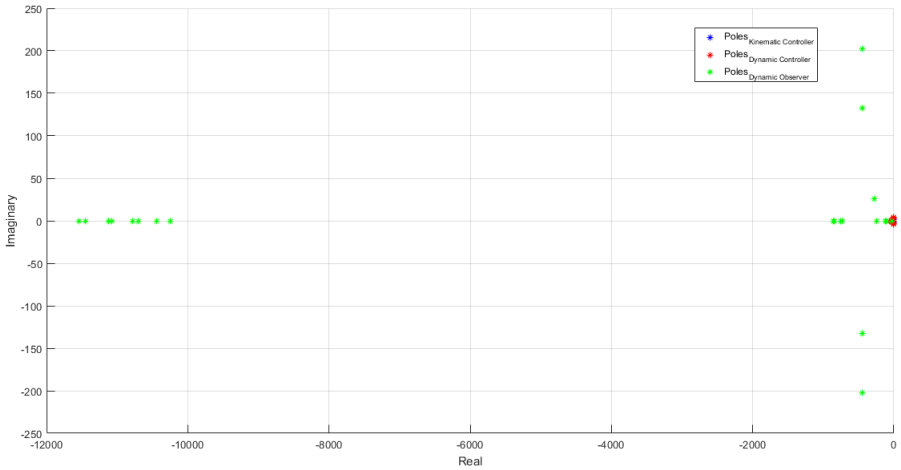

Figure 6.28: Lateral FTC with Augmented State Observer - Poles

As we can see, the poles of the observer in both cases present some fast dynamics. It is needed to mention that, the above results are the best among all the trials that have been done, since the rest of the experiments had much bigger pole values or even positive poles. For this

reason, the system had some singularities and it could not be simulated. Additionally, some experiments with the state α to be measurable have been carried out. Although, the results were similar with the former ones. It means that this method (Augmented State Observer), arranging the faults as states, is not suitable for the specific vehicle system.

6.4.3 Fault Estimation with Least Squares Parameter Estimation

In this experiment the, LSPE technique is used in combination with the system simulated in Section 6.3.2. This method is different from the previous ones. In order to estimate the faults, it is taken into account that the actuator faults affect the system parameters. The procedure starts from the hypothesis that it is possible to find a state of the system that is directly influenced by the faulty actuator. Consequently, the faults can be estimated in different ways by using the system equations. Specifically, in our case, for the longitudinal fault there are two different formulas that affect the states v and α respectively, and for lateral fault there are three different formulas that affect states v, α and ω. During the experimental procedure, all these formulas have been used to estimate the faults, as part of the LSPE method. These different cases have been tested whether they are suitable to be used for the faults estimation. Although, only one of them it was used, the case where the lateral fault affects the state ω. It has been ascertained, that for all the other cases, the system has singularities which make the simulation to shut down. In order to explain why is this happening, we have to check the (eq.4.11)-(eq.4.12) and to realize that the cases that have singularities, depend on the state α which cannot be measured. The following Figures 6.29 and 6.30 present the states with the lateral fault (δ_{fault}) compensated, as well as the fault estimation by the LSPE.

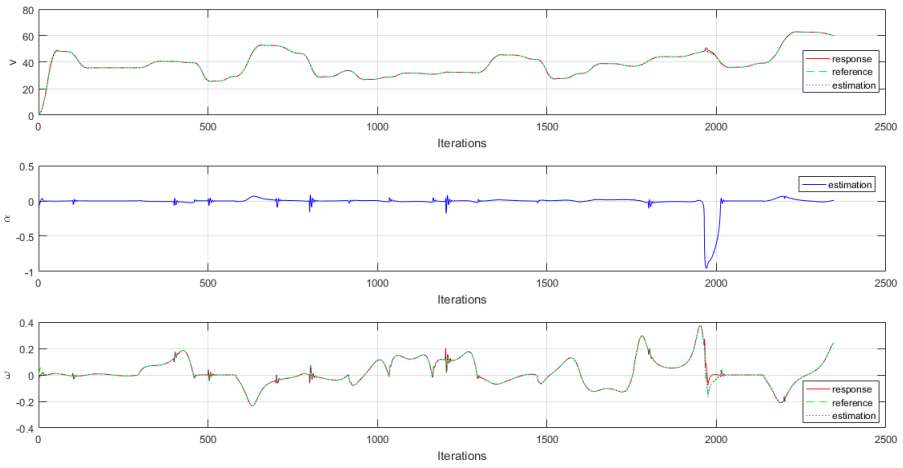

Figure 6.29: Lateral FTC with LSPE - States of the Vehicle

As we can see in the above figure 6.29, the lateral fault is compensated in a effective way in state ω which is mainly affected. Although, we can observe that there are some oscillations before the fault is compensated completely. These oscillations are justified similarly to the previous experiments, because of the fast dynamics of the lateral actuator. This conjecture can be confirmed by checking the Figure 6.30 as well. We can observe in the fault estimation plot, that the oscillations occur at the same time with the oscillations of the states (Figure 6.29), which depend on the fast lateral actuator dynamics in the presence of the fault. Additionally there are some more oscillations which depend on the abrupt change of state ω. Although, if someone observe carefully, they can see an abrupt big change of state α and the estimated fault approximately at iteration 1950, as well as change to the value of states v and ω at the same time. Additionally, they can see a shut down of the simulation around the iteration 2350 (in both figures). After careful consideration, it has been ascertained that even if the lateral actuator fault is equal to zero, we have the same incidents at the same time instants. Investigating the reason why we have these, it has been found that they depend on some disturbances that come from the longitudinal actuator. This ascertainment is analyzed and depicted in the next Section 6.4.4.

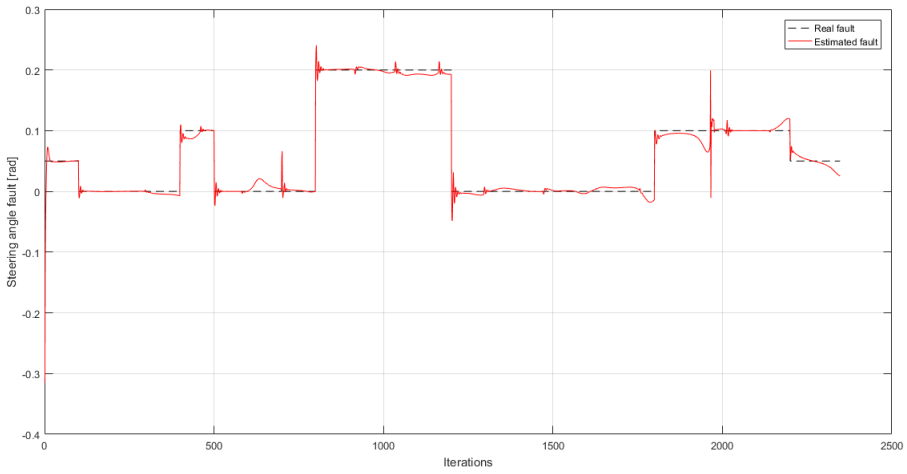

Figure 6.30: Lateral Actuator Fault Estimation with LSPE

6.4.4 Complete Fault Tolerant Control Scheme

FTC for Lateral Actuator Fault and Longitudinal Actuator Disturbances

In this section, the lateral actuator fault (δ_{fault}) estimation as well as the longitudinal disturbance estimation are used, in order to compensate the lateral actuator fault to the system. In order to perform this experiment, the UIO control scheme simulated in Section 6.4.1 (Figure 6.16), combined with the LSPE method are employed, and the new control scheme is depicted in Figure 6.31. In order to detect the disturbances that come from the longitudinal actuator, the same UIO which is used for the longitudinal actuator fault ($F_{xRfault}$) is employed. The only difference from the simulation in Section 6.4.1 is that the real longitudinal actuator fault is arranged to be equal to zero. The results of this experiment, are presented in the following Figures 6.32, 6.33, 6.34, 6.35, 6.36 and 6.37.

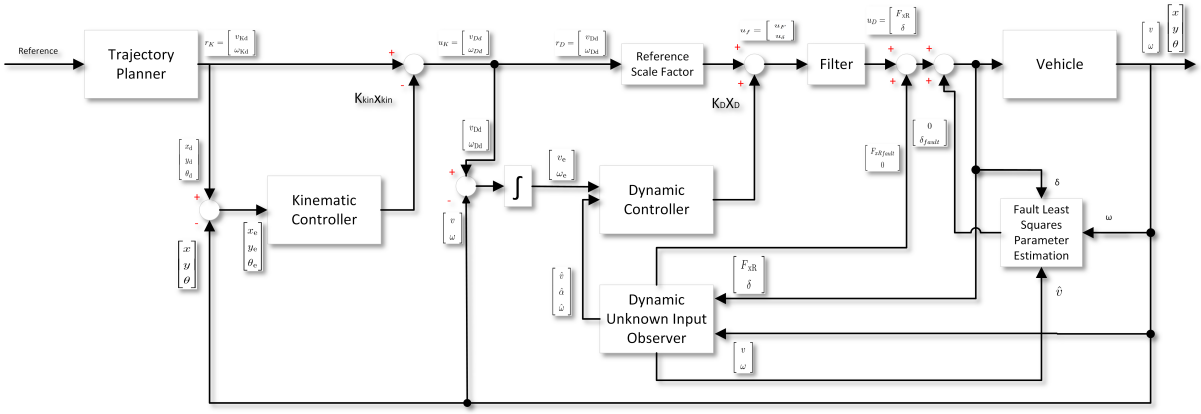

Figure 6.31: FTC with LSPE and UIO - Block Diagram

As it is analyzed in Section 6.4.3, in order to compensate the lateral actuator fault, it is needed to estimate the longitudinal actuator disturbances through the UIO, and simultaneously to estimate lateral actuator fault via the LSPE. The result of the disturbance estimation is presented in the following Figure 6.32.

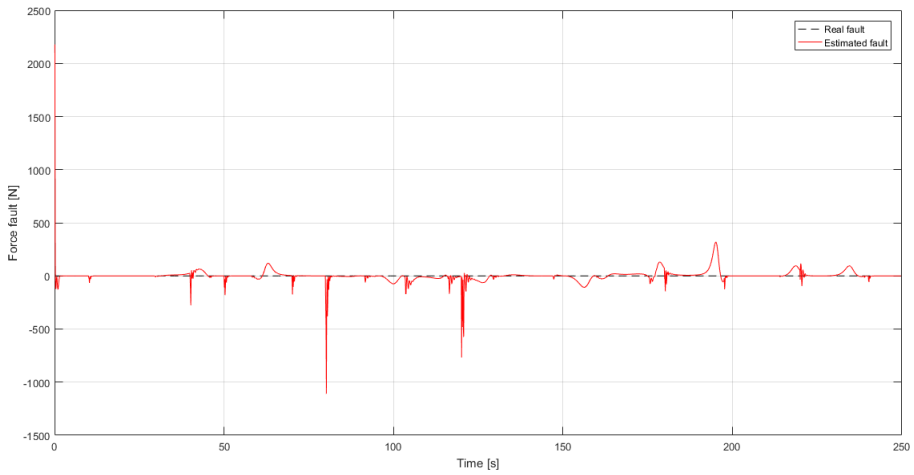

Figure 6.32: Longitudinal Actuator Disturbance Estimation with UIO

We can observe that the longitudinal actuator disturbance incidents occur when there are some abrupt changes to the lateral actuator fault (Figure 6.33), as well as when the states v and ω have abrupt changes (Figure 6.34). Compensating the longitudinal actuator disturbance

to the system, the lateral actuator fault estimation is done more effectively, and it is presented in the next Figure 6.33.

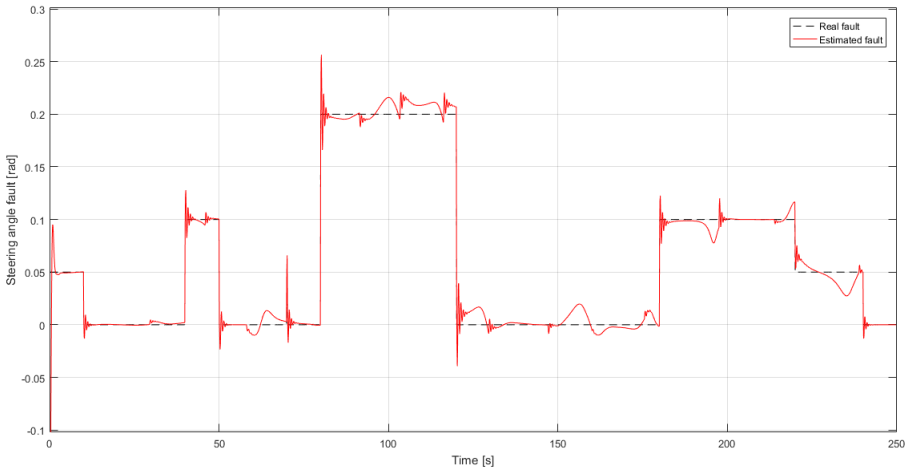

Figure 6.33: Lateral Actuator Fault Estimation with LSPE

As we can observe (Figure 6.33), the lateral fault estimation, is quite satisfactory. The only problems (oscillations, overshoots) arise, when the the state ω have abrupt changes, as it can be seen in the next Figure 6.34, and it happens because of the fast dynamics of the lateral actuator. Additionally, the simulation does not shut down because of the singularities, as before (Section 6.4.3) and thus the experiment is performed for all the desired values of the position, orientation and velocities.

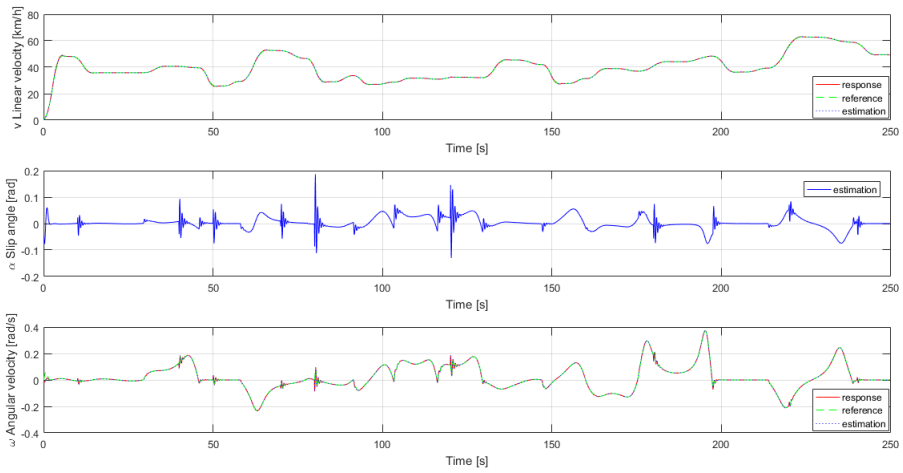

Figure 6.34: Lateral FTC with LSPE and UIO - States of the Vehicle

In the above Figure 6.34, we can observe that the response and the estimation of the states follow the references in a proper way. The only adverse issue is when the fault value changes and so some oscillations appear. However, these oscillations follow the generated references from the kinematic controller, in order to follow the desired position and orientation values and they are depreciated quite quickly. Another, important issue, is the increase of the slip angle value. It appears when there is change to the ω state, but it is not so significant as it is depreciated quickly and also, does not take big values (it is in the desired range [-0.1, 0.1]).

In order to evaluate the behavior of the vehicle in a better way, we can check the following Figures 6.35 and 6.36. The first figure depicts the desired trajectory of the vehicle and its real response, and the second one shows the errors of positions and orientation response with respect to the reference.

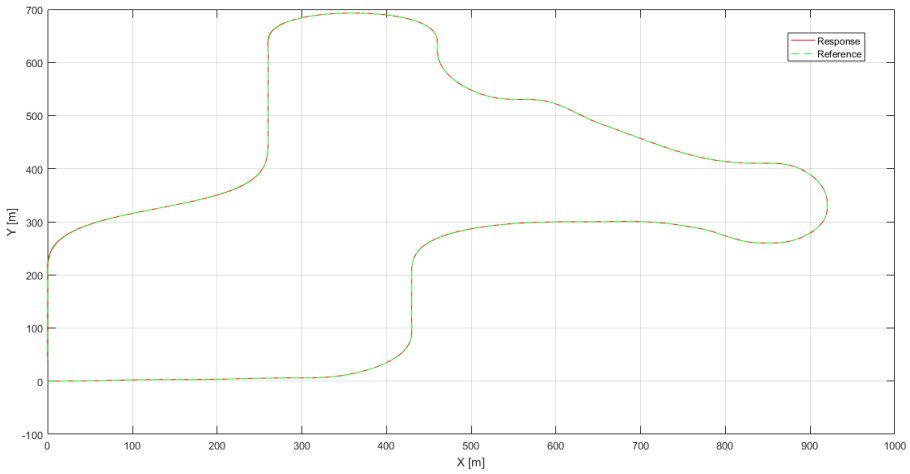

Figure 6.35: Lateral FTC with LSPE and UIO - Trajectory of the Vehicle

Observing the above Figure 6.36, one could say that the vehicle follows the trajectory reference properly. It seems that it does not depart from the desired positions. Although, it is not clear what happens with the orientation. For this reason we have to check the plot of the errors (see Figure6.36).

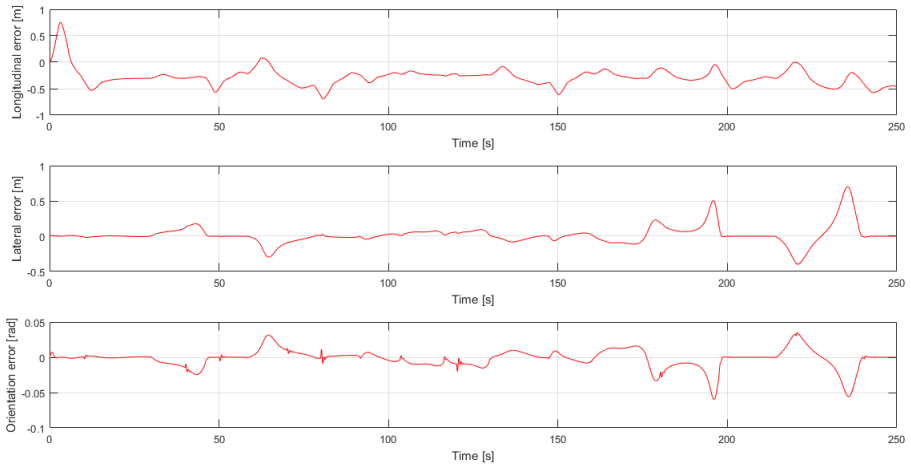

Figure 6.36: Lateral FTC with LSPE and UIO - Position and Orientation Errors

As we can see in Figure 6.36, the longitudinal position and orientation errors do not have

such values that affect the behavior of the vehicle. Although, the lateral position values in some cases have some values which are undesirable (almost 0.5 m) which could affect the behavior of a real vehicle to follow the desired trajectory. In order to understand better the errors effect to the vehicle behavior we have to check the root mean square value of these errors in Table 6.8.

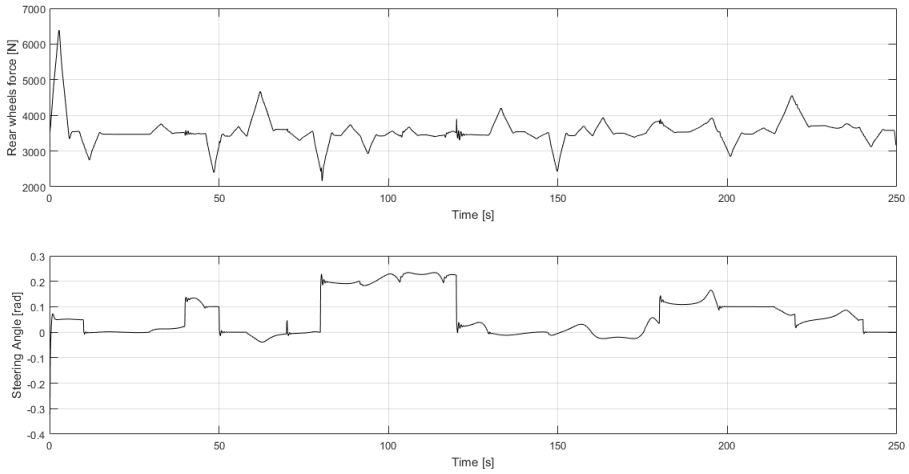

Figure 6.37: Lateral FTC with LSPE and UIO - Inputs of the Vehicle

In the above Figure 6.37 we can observe the values of the actuators during the experiment. We can see that the lateral actuator values change abruptly and have big values in some cases, which happens because of the actuator fault, and this is actually the way the fault is compensated.

$Q_{controllerK}$	$R_{controllerK}$	v_{RMSE}	ω_{RMSE}	X_e	Y_e	θ_e
$\begin{bmatrix} 0.1 & 0 & 0 \\ 0 & 0.1 & 0 \\ 0 & 0 & 0.1 \end{bmatrix}$	$\begin{bmatrix} 0.1 & 0 \\ 0 & 0.1 \end{bmatrix}$	**0.0441**	**0.0058**	**0.3287**	**0.1314**	**0.0132**
$Q_{controllerD}$	$R_{controllerD}$					
$\begin{bmatrix} 0.05 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.01 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.01 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.01 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 100000 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 100000 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$	$\begin{bmatrix} 0.01 & 0 \\ 0 & 100 \end{bmatrix}$					
Q_{UIO}	R_{UIO}					
$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$	$\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$					

Table 6.8: Lateral FTC with UIO and LSPE - RMSE

So as to understand better the errors with the influence of the actuator fault, we have to

compare the RMSE values of the previous Table 6.8 with the RMSE of the experiment in which the actuator fault does not exist (see Table 6.6). Through this comparison we can say that the results in both cases are almost the same, except a minor difference in the angular velocity error, which is 0.0044 rad/s bigger when there is lateral actuator fault. However, this minor difference does not affect in a significant manner the behavior of the vehicle.

FTC with Simultaneous Presence of Longitudinal and Lateral Actuator Faults

In this experiment, the system (vehicle) has faults to both its actuators (longitudinal, lateral). It has been arranged for both the faults to emerge at the same time instants. In order to detect and estimate the faults, the techniques that have been used in the previous experiment are employed. Specifically, exactly the same control scheme from Figure 6.31 is used. The only difference is that instead of setting zero longitudinal actuator fault, it is arranged to have the existence of it, in the same way as in Section 6.4.1 (see Figure 6.22). The lateral actuator fault, is arranged to have the same values as in Section 6.4.4 (see Figure 6.33). The results of this experiment, are presented in the following Figures 6.38, 6.39, 6.40, 6.41, 6.42 and 6.43.

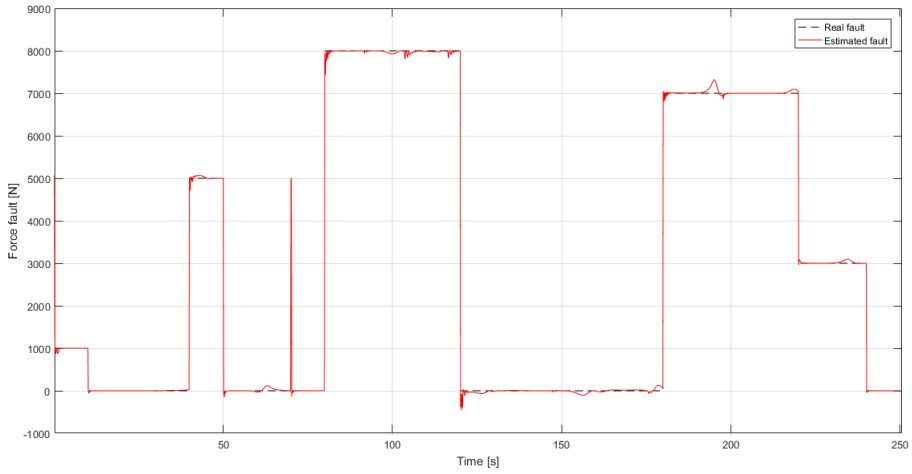

Figure 6.38: Complete FTC - Longitudinal Actuator Fault Estimation with UIO

In the above Figure 6.38, we can observe the longitudinal fault estimation ($F_{xRfault}$) using the UIO. Comparing it with the experiment where the lateral fault (δ_{fault}) is not taken into account (Figure 6.22), we can say that the estimation is the same except some tiny differences which are insignificant to affect the behavior of the vehicle. The existence of these tiny

oscillations depend on the presence of δ_{fault}.

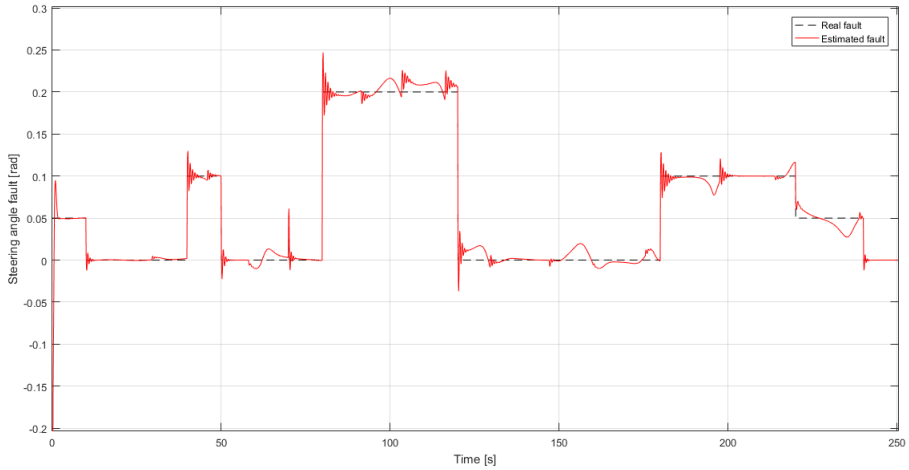

Figure 6.39: Complete FTC - Lateral Actuator Fault Estimation with LSPE

The estimation of the lateral actuator fault δ_{fault} is depicted in the previous Figure 6.39. Comparing it with the estimation in the experiment where the longitudinal actuator fault ($F_{xRfault}$) is not taken into account (see Figure 6.33), we can say that the estimation is the same despite the presence of $F_{xRfault}$. Hence, one could assume that the vehicle should have satisfactory behavior, without deviating from the references. In order to confirm the latter assumption it is needed to check the states of the vehicle in the next Figure 6.40.

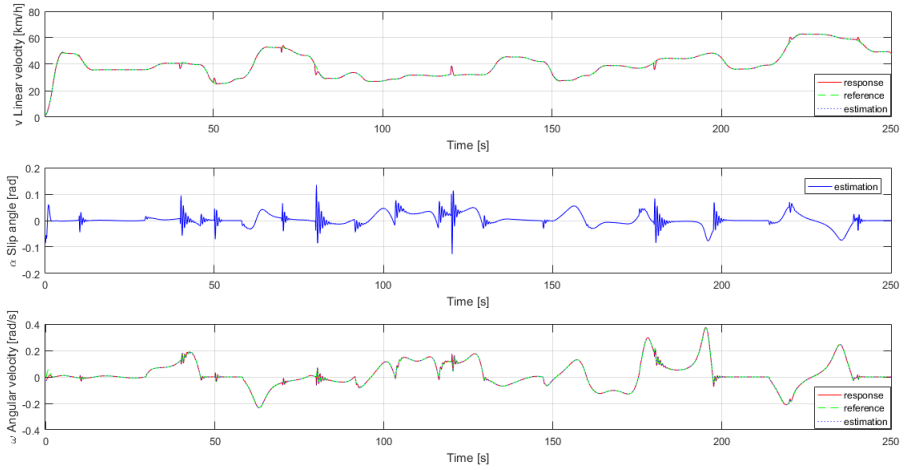

Figure 6.40: Complete FTC with LSPE and UIO - States of the Vehicle

We can observe in the above Figure 6.40 that the presence of both faults affects all the three states v, α, ω. The faults are compensated is a very efficient way, almost the same as in the experiments where the presence of the faults is arranged separately (see Figures 6.20 and 6.34) and so the behavior of the vehicle is satisfactory without departing from the references or having undesired slip angle values. This ascertainment is confirmed by checking the next Figure 6.41.

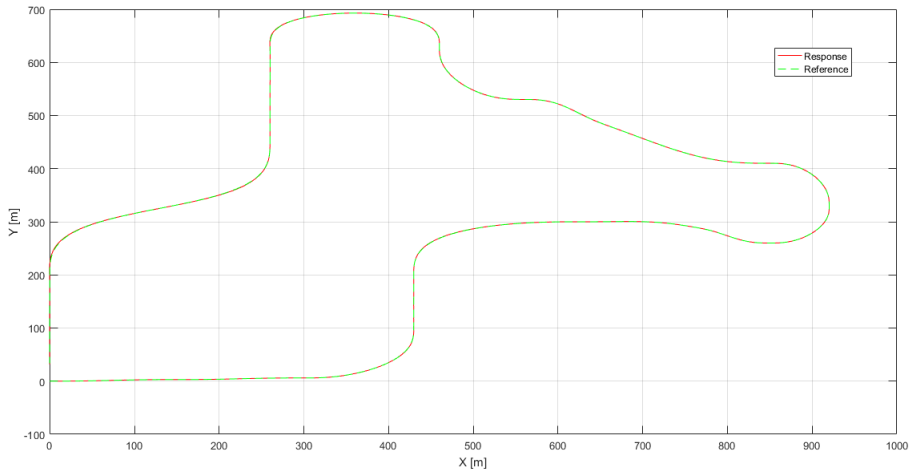

Figure 6.41: Complete FTC with LSPE and UIO - Trajectory of the Vehicle

In the previous figure where the desired trajectory is depicted, as well as the estimation of it, we can confirm that the behavior of the vehicle is satisfactory, as it follows the desired path. This ascertainment verifies the previous testimony (see Figure 6.40) that the vehicle follows the references. Although, in order to investigate it in a better way, it is needed to check the next Figure 6.42 where the position and orientation errors are presented.

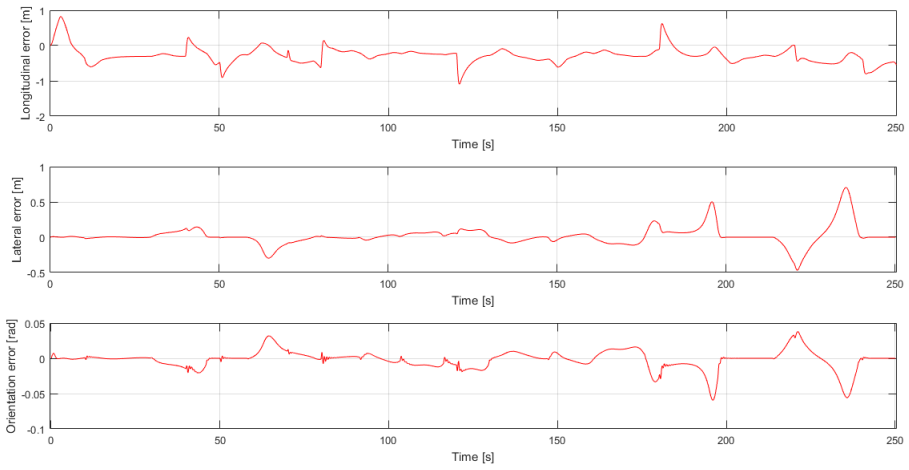

Figure 6.42: Complete FTC with LSPE and UIO - Position and Orientation Errors

As it can be seen in the above Figure 6.42 the errors have similar values with the experiments where the faults are arranged separately (see Figures 6.19 and 6.36). One could assume that with the simultaneously presence of faults, these errors would be larger or different. Although, both the faults are compensated properly, even in their concurrent existence, which can be verified in the next Figure 6.43.

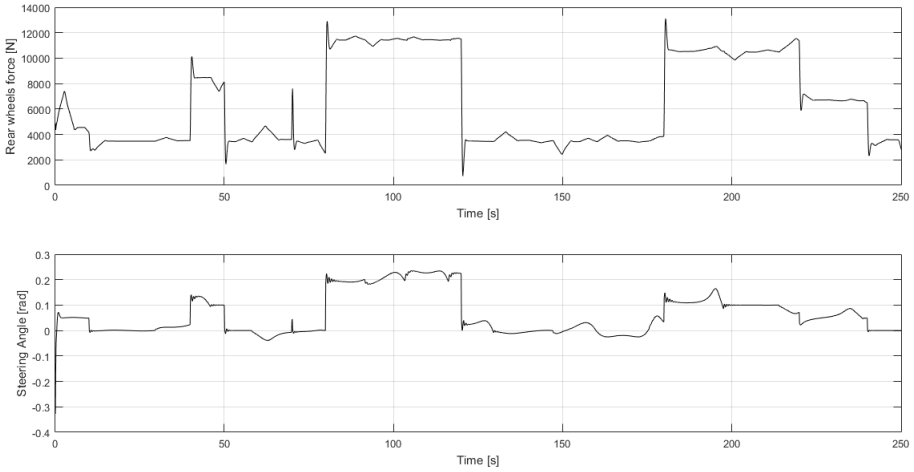

Figure 6.43: Complete FTC with LSPE and UIO - Inputs of the Vehicle

Observing the above figure were the values of the actuators are presented we can confirm that the faults are compensated effectively. We can verify it by checking the simulation without the faults (see Figure 6.15). As we can see, the values of the actuator increments, is exactly the value of the estimated faults (see Figures 6.38 and 6.39), which confirm the satisfactory fault compensation.

$Q_{controllerK}$	$R_{controllerK}$	v_{RMSE}	ω_{RMSE}	X_e	Y_e	θ_e
$\begin{bmatrix} 0.1 & 0 & 0 \\ 0 & 0.1 & 0 \\ 0 & 0 & 0.1 \end{bmatrix}$	$\begin{bmatrix} 0.1 & 0 \\ 0 & 0.1 \end{bmatrix}$					
$Q_{controllerD}$	$R_{controllerD}$					
$\begin{bmatrix} 0.05 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.01 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.01 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.01 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 100000 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 100000 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$	$\begin{bmatrix} 0.01 & 0 \\ 0 & 100 \end{bmatrix}$	**0.1573**	**0.0055**	**0.3594**	**0.1320**	**0.0132**
Q_{UIO}	R_{UIO}					
$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$	$\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$					

Table 6.9: Complete FTC with UIO and LSPE - RMSE

In the previous Table 6.9, we can see that the RMSEs are affected by the presence of the faults. Specifically, the two states that are influenced are the linear and angular velocity. It can be confirmed by comparing these results with the ones without the presence of faults (Table 6.6). The difference between them, is for the v_{RMSE} 0.1235 km/h and for ω_{RMSE} 0.0041 rad/s. It means that in the case where there are actuator faults and the one without them the difference between the errors is not so significant to affect the behavior of the vehicle. The latter ascertainment means that the compensation of the faults by the system is quite effective and consequently the vehicle has the desired behavior.

Chapter 7

Conclusions and Future work

This thesis has proposed gain-scheduling Takagi-Sugeno control strategies, so as to handle the stability and performance of the system (vehicle), as well as to compensate the actuators disturbances and faults. The T-S models obtained from sector non-linearity approach has been proven to be a suitable choice to control the vehicle behavior. In order to design the controllers, the selection of the appropriate LMIs was needed, although it is not a straightforward process. The correct choice of them has been attained by following the experimental procedure. Consequently, as one can ascertain, for each part of the control scheme (Controllers and Observers) different LMI techniques have been employed. Eventually, the complete cascade control scheme has shown remarkable performance. Going further, and taking into account the actuators disturbances and faults, the FTC scheme was constructed. In order to attain this, many techniques have been tested and the more appropriate ones, for each different case of faults and disturbances, have been adopted. Specifically, the UIO suits better than the other methods for the longitudinal actuator fault and disturbance estimation. For the lateral actuator fault estimation, the most appropriate technique is the LSPE. Consequently, the complete FTC scheme, for the concurrent existence of longitudinal and lateral actuator faults, combine the UIO and LSPE methods. The ultimate implementation has been proven to be reliable and quite effective, concerning the faults estimation and compensation, and in general the behavior of the vehicle.

Continuing the work in this thesis one would improve of the FTC implementation. Specifically, they could investigate better why the singularity and numerical issues arose in some of fault estimation methods. In addition, some other methods which are not developed in this work could be tested, and finally the best combination of some of them could be examined.

Finally, the implementation of the ideas developed in this work in a real system/environment, would be another case study.

Bibliography

[1] Fuzzy Control Systems Design and Analysis: A Linear Matrix Inequality Approach. Kazuo Tanaka, Hua O.Wang

[2] LMIs in Control Systems: Analysis, Design and Applications. Guang-Ren Duan, Hai-Hua Yu

[3] Mono- and Multivariable Control and Estimation Linear, Quadratic and LMI Methods. Ostertag, Eric

[4] Advanced Takagi–Sugeno Fuzzy Systems Delay and Saturation. Benzaouia, Abdellah, El Hajjaji, Ahmed

[5] Fuzzy Gain Scheduling: Controller and Observer Design Based on Lyapunov Method and Convex Optimization. Petr Korba, Robert Babuska, Henk B. Verbruggen, and Paul M. Frank

[6] Gain Scheduling LPV Control Scheme for the Autonomous Guidance Problem using a Dynamic Modelling Approach. E.Alcala1 V. Puig1 J. Quevedo1 T. Escobet1

[7] P. Apkarian, P. Gahinet, G. Becker, Self-Scheduled H_∞ Control of Linear Parameter-Varying Systems: A Design Example Automatica, vol. 31, no. 9, pp. 1251-1261, 1995.

[8] An approach to fuzzy control of nonlinear systems: Stability and design issues. HO Wang, K Tanaka, MF Griffin IEEE transactions on fuzzy systems 4 (1), 14-23 1996

[9] S. Boyd, L. E. Ghaoui, E. Feron, and V. Belakrishnan, "Linear matrix inequalities in system and control theory," in SIAM: Studies In Applied Mathematics. Philadelphia, PA: SIAM, 1994, vol. 15.

[10] C. G. L. Bianco, A. Piazzi, M. Romano, Velocity planning for autonomous vehicles, IEEE Intelligent Vehicles Symposium.

[11] Autonomous Intelligent Vehicles Theory, Algorithms, and Implementation Authors: Cheng Hong, 2011

[12] Nonlinear Control of Wheeled Mobile Robots Authors: Dixon, W.E., Dawson, D.M., Zergeroglu, E., Behal, A. 2001

[13] A Mathematical Introduction to Robotic Manipulation Authors: Richard M. Murray, Zexiang Li,S. Shankar Sastry, 1994

[14] Diagnosis and Fault-Tolerant Control Authors: Blanke, M., Kinnaert, M., Lunze, J., Staroswiecki, M., 2006

[15] Fault Diagnosis and Fault-Tolerant Control Strategies for Non-Linear Systems Analytical and Soft Computing Approaches Authors: Witczak, Marcin, 2014

[16] Fault Estimation using a Takagi-Sugeno Interval Observer: Application to a PEM Fuel Cell Authors: C. Martınez Garcıa, V. Puig, C.Astorga Zaragoza

[17] Fault-tolerant control design using the linear parameter varying approach Authors: Saúl Montes de Oca, Sebastian Tornil-Sin, Vicenç Puig, Didier Theilliol

[18] A Fault-Hiding Approach for the Switching Quasi-LPV Fault-Tolerant Control of a Four-Wheeled Omnidirectional Mobile Robot Authors: Damiano Rotondo, Vicenç Puig, Fatiha Nejjari, Juli Romera

[19] T. Takagi and M. Sugeno, "Fuzzy identification of systems and its applications to modeling and control," IEEE Trans. Syst., Man, Cybern., vol. SMC-15, no. 1, pp. 116–132, Jan./Feb. 1985.

[20] M. Sugeno, Fuzzy Control. North-Holland, 1988.

[21] M. Sugeno and G. Kang, "Structure identification of fuzzy model," Fuzzy Sets and Systems, vol. 28, no. 1, pp. 15–33, 1988.