Ecole Nationale Polytechnique

Automatics department

[SEM] Seminary Report

Sliding Mode Observer-Based Control for Takagi-Sugeno Fuzzy Systems

Supervised by :

- Dr. Hakim ACHOUR
- Pr. Djamel BOUKHETALA

Realised by :

- Amine Rami BENNACER
- Yasser HADJ

Contents

1 Introduction **4**

2 Takagi-Sugeno Fuzzy Systems **5**
2.1 Fuzzy Systems . . . 5
2.1.1 Mamdani vs TSK models . . . 5
2.1.2 TS Approach in control theory . . . 5
2.2 Obtainment Of A TS-Fuzzy Model . . . 6
2.2.1 Linearisation . . . 6
2.2.2 Sector non-linearity . . . 8
2.2.3 Remarks . . . 12

3 Linear Matrix Inequalities (LMI) **14**
3.1 Definition . . . 14
3.2 LMI problems . . . 14

4 Sliding Mode Observes **15**
4.1 Observer Synthesis for Certain Linear Systems . . . 15
4.2 Classical Sliding Mode Control . . . 16
4.3 Examples of sliding control . . . 17
4.3.1 example 01 . . . 17
4.3.2 example 02 . . . 18
4.4 Sliding Mode Observers . . . 19
4.4.1 Utkin Discontinuous Observer . . . 19
4.4.2 Walcott and Zak Discontinuous Observer . . . 20
4.5 Example of Walcott and Zak's observer . . . 22

5 Applcation : The Cart and Double Pendulum **25**
5.1 TS fuzzy model . . . 25
5.2 Sliding Mode observation . . . 26
5.3 Command example: LQR . . . 28

6 Conclusion and perspectives **30**

7 Footmarks **32**

8 References **33**

List of Figures

1 Fuzzy system data evaluation [J.S Castillo] 5
2 Membership functions of two-rule model[6] 7
3 Inverted pendulum fuzzy simulink model 8
4 Subsystem used for inverted pendulum fuzzy simulink model 8
5 Fuzzy states and real states 9
6 Membership functions of $z_1(t)$ and $z_2(t)$ 11
7 Simulink models of fuzzy system and real system 12
8 Subsystem of fuzzy model 12
9 Plot of fuzzy states and real states 13
10 Observation Scheme for a Certain Linear System 15
11 Phase Plan of a sliding mode system 17
12 Phase Plan Plot of the system's response to a: (a) big command, (b) small command, (c) sliding mode command 18
13 Phase Plan Plot of the system's response to a: (a) positive feedback, (b) negative feedback, (c) sliding mode command 18
14 Walcott and Zak Discontinuous Observer 23
15 Simulations Results for Walcott and Zak observer 24
16 Cart and double pendulum[11] 25
17 Cart and double pendulum fuzzy states vs real states 26
18 Sliding Mode Observation Scheme 27
19 Global Application Scheme 27
20 Observations Results : Errors and Commutations 28
21 Stabilised states of the double pendulum and cart system 29

1 Introduction

In this paper our main purpose as the title suggests is to make a command based on sliding modes observers for fuzzy systems of type Takagi-Sugeno.
We will try along the chapters to explain every aspect of our work individually, from fuzzy systems, TS model and LMI's to sliding modes and observers.We will then in the end combine everything to answer our problematic.
Uncovering the specifications and characteristics of every aspect of this research is also included in our work. Finding the advantages of the method, the disadvantages and limitations, and trying to come up with perspectives for future works.
We will do our best to simplify each concept for the reader, and try to give as much examples and illustrations as possible to make the lecture easy on the mind.

2 Takagi-Sugeno Fuzzy Systems

2.1 Fuzzy Systems

Fuzzy Logic has known great success ever since the first works of L.Zadeh in the late 60's. This success led to it being used in a variety of fields and domains, but in the control field specifically it brought a new way to look at system modelisation. The fuzzy approach is an approach based on logical rules instead of numerical rules, hence the name, it is a way to represent the vague and the unclear and to turn linguistic thoughts into mathematical equations.

2.1.1 Mamdani vs TSK models

We recognize two main categories of fuzzy models, the Mamdani model and the Takagi-Sugeno-Kang model (usually referenced as TS or TSK), both approaches are similar in the process of getting the model , which is :

1. Fuzzification : Which is to turn our real inputs (crisp inputs[1]) into fuzzy inputs[2]using membership functions[3].
2. Output Computing : we compute the output using a set of logical rules (ex: if A and B then y) that will be turned into mathematical rules (examples will be given along this chapter) resulting in fuzzy outputs.
3. Defuzzification : as the name suggests it, this step is the opposite of the first one, we turn our fuzzy outputs into crisp outputs.

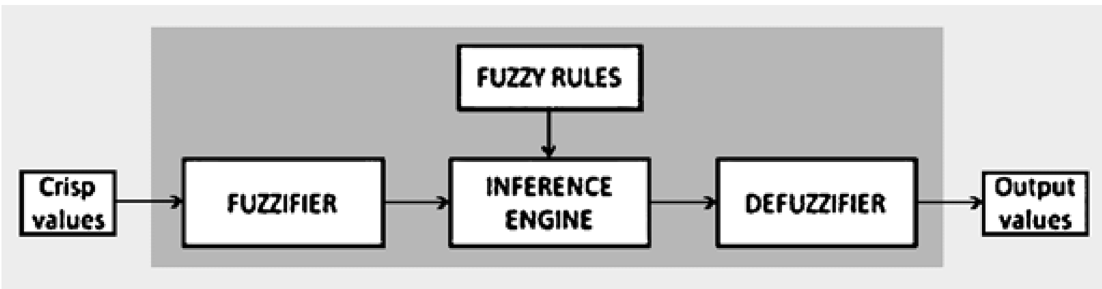

Figure 1: Fuzzy system data evaluation [J.S Castillo]

The main difference between the 2 approaches mentioned previously is that the Mamdani approach uses sub-ensembles to get the fuzzy outputs, while the TS approach uses a linear function (usually polynomial) of the input variables to combine the 2^{nd} and 3^{rd} steps and get directly a crisp output.

2.1.2 TS Approach in control theory

In control theory the TS approach is the most widely used between the 2 approaches because it allows to turn non-linear systems into an interpolation of local linear model, each of these models being an LTI model. The general form of a TS state model is :

$$\begin{cases} \dot{x}(t) = \sum_{i=1}^{n} \mu_i(z(t))(A_i x(t) + B_i u(t)) \\ y(t) = Cx(t) \end{cases} \tag{1}$$

With $z(t) \in \mathbb{R}^q$ being the decision vector variable (input for fuzzy system) usually containing a combination of x(t) and u(t), and $\mu_i(.), i \in \mathbb{I}_n$ is the activation function[4]for the state model i. We often say that $\mu(.)$ is normalised , meaning :

$$\begin{cases} 0 < \mu_i(.) < 1 \\ \sum_{i=1}^{n} \mu_i(.) = 1 \end{cases}$$

And so we write that :

$$\begin{cases} \mu_i(.) = \frac{\omega_i}{\sum_{i=1}^{n} \omega_i} \\ \omega_i > 0 \end{cases}$$

With ω_i being the consequent weights resulting from the rules and membership functions.
Remark : the activation function isn't a probability rather it's how much the model belongs to the set i.

2.2 Obtainment Of A TS-Fuzzy Model

Given a non-linear mathematical representation of a model, the question here is how to get the state models (A_i, B_i) and the activation functions $\mu_i(.)$ (of course in regards of our choice of z(t)) to represent the non-linear system as a TS model.For that 2 main methods have been used :

- Linearisation.
- Sector non-linearity.

2.2.1 Linearisation

The idea of this method is to linearise the non-linear state space model at different local points to get the linear state space model to interpolate, meanwhile the rules and the membership functions are defined by logic and by our knowledge of the system.
Let's say we have the non linear system defined by the equation :

$$\dot{x}(t) = f(x(t), u(t)) \tag{2}$$

With f(.) being a derivable function, the linearisation of our system at the local point (x_i, u_i) will give us :

$$\dot{x}(t) = A_i(x(t) - x_i) + B_i(u(t) - u_i) + f(x_i, u_i) \tag{3}$$

With

$$Ai = \left.\frac{\partial f}{\partial x}\right|_{(x,u)=(x_i,u_i)}, B_i = \left.\frac{\partial f}{\partial u}\right|_{(x,u)=(x_i,u_i)}$$

Let's put $d_i = f(x_i, u_i) - A_i x_i - B_i u_i$ and rewrite the equation (3) as :

$$\dot{x}(t) = A_i x(t) + B_i u(t) + d_i \tag{4}$$

Therefore for n local points we can write the equation :

$$\dot{x}(t) = \sum_{i=1}^{n} \mu_i(z(t))(A_i x(t) + B_i u(t) + d_i) \tag{5}$$

To note that our choice of the number of local points (therefore local models) "n" is based on the desired precision , and the complexity of our non-linear model.
Example 1.1: Given the non-linear equations of an inverted pendulum [8] :

$$\begin{cases} \dot{x}_1(t) = x_2(t) \\ \dot{x}_2(t) = \frac{g.sin(x_1(t)) - a.m.l.x_2(t)^2.sin(2x_1(t))/2 - a.cos(x_1(t)).u(t)}{4l/3 - a.m.l.cos^2(x_1(t))} \end{cases}$$

Where $x_1(t)$ represents the angle of rotation of the pendulum, $x_2(t)$ is it's derivative (meaning the angular velocity), and u is the force applied .g is the gravity constant, a=1/(m+M) where m is the mass of the pendulum and M the mass of the cart, and 2l is the length of the pendulum.
The linearised model when x_1 is around 0 will give us :

$$A_1 = \begin{bmatrix} 0 & 1 \\ \frac{g}{4l/3 - aml} & 0 \end{bmatrix}, B_1 = \begin{bmatrix} 0 \\ -\frac{a}{4l/3 - aml} \end{bmatrix}$$

While around $\pm\frac{\pi}{2}$ we get :

$$A_2 = \begin{bmatrix} 0 & 1 \\ \frac{2g}{\pi(4l/3 - aml\beta^2)} & 0 \end{bmatrix}, B_2 = \begin{bmatrix} 0 \\ \frac{-a\beta}{4l/3 - aml} \end{bmatrix}$$

with $\beta = cos(88)$ as an approximation.
For the computing of the membership functions we consider $z(t) = x_1(t)$ and we assume the two following rules :

- **Rule 1:** if x_1 is near 0, $\dot{x}(t) = A_1 x(t) + B_1 u(t)$
- **Rule 2:** if x_2 is near $\pm\frac{\pi}{2}$, $\dot{x}(t) = A_2 x(t) + B_2 u(t)$

Now we can define our membership function as two triangular functions : So we can sim-

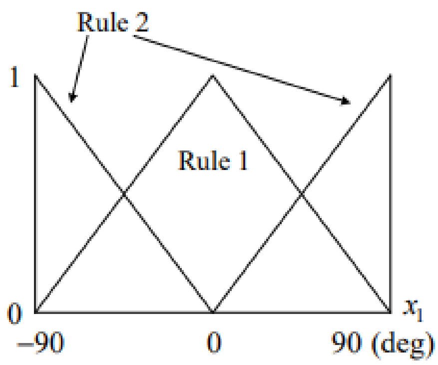

Figure 2: Membership functions of two-rule model[6]

ulate this model on simulink using the following structure in figure 3 with the subsystem used is in figure 4 giving us the results in figure 5 that show that the fuzzy satates are alligned with the real non-linear system, but we should note that with time they start to diverge as we only used 2 local points which is insufficient for the whole system, and adding more points would add to the complexity of the method.

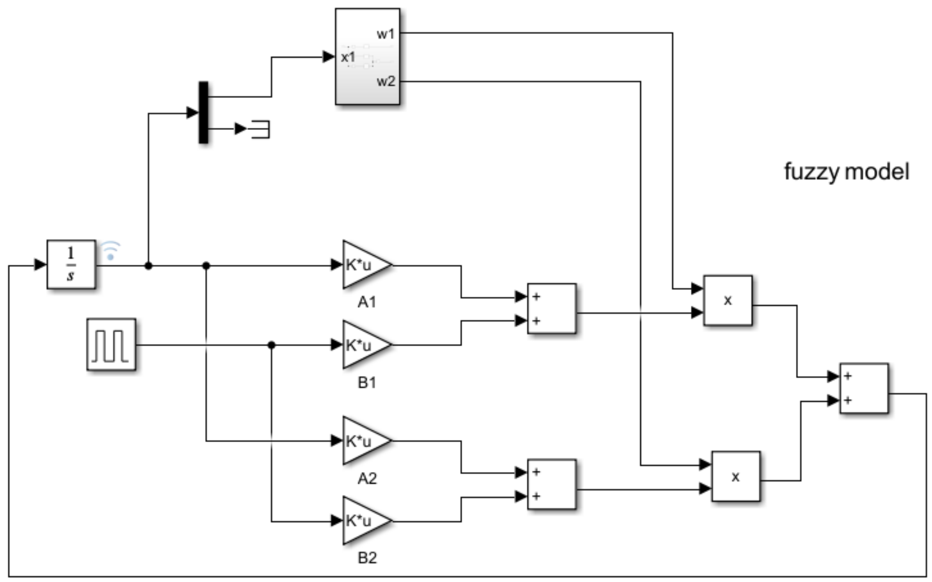

Figure 3: Inverted pendulum fuzzy simulink model

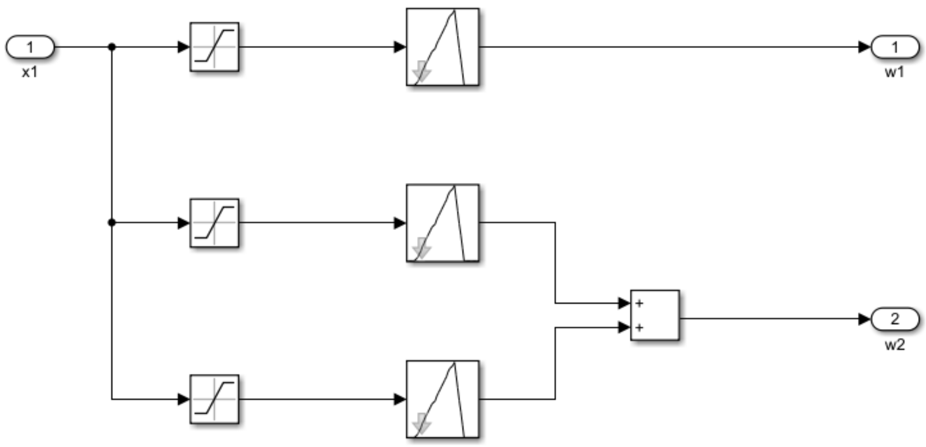

Figure 4: Subsystem used for inverted pendulum fuzzy simulink model

2.2.2 Sector non-linearity

In this method the goal is to generate the state models by bounding the non-linearities, meaning for $z_i(t)$ a non linearity (and a decision variable), we replace each $z_i(t)$ with $\alpha = min(z_i(t))$ and $\beta = max(z_i(t))$. Of course this requires the non-linearities to be bounded, if not then we reduce the borders of our states x_i.
So if given for example :

$$A = \begin{bmatrix} z_1 & a_{12} & a_{13} \\ a_{21} & a_{22} & z_2 \\ a_{31} & z_3 & a_{33} \end{bmatrix}$$

with $a_i i$ constants , we would get :

$$A_1 = \begin{bmatrix} \alpha_1 & a_{12} & a_{13} \\ a_{21} & a_{22} & \alpha_2 \\ a_{31} & \alpha_3 & a_{33} \end{bmatrix} A_2 = \begin{bmatrix} \alpha_1 & a_{12} & a_{13} \\ a_{21} & a_{22} & \alpha_2 \\ a_{31} & \beta_3 & a_{33} \end{bmatrix} A_3 = \begin{bmatrix} \alpha_1 & a_{12} & a_{13} \\ a_{21} & a_{22} & \beta_2 \\ a_{31} & \alpha_3 & a_{33} \end{bmatrix} A_4 = \begin{bmatrix} \alpha_1 & a_{12} & a_{13} \\ a_{21} & a_{22} & \beta_2 \\ a_{31} & \beta_3 & a_{33} \end{bmatrix}$$

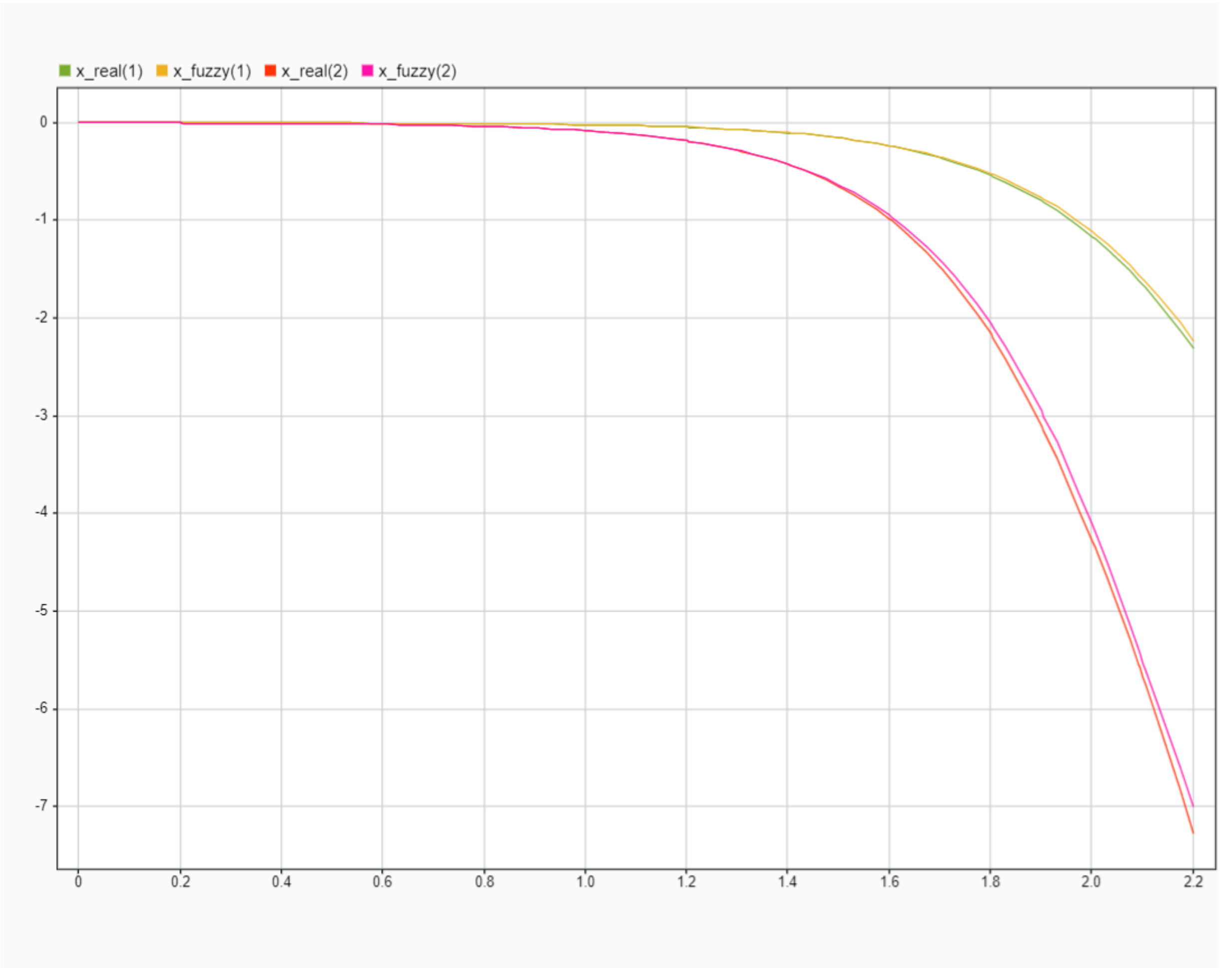

Figure 5: Fuzzy states and real states

$$A_1 = \begin{bmatrix} \beta_1 & a_{12} & a_{13} \\ a_{21} & a_{22} & \alpha_2 \\ a_{31} & \alpha_3 & a_{33} \end{bmatrix} A_2 = \begin{bmatrix} \beta_1 & a_{12} & a_{13} \\ a_{21} & a_{22} & \alpha_2 \\ a_{31} & \beta_3 & a_{33} \end{bmatrix} A_3 = \begin{bmatrix} \beta_1 & a_{12} & a_{13} \\ a_{21} & a_{22} & \beta_2 \\ a_{31} & \alpha_3 & a_{33} \end{bmatrix} A_4 = \begin{bmatrix} \beta_1 & a_{12} & a_{13} \\ a_{21} & a_{22} & \beta_2 \\ a_{31} & \beta_3 & a_{33} \end{bmatrix}$$

Remark : usually for n non-linearities we get 2^n models at most.
The advantage of this method is that the membership functions are explicitly calculated and not chosen. Given our decision vector z(t), each element $z_i(t)$ is represented by two membership functions defined as follows :

$$M_i(z_i(t)) = \frac{z_i(t) - \alpha}{\beta - \alpha} \tag{6}$$

and

$$N_i(z_i(t)) = \frac{\beta - z_i(t)}{\beta - \alpha} \tag{7}$$

And we can write that :

$$z_i(t) = M_i(z_i(t)).\beta + N_i(z_i(t)).\alpha \tag{8}$$

With $M_i(z_i(t)) + N_i(z_i(t)) = 1$.
And finally we have the activation functions defined as the products of the membership functions , i.e if we have two decision variables then :

$$\mu_1(z(t)) = M_1(z_1(t)).M_2(z_2(t))$$

$$\mu_2(z(t)) = M_1(z_1(t)).N_2(z_2(t))$$
$$\mu_3(z(t)) = N_1(z_1(t)).M_2(z_2(t))$$
$$\mu_4(z(t)) = N_1(z_1(t)).N_2(z_2(t))$$

Example 1.2: Let's take the following non-linear system [6] :

$$\begin{cases} \dot{x_1}(t) = -x_1(t) + x_1(t).x_2^3(t) \\ \dot{x_2}(t) = -x_2(t) + (3 + x_2(t)).x_1^3(t) \end{cases}$$

For simplicity we will assume that $x_i \in [-1; 1]$.
Let's write the non-linear state space model :

$$\begin{pmatrix} \dot{x_1}(t) \\ \dot{x_2}(t) \end{pmatrix} = \begin{pmatrix} -1 & x_1(t).x_2(t)^2 \\ (3 + x_2(t)).x_1(t)^2 & -1 \end{pmatrix} . \begin{pmatrix} x_1(t) \\ x_2(t) \end{pmatrix} = \begin{pmatrix} -1 & z_1(t) \\ z_2(t) & -1 \end{pmatrix} . \begin{pmatrix} x_1(t) \\ x_2(t) \end{pmatrix}$$

With $z_1(t) = x_1(t).x_2(t)^2$ and $z_2(t) = (3 + x_2(t)).x_1(t)^2$ our decision variables, so we get $\alpha_1 = min(z_1(t)) = -1$, $\beta_1 = max(z_1(t)) = 1$ and $\alpha_2 = min(z_2(t)) = 2$, $\beta_1 = max(z_1(t)) = 4$.
So our state models will be :

$$A_1 = \begin{pmatrix} -1 & 1 \\ 4 & -1 \end{pmatrix}, A_2 = \begin{pmatrix} -1 & 1 \\ 2 & -1 \end{pmatrix}$$

$$A_3 = \begin{pmatrix} -1 & -1 \\ 4 & -1 \end{pmatrix}, A_4 = \begin{pmatrix} -1 & -1 \\ 2 & -1 \end{pmatrix}$$

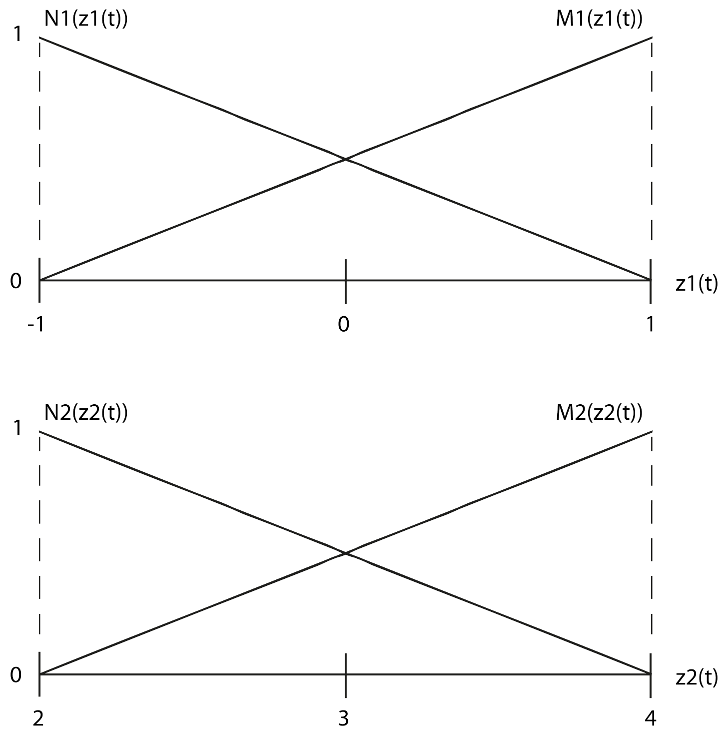

Figure 6: Membership functions of $z_1(t)$ and $z_2(t)$

As defined previously our membership functions will be :

$$M_1(z_1(t)) = \frac{z_1(t)+1}{2}, N_1(z_1(t)) = \frac{1-z_1(t)}{2}$$
$$M_2(z_2(t)) = \frac{z_2(t)-2}{2}, N_2(z_2(t)) = \frac{4-z_2(t)}{2}$$

Represented by the figure 6 .

From this we can define our set of rules as :

- **Rule 1:** If $M_1(z_1(t))$ and $M_2(z_2(t))$ then $\dot{x}(t) = A_1x(t)$
- **Rule 2:** If $M_1(z_1(t))$ and $N_2(z_2(t))$ then $\dot{x}(t) = A_2x(t)$
- **Rule 3:** If $N_1(z_1(t))$ and $M_2(z_2(t))$ then $\dot{x}(t) = A_3x(t)$
- **Rule 4:** If $N_1(z_1(t))$ and $N_2(z_2(t))$ then $\dot{x}(t) = A_4x(t)$

And we have our activation functions :

$$\mu_1(z(t)) = M_1(z_1(t)).M_2(z_2(t))$$
$$\mu_2(z(t)) = M_1(z_1(t)).N_2(z_2(t))$$
$$\mu_3(z(t)) = N_1(z_1(t)).M_2(z_2(t))$$
$$\mu_4(z(t)) = N_1(z_1(t)).N_2(z_2(t))$$

So finally :

$$\dot{x}(t) = \sum_{i=1}^{n} \mu_i(z(t))A_ix(t)$$

We can thus represent the system by the simulink model in figures 7 and 8 that gives us the results in figures 9 where we can clearly see that our fuzzy states are alligned with the real states.

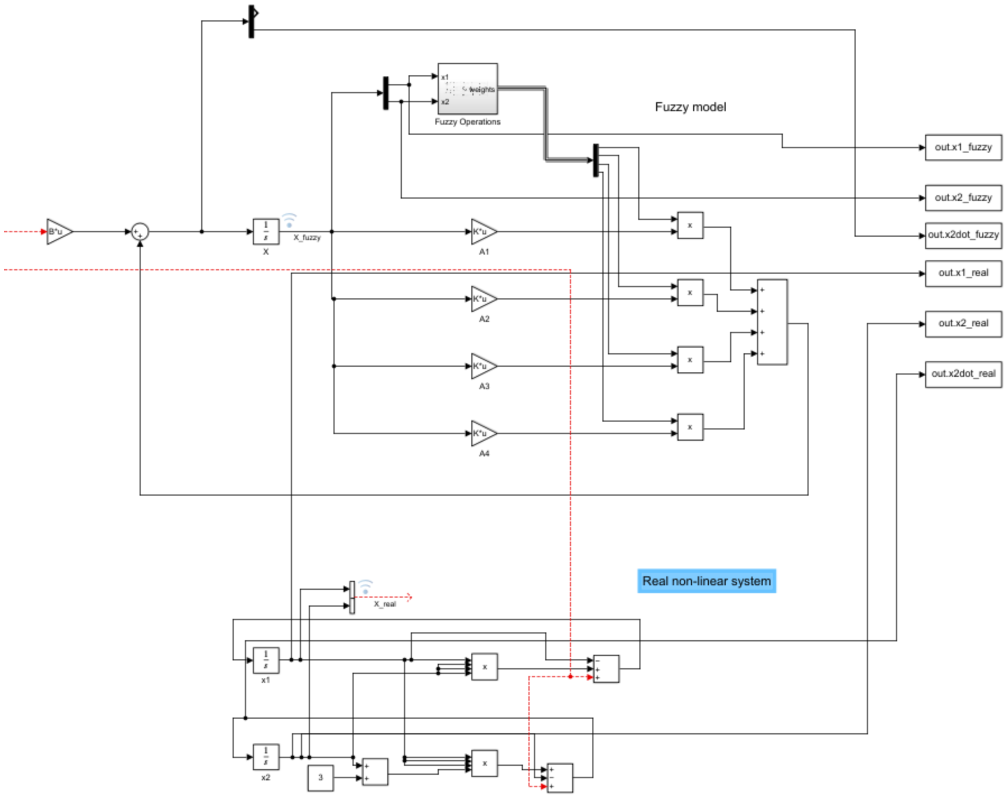

Figure 7: Simulink models of fuzzy system and real system

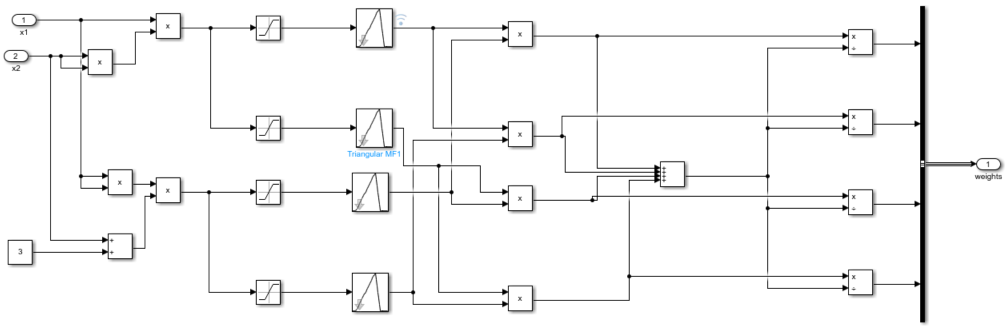

Figure 8: Subsystem of fuzzy model

2.2.3 Remarks

The two approaches previously mentioned both present advantages and disadvantages, those are some of them :

- The linearisation approach is good for reducing the amount of rules needed , but it's a method based on trial and error as some parameters need to be tuned (number of local points, decision variables, membership functions, rules...).
- The sector non-linearity approach is pure computing no guessing is needed, but the number of rules needed increases with the number of non-linearities, for example for the inverted pendulum problem 16 rules will be needed to compute the model.

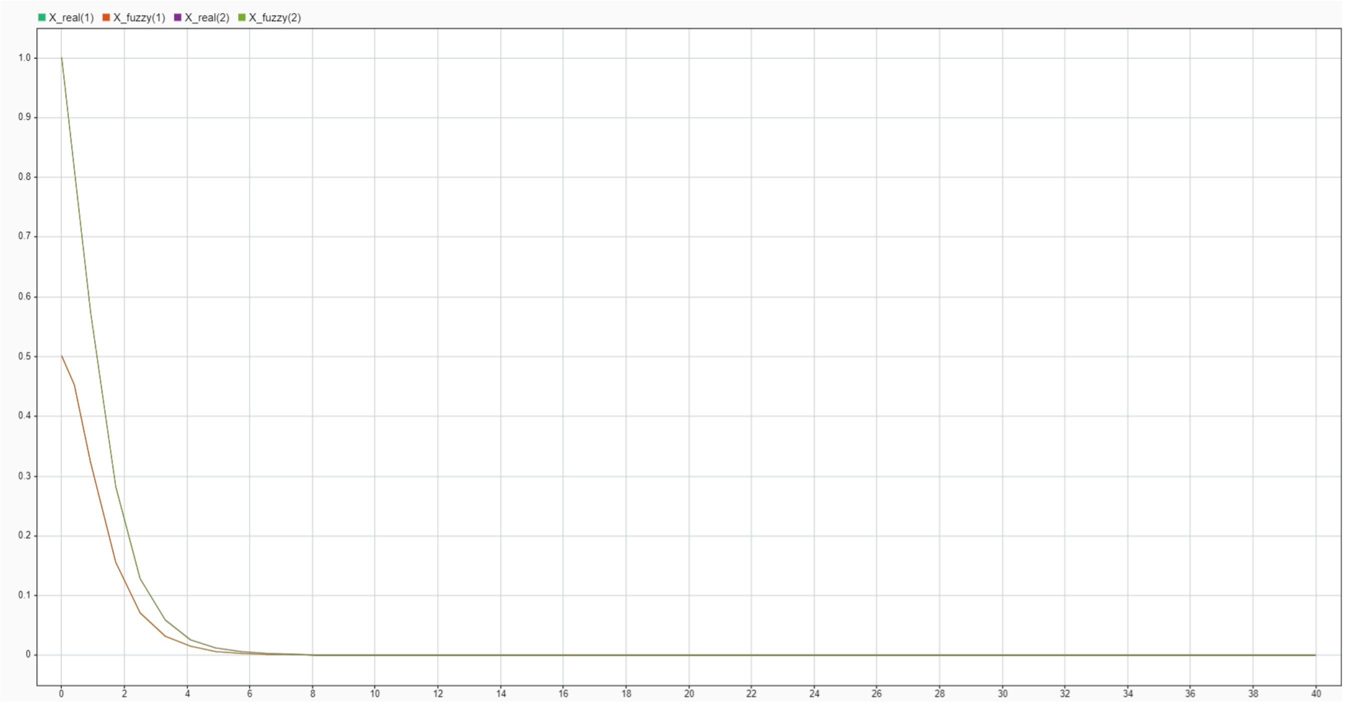

Figure 9: Plot of fuzzy states and real states

- The range of the states is important for the validity of both approaches.

3 Linear Matrix Inequalities (LMI)

Before we continue any further, we need to stop and talk about Linear Matrix Inequalities or LMI for short since a lot of our problems will be in the form of LMI's.
A Linear Matrix Inequality is a type of convex problems[5], it has been at the center of attention of control theorists since it's first appearance in the late 1800's. The reason being that most control theory problems can be written as LMI problems, so solving them became a matter of high interest.

3.1 Definition

Given a set of symmetrical matrices P_0 and $P_i \in \mathbb{R}^{p.p}$ with $i \in I_n$, and a vector $x = (x_0, x_1, x_2, ..., x_n)^T \in \mathbb{R}^n$, a strict LMI (resp. non-strict) in x is written as :

$$F(x) = P_0 + \sum_{i=1}^{n} x_i P_i > 0 (resp \geq) \tag{9}$$

3.2 LMI problems

The most knonw (and the first) LMI problem is the lyapunov stability LMI that states that in order for a problem of the form $\dot{x} = Ax$ to be exponentially stable if and only if a positive-definite matrix P exists such that :

$$A^T P + PA < 0$$

Which is a special form of an LMI, and he found that the solution is obtained by solving for P the equation $A^T P + PA = -Q$ with Q being any matrix that satisfies $Q = Q^T > 0$.
Apart from that the most encountered LMI problems are :

- **Realisability problems :** It is about finding a vector x such as F(x)¿0 is satsified.
- **Eigen values problems (EVP) :**it is about minimising the greatest eigen value of a symmertical matrix such that :

$$\begin{cases} \lambda I - A(x) > 0 \\ B(x) > 0 \end{cases}$$

- **Generalised eigen values problems (GEVP) :** it is the same as the previous one but working with a pair of matrices instead :

$$\begin{cases} \lambda B(x) - A(x) > 0 \\ B(x) > 0 \\ C(x) > 0 \end{cases}$$

4 Sliding Mode Observes

4.1 Observer Synthesis for Certain Linear Systems

Observers are used when we want to estimate the state of a system or a linear function of it (such as the output of a system) cause we don't have or can't install appropriate sensors. As shown in figure (10), a state reconstructor (or an estimator) is a system that takes the inputs and outputs of the process and produces an estimation of the process state as its output.

The conception of an observer involves correcting the estimation error between the actual output and the reconstructed output, and fusing it with the system's dynamics to make the estimation error converge to zero.

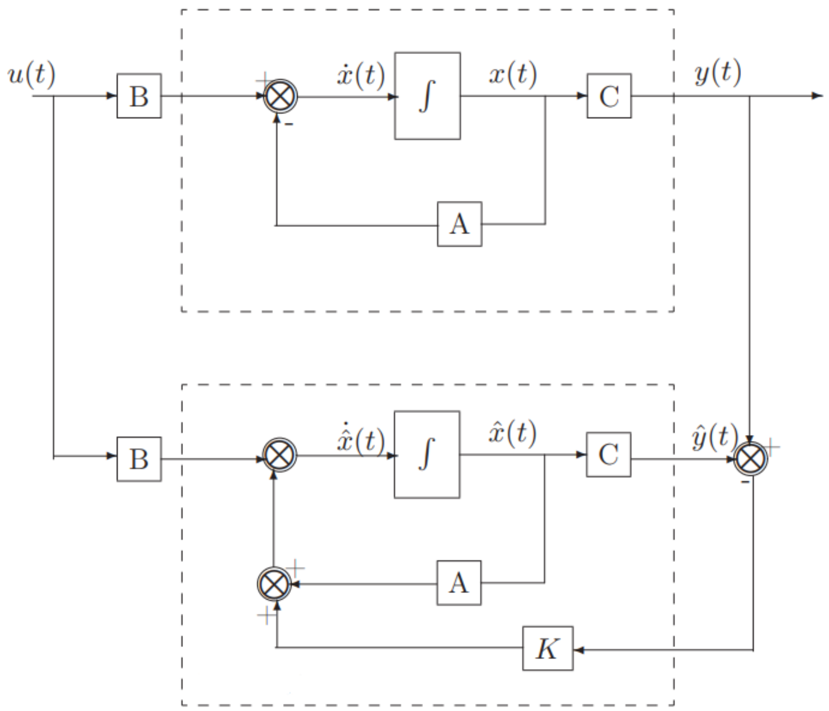

Figure 10: Observation Scheme for a Certain Linear System

Consider a certain LTI system defined by:

$$\begin{cases} \dot{x}(t) = Ax(t) + Bu(t) \\ y(t) = Cx(t) \end{cases} \tag{10}$$

where $x(t) \in \mathbb{R}^n$ is the state vector and $y(t) \in \mathbb{R}^p$ is the output vector.

The corresponding linear observer is defined by:

$$\begin{cases} \dot{\hat{x}}(t) = A\hat{x}(t) + Bu(t) + Ke_y(t) = (A - KC)\hat{x}(t) + Bu(t) + Ky(t) \\ \hat{y}(t) = C\hat{x}(t) \end{cases} \tag{11}$$

where $e_y = y(t) - \hat{y}(t) \in \mathbb{R}^p$ is the estimation error that needs to converge to zero and $K \in \mathbb{R}^{n.p}$ is the associated observer gain.

Every type of observer has a different way of calculating K: whether it be with pole placement approach or using Riccatti equations or Lyapunov matrix inequalities. What is important is that :

- $K \in \mathbb{R}^{n.p}$ exists if and only if the pair (A, C) is totally observable.
- K is chosen in a way that $A_{CL} = A - KC$ is exponentially stable.
- K is of bigger value for less accurate system estimation, and of higher value for noisy output signals.

4.2 Classical Sliding Mode Control

Sliding mode control is fundamentally a consequence of discontinuous control. In fact, it was a result of robust control laws that the sliding mode control was discovered to overcome the gain limitations in the existing regulators.

Typically, the command used to act on the error signal e_x of the feedback loop, were defined by:

$$u = |F_1(x, \dot{x}, ...)| sgn(F_2(x, \dot{x}, ...)) \tag{12}$$

where || deontes the absolute value and the paire (F_1, F_2) are appropriate linear filters. Hence the output was discontinuous but modulated by a function of the error signal and its derivatives. Under its simplest form :

$$u = -|x| sgn(x + k\dot{x}) \tag{13}$$

Under the approximation of the first harmonic (for $x(t) = x_0 \sin(wt)$), the gain of the controller depends only on the pulsation w just like a linear network, hence their denomination as "pseudo linear network". The base concept is easy to understand. Consider a system with the dynamics defined by:

$$\dot{x}(t) = f(x(t), u(t)) \tag{14}$$

that is continuous piecewise, where f is a discontinuous function on either sides of a surface $S(x(t)) = 0$ in a way that:

$$\dot{x}(t) = \begin{cases} f^+(x(t), u(t)) & ; \quad S(x(t)) > 0 \\ f^-(x(t), u(t)) & ; \quad S(x(t)) < 0 \end{cases} \tag{15}$$

If the trajectories of these equations always converge to the surface $S(x(t)) = 0$, there will be no way out and the system is stuck to evolve there. As illustrated in figure (11).

A system defined by [15] is called of variable structure, in which the dynamic in the sliding surface is fundamentally different from both dynamics outside it.

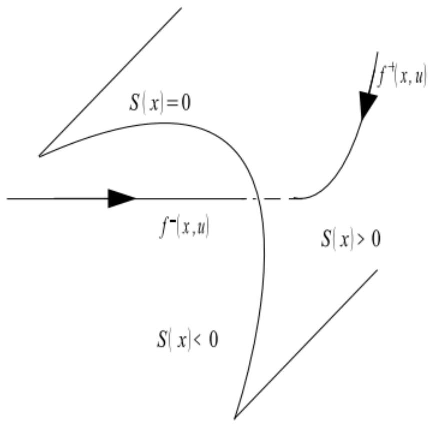

Figure 11: Phase Plan of a sliding mode system

4.3 Examples of sliding control

The synthesis of such systems is defined by a commutation function and commutation logic. This enables the system to commute, at any given moment, between both structures in order to combine their advantages and obtain the desired behavior of the system. Not only that but a system of variable structure can also generate new properties that are not present in any of the used structures.

The following examples best illustrate the usefulness of sliding mode systems:

4.3.1 example 01

Consider a second order non linear system defined by:

$$\ddot{x}(t) = -u(t)x(t) \tag{16}$$

The behavior of such a system depends on the command $u(t)$, which can have two values α_1^2 and α_2^2 with $\alpha_2^2 << \alpha_1^2$. Figure (12) shows that the behavior of the system is not asymptotically stable for either of the values of the command. Meanwhile, it becomes asymptotically stable (AS) if the system's structure changes with the change of the command (commutation function) following a commutation logic defined by:

$$u(t) = \begin{cases} \alpha_1^2 & ; \quad x(t)\dot{x}(t) > 0 \\ \alpha_2^2 & ; \quad x(t)\dot{x}(t) < 0 \end{cases} \tag{17}$$

where the commutation logic is defined by $x(t)\dot{x}(t)$ that divides the phase plan into its 4 quadrants. The resulting behavior is stable and the phase plan plot converges to zero:

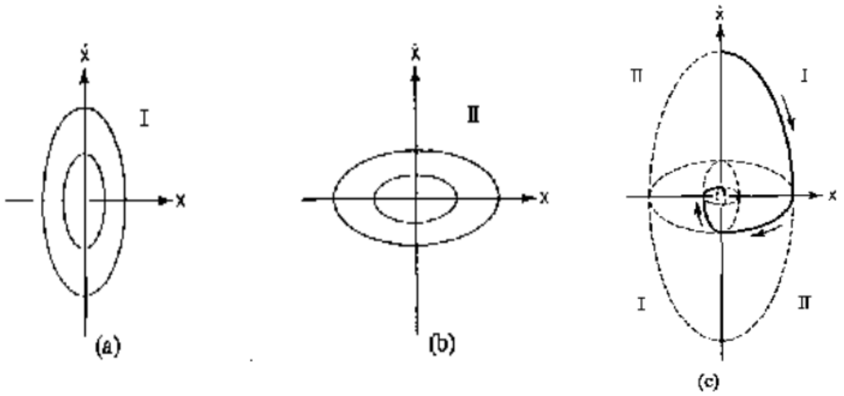

Figure 12: Phase Plan Plot of the system's response to a:
(a) big command, (b) small command, (c) sliding mode command

4.3.2 example 02

Consider the 2nd order system defined by its linear differential equation:

$$\ddot{x}(t) - \xi\dot{x}(t) + \psi x(t) = 0 \qquad /\xi > 0 \tag{18}$$

where the state feedback control is defined by $u(t) = \psi x(t)$. It is positive when $\psi = cte > 0$ and is negative when $\psi = cte < 0$ but both structures are unstable. Meanwhile, if we take ψ as commutation function and we change it following the commutation logic defined by:

$$\psi = \begin{cases} \alpha & if \quad x(t)s(t) > 0 \\ -\alpha & if \quad x(t)s(t) < 0 \end{cases} \tag{19}$$

where $\alpha > 0$ is the state feedback gain and

$$s(t) = cx(t) + \dot{x}(t) \qquad /c = -\frac{\xi}{2} \pm \sqrt{\frac{\xi^2}{4} + \alpha} \tag{20}$$

the behavior of the system changes and the phase trajectories goes from constructing a saddle point and an unstable focus, to converging to the commutation line $s(t)$. The following figure illustrates best the mentionned behaviors.

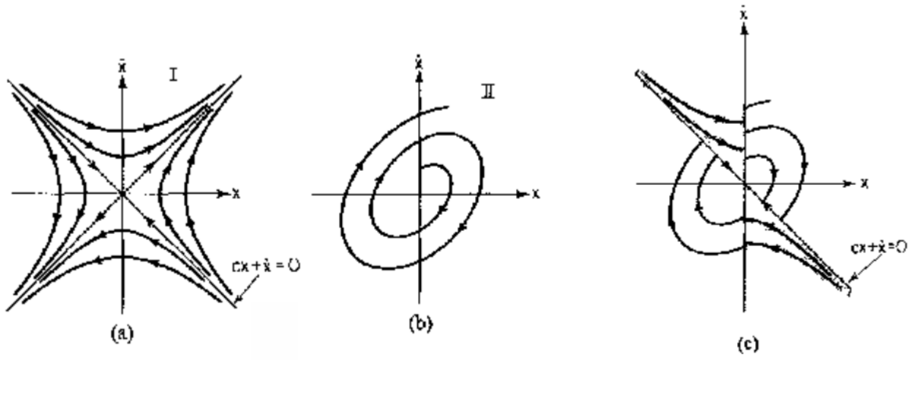

Figure 13: Phase Plan Plot of the system's response to a:
(a) positive feedback, (b) negative feedback, (c) sliding mode command

4.4 Sliding Mode Observers

Now that we know how to design a discontinuous command, let us do the same for discontinuous observers. They are generally used for system with uncertaintenties. For that consider the linear system with uncertaintenties defined by:

$$\begin{cases} \dot{x}(t) = Ax(t) + Bu(t) + f(x,u,t) \\ y(t) = Cx(t) \end{cases} \tag{21}$$

where $x(t) \in \mathbb{R}^n$ is the state vector, $u(t) \in \mathbb{R}^m$ the command vector and $y(t) \in \mathbb{R}^p$ the measured output vector. The function $f : \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R}_+ \to \mathbb{R}^n$ represents the uncertainties and satisfies, which are supposed to be bouned:

$$\exists \rho \in \mathbb{R} : \qquad ||f(x,u,t)|| \leq \rho \qquad \forall x \in \mathbb{R}^n, u \in \mathbb{R}^m, t \in \mathbb{R}_+ \tag{22}$$

In this section we will go through 2 discontinuous observers and then one that combines them to utilize both their advantages and compensate their disadvantages.

4.4.1 Utkin Discontinuous Observer

Utkin presents a method of discontinuous observer design that forces the estimation error to converge.

Let us start by augmenting the state vector to include the output signals in order for us to use them directly from the state vector. The needed transformation is defined by :

$$\begin{bmatrix} x_1(t) \\ y(t) \end{bmatrix} = \begin{bmatrix} I_{n-p} & 0 \\ C_1 & C_2 \end{bmatrix} \begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix} = T_1\, x(t) \tag{23}$$

where $x_1(t) \in \mathbb{R}^{n-p}$ are the $(n-p)$ first elements of the state vector and $T_1 \in \mathbb{R}^{n.n}$ the corresponding transformation matrix. $C_1 \in \mathbb{R}^{p.(n-p)}$ and $C_2 \in \mathbb{R}^{p.p}$ are the corresponding bloc of the output matrix C with $det(C_2) \neq 0$.

The transformed state matrixes are then:

$$\begin{cases} A_1 = T_1\, A\, T_1^{-1} \\ B_1 = T_1\, B \\ C_1 = C\, T_1^{-1} = \begin{bmatrix} 0 & I_p \end{bmatrix} \end{cases} \tag{24}$$

The proposed Utikin discontinuous observer is defined by :

$$\begin{cases} \dot{\hat{x}}_1(t) = A_{11}\hat{x}_1(t) + A_{12}\hat{y}(t) + B_1 u(t) + Lv(t) \\ \dot{\hat{y}}(t) = A_{21}\hat{x}_1(t) + A_{22}\hat{y}(t) + B_2 u(t) - v(t) \end{cases} \tag{25}$$

where $(\hat{x}(t), \hat{y}(t))$ are the estimates of $(x(t), y(t))$ and $L \in \mathbb{R}^{(n-p).p}$ is the observer gain. The discontinuous vector $v(t) \in \mathbb{R}^p$ is defined by :

$$v(t) = M\ sgne(\hat{y}(t) - y(t)) \qquad /M \in R_+ \tag{26}$$

If we suppose the estimation error vectors

$$e_1(t) = \hat{x}_1(t) - x_1(t) \quad ; \quad e_y(t) = \hat{y}(t) - y(t) \tag{27}$$

then their dynamics will be given by:

$$\begin{cases} \dot{e}_1(t) = A_{11}e_1(t) + A_{12}e_y(t) + Lv(t) \\ \dot{e}_y(t) = A_{21}e_1(t) + A_{22}e_y(t) - v(t) \end{cases} \tag{28}$$

Since supposedly the pair (A, C) is totally observable, the pair (A_{11}, A_{21}) is toally observable as well. L is then chosen so that $A_{11} - LA_{21}$ is an exponentionally stable matrix.

Let us define a 2nd transformation :

$$\begin{bmatrix} x_1^{'}(t) \\ y(t) \end{bmatrix} = \begin{bmatrix} I_{n-p} & L \\ 0 & I_p \end{bmatrix} \begin{bmatrix} x_1(t) \\ y(t) \end{bmatrix} = T_2 \begin{bmatrix} x_1(t) \\ y(t) \end{bmatrix} \tag{29}$$

with $x_1^{'}(t) = x_1(t) + Ly(t)$ This 2nd transformation will change the dynamics of the estimation errors to:

$$\begin{cases} \dot{e}_1^{'}(t) = A_{11}^{'}e_1^{'}(t) + A_{12}^{'}e_y(t) \\ \dot{e}_y(t) = A_{21}e_1^{'}(t) + A_{22}^{'}e_y(t) \end{cases} \tag{30}$$

with

$$\begin{cases} e_1^{'}(t) = e_1(t) + Le_y(t) \\ A_{11}^{'} = A_{11} + LA_{21} \\ A_{12}^{'} = A_{12} + LA_{22} - A_{11}^{'}L \\ A_{22}^{'} = A_{22} - A_{21}L \end{cases} \tag{31}$$

Combining the two transformations and using big sliding gain M, a sliding behavior can take place at the output estimation error, then with a good observer gain L the $(n - p)$ states' estimation errors will converge to zero with time. The (p) remaining states are reconstructed then using:

$$\hat{x}_2(t) = C_2^{-1}(y(t) - C_1\hat{x}_1(t)) \tag{32}$$

The inconvenience of this observer relies upon the fact that it is hard to choose a good commutation gain M that generates the desired sliding behavior without excessive commutations.

4.4.2 Walcott and Zak Discontinuous Observer

Wallcot and Zak presents a discontinuous observer that forces its estimation error to converge to zero even in the presence of uncertainties. In this section let us suppose :

$$f(x, u, t) = R\xi(x, t) \tag{33}$$

where $\xi : \mathbb{R}^n \times \mathbb{R}_+ \to \mathbb{R}^q$ is a bounded unknown function and $R \in \mathbb{R}^{n.q}$ is the distribution matrix of these uncertainties.

The proposed observer by Walcott and Zak is defined by:

$$\begin{cases} \dot{\hat{x}}(t) = A\hat{x}(t) + Bu(t) - G(\hat{y}(t) - y(t)) + v(t) \\ \hat{y}(t) = C\hat{x}(t) \end{cases} \tag{34}$$

with the sliding function given by:

$$v(t) = \begin{cases} -\rho\frac{P^{-1}C^TF^TFCe(t)}{||FCe(t)||} & ; \quad FCe(t) \neq 0 \\ 0 & \quad otherwise \end{cases} \tag{35}$$

with

$$e(t) = \hat{x}(t) - x(t) \tag{36}$$

and $F \in \mathbb{R}^{p.q}$ is obtained from the following structural constraint :

$$C^T F^T = PR \tag{37}$$

with an existing Lyapunov SDP matrixes $(P, Q) \in (\mathbb{R}^{n.n})^2$ such that:

$$(A - GC)^T P + P(A - GC) = -Q \tag{38}$$

Let's consider the following Lyapunov function:

$$V(e(t)) = e^T(t)Pe(t) \tag{39}$$

then its derivative along the estimation's error trajectory is:

$$\dot{V}(e(t)) = \dot{e}^T(t)Pe(t) + e^T(t)P\dot{e}(t) \tag{40}$$

from [36], [34] and [21]:

$$\dot{V}(e(t)) = -e^T(t)Qe(t) + 2e^T(t)Pv(t) - 2e^T(t)C^T F^T \xi(x,t) \tag{41}$$

following the commutation function, we distinguish two cases:

case 01. $FCe(t) \neq 0$ then from [35] we get:

$$\dot{V}(e(t)) = -e^T(t)Qe(t) - 2\rho||FCe(t)|| - 2e^T(t)C^T F^T \xi(x,t) \tag{42}$$

since $\xi(x,t)$ is supposedly bounded, the derivative of the Lyapunov function can be upper bounded :

$$\dot{v}(e(t)) \leq -e^T(t)Qe(t) < 0 \tag{43}$$

case 02. $FCe(t) = 0$ then from [35] we get:

$$\dot{v}(e(t)) = -e^T(t)Qe(t) < 0 \tag{44}$$

In both cases, the derivative of the Lyapunov function is negative, which means the states' estimation error converges to zero. The only conditions to verify are then:

- the pair (A, C) is totally observable.
- there exists a pair of Lyapunov matrixes (P, Q) and a matrix F than respects the constraints [37] and [38]

The major inconvenient of Wallcot and Zak's discontinuous observer is that when $FCe(t) = 0$ is happens before $e_y(t) = Ce(t) \neq 0$. This means $F \perp Ce(t)$ hence the convergence of this observer will not be always true.

4.5 Example of Walcott and Zak's observer

Consider the following linear system with uncertain input defined by:

$$\begin{cases} \dot{x}(t) = Ax(t) + Bu(t) + R\xi(x,t) \\ y(t) = Cx(t) \end{cases} \tag{45}$$

with

$$A = \begin{bmatrix} -2 & 1 & 1 \\ 1 & -3 & 0 \\ 2 & 1 & -6 \end{bmatrix} \quad ; \quad B = \begin{bmatrix} 1 \\ 0.5 \\ 0.5 \end{bmatrix} \quad ; \quad C = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 0 & 1 \end{bmatrix} \quad ; \quad R = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$$

First things first we notice that:

```
rank_Ob =

     3

fx >>
```

which means the pair (A, C) is totally observable. Now by solving the constraints [37] and [38], we find that:

```
P =

    0.7300    0.1000    0.0100
    0.1000    0.1000    0.1000
    0.0100    0.1000    0.7300


G =

    2.0400   -2.0100
    1.5400    0.1900
    2.8000   -2.1700


F =

    0.3100    0.5300

fx >>
```

since all of the system is Linear Time Invariant (LTI), all the previous matrixes are constant. Let us calculate the coefficients of the sliding function so that we can pass it between MATLAB and Simulink during simulations :

```
clc
clear

%system
A=[-2 1 1; 1 -3 0; 2 1 -6];
B=[1; 0.5; 0.5];
C=[1 1 1; 1 0 1];
R=[1; 1; 1];
```

```
rank_Ob = rank(obsv(A,C));

%observer
rho=1.5;

P=[0.73 0.1 0.01; 0.1 0.1 0.1; 0.01 0.1 0.73]
G=[2.04 -2.01; 1.54 0.19; 2.8 -2.17]
F=( inv(C*C') * C * P * R )'

%slding function
coeff_1=-rho*inv(P)*(C')*(F')*F*C;
coeff_2=F*C;
```

and we insert them in the function whose bloc is used in figure (14) :

```
function y = fcn(u)

    %u is the state error vector e=x-x_hat
    %y is the sliding mode function v+ for FCe !=0

    coeff_1=[-1.2367  -0.4564  -1.2367;
             -1.4327  -0.5287  -1.4327;
             -1.2367  -0.4564  -1.2367];

    coeff_2=[0.8400  0.3100   0.8400];

y = coeff_1*u / norm(coeff_2*u);
```

The Simulink Scheme for this whole application is illustrated in the following figure:

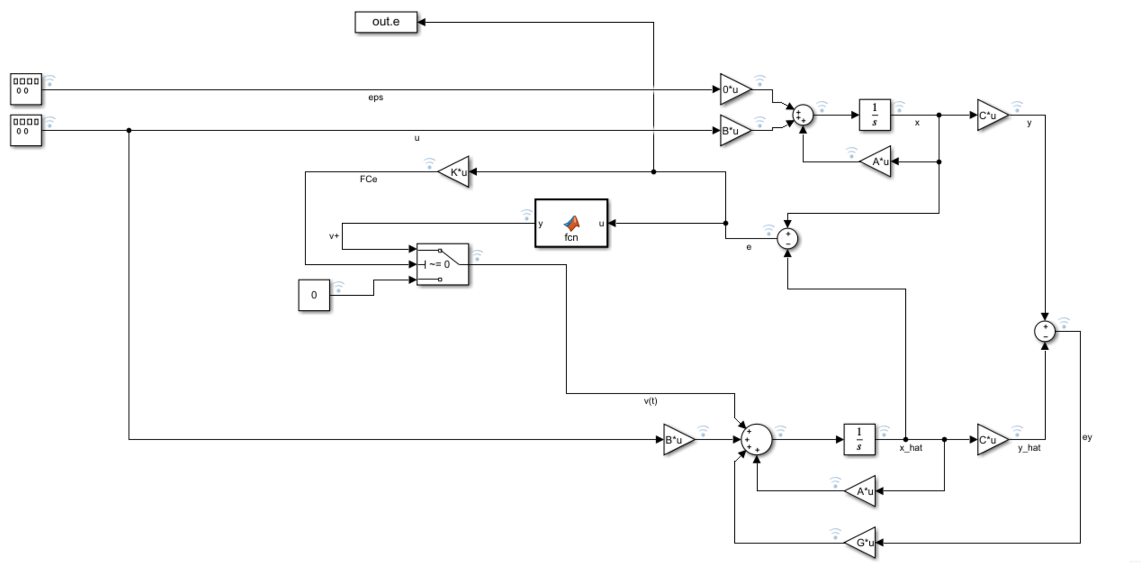

Figure 14: Walcott and Zak Discontinuous Observer

For a random input signal $u(t)$ and bounded input uncertainties $\xi(x,t)$, the simulation results are illustrated in the following figure:

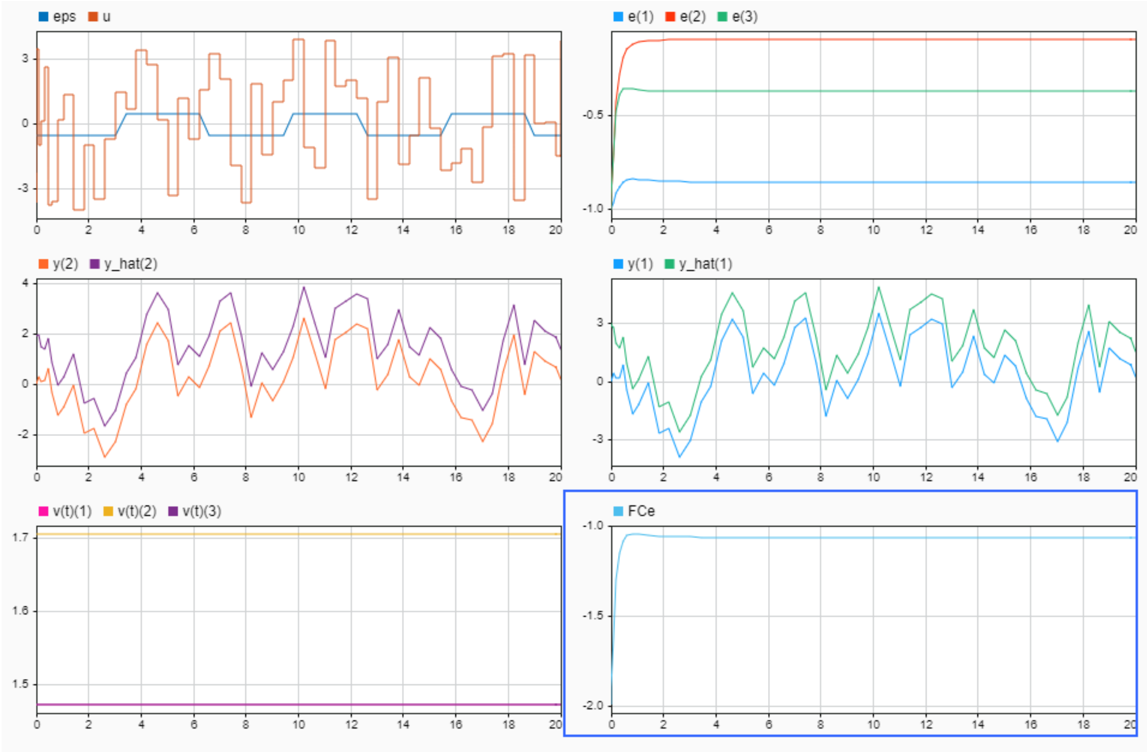

Figure 15: Simulations Results for Walcott and Zak observer

We notice that the error is stable and almost equals zero, and that the gap between the real and estimated output signals is small even in the presence of input uncertainties.

It is also important to note that the quality of convergence relies on the Lyapunov (P, Q) matrixes. Since there are many that satisfies [38], an optimal pair would give an optimal convergence of the estimation error.

5 Applcation : The Cart and Double Pendulum

5.1 TS fuzzy model

Let's take a classic application to combine what we have already seen, the cart and double pendulum inspired from [11].
The system as shown in figure (16) is represented by the set of non-linear state space model shown in (46).

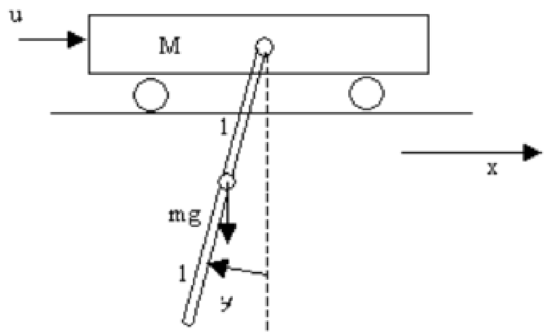

Figure 16: Cart and double pendulum[11]

$$\begin{pmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \end{pmatrix} = \begin{pmatrix} x_2 \\ \frac{-g.sin(x_1)-m.l.a.x_2^2.sin(2x_1)/2}{4l/3-m.l.a.cos^2(x_1)} \\ x_4 \\ \frac{m.a.g.sin(2x_1)/2+4.m.l.a.x_2^2.sin(x_1)/3}{4/3-m.a.cos^2(x_1)} \end{pmatrix} + \begin{pmatrix} 0 \\ \frac{-a.cos(x_1)/2}{4l/3-m.l.a.cos^2(x_1)} \\ 0 \\ \frac{4a/3}{4/3-m.a.cos^2(x_1)} \end{pmatrix} (u - f_c) \tag{46}$$

With : $g = 9.81 m/s^2$, m=2kg, M=8kg, a=1/(m+M), l=0.5m and $f_c = \mu_c sgn(x_4)$ (the friction of the cart with $\mu_c = 0.005$.
To compute the fuzzy model we will use the linearisation method around two local points, $x_1 = 0$ and $x_1 = \pm\frac{\pi}{4}$ and we obtain the two following space models :

$$A_1 = \begin{pmatrix} 0 & 1 & 0 & 0 \\ -17.31 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 1.7312 & 0 & 0 & 0 \end{pmatrix}, B_1 = \begin{pmatrix} 0 \\ -0.1765 \\ 0 \\ 0.1176 \end{pmatrix} \tag{47}$$

$$A_2 = \begin{pmatrix} 0 & 1 & 0 & 0 \\ -14.32 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0.716 & 0 & 0 & 0 \end{pmatrix}, B_2 = \begin{pmatrix} 0 \\ -0.1147 \\ 0 \\ 0.1081 \end{pmatrix} \tag{48}$$

And from the research of experts, we are given the corresponding weights :

$$\omega_1(x_1) = \frac{1 - 1/(1 + e^{-14(x_1 - \pi/8)})}{1 + e^{-14(x_1 + \pi/8)}} \tag{49}$$

$$\omega_2(x_1) = 1 - \omega_1(x_1) \tag{50}$$

So finally our multi-model is represented as :

$$\dot{X} = (\omega_1(x_1)A_1 + \omega_2(x_1)A_2)X + (\omega_1(x_1)B_1 + \omega_2(x_1)B_2)(U - f_c + m.l.x_2^2 sin(x_1)) \tag{51}$$

To check the validity of our model we put both the real model and the fuzzy model is simulink and we plot the states to compare, we obtain the results in figure (17).

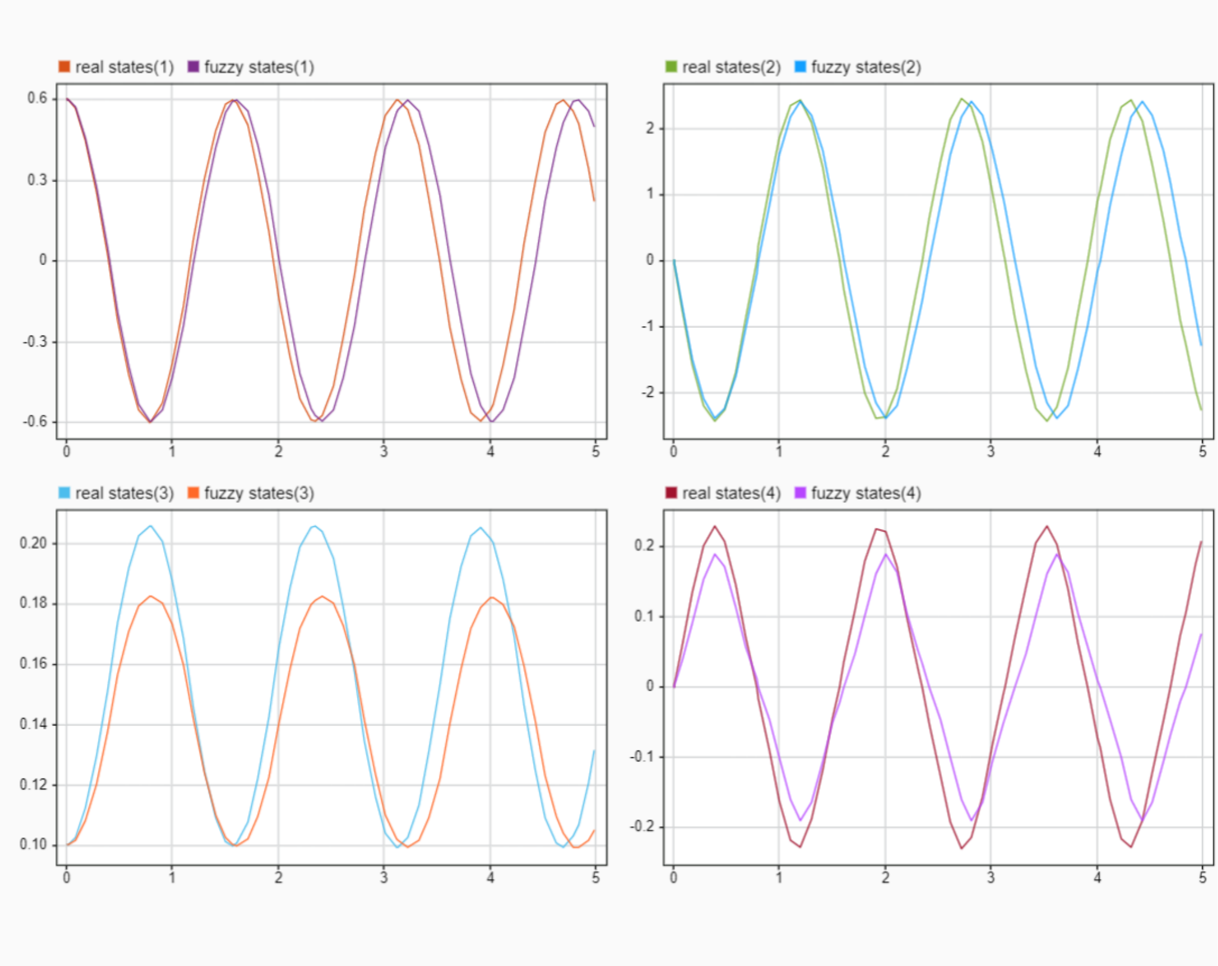

Figure 17: Cart and double pendulum fuzzy states vs real states

We can see that the two set of states are matching and therefore our fuzzy system is indeed accurate.

5.2 Sliding Mode observation

Now that we've built our multi-model TS fuzzy system. Let us add the observing terms to build the Sliding Mode (SM) observer. for that consider the sliding function:

$$M = \begin{cases} -\eta ||P_2|| P_2^{-1} \frac{e_y}{||e_y||} & ; \ \eta \neq 0 \\ 0 & ; \ \eta = 0 \end{cases} \tag{52}$$

that we are going to input in our observer, we obtain :

$$\begin{cases} \dot{\hat{x}} = \sum_{i=1}^{l} w_i(\hat{x}, u) \Big(A_i \hat{x} + B_i u + con_i \Big) + G_l(y - \hat{y}) - G_{nl} M \\ \hat{y} = C\hat{x} \end{cases} \tag{53}$$

it is to note, as indicated in [53] and illustrated in (18) that, for this application, the designed discontinuous observer's terms are added to the integrity of the TS multi-model and not to every local model of it.

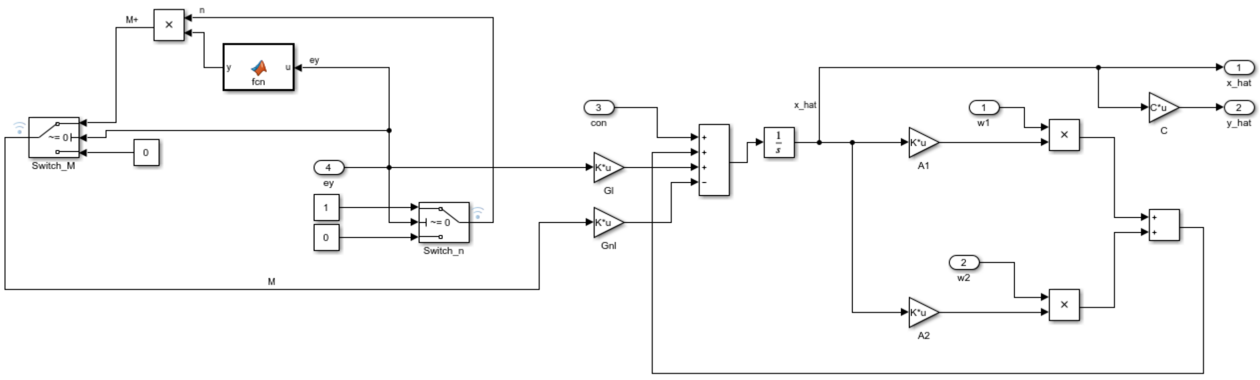

Figure 18: Sliding Mode Observation Scheme

Now that we have constructed the SM observation bloc, let us integrate it to the global scheme as follows:

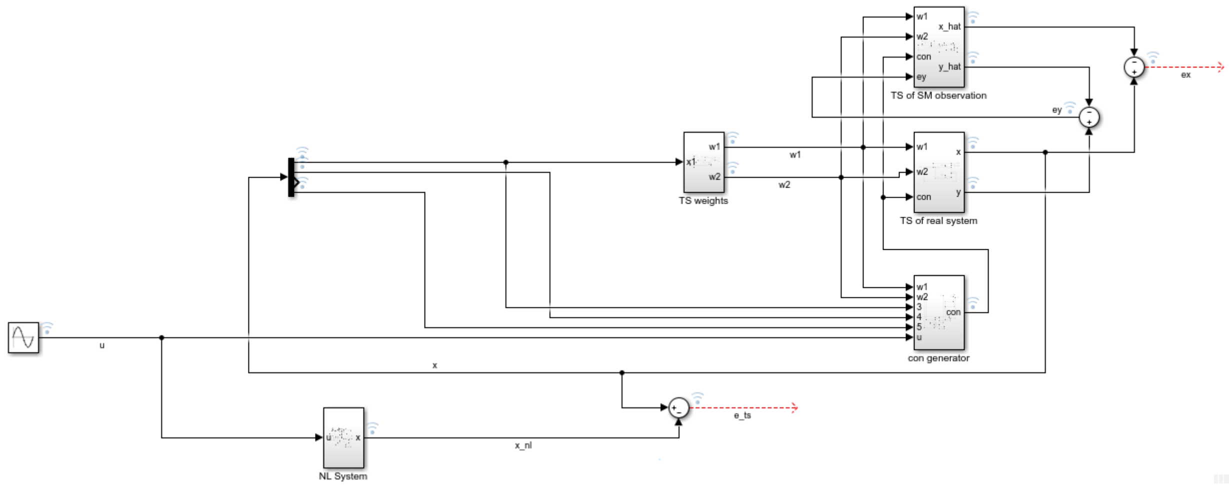

Figure 19: Global Application Scheme

Now if we take $\eta = 1$ and the calculated gains:

$$G_{nl} = \begin{pmatrix} 0.1176 & 0 & 0 \\ 0.3529 & 0 & -0.1765 \\ 0 & 0.1176 & 0 \\ 0 & 0 & 0.1176 \end{pmatrix} \quad ; \quad G_l = \begin{pmatrix} 4 & 0 & -1.5 \\ -14.31 & 0 & -6 \\ 0 & 2 & 1 \\ 1.73 & 0 & 4 \end{pmatrix}$$

and

$$P_2 = \begin{pmatrix} 0.5 & 0 & 0 \\ 0 & 0.25 & 0 \\ 0 & 0 & 0.125 \end{pmatrix}$$

The simulation's results are illustrated as shows the following figure:

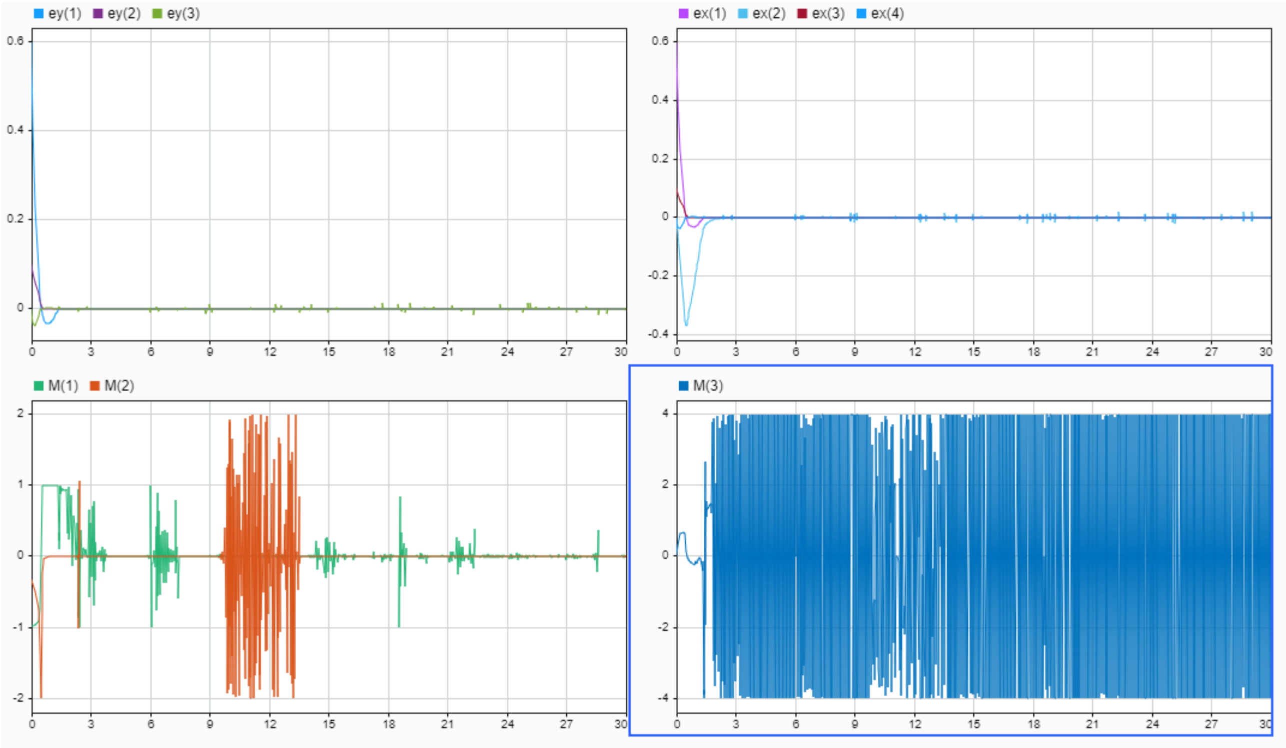

Figure 20: Observations Results : Errors and Commutations

From observing the previous figure, we can comment the following:

- The observation errors converges in a fast manner to zero. With few perturbances here and there but they are reduced instantly, which goes to show the robustness of the designed sliding mode observer.

- For the first state $x_1(t)$, few commutations on the discontinuous function M have been made to ensure the convergence of the associated Lyapunov function described in [40]. These commutations stop as the observation error finally hits zero.

- The same could be said about the commutations for the 2nd state $x_2(t)$ except the fact that they are excessive for just a limited period of time. That can pass since our system is not very sensitive to it.

- For the 3rd state $x_3(t)$, the commutations are excessive, of a large gap, and continuous. Even though this gives good observations, it is not practical as it requires a lot of energy from the system and can destroy it in the long run. That is why an "equivalent control u_e" has been developed and is used in practice. But it is not the topic of this project.

5.3 Command example: LQR

As the goal of our project is reached, we should implement a command to use the obtained results. So, in this section, we will execute a Linear Quadratic (LQR) command based on the SM observer of the TS multi-model of the Cart and Double Pendulum.

As the simulation results showed an unstable behavior (since the stability study is a different notion than that of simple state space systems, that we won't get into, we'll be satisfied with observing the simulations' results), we've opted for an LQR command for stabilization. The optimal gains K_1 and K_2 are obtained by using MATLAB's $lqr()$ function, and that for each local model:

$$K_1 = \begin{pmatrix} -0.3677 \\ -0.5682 \\ 1 \\ 4.5919 \end{pmatrix} \quad ; \quad K_2 = \begin{pmatrix} -0.4287 \\ -0.8009 \\ 1 \\ 4.5393 \end{pmatrix}$$

That we implement in our LQR feedback for every local model: $\bar{u} = -K_i\hat{x} + u$. We obtain the following simulation results:

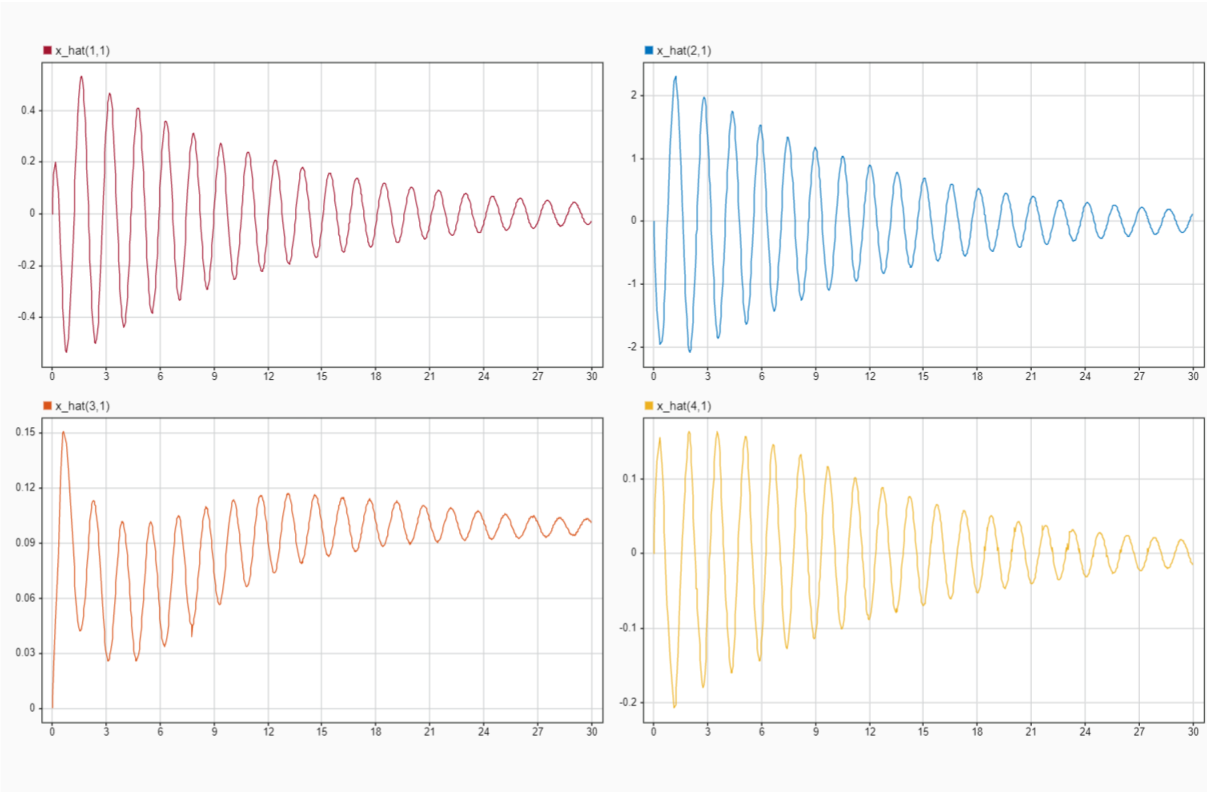

Figure 21: Stabilised states of the double pendulum and cart system

Now we can observe that all of our states became stable. Of course, we can make a better choice of the weighted matrices Q and R to construct optimal feedback by following Bryson's rule for example, but we will be satisfied with the obtained results as the stabilizing goal has been met.

6 Conclusion and perspectives

All along this research, we have seen many concepts and ideas, we have shown the "how" and the "why" of the latter, and we've highlighted the advantages of our methods.

But as any research goes the work is never done, so here are our remarks and perspectives for future works :

- The TS fuzzy model allows us to represent a non-linear system with an interpolation of linear state space models with high precision, this benefits us greatly as it allows for linear approaches to be applied on a non-linear system. To add to that both approaches (linearisation and sector non-linearity) have their own uses , based on the problem itself, which in turn provides a certain variety in computing the multi-modal.

- The fuzzy multi-modal approach has proven it's usefulness once again , but it has to be said that the method itself isn't as easy to execute , as it often requires trial and error, and the results are sometimes constrained by certain specifications. It is already a work in progress , but it needs to be mentioned that finding a more balanced way to compute fuzzy models is needed.

- The fuzzy linearisation approach relies heavily on the choice of it's parameters therefore some experience and data is needed before hands , some works for providing auto-parameters-tuning already exist , so taking advantage of them should be in our sight.

- The fuzzy sector non-linearity approach relies on a certain configuration for non-linear state space models , but under certain circumstances and for some models those specifications are not met so the approach becomes unusable.

- Sliding Mode Control offers robust control strategies using discontinuous commands. The commutation function and logic are chosen well in order to reduce excessive commutations and to indirectly combine the behavior of the piecewise continuous functions into a desired, stable and optimal behavior.

- The Sliding motion is done within a specific surface, where the system can't escape it once it gets into it. But that is what guarantees the desired behavior since this surface is designed for that.

- Aside from sliding mode control commands, there are sliding mode observers. They implement the classical gains to the estimation errors and add a sliding command that will guarantee the convergence of this error in different ways.

- Within the many researches about sliding Mode Observers we can cite Utkin's discontinuous observer. It goes by two transformations in order to include the output signals into the state vector then change the dynamic equation of the estimation error. Then it easily calculate the observer gains by pole placement into the last system. And finally rebuilds the rest of the state vector from the observed portion of it.

- Another Sliding Mode observer is the one proposed by Walcott and Zak. It forces the estimation error to converge to zero by realising the stability of the corresponding quadratic Lyapunov function.

- An interesting application of all the notions and concepts presented in this project is the Fault Tolerant Control. It helps the system keep intact from potential sensor or actuator faults.

- The fault tolerant control relies on using a Fault Detection bloc that detects and estimates the faults, then it used its command in an active approach to compensate the faults right away in two steps: give good estimation of the faults and the resulted faulty state, then command the faulty state to behave as the clean state (as if no faults had occured).

- Lastly we provided examples for most of the concepts presented to visualise the evolution of the system's dynamics depending on the present notion.

7 Footmarks

(1) Crisp input/output : in a fuzzy approach we call a crisp input/ouput the value of the variable in the real domain, in our case it is the time domain.

(2) Fuzzy input/output : it is a 1-D vector where each element represents how much the variable belongs to the corresponding set.

(3) Membership function : it is a function that represents a fuzzy state of a variable, many membership functions exit but the most known are : Triangular, Trapezoidal and Gaussian functions.

(4) Activation function : similar to the membership function, the activation function shows how much our input belongs to a certain model

(5) Convex problem : a convex problem is defined by the two following notions :
Convex ensemble : given an ensemble $E \subset \mathbb{R}$, E is convex if and only if :$\forall \lambda \in [0; 1] \subset \mathbb{R}, \forall (x_1, x_2) \in E^2, \lambda x_1 + (1 - \lambda) x_2 \in E$
Convex function : given a function $f : E \subset \mathbb{R}^n \Rightarrow \mathbb{R}$, with E being a convex ensemble, f is convex if and only if :
$\forall \lambda \in [0; 1] \subset \mathbb{R}, \forall (x_1, x_2) \in E^2, f(\lambda x_1 + (1 - \lambda) x_2) \leq \lambda f(x_1) + (1 - \lambda) f(x_2)$
Convex problem : a convex problem is of the form $min_{x \in E}(f(x))$ for f a convex function and E a convex ensemble.

8 References

(1) Abdelkader Akhenak, Mohammed Chadli, Didier Maquin, José Ragot. State estimation via multiple observer with unknown inputs : Application to the three tank system. 5th IFAC Symposium on Fault Detection, Supervision and Safety for Technical Processes, Safeprocess'2003, Jun 2003, Washington, United States. pp.CDROM. ffhal-00278202

(2) Atef Khedher, Ilyes Elleuch, Kamal Ben Othman : Observer Desing for Non Linear Takagi-Sugeno Fuzzy Systems. Application to Fault Tolerant Control. Studies in Systems, Decision and Control (SSDC), Volume 410.

(3) Abdelkader Akhenak, Mohammed Chadli, Jose Ragot and Didier Maquin ; Fault detection and isolation using sliding mode observer for uncertain Takagi-Sugeno fuzzy model.

(4) Jaime A. Moreno : On Discontinuous Observers for Second Order Systems: Properties, Analysis and Design, Lecture Notes in Control and Information Sciences, Jan 2013. B. Bandyopadhyay et al. (Eds.): Advances in Sliding Mode Control, LNCIS 440, pp. 243–265

(5) Jean Pierre Barbot : Sliding mode control in engineering. Applied Control Engineering, University of Manchester Institue of Science and Technology, Manchester, United Kingdom. Wilfrid Perruquetti. p. cm. — (control engineering)

(6) Kamyar Mehran : Takagi-Sugeno Fuzzy Modeling for Process Control

(7) Ibtissem Abdelmalek , Noureddine Golea : LMI-Based Design of Fuzzy Controller and Fuzzy Observer forTakagi-Sugeno Fuzzy Systems: New Non-Quadratic StabilityApproach

(8) R. Cannon, Dynamics of Physical Systems. Dover Publications, 2003.

(9) Mohammed CHADLI : Stabilité et Commande de Systèmes décrits par des Multimodèles.

(10) Stephen Boyd, Laurent El Ghaoui, Eric Feron, and Venkataramanan Balakrishnan Linear Matrix Inequalities in System and Control Theory

(11) Rainer Palm : Sliding Mode Observer for a Takagi Sugeno Fuzzy system.