University of Coimbra

Doctoral Program in Information Science and Technology

Real Time Learning in Intelligent Systems

Assignment #4 - Process Modelling with Neuro-Fuzzy Systems

Joaquim Pedro Bento Gonçalves Pratas Leitão - 2011150072

May 29, 2017

1 Introduction

Dynamic systems are often described and modelled by different equations, where the outputs in a given time instant (say k) are conditioned by previous outputs (in instants $k-1$, $k-2$, ...) and inputs (in instants k, $k-1$, $k-2$, ...). A system is also said to have *inertia* when the inputs in a given instant only affect its outputs in posterior instants (that is, input at instant k can only influence the output at instants $k+1$, $k+2$, ...).

The current assignment proposes the development of a Sugeno-type Neuro-Fuzzy System (NFS) to model the dynamics of a given process or system with inertia. NFSs of this nature are characterised by mapping their input space to an output space using a series of fuzzy *if-then* rules. In the particular case of Sugeno NFSs the output of each rule is written as a linear combination of the input variables. In simpler Sugeno systems, this linear combination consists in a constant value.

By collecting data containing pairs of the system's input and corresponding output values the rules that define the NFS can be learned in such a way that they describe the system's behaviour.

In other words, exploring collected data from the system the mentioned fuzzy rules can be learned, resulting in the development of a Sugeno NFS that models the dynamics of the desired process or system, as it is the objective of this assignment.

The remainder of this document is organised as follows: Section 2 presents the system to be modelled in this work; Section 3 describes the methodology followed throughout the project; Sections 4 and 5 cover the main steps of the work, presented in the Methodology section. Finally, Section 6 concludes the document.

2 Modelled System

The system to be modelled in this work can be defined by the following transfer function:

$$G(s) = \frac{2}{s^3 + 5s^2 + 6.75s + 2.25}$$

Since the provided transfer function features 3 poles and no zeros - therefore it has more poles than zeros - the presented system is a third-order system. It can be proved that the provided system has memory of 3 instants in both the inputs and outputs and *inertia* (meaning that inputs at a given time instant only influence future outputs). In this sense, the output of the system at any time instant k can be written as:

$$y(k) = f(y(k-1), y(k-2), y(k-3), u(k-1), u(k-2), u(k-3))$$

where $y(k)$ represents the system's output at instant k and $u(k)$ represents the system's input at instant k.

3 Methodology

The project and development of a Neuro-Fuzzy System can be a complex and extensive task. Nevertheless, in the scope of the current work, the following two main steps will be considered:

1. Learning Stage, where a model of the system is learned using data collected from the system. This stage comprises three main sub-tasks:

    - *Data collection*, where pairs of system's *input-output* data are collected, which will be used as a labelled dataset during the learning. Collecting a good set of *input-output* data is critical in the final outcome. These data should be representative of the dynamics of the system, making the input and output vary over the entire permissible domain. A good technique is to use a random input sequence that forces the output to move through the entire space.

    - *Fuzzy rules initialisation.* An initial estimate for the NFS's rules is obtained by applying a *clustering* algorithm to the labelled dataset collected in the previous point. Common algorithms used at this stage include subtractive, c-means and fuzzy c-means clustering.

    - *Fuzzy rules optimisation.* A further optimisation of the fuzzy rules computed in the previous point is usually performed, aiming to reduce the modelling error. An *ANFIS* structure is commonly used along with either a *backpropagation* or *hybrid* optimisation method (although other alternatives maybe considered at this point).

2. Assessment Stage, where different input signals are provided to both the developed model and the real system with the objective of comparing the real and simulated outputs. The ultimate goal of this task is to certify that the developed model of the system can accurately describe its behaviour.

4 Learning Stage

As presented in the previous section, the goal of the *Learning Stage* is to train a Neuro-Fuzzy System to model the target system's behaviour based on pairs of *input-output* signals collected from the target system.

Therefore the first step of this stage is related with data collection. Since no access to the *"real", physical* system is available data will be collected with *Simulink*, a graphical programming and simulation environment available with the *MATLAB* software.

4.1 Data Collection

Usually, the process of collecting data from a given system assumes that the collection is made at a given frequency, meaning that samples are collected with a given interval. It is recommended that the sampling interval (also referred to as discretization interval) be less than the inverse of the smaller pole of the system. As such, a sampling interval equal to one-third of the inverse of the smaller pole of the system was used.

After setting the sampling interval, a discrete version of the system's transfer function must be computed. Resorting to the *MATLAB* software environment, such computation can be performed with the function *c2dm*. The following discrete transfer function was obtained:

$$G(z) = \frac{0.1079z^2 + 0.1413z + 0.009003}{z^3 - 0.8794z^2 + 0.1766z - 0.006738}$$

In this sense, the block diagram presented in Figure 1 was created. A random number generator is used to produce random inputs to the system to which the corresponding outputs are obtained. Both the generated inputs and outputs are stored in memory. The system dynamics are simulated during a total simulation time of 500 instants. The generated data was further divided into a *training* and *testing* dataset, with a $70 - 30$ division.

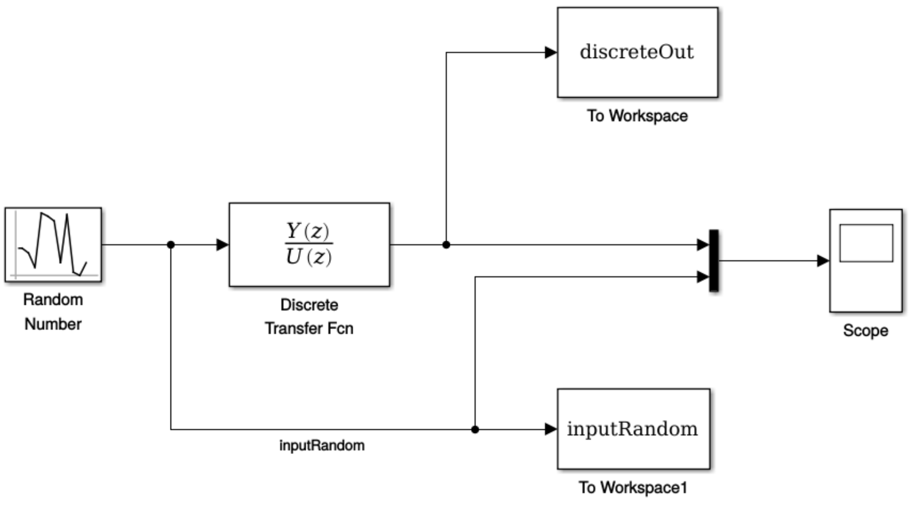


Figure 1: Block diagram developed in Simulink to generate training and testing data.

The generated input signal and correspondent system's output is presented in Figure 2:

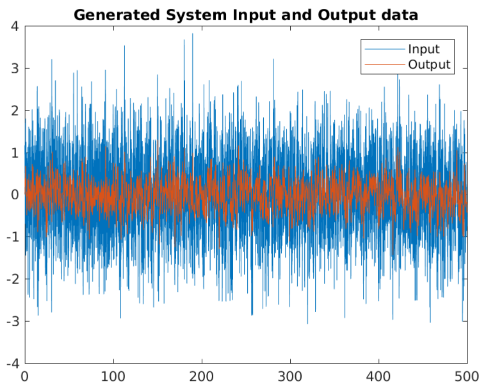


Figure 2: Generated input signal and system output.

4.2 Fuzzy Rules Initialisation

Having generated input and output samples representative of the dynamics of the system being modelled, the actual learning process can be started. As mentioned in Section 3, in an initial step, estimations for the fuzzy rules are produced as a result of a clustering process of the collected data. Two clustering algorithms were studied at this point, featuring the following parameters:

- Subtractive Clustering, where a *cluster influence range* of 0.5 was defined; s *squash factor* of 1.25 was considered, along with and an accept ratio of 0.5 and reject ratio of 0.15.
- Fuzzy C-Means Clustering, considering an exponent for the fuzzy partition matrix of 2.0, maximum number of iterations of 100, minimum improvement in objective function between two consecutive iterations of 10^{-5} and the number of clusters being automatically estimated using subtractive clustering.

In terms of implementation, the initial estimation of the fuzzy rules was performed using the *MATLAB's genfis* function, properly specifying the training data, the type of clustering to be performed and the respective parameters. The last two arguments (type of clustering and clustering parameters) were defined by creating a proper *genfisOptions* object.

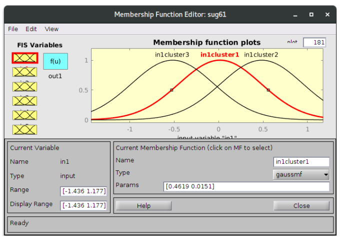
The remainder of the current subsection presents a detailed view of the fuzzy inference systems obtained after the initial rules estimation.

Table 1: Fuzzy Inference Systems obtained after the initial rules estimation.

Cluster Technique	**AND Method**	**OR Method**	**Order**
Subtractive	Product	Probabilistic OR of fuzzified input values	1
Fuzzy C-Means	Product	Probabilistic OR of fuzzified input values	1

Table 2: Input Membership Functions - Subtractive Clustering.

Input Number	**Membership Function Type**	**Parameters**
in1	Gaussian Curve (gaussmf)	$\sigma = [0.4619; 0.4619; 0.4619]$ $C = [0.0151; 0.4848; -0.5181]$
in2	Gaussian Curve (gaussmf)	$\sigma = [0.4834; 0.4834; 0.4834]$ $C = [0.0252; 0.4486; -0.5278]$
in3	Gaussian Curve (gaussmf)	$\sigma = [0.4834; 0.4834; 0.4834]$ $C = [-0.0177; 0.2468; -0.5145]$
in4	Gaussian Curve (gaussmf)	$\sigma = [1.2188; 1.2188; 1.2188]$ $C = [-0.3147; 0.6766; 0.03]$
in5	Gaussian Curve (gaussmf)	$\sigma = [1.2129; 1.2129; 1.2129]$ $C = [-0.0699; 0.5373; 0.1799]$
in6	Gaussian Curve (gaussmf)	$\sigma = [1.2129; 1.2129; 1.2129]$ $C = [-0.0133; 0.4407; -1.1277]$

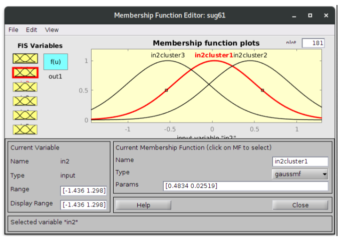


Current Variable
Name in1
Type input
Range [-1.436 1.177]
Display Range [-1.436 1.177]

Current Membership Function (click on MF to select)
Name in1cluster1
Type gaussmf
Params [0.4619 0.0151]

Ready

(a) Membership Function for input *in*1

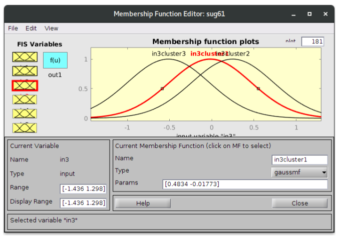


Current Variable
Name in2
Type input
Range [-1.436 1.298]
Display Range [-1.436 1.298]

Current Membership Function (click on MF to select)
Name in2cluster1
Type gaussmf
Params [0.4834 0.02519]

Selected variable "in2"

(b) Membership Function for input *in*2

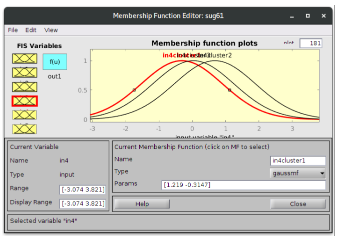


Current Variable
Name in3
Type input
Range [-1.436 1.298]
Display Range [-1.436 1.298]

Current Membership Function (click on MF to select)
Name in3cluster1
Type gaussmf
Params [0.4834 -0.01773]

Selected variable "in3"

(c) Membership Function for input *in*3

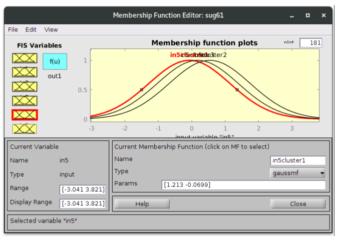


Current Variable
Name in4
Type input
Range [-3.074 3.821]
Display Range [-3.074 3.821]

Current Membership Function (click on MF to select)
Name in4cluster1
Type gaussmf
Params [1.219 -0.3147]

Selected variable "in4"

(d) Membership Function for input *in*4

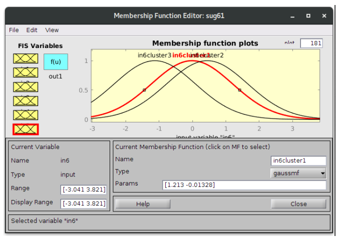


Current Variable
Name in5
Type input
Range [-3.041 3.821]
Display Range [-3.041 3.821]

Current Membership Function (click on MF to select)
Name in5cluster1
Type gaussmf
Params [1.213 -0.0699]

Selected variable "in5"

(e) Membership Function for input *in*5



Current Variable
Name in6
Type input
Range [-3.041 3.821]
Display Range [-3.041 3.821]

Current Membership Function (click on MF to select)
Name in6cluster1
Type gaussmf
Params [1.213 -0.01328]

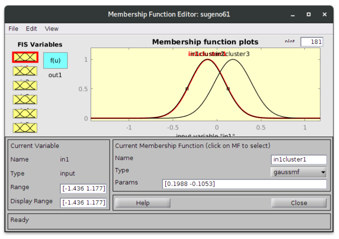
Selected variable "in6"

(f) Membership Function for input *in*6

Figure 3: Input Membership Functions for the initial rules obtained with the Subtractive Clustering method.

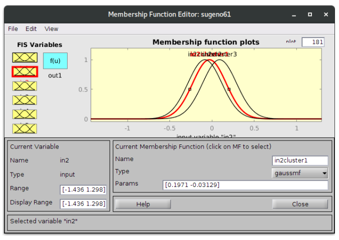
Table 3: Input Membership Functions - Fuzzy C-Means Clustering.

Input Number	**Membership Function Type**	**Parameters**
in1	Gaussian Curve (gaussmf)	$\sigma = [0.1988; 0.1984; 0.2096]$ $C = [-0.1053; -0.1094; 0.1818]$
in2	Gaussian Curve (gaussmf)	$\sigma = [0.1971; 0.1999; 0.2]$ $C = [-0.03129; -0.0849; 0.0888]$
in3	Gaussian Curve (gaussmf)	$\sigma = [0.198; 0.1983; 0.1987]$ $C = [-0.0214; -0.0277; 0.0248]$
in4	Gaussian Curve (gaussmf)	$\sigma = [0.54; 0.5405; 0.5408]$ $C = [-0.0318; 0.0294; -0.0245]$
in5	Gaussian Curve (gaussmf)	$\sigma = [0.6063; 0.5544; 0.5526]$ $C = [-0.6561; 0.3178; 0.287]$
in6	Gaussian Curve (gaussmf)	$\sigma = [0.5256; 0.5695; 0.5786]$ $C = [-0.0714; -0.5193; 0.5411]$



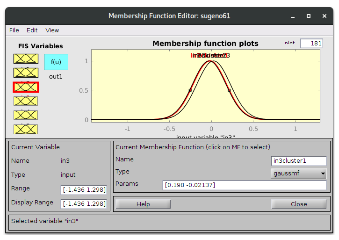
Current Variable
Name in1
Type input
Range [-1.436 1.177]
Display Range [-1.436 1.177]
Current Membership Function (click on MF to select)
Name in1cluster1
Type gaussmf
Params [0.1988 -0.1053]
Ready

(a) Membership Function for input $in1$



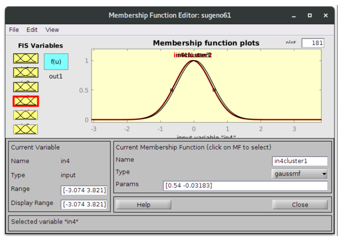
Current Variable
Name in2
Type input
Range [-1.436 1.298]
Display Range [-1.436 1.298]
Current Membership Function (click on MF to select)
Name in2cluster1
Type gaussmf
Params [0.1971 -0.03129]
Selected variable "in2"

(b) Membership Function for input $in2$



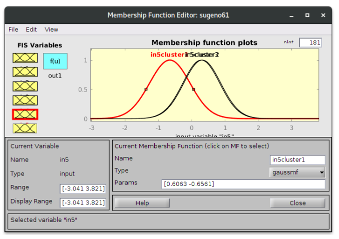
Current Variable
Name in3
Type input
Range [-1.436 1.298]
Display Range [-1.436 1.298]
Current Membership Function (click on MF to select)
Name in3cluster1
Type gaussmf
Params [0.198 -0.02137]
Selected variable "in3"

(c) Membership Function for input $in3$



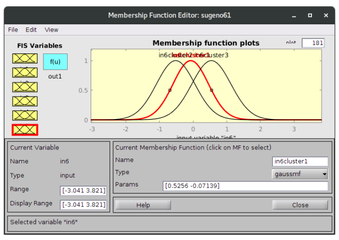
Current Variable
Name in4
Type input
Range [-3.074 3.821]
Display Range [-3.074 3.821]
Current Membership Function (click on MF to select)
Name in4cluster1
Type gaussmf
Params [0.54 -0.03183]
Selected variable "in4"

(d) Membership Function for input $in4$



Current Variable
Name in5
Type input
Range [-3.041 3.821]
Display Range [-3.041 3.821]
Current Membership Function (click on MF to select)
Name in5cluster1
Type gaussmf
Params [0.6063 -0.6561]
Selected variable "in5"

(e) Membership Function for input $in5$



Current Variable
Name in6
Type input
Range [-3.041 3.821]
Display Range [-3.041 3.821]
Current Membership Function (click on MF to select)
Name in6cluster1
Type gaussmf
Params [0.5256 -0.07139]
Selected variable "in6"

(f) Membership Function for input $in6$

Figure 4: Input Membership Functions for the initial rules obtained with the Fuzzy C-Means Clustering method.

4.3 Fuzzy Rules Optimisation

In the final step of the learning stage an ANFIS structure is used to optimise the estimations for the fuzzy rules obtained in the previous subsection. Two optimisation methods were considered at this point: the well-known *Backpropagation* and an *Hybrid* method.

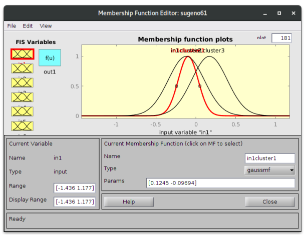
In terms of implementation, the optimisation of the inference rules was performed using *MATLAB*'s *anfis* function, properly specifying the training data, the type of optimisation method and the number of epochs[1]. The last two arguments (type of optimisation and the number of epochs) were defined by creating a proper *anfisOptions* object.

Analysing the input membership functions of the optimised Neuro-Fuzzy Systems (NFS) no changes were registered in the NFSs where subtractive clustering was used to initialise the inference rules; however, visible changes in the mean and deviation of the gaussians were registered when the hybrid optimisation method was used in a NFSs where the fuzzy c-means clustering was used to initialise the inference rules. Table 4 presents the new input membership functions.

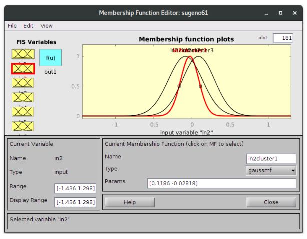
[1]A value of 200 was defined for the epoch number.

Table 4: Final Input Membership Functions - Fuzzy C-Means Clustering and Hybrid Optimisation Method.

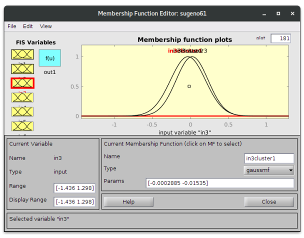
Input Number	**Membership Function Type**	**Parameters**
in1	Gaussian Curve (gaussmf)	$\sigma = [0.1245; 0.2246; 0.222]$ $C = [-0.0969; -0.1212; 0.1701]$
in2	Gaussian Curve (gaussmf)	$\sigma = [0.1186; 0.2227; 0.2101]$ $C = [-0.0282; -0.0854; 0.0858]$
in3	Gaussian Curve (gaussmf)	$\sigma = [-0.0002885; 0.2296; 0.2105]$ $C = [-0.01535; -0.0312; 0.0238]$
in4	Gaussian Curve (gaussmf)	$\sigma = [0.478; 0.5723; 0.5579]$ $C = [-0.03009; 0.028; -0.0258]$
in5	Gaussian Curve (gaussmf)	$\sigma = [0.5866; 0.6097; 0.5833]$ $C = [-0.6511; 0.2963; 0.2749]$
in6	Gaussian Curve (gaussmf)	$\sigma = [0.4937; 0.582; 0.5829]$ $C = [-0.0727; -0.5151; 0.5403]$


Membership Function Editor: sugeno61
File Edit View
FIS Variables
f(u)
out1
Membership function plots
plot 181
in1cluster1 in1cluster2 in1cluster3
input variable "in1"
Help
Close
Current Variable: Name in1; Type input; Range [-1.436 1.177]; Display Range [-1.436 1.177]
Current Membership Function (click on MF to select): Name in1cluster1; Type gaussmf; Params [0.1245 -0.09694]
Ready

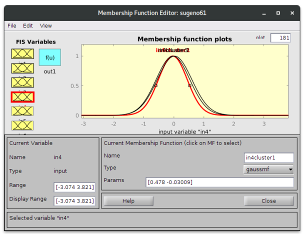
(a) Membership Function for input $in1$


Membership Function Editor: sugeno61
File Edit View
FIS Variables
f(u)
out1
Membership function plots
plot 181
in2cluster1 in2cluster2 in2cluster3
input variable "in2"
Help
Close
Current Variable: Name in2; Type input; Range [-1.436 1.298]; Display Range [-1.436 1.298]
Current Membership Function (click on MF to select): Name in2cluster1; Type gaussmf; Params [0.1186 -0.02818]
Selected variable "in2"

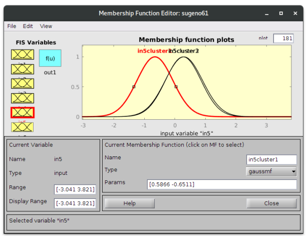
(b) Membership Function for input $in2$


Membership Function Editor: sugeno61
File Edit View
FIS Variables
f(u)
out1
Membership function plots
plot 181
in3cluster1 in3cluster2 in3cluster3
input variable "in3"
Help
Close
Current Variable: Name in3; Type input; Range [-1.436 1.298]; Display Range [-1.436 1.298]
Current Membership Function (click on MF to select): Name in3cluster1; Type gaussmf; Params [-0.0002885 -0.01535]
Selected variable "in3"

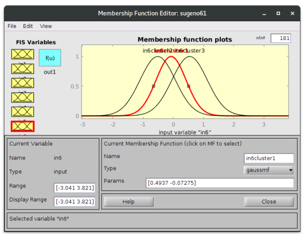
(c) Membership Function for input $in3$


Membership Function Editor: sugeno61
File Edit View
FIS Variables
f(u)
out1
Membership function plots
plot 181
in4cluster1 in4cluster2 in4cluster3
input variable "in4"
Help
Close
Current Variable: Name in4; Type input; Range [-3.074 3.821]; Display Range [-3.074 3.821]
Current Membership Function (click on MF to select): Name in4cluster1; Type gaussmf; Params [0.478 -0.03009]
Selected variable "in4"

(d) Membership Function for input $in4$


Membership Function Editor: sugeno61
File Edit View
FIS Variables
f(u)
out1
Membership function plots
plot 181
in5cluster1 in5cluster2 in5cluster3
input variable "in5"
Help
Close
Current Variable: Name in5; Type input; Range [-3.041 3.821]; Display Range [-3.041 3.821]
Current Membership Function (click on MF to select): Name in5cluster1; Type gaussmf; Params [0.5866 -0.6511]
Selected variable "in5"

(e) Membership Function for input $in5$


Membership Function Editor: sugeno61
File Edit View
FIS Variables
f(u)
out1
Membership function plots
plot 181
in6cluster1 in6cluster2 in6cluster3
input variable "in6"
Help
Close
Current Variable: Name in6; Type input; Range [-3.041 3.821]; Display Range [-3.041 3.821]
Current Membership Function (click on MF to select): Name in6cluster1; Type gaussmf; Params [0.4937 -0.07275]
Selected variable "in6"

(f) Membership Function for input $in6$

Figure 5: Input Membership Functions after Hybrid optimisation when the Fuzzy C-Means Clustering method was used in the rules initialisation.

Once the optimisation was complete, the performance of the final Neuro-Fuzzy Inference System was assessed using the test data. Table 5 presents a summary of the results of the optimisation process. The error metric used to compute both the training and testing error was the *Root Mean Squared Error* (RMSE).

Table 5: Final results of the optimisation process.

Cluster Technique	**Optimisation Type**	**Minimum Training Error**	**Test Error**	**AND Method**	**OR Method**	**Number Rules**	**Order**
Subtractive	Backpropagation	1.6813e-16	0.0054	prod	probor	3	1
Subtractive	Hybrid	4.0359e-07	3.769e-07	prod	probor	3	1
Fuzzy C-Means	Backpropagation	8.1942e-17	0.007	prod	probor	3	1
Fuzzy C-Means	Hybrid	2.3616e-07	3.3348e-05	prod	probor	3	1

5 Assessment Stage

Having developed and trained several NFSs to model the behaviour of the system presented in Section 2 more diverse and intensive tests must be performed. With the goal of determining how close the developed NFSs were from the actual system a *Simulink* model was developed and is presented in Figure 6.

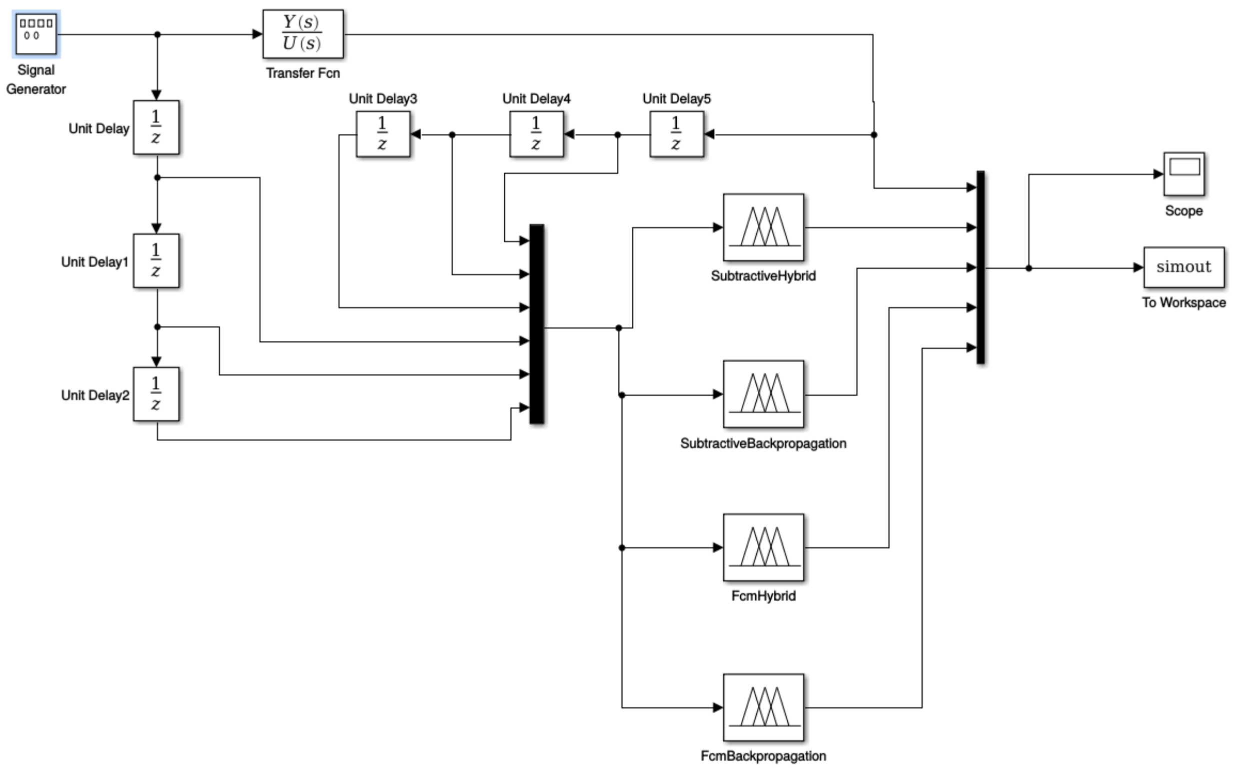


Figure 6: Block diagram developed in Simulink to assess how close the developed NFSs were to the actual system to be modelled.

In this exercise, a signal generator produces an input signal that is provided to the system being modelled (contained in the *Transfer Func* block, described by the transfer function presented in Section 2) and to each one of the four NFSs developed in the previous stage. The five computed outputs were then inspected and plotted in the *Scope* block.

The following functions were used in the *Signal Generator* block to produce the input signals:

- *Sawtooth* wave, with an amplitude of 5, units defined in *Hertz* and a frequency set to the sampling interval whose computation was described in Section 4.1.
- *Sine* wave, with an amplitude of 5, units defined in *Hertz* and a frequency set to the sampling interval whose computation was described in Section 4.1.
- *Square* wave, with an amplitude of 5, units defined in *Hertz* and a frequency set to the sampling interval whose computation was described in Section 4.1.

Table 6 presents the *Root Mean Squared* (rmse) errors registered with each developed Neuro-Fuzzy System, for each input wave.

Table 6: Root Mean Squared Errors of the different Neuro-Fuzzy Systems for the three input waves.

Wave Type	**Cluster Technique**	**Optimisation Type**	**RMSE**
Sawtooth	Subtractive	Hybrid	0.7438
Sawtooth	Subtractive	Backpropagation	0.7386
Sawtooth	Fuzzy C-Means	Hybrid	0.7437
Sawtooth	Fuzzy C-Means	Backpropagation	0.7553
Sine	Subtractive	Hybrid	0.9182
Sine	Subtractive	Backpropagation	0.9138
Sine	Fuzzy C-Means	Hybrid	0.9181
Sine	Fuzzy C-Means	Backpropagation	0.9305
Square	Subtractive	Hybrid	1.2836
Square	Subtractive	Backpropagation	1.2727
Square	Fuzzy C-Means	Hybrid	1.2834
Square	Fuzzy C-Means	Backpropagation	1.3008

Analysing the results presented in Table 6 it can be seen that, for the same input values, the four different NFSs produce very similar outputs. Such a statement is supported by the fact that *rmse* values for the same wave type are very similar.

The *Sawtooth* wave recorded the smallest *rmse* values, with error values between 0.73 and 0.75. Even though the values are small, a closer inspection to the produced outputs presented in Figure 7 reveals the existence of an antecipation in the NFSs response: changes in the NFSs outputs occur slightly before the change in the outputs of the system described the transfer function (represented in blue).

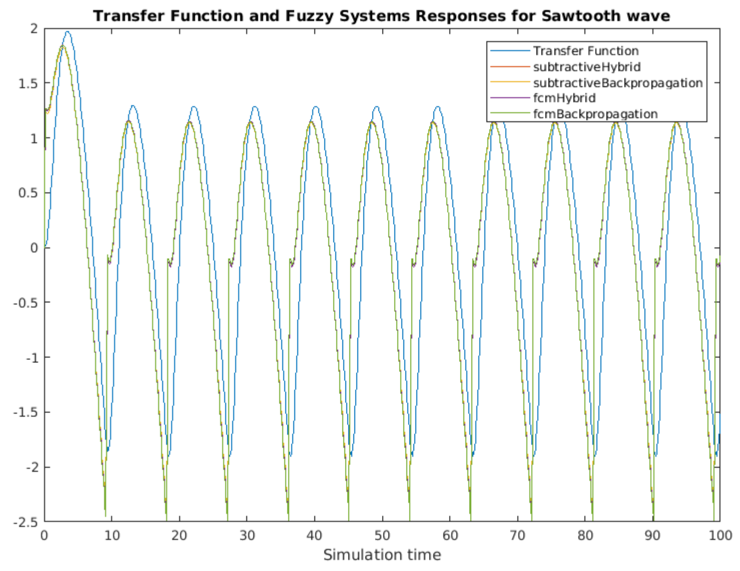


Figure 7: Outputs of the system being modelled and the four NFSs developed when using the *Sawtooth* wave to produce the inputs.

Indeed, such a behaviour can be identified for both the square and sin waves: whenever the input signal changes its direction the developed NFSs always react before the actual system being modelled. Figures 8 and 9 illustrate this phenomena.

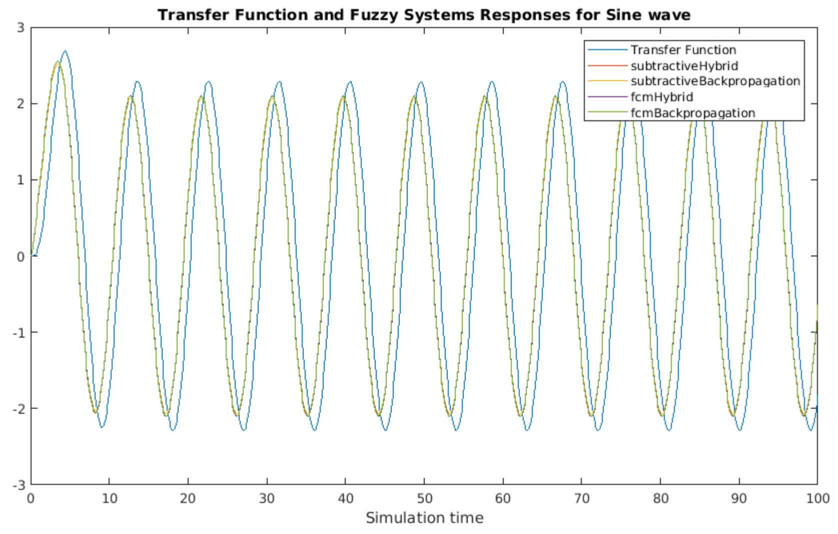


Figure 8: Outputs of the system being modelled and the four NFSs developed when using the *Sine* wave to produce the inputs.

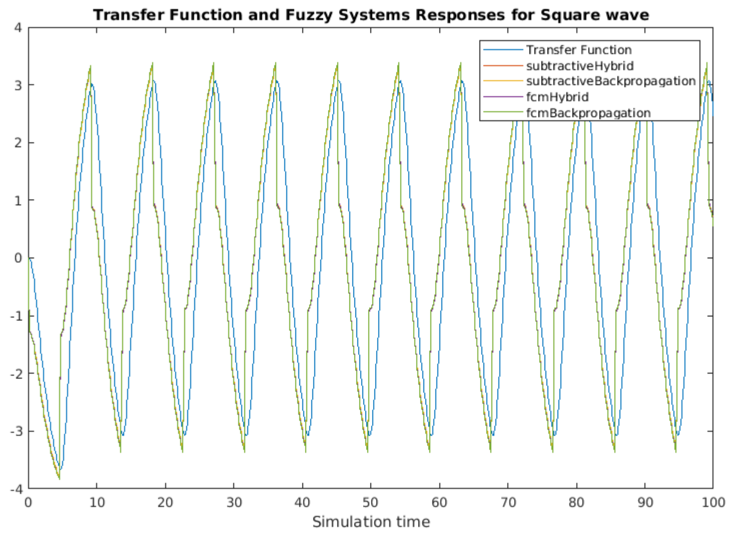


Figure 9: Outputs of the system being modelled and the four NFSs developed when using the *Square* wave to produce the inputs.

6 Conclusion

The current work focused on the task of modelling dynamic processes by means of Neuro-Fuzzy Systems. In the course of this work, such fuzzy logic controllers were projected and developed, based on the dynamics of a given system, described by a transfer function.

In an initial stage, a random input signal was provided to the *"real"* system (that is, the one to be modelled) and its corresponding outputs were recorded, creating a dataset to be used in the development of the Neuro-Fuzzy Systems. This development consisted in two main stages:

At first, estimates to the fuzzy rules that govern the operation of these systems were obtained by means of a clustering technique - two distinct clustering algorithms were considered: *Subtractive* and *Fuzzy C-Means* clustering.

In a second step, an optimisation of the initial rules estimates was performed using an *ANFIS* structure. Two optimisation methods were also considered at this point - *Backpropagation* and an *Hybrid* optimisation method.

After the training, an initial performance assessment was conducted with part of the data initially collected. As presented in Table 6, the use of the *Backpropagation* optimisation technique registered the lowest errors during training; however substantially worst results were registered during testing, suggesting a lack of generalisation for this optimisation method. On the other hand, the *Hybrid* optimisation method was still capable of producing acceptable results in training which remained more stable during testing, suggesting a more general (and therefore desirable) model.

Based on the information mentioned, models developed using the *Hybrid* optimisation method could be seen as good candidates to model the dynamics of the system in question. Nevertheless, as described and discussed in Section 5, when subjecting the system to input signals with different characteristics no major differences between the four trained

NFSs could be identified. Indeed, **FIXME: Falar aqui do atraso?? O que dizer ao certo???**

FIXME: Confirmar esta última frase - Ver resposta Blue