Machine Learning
Assignment 3 - Fuzzy Control Systems

André Carvalho
andrecarvalho@student.dei.uc.pt
no. 2019216156

Paulo Cortesão
paulocortesao@student.dei.uc.pt
no. 2019216517

Class PL2, Group 1

December 2022

Contents

1 Part A

1.1 Introduction

This report aims to describe the process undertaken in the design of fuzzy systems for generic control actions that would be resilient to actuator and load disturbances. For this purpose, the *MATLAB Simulink* package was used, as well as the respective *Fuzzy Logic Toolbox*. The steps taken in the design of the relevant circuits and the results obtained shall be registered here; and a brief discussion of the results is included.

1.2 Construction of the Experimental Setup

Considering the statement and the slides for the course, an experimental setup was devised in order to allow for the correct functioning of the fuzzy system, while favouring the obtention of relevant data for drawing conclusions. The experimental setup in 1 was obtained.

In order to test the various types of fuzzy systems and compare them, the *fuzzyLogicDesigner* application was used, and 9-, 25- and 49-rule Mammdani systems were devised. After this, the conversion tool included in this software was used to obtain the Sugeno version of these systems.

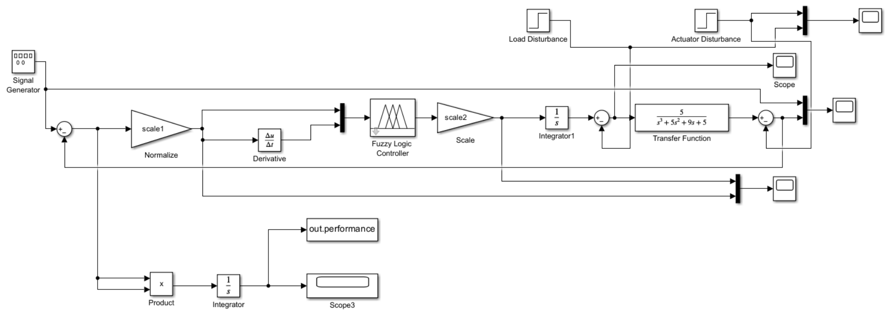


Figure 1: Base circuit for input generation

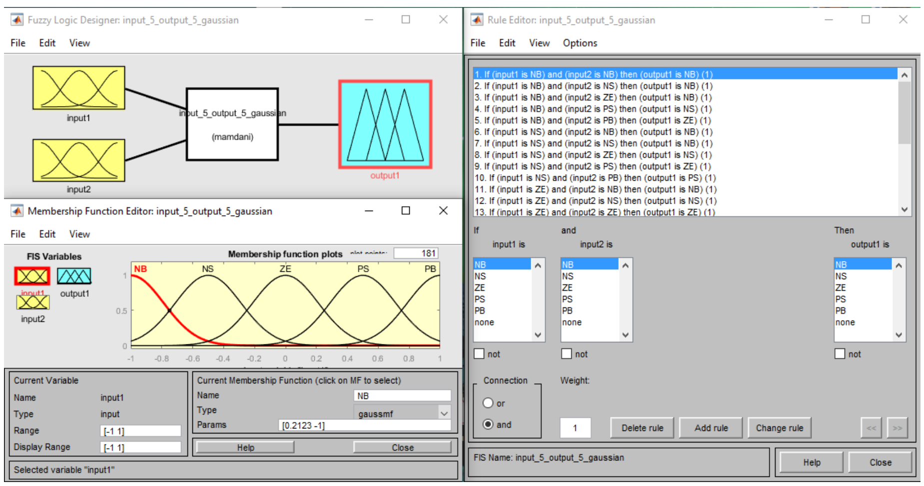


Figure 2: Editing GUI for building a Mamdani system

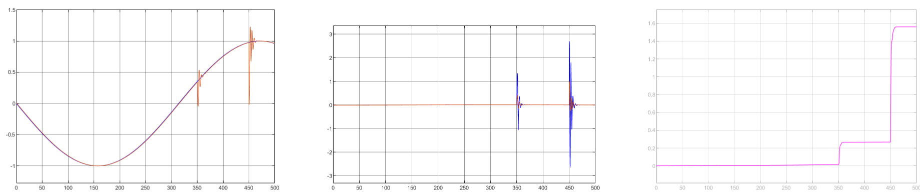
1.3 Results

The following table shows the results obtained in the various systems tested. The values for the input factor ($factor1$) and the actuator factor ($factor2$) are somewhat disperse due to the fact that a heuristic search was applied to reach the optimal systems in terms of overshoot and SSE. It is important to note that the waves used as reference had a frequency of 0.01 radians and that the simulation time was 500 seconds in all trials, except the ones where the saw-tooth function was used as a reference (in order to allow for the visualization of the error in the step part of the function, the simulation time was increased to 700 seconds). The load disturbance occurred at $t = 350s$ and the actuator disturbance was programmed to take place from $t = 450s$ on.

Ctrl. Type	No. rules	Memb. Func	Ref. Func	$factor1$	$factor2$	SSE
mamdani	25	Gaussian	square	0.60	7.125	13.967485
mamdani	25	Gaussian	square	0.60	2.00	11.402784
mamdani	25	Gaussian	square	0.60	0.50	25.857485
mamdani	25	Gaussian	square	0.60	3.00	9.670112
mamdani	25	Gaussian	square	0.60	5.00	8.752938
mamdani	25	Gaussian	square	0.60	6.00	8.467538
mamdani	25	Gaussian	square	0.60	2.00	11.701809
mamdani	9	Gaussian	square	0.50	15.00	10.003034
mamdani	9	Gaussian	sawtooth	0.50	15.00	6.440346
mamdani	9	Gaussian	sawtooth	0.50	15.00	37.125619
sugeno	25	Gaussian	square	0.60	2.00	10.642364
sugeno	25	Gaussian	square	0.60	4.00	8.777417
sugeno	25	Gaussian	sin	0.60	4.00	1.576537
sugeno	25	Gaussian	sin	0.60	2.00	1.987883
mandani	25	Gaussian	sin	0.60	2.00	2.340998
mandani	25	Gaussian	sin	0.60	4.00	1.652202
mandani	9	Gaussian	square	0.60	4.00	13.158141
mandani	9	Gaussian	square	0.65	6.00	10.967875
mandani	9	Gaussian	sin	0.65	6.00	9.913078
mandani	9	Gaussian	sin	0.65	8.00	8.422704
mandani	9	Gaussian	sin	0.65	10.00	7.395736
mandani	9	Gaussian	sin	1.00	7.00	4.873267
sugeno	9	Gaussian	sin	1.00	4.00	1.560859
sugeno	9	Gaussian	sin	1.00	3.00	1.668104
sugeno	9	Gaussian	sin	0.90	4.00	1.581950
sugeno	9	Gaussian	square	0.70	4.00	8.914960
sugeno	9	Gaussian	square	0.60	3.00	9.873211
sugeno	9	Gaussian	square	0.80	4.00	8.906654
mandani	25	Gaussian	square	0.60	6.00	8.488450
sugeno	25	Triangular	sin	0.60	4.00	5.647989
sugeno	25	Triangular	sin	0.60	2.00	1.906491
sugeno	25	Triangular	square	0.60	6.00	73.505326
sugeno	25	Triangular	square	0.60	2.00	10.498060
sugeno	25	Triangular	sawtooth	0.60	2.00	10.724236
mandani	25	Triangular	sawtooth	0.60	2.00	11.135762
sugeno	49	Gaussian	square	0.60	2.00	10.418216
sugeno	49	Gaussian	square	0.60	3.00	9.053528
sugeno	49	Gaussian	square	0.60	4.00	8.818157
mamdani	49	Gaussian	square	0.60	2.00	11.106783
mamdani	49	Gaussian	sawtooth	0.60	2.00	10.846574
mamdani	49	Gaussian	sin	0.60	2.00	1.898087
sugeno	49	Triangular	square	0.60	2.00	10.421963
sugeno	49	Triangular	square	0.60	1.50	11.744757
mamdani	49	Triangular	sin	0.60	2.00	1.882157
mamdani	49	Triangular	square	0.60	2.00	10.622498
mamdani	49	Triangular	sawtooth	0.60	2.00	10.725070
sugeno	9	Triangular	square	0.60	2.00	25.984022
mamdani	9	Triangular	sawtooth	0.60	1.00	48.977624
mamdani	9	Triangular	sin	0.60	1.00	44.708761
mamdani	9	Triangular	square	0.60	1.00	54.292702

Total number of controllers delivered: 12

Some pictures of the graphs obtained in the process are included. The graphs plotting the errors do not include the derivative due to the fact that there were instants in which its magnitude was too large, leading to poor visualization of the data.

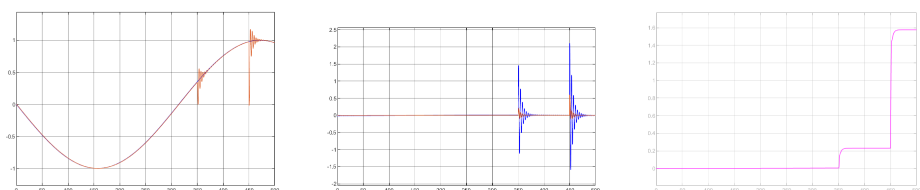


(a) Reference vs. output

(b) Control vs. error

(c) Evolution of SSE

Figure 3: Sugeno system with 9 rules

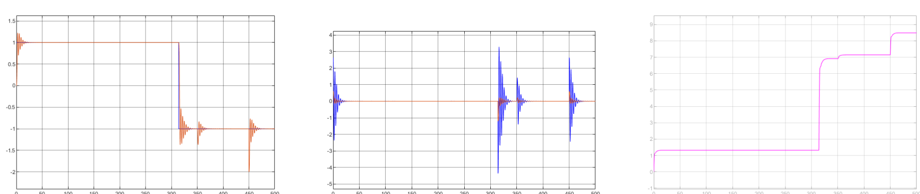


(a) Reference vs. output

(b) Control vs. error

(c) Evolution of SSE

Figure 4: Sugeno System with 25 rules using a sine function as reference

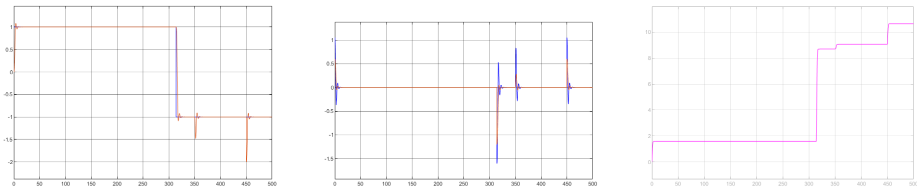


(a) Reference vs. output

(b) Control vs. error

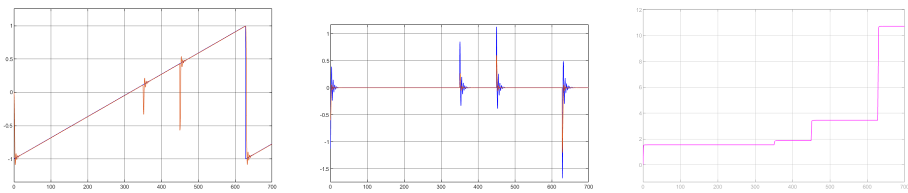
(c) Evolution of SSE

Figure 5: Mamdani system with 25 rules using a square function as reference



(a) Reference vs. output (b) Control vs. error (c) Evolution of SSE

Figure 6: Sugeno system with 25 rules using a square function as reference



(a) Reference vs. output (b) Control vs. error (c) Evolution of SSE

Figure 7: Sugeno system with 25 rules using a sawtooth function as reference

1.4 Discussion

From the results obtained, it is possible to draw some conclusions on the functioning of fuzzy control systems:

- There is a compromise between the error and the overshoot in the values obtained by the controller, which is regulated by the value of the factor affecting the control action. As can be seen from the result table, an excessively small factor leads to a very slow convergence from the controller to the reference value, which is reflected in a very slow stabilizing time (about 20 s), and a very high SSE. On the other hand, when the factor is higher, about 5, the SSE is smaller, even though the convergence is still somehow slow, and overshooting phenomena are observed. Considering intermediate values for the second factor, it can be seen from 5 and 6 that choosing a higher number will lead to more overshooting and a lower SSE. On the other hand, choosing a lower factor 2 will reduce overshooting with a slight increase in SSE.

- The Sugeno controllers consistently outperform the Mamdani controllers. This means that fuzziness is more beneficial in the controller input than in the respective output. It is possible that the coefficients and normalization algorithms associated to the TSK controller have greater precision than the defuzzification methods applied in the Mamdani systems.

- The number of rules also affects the performance of the system. Even though good results can be found with 9, 25 and 49 rules, controllers with 25 and 49 rules normally have smaller SSE values when tested. This reflects the finer granularity allowed by a greater number of rules, as the input and output space are covered with more specific rules.

- As can be seen from the table, the type of membership function in the input affects the result obtained. In this case, the triangular function yields poorer results than the gaussian function. This can be explained by the fact that the values close to 0 are not well represented in the non-zero classes, which means that the system will produce near-zero output and will not improve its error. The gaussian membership function avoids this problem with a smoother descent throughout the set support.

- The performance using sine as the reference function is always higher than when other functions are used. This can be shown by the multiple experiences performed, as sine always yielded better results when the fittest values for the factors were used. This reference even had approximately the same performance using both 9 or 25 rules as can be seen in 3 and 4.

- The actuator disturbance produces a greater error in this specific system than the load disturbance. Even though it was possible to overcome both difficulties in the system, it can be seen in the graphs (especially the ones related to the sine function, as it does not have any other discontinuities) that the actuator disturbance causes an error that is 3 to 4 times greater than the error caused by the load disturbance.

2 Part B

2.1 Introduction

In this part of the work, a TSK fuzzy system had its parameters trained in order to fit a given reference function. For this, the Neural Fuzzy Toolbox in MATLAB was used. The results obtained and conclusions drawn from the experimental setup shall be discussed here.

2.2 Experimental Setup

In order to produce data for training the fuzzy system, a small circuit was devised. The transfer function associated with the system was converted to discrete values and the system's input and output data were processed so they would fit the purpose of the project: in this case, after the two temporal series were obtained, seven series were extracted from them by shifting them in time and removing some elements in the extremes. This allowed for the generation of the dataset that the ANFIS would be based on:

After this, the circuit for testing was designed. Based on previous circuits and on the course's slides, the following was obtained:

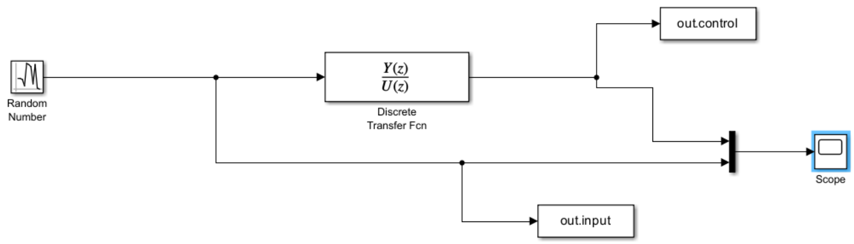


Figure 8: Base circuit for input generation

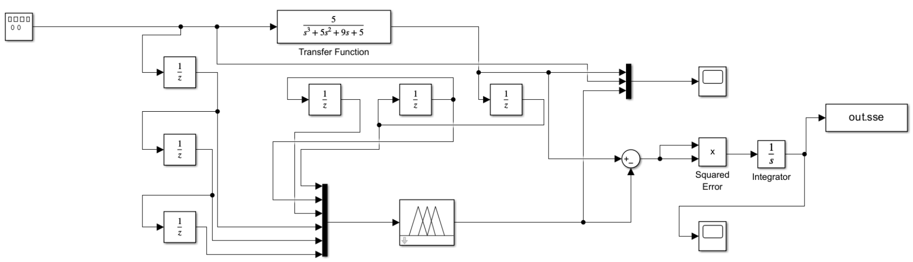


Figure 9: Base circuit for input generation

2.3 Results

The following table shows the results obtained with the various fuzzy inference systems trained. In all the cases, the simulation time was 100 seconds and the frequency of the signal was 0.1 radians. A total of 6 FIS were tested, across three different reference functions.

Clustering	Rules	Func.Type	Optimization	SSE
FCM	3	Square	Backpropagation	0.0360097
FCM	3	Sawtooth	Backpropagation	0.0119186
FCM	3	Sine	Backpropagation	0.00957796
FCM	3	Square	Hybrid	0.0212037
FCM	3	Sawtooth	Hybrid	0.00783114
FCM	3	Sine	Hybrid	0.00261489
Grid Partition	64	Square	Backpropagation	82.2343
Grid Partition	64	Sawtooth	Backpropagation	25.5466
Grid Partition	64	Sine	Backpropagation	41.954
Grid Partition	64	Square	Hybrid	0.00299705
Grid Partition	64	Sawtooth	Hybrid	0.00804986
Grid Partition	64	Sine	Hybrid	0.00299705
Subtractive	3	Square	Hybrid	0.0212037
Subtractive	3	Sawtooth	Hybrid	0.00783091
Subtractive	3	Sine	Hybrid	0.00261481
Subtractive	3	Square	Backpropagation	0.0280667
Subtractive	3	Sawtooth	Backpropagation	0.0140023
Subtractive	3	Sine	Backpropagation	0.0110555

The graphical result for the best three results, one for each reference function (square, sin and sawtooth, respectively), are shown below. In these cases, it is impossible to see the system output graph, since it practically coincides with the model output (which also reflects the quality of the models generated).

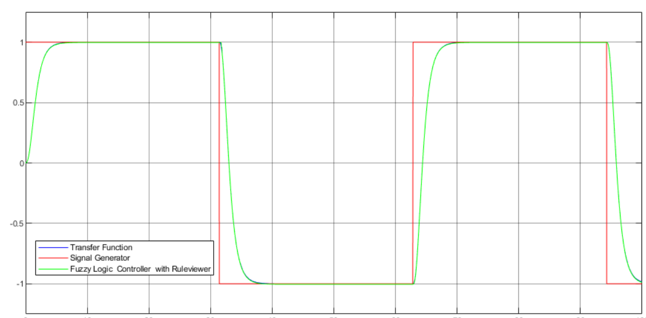


Figure 10: Grid partition with Hybrid optimization

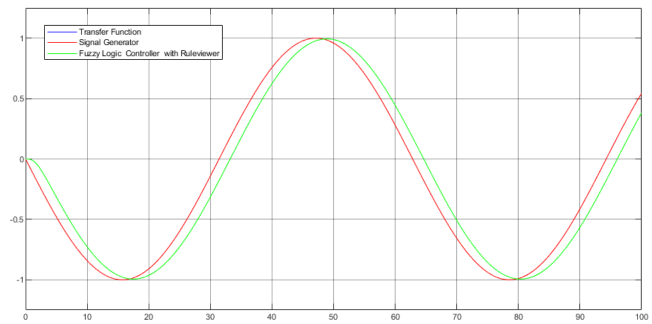


Figure 11: Subtractive clustering with Hybrid optimization

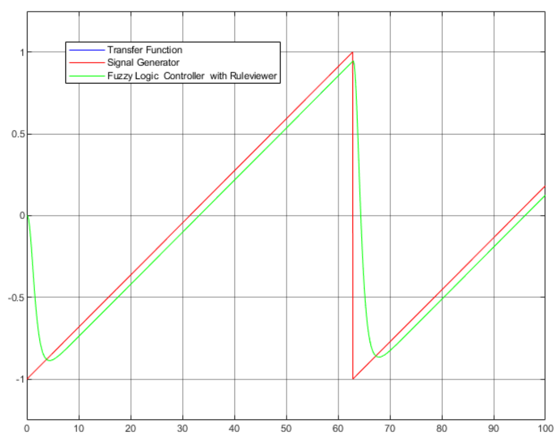


Figure 12: Subtractive clustering with Hybrid optimization

2.4 Discussion

Some conclusions can be drawn from the results and the experimental process in general:

- Even with a few rules (for example, 3), it is possible to use these models to replicate the functioning of systems with great precision ($SSE < 0.01$). Thus, this experiment reveals their broad applicability in system simulation and in control systems.

- Using the default parameters for the clustering methods, there is great similarity between the sum of squared errors. Still, there is an improvement in performance when a 64-rule, grid partition-based system is used. The greater number of rules allows for a better description of the system. However, the training and testing time associated with these systems is about four times greater, so the choice of the type of system to use should bear this in mind.

- As far as the training method is concerned, it appears that the hybrid method outperforms backpropagation. This shows that, in this case, the update of the coefficients associated with the rule antecedents is more suitable using the least squares method.

- It was also noted that using Backpropagation with Grid partition clustering was the only case that the controller did not yield good results. One example using a square reference function can be seen as follows:

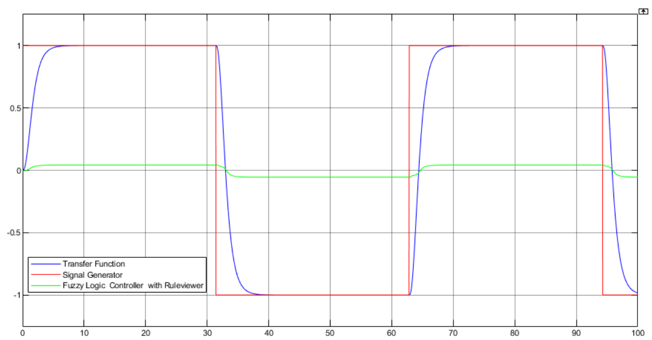


Figure 13: Grid partition clustering with Backpropagation

3 Conclusion

In this work, low-error systems and high-fidelity fuzzy models were devised, and their testing errors were determined. The potential of these methods for modelling systems where a transfer function is unknown became clear, given the low SSE verified in most of the cases.

As such, it is considered that the objectives defined for the work have been achieved.