Soft Computing Assignment 2

Mustafa Caner Sezer — 504191123

June 11, 2021

1 Question 1

In the first question, a neuron based nonlinear PID Controller is given to be used to control a paper making process.The controller parameters and the initial conditions were stated. A procedure which consists 7 different steps were created to control the system for different paper densities. The controller was tested under two different reference signals and in this paper the results were given for only 100 g/m^2. Nevertheless, the results for 80 g/m^2 and 120 g/m^2 were also validated but not mentioned in this paper.

1.1 Section A

In the first section, given reference signal was used as the reference. The algorithm was created and can be seen in the attachments. It basically consists 7 different steps that were mentioned in the fundamental informations pdf. After implementing the algorithm, the results were obtained and can be seen in the figures 1a, 1b, 1c, 1d.

It is obvious that if the change occurs in the reference signal, the control signal responds to the change extremely aggressive so the controller must be able to generate this control signal. The designer should consider this if she wants to implement it in the real life. However as a result, the controller could be able to control the system under the desired reference signal.

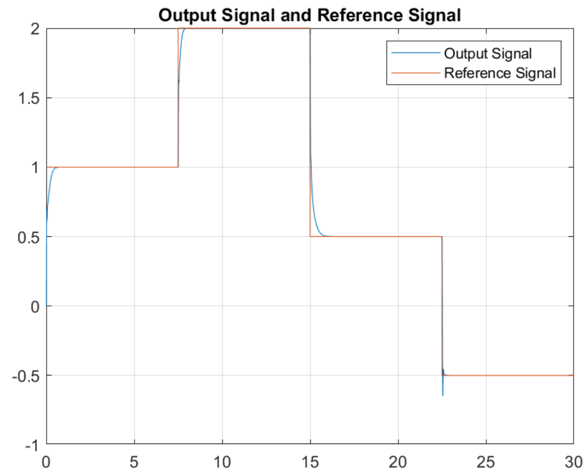
1.2 Section B

In the section B, the only difference was the reference signal. A sinusoidal reference signal was used as the reference. After configuring the algorithm, the results were obtained and can be seen in the figures 2a, 2b, 2c, 2d.

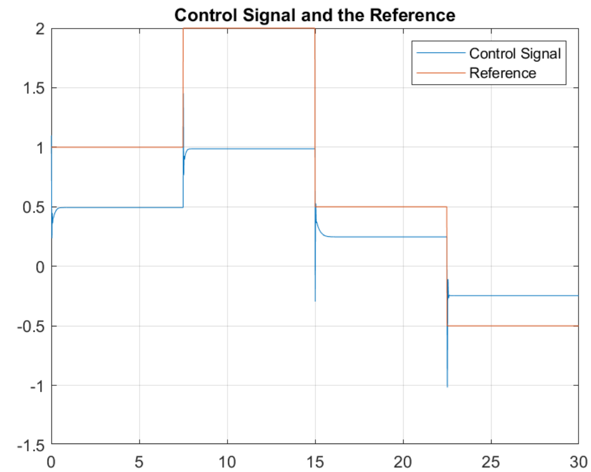
As well as the first reference, the system could also be able to follow the reference under the desired reference signal.

2 Question 2

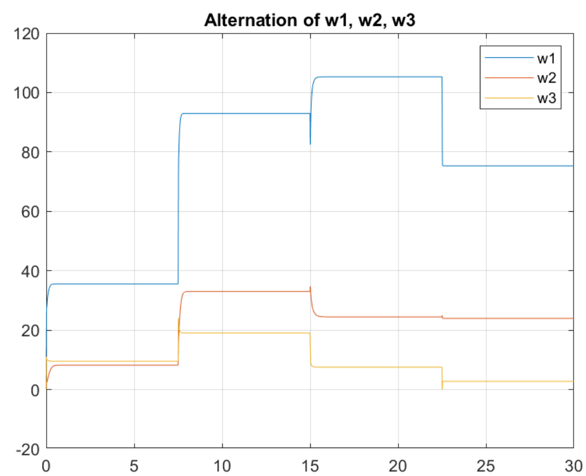
In the second question, a CSTR system was aimed to be controlled with a fuzzy PID controller. The fuzzy PID controller takes two inputs as the error and the change of the error and produces an output. To create an integral term, the output of the controller is integrated and multiplied by the term β. Also to fulfill all of the PID terms, u is multiplied by the other term α. Since the system is discrete, an integrator was added to the end and the output of the fuzzy logic pid controller was obtained.



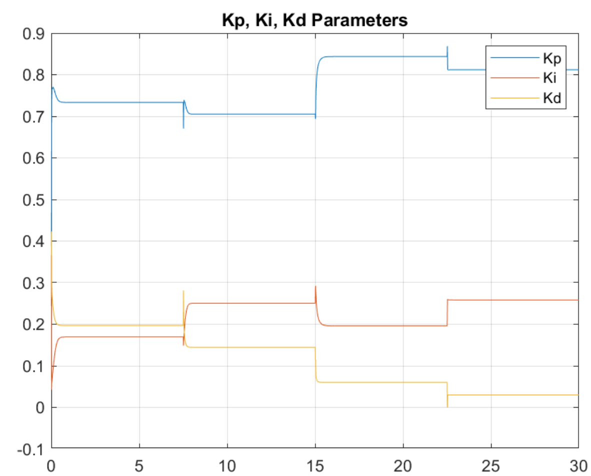
(a) Output vs Reference



(b) Control signal vs Reference

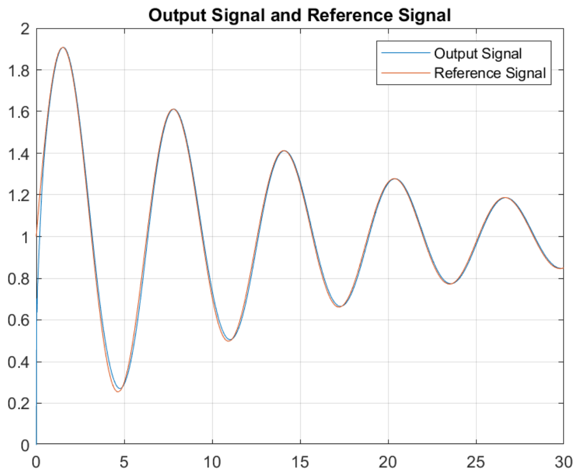


(c) Alternation of w_1,w_2 and w_3

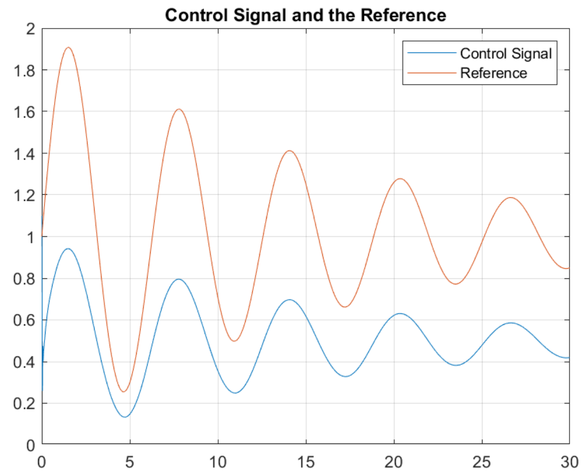


(d) Change of the PID parameters

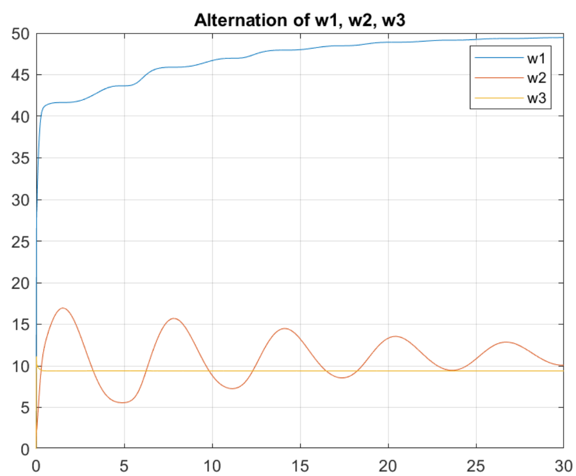
Figure 1: The plots for the first reference signal



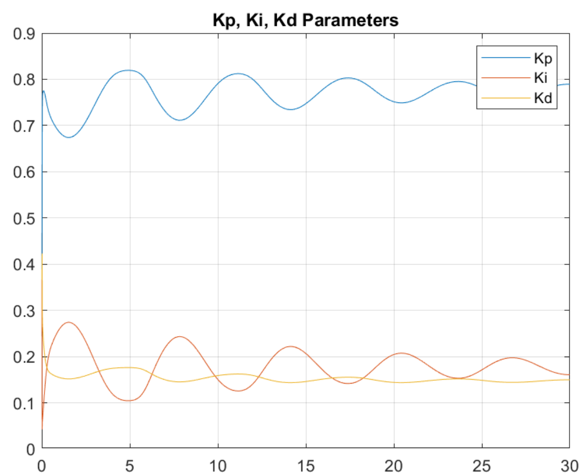
(a) Output vs Reference



(b) Control signal vs Reference



(c) Alternation of w_1,w_2 and w_3



(d) Change of the PID parameters

Figure 2: The plots for the second reference signal

The x_3 was selected as the output as it is mentioned in the reference paper. A Sugeno type FLC structure was created. Four different membership functions for the error and the change of the error was created with the cores -1, -0.4, 0, 0.4, 1. The output is selected as a crisp type output. Seeing that the output is a crisp type output, the control signal can be calculated as the firing strength of the rules times the value of the corresponding rule. As the nature of the sugeno type FLC structure, at most four different rules can be fired simultaneously. The weighted average of the fired rules can be used as the deffuzification phase.

2.1 Section A

In the first section, the fuzzy logic controller algorithm was created in Simulink. The structure can be seen in the figure 3. The system was given as a discrete type system. Therefore, the integral and derivative terms were created discrete type as well as the system and the sampling time was defined as 0.1. The algorithm works as follows,

1. Calculates the error and the change of the error
2. Multiply them by the scaling terms and uses saturation to ensure that they are within the interval $[-1, 1]$.
3. The subsystem in the middle calculates the membership values of e, $\dot{e}$ and calculates the output by multiplying the membership values and the firing strength.
4. Output scaling factors α, β and discrete integral term
5. There is another term to calculate the u_{FPID}
6. The control signal is applied to the system and x_3 was used to feedback.

The given code was used for the CSTR system and in the CSTR block, the initial conditions and the constants were stated. The simulink file can be found in the attachments.

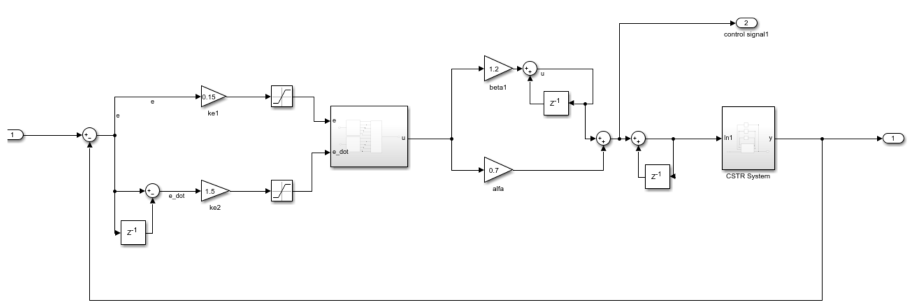


Figure 3: FLC type PID Controller Structure

2.2 Section B

To simulate the system, the given values $K_e = 0.15$, $K_{de} = 1.5$, $\alpha = 0.7$ and $\beta = 1.2$ was used. The simulation results can be seen in the figure 4. The system response starts from 0.31 due to the inital conditions as it supposed to be. As a consequence, the FLC type PID controller can handle a highly nonlinear system like CSTR.

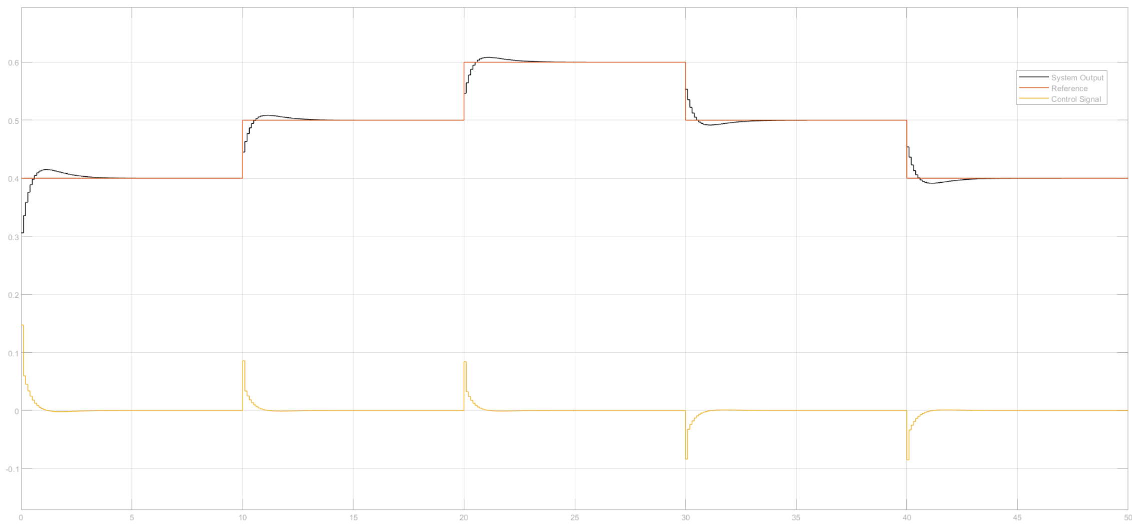


Figure 4: The reference signal, system response and the control signal

2.3 Section C

In the section C, the parameters of the fuzzy logic controller were determined by the genetic algorithm so that to minimize the cost function with different lambda values given as follows.

$$J = 0.5 \sum_{n=1}^{\infty} (r_n - y_n)^2 + 0.5\lambda \sum_{n=1}^{\infty} (u_n - u_{n-1})^2$$

To simulate genetic algorithms, a simulink file like figure 4 was created and the cost function value was transferred to the workspace like in the figure 5. First, the parameters K_e, K_{de}, α and β were defined as global parameters and a cost function was created in MATLAB. With the help of MATLAB optimtool, the simulation has started for different λ values. The final values for various λ values can be seen in the following table.

λ	K_e	K_d	α	β
0.1	3.22	0.017	1.77	0.68
0.5	0.72	0.017	3.34	0.74
1	0.12	0.77	0.01	2.31
10	0.02	0.26	0.01	2.75
100	1.49	0.017	0.21	0.01

For these values, simulations were made and the results can be seen in the figures 6 and 7 . When we increase the λ value, the system gets slower and the change in the control signal decreases as expected. The physical meaning of increasing λ value is the weight of importance in the change of the control signal. If we say that λ is 0.1, it means that i value reference tracking 10 times more than the change in the control signal. Therefore, the system wants to reach the reference as quick as possible. Similarly if we say λ is 100, We say that a smooth control signal is 100 times more important than reference tracking. Hence, the system response is smoother.

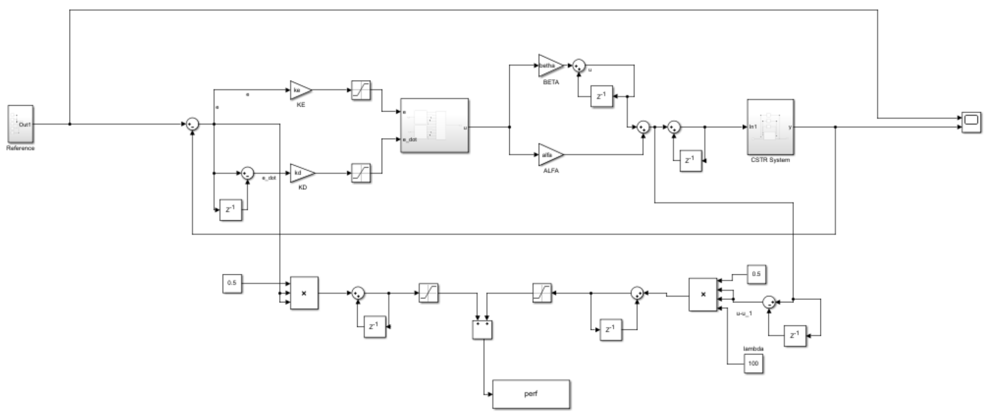


Figure 5: Genetic Algorithm Structure

2.4 Section D

In the last section, an adaptation mechanism was demanded to create. For this purpose, a function tuner structure that was mentioned in the paper "A PID type fuzzy controller with self-tuning scaling factors" was used. The method simply tunes the scaling factors by writing them as the functions of error and the change of the error. We define two different functions like $f()$ and $g()$.

$$f(e(t)) = a_1 abs(e(t)) + a_2$$

$$g(e(t)) = b_1(1 - abs(e(t))) + b_2$$

The parameters β and K_{ds} are aimed to be updated. Therefore the update can be made by the following equations,

$$k_d = k_{ds} g(e(t))$$

$$\beta_s = \beta f(e(t))$$

After defining these functions, the system structure can be given in the figure 8. It is obvious that by using function tuner, one must determine four additional parameters. These parameters can also be determined by a genetic algorithm or by try and error. In this case, the parameters were determined with the second method and the simulation results were compared with the system in section B. The parameters were selected as $a_1 = 2$, $a_2 = 0.7$, $b_1 = 3$ and $b_2 = 0.8$. The comparsion of these two systems can be observed in the figure 9.

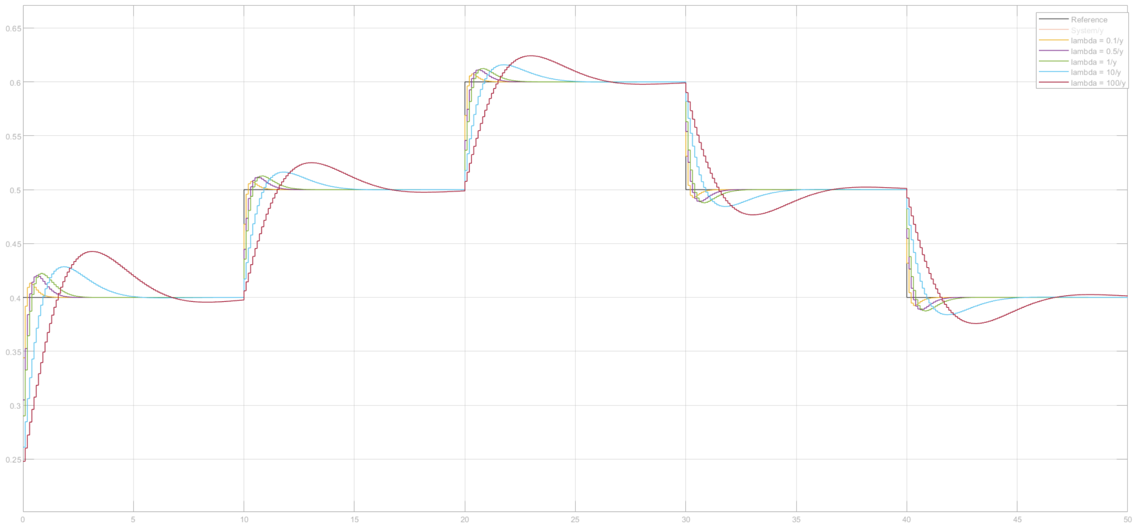


Figure 6: System responses for different lambda values

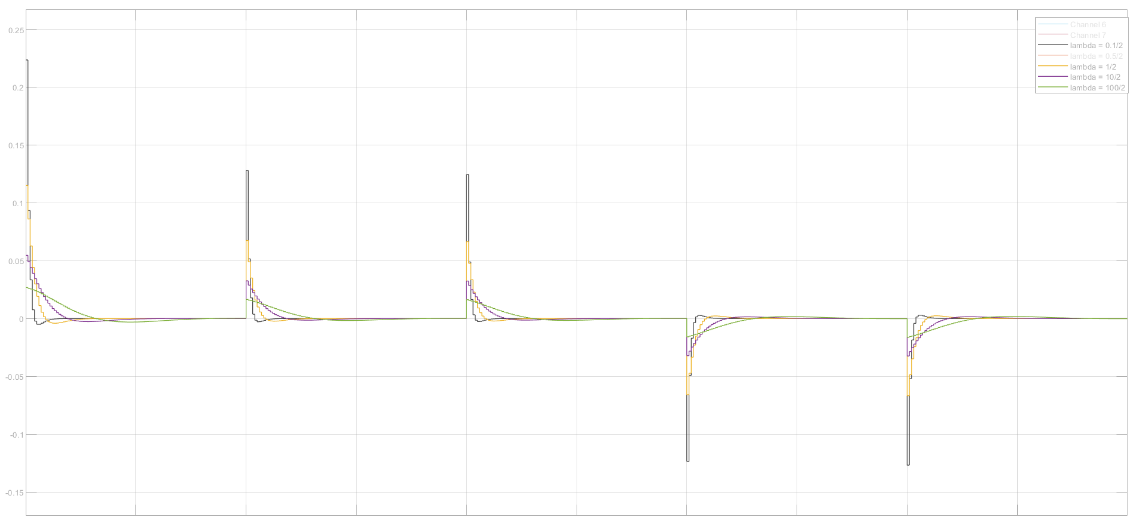
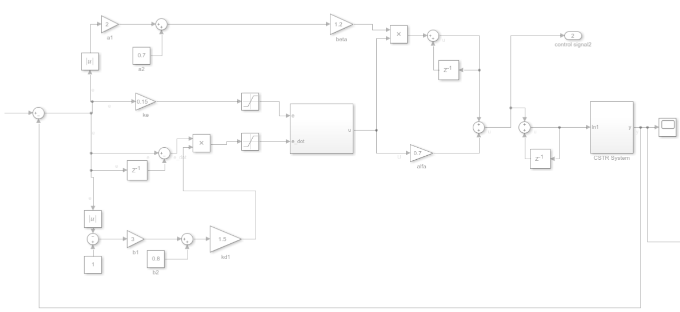
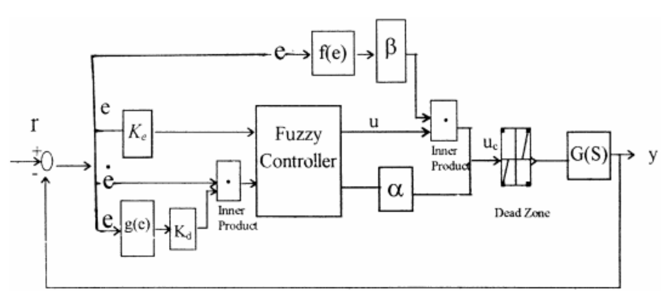


Figure 7: Control signals for different lambda values



(a) Function tuner in simulink



(b) Function Tuner Structure

Figure 8: Function Tuner structures

Figure 9: The comparison of function tuner and classical FLC PID