Mamdani Fuzzy Logic for Reaction Pendulum

January, 2025

Author:
Pawel Wiktor

Email:
pawelwiktor.kontakt@gmail.com

GitHub:
https://https://github.com/pawelwiktor-github

Table of Contents

1 Creating Fuzzy Logic Mamdani Controller (MISO) for Pendulum Control Signal

Creation of *Fuzzy Logic Mamdani-Type Controller (MISO)* for *Pendulum* Control Signal was made in *Matlab - Simulink* environment. The results were tested in a model that was created for Reaction Pendulum identification and as a point of reference *LQR Controller* was included in model aswell.

1.1 Theoretical Introduction

The *Mamdani-Type Fuzzy Logic Controller* (FLC) is widely used in control applications due to its simplicity and interpretability. For a system such as a *Reaction Pendulum*, which is inherently nonlinear and unstable, the *Mamdani-Type FLC* provides a robust framework for designing a *Multi-Input Single-Output* (MISO) control system.

In the context of a *Reaction Pendulum*, inputs includes the angular placement and angular velocity of the *Pendulum*. These are mapped to fuzzy sets using membership functions. According to *Matlab Documentation* there are 15 functions that can be chosen (but only thirteen for *Mamdani-Type*). Couple of main ones are:

- Triangular membership function [*"trimf"*]
- Trapezoidal membership function [*"trapmf"*]
- Gaussian membership function [*"gaussmf"*]
- Generalized bell-shaped membership function [*"gbellmf"*]

The triangular membership function is simple, computationally efficient and ideal for systems with clear boundaries between states.

The trapezoidal membership function is similar to the triangular but with flatter tops, making them less sensitive to small variations and useful for broader regions of influence in the input or output space.

Gaussian membership function is smooth, continuous, suitable if you want to avoid abrupt changes in control behavior, and good for systems where overlapping states contribute significantly to stability.

The generalized bell-shaped membership function is flexible, tunable, and great for finely tuned control systems, but slightly more complex to configure than previously listed.

For a MISO system, each input is fuzzified independently, allowing for linguistic representation such as "small angle"or "high velocity."

The *Mamdani FLC* uses a rule base of *If-Then* rules, for example: "*If Position value is negatively small and Velocity value is negatively large, then Control Signal is negatively small*". Based on that, the user can easily influence the specific cases of the system.

The benefits of using *Mamdani-Type FLC* for *Reaction Pendulum* Control are:

- The Mamdani FLC inherently handles nonlinearities of the reaction pendulum, avoiding the need for complex mathematical models
- Designing the controller involves defining intuitive rules and membership functions, which can be iteratively refined based on simulation or experimental results
- The fuzzy controller adapts well to uncertainties and parameter variations, making it suitable for a system prone to disturbances

1.2 Creating *Mamdani-Type FLC*

After properly loading the data and creating the FIS structure, the first important step is to analyze how the *Reaction Pendulum* model works. Studies of the reaction pendulum object showed that this DC motor is able to swing back to the top position (180° shifted from the normal equilibrium point) when the Pendulum Angle is equal to 30° (or to $-30°$).

The analysis process was based on *LQR Controller* which reacted to the 30° swing of the Pendulum. The swing in oposite way showed exactly opposite behaviour (same ranges, different direction).

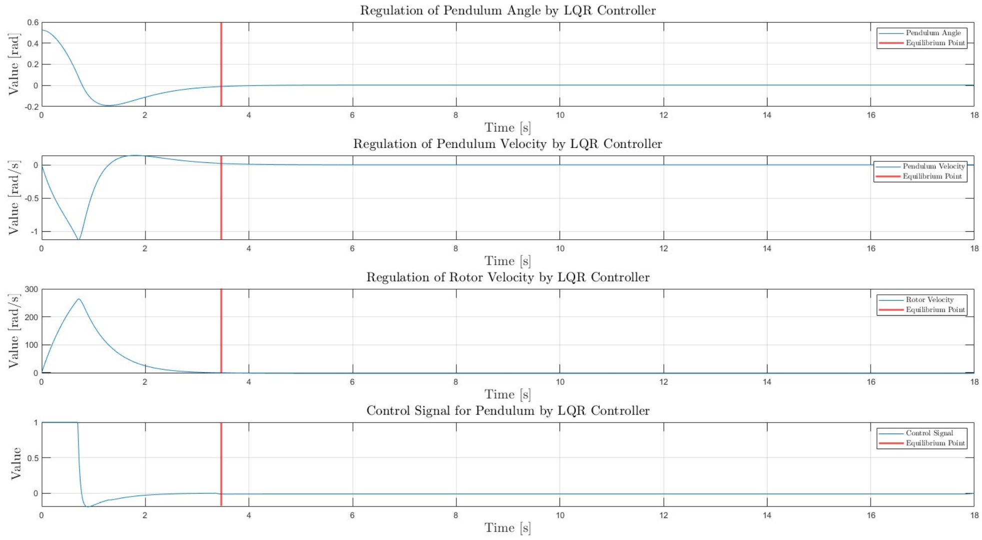

Fig. 1. *LQR* regulation of *Pendulum* after 30° swing

The first thing observed is that when the Pendulum is swinged, to bring it back to equilibrium point, the control signal needs to spin the rotor disc in the opposite direction.

In this case, when the Pendulum Angle is equal to 30° and it is starting to go to the equilibrium point, the Control Signal is maximum and the motor is starting to spin in the opposite direction. Accordingly, the Pendulum arm is heading to the left. During the process, the control signal is lower in value.

When Pendulum Angle slightly shifts to the left side, the rotor is starting to brake and Control Signal is coming back to zero.

The equilibrium point is reached at 3.472 [s].

With that knowledge, it is possible to create membership functions to influence the behavior of the model for specific cases.

1.3 Membership Functions

Membership functions were created based on previous knowledge of the model. It was realized that the least number of membership functions for the Fuzzy Logic Controller to work properly is 3 (for inputs and output). The triangular membership function were chosen because of its simplicity.

Specifications are shown in the tables 1, 2, 3, 4.

Table 1. Pendulum Angle Membership Functions

Name	Type	Values
Pendulum Angle is Left	Triangular	$[-3.14159, -0.5, -0.01]$
Pendulum Angle is Center	Triangular	$[-0.02, 0, 0.02]$
Pendulum Angle is Right	Triangular	$[0.01, 0.5, 3.14159]$

Table 2. Pendulum Velocity Membership Functions

Name	Type	Values
Pendulum Velocity is Left	Triangular	$[-10, -5, -0.01]$
Pendulum Velocity is NoMovement	Triangular	$[-0.02, 0, 0.02]$
Pendulum Velocity is Right	Triangular	$[0.01, 5, 10]$

Table 3. Rotor Velocity Membership Functions

Name	Type	Values
Rotor Velocity is Negative	Triangular	$[-450, -130, -5]$
Rotor Velocity is Zero	Triangular	$[-10, 0, 10]$
Rotor Velocity is Positive	Triangular	$[5, 130, 450]$

Table 4. Control Signal Membership Functions

Name	Type	Values
Control Signal is Negative	Triangular	$[-1.2, -1, -0.01]$
Control Signal is Zero	Triangular	$[-0.02, 0, 0.02]$
Control Signal is Positive	Triangular	$[0.01, 1, 1.2]$

1.4 Rules

Rules specify the behavior of the whole system based on membership functions. That means the controller should react this way if this parameter has this value. It is necessary to cover all the cases that are necessary for the control signal to be working properly. To be able to cover all cases, you need to visualize how system should work and have a priori knowledge of the model that you are working on (or have specific experiments that show types of behavior).

Created rules are shown in the table 5.

Table 5. Fuzzy Rules for Pendulum Control

Rule	Description	Weight
1	If PendulumAngle is Left and PendulumVelocity is Left and RotorVelocity is Negative then Control is Negative	1
2	If PendulumAngle is Center and PendulumVelocity is Left and RotorVelocity is Negative then Control is Negative	1
3	If PendulumAngle is Right and PendulumVelocity is Left and RotorVelocity is Negative then Control is Positive	1
4	If PendulumAngle is Left and PendulumVelocity is Zero and RotorVelocity is Negative then Control is Negative	1
5	If PendulumAngle is Center and PendulumVelocity is Zero and RotorVelocity is Negative then Control is Negative	1
6	If PendulumAngle is Right and PendulumVelocity is Zero and RotorVelocity is Negative then Control is Positive	1
7	If PendulumAngle is Left and PendulumVelocity is Right and RotorVelocity is Negative then Control is Negative	1
8	If PendulumAngle is Center and PendulumVelocity is Right and RotorVelocity is Negative then Control is Positive	1
9	If PendulumAngle is Right and PendulumVelocity is Right and RotorVelocity is Negative then Control is Positive	1
10	If PendulumAngle is Left and PendulumVelocity is Left and RotorVelocity is Zero then Control is Negative	1
11	If PendulumAngle is Center and PendulumVelocity is Left and RotorVelocity is Zero then Control is Negative	1
12	If PendulumAngle is Right and PendulumVelocity is Left and RotorVelocity is Zero then Control is Positive	1
13	If PendulumAngle is Left and PendulumVelocity is Zero and RotorVelocity is Zero then Control is Negative	1
14	If PendulumAngle is Center and PendulumVelocity is Zero and RotorVelocity is Zero then Control is Zero	1
15	If PendulumAngle is Right and PendulumVelocity is Zero and RotorVelocity is Zero then Control is Positive	1
16	If PendulumAngle is Left and PendulumVelocity is Right and RotorVelocity is Zero then Control is Negative	1
17	If PendulumAngle is Center and PendulumVelocity is Right and RotorVelocity is Zero then Control is Positive	1
18	If PendulumAngle is Right and PendulumVelocity is Right and RotorVelocity is Zero then Control is Positive	1
19	If PendulumAngle is Left and PendulumVelocity is Left and RotorVelocity is Positive then Control is Negative	1
20	If PendulumAngle is Center and PendulumVelocity is Left and RotorVelocity is Positive then Control is Negative	1
21	If PendulumAngle is Right and PendulumVelocity is Left and RotorVelocity is Positive then Control is Positive	1
22	If PendulumAngle is Left and PendulumVelocity is Zero and RotorVelocity is Positive then Control is Negative	1
23	If PendulumAngle is Center and PendulumVelocity is Zero and RotorVelocity is Positive then Control is Positive	1
24	If PendulumAngle is Right and PendulumVelocity is Zero and RotorVelocity is Positive then Control is Positive	1
25	If PendulumAngle is Left and PendulumVelocity is Right and RotorVelocity is Positive then Control is Negative	1
26	If PendulumAngle is Center and PendulumVelocity is Right and RotorVelocity is Positive then Control is Positive	1
27	If PendulumAngle is Right and PendulumVelocity is Right and RotorVelocity is Positive then Control is Positive	1

1.5 Conlcusions

The first figure represents fuzzy regulation of the maximum swing of the Pendulum (equal to 30°) and the second figure represents comparison to LQR (under the same conditions).

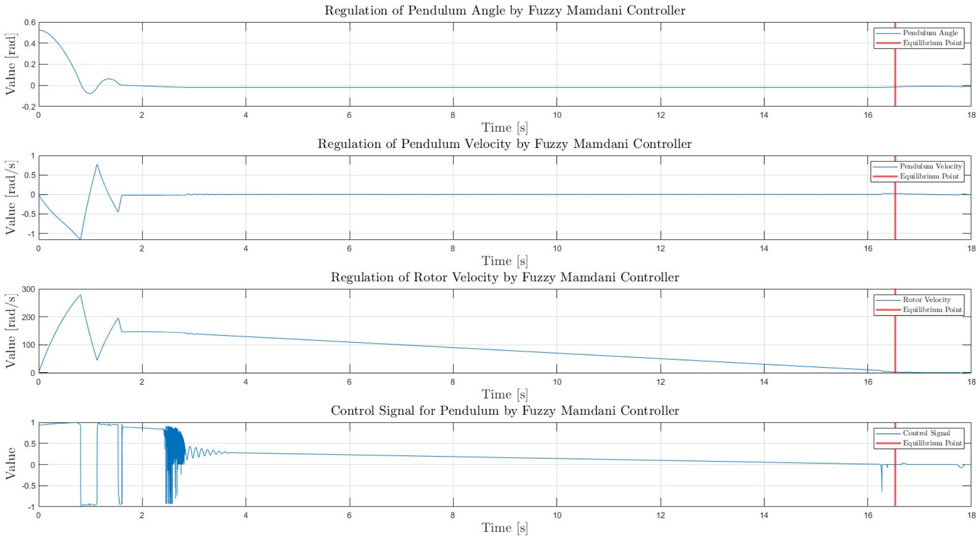

Fig. 2. *Fuzzy Mamdani* regulation of Pendulum after 30° swing

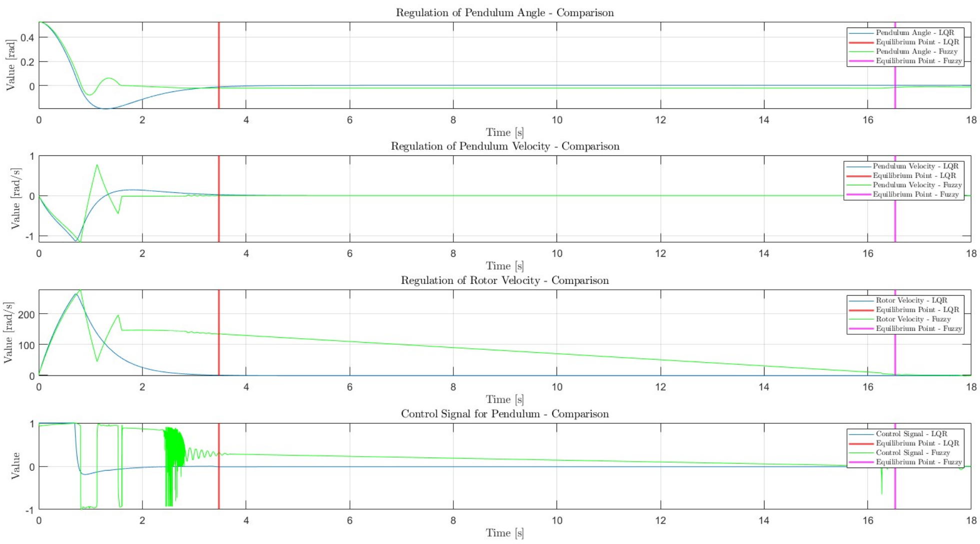

Fig. 3. Comparison of controllers regulation of Pendulum after 30° swing

The first visible issue with Fuzzy is that when the angle reaches around the center position, there are some overregulation in Pendulum Angle, Pendulum Velocity, Rotor Velocity, and Control Signal. In my opinion, it is because of insufficient regulation, i.e. poorly matched ranges of membership functions. It can cause engine overheating and loss of efficiency of the system.

Comparison of controllers shows that apart from overregulations Fuzzy stabilized Pendulum 18.6640 [s] later then original LQR. The problem can occur through misalignment of membership functions.

1.6 Fuzzy Logic Mamdani Controller adjustments

1.6.1 Membership Functions adjustments

Firstly, Membership Functions were adjusted, based on the object knolwedge and trial-and-error method. The results and changes are shown below.

Table 6. Pendulum Angle adjusted Membership Functions

Name	Type	Values
Pendulum Angle is Left	Triangular	$[-3.14159, -0.5, 0]$
Pendulum Angle is Center	Triangular	$[-0.08, 0, 0.08]$
Pendulum Angle is Right	Triangular	$[0, 0.5, 3.14159]$

The slope of external Membership Functions was reduced and the zero range in the center Membership Function was extended.

Table 7. Pendulum Velocity adjusted Membership Functions

Name	Type	Values
Pendulum Velocity is Left	Triangular	$[-4.2, -2.1, 0]$
Pendulum Velocity is Zero	Triangular	$[-0.12, 0, 0.12]$
Pendulum Velocity is Right	Triangular	$[0, 2.1, 4.2]$

The range of Pendulum Velocity was changed (because it was incorrectly defined). As in the previous case, the slope of external Membership Functions was reduced and the zero range in the center Membership Function was extended.

Table 8. Rotor Velocity adjusted Membership Functions

Name	Type	Values
Rotor Velocity is Negative	Triangular	$[-450, -250, 0]$
Rotor Velocity is Zero	Triangular	$[-30, 0, 30]$
Rotor Velocity is Positive	Triangular	$[0, 250, 450]$

In addition to extending the zero range in the center Membership Function, the slope was increased of external Membership Functions.

Table 9. Control Signal adjusted Membership Functions

Name	Type	Values
Control Signal is Negative	Triangular	$[-1.45, -0.95, -0.45]$
Control Signal is Zero	Triangular	$[-0.5, 0, 0.5]$
Control Signal is Positive	Triangular	$[0.45, 0.95, 1.45]$

Adjustments of Control Signal Membership Functions was the most important one. Based on how the Control Signal value is multiplied, the zero range was dramatically increased to implement *Low Positive* and *Low Negative* Control values to avoid overregulations.

1.6.2 Fuzzy Logic Mamdani Controller adjustments results

After tuning, the Mamdani Controller is much more efficient. The comparison was performed exactly the same as in the previous version of Fuzzy.

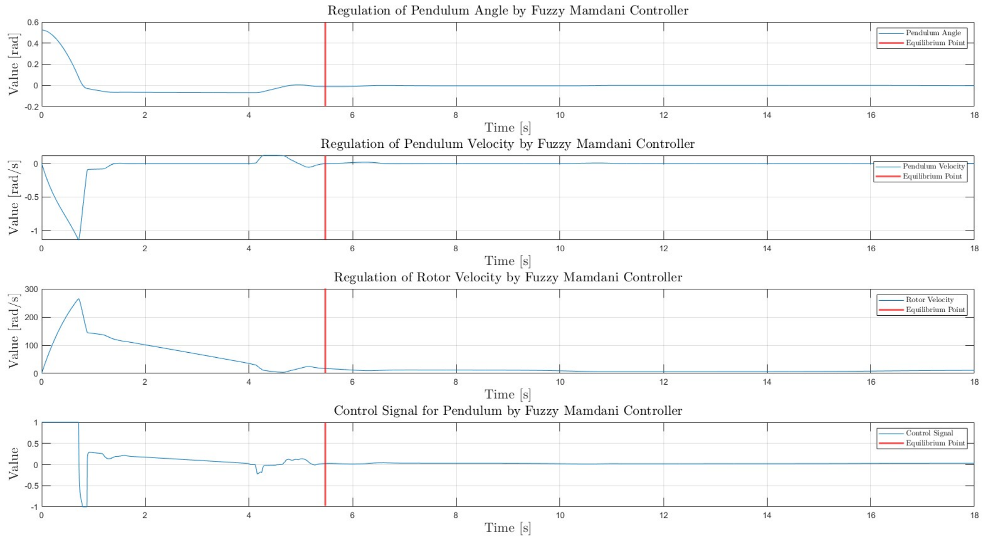

Fig. 4. *Fuzzy Mamdani* regulation of Pendulum after 30° swing

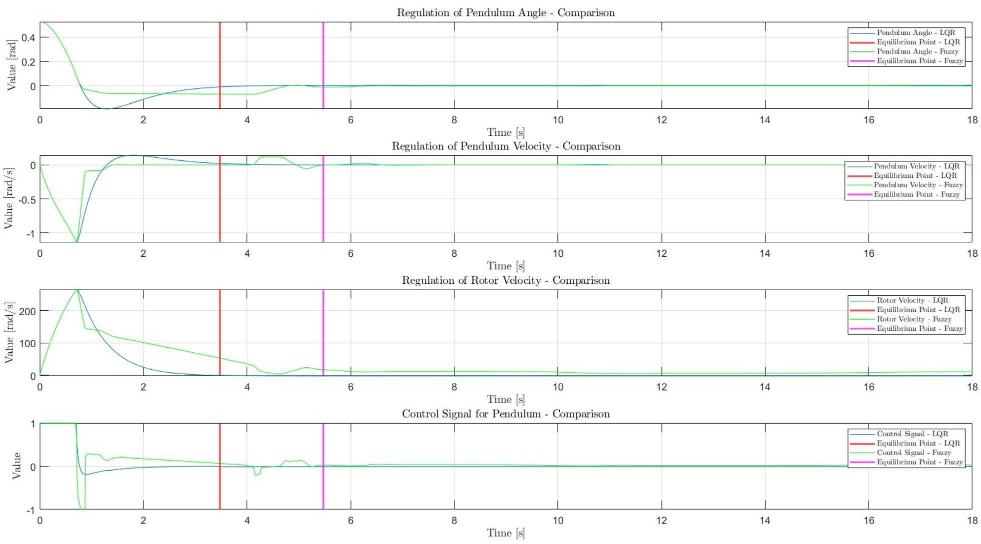

Fig. 5. Comparison of controllers regulation of Pendulum after 30° swing

1.6.3 Conclusions

After the tuning process, performance of Fuzzy was much more satisfying. There are no overregulations as mentioned earlier, and Fuzzy performs stabilization much more faster.

As a result validation criterion, the mean squared error was chosen, which is calculated by the equation 1.

$$\sigma_x^2 = \frac{1}{N} \sum_{k=1}^{N} \left\| x_L - x_F \right\|^2 \tag{1}$$

where:

σ – mean squared error

N – number of values
x_L – values provided by LQR controller
x_F – values provided by Fuzzy controller

Considering only Pendulum Angle stabilization, the mean squared error was calculated only as the squared difference between the values provided by LQR and the values provided by Fuzzy, but considering all input stabilization, the mean squared error was the sum of the individual input values. The results are shown in the table 10.

Table 10. Mean squared error results

Name	Values
Pendulum Angle stabilization	$2.01 \cdot 10e^{-7}$
Pendulum Angle, Pendulum Velocity and Rotor Velocity stabilization	$5.57 \cdot 10e^{-2}$

By the results it is possible to say that created Fuzzy Controler works simillary to LQR which provides good quality regulation. The difference in mean squared errors is caused by the fact that in Fuzzy, Rotor Velocity is stabilized at around zero values instead of directly in zero. The difference in time stabilization is 1.22 [s], but LQR performs a smoother transition to the equilibrium point near zero values.

Reviewing theory of fuzzy logic reminded that because of it's accuracy of control selection by construction of *if-then* rules, the Fuzzy should work faster and smoother than LQR. It was discovered that the issue is caused by too few membership functions for the control with 27 rules (there should be 9 membership functions for the control in this scenario). It is caused due to used Fuzzy Creation Parameters. In this project it was used respectively: And method(min), Implication method(prod), Aggregation method(sum), Defuzzification method(centroid). Description can be found below:

- And Method == min -> Uses the minimum value of the membership degrees
- Implication Method == prod -> Uses the product of the antecedent truth value and the membership function of the consequent
- Aggregation Method == sum -> Adds the outputs of the rules together
- Defuzzification Method == centroid -> Calculates the center of area (COA) of the aggregated fuzzy set, which is a weighted average

By using this approach, the output is calcualted by degree of membership and rule weight multiplication and then there is a sum of applied rules. In the end, by defuzzification method, output value is a weighted average of mentioned above mathematical operations. Based on this observation, it was found that far fewer rules could be designed i.e. 3 for each input.

1.7 Fuzzy Logic Mamdani Controller correction

1.7.1 Membership Functions correction

The slope was increased of the external Membership Functions for Rotor Velocity (table 11) and completely changed Control Signal Membership Functions (table 12).

Table 11. Rotor Velocity corrected Membership Functions

Name	Type	Values
Rotor Velocity is Negative	Triangular	$[-450, -260, 0]$
Rotor Velocity is Zero	Triangular	$[-30, 0, 30]$
Rotor Velocity is Positive	Triangular	$[0, 260, 450]$

Table 12. Control Signal corrected Membership Functions

Name	Type	Values
Control Signal is Negative	Triangular	$[-10, -10, 0]$
Control Signal is Zero	Triangular	$[-0.01, 0, 0.01]$
Control Signal is Positive	Triangular	$[0, 10, 10]$

1.7.2 Rules correction

Table 13. Corrected Fuzzy Rules for Pendulum Control

Rule	Description	Weight
1	If PendulumAngle is Left then Control is Positive	0.0799
2	If PendulumAngle is Center then Control is Zero	0.0799
3	If PendulumAngle is Right then Control is Negative	0.0799
4	If PendulumVelocity is Left then Control is Positive	0.161
5	If PendulumVelocity is Zero then Control is Zero	0.161
6	If PendulumVelocity is Right then Control is Negative	0.161
7	If RotorVelocity is Negative then Control is Positive	0.101
8	If RotorVelocity is Zero then Control is Zero	0.101
9	If RotorVelocity is Positive then Control is Negative	0.101

Rule wages were changed in order to adapt Control Signal multiplication for specified parameter in addition to reducing the number of rules.

1.7.3 Fuzzy Logic Mamdani Controller correction results

After the correction, the performance of Fuzzy was much better. Trasitions between states are smoother, and stabilization is achieved faster.

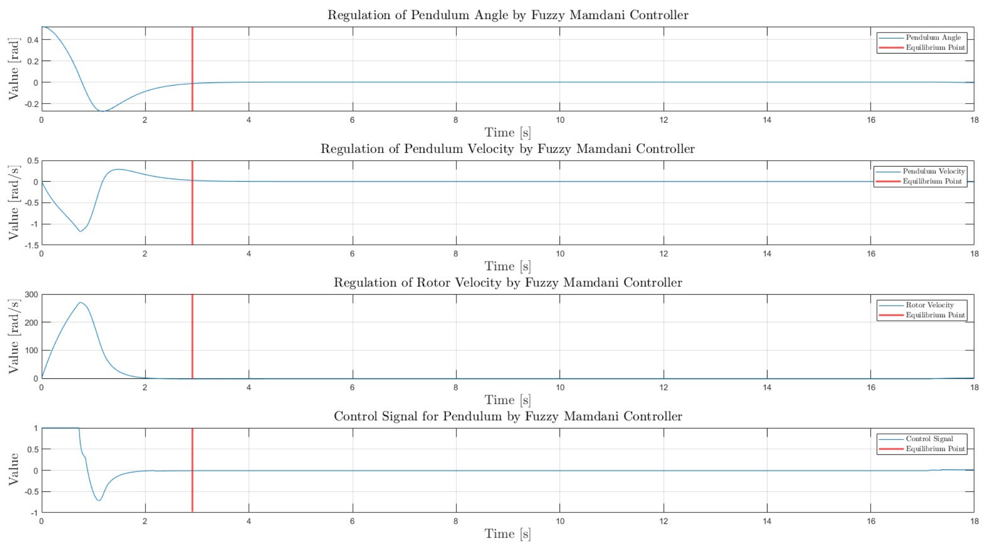

Fig. 6. *Fuzzy Mamdani* regulation of Pendulum after 30° swing

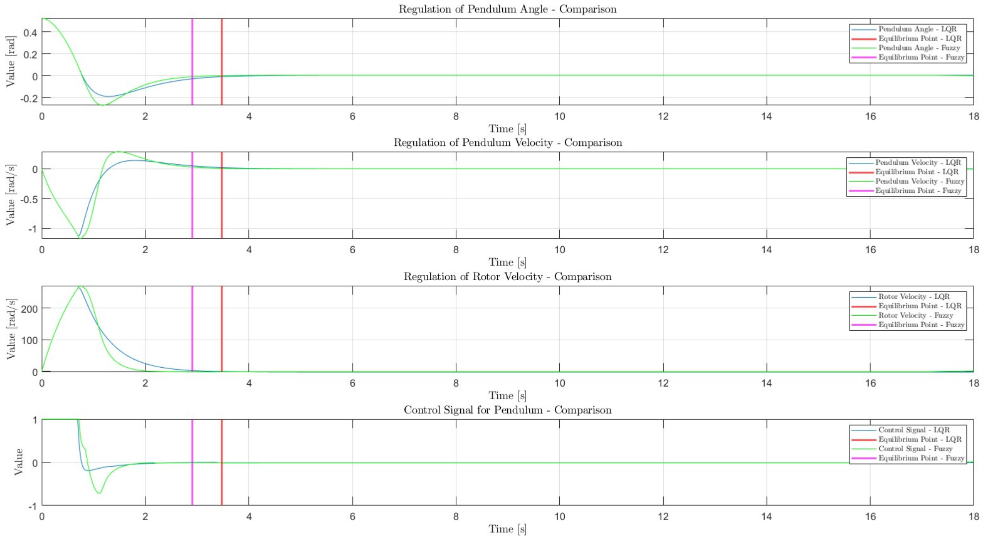

Fig. 7. Comparison of controllers regulation of Pendulum after 30° swing

Fuzzy performs stabilization at 2.912 [s], which is 0.56 [s] faster than LQR.

1.8 Fuzzy Logic structure review

After correction, Fuzzy Logic structure is shown in figure 8.

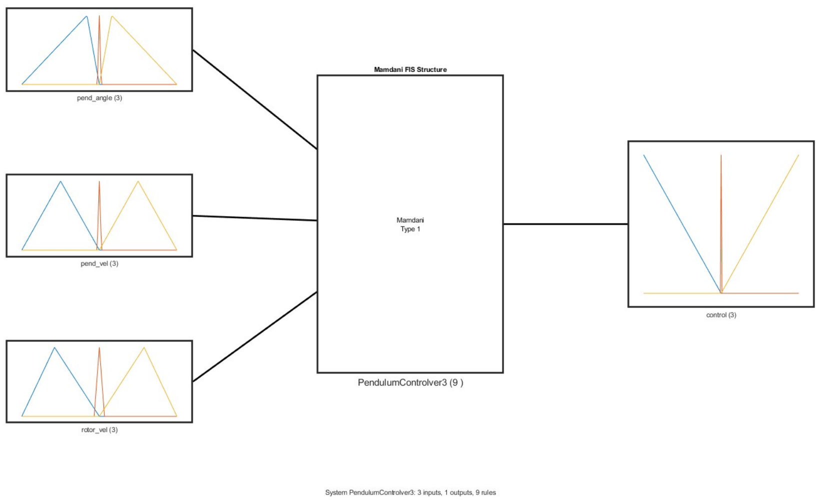

Fig. 8. *Mamdani Fuzzy Controller structure*

Matlab environment enables the user to visualize graphically the membership functions created using the *plotmf* function. Those graphical illustrations are shown in figure 9.

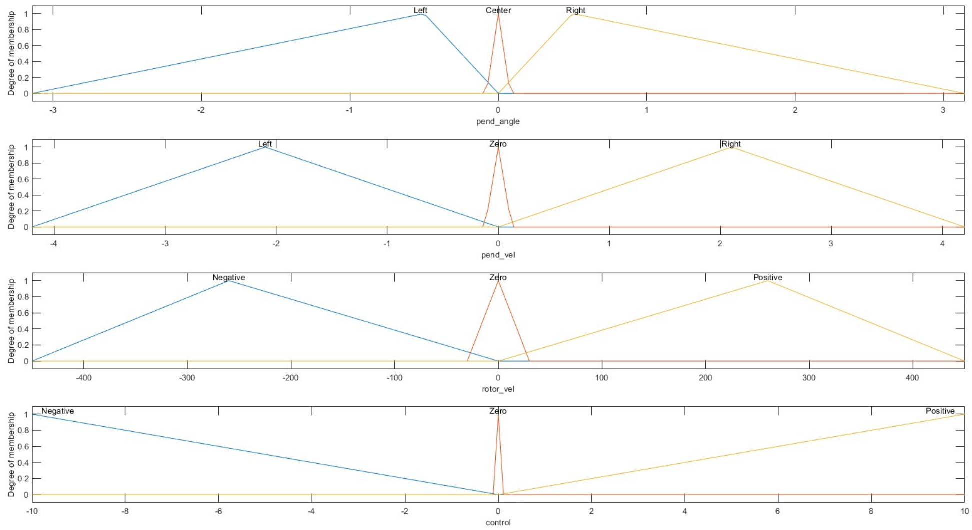

Fig. 9. Membership Function visualization

After rules creation there is a possibility to visualize control surface which shows how Control Signal is reacting for specific values of parameters (followed by the rules). Because we have 4-dimensional parameters, Matlab does not allow user to create that type of graph. That is why there are 3 figures that contain all possibilities of input parameters compared to output one. It is shown in figure 10.

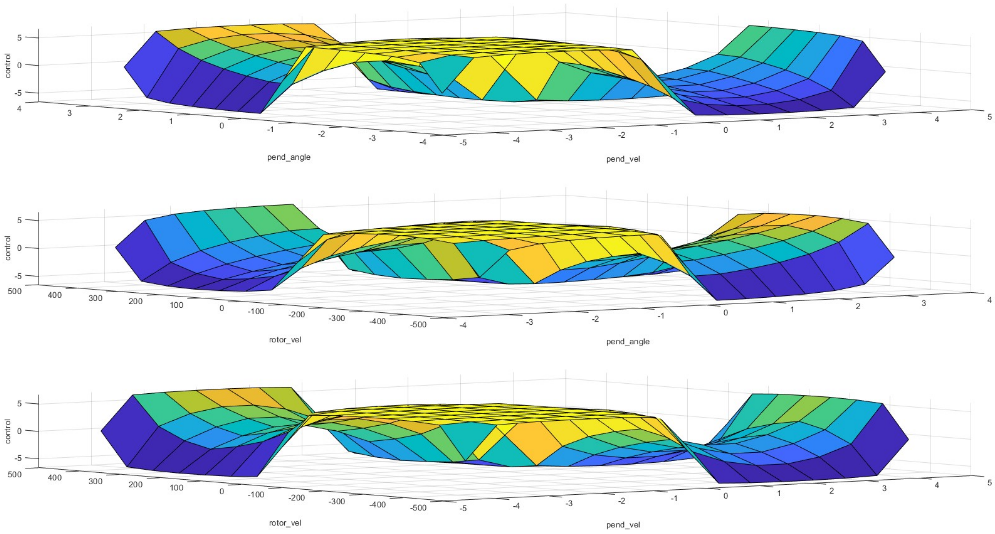

Fig. 10. Control surface visualization

1.9 Quality indicators

In order to check the robustness of the controller, two integral quality indices were calculated each for the different experiments carried out and the stabilization time of the pendulum was manually read and analyzed:

- Integral from the square of the control deviation (J_1)
- Integral of the absolute value of the control multiplied by time (J_2)
- Stabilization time (t_s)

Equations for quality indicators:

$$J_1 = \int_0^T u^2(t)\,dt \tag{2}$$

$$J_2 = \int_0^T |u(t)| \cdot t\,dt \tag{3}$$

where:

J_1 – Integral from the square of the control deviation
J_2 – Integral of the absolute value of the control multiplied by time
$u(t)$ – control value in time
t – simulation time

Description of experiments:

1) Constant disturbance in the deflection of the pendulum from equilibrium by 0.2 rad every 10 seconds
2) Constant disturbance in the deflection of the pendulum from equilibrium by 0.2 rad every 10 seconds + increased distance from the pivot point
3) Constant disturbance in the deflection of the pendulum from equilibrium by 0.2 rad every 10 seconds + constant motor disc rotation speed of 100 rad/s

Results for each experiment (per one swing):

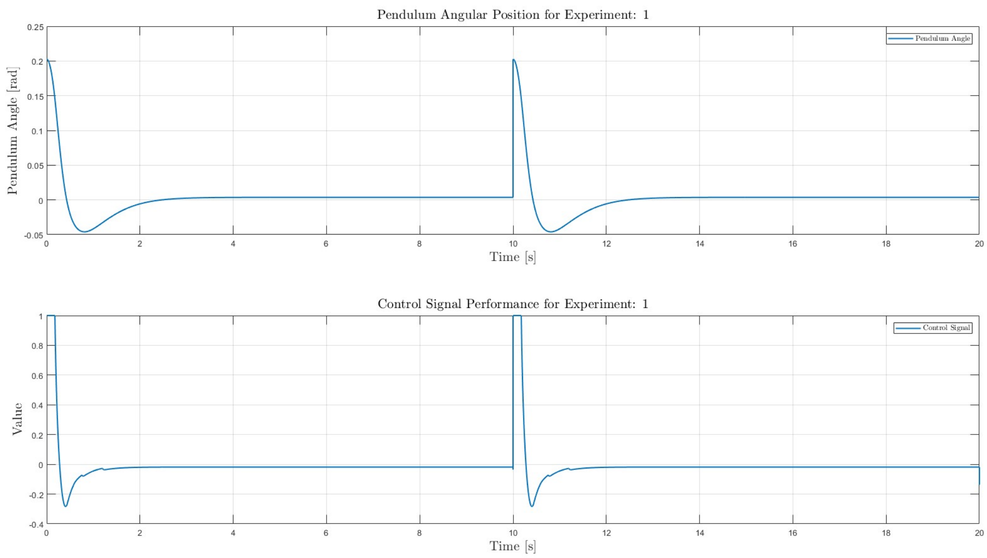

Fig. 11. Results for first experiment

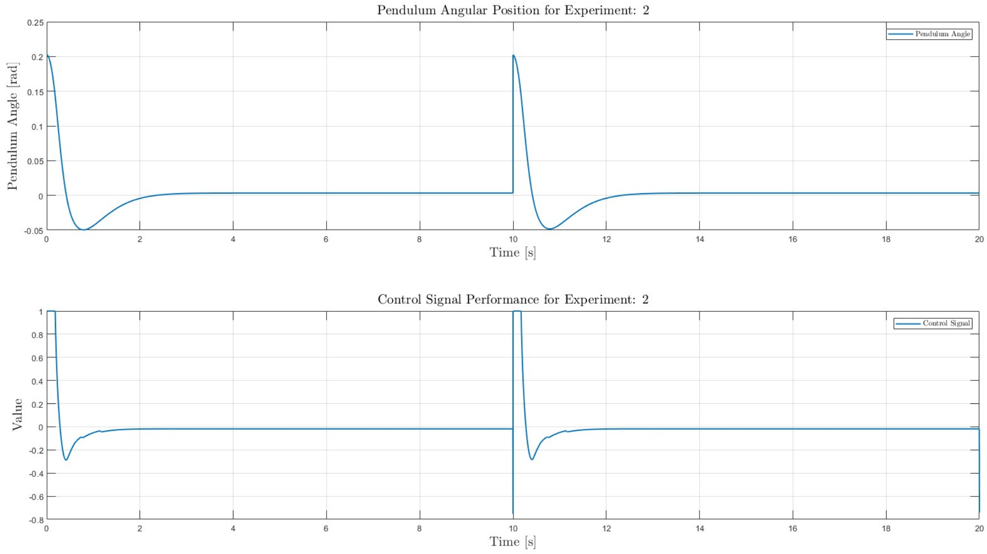

Fig. 12. Results for second experiment

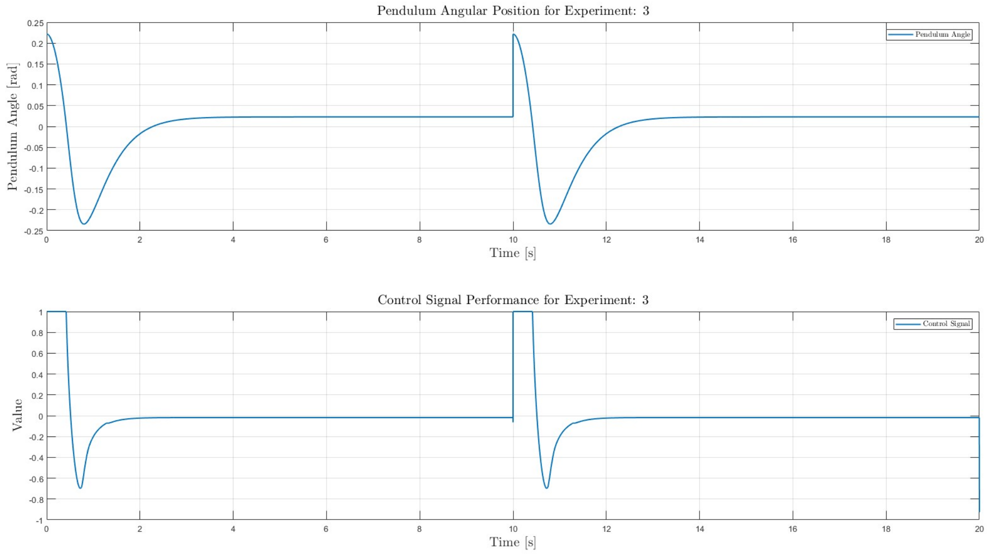

Fig. 13. Results for third experiment

Quality indicators were calculated and shown in the table 14.

Table 14. Quality indicators calculation results

Experiment indicator	J_1	J_2	t_s
First experiment	0.4372	1.7553	1.8157
Second experiment	0.4452	1.7857	1.8178
Third experiment	1.1234	3.7214	2.1076

1.9.1 Conclusions

Quality indicators validated the robustness and efficiency of the controller under varying disturbances. The stabilization time t_s revealed that the controller performed the best in the first experiment. The increased distance from the pivot point slightly influenced the stabilization performance within an acceptable range. The values of J_1 and J_2 indicated minimal control deviation and stable control application, demonstrating the effectiveness of the Mamdani controller in managing disturbances for the first and second experiments.

The results of the third experiment have shown that Fuzzy is trying to stabilize the Pendulum arm in the top position, but with a constant motor disc rotation speed set to 100 $[\frac{rad}{s}]$ it is impossible for the system to stabilize in 0. It is caused by an insufficient control signal shown by the J_1 indicator (which exceeded the [-1 1] range). Instead, Fuzzy stabilize Pendulum at around 11[°] (figure 13).

1.10 Summary

The development and analysis of the Fuzzy Logic Mamdani Controller for the Reaction Pendulum demonstrated its capacity to effectively handle the system's inherent non-linear dynamics and instability. Throughout the iterative design process, significant improvements in controller performance were achieved, as evidenced by reduced stabilization times and improved robustness under varying disturbances.

The adjustment and correction of membership functions and fuzzy rules improved the controller's stability and response. Initial overregulations and inefficiencies were mitigated by careful expansion of zero ranges and refinement of control signal parameters. After correction, the controller achieved stabilization faster than the benchmark LQR controller in certain scenarios, showcasing the potential for fuzzy logic in advanced control systems.

Quality indicator evaluations across different experimental setups highlighted the controller's robustness. The minimal deviations in control signal performance (as measured by the J1 and J2 indices) confirmed the system's ability to maintain effective regulation despite disturbances like increased pendulum deflection or changes in pivot point distance.

The third experiment illustrated the controller's limitations when faced with extreme disturbances, such as constant high-speed motor disc rotation. The inability to stabilize the pendulum in the top position under these conditions underscores the need for further refinement or integration with auxiliary control strategies to enhance performance in such scenarios.

Mamdani controller demonstrated substantial promise for managing non-linear dynamic systems like the Reaction Pendulum. Further refinements in the granularity of the membership function and the design of rules could enhance its efficacy and adaptability, making it a viable alternative to traditional controllers in specific applications.