Compound Pendulum: A Delivery Exercise in Fuzzy Computing

Mauro VÁZQUEZ CHAS, Dániel MÁCSAI

Master in Artificial Intelligence

Planning and Approximate Reasoning

10th January 2024

Contents

1 Introduction

The compound pendulum models robotic arm movement, important for control systems and mechanical modeling. It consists of a uniform bar with mass m and length l, suspended at a sliding pivot point A, adjustable by distance h from the center of gravity. This configuration, combined with a motor and propeller, results in oscillations requiring advanced control for stabilization.

This report focuses on designing a Mamdani fuzzy inference system (FIS) controller using Simulink and MATLAB's fuzzy toolbox to improve the pendulum's transient response and stabilize its angular position. The open-loop response shows sustained oscillations, highlighting the need for feedback control. Real-time angle measurements from an encoder are used for implementing control strategies.

The report covers defining fuzzy membership functions for error, error derivative, and thrust, establishing a rule base, and simulating the system's behavior. It also compares fuzzy control with the open-loop system and explores the impact of refining membership function granularity.

2 Design of the Fuzzy Controller

2.1 Membership Functions

Here we provide the membership functions we used for the fuzzy controller. We have three input variables: Error, Error Derivative, and one output variable, the Thrust.

2.1.1 Error

The error variable represents the difference between the desired and actual angular position of the pendulum. Following the instructions, we defined the membership functions as detailed below:
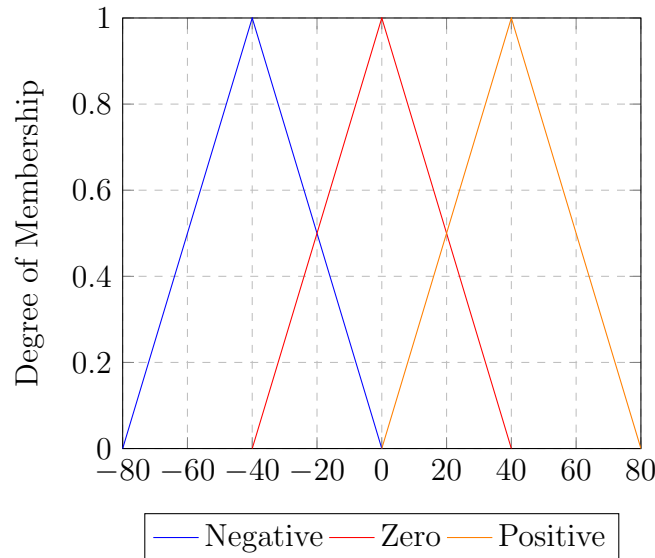

Figure 1: Membership Functions for Input Variable "Error"

We believe this variable could model reality more accurately using trapezoidal membership functions for the lowest and highest values. For instance, when the error exceeds 40 degrees, the corresponding membership function starts decreasing, which might influence rule activation. However, we adhered to the instructions and used triangular membership functions.
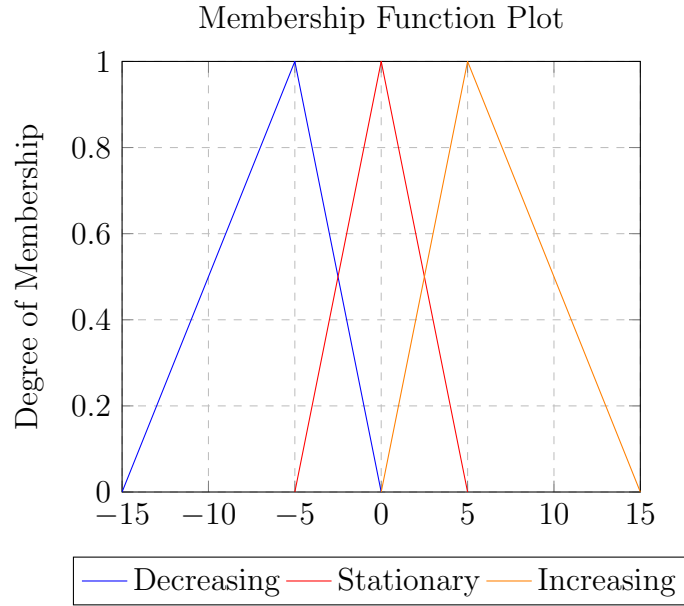
2.1.2   Error Derivative

Figure 2: Membership Functions for Input Variable "Error Derivative"

The same observation applies to the Error Derivative variable. Initially, the range was set to [-5, 5], but during simulations, the derivative often exceeded these bounds, causing the fuzzy controller to output zero. We adjusted the range to [-15, 15] for improved results.
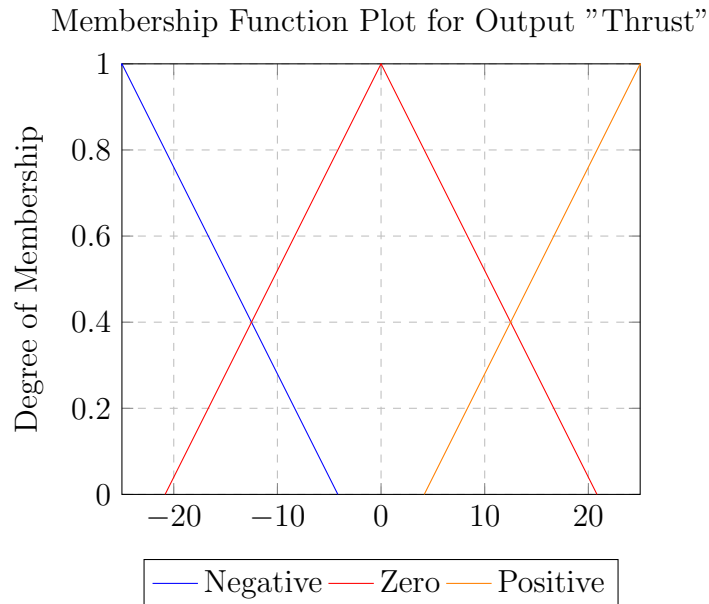
2.1.3   Thrust

Figure 3: Membership Functions for Output Variable "Thrust"

Since the design of this variable was not provided, we made the Negative and Positive variables activate at the highest levels for the lowest and highest values, respectively. The three variables are evenly distributed across the interval $[-25, 25]$.

2.2 Rules

We designed the 9 rules required for this exercise as follows:

Error	ErrorDerivative	Thrust
Negative	Decreasing	Negative
Zero	Decreasing	Negative
Positive	Decreasing	Positive
Negative	Stationary	Negative
Zero	Stationary	Zero
Positive	Stationary	Positive
Negative	Increasing	Negative
Zero	Increasing	Positive
Positive	Increasing	Positive

Table 1: Rules and Corresponding Weights

When the Error is not *Zero*, the Thrust is set accordingly: if the Error is *Negative*, the Thrust is *Negative*; if the Error is *Positive*, the Thrust is *Positive*. The Error Derivative variable is used to fine-tune the Thrust. For instance, if the Error is *Zero* but *Increasing*, the Thrust is *Positive*; if the Error is *Decreasing*, the Thrust is *Negative*. When both Error and Error Derivative are *Zero*, the Thrust is set to *Zero*.

3 Results

We ran the simulation for a stop time of 80 seconds using the provided controller and parameters. The results are shown in figures 4 and 5.
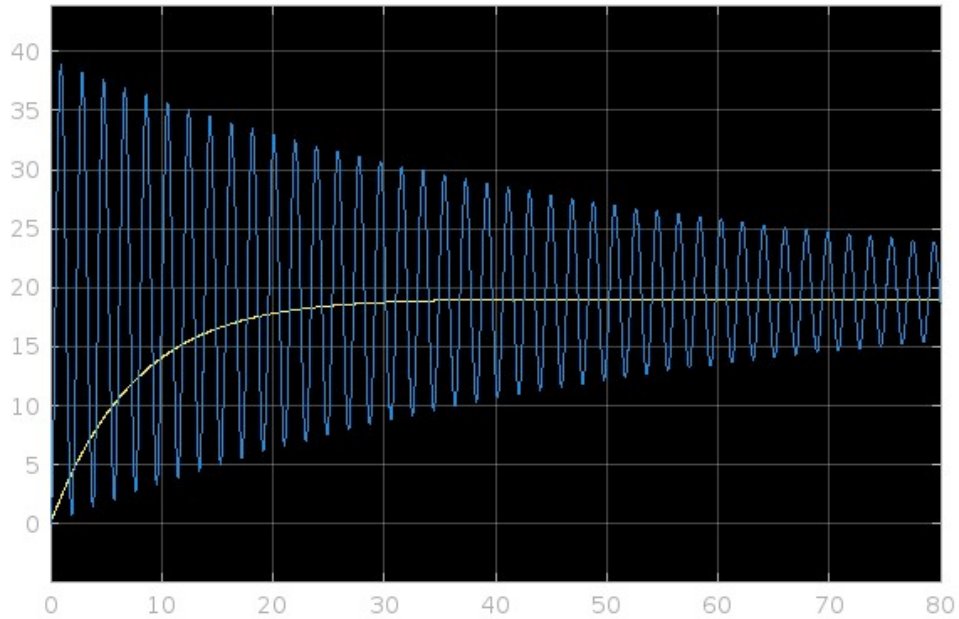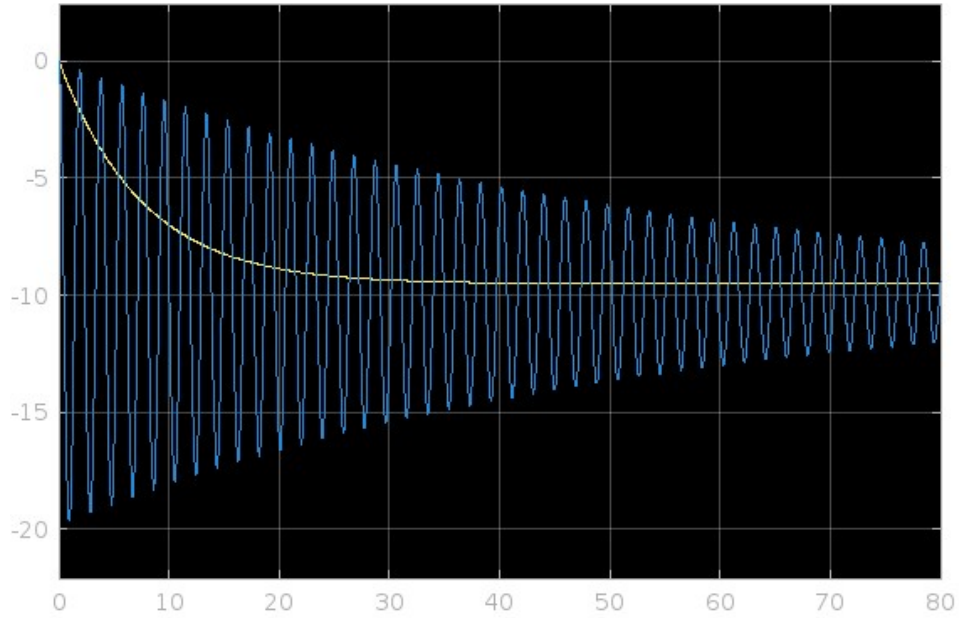

Figure 4: theta_ref = 20 and thrust = 0.123

Figure 5: theta_ref = -10 and thrust = -0.062

In both cases, our designed control system was able to stabilize the pendulum at the desired angle.
We don't see any oscillations, and the system reaches the desired angle in a reasonable time. We can
clearly see how the thrust is adjusted according to the error and its derivative.

4 More complex controller

To test out the cababilities of the fuzzy controller, we designed a more complex controller with 7
membership functions for the Thrust output variable. The plots frim the simulation are shown in
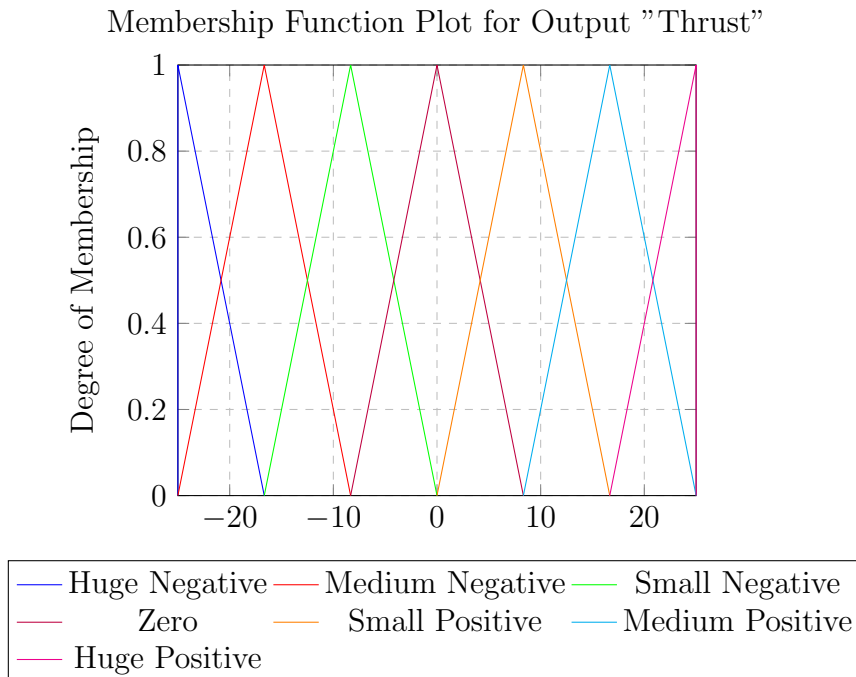figures 7 and 8.

Figure 6: Membership Functions for Output Variable "Thrust"

4

To make a change, we need to modify the rules too. Here are the updated rules:

Error	ErrorDerivative	Thrust
Negative	Decreasing	Huge Negative
Zero	Decreasing	Small Negative
Positive	Decreasing	Medium Negative
Negative	Stationary	Medium Negative
Zero	Stationary	Zero
Positive	Stationary	Medium Positive
Negative	Increasing	Medium Negative
Zero	Increasing	Small Positive
Positive	Increasing	Huge Positive

Table 2: Fuzzy Logic Rules for Thrust Calculation

Both simulations were conducted using the new controller, and the results are presented in the following figures:
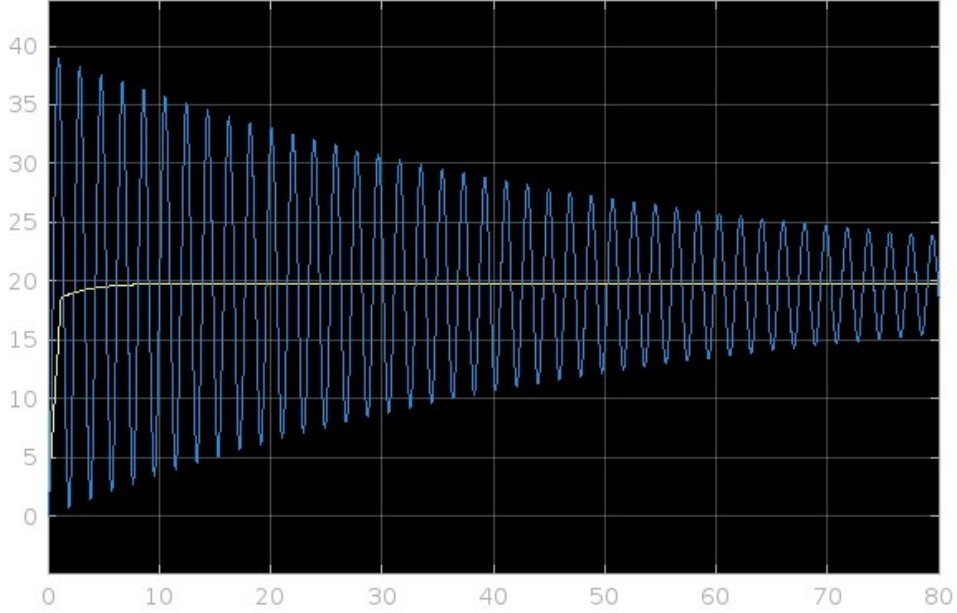

Figure 7: theta_ref = 20 and thrust = 0.123

Initially, it was hypothesized that improvements would require increasing the sensitivity of the input variables by incorporating additional membership functions and more complex rules. However, a significant improvement was observed by modifying only the output variable. This adjustment enabled the system to reach the desired angle more rapidly while maintaining stability. While further optimization of the input variables could enhance performance, the current results are already highly promising.

5   Conclusions

In this study, we successfully designed and implemented a Mamdani fuzzy inference system (FIS) to control a compound pendulum, achieving significant improvements in stability and transient response compared to the open-loop system. The fuzzy controller effectively reduced oscillations and
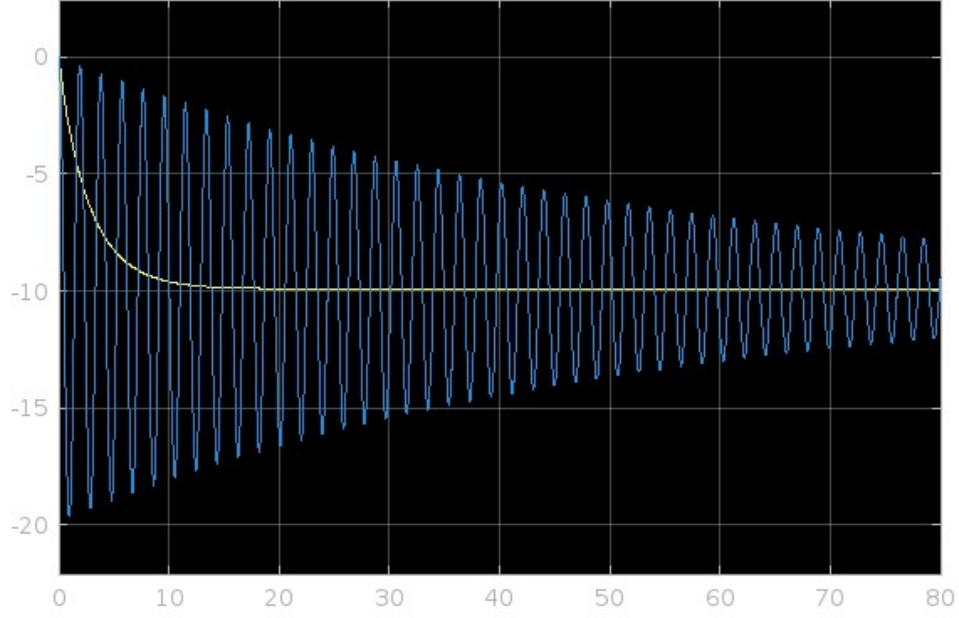
Figure 8: theta_ref = -10 and thrust = -0.062

achieved the desired angular position, demonstrating robustness and adaptability. By experimenting with enhanced membership functions and a more granular rule base, we observed that increasing the complexity of the output variable improved response speed and stability without requiring additional adjustments to input variables. This highlights the potential of fuzzy logic systems to handle nonlinear control problems efficiently. The results underscore the versatility and practicality of fuzzy computing in real-world dynamic systems.

6 Usage of generative AI models

We used ChatGPT for spelling checks, grammar corrections, and sentence structure improvements.