Intuitive Fuzzy Logic Controller Design using MATLAB/Simulink

Jared Poole, and Symoom I. Saad

***Abstract*—The objective of this paper was to demonstrate a fuzzy logic design technique that embeds the knowledge and intuition of human thinking into the resultant controller. The proposed technique was utilized to design a balance and positional controller for a two-wheeled robot using fuzzy logic without depending on an accurate mathematical model. Instead, a physics-based model was utilized. Simulation results indicated that the fuzzy logic controller successfully self-balanced the two-wheeled robot by ensuring that the robot stayed upright throughout its operation with negligible steady state error. The controller was also successful in guiding the robot to its target destination with minimal overshoot. These results demonstrated the practicality of the intuitive controller design technique and will raise awareness of the technique's merits in the scientific community.**

***Index Terms*—Controller Design Technique, Fuzzy Logic Control, Intelligent Control, Intuitive Control, Stability Control, Two-wheeled Robot**

I. Introduction

ALTHOUGH classical and modern control techniques are still heavily utilized in the industry, intelligent control is slowly rising in popularity [1]. Mamdani's published literature [2] on the application of fuzzy logic control pioneered the usage of fuzzy logic in control theory and in less than five years, its application started spanning across multiple industries [3].

Fuzzy logic is a way to model logical reasoning where the truth of a statement is not binary, but rather lies within a range of scale [4]. This approach maps out the qualitative human experience into the rules defining the fuzzy inference system by expressing thoughts in the form "IF…THEN…" [4]. This means that only a general understanding of the plant and its behavior is required to design a fuzzy controller. Since a precise mathematical model of the plant is not required for the design process, the industry will be more willing to employ this technique as the development of such models is often a costly and time-consuming process [5].

The concept of model-free control has been around for over a decade [6] [7], and with the advancements in faster computing power and the introduction of more capable computer-based software simulation packages, this concept should see a surge in popularity. This is because the controller designs can now be easily validated using specialized computer-aided simulations. For example, controllers designed for mechanical plants can be verified using the Simscape Multibody toolbox within MATLAB/Simulink, which provides a multibody simulation environment for 3D mechanical systems.

In fact, at the time of writing, fuzzy logic control is already very popular in literature and has been studied in a wide variety of applications.

In [8], a permanent magnet DC motor's loaded output speed was controlled using a fuzzy logic controller. A linearized mathematical model was developed and studied to help develop the fuzzy inference system. Simulation results indicated that the fuzzy logic controller had no overshoot or steady state error. Contrasting that to the PID controller, which had an overshoot of 9.23% and steady state error of 4%, the fuzzy logic controller had a smoother and more precise DC motor control.

In [9], a high powered and highly non-linear AC/DC converter's output voltage was controlled using a fuzzy logic controller. A non-linear mathematical model of the plant was studied to develop rules for the fuzzy inference system. Simulation results indicated that the fuzzy logic controller had a response time 0.09 seconds shorter than a PI controller. As a result, the fuzzy logic controller was deemed to be more robust.

In [10], a proton exchange membrane fuel cell (PEMFC)'s output voltage fluctuations were minimized using an adaptive fuzzy logic controller. A dynamic model of PEMFC fuel cell had to be developed and studied to help develop the rules for the fuzzy inference system. Simulation results indicated that the fuzzy logic controller had no steady state error. Contrasting that to a PID controller, which had a steady state error of 1.84%, the fuzzy logic controller was more power efficient.

In [11], a two-wheeled robot's balance in the presence of disturbance was stabilized using a fuzzy PD controller. A mathematical model had to be developed and simulated on to obtain the optimal coefficients for the controller. Simulation results indicate that the fuzzy logic controller had a settling time of at most 5 seconds when subjected to various degrees of disturbances. An actual implementation also validated the simulation results obtained above.

In [12], a DC-DC converter's output voltage was compensated against input voltage disturbances and load variations using a fuzzy logic controller. A mathematical model of a PI controller was studied to develop said fuzzy logic controller. Simulation results indicated that the fuzzy logic controller had reduced the overshoot by at least 74% when compared to the PI controller that was being studied, helping the plant be more power efficient.

This paper will diverge from the above literature by first demonstrating how to employ an intuition-based controller design approach rather than a mathematical one. The example used with be that of a balance and positional fuzzy controller for a two-wheeled robot. The design's effectiveness will then be verified using a physic-based computer model of a two-wheeled robot in Simscape Multibody.

II. Plant Operations and Design

In the absence of a mathematical model, the easiest method for verification of an intuitively developed controller would be to simulate a digital model of a plant inside a highly accurate simulation environment. The plant selected for control in this paper was a 3D mechanical two-wheeled robot. Simscape Multibody, a parametric solid physics engine, was selected to simulate this plant due to its tight integration with MATLAB/Simulink. The following sections expand on the various design choices made when capturing the plant in Simscape Multibody and any compromises that were required.

A. Plant Definition

Figure 1 depicts a rendering of the two-wheeled robot considered in this paper.

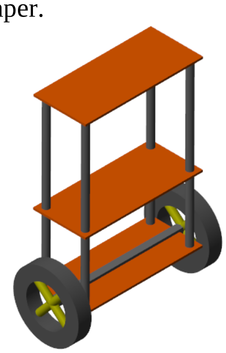

Figure 1- Simulink rendering of the two-wheeled robot

Key features included a symmetrical chassis for even weight distribution and a single axle to restrict travel to a single axis. These constraints helped make the intuition more straightforward when developing rules for the fuzzy logic controller. All other aspects of the robot were kept as realistic as possible. For example,

- The chassis was constructed of several rods and plates to allow the robot to have a realistic mass distribution.
- All surfaces, but particularly the wheels, were configured to experience both static and dynamic friction on contact with the simulation environment.

More details on relevant plant parameters can be found in Table 1.

Table 1 - Parameters for the two-wheeled robot

Parameter	Value
Coefficient of Static Friction	1
Coefficient of Dynamic Friction	1
Mass	0.950 kg

B. Plant Realization

The plant was realized in Simulink using a combination of built-in Simscape Multibody blocks and custom blocks as shown in Figure 2.

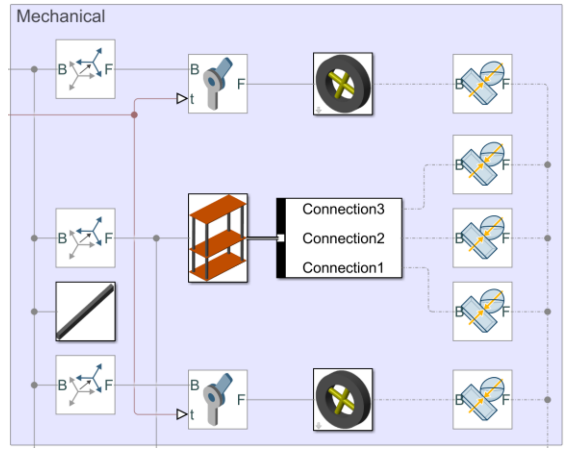

Figure 2 - Mechanical plant model realization in Simulink

Note that the chassis and wheels were themselves multi-block assemblies which were hidden here for brevity. Also observe that although the wheels were attached using revolute joints granting them each a degree of freedom along the axis of rotation, the same torque input was provided to both, forcing the robot to travel in one direction only.

C. Environment Definition

Figure 3 depicts the environment in which the robot will be simulated.

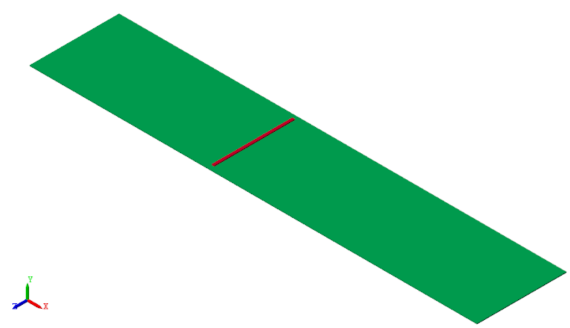

Figure 3 - Simulink rendering of the environment

In addition to the long patch of green floor and the red positional target indicator line, the environment also applied a constant force of gravity of $9.81 m/s^2$ to all objects with mass, namely the robot.

D. Environment Realization

The environment was realized in Simulink as shown in Figure 4.

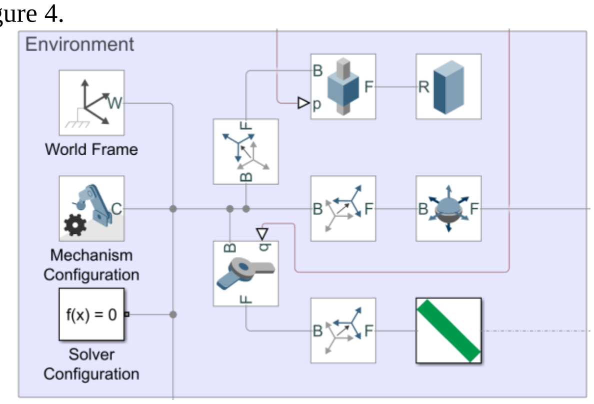

Figure 4 - Environment model realization in Simulink

The most important blocks were the "World Frame" which defined the global coordinate system and the "Mechanism Configuration" which defined gravity. Notice that a revolute joint was used to add a slope to the floor, which was utilized to test the controller's performance in the presence of a disturbance inclination.

III. Design of Fuzzy Logic Controller

A fuzzy logic controller uses fuzzy logic inferencing rules to make decisions that help control the overall system. A primary claim is that a highly effective controller can be developed by utilizing an expert's intuition at every step of the process. The following sections outline each step and how intuition played a role.

A. Fuzzification

Fuzzification is a process by which crisp input values are converted into fuzzy input values using fuzzy input membership functions. A sample input membership function can be found in Figure 5.

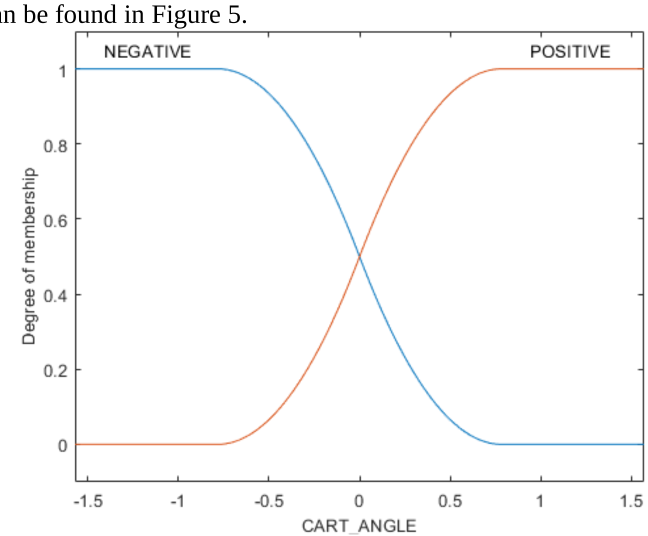

Figure 5 - Sample input membership function

Four possible inputs for the fuzzy logic controller were identified, namely angle, angular velocity, position, and velocity. All inputs utilized a pair of z-shaped and s-shaped membership functions to perform the fuzzification process to obtain POSITIVE and NEGATIVE degrees of membership. These curves were chosen because this paper claims that no mathematical rigor is required when designing the controller and that generally these curves should result in the input being classified as POSITIVE or NEGATIVE a lot faster as it deviates from zero.

The domain of each input has been compiled in Table 2. The maximum value in each domain encouraged the variables to operate within the specified region. Some of these maximum values were developed with the help of intuition. For example, it can be inferred that the robot cannot tilt more than 90° since it will be obstructed by the floor after that point, and as such $\pi/2$ was chosen to be the maximum CART_ANGLE value. Similarly, the size of the simulation environment would prevent the robot from rolling for more than 1m before falling off the edge. The rest of the values were derived by simulating the model in open loop mode.

Table 2 - Domain of inputs for the fuzzy logic controller.

Input	**Domain**
CART_ANGLE (rad)	[$-\pi/2$ $\pi/2$]
CART_ANGULAR_VELOCITY (rad/s)	[-10 10]
CART_POSITION (m)	[-1 1]
CART_VELOCITY (m/s)	[-10 10]

B. Fuzzy Inference

Fuzzy inference is a method by which fuzzy inputs are transformed into fuzzy outputs using a set of fuzzy inference rules. In this paper, the rules were defined with IF-THEN conditions to produce a single output, torque, which will be discussed in depth in section C.

Crucially, to simplify any intuition about the system, a set of conventions was established. The direction of movement was defined as the x-axis. A positive angle was defined as tipping towards the positive x-axis. A positive position was defined as a position within the positive x-axis. A positive torque was defined as a torque which would move the robot in the positive x direction.

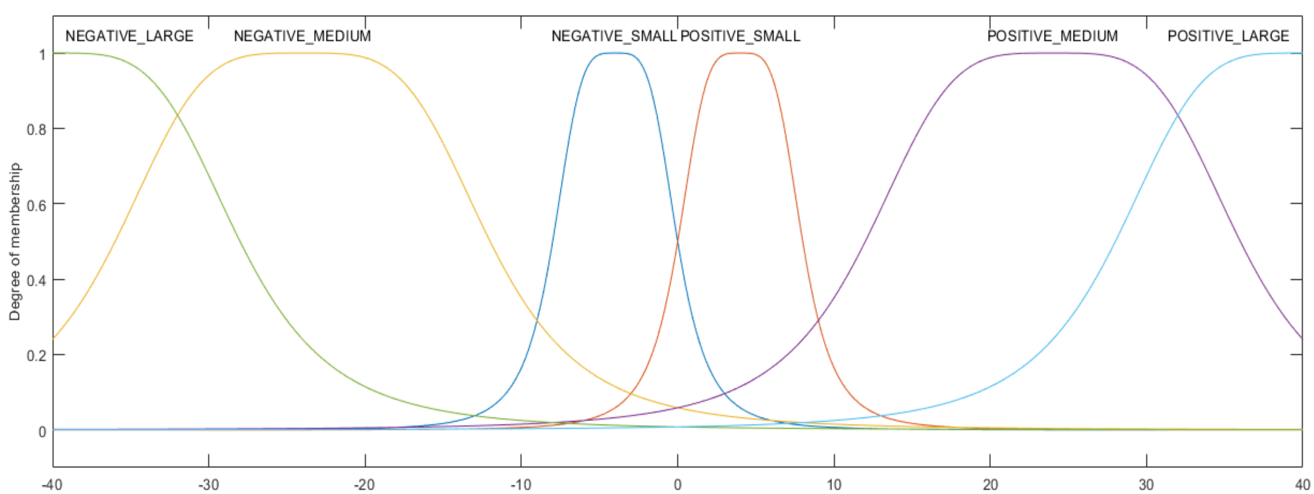

Figure 6 - Output membership function

A summary of the rules developed in this paper which follow the conventions described above is as follows:

1. If CART_ANGLE is NEGATIVE then TORQUE is NEGATIVE_MEDIUM
2. If CART_ANGLE is POSITIVE then TORQUE is POSITIVE_MEDIUM
3. If CART_ANGULAR_VELOCITY is NEGATIVE then TORQUE is NEGATIVE_LARGE
4. If CART_ ANGULAR_VELOCITY is POSITIVE then TORQUE is POSITIVE_LARGE
5. If CART_POSITION is NEGATIVE then TORQUE is POSITIVE _SMALL
6. If CART_ POSITION is POSITIVE then TORQUE is NEGATIVE_ SMALL
7. If CART_VELOCITY is NEGATIVE then TORQUE is NEGATIVE _MEDIUM
8. If CART_VELOCITY is POSITIVE then TORQUE is POSITIVE_MEDIUM

Each of these rules was developed with the help of intuition. For example, in rule 2 it was inferred that when the robot tilts to the right, thus shifting its center of mass to the right with respect to the wheels, it must move to the right to reposition the center of mass directly above the wheels. This was achieved by providing a positive torque to the wheels of the robot. A similar intuitive argument was made for applying a negative torque when the robot tilted to the left.

Note that rules 5 and 6 are perhaps slightly unexpected. To move towards the target position, the robot was actually nudged slightly in the opposite direction. This is because in order to drive in a certain direction while staying balanced, the robot must already be tilting in that direction. Thus, the purpose of the CART_POSITION rules 5 and 6 was to establish the correct tilt, allowing the CART_ANGLE rules 1 and 2 to kick in and push the robot towards the target position.

Finally, the angular velocity and linear velocity were heavily compensated for to stabilize the control of the robot by preventing sudden change in angle and position.

C. Defuzzification

Defuzzification is a process by which fuzzified outputs are converted back into crisp real-world outputs using fuzzy output membership functions. The plant under study in this paper requires only a single output, namely torque and thus only one single output membership function was required as shown in Figure 6. Note that there are various strategies for defuzzifying a fuzzy value. This paper utilized MATLAB's "centroid" method where the centroid of the area under the output fuzzy set was calculated and used as the crisp output value. The area under consideration came directly from the fuzzy inference system, which produced the final shape based on the fuzzy inference rules. The crisp output signified the amount of torque that must be applied to the two-wheeled robot to balance and lead the robot to the target position.

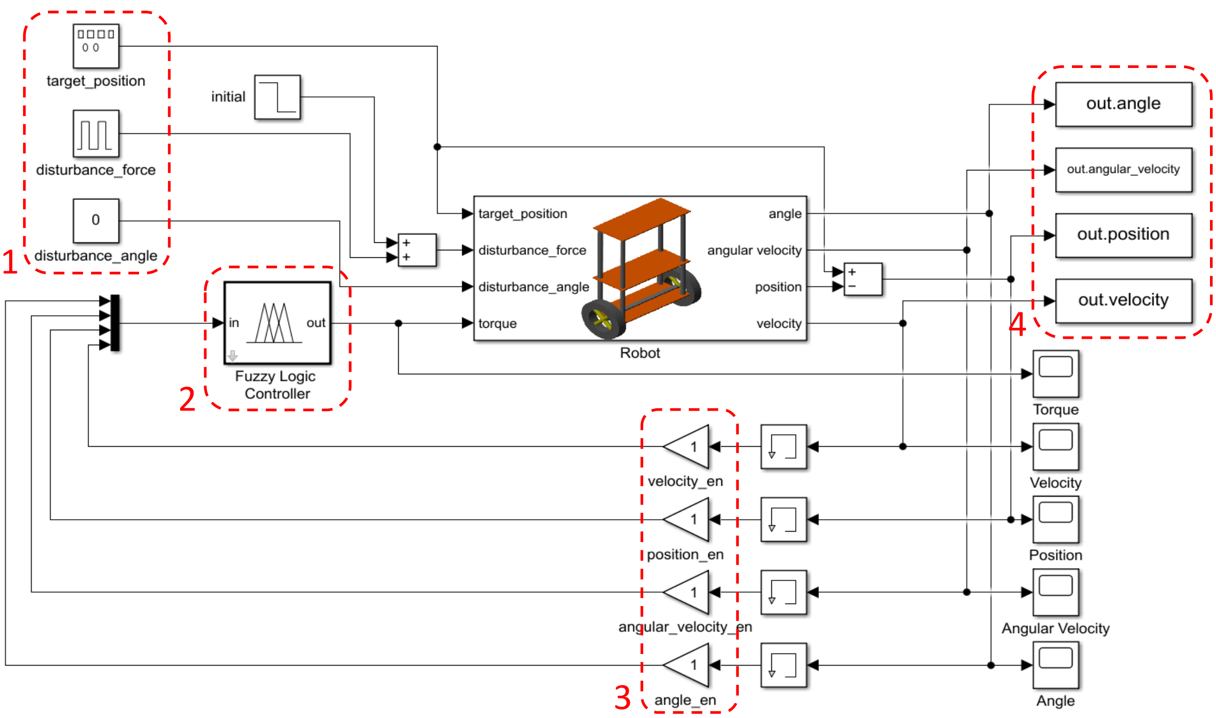

Figure 7 - Test harness to aid in the simulation of the fuzzy logic controller

IV. Simulation Results

To validate the implementation of the proposed fuzzy logic controller, a test harness was constructed in Simulink. The test harness utilized the Simscape Multibody plant model developed in section II and the Mamdani fuzzy logic controller developed in section III and is shown in Figure 7. The test harness consisted of four important regions:

- Region 1 contained input blocks to ease the introduction of disturbances in the simulations.

- Region 2 contained the fuzzy logic controller developed in section III.
- Region 3 contained enable blocks that helped isolate the influence of individual inputs to ease the study of their effects on the plant.
- Region 4 contained the output blocks that logged all simulation data for analysis.

The test harness was utilized to validate the operation, stability and robustness of the proposed fuzzy logic controller, and their results have been outlined in the following sections.

A. Controller Operation Analysis

To validate the stationary balance and positional accuracy of the fuzzy logic controller, the robot was simulated against various target positions without the presence of any disturbances. The controller's operation was analyzed by plotting the positional error and balance error against time as shown in Figure 8 and Figure 9 respectively.

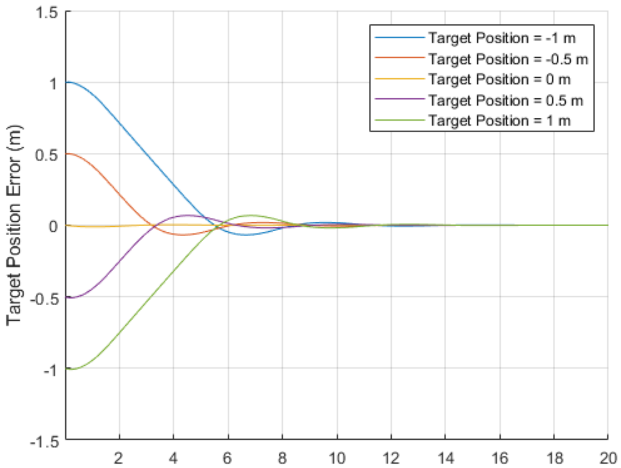

Figure 8 - Fuzzy logic controller's positional control error at various operating points

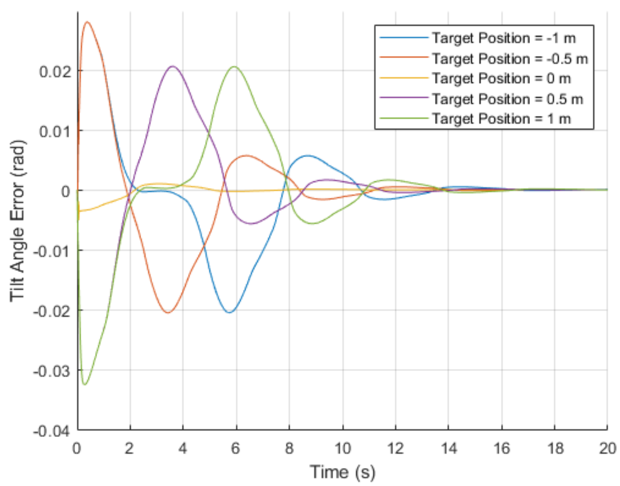

Figure 9 - Fuzzy logic controller's balance control error at various operating points

The above plots confirm that the controller can balance the robot while simultaneously leading the robot to the target position with minimum overshoot and negligible steady state error.

B. Controller Stability Analysis

To validate the stability of the balancing component of the fuzzy logic controller, the robot was simulated in various modes, namely:

- Open loop mode
- Closed loop mode without angular velocity control
- Closed loop mode with angular velocity control

The stability of the controller was analyzed by plotting the tilt angle error of these modes against time as shown in Figure 10.

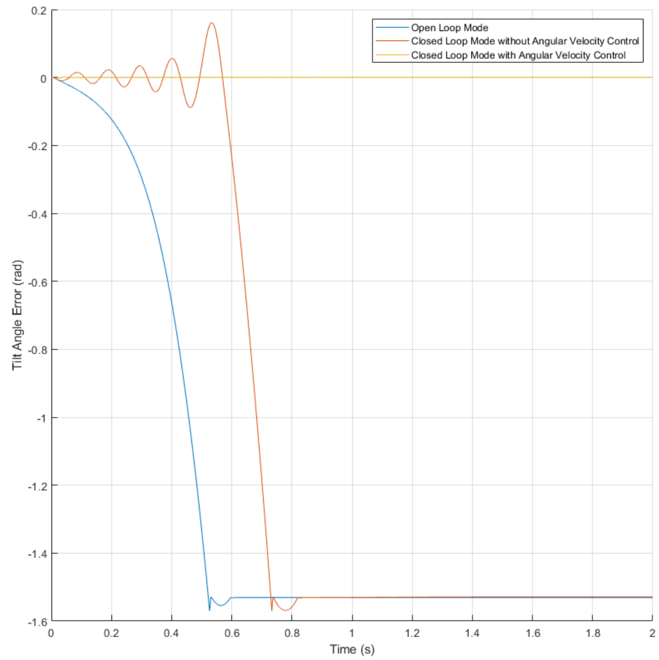

Figure 10 - Stability of the tilt angle control

The plot above confirms that the proposed intuitive fuzzy logic controller (yellow line) is stable as intended. However, without angular velocity control, the robot oscillates with increasing amplitude (red line) until compensation for the extreme tilts is no longer possible and the robot topples over, indicating an unstable controller.

A similar simulation was performed to validate the stability of the positional control, results of which can be found in Figure 11. The plot validates that the velocity control is successful in minimizing the overshoot of the target position (yellow line), making the controller's response to positional error stable.

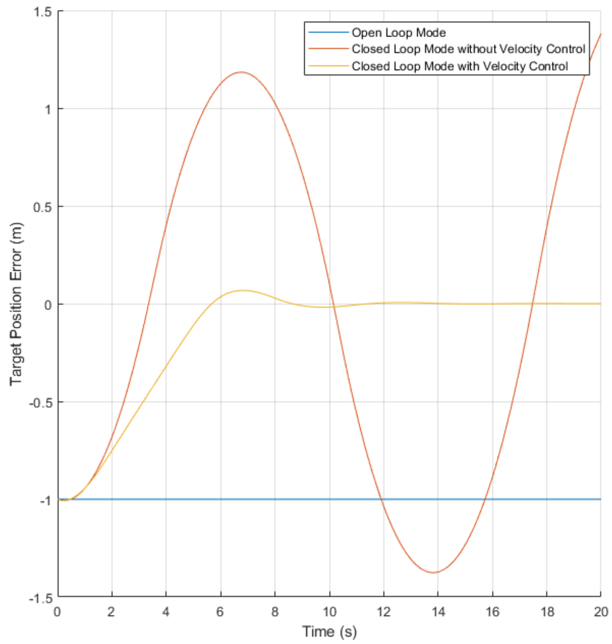

Figure 11 - Stability of the target position control

C. Controller Robustness against Disturbance Force

To validate the robustness of the fuzzy logic controller in the presence of a disturbance force, the robot was subjected to various instantaneous forces applied to its upper plate. The controller's ability to compensate for the disturbance was analyzed by plotting the target position error against time as shown in Figure 12.

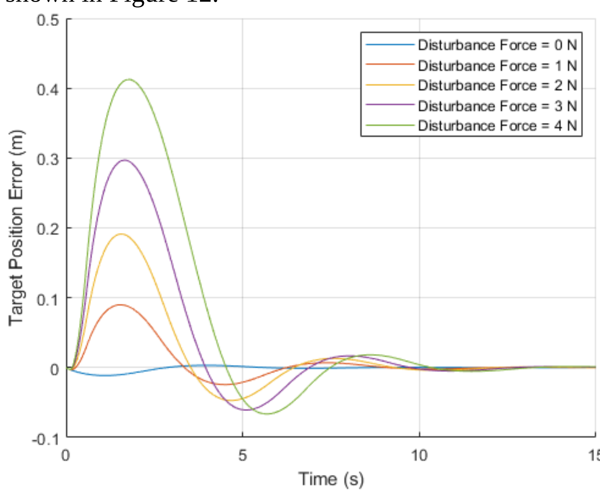

Figure 12 - Robustness of the fuzzy logic controller against disturbance force

The plot above confirms that the proposed controller is robust against disturbance force as it was able to eventually drive the robot back to the target position. It should be noted that increasing the instantaneous force by a huge margin caused the robot to simply topple over, which is an irrecoverable state. This outcome was not plotted as it is unreasonable to apply such large forces to a small object weighing less than 1kg anyways.

D. Controller Robustness against Disturbance Inclination

To validate the robustness of the fuzzy logic controller in the presence of a disturbance inclination, the slope of the floor was varied. The controller's ability to account for the disturbance inclination was analyzed by plotting the target position error against time as shown in Figure 13.

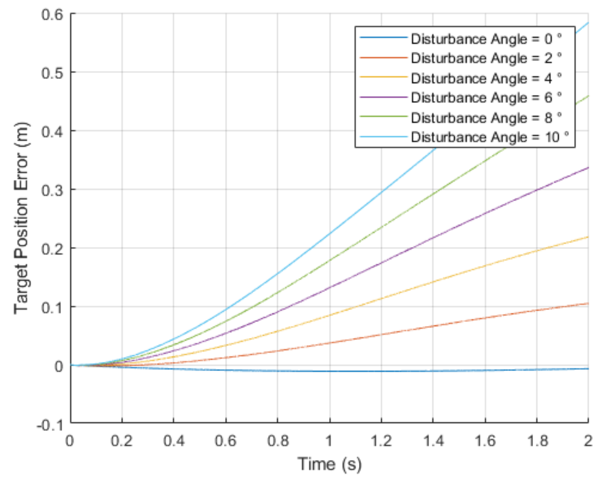

Figure 13 - Robustness of the fuzzy logic controller against disturbance inclination

The plot above clearly indicates that the controller is not robust to changes in inclination. Any change leads to a steady state error in the target position, and in worse cases, drives the robot even further from the target position.

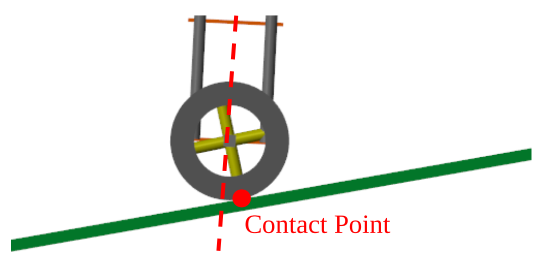

Figure 14 – Cart balance characteristics on an inclined floor

This control failure occurs because the point of contact with the floor and the axis of symmetry of the robot are no longer aligned when balancing on a slope as shown in Figure 14. Thus, to balance, the robot must tilt to compensate. However, the intuitive design process followed in this paper started with the presumption that a cart at 0° is perfectly balanced. The controller therefore fundamentally has the wrong goal in this case and cannot balance the robot on any slope. This result highlights a fundamental limitation of the intuitive design approach when applied to fuzzy logic control. If the designer does not fully understand the system being controlled, fails to imagine some of the types of disturbance which may be encountered or makes assumptions which are not valid in all circumstances, the resulting control system will not be as robust as desired. For critical applications, a more traditional mathematically rigorous design process is therefore still advisable.

V. Conclusion

In this paper, an intuitive technique to develop a fuzzy logic controller was explored. This technique was applied to implement balance and positional control of a two-wheel robot modeled in MATLAB/Simulink using the Simscape Multibody physics toolbox. The simulation results revealed that the proposed fuzzy logic controller worked as intended with minimal overshoot and negligible steady state error. Furthermore, the claim that no mathematical analysis would be required was verified. Key insights included the importance of a simple set of conventions for use in the development of fuzzy rules and the insignificance of the shape of the membership functions beyond having the right general characteristics. The controller was also confirmed to be stable and robust against various disturbance forces up to 4N but was not stable or robust against a change in inclination. The downside to intuition-based design, namely the reliance on the designer for a perfect understanding of the system was clearly outlined.

VI. References

[1] Š. Kozák, "State-of-the-art in control engineering," *Journal of Electrical Systems and Information Technology*, vol. 1, no. 1, pp. 1-9, 2014.

[2] E. H. Mamdani, "Applications of fuzzy algorithms for control of simple dynamic plant," *Proceedings of the IEEE*, vol. 121, no. 12, pp. 1585-1588, 1974.

[3] R.-E. Precup and H. Hellendoorn, "A survey on industrial applications of fuzzy control," *Computers in Industry*, vol. 62, no. 3, pp. 213-226, 2011.

[4] M. Sugeno, "An introductory survey of fuzzy control," *Information Sciences*, vol. 36, no. 1-2, pp. 59-83, 1985.

[5] M. Fliess and C. Join, "Model-free control," in *IFAC*, Saint-Malo, 009,.

[6] S. Syafiie, F. Tadeo, E. Martinez and T. Alvarez, "Model-free control based on reinforcement learning for a wastewater treatment problem," *Applied Soft Computing*, vol. 11, no. 1, pp. 73-82, 2011.

[7] C. Lauwerys, J. Swevers and P. Sas, "A model free control design approach for a semi-active suspension of a passenger car," in *American Control Conference*, 2005.

[8] K. Al-mutib, E. A. Mattar and N. M. A. Shah, "Performance Analysis of Fuzzy Logic And PID Controller for PM DC Motor Drive," 2019.

[9] M. U. Hassana, P. Fua, Z. Songa, X. Chena, X. Zhanga and M. Humayum, "Feasibility analysis of fuzzy logic control for ITER Poloidal field (PF) AAC/DC converter system," *Fusion Engineering and Design*, vol. 118, pp. 11-19, 2017.

[10] N. E. Benchouia, A. Derghal, B. Mahmah, N. E. Benchouia, A. Derghal and B. Mahmah, "An adaptive fuzzy logic controller (AFLC) for PEMFC fuel cell," *International Journal of Hydrogen Energy*, vol. 40, no. 39, pp. 13806-13819, 2015.

[11] J. Wu, W. Zhang and S. Wang, "A Two-Wheeled Self-Balancing Robot with the Fuzzy PD Control Method," *Mathematical Problems in Engineering*, vol. 2012, pp. 21-13, 2012.

[12] A. G. Perry, G. Feng, Y.-F. Liu and P. C. Sen, "A Design Method for PI-like Fuzzy Logic Controllers for DC–DC Converter," *IEEE*, vol. 54, no. 5, pp. 2688-2696, 2007.