STATISTICS, OPTIMIZATION AND INFORMATION COMPUTING
Stat., Optim. Inf. Comput., Vol. **12**, January 2024, pp 182–199.
Published online in International Academic Press (www.IAPress.org)

Implementation of Fuzzy Logic Controller Algorithm with MF Optimization on FPGA

Samet Ahmed *, Kourd Yahia

Laboratory of Electrical Engineering and Renewable Energy LEER,Mohamed-Cherif Messaadia University,Souk Ahras, Algeria

Abstract In this work, we propose the design and implementation of a parallel-structured fuzzy logic controller with integral action and anti-windup. The Grey Wolf Optimization (GWO) technology is used to optimize fuzzy rules, which allows for the complicated algebraic ideas of type 1 fuzzy logic algorithms to be reduced to straightforward numerical equations for FPGA target implementation. The techniques for operating a geared DC motor are optimized by the membership function structure of our controller's data propagation. Our proposed controller was implemented in Xilinx System Generator (XSG) and co-simulated on hardware and software with VIVADO and XSG tools.

Keywords Geared DC Motor, GWO, Fuzzy Logic Controller, Fixed Point Design, Xilinx System Generator, FPGA

DOI: 10.19139/soic-2310-5070-1790

1. Introduction

Due to their simplicity, reliability, and ease of control, Direct Current (DC) motors find extensive utilization across a diverse range of industries and applications. These motors are commonly employed in electric vehicles, robotics, industrial equipment, appliances, aerospace and defense systems, renewable energy systems, medical devices, and amateur projects. The primary function of DC motors is to convert electrical energy into mechanical motion, enabling efficient and precise control for various tasks, including propulsion, robotic movements, material handling, fluid transportation, and more [30, 21, 14]. Their widespread adoption and adaptability have made them an indispensable component of modern technologies spanning different sectors.

Ongoing research on DC motors aims to enhance their performance and control in diverse applications. This research involves gaining insights into engine characteristics, developing mathematical models and simulations, exploring control techniques and algorithms, and investigating sensorless control methods. By addressing challenges such as nonlinearities, uncertainties, and parameter variations, the ultimate goal is to improve the efficiency, reliability, and controllability of DC motors [12]. Research plays a crucial role in advancing the development of Direct Current (DC) motor systems that offer enhanced reliability and precision, tailored to the specific requirements of diverse industries [12, 7]. One area of focus in DC motor research involves achieving precise and efficient speed control in various industrial applications. Initially, open-loop control methods were utilized, but their accuracy and performance were compromised in the presence of disturbances and load variations. Consequently, closed-loop control techniques, notably PID control, gained popularity. PID control involves using feedback to continuously adjust control inputs based on the disparity between the desired motor speed and its actual speed. Researchers have explored various approaches to fine-tune PID parameters for achieving improved performance [6, 19, 10]. Fuzzy logic control is a control technique used to regulate DC motors and other systems

*Correspondence to: Samet Ahmed (Email: a.Samet@univ-soukahras.dz). Laboratory of Electrical Engineering and Renewable Energy LEER, Mohamed-Cherif Messaadia University, Souk Ahras, Algeria.

ISSN 2310-5070 (online) ISSN 2311-004X (print)
Copyright © 2024 International Academic Press

by incorporating human-like reasoning and linguistic variables. This approach offers flexible and robust control strategies, effectively handling imprecision and uncertainty present in real-world systems. Fuzzy logic control finds applications in various areas, such as speed control, position control, torque control, robotics, automation, and energy efficiency optimization for DC motors. By enabling precise and adaptive control, it effectively addresses non-linearities and varying operating conditions, providing intelligent control strategies for diverse applications [33, 29, 27].

The main challenges posed by the problem can be summarized into two key issues: the non-linearity of the DC motor and the drawbacks associated with fuzzy controller systems, which are expressed as follows [27, 16, 34]:
1. Complex mathematical computations are necessary for fuzzy controllers.
2. The essential information provided by experts may not always be optimal or precise.
3. The challenge lies in determining the ideal values for input and output gains due to the numerous multiplications and additions involved in fuzzy calculations.

Xilinx provides a high-level design tool called XSG, which is specifically designed for rapid prototyping of Digital Signal Processing (DSP) algorithms using their FPGAs. This tool seamlessly integrates with MATLAB/Simulink, enabling designers to create and simulate intricate DSP algorithms within a user-friendly graphical environment. By automating low-level implementation details, XSG simplifies the design process and supports various Xilinx FPGA families. Leveraging FPGA parallelism and configurability, System Generator (XSG) facilitates efficient implementation, resulting in reduced development time and a higher level of abstraction compared to traditional RTL design. This feature allows for faster iterations and system exploration, making it a potent tool for DSP algorithm design and implementation on Xilinx FPGAs [4, 13, 17, 15].

Incorporating FPGAs in DC motor speed control offers several advantages, such as enhanced flexibility, high-speed processing capabilities, integration of multiple functions, customizability, real-time monitoring, reduced system cost, and potential energy efficiency [39, 23, 22, 36, 11, 25, 1, 38, 32, 20].

1.1. Literature review

DC motor speed control has been extensively studied in the field of control systems. Researchers have explored various control techniques and algorithms to achieve precise and efficient speed regulation. The literature review highlights significant findings and approaches, including PID control [8], adaptive control [7], fuzzy logic control [35], neural network control [30], sliding mode control [15], optimal control [5], and hybrid control methods [10]. The main focus of these studies is on optimizing control performance, dealing with non-linearities and uncertainties, and achieving robust and efficient speed control in DC motors.

In the subsequent sections, we will briefly examine the utilization of Fuzzy Logic Controller (FLC) structures in the control of DC motors. Additionally, we explore the application of Field-Programmable Gate Arrays (FPGAs) for control purposes in embedded systems.

In [24] the primary aim of this research is to introduce and justify a defuzzification approach founded on the global structure of a fuzzy controller. Furthermore, this study demonstrates how this method can assist designers in attaining their objectives with ease and in a systematic manner. Various defuzzification methods exist, including COG (Center of Gravity), MOM (Mean of Maxima), WAF (Weighted Average Formulae), QM (Quality Method), and WABL (Weighted Average Based on Levels) , among others. However, the challenge arises from the fact that each method yields distinct outcomes for the same problem. Consequently, determining which one is the most suitable becomes a crucial question.

In one study [28], the primary objective is to enhance the performance of Electric Vehicles (EV) battery chargers by optimizing the Vienna rectifier with a Voltage-Oriented Controller (VOC) PI controller. The researchers aim to optimize the VOC PI controller using the Particle Swarm Optimization (PSO) technique implemented through MATLAB software [37]. Comparing it to the Genetic Algorithm (GA) technique, the PSO technique offers a simpler optimization process with fewer heuristic variables, making it the preferred method for optimizing the PI controller of VOCs.

In another study [11], the authors investigate the performance of Induction Motor (IM) drives utilizing Indirect Field Oriented Control (IFOC) along with a simple self-tuning mechanism for adjusting the output Scaling Factor (SF) of the primary FLC. They analyze the impact of varying rule sizes within the FLC on both the drive's

performance and the computational load of the system. The study provides a comprehensive analysis of these effects.

In [18], the authors set out to present a prototype of a unicycle robot. They initiated the process by developing a model of a two-armed inverted pendulum. Subsequently, they made enhancements by incorporating various components, ultimately intending to replace the arms with a gyroscope. As mentioned earlier, when a perturbation τ was applied to the two arms, it led to rotations of the arms, denoted as α and β, and caused the pendulum angle θ to oscillate around its equilibrium point. To ensure stability in the system, the authors proposed and modeled a fuzzy logic control technique.

In [25], the authors propose a straightforward Fuzzy Control Strategy (FCS) for Self-Balancing Robot (SBR) systems and demonstrate an implementation technique using Look-Up Tables (LUTs) for Fuzzy Logic Controllers (FLCs). This approach is specifically tailored for educational purposes.

In [1], the authors employ a Fractional-Order PID (FOPID) controller to regulate the reference current of a Brushless Direct Current (BLDC) motor, and a Fuzzy Logic Controller (FLC) is utilized to manipulate the DC bus voltage of the inverter. To tune the FOPID controller's parameters, they develop a modified Harmony Search (HS) metaheuristic technique. The article presents a strategy for controlling the speed of a BLDC motor, involving the use of a FLC to adjust the DC inverter voltage and a FOPID controller to regulate the BLDC motor's reference current through the inverter gate circuit.

In [38], the authors introduce a novel PID controller called DFPID-HSA (Dual Fuzzy Logic Systems with Harmony Search Algorithm optimization). They incorporate dual Fuzzy Logic Systems (FLSs) and utilize the Harmony Search Algorithm (HSA) optimization technique to enhance the speed control performance of BLDC motors. DFPID-HSA comprises two FLSs: FLS1, which adjusts the three coefficients of the PID controller based on the system error and error change rate, and FLS2, which is optimized using HSA (referred to as HSA-F2) to achieve precise correction of the three coefficients.

The key works in [32] and [20], representing a branch of research in FPGA implementation, are the subject of our study. In [32], the authors propose a new fuzzy logic algorithm and compare it with the traditional fuzzy logic algorithm. They provide a detailed discussion of the comparative results obtained from the analysis. In [20], the authors introduce optimization techniques using an enhanced Genetic Algorithm (GA) to improve the inherent qualities of fuzzy controllers. They compare the outcomes achieved with various types of fuzzy controllers mentioned in other research papers and introduce specific strategies for implementing an optimal fuzzy controller on FPGA. The proposed method's scheme is represented in Figure 1.

1.2. Motivations and Contributions

Reasons for regulating the speed of DC motors encompass the necessity for exacting command, enhanced energy usage, torque management, and seamless system amalgamation. This facilitates meticulous performance, optimized energy utilization, precise torque generation, and effortless integration into intricate setups. Difficulties associated with governing the speed of DC motors involve intricate control demands, dynamic responsiveness, motor traits and confines, susceptibility to Electromagnetic Interference (EMI), as well as the expenses and intricacy linked with control hardware, along with upkeep and dependability concerns. To surmount these hurdles, it necessitates advanced control algorithms to tackle nonlinear tendencies, counterbalance motor restrictions, handle EMI susceptibilities, consider cost and intricacy aspects, and ensure appropriate maintenance to ensure dependable functionality. Fixed-point arithmetic plays a significant role in influencing the cost and performance of a circuit. As the size of operators decreases, the required precision becomes more critical. The main goal is to achieve a small and fast circuit while ensuring an acceptable level of precision. Consequently, this research paper aims to make two contributions:

1. Control algorithms: The primary focus is on developing a novel control algorithm based on simple and easily implementable equations to enhance the accuracy, stability, and response characteristics for regulating the speed of DC motors.

2. Power consumption strategies: To minimize power losses and improve overall energy efficiency, implementation techniques on FPGA boards, such as parallelism and data propagation pipeline, are employed. These strategies are designed to optimize power consumption by capturing global energy efficiency.

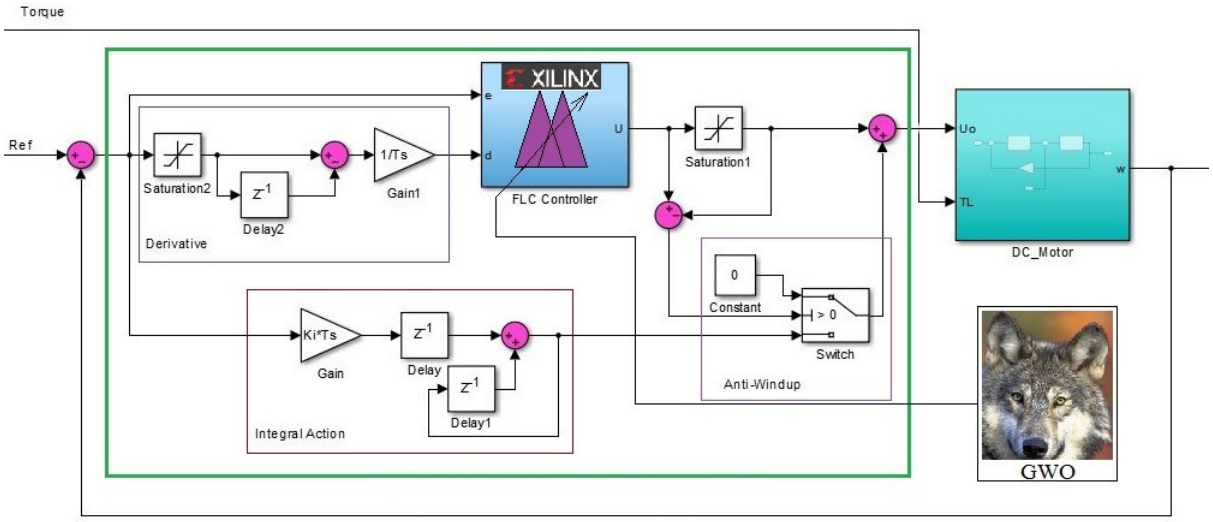

Figure 1. Integral FL controller with anti-windup technique.

2. Electromachanical system

Using an electrical input to create a mechanical output or a displacement output from a voltage input, a motor is an electromechanical component. The following graphic displays the geared DC motor's schematic [30, 2].

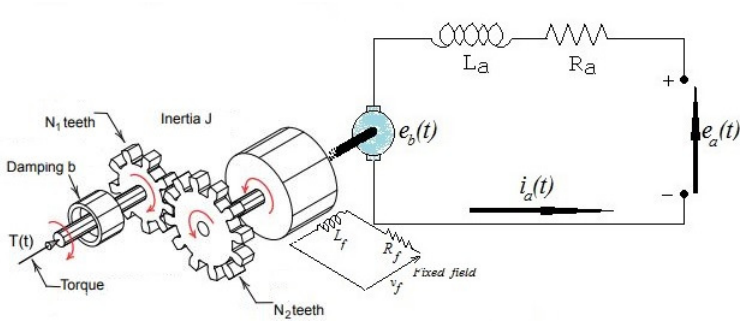

Figure 2. Electromechanical system modeling.

In industrial applications, a motor that drives the load is typically linked to gears. You may increase speed with less torque or decrease speed with more torque while using gears. When referring to the motor with gear, J_T stands for total inertial momentum and B_T for total viscous friction. The electromechanical system's parameters are presented in table 1. A second order system is defined as any system that is governed by a second order linear

Table 1. The Electromechanical System's Parameters.

$R_a[\Omega]$	0.5	ki[N.m/A]	0.05
L_a[H]	1.5	k_b[V.s/rad]	0.05
J_T[Kg.m^2]	0.0025	*Nominal Voltage*[V]	23.5
B_T[N.m.s/rad]	0.0010	*Operating Range* [V]	**8-35**

differential equation with constant coefficients whose transfer function is:

$$H(s) = \frac{Y(s)}{U(s)} = \frac{K}{\frac{1}{\omega_n^2}s^2 + \frac{2\zeta}{\omega_n}s + 1} \tag{1}$$

The following is the closed-loop transfer function:

$$H_C(s) = \frac{H_O(s)}{1 + H_O(s)} \tag{2}$$

Hence:

$$H_C(s) = \frac{\Omega(s)}{E_a(s)} = \frac{\frac{100}{106}}{\frac{7.5}{106}s^2 + \frac{5.5}{106}s + 1}\left[\frac{rad}{s.V}\right] \tag{3}$$

Figure 3 show the dynamic behavior of the error and its derivative of our system studied later.

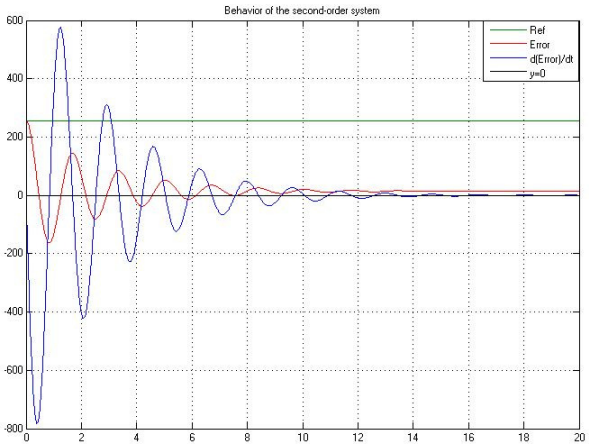

Figure 3. Second-Order System Behavior(time scale 2s).

Where:

$$\begin{cases} K_C = \frac{100}{106} = 0.9434\frac{rad}{s.V} \\ \omega_n = 3.7594\frac{rad}{s}\ ,\ \zeta = \ 0.0975 \\ \omega_d = 3.7415\frac{rad}{s}\ ,\ T_d = 1.6793s \end{cases} \tag{4}$$

We have the following continuous dynamic system with *u(t)* is the applied control (see Figure 3), the time response is given by:

$$y(t) = KU_0(1 - e^{-\zeta\omega_n t}(\cos\omega_n\sqrt{1-\zeta^2}t + \frac{\zeta}{\sqrt{1-\zeta^2}}\sin\omega_n\sqrt{1-\zeta^2}t)) \tag{5}$$

The equation that represents the error applied to system is expressed by:

$$e(t) = U_0(1-K) + KU_0e^{-\zeta\omega_n t}\left(\cos\omega_n\sqrt{1-\zeta^2}t + \frac{\zeta}{\sqrt{1-\zeta^2}}\sin\omega_n\sqrt{1-\zeta^2}t\right) \tag{6}$$

The error's derivative function is:

$$\dot{e}(t) = -KU_0\frac{\omega_n}{\sqrt{1-\zeta^2}}e^{-\zeta\omega_n t}\sin\omega_n\sqrt{1-\zeta^2}t \tag{7}$$

If the derivative function's derivative is equal to zero, the derivative function's minimum is equal to a quarter of the system's pseudo-period. This is shown in (8)

$$\sin \omega_n \sqrt{1-\zeta^2} t = 1 \;\Rightarrow\; t = \frac{T}{4} \tag{8}$$

from which the numerical application gives the results in (9)

$$\dot{e}\left(\frac{T}{4}\right) = \dot{e}_{Min} = -KU_0 \frac{\omega_n}{\sqrt{1-\zeta^2}} e^{-\zeta\omega_n \frac{T}{4}} = -K_0 U_0 = -3.0552 U_0 \tag{9}$$

This involves:

$$e_{Max} = 250 \;\Rightarrow\; \dot{e}_{Min} = -763.7966 \tag{10}$$

Finally, we can deduce the range of work to which we will apply our controller in the next part of this article. for practical reasons, that is to say, reducing noise and exploiting average speeds in order to have performance in the results of simulations (11).

$$\begin{cases} -256 \leq e \leq 256 \\ -1024 \leq \dot{e} \leq 1024 \end{cases} \tag{11}$$

3. Fuzzy Logic Controller

3.1. Type-1 fuzzy logic systems

The following membership function, *μA(x)*, which accepts values in the range [0,1], describes a type-1 fuzzy set in the universe *X*. It may be modeled as a collection of ordered pairs of an element and the degree to which the element is a member of the set.

$$(A = \{A, \mu A(x) |\; x \in X\} \tag{12}$$

Where μA: $X \in [0, 1]$, in this expression, μA(x) stands for the degree of membership of the element $x \in X$ to the set A. Any fuzzy logic-based system has three phases: fuzzy inference, fuzzy inference, and fuzzy defuzzification. The set point andv the measured and computed signals from the encoder are expressed linguistically via the input linguistic variables for the fuzzy logic controller (FLC). Fuzzification involves employing membership functions to convert entries into linguistic variables. The two inputs to FLC are derivative of Error equal to d and Error equal to e (Set point-measured speed, derivative of error). In the output linguistic variable, the value applied to the FLC actuator for dc motor control is stated *(u)*. Triangular membership functions are used to fuzzily the input or output. For the fuzzifier application, it is necessary to identify the range of fuzzy variables associated to the crisp inputs and output values. NS=negative small, NB=negative big, NM=negative mean the fuzzy sets utilized are tiny, ZE=zero, PS=positive small, PM=positive mean, and PB=positive big. The Membership function for input and output linguistic variables is shown in equations [27, 18, 18].

Inputs *e* and *de* are represented by a symmetrical triangular function:

$$\mu_{e_i}\left(x, h_{i\,i}^1, a_i, l_i^1\right) = \begin{cases} 0 & x \leq h_i^1 \\ \frac{x-h_i^1}{a_i-h_i^1} & h_i^1 \leq x \leq a_i \\ \frac{a_i-x}{l_i^1-a_i} + 1 & a_i \leq x \leq l_i^1 \\ 0 & l_i^1 \leq \quad x \end{cases} \tag{13}$$

$$\mu_{de_i}\left(x, h^2_{i\,i}, b_i, l^2_i\right) = \begin{cases} 0 & x \le h^2_i \\ \frac{x-h^2_i}{b_i-h^2_i} & h^2_i \le x \le b_i \\ \frac{b_i-x}{l^2_i-b_i}+1 & b_i \le x \le l^2_i \\ 0 & l^2_i \le \ x \end{cases} \tag{14}$$

Triangular function that represents the output *u*:

$$\mu_{u_i}\left(x, h^3_{i\,i}, c_i, l^3_i\right) = \begin{cases} 0 & x \le h^3_i \\ \frac{x-h^3_i}{c_i-h^3_i} & h^3_i \le x \le c_i \\ \frac{c_i-x}{l^3_i-c_i}+1 & c_i \le x \le l^3_i \\ 0 & l^3_i \le \ x \end{cases} \tag{15}$$

With

$$\begin{cases} a_i = (h^1_i + l^1_i)/2 \\ b_i = (h^2_i + l^2_i)/2 \\ c_i = (h^3_i + l^3_i)/2 \end{cases} \tag{16}$$

The components of the vectors to be optimized are given by the following system of equations:

$$\begin{cases} x_i = (l^1_i - a_i) \quad a[-256 \ , 256] \\ y_i = (l^2_i - b_i)\ a[-1024, 1024] \\ z_i = (l^3_i - c_i) \qquad a[-12 \ , 12 \ ] \end{cases} \tag{17}$$

It is a technique that characterizes the study of this article that the values in the intervals in the function pertaining to the linguistic variables of inputs and outputs are in power of 2.

$$\begin{cases} a_i \ \ a \ \{-256, \ -32, \ 0, \ 32, \ 256\} \\ b_i \ a \ \{-512, \ -128, \ 0, \ 128, \ 512\} \\ \quad c_i \ a \ \{-8, \ -4, \ 0, \ 4 \ , 8 \ \} \end{cases} \tag{18}$$

Inference connects input linguistic variables to output linguistic variables, which are then expressed as linguistic variables (the inference engine used is Max-min from Mamdani) [27]. The rule base for FLC is shown in Figure 4 Which command to employ is indicated by a fuzzy controller's fuzzy output. The predicted output is also a real

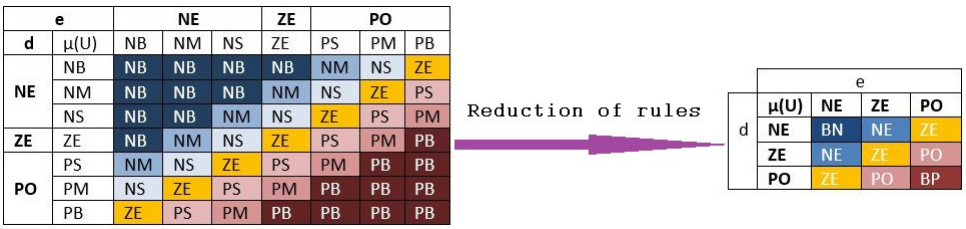

Figure 4. Fuzzy logic controller rules reduction.

number if the input is a real number. However, since the output remains fuzzy even when the input is crisp, this generally does not happen with fuzzy controllers. Determining a method for defuzzing the output and generating a true number that will finally show which control to employ will be necessary. There are several defuzzification methods available.

$$u = \frac{\sum_{j=1}^{n} \mu_j(u) C_j}{\sum_{j=1}^{n} \mu_j(u)} \tag{19}$$

U is the centroid, which is sometimes referred to as the center of mass or gravity. It is suggested to use this defuzzification method even if it is the most complex. The three parts of a fuzzy controller are shown in the following diagram. By dissecting the building blocks, we can observe that the intermediate part houses the two main working components of the controller, the rules and inference engine. This sort of controller uses the centroid technique in Figure 5.

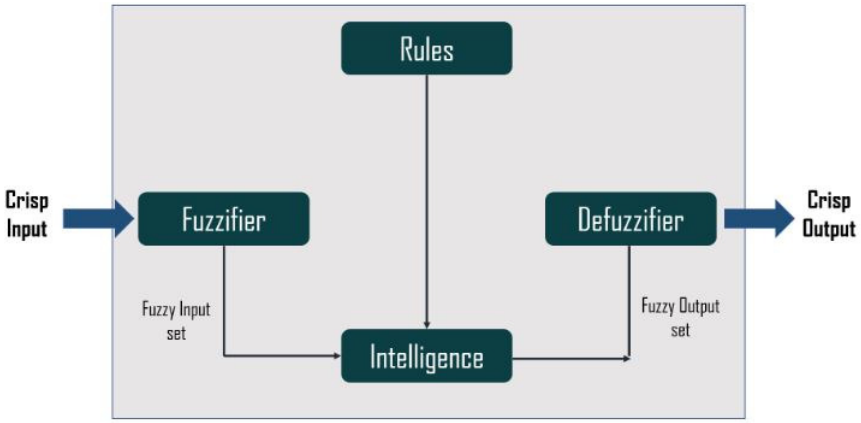

Figure 5. type I fuzzy Logic System Structure.

The membership function for the input or output linguistic variables is represented graphically in Figure 6.

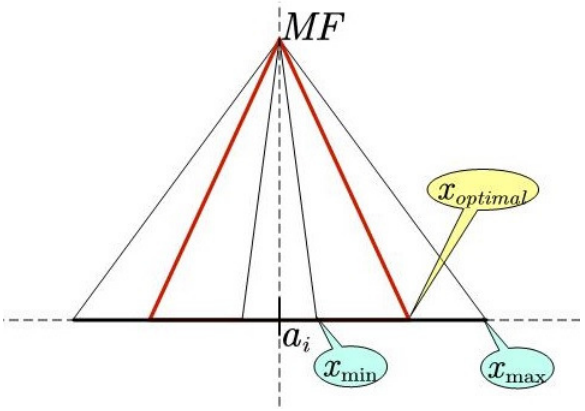

Figure 6. Behavior of Membership function for inputs and outputs linguistic variable of FL Controller.

3.2. Proposed Centroid Calculation Algorithm

The Mamdani's min-max implication is a fuzzy inference technique that use the min operator to determine the least value of the membership degree between two inputs obtained from the rule selector, and the max operator to aggregate all rules for each output variable. To obtain the crisp value of the fuzzy value, the final output is determined using the weighted average defuzzification approach as shown in (20). The value of each resulting surface of each rule is computed in the first stage of this procedure (19). The determined surface is the value of $R_k(u)$, which corresponds to the controller's $r_k(u)$ rule's membership function $\mu_i(u)$. All the values of the surfaces

$R_k(u)$ calculated as follows will be stored in a table R:

$$\begin{cases} R(u) = \int_{-L}^{L} (\mu_k(u) - f(u))du \\ R_k(u) = A(k)(2\mu_k(u) - \mu_k(u)^2) \\ k = 1, 2, \ldots n = 9 \end{cases} \tag{20}$$

In the second step an R array is computed of which:

$$H_k(u) = c_k R_k(u) \tag{21}$$

Finally, we determine the centroid using ci, the coefficient of the triangle function of (22).

$$u = \frac{\sum_{k=1}^{n} c_k R_k(u)}{\sum_{k=1}^{n} S_k(u)} = \frac{\sum_{k=1}^{n} H_k(u)}{\sum_{k=1}^{n} R_k(u)} \tag{22}$$

If we utilize the previous equation's two-dimensional notation, we get:

$$u = \sum_{1}^{n} \sum_{1}^{n} \frac{H(i,j)}{R(i,j)} \tag{23}$$

The first initialization, the second fuzzification, the third defuzzification process, and the fourth computation of the center of gravity are the four parts of the algorithm.

4. Grey Wolf Optimization

4.1. Overview

Mirjalili et al. [3] suggested GWO. The social order and hunting practices of wolves served as the basis for the mathematical model that makes up the Grey Wolf algorithm. Alpha, beta, delta, and omega make up the four tiers of the grey wolf social structure, from highest to lowest. The top wolves in a wolf pack are called alphas. They are in charge of the wolf pack, and the others must obey their orders. Beta is the next position in the grey wolf hierarchy. The alpha wolf's beta can help with decision-making or other group tasks. It provides feedback to the alpha and reinforces his or her directives. Omega is the grey wolf with the lowest rating. They are obligated to constantly yield to the other dominant wolves. Next to the alphas and betas in the pack hierarchy, delta wolves are in charge of the omega. First, the objective function is built in accordance with the challenges and the best answer is considered to be alpha in order to mathematically simulate the social hierarchy of grey wolves.

4.2. Encircling Prey

When hunting, grey wolves encircle their victim. The following equations allow a grey wolf to update its position within the area around the prey at any random place during this procedure [3, 31].

$$D = \left| \overrightarrow{C}.\overrightarrow{X_p}(t) - \overrightarrow{X(t)} \right| \tag{24}$$

$$\overrightarrow{X}(t+1) = \overrightarrow{X_p}(t) - \overrightarrow{A}.\overrightarrow{D} \tag{25}$$

Where t is the current iteration, $X(p)$ denotes the position vector of the prey, X denotes the position vector of a grey wolf, C and A are the coefficient vectors that are determined using the following equations.

$$\overrightarrow{C} = 2.\overrightarrow{r_2} \tag{26}$$

$$\overrightarrow{A} = 2.\overrightarrow{d}.\overrightarrow{r_1} - \overrightarrow{d} \tag{27}$$

where r_1 and r_2 are random vectors between [0, 1], and components of d are linearly lowered from 2 to 0 during the course of repetitions.

4.3. Hunting

The dominant wolf leads the hunt. The beta and delta wolves occasionally go hunting. It is assumed that the alpha, beta, and delta wolves have superior knowledge of the possible location of prey in order to mathematically describe the hunting behavior of grey wolves. The first three best answers obtained are therefore kept, and the other search agents are compelled to update their positions in accordance with the best search agents' locations. The following equations can be used in this regard [3, 31].

$$\begin{cases} \overrightarrow{D_\alpha} = \left|\overrightarrow{C_1}.\overrightarrow{X_\alpha} - \overrightarrow{X}\right|, & \overrightarrow{X_1} = \overrightarrow{X_\alpha} - A_1.\overrightarrow{D}_\alpha \\ \overrightarrow{D_\beta} = \left|\overrightarrow{C_2}.\overrightarrow{X_\beta} - \overrightarrow{X}\right|, & \overrightarrow{X_2} = \overrightarrow{X_\beta} - A_2.\overrightarrow{D}_\beta \\ \overrightarrow{D_\delta} = \left|\overrightarrow{C_3}.\overrightarrow{X_\delta} - \overrightarrow{X}\right|, & \overrightarrow{X_3} = \overrightarrow{X_\delta} - A_3.\overrightarrow{D}_\delta \end{cases} \tag{28}$$

4.4. Attacking Prey

When the victim stops moving, the grey wolves attack it to complete their hunt. The value of D is gradually dropped over the course of iterations from 2 to 0, which also reduces the fluctuation range of A, in order to build the mathematical model of a Gray Wolf approaching prey. The next position of a search agent can be at any location between its present position and the position of the prey when the random values of A are in the range [-1, 1]. When A is smaller than 1, grey wolves attack the victim [9, 26].

4.5. Search for Prey

Grey wolves split from one another in order to escape the local optimal solution and go for a more suitable prey. A is used with random values bigger than 1 or less than -1 to force the search agent to diverge from the prey in order to mathematically simulate the grey wolf's divergence traits. As a result, the GWO algorithm may search worldwide and emphasizes exploratory qualities. When abs(A) is larger than 1, grey wolves depart from their prey in pursuit of fitter prey [3, 31]. In (28) the prey position corresponds, respectively, to the best position attained by each of the wolf ranks, α, β and γ. All wolf population positions are updated using the following (29) and the general GWO is presented in Table 2.

$$\overrightarrow{X}(t+1) = \frac{\overrightarrow{X_1} + \overrightarrow{X_2} + \overrightarrow{X_3}}{3} \tag{29}$$

4.6. Cost and parameters

In this paper the construction of objective function is shown in (31)

$$J = \int_0^\infty e(t)^2 dt \tag{30}$$

More precisely:

$$J = \sum_{k=0}^{L} e(k)^2 \tag{31}$$

The GWO method must minimize the cost J while optimizing the vectors X_{in} and X_{out}. Our approach entails optimizing the input and output individually. With maximum number of iteration equal to 100 and number of search agent is 30.

$$\overrightarrow{X}_{in} = \begin{bmatrix} x_i \\ y_i \end{bmatrix} \quad i = -1, 0, 1 \tag{32}$$

$$\overrightarrow{X}_{out} = [z_i] \quad i = -2, -1, 0, 1, 2 \tag{33}$$

Table 2. GWO Algorithmrs

	Pseudo Code
1	*Initialize the grey wolf population X_i, i=1,n*
2	*Initialize a, A and C*
3	*Calculate the fitness of each search agent*
4	*X_α the best search agent*
5	*X_β the best search agent*
6	*X_γ the best search agent*
7	----------
8	***While*** *t< iteration* ***do***
9	***For*** *each search agent* ***do***
10	*Randomly initialize r_1 and r_2*
11	*Update the position of the current search agent by Equation (29)*
12	***End***
13	----------
14	*Update a, A and C*
15	*Calculate the fitness of all search agent*
16	*Update X_α, X_β and X_γ*
17	*T=t+1*
18	***End***
19	----------
20	*Return X_α*

The bounds of the vectors X_{in} and X_{out} are given by the following memberships:

$$x_i \in \{2^{-4}, 2^{-3}, 2^{-2}, 2^{-1}, 2^0, 2^1, 2^2, 2^3, 2^4, 2^5, 2^6, 2^7, 2^8\} \tag{34}$$

$$y_i \in \{2^{-4}, 2^{-3}, 2^{-2}, 2^{-1}, 2^0, 2^1, 2^2, 2^3, 2^4, 2^5, 2^6, 2^7, 2^8, 2^9, 2^{10}\} \tag{35}$$

$$z_i \in \{2^{-4}, 2^{-3}, 2^{-2}, 2^{-1}, 2^0, 2^1, 2^2, 2^3, 2^4\} \tag{36}$$

5. Implementation Using Xilinx System Generator

A powerful tool for putting control methods that have been theoretically validated in Matlab Simulink into practice is the Xilinx® System Generator. The co-simulation of the researched models is carried out directly on the FPGA board, and parallel VHDL code generation is done in accordance with the model's methodology. The Xilinx Vivado platform and Matlab Simulink are used to polish the method.

5.1. FLC Controller Block Diagram

All of the components of our proposed controller are of the same sort of data, whether in input or output. Understanding binary arithmetic and its accuracy is necessary for fixed point data transfer. T_s=0.001s, K_i=0.0250 will be used for the remainder of the co-simulation. The division block used in this design, which is why the radix algorithm is the simplest, is the most basic unit of the system generator. As a result, excessive use of DSPs should be avoided (Figure 7).

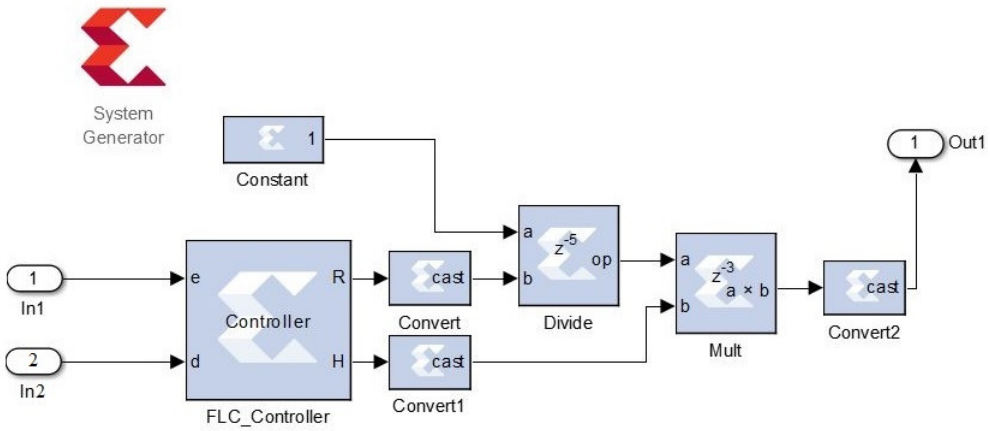

Figure 7. Proposed FLC controller using Xilinx® System Generator.

5.2. FL Controller Arithmetic Configuration

One particular form of controller for FPGA implementation is the subject of the practical inquiry. We discover an optimum architecture by reducing the quadratic errors on the signals that transmit data through the controller. establishing a fixed point (64, 32) is the best option out of the total of 32 bits, of which one bit is the sign bit and fifteen bits are the fractional part. The details will be discussed in the following section of this article. See Table 3.

Table 3. The proposed controller arithmetic structures' terminology.

Notation	FLC_64_32
Designation	Fuzzy Logic Controller
Type	Fixed-point
Precisions	(64 bits, 32)
Arithmetic type	Signed (2's comp)
Quantization	Round

6. Explanation of Results

Table 4 and 5 show the comparison between the standard vectors of the fuzzy controller and the vectors optimized by (GWO).

Table 4. Inputs Vector variables.

	Standard	GWO	Standard	GWO
i	X_i	X_i	Y_i	Y_i
-1	256	256	512	512
0	256	32	512	128
1	256	256	512	512

Table 5. Output Vector variable.

	Standard	GWO
I	Z_i	Z_i
-2	4	4
-1	4	4
0	4	0.125
1	4	4
2	4	4

The linguistic input and output variables are given in Table 6 and 7.

Table 6. Inputs Linguistic variables.

		Crisp Input Range	Crisp Input Range
i	Fuzzy Variable	e=Error $\in$ [-256,256]	d=dError $\in$ [-1024,1024]
-1	NE	$\in$ [. . . ,-250,0]	$\in$ [. . . ,-512,0]
0	ZE	$\in$ [-32,0,32]	$\in$ [-128,0,128]
1	PO	$\in$ [0,256,. . .]	$\in$ [0,512,. . .]

Table 7. Output Linguistic Variable.

	Crisp Output Range	
i	Fuzzy Variable	u=Control $\in$ [-16,16]
-2	BN	$\in$ [-16,-8,-4]
-1	NE	$\in$ [-8,-4,0]
0	ZE	$\in$ [-1/8,0,1/8]
1	PO	$\in$ [0,4,8]
2	BP	$\in$ [4,8,16]

The membership function for the input and output linguistic variables is represented graphically in Figure 8 and 9.

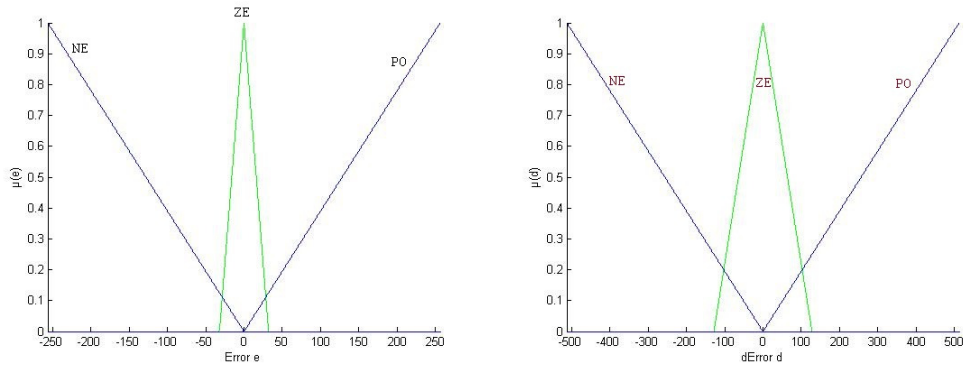

Figure 8. Optimized Membership function for inputs linguistic variable of FL Controller.

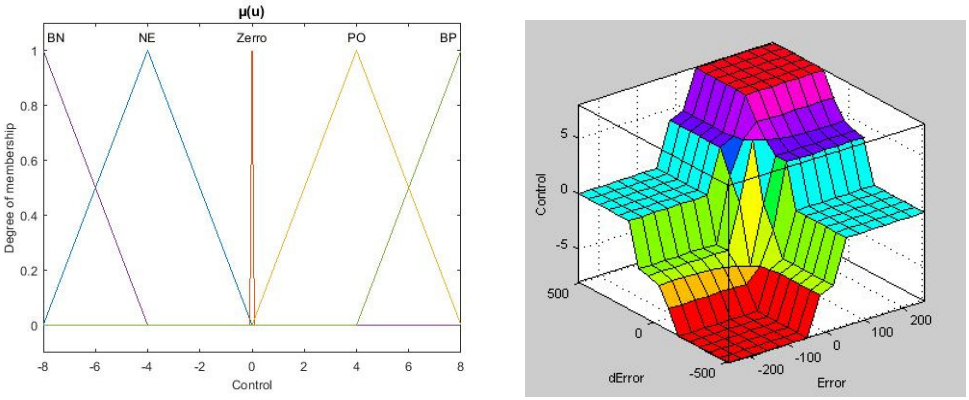

Figure 9. Optimized Membership function for output linguistic variable and control surface of FL controller.

The simplified and optimized algorithm for our fuzzy controller is given by the table 8:

Table 8. FLC Algorithmr

	Pseudo Code
1	*Initialize Vectors*
2	*a= [-256 -32 0 32 256];*
3	*b= [-512 -128 0 128 512];*
4	*c= [-8 -4 0 4 8];*
5	*A= [4 4 0.125 4 4];*
6	*E= [0 0 0];*
7	*D= [0 0 0];*
8	*U= [0 0 0; 0 0 0; 0 0 0];*
9	*R= [0 0 0; 0 0 0; 0 0 0];*
10	*H= [0 0 0; 0 0 0; 0 0 0];*
11	*----------*
12	*Calculate (E, D)*
13	*----------*
14	*k=0;*
15	***for*** *i = 1:3*
16	***for*** *j = 1:3*
17	*U(i,j)= min(E(i),D(j));*
18	*R(i,j)=A(j+k)*(2*U(i,j)-(U(i,j)^2));*
19	*H(i,j)=c(j+k)*R(i,j);*
20	***End***
21	*k=k+1;*
22	***End***
23	*----------*
24	*U_out=sum(H)/sum(R);*

6.1. Co-simulation Theorical Study of FL Controller

At low speeds, the signals are normal; but, as the speeds rise, the Windup phenomenon becomes more pronounced. This is explained by the presence of saturation in the Control loop. (Figure 10).

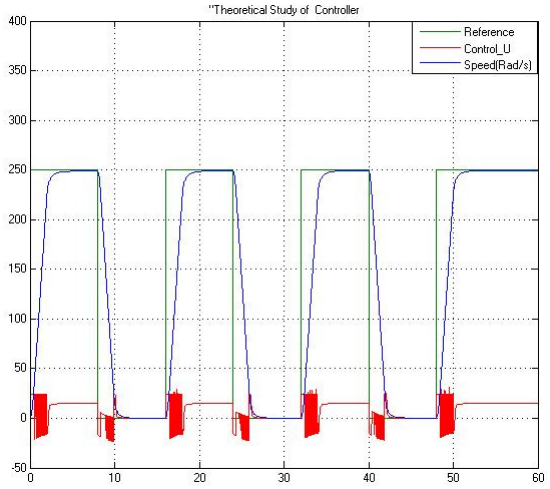

Figure 10. Simulink theorical study of FL Controller (speed and control curve Ref=250).

6.2. Co-simulation With Disturbance

The controller's behavior is influenced by sudden changes in speed due to the limits of the used speeds, which are disrupted by statistical errors connected to data propagation in the FPGA constructed in fixed point illustrated in Figure 11. The next graph depicts the behavior of the control u as a function of torque disturbance; we're talking about speed control here.

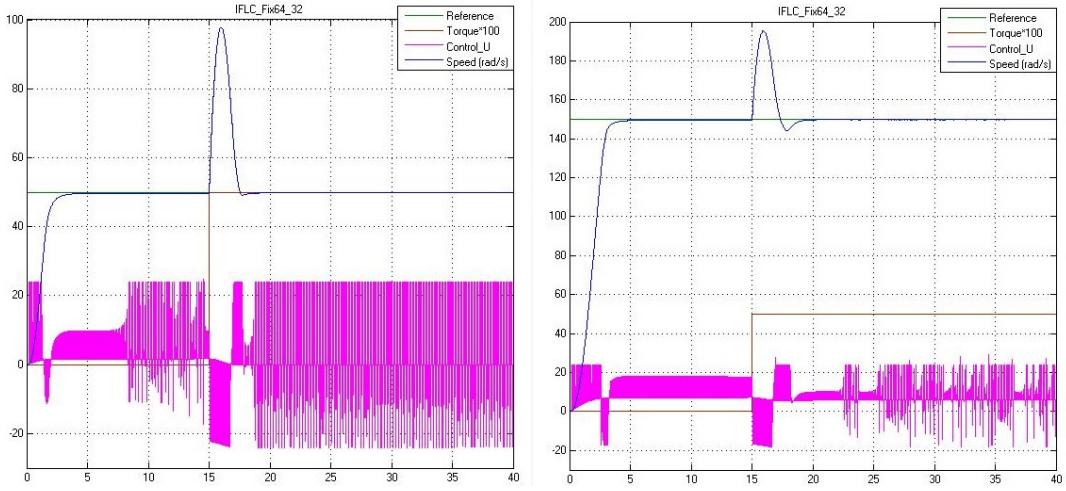

Figure 11. Speed and control results of the proposed IFLC with disturbance using.

6.3. Co-simulation Using Reference Variation

The limitations of the speeds utilized, which are disturbed by statistical mistakes related to data propagation in the FPGA built in fixed point, have an impact on the controller's behavior, (Figure 12).

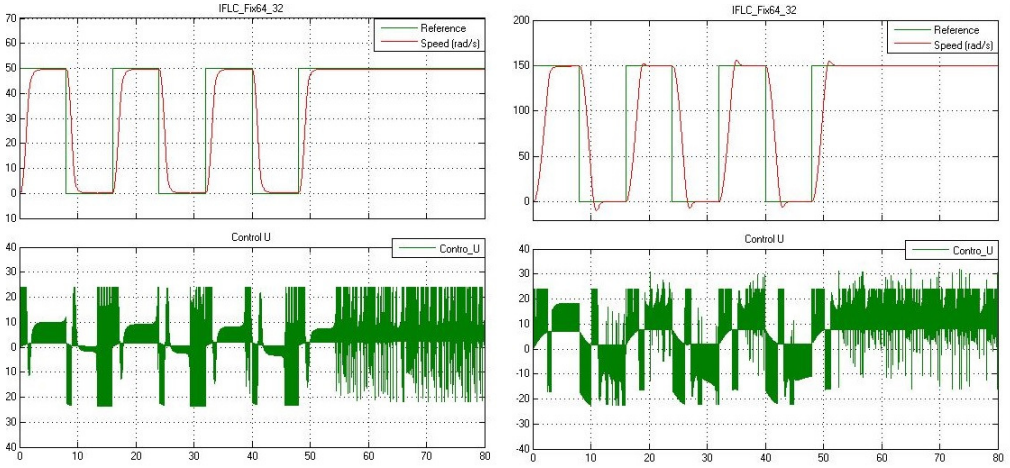

Figure 12. Speed and control results of the proposed IFLC without disturbance using.

6.4. FPGA hardware statistic

The Vivado platform handles the modification and optimization of the programs transmitted by System Generator from Simulink, after synthesis and analysis of the project, we acquire the following estimates Table 9 and Figure 13 show the programmable logic used in our realization.

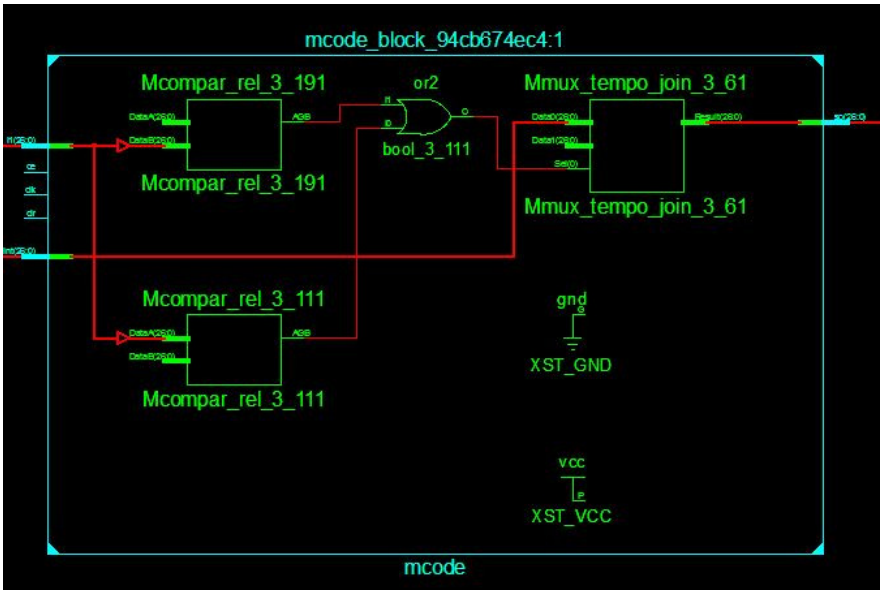

Figure 13. Top level RTL schematic of proposed hardware implementation (FLC).

7. Conclusion

Our hardware was built using VHDL and an FPGA. With the help of this endeavor, we were able to apply the majority of electromechanical systems theory and discover the centroid principle for the first time in real-world applications. The system's behavior has been modeled and specified. In actuality, we impose the moment of inertia

Table 9. FLC Design Utilization Summary.

	Estimated Values: FLC_64_32		
Logic utilization	Used	Available	Utilization
Number of Slice registers table	286	106400	0%
Number of Slice LUTs	8132	53200	15%
Number of fully used LUT.ff pairs	183	8235	2%
Number of bonded IOBs	129	200	64%
Number of BUFG/BUFGCTRLs	1	32	3%
Numbre of DSP48Es	149	220	67%

of the load while dispensing with the friction of the support. The emphasis was subsequently placed on fuzzy control with integral action, which let us outperform a conventional PID optimized using a Matlab program. To compare, a generic floating-point structure was employed. The system was followed from modeling through implementation of the many correctors that needed saturation control to lessen the impact of non-linearities, which made this assignment exciting. Additionally, by including its Anti windup in the tutorials and courses, it helped us comprehend the concepts of saturation. Results of the experiments demonstrate the dependability and efficiency of the developed controllers. The implementation, on the other hand, demonstrates that there is less consumption and a quicker execution time on a fixed-point structure (Fix 64,32). There is less variation in the execution time, a fixed sample time, less usage of the available resources, and only a brief setup period for the program and apparatus. Overall, the FPGA target offers a versatile method for developing complicated control algorithms, especially for highly dynamic processes. Future study will analyze the practical hidden components of this article, including the PWM and encoder signals, as well as the GPO of the FPGA used in the mechanism control. Not to mention the power supply voltage needed for the assembly to operate, the motor's dead time, and the incremental encoder's noises.

ACKNOWLEDGEMENT

This project has received support from the Directorate General of Scientific Research and Technological Development (DGRSDT) and the Laboratory of Electrical Engineering and Renewable Energy (LEER) in Algeria.

REFERENCES

1. M A Mohammed Eltoum, A Hussein, and MA Abido. Hybrid fuzzy fractional-order PID-based speed control for brushless DC motor. *Arabian Journal for Science and Engineering*, 46(10):9423–9435, 2021.
2. Hakan Acikgoz. Speed control of DC motor using interval type-2 fuzzy logic controller. *International Journal of Intelligent Systems and Applications in Engineering*, 6(3):197–202, 2018.
3. Qasem Al-Tashi, Helmi Md Rais, Said Jadid Abdulkadir, Seyedali Mirjalili, and Hitham Alhussian. A review of Grey Wolf Optimizer-based feature selection methods for classification. *Evolutionary Machine Learning Techniques: Algorithms and Applications*, pages 273–286, 2020.
4. Aiman Alabdo, Javier Pérez, Gabriel J Garcia, Jorge Pomares, and Fernando Torres. FPGA-based architecture for direct visual control robotic systems. *Mechatronics*, 39:204–216, 2016.
5. Belgacem Bekkar and Khale Ferkous. Design of Online Fuzzy Tuning LQR Controller Applied to Rotary Single Inverted Pendulum: Experimental Validation. *Arabian Journal for Science and Engineering*, 48(5):6957–6972, 2023.
6. Tomàs Pallejà Cabré, Albert Saiz Vela, Marcel Tresanchez Ribes, Javier Moreno Blanc, Jose Ribó Pablo, and Francisco Clariá Sancho. Didactic platform for DC motor speed and position control in Z-plane. *ISA transactions*, 118:116–132, 2021.
7. Gonggui Chen, Zhijun Li, Zhizhong Zhang, and Shuaiyong Li. An improved ACO algorithm optimized fuzzy PID controller for load frequency control in multi area interconnected power systems. *IEEE Access*, 8:6429–6447, 2019.
8. Lucian R da Silva, Rodolfo CC Flesch, and Julio E Normey-Rico. Analysis of anti-windup techniques in PID control of processes with measurement noise. *IFAC-PapersOnLine*, 51(4):948–953, 2018.
9. PB de Moura Oliveira, Hélio Freire, and EJ Solteiro Pires. Grey wolf optimization for PID controller design with prescribed robustness margins. *Soft Computing*, 20(11):4243–4255, 2016.
10. Serdar Ekinci, Davut Izci, and Baran Hekimoğlu. Optimal FOPID speed control of DC motor via opposition-based hybrid manta ray foraging optimization and simulated annealing algorithm. *Arabian Journal for Science and Engineering*, 46(2):1395–1409, 2021.

11. Nabil Farah, Md Hairul Nizam Talib, Zulkifilie Ibrahim, Qazwan Abdullah, Ömer Aydoğdu, Maaspaliza Azri, Jurifa Binti Mat Lazi, and Zainuddin Mat Isa. Investigation of the computational burden effects of self-tuning fuzzy logic speed controller of induction motor drives with different rules sizes. *Ieee Access*, 9:155443–155456, 2021.
12. A Fathima and G Vijayasree. Design of BLDC motor with torque ripple reduction using spider-based controller for both sensored and sensorless approach. *Arabian Journal for Science and Engineering*, 47(3):2965–2975, 2022.
13. Sikender Gul, Muhammad Faisal Siddiqui, and Naveed Ur Rehman. FPGA based real-time implementation of online EMD with fixed point architecture. *IEEE Access*, 7:176565–176577, 2019.
14. Ahmet Gundogdu, Resat Celikel, and Omur Aydogmus. Comparison of Si-ANN and extended Kalman filter-based sensorless speed controls of a DC motor. *Arabian Journal for Science and Engineering*, 46(2):1241–1256, 2021.
15. Handan Gürsoy and Mehmet Önder Efe. Control system implementation on an FPGA platform. *IFAC-PapersOnLine*, 49(25):425–430, 2016.
16. Mahammad A Hannan, Zamre ABD Ghani, Md Murshadul Hoque, Pin Jern Ker, Aini Hussain, and Azah Mohamed. Fuzzy logic inverter controller in photovoltaic applications: Issues and recommendations. *Ieee Access*, 7:24934–24955, 2019.
17. Afshan Ilyas, M Rizwan Khan, and Mohammad Ayyub. Original research article FPGA based real-time implementation of fuzzy logic controller for maximum power point tracking of solar photovoltaic system. *OPTIK*, 213, 2020.
18. Boutaina EL Kinany, Mohamed Alfidi, and Zakaria Chalh. Fuzzy Logic Control for Balancing a Two-Armed Inverted Pendulum. *Statistics, Optimization & Information Computing*, 11(1):136–142, 2023.
19. Jesus Lopez-Gomez, M Aurora D Vargas-Treviño, Sergio Vergara-Limon, Marciano Vargas-Treviño, Jaime Gutierrez-Gutierrez, AD Palomino-Merino, Fermín Martínez-Solis, and Olga Guadalupe Felix-Beltran. Influence of PWM torque control frequency in DC motors by means of an optimum design method. *IEEE Access*, 8:80691–80706, 2020.
20. A Lotfy, M Kaveh, MR Mosavi, and AR Rahmati. An enhanced fuzzy controller based on improved genetic algorithm for speed control of DC motors. *Analog Integrated Circuits and Signal Processing*, 105:141–155, 2020.
21. Ahmad A Masoud, Mohammad Abu-Ali, and Ali Al-Shaikhi. Experimental determination of an extended DC servo-motor state space model: an undergraduate experiment. *IEEE Access*, 8:4908–4923, 2019.
22. A Messai, I Abdellani, and A Mellit. FPGA-based real-time implementation of a digital reactivity-meter. *Progress in Nuclear Energy*, 150:104313, 2022.
23. Eric Monmasson, Lahoucine Idkhajine, Marcian N Cirstea, Imene Bahri, Alin Tisan, and Mohamed Wissem Naouar. FPGAs in industrial control applications. *IEEE Transactions on Industrial informatics*, 7(2):224–243, 2011.
24. Mourad Nachaoui, Abdeljalil Nachaoui, RY Shikhlinskaya, and Abdelali Elmoufidi. An improved hybrid defuzzification method for fuzzy controllers. *Statistics, Optimization & Information Computing*, 11(1):29–43, 2023.
25. Ákos Odry, Róbert Fullér, Imre J Rudas, and Péter Odry. Fuzzy control of self-balancing robots: A control laboratory project. *Computer Applications in Engineering Education*, 28(3):512–535, 2020.
26. Sasmita Padhy and Sidhartha Panda. Application of a simplified Grey Wolf optimization technique for adaptive fuzzy PID controller design for frequency regulation of a distributed power generation system. *Protection and Control of Modern Power Systems*, 6:1–16, 2021.
27. K Raja and S Ramathilagam. Washing machine using fuzzy logic controller to provide wash quality. *Soft Computing*, 25(15):9957–9965, 2021.
28. Gowthamraj Rajendran, Chockalingam Aravind Vaithilingam, Kanendra Naidu, Ahmad Adel Alsakati, Kameswara Satya Prakash Oruganti, and Mohd Faizal Fauzan. Dynamic voltage stability enhancement in electric vehicle battery charger using Particle Swarm Optimization. *IEEE Access*, 10:97767–97779, 2022.
29. Yesset Raziyev, Ramil Garifulin, Almas Shintemirov, and Ton Duc Do. Development of a power assist lifting device with a fuzzy PID speed regulator. *IEEE Access*, 7:30724–30731, 2019.
30. Juan Reyes-Reyes, Carlos-M Astorga-Zaragoza, Manuel Adam-Medina, and Gerardo-V Guerrero-Ramírez. Bounded neuro-control position regulation for a geared DC motor. *Engineering Applications of Artificial Intelligence*, 23(8):1398–1407, 2010.
31. Muhammed Arif ŞEN and Mete KALYONCU. Optimal tuning of PID controller using Grey Wolf Optimizer algorithm for quadruped robot. *Balkan Journal of Electrical and Computer Engineering*, 6(1):29–35, 2018.
32. André Sanches Fonseca Sobrinho and Francisco Granziera Junior. Type-1 fuzzy logic algorithm for low cost embedded systems. *Computers & Electrical Engineering*, 88:106861, 2020.
33. Albert Alexander Stonier, Srinivasan Murugesan, Ravi Samikannu, Vinoth Krishnamoorthy, Senthil Kumar Subburaj, Gnanavel Chinnaraj, and Geetha Mani. Fuzzy logic control for solar PV fed modular multilevel inverter towards marine water pumping applications. *IEEE Access*, 9:88524–88534, 2021.
34. Muhammad Talha, Furqan Asghar, Ali Rohan, Mohammed Rabah, and Sung Ho Kim. Fuzzy logic-based robust and autonomous safe landing for UAV quadcopter. *Arabian Journal for Science and Engineering*, 44:2627–2639, 2019.
35. Alper Kadir Tanyildizi, Oguz Yakut, B Taşar, and Ahmet Burak Tatar. Control of twin-double pendulum lower extremity exoskeleton system with fuzzy logic control method. *Neural Computing and Applications*, 33:8089–8103, 2021.
36. MA Vargas-Treviño, J Lopez-Gomez, S Vergara-Limon, A Palomino-Merino, R Torres-Reyes, and P Garcia-Ramirez. A mechatronic approach for ball screw drive system: modeling, control, and validation on an FPGA-based architecture. *The International Journal of Advanced Manufacturing Technology*, 104:2329–2346, 2019.
37. Ru Wang, Chao Tan, Jing Xu, Zhongbin Wang, Jingfei Jin, and Yiqiao Man. Pressure control for a hydraulic cylinder based on a self-tuning PID controller optimized by a hybrid optimization algorithm. *Algorithms*, 10(1):19, 2017.
38. Tingting Wang, Hongzhi Wang, Chuhang Wang, and Huangshui Hu. A novel PID controller for BLDCM speed control using dual fuzzy logic systems with HSA optimization. *Scientific Reports*, 12(1):11316, 2022.
39. Ayman Youssef, Mohammed El Telbany, and Abdelhalim Zekry. Reconfigurable generic FPGA implementation of fuzzy logic controller for MPPT of PV systems. *Renewable and Sustainable Energy Reviews*, 82:1313–1319, 2018.