https://doi.org/10.31436/iiumej.v26i1.3436

OPTIMIZED TYPE 2 FUZZY LOGIC CONTROL FOR LOW-SPEED VEHICLE PEDAL PRESSING AUTOMATION USING HYBRID SPIRAL SINE COSINE ALGORITHM

AZRUL AZIM ABDULLAH HASHIM[1], NOR MANIHA ABDUL GHANI[1*], SALMIAH AHMAD[2,3], MOHD RUZAINI HASHIM[4], NOOR ZIRWATUL AHLAM NAHARUDDIN[1], ADDIE IRAWAN[1]

[1]Faculty of Electrical and Electronics Engineering Technology, Universiti Malaysia Pahang Al-Sultan Abdullah, Pahang, Malaysia
[2]College of Engineering and Technology, University of Doha for Science and Technology, Qatar
[3]Kulliyyah of Engineering, International Islamic University Malaysia, Malaysia
[4]Faculty of Electrical Technology and Engineering, Universiti Teknikal Malaysia Melaka, Melaka, Malaysia

**Corresponding author: normaniha@umpsa.edu.my*

(Received: 3 September 2024; Accepted: 6 October 2024; Published online: 10 January 2025)

ABSTRACT: This paper describes the systematic design and experimental analysis of a Fuzzy Logic Controller (FLC) to govern vehicle speed for low-speed driving by adjusting an attached linear actuator that governs the vehicle's pedal. The research investigates two FLC approaches: the standard Type 1 FLC and the advanced Type 2 FLC, both optimized using the Hybrid Spiral Sine Cosine Algorithm (SSCA). The integrated system linking the actuator to the dynamics model of the vehicle shows improved ability in the manner in which control is done. Physical modeling and simulation were done in Simscape MATLAB, which provides an opportunity for modeling and visual description of the actuator system's relationship with the dynamics of the car. The results presented in this paper prove the fact that the analyzed Type 2 FLC optimized by the SSCA method performs better than the traditional Type 1 FLC in terms of the key metrics, with improvements of 32.4242% in overshoot, 0.364 seconds in settling time, and a reduction of 0.002009 in steady-state error at 2 km/h reference speed. This superior performance highlights the potential of the SSCA-optimized Type 2 FLC to automate pedal pressing for vehicle speed control, effectively replacing repetitive pedal actions and reducing driver fatigue, as this mechanism proves capable of controlling vehicle speed with high precision.

ABSTRAK: Kertas kerja ini menerangkan reka bentuk sistematik dan analisis eksperimen Pengawal Logik Kabur (FLC) untuk mengawal kelajuan kenderaan untuk pemanduan berkelajuan rendah dengan melaraskan penggerak linear yang dipasang yang mengawal pedal kenderaan. Penyelidikan ini menyiasat dua pendekatan FLC: FLC Jenis 1 standard dan FLC Jenis 2 lanjutan, kedua-duanya dioptimumkan menggunakan Algoritma Kosinus Sinus Lingkaran Hibrid (SSCA). Sistem bersepadu yang menghubungkan penggerak kepada model dinamik kenderaan menunjukkan keupayaan yang lebih baik dalam cara kawalan dilakukan. Pemodelan dan simulasi fizikal telah dilakukan dalam Simscape MATLAB di mana ia menyediakan peluang pemodelan dan penerangan visual tentang hubungan sistem penggerak dengan dinamik kereta. Keputusan yang dibentangkan dalam kertas kerja ini membuktikan fakta bahawa FLC Jenis 2 yang dianalisis yang dioptimumkan oleh kaedah SSCA menunjukkan prestasi yang lebih baik daripada FLC Jenis 1 tradisional dari segi metrik utama, dengan peningkatan sebanyak 32.4242% dalam overshoot, 0.364 saat dalam masa penyelesaian, dan pengurangan 0.002009 dalam ralat keadaan mantap pada kelajuan rujukan 2 km/j. Prestasi unggul ini menyerlahkan potensi Type 2 FLC yang dioptimumkan SSCA

https://doi.org/10.31436/iiumej.v26i1.3436

untuk mengautomasikan penekanan pedal untuk kawalan kelajuan kenderaan, menggantikan tindakan pedal berulang dengan berkesan dan mengurangkan keletihan pemandu, kerana mekanisme ini terbukti mampu mengawal kelajuan kenderaan dengan ketepatan tinggi.

KEYWORDS: *Fuzzy Logic Controller Type 2, Hybrid Spiral Sine Cosine, Automate Pedal Pressing, Vehicle Speed Control.*

1. INTRODUCTION

Extensive progress in vehicle control technology has resulted in the production of advanced systems, such as adaptive cruise control, designed to perform automated speed control and provide higher driving comfort on highways. ACC systems, especially those supporting stop-and-go functionality, are intended to decrease the driver's workload by independently responding to traffic and even stopping with the following movement on congested roads [1,2]. Unfortunately, these sophisticated systems are generally supported in high-end car models [3] and, hence, not available to middle-end car owners.

In markets like Malaysia, where middle-end vehicles are more prevalent, such disparities hinder the use of such technologies and their advantages. However, there are issues in frequent acceleration and deceleration cases, such as in stop-and-go traffic. Current systems may not support dynamic traffic changes and rapid responses, leading to fatigue among drivers and discomfort [4,5]. For the optimal benefit of these technologies, more advancement is required to enable improved availability and performance of automated vehicle control, especially in the urban environment where a high level of control is desirable.

Several studies have been conducted to improve and create new developments in vehicle control systems. Specifically, several solutions for enhancing the performance and reliability of technologies, including adaptive cruise control, have been developed. For example, Idzham et al. [6] explored a deep learning technique for diagnosing the consequences of repetitive brake pedal pressing and leg positions during traffic jams. The authors employed LGBM to analyze leg position and brake pressure data. Subsequently, the authors ascertained that this model based on AI could achieve about 80% forecast of vehicle speed. It can be employed to fine-tune the adaptive cruise control systems and improve vehicle accident prevention. On the other hand, Isa et al. [7] attempt to develop a linear electromechanical actuator to simulate brake and accelerator pedal pressing needs during traffic congestion to reduce a driver's fatigue leading to accidents. The contrast between the actuator force and the force produced by the human leg was deemed acceptable, and the response time was measured to be less than a second. In general, the advancements in such technologies are mainly focused on their reliability and response characteristics. This is important for driving scenarios such as stop-and-go, where technologies such as adaptive cruise control should work quickly and consistently to provide the ability to brake or accelerate. In addition, they strive to introduce cost-effective solutions that can be implemented on mid-range vehicles.

The pivotal part of any advanced automated pedal pressing for a vehicle speed control system is the controller, which is adjusted to the system to reach desired performance outcomes. The controller aims to process input data from sensors and change the actuator's performance to control the vehicle's pedals effectively and smoothly. In some situations, changing the car's pedals may be desirable, but they will not change due to road conditions. This is when the given control system is designed to provide efficient ways to control the functionality. Since driving is a rather complex activity and vehicles move in a dynamic manner, the traffic situation and the driving environment may change rapidly. Many control

systems today are based on fuzzy logic control as it has proven to manage more complex and nonlinear systems than traditional controllers, which require precise mathematical models and continuous adjustments such as PID controllers [8]. Fuzzy logic controllers operate on linguistic rules and approximate reasoning to adjust the system's behavior [9]. Thus, they can be based on the fact that real-time data can be approximate and variable. The implications for real-time data uncertainty and variability are embedded in fuzzy logic controllers' rules and rough estimates [10]. They can change and react to ever-changing conditions due to data from precise models' algorithms. This means a rather powerful solution in such a system as an automated pedal adjusting system since the behavior of the system is rather flexible and not as rigid as with standard algorithm-based systems. The main benefit of using fuzzy logic in control systems is better adaptability for dealing with imprecise data.

One of the significant difficulties of fine-tuning automatic driving systems is the proper selection of the controller's parameters [11]. It is also noted that properly tuning the control parameters poses a challenge no matter the control technique used, such as a fuzzy logic controller or other conventional methods [12]. In this context, researchers have applied optimization methods to solve this problem. Genetic Algorithms (GA) [13,14], Spiral Dynamic Algorithm (SDA) [15], Artificial Bee Colony (ABC) [16,17], Firefly Algorithm (FA) [18,19], Cuckoo Search (CS) [20,21], and Particle Swarm Optimization (PSO) [22,23] have been used to explore and optimize the parameter space quickly. These algorithms explore the parameter space systematically and efficiently, imitating natural processes or based on mathematical equations to perform optimizations and come up with the best setting. Applying all these optimization techniques can enhance controller performance, reliability, and flexibility, leading to generally enhanced, reliable, and high-quality automated systems.

Advanced optimization, on the other hand, is a technique of combining many optimizations in a bid to enhance the results while reducing the challenges that affect single optimization techniques. Hybrid optimization, therefore, is a process where two or more optimization techniques are used to take advantage of the strengths while avoiding some difficulties like getting stuck in local optima, where instead of getting the best solution, the algorithms end up getting the worst one. For example,[24]] one study integrated the spiral equation from SDA directly into the Bacterial Foraging Algorithm (BFA). It showed that it enhanced both the exploration and exploitation steps and the accuracy level. Another research work [25] employed the integration of SDA with GA to achieve greater accuracy using the mutation and crossover features of GA. These hybrid techniques have been observed to perform better in challenging optimization conditions.

Motivated by the latest developments and findings, this work aims to assess the hybrid optimization approaches for refining the parameters of advanced fuzzy logic type 2 controllers for self-controlled pedals. The study will particularly assess how SSCA can enhance the automated pedal pressing operation in low-traffic circumstances. This work aims to improve the fuzzy logic controller's performance by applying hybrid optimization to solve real-world and time-varying driving conditions more efficiently. The focus will be to improve the control system to ensure the driver is comfortable in traffic-sensitive conditions and address the practical issues identified in the low-traffic context.

This research introduces a novel integration of the Hybrid Spiral Sine Cosine Algorithm (SSCA) with a Type-2 Fuzzy Logic Controller (FLC) to optimize automated pedal pressing systems for low-speed driving conditions in a simulation environment. Unlike previous studies that focus on single optimization techniques or conventional controllers, this study utilizes the strengths of SSCA to improve the adaptability and precision of Type-2 FLC in handling dynamic and nonlinear traffic scenarios. This work enhances the performance and reliability

https://doi.org/10.31436/iiumej.v26i1.3436

of automated pedal systems in simulated environments by addressing challenges such as parameter tuning, imprecise data, and real-time response. Additionally, it contributes to the field of vehicle control by demonstrating how hybrid optimization techniques can refine controller parameters to achieve superior results under variable and time-sensitive driving conditions, offering insights into potential future applications for Advanced Driver Assistance Systems (ADAS).

2. METHODOLOGY

2.1. Design of Pedal Subsystem Prototype

The pedal subsystem was designed based on the brake pedal system commonly found in the Proton Saga, a popular sedan model in Malaysia. Figure 1 shows the process of measuring the available space in the car's footwell, a crucial step to ensure that the subsystem design fits the spatial limitations typical of sedan vehicles. By capturing accurate dimensions from the Proton Saga's pedal system, this design approach can be effectively applied to similar sedan models.

Figure 1. Measuring available space

After the measurements, key components were selected and integrated into the design. Figure 2 and Figure 3 illustrate the linear actuator model L16-P, chosen for its compact size and suitable force output for controlling the pedal. The Wilwood swing mount brake pedal, shown in Figure 4 and Figure 5, was selected to simulate realistic pedal movement. The complete pedal subsystem was then modeled in SolidWorks Figure 6, with each component positioned and scaled to match the measured dimensions.

Figure 2. Linear actuator L16-P

https://doi.org/10.31436/iiumej.v26i1.3436

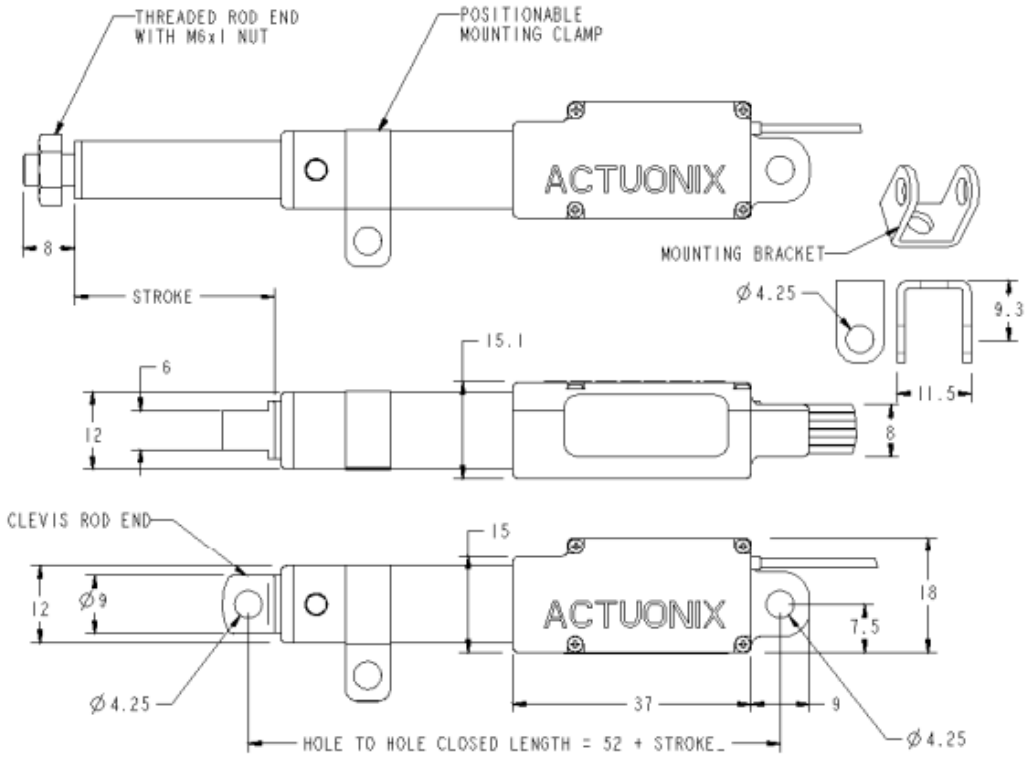


Figure 3. Dimension of linear actuator L-16P

Figure 4. Wilwood swing mount brake pedal

https://doi.org/10.31436/iiumej.v26i1.3436

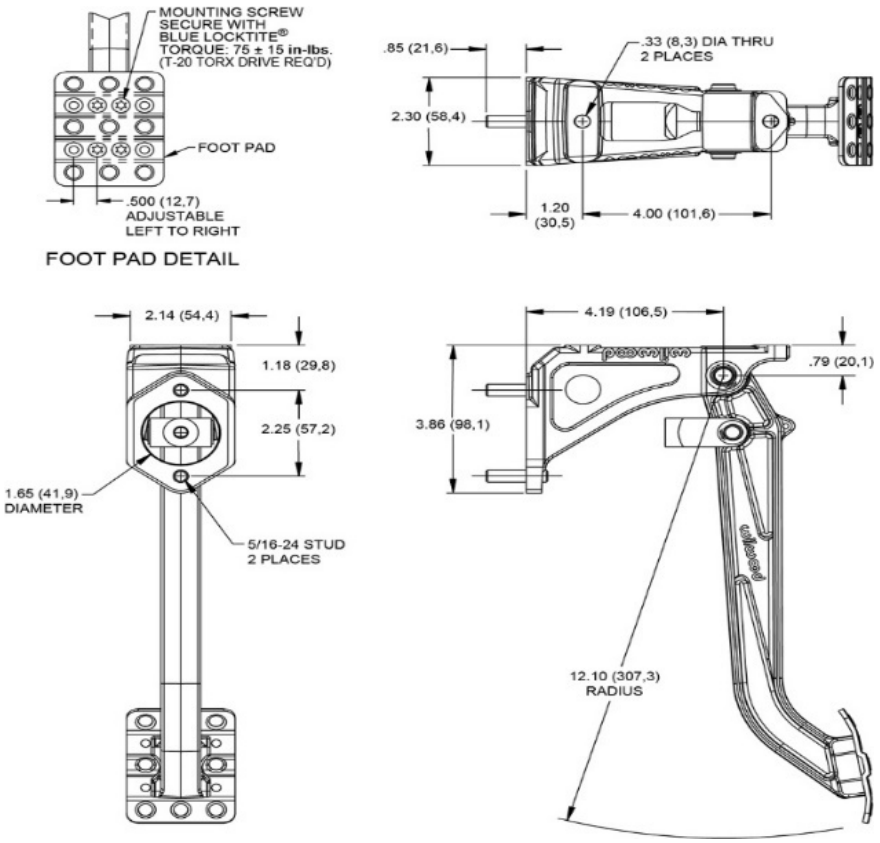


Figure 5. Dimension of Wilwood swing mount brake pedal

This CAD model is an efficient alternative to complex mathematical modeling before implementation on real hardware, as it requires a shorter time to derive accurate equations from complex variations [26,27]. Additionally, the 3D CAD model in SolidWorks brings the advantage of closely replicating the real system by incorporating essential factors such as nonlinearity, mass, gravity, and friction within the software environment. This enables more realistic simulations throughout the study, enhancing the model's practical applicability to real-world conditions.

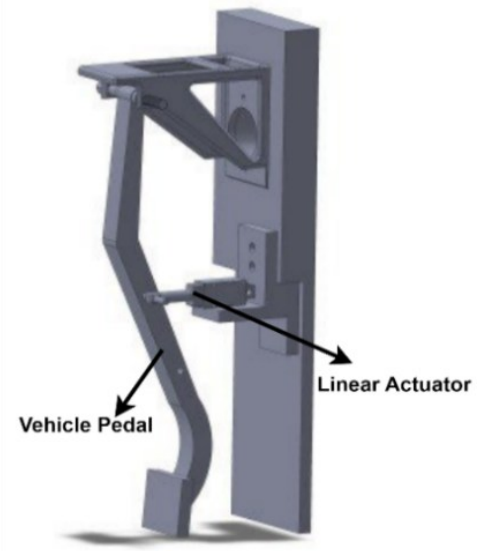


Figure 6. Automate Pedal Pressing Prototype design in 3D SolidWorks

https://doi.org/10.31436/iiumej.v26i1.3436

2.2. Physical Component Modeling of Linear Electromechanical Actuator in Simscape.

Figure 7, which employs Simscape component models for its systems representation, illustrates the physical components of this linear actuator. The lead screw, DC motor and gearbox transmission, force sensor, and voltage source are all components of this linear actuator. All of these components are essential for the actuation of responses, including the pressing and releasing of the accelerator. Thus, Table 2 presents the parameters of the chosen linear electromechanical actuator, with values that will be utilized in the modeling represented in Figure 2.

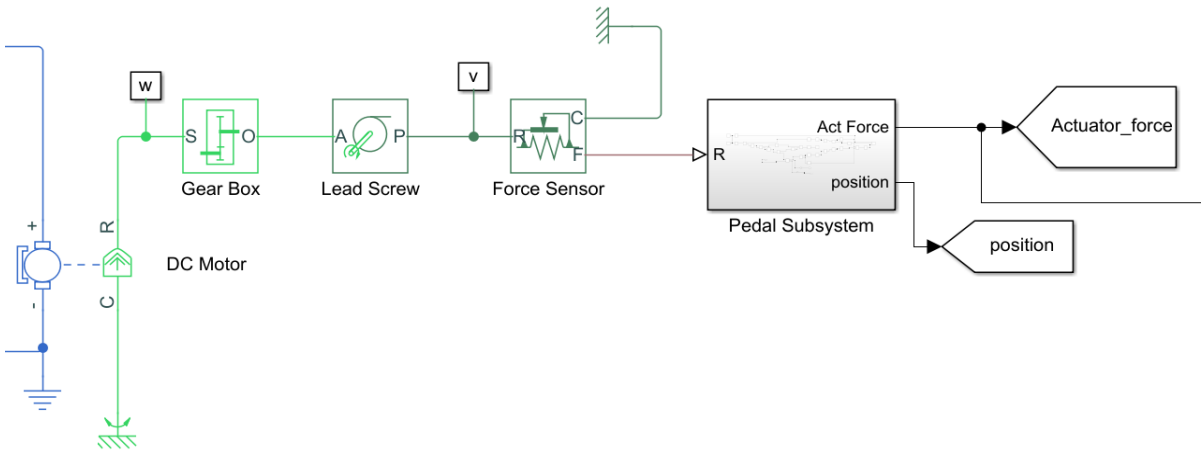


Figure 7. Modeling of Linear Actuator component in Simscape Matlab

In Figure 7, the system begins with the DC motor, which converts electrical energy into mechanical energy. The motor generates torque $T_{motor}(S)$ based on the input voltage $V(s)$, as described by the relationship in Eq. (1).

$$\frac{T_{motor}(S)}{V(s)} = \frac{K_{motor}}{sL+R} \tag{1}$$

where K_{motor} is the motor constant, L is the winding inductance, and R is the winding resistance.

The gearbox then reduces or increases the angular speed and torque output of the motor. The angular speed of the shaft $W_{shaft}(S)$ is related to the input voltage $V(s)$ as shown in Eq. (2).

$$\frac{W_{shaft}(S)}{V(S)} = \frac{1/b}{s(J/b)+1} \tag{2}$$

where b and J are shaft's rotational friction constant and moment of inertia, respectively.

Then the lead screw converts rotational motion into linear motion. The relationship between the gear ratio G, input torque τ_{in}, and output torque τ_o is given by Eq. (3).

$$G = \frac{\tau_o}{\tau_{in}} \tag{3}$$

The linear force, F, generated by the rotating lead screw can be calculated using Eq. (4)

$$F = G\,\frac{2\pi}{l} \tag{4}$$

where l is the length of one pitch on the leadscrew. As for the linear position X_{linear} of the leadscrew can be expressed by the following Eq. (5).

https://doi.org/10.31436/iiumej.v26i1.3436

$$X_{linear} = \theta_{linear} \frac{l}{2\pi} \quad (5)$$

Thus, Table 1 presents the parameters for the chosen linear electromechanical actuator, providing essential values for the system's performance.

Table 1. Parameter for the Chosen Linear Electromechanical Actuator

No	Parameter	Value
1	Torque Constant	0.4 N.m
2	Winding Resistance	1 Ω
3	Winding Inductance	0.05 H
4	Screw lead pitch	0.008 m
5	Gearbox ratio	2
6	Lead screw efficiency	40 %
7	Shaft moment of inertia	0.01 Kg.m^2
8	Rotational friction coefficient	0.1 N.m. s

2.3. Type-2 Fuzzy Logic Controller

Type 2 Fuzzy Logic Controllers (FLCs) offer a distinct advantage over Type 1 FLCs, particularly in handling uncertainty and variability in complex systems. While Type 1 FLCs use crisp membership functions, Type 2 FLCs employ fuzzy sets with fuzzy boundaries such as lower scale and lower lag, allowing them to manage the uncertainties inherent in real-world applications better. This capability improves performance, especially in systems with highly uncertain or dynamic input-output relationships.

Lower scale and lower lag are critical features of Type 2 FLCs. The lower scale refers to the finer granularity in the fuzzy membership functions, enabling the controller to respond more precisely to small variations in input data. This precision is particularly beneficial in systems requiring high accuracy and smooth control, such as in-vehicle speed control, where minor deviations must be corrected quickly.

Lower lag in a Type 2 FLC reduces the delay between the input signal and the system's response, leading to a more responsive control system. This is especially important in dynamic environments where the system must adapt rapidly to changes, such as sudden decelerations in a vehicle. Selecting lower scale and lower lag values typically involves tuning the fuzzy membership functions and rules to optimize the controller's performance. By including these values in the Type 2 FLC, engineers can test and validate whether the controller provides improved response time, stability, and accuracy compared to a Type 1 FLC. This comparison often reveals that Type 2 FLCs offer enhanced robustness and flexibility, making them more suitable for complex and uncertain control tasks.

Step 1: Specify the number of inputs and outputs and the number of Fuzzy sets. As shown in Figure 8, there are 2 inputs: Error of vehicle speed and Change of error of vehicle speed, 1 output: Voltage, and 5 number of fuzzy set: Negative Big (NB) and Negative Small (NK), Zero (Z), Positive Small (PK), and Positive Big (PB) are the five fuzzy sets with which each such input value is associated.

Step 2: Selecting the type of membership functions, such as Gaussian, trapezoidal, and others, for imprecise sets that include inputs (error and change of error) and outputs (voltage supply) is essential. Furthermore, applying lower scale and lower lag values, randomly specified within the range of 0 to 1, is crucial for Type 2 FLC.

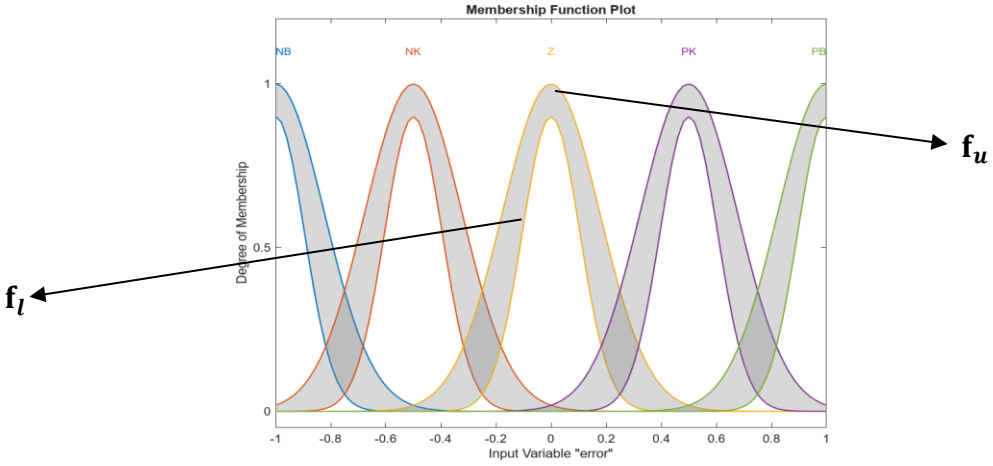
https://doi.org/10.31436/iiumej.v26i1.3436

This approach enhances the quality of uncertainty handling by providing an ideal transition between membership categories.

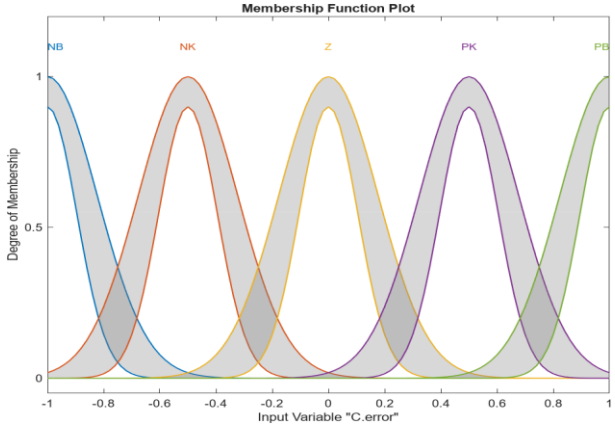
Step 3: To determine the outputs, 25 fuzzy rules are implemented by combining the five fuzzy sets for each input. The rule foundation provided corresponds to the pedal control subsystem's specifications. It consists of a 5x5 matrix in which each cell corresponds to a specific pair of input fuzzy values and describes the amount of voltage that should be applied to the adjusting linear actuator.

Step 4: Transform the fuzzy output data into a precise voltage value. This technique transforms the fuzzy results from the rule application into a precise and accurate control value for the actuator, ensuring precise and appropriate pedal control. The defuzzification process, supported by the lowered scale and lower lag, effectively controls dynamic changes in the system.

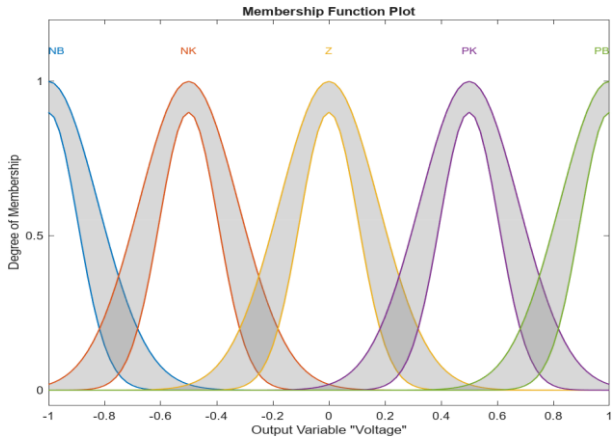
In fuzzy logic controller (FLC) design, normalized values, typically ranging from -1 to 1, simplify the design process and ensure consistency across varying system conditions. This normalization allows for easier handling of inputs and outputs, making defining membership functions and process control actions straightforward without being influenced by the absolute scales of the parameters involved. However, the range of input and output values needs to be scaled according to the system's specific characteristics, as it may not naturally produce values within this normalized range.



(a)



(b)



(c)

Figure 8. Design of Membership Function: (a). Error Input, (b). Change of error input and (c). Output (Voltage)

Table 2. Fuzzy Rules Table

Error	**Change of Error**				
	NB	**NK**	**Z**	**PK**	**PB**
NB	NB	NB	NB	NK	Z
NK	NB	NB	NK	Z	PK
Z	NB	NK	Z	PK	PB
PK	NK	Z	PK	PB	PB
PB	Z	PK	PB	PB	PB

The block diagram in Figure 3 (a) illustrates the structure of a Type-2 Fuzzy Logic Controller (FLC) specifically designed to manage uncertainties through interval type-2 fuzzy sets. The system receives two input signals: the error (e) and the change of error (Δe), representing the difference between the desired and actual values. These inputs are fuzzified using interval type-2 membership functions, which consist of an upper membership function (UMF) and a lower membership function (LMF). This dual-layer approach forms the foundation of interval type-2 fuzzy sets, allowing for the representation of uncertainty through a footprint of uncertainty (FOU). During the fuzzification process, each input is assigned a degree of membership across both the UMF and LMF, represented by the functions.

$$f_u(x) \text{ and } f_l(x) \tag{6}$$

where $f_u(x)$ and $f_l(x)$ denote the upper and lower membership values, respectively, for a given input x.

Once fuzzified, the inputs are processed through a rule evaluation (inference) stage. Each input's membership values interact within a set of fuzzy rules to produce outputs. Each rule acts as a conditional statement linking the inputs to the control output, with the strength of each rule defined by a range of values due to the FOU. The firing strengths for each rule, represented as intervals, are calculated by applying the minimum operator to both UMF and LMF values. For a given rule i, the upper and lower firing strengths $w_{u,i}$ and $w_{l,i}$

$$w_{u,i} = min(f_u(e)\ , f_u(\Delta e)) \tag{7}$$

$$w_{l,i} = min(f_l(e)\ , f_l(\Delta e)) \tag{8}$$

https://doi.org/10.31436/iiumej.v26i1.3436

Here, $f_u(e)$ and $f_u(\Delta e)$ represent the upper membership values for the error and change of error while $f_l(e)$ and $f_l(\Delta e)$ denote the lower membership values.

The outputs from each rule are then aggregated to form an output fuzzy set, resulting in an interval type-2 fuzzy set. This aggregated output undergoes type reduction to yield an interval type-1 fuzzy set. Type reduction determines the limits of the resulting fuzzy set, typically computed using the left and right centroids, denoted as c_L and c_R. These are calculated using the following equations.

$$c_L = \frac{\sum_{i=1}^{L} x_i \cdot f_{u,i} + \sum_{j=L+1}^{N} x_J \cdot f_{l,j}}{\sum_{i=1}^{L} f_{u,i} + \sum_{j=L+1}^{N} f_{l,j}} \tag{9}$$

$$c_R = \frac{\sum_{i=1}^{R} x_i \cdot f_{l,i} + \sum_{j=R+1}^{N} x_J \cdot f_{u,j}}{\sum_{i=1}^{R} f_{l,i} + \sum_{j=R+1}^{N} f_{u,j}} \tag{10}$$

In these expressions, x_i represents the sampled values over the output universe of discourse, while $f_{u,i}$ and $f_{l,i}$ correspond to the UMF and LMF values within the output set.

Finally, the defuzzification process produces a crisp output by averaging the lower and upper centroid values, c_L and c_R, to determine a single control action. The resulting control output y is expressed as:

$$y = \frac{c_L + c_R}{2} \tag{11}$$

This output y is then connected to the integrator block and the gain K3 in the control system. The integration process allows the system to accumulate the effect of the control action over time, effectively enhancing the system's response. Given that the output can range from [-1,1], K3 amplifies the signal as needed until the system achieves the desired output. This configuration ensures that force control is appropriately managed in response to input conditions.

2.4. Spiral Sine Cosine Tuning Approach.

Problems such as getting stuck in local optima have always posed a great challenge for optimization algorithms, especially when numerous obstacles exist in the search space. This difficulty arises when one optimizes only for a specific region in the solution space rather than the entire space, leading the algorithm to take an optimal measure in that region but not the best one. In other words, the algorithm may converge to a 'solution' that does not prevent it from searching for a more optimal solution across the entire search space, a globally optimal solution for the problem.

The SSCA algorithm is another modern meta-heuristic optimization algorithm that addresses the issue of premature convergence. While the SCA is responsible for exploring the search space, the SCA overcomes the exploration capabilities of the Spiral Dynamic Algorithm to prevent the algorithm from falling into suboptimal solutions. The SCA component encourages a large exploration of the solution space and frees it from the local optimum by consistently updating the direction and amplitude of the search. In addition, the SDA part of the algorithm carries the burden of the detailed adjustment of the search in the promising regions, making the algorithm fast in converging to the best solutions.

https://doi.org/10.31436/iiumej.v26i1.3436

The SSCA starts by creating a population of agents in the search space, each representing a solution. The first step of the algorithm used, called "spiral movement," can be described in Eq. (12)

$$X(i)'' = (rz \times Rz \times X(i)') - \Big(\big((rz \times Rz) - I\big) \times xStar\Big) \tag{12}$$

where $xStar$ represents the center of the spiral, X(i)″the updated position of the searching agent, rz is a spiral radius, RZ is the rotational matrix, I is $n \times n$ identity matrix, and X(i)′ is the current position of the searching agent. This equation ensures that agents move in a spiral trajectory toward the best solution, promoting global exploration.

After updating the positions using the spiral equation, SSCA performs another step to refine the updated position of the agents using sine and cosine functions, as shown in Eq. (13) and Eq. (14).

$$X(i)''' = X(i)'' + r1 + sin(r2) + |r3 \times (P(i) + X(i)'')| \tag{13}$$

$$X(i) = X(i)''' + r1 + cos(r2) + |r3 \times (P(i) + X(i)''')| \tag{14}$$

where X(i) is the updated position of the current agent, r1 is the adaptive step size that decreases over iteration, r2 is a random angle between 0 and 2π, r3 random factor, and P(i) is the best-known position. The r1 is defined as Eq. (15).

$$r1(t) = a + t\frac{a}{T} \tag{15}$$

where a is a predefined constant, t represents the current iteration index, and T denotes the maximum number of iterations specified for the algorithm. The equation for $r1$ is structured to decrease as the iteration count increases. Equation $r2$ is formulated as follows in Eq. (16)

$$r2 = 2 \times \pi \times rand \tag{16}$$

where $rand$ is a random number defined between [0,1]. It depicts the algorithm's random behavior in deciding the agent's direction. Additionally, the parameter $r3$ is a separate random value applied to each agent, adding individual variability to each agent's behavior.

These equations ensure the local searches around the best-known positions, allowing the algorithm exploitative capability. Through these procedures, the cost function values are updated as the algorithm converges towards the optimal solution of the target problem, allowing SSCA to be suitable for solving complex optimization problems.

Figure 9 shows the overall block diagram of the automated pedal pressing for the vehicle speed control system. This system also incorporates Fuzzy Logic Control to control the vehicle pedal, where the FLC has been optimized by applying SSCA to fine-tune the control model. Figure 10 illustrates the design of this system in the Simulink MATLAB environment. This design mimics the block diagram of Figure 9, where the control logic, the sensors' input, and the actuators' output are performed in simulation.

Since the normalized values in the Type 2 FLC design range from (-1, 1) for both input and output, there is a need to scale these values to adapt to the system's requirements. The parameters K1, K2, and K3 are employed for this purpose. K1 adjusts the proportional response to the current error, while K2 accumulates the error over time to eliminate steady-state discrepancies. Additionally, K3 is associated with the integral aspect, facilitating the accumulation of past errors to ensure that any lingering steady-state error is effectively corrected. Optimization techniques are then utilized to fine-tune these parameters, ensuring they align with the system's specific dynamics and performance needs. This approach allows

https://doi.org/10.31436/iiumej.v26i1.3436

the FLC to adapt its control actions effectively, achieving robust and precise performance tailored to the unique operational conditions of the application.

Thus, Table 4 depicts the important parameters that should be defined within the Spiral Sine Cosine Algorithm (SSCA) for optimizing the values of K1, K2, and K3. These parameters play a critical role in ensuring that the Type 2 FLC can operate effectively within the normalized range while adapting to the system's specific requirements. By fine-tuning these parameters, the SSCA facilitates improved performance and stability of the FLC, enabling enhanced control actions that are responsive to varying operational conditions.

Figure 11 presents the flowchart of the SSCA implementation. The SSCA flowchart begins with the agent population being initialized at various locations in the solution space. Following initialization, the algorithm updates the agents' positions in a loop, where the direction of movement is determined by the spiral movement combined with sine-cosine formulas. During each iteration, the agents explore the problem space and leverage the most promising points to refine the solutions. This iterative process continues until a termination condition is met, at which point the best solution is identified.

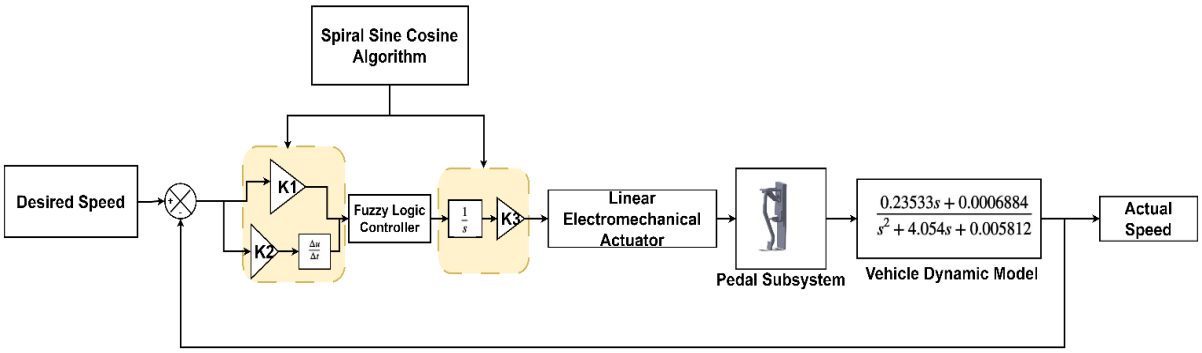


Figure 9. Block diagram structure of the Automated Pedal Pressing for Vehicle Speed Control

https://doi.org/10.31436/iiumej.v26i1.3436

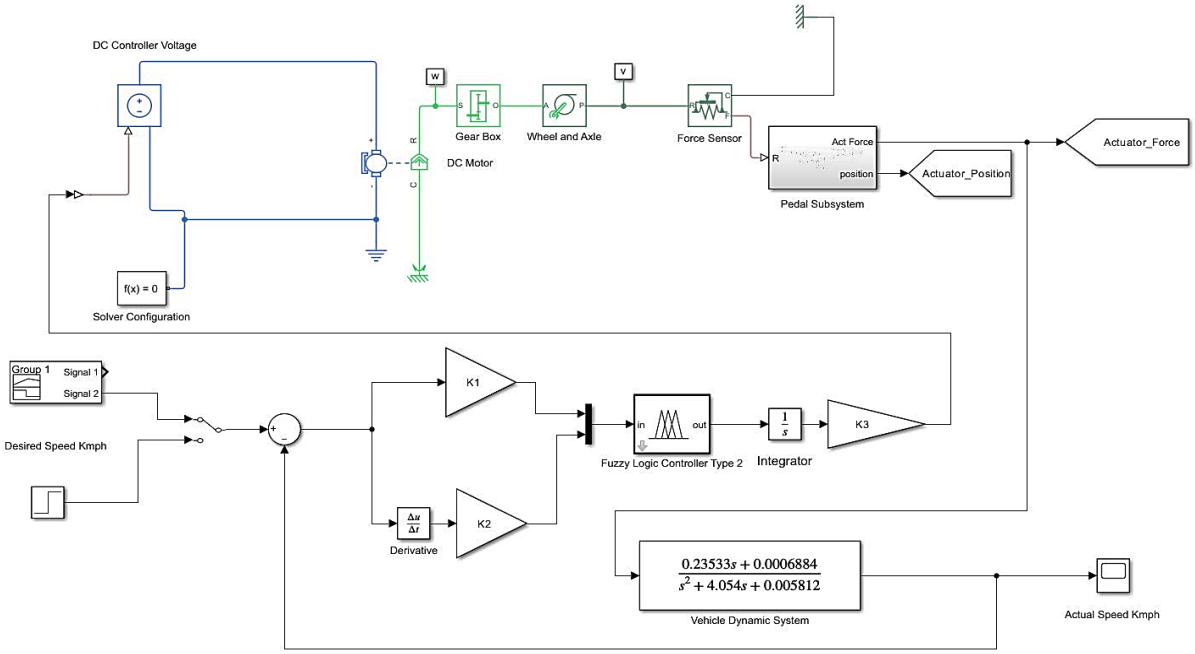


Figure 10. Detailed Simscape Model of the Automated Pedal Pressing for Vehicle Speed Control in MATLAB Simulink

Table 3. Parameters for the SSCA

Parameter	**Value**	
Number Iteration	30	
Number of Agents	10	
Dimensions	3	
Boundaries	Min	Max
K1	0	1
K2	0	1
K3	0	30
Spiral Parameter		
rz	0.96	
θ	π/2	
RZ	1	

https://doi.org/10.31436/iiumej.v26i1.3436

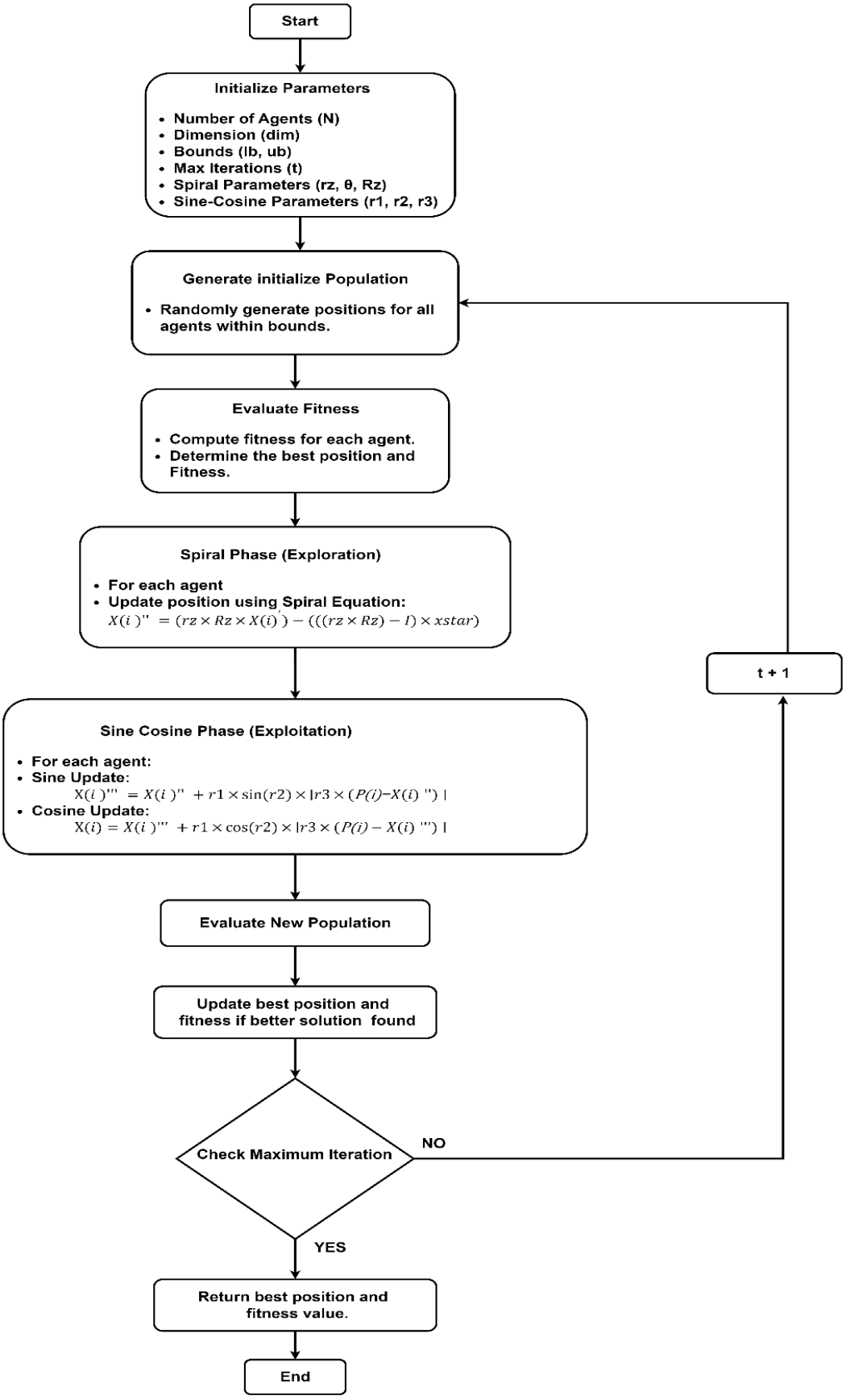


Figure 11. Flowchart of implementation Spiral Sine Cosine Algorithm

https://doi.org/10.31436/iiumej.v26i1.3436

3. RESULTS AND DISCUSSION

The following section provides the outcomes of the simulations on the automated pedal pressing for vehicle speed control using a linear actuator intended to control the vehicle's speed during low-speed driving situations. The system replaces leg activities by pressing and releasing the automobile's pedals. The research mainly focuses on testing a Type-2 FLC, tuned by the SSCA, to bring the actuator to stability and maintain the vehicle's speed of the vehicle constant. To demonstrate the efficiency of the Type-2 FLC, it is compared with a Type-1 FLC. The comparison of the Type-2 FLC design optimized using the SSCA with a conventional Type-1 FLC controller allows us to observe the benefits in the overshoot, settling time, rise time, and steady-state error that will give stability and accuracy of speed control, especially in low-speed driving.

3.1. Convergence Analysis of Spiral Sine Cosine Algorithm

The convergence plot of the spiral sine cosine algorithm with two fuzzy logic controllers, Type-2 and Type-1, was performed as shown. The parameters K1, K2, and K3 were optimized in the process. The tuning was done through 30 iterations with a Root Mean Square Error as a fitness function.

Figure 12 presents the convergence behavior of the Type-2 FLC, showing the results from 10 cycles of the SSCA optimization algorithm. As indicated in the figure, the best performance was achieved during the 8th cycle, with the RMSE initially stabilizing at 0.945 for the first 5 cycles. From cycle 6 onward, a significant improvement was observed, with the RMSE decreasing to 0.903, and this value was maintained consistently throughout cycle 30. This indicates that the SSCA effectively converged to an optimal solution, improving the control parameters of the Type-2 Fuzzy Logic Controller.

Conversely, the Type-1 FLC did not exhibit the same convergence properties. Throughout all 30 iterations, the RMSE value remained constant at 0.905, showing no further reduction. This lack of convergence suggests that the SSCA was less effective in optimizing the Type-1 FLC parameters, as it failed to achieve significant improvements throughout the iterations.

https://doi.org/10.31436/iiumej.v26i1.3436

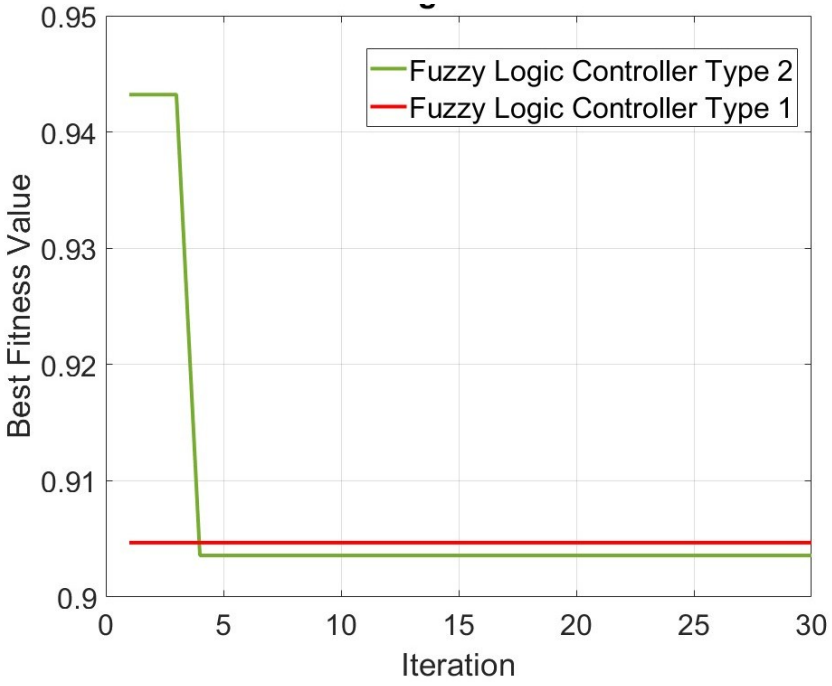


Figure 12. Comparison of Convergence Curve for both Type1 and Type 2 FLC

3.2. Performance Analysis with Different Speed

Table 5 shows some of the parameters tuned by the SSCA for both Type-1 and Type-2 Fuzzy Logic Controllers (FLC). Another point that distinguishes the Type-2 FLC is the existence of extra parameters, such as the lower lag and the lower scale value, which are randomly specified within the range of 0 to 1. In contrast, the Type-1 FLC does not include these parameters, contributing to the Type-2 FLC's improved performance and stability. Additionally, for both FLC types, the three constants designated as K1, K2, and K3, as shown in Figure 9, were adjusted by SSCA over defined boundaries, as depicted in Table 4.

Figure 13 illustrates the system's response in three key areas: vehicle speed (km/h), force applied by the actuator (N), and actuator position (mm). A comparative analysis of the two FLC types shows that FLC Type-1, even when tuned by SSCA, exhibits a certain level of oscillation with regard to system response. This oscillation is particularly evident when comparing the speed and actuator force graphs.

The presence of oscillations in the FLC Type-1 controller is significant due to its higher overshoot, which renders the speed control system unstable. The actuator force also fluctuates and follows a cycle, indicating consistent control. The Type-2 FLC, with parameters tuned by SSCA, offers a less oscillatory response compared to its basic counterpart.

A comparison between FLC Type 1 and FLC Type 2 has revealed improved performance in terms of the following indices, achieved using the Spiral Sine Cosine Algorithm (SSCA). Table 5 shows that at a speed of 2 Kmph, FLC Type 1 had an overshoot of 36.0198%. Meanwhile, (FLC Type 2) reduced the overshoot by 3.5956 %, improving about 32.4242%. Concerning rise time, FLC Type 1 was observed to be better than FLC Type 2 at certain speeds, exhibiting a shorter rise time. For instance, at 2 Kmph, FLC Type 1 reached a rise time of 0.2093 seconds compared to 0. 5573 seconds for FLC Type 2. However, one obvious disadvantage of the faster rise time in FLC Type 1 is that it exhibits a larger overshoot, causing the pedal and the actuator to oscillate and resulting in poor performance of vehicle speed control. The Simscape visual representation of the automated pedal-pressing vehicle speed

https://doi.org/10.31436/iiumej.v26i1.3436

control is depicted in Figure 14The figure illustrates how the actuator responds to various vehicle speeds. If the desired speed is 5 km/h, the actuator retracts, and the pedal attached to it extends, thereby releasing the pedal vehicle and increasing the speed.

In terms of settling time, FLC Type 2 was slightly better on average, especially at 2 Kmph, where settling time was reduced by 0.364 seconds compared with FLC Type 1. As such, it allows a steady state to be achieved faster and makes the overall structure more stable.

As for the steady-state error, FLC Type 2 had lower and constant results, providing a more accurate control. At 2 Kmph, the steady-state error was -0.0018, compared to -0.0038 for FLC Type 1, reflecting an improvement of 0.002009.

Table 4. Parameter Tune by Spiral Sine Cosine Algorithm for both Type FLC

Controller	Parameter Tune By SSCA			Randomly Specified	
	K1	K2	K3	Lower Scale	Lower Lag
FLC Type 1	0.1188	0.0099	21.9156	-	-
FLC Type 2	0.0678	0.0100	30	1	0.2

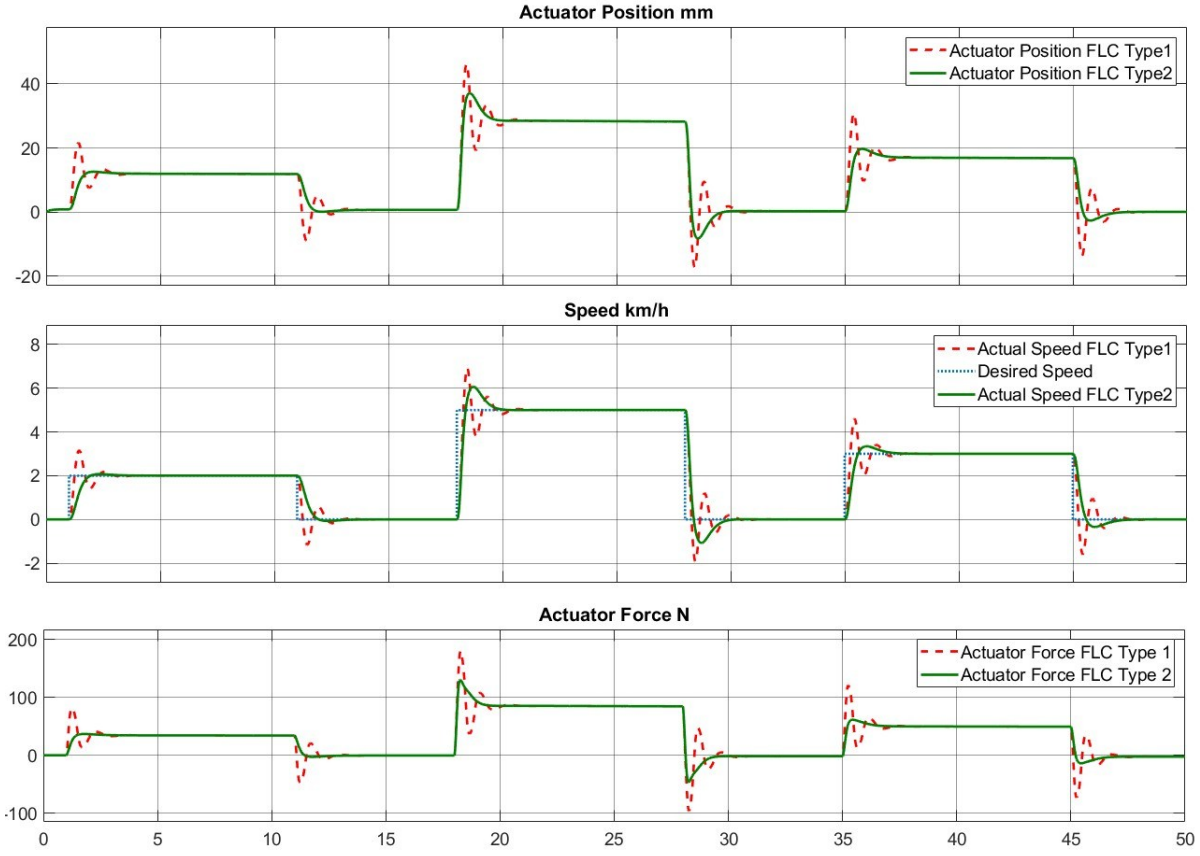
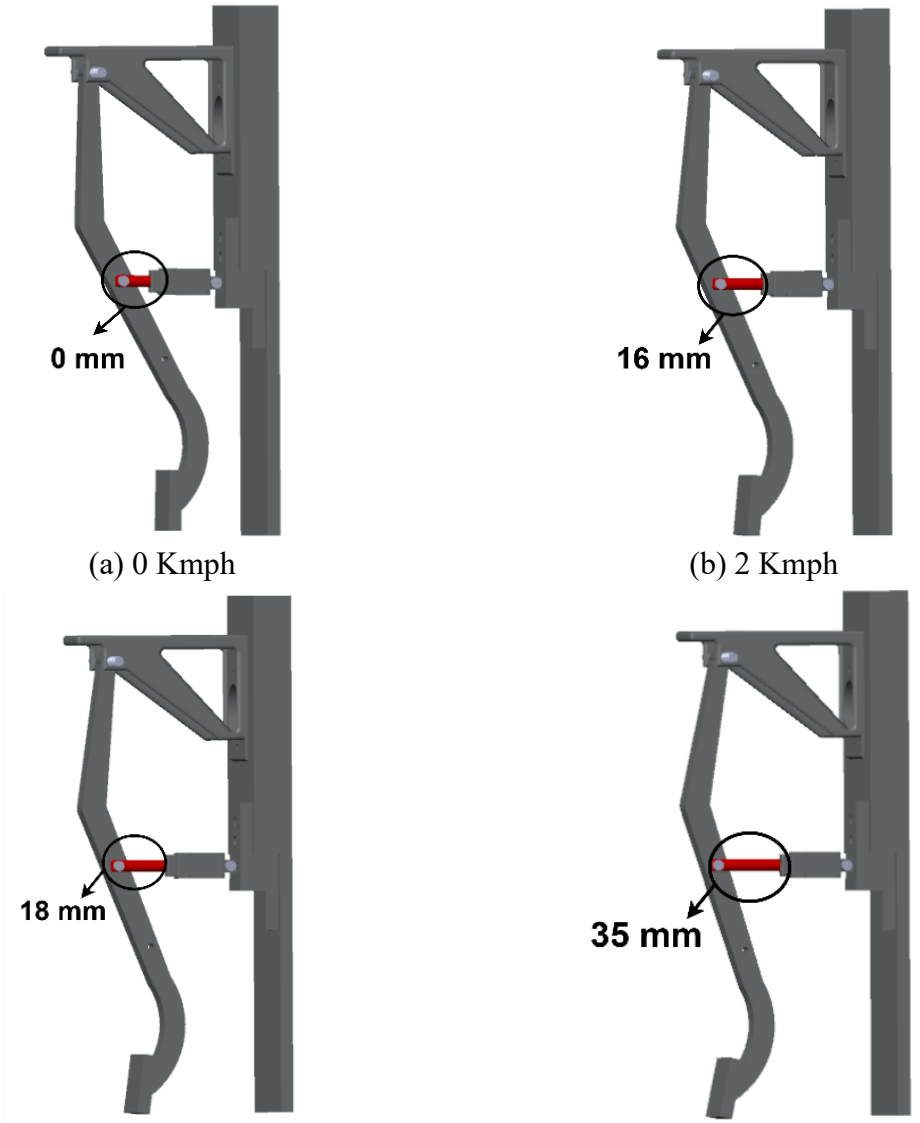


Figure 13. Vehicle Speed Response with Different Reference Speed

https://doi.org/10.31436/iiumej.v26i1.3436



(a) 0 Kmph (b) 2 Kmph

(c) 3 Kmph (d) 5 Kmph

Figure 14. Visualization of Automate Pedal Pressing of Vehicle Response in Simscape

Table 5. System Performance Analysis tuned by SSCA for both Type FLC

Reference Speed	System Performance Analysis			
	Parameter	**FLC Type 1**	**FLC Type 2**	**FLC Type 2 Improvement**
2 Kmph	Overshoot (%)	36.0198	3.5956	32.4242
	Rise time (s)	0.2093	0.5573	Not reduced
	Settling time (s)	3.2538	2.8894	0.3644
	Steady-State Error	-0.0044	-0.0020	0.0024
3 Kmph	Overshoot (%)	41.6032	11.4938	30.1094
	Rise time (s)	0.1799	0.3641	Not reduced
	Settling time (s)	37.6442	36.9800	0.6642
	Steady-State Error	-0.0011	-0.0025	Not reduced
	Steady-State Error	-0.003809	-0.0018	0.002009
5 Kmph	Overshoot (%)	33.2372	21.4962	11.741
	Rise time (s)	0.23740	0.2661	Not reduced
	Settling time (s)	19.6979	19.5361	0.1618

4. CONCLUSION

A new approach towards the development of controlling a linear actuator is presented in the study in the form of a Fuzzy Logic Controller (FLC), which is intended to impose control over vehicle speeds, especially during low-speed driving in heavy traffic conditions where frequent pressing and releasing of the brake pedal is required. Integrating the actuator with the vehicle dynamics model enables the control system to maintain optimal speed by applying the appropriate force on the pedal, alleviating the driver's need for repetitive pedal activities, and is especially beneficial in traffic congestion. The system's performance was evaluated using two different FLC approaches: the traditional FLC Type 1 and the more advanced FLC Type 2. The results obtained effectively demonstrate the concept proposed in this research and confirm that the FLC Type 2 optimized with the SSCA exhibits a significantly lower overshoot than the conventional FLC Type 1 algorithm and a significantly faster settling time while maintaining a considerably small steady-state error. However, the settling time is influenced by the presence of variation. This advancement suggests that it could be possible to substitute the tiresome procedure of the presses and releases of the pedal vehicle among drivers. This system's enabled combination of FLC Type 2 with SSCA improves the overall performance of a real hardware solution. However, transitioning from exercise to application poses many difficulties that can be outlined as follows. Thus, future research will focus on the hardware-in-the-loop testing of the presented system to ensure a comprehensive analysis of the system's behavior and efficiency in a realistic environment. Additionally, the current model is primarily tested for very low-speed scenarios, which is effective in simulating stop-and-go traffic but may limit its broader applicability. Future studies should consider testing the system across different terrains, including urban areas, rural environments, and highways, to evaluate its performance in various driving conditions. This approach aims to bridge the gap between simulation and the implementation of the system on the actual hardware to provide a more reliable and efficient subsystem for usage.

ACKNOWLEDGMENT

This work has been financially supported by the Postgraduate Research Grant Scheme, PRGS, with a university reference number (PGRS230341), and the UMPSA Research Grant Scheme, with a university reference number (RDU220328). The Research and Innovation Department, Universiti Malaysia Pahang Al-Sultan Abdullah, has awarded both grants.

REFERENCES

[1] Stanton NA, Dunoyer A, Leatherland A. Detection of new in-path targets by drivers using Stop & Go Adaptive Cruise Control. Appl Ergon 2011;42:592–601. https://doi.org/https://doi.org/10.1016/j.apergo.2010.08.016.

[2] Yu L, Wang R. Researches on Adaptive Cruise Control system: A state of the art review. Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering 2022;236:211–40. https://doi.org/10.1177/09544070211019254.

[3] Sankar V. REVIEW ON ADAPTIVE CRUISE CONTROL IN AUTOMOBILES. International Journal of Mechanical Engineering & Robotics Research 2014;3:404–9.

[4] Tjolleng A, Yang J, Jung K. Analysis of Leg Muscle Activities and Foot Angles while Pressing the Accelerator Pedal by Different Foot Postures. Applied Sciences (Switzerland) 2022;12:13025. https://doi.org/10.3390/app122413025.

[5] Jammes Y, Weber JP, Behr M. Consequences of Car Driving on Foot and Ankle Mobility and Reflexes. Clin Res Foot Ankle 2017;05. https://doi.org/10.4172/2329-910x.1000233.

[6] Idzham MK, Ahmad S, Abdullah M. Deep learning based prediction model of recurrent pedal pressing for low speed. 8th International Conference on Mechatronics Engineering (ICOM 2022), vol. 2022, 2022, p. 63–7. https://doi.org/10.1049/icp.2022.2266.

[7] Isa HM, Ahmad S, Abdullah M, Rahman A, Ghani N, Noor NMM. Modelling of Pedals Pressing Mechanisms for Low Speed Driving in Road Traffic Delay. IOP Conf Ser Mater Sci Eng 2022;1244:012010. https://doi.org/10.1088/1757-899x/1244/1/012010.

[8] Godjevac J. Comparison between PID and fuzzy control. Ecole Polytechnique Fédérale de Lausanne, Département d'Informatique, Laboratoire de Microinformatique, Internal Report 1993;93.

[9] Coleman CP, Godbole D. A comparison of robustness: fuzzy logic, PID, and sliding mode control. Proceedings of 1994 IEEE 3rd International Fuzzy Systems Conference, 1994, p. 1654–9 vol.3. https://doi.org/10.1109/FUZZY.1994.343945.

[10] Kaur A. Comparison between Conventional PID and Fuzzy Logic Controller for Liquid Flow Control: Performance Evaluation of Fuzzy Logic and PID Controller by Using MATLAB/Simulink. International Journal of Innovative Technology and Exploring Engineering (IJITEE) 2012:2278–3075.

[11] George T, Ganesan V. Optimal tuning of PID controller in time delay system: a review on various optimization techniques. Chemical Product and Process Modeling 2022;17:1–28. https://doi.org/doi:10.1515/cppm-2020-2001.

[12] Hashim AAA, Ghani NMA, Ahmad S, Nasir ANK. System Identification and Control of Linear Electromechanical Actuator Using PI Controller Based Metaheuristic Approach. Applications of Modelling and Simulation 2024;8:213–24.

[13] Wang H, Xu S, Hu H. PID Controller for PMSM Speed Control Based on Improved Quantum Genetic Algorithm Optimization. IEEE Access 2023;11:61091–102. https://doi.org/10.1109/ACCESS.2023.3284971.

[14] Dogruer T, Can MS. Design and robustness analysis of fuzzy PID controller for automatic voltage regulator system using genetic algorithm. Transactions of the Institute of Measurement and Control 2022;44:1862–73. https://doi.org/10.1177/01423312211066758.

[15] Ghani NMA, Nasir NM, Hashim AAA. Comparative Analysis of Spiral Dynamic Algorithm and Artificial Bee Colony Optimization for Position Control of Flexible Link Manipulators. Journal Europeen Des Systemes Automatises 2024;57:773–9. https://doi.org/10.18280/jesa.570315.

[16] Mahmoodabadi MJ, Shahangian MM. A New Multi-objective Artificial Bee Colony Algorithm for Optimal Adaptive Robust Controller Design. IETE J Res 2022;68:1251–64. https://doi.org/10.1080/03772063.2019.1644211.

[17] Yao LJ, Hashim MR, M. O. Tokhi. Artificial Bee Colony Optimization Algorithm with Flexible Manipulator System. International Journal of Electrical Engineering and Applied Sciences (IJEEAS) 2020;3.

[18] Huang M, Tian M, Liu Y, Zhang Y, Zhou J. Parameter optimization of PID controller for water and fertilizer control system based on partial attraction adaptive firefly algorithm. Sci Rep 2022;12. https://doi.org/10.1038/s41598-022-16425-7.

[19] Jamshidpour Ehsan and Parameshachari BD and SE and SC-L and RSH and SMH. Modeling and Optimal Control of Power System Frequency Load Controller by Applying Disturbance in the System by a Modified Version of Firefly Algorithm. In: Razmjooy Navid and Ghadimi N and RV, editor. Metaheuristics and Optimization in Computer and Electrical Engineering: Volume 2:

Hybrid and Improved Algorithms, Cham: Springer International Publishing; 2023, p. 199–240. https://doi.org/10.1007/978-3-031-42685-8_6.

[20] Dhal KG, Das A, Gálvez J. A Novel Fuzzy Logic-Based Improved Cuckoo Search Algorithm. International Journal of Applied Metaheuristic Computing 2022;13:1–29. https://doi.org/10.4018/ijamc.292516.

[21] Ou X, Wu M, Pu Y, Tu B, Zhang G, Xu Z. Cuckoo search algorithm with fuzzy logic and Gauss–Cauchy for minimizing localization error of WSN. Appl Soft Comput 2022;125:109211. https://doi.org/https://doi.org/10.1016/j.asoc.2022.109211.

[22] Pozna C, Precup R-E, Horváth E, Petriu EM. Hybrid Particle Filter–Particle Swarm Optimization Algorithm and Application to Fuzzy Controlled Servo Systems. IEEE Transactions on Fuzzy Systems 2022;30:4286–97. https://doi.org/10.1109/TFUZZ.2022.3146986.

[23] Sathish Kumar A, Naveen S, Vijayakumar R, Suresh V, Asary AR, Madhu S, et al. An intelligent fuzzy-particle swarm optimization supervisory-based control of robot manipulator for industrial welding applications. Sci Rep 2023;13:8253. https://doi.org/10.1038/s41598-023-35189-2.

[24] Kasruddin Nasir AN, Ahmad MA, Tokhi MO. Hybrid spiral-bacterial foraging algorithm for a fuzzy control design of a flexible manipulator. Journal of Low Frequency Noise Vibration and Active Control 2022;41:340–58. https://doi.org/10.1177/14613484211035646.

[25] Nasir ANK, Razak AAA, Ismail MTR, Ahmad MA. A Hybrid Spiral-Genetic Algorithm for Global Optimization. Journal of Telecommunication, Electronic and Computer Engineering (JTEC) 2018;10:93–7.

[26] Ghani NMA, Othman A, Hashim AAA, Nasir ANK. Comparative Analysis of PID and Fuzzy Logic Controllers for Position Control in Double-Link Robotic Manipulators. Journal of Intelligent Systems and Control 2023;2:183–96. https://doi.org/10.56578/jisc020401.

[27] Karim MI, Hashim AAA, Ghani NMA. Control of Double Link Flexible Robotic Manipulator System. Journal Europeen Des Systemes Automatises 2022;55:759–63. https://doi.org/10.18280/jesa.550607.