Research Article

Optimal Position Fuzzy Control of an Underactuated Robotic Finger

Francisco J. Espinosa Garcia,1,2 Esther Lugo-González,3 Arturo Telléz-Velázquez,4 Manuel Arias-Montiel,3 and Marco Ceccarelli2

1Postgraduate Division, Technological University of the Mixteca, Huajuapan de León 69000, Mexico
2LARM 2, University of Rome Tor Vergata, Rome 00133, Italy
3Institute of Electronics and Mechatronics, Technological University of the Mixteca, Huajuapan de León 69000, Mexico
4Institute of Computing, Technological University of the Mixteca, Huajuapan de León 69000, Mexico

Correspondence should be addressed to Manuel Arias-Montiel; mam@mixteco.utm.mx

Received 18 March 2022; Revised 27 May 2022; Accepted 28 June 2022; Published 20 July 2022

Academic Editor: Ardashir Mohammadzadeh

Copyright © 2022 Francisco J. Espinosa Garcia et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

In this paper, the optimal position control of an underactuated robotic finger is presented. Two trajectories, one for the proximal and the other for the medial phalanx, are proposed in order to emulate the finger’s flexion/extension movements. A Mandani fuzzy control is proposed due to the lack of a precise dynamical model of the system. In order to obtain the control parameters, an optimization strategy based on the membership functions is applied. Genetic algorithms (GA) are commonly used as an optimization method in diverse applications; however, in this case, the use of an autoadaptive differential evolution method is proposed in order to obtain a superior convergence behavior. Simulations of the virtual prototype are carried out using MATLAB/Simulink software to display the trajectory tracking. The results show that the maximum error between the proposed and obtained trajectories is $3.1352 \times 10^{-4}$ rad.

1. Introduction

Currently, control development for robotic hands continues to be a topic of interest as researchers seek to recreate human hand interaction with their environment when developing prototypes and their interaction with the environment. This objective is difficult to achieve due to the challenging nature of controlling robotic hands, as they are relatively complex mechatronic systems, which allow the user to hold, manipulate, and make use of different objects and tools. As a possible solution, some researchers have focused on improving control tasks in order to generate a robust grasp to reliably hold any object. In addition, the selection criteria of actuators are an important factor because the size, weight, and torque are variables that must be considered. Generally, the control systems have a closed-loop structure, since they seek to reduce errors so that the hand finger can maintain a specific desired position [1, 2], by using pneumatic [3–5] and touch [6–8] sensors. In the literature, the most common control schemes are the proportional-integral-derivative (PID) control [9–11] and fuzzy control [12–14]. Controllers based on fuzzy logic are an alternative solution that does not require a mathematical model such as the PID [15]. A fuzzy logic controller (FLC) is a heuristic approach composed of a rules base proposed by the designer. The FLC is a nonlinear system with a knowledge based on fuzzy If-Then rules, and in most cases, the fuzzy rules are proposed by an expert who knows the process. In order to generate an output, membership functions are used to specify the degree of membership based on inputs. FLC must have a flexible behavior to adapt to various situations, as well as being robust to maintain the state of the desired output. The implementation of FLC is fairly common for solving problems, where (a) the systems are partially defined, (b) systems with variables that cannot be measured, and (c) system with large disturbances [12, 14, 16]. The principal fuzzy systems are Mandani and...
Takagi-Sugeno. In particular, the Mandani systems use various techniques that allow fuzzy set membership function tuning, such as genetic algorithms (GA) [16], adaptive neural networks [17, 18], artificial bee colony optimization [19–21], ant colony optimization [22–24], and evolutionary algorithms [25–27].

Recently, fuzzy logic systems have been combined with nonlinear approaches in order to obtain adaptive control systems. Zhang et al. [28] proposed an FLC combined with sliding-mode technique for controlling underactuated systems. The authors provided a detailed stability analysis and a numerical example of a nonspecified system to prove the performance of the closed-loop system. However, the fuzzy system is not optimized. Li et al. [29] combined a fuzzy logic system with a back-stepping controller and a fuzzy state observer to solve the tracking problem for a class of uncertain nonlinear systems with unmodeled dynamics and disturbances. The stability of the systems is proven to be input-to-state practically stable based on the small-gain theorem, but the fuzzy logic system’s membership functions are not optimized. Chang et al. [30] presented an adaptive fuzzy tracking control for a class of pure feedback and switched nonlinear systems with unknown gain. The authors did not provide details on the fuzzy system, such as the procedure to determine the membership functions.

Optimization methods have helped with the complex task of finding the appropriate parameter values and structure for the fuzzy systems. The most used algorithms for fuzzy systems optimization are GA because they are considered as global optimizers, unlike derivative methods that have the disadvantage of being prone to local minimum stagnation. They are also considered as good optimizers because of the existence of a population of potential solutions, which may be used to discard several local minima, meaning that there is a higher probability of finding the global minimum [31]. Several proposals search for optimal membership function parameters. For example, in [32] the authors proposed a methodology for optimizing a fuzzy controller for the speed of a permanent magnet synchronous motor (PMSM) without expert knowledge. For this approach, 14 variables are used for the optimization, 5 for the rule base parameters, 6 for the membership functions, and 3 for the scaling factors. A method for automatically generating fuzzy rules is presented for a system with 25 fuzzy rules needed, and each rule needs 3 bits [33]. A fuzzy controller applied to the wellhead back pressure control systems is optimized through the membership function, control rules, quantization, and scaling factors. The parameters are error, rate, scale, integral, and differential scale factors [34]. Despite this, there are other methods, such as the differential evolution (DE), which have more advantages compared to GA, one example being that they are more effective at searching for numerical problems and finding the optimum global solution [35, 36].

Motivated by the limitations of GA and other optimization methods, this paper presents the implementation of an auto adaptive differential evolution algorithm (AADE) [37] for optimizing the parameters of a fuzzy logic controller (FLC) for an underactuated robotic finger in order to track the trajectory that represents the flexion-extension finger movements. This optimization algorithm does not require gradient information, which means that the optimization problem does not need to be differentiable. In addition, AADE is based on symmetrical discourses and fuzzy normalized systems, and the parameters F (factor scale) and Cr (crossover constant) show evidence of self-adaptation during the evolution phase, unlike GA or classical differential evolution (DE) algorithms, increasing its power and execution.

The remainder of this paper is organized as follows. In Section 1, a brief description of the prototype is presented. In Section 2, the development of the FLC and the application of AADE for parameter optimization is described. In Section 3, details of the numerical implementation of the optimized FLC are provided. Final remarks and some proposed future works for this research are given in Section 4.

2. Materials and Methods

2.1. Prototype Design. The robotic hand AMH-II (anthropomorphic metamorphic hand), originally presented in [38], has four fingers, all of which have three joints except for the thumb. The index, middle, and ring fingers are powered by an underactuated mechanism in order to use a single motor, thereby reducing the weight of the finger. This configuration allows the hand to perform the flexion/extension finger movements. The thumb has three joints, although only two of them are used for the flexion/extension movements, while the other one is used for the rotational movement. Details on the design and kinematic analysis of the AMH-II are presented in [38]. The virtual prototype is shown in Figure 1, and as can be observed, each finger has two motors to control movement. The motors employed are Pololu micro motors 100:1360 RPM 6V, and the joints are connected to these motors through a transmission system comprised of a worm-gear mechanism. The palm, unlike other robotic hands, can be moved through a motor fixed at the joint of the link, where the fingers are located and the link, where the thumb is located (thumb motor). Table 1 shows the ranges of motion for flexion/extension movements of the hand, corresponding to each finger joint (except for the thumb). Similarly, the movement measurements and ranges of the phalanges of the presented prototype are also shown.

2.2. Implemented Controller. For the control system, a Mandani fuzzy system was implemented, which includes a micro-motor that controls the position of the fingers. The transfer function corresponding to the servo motor used was obtained from [39].

\[
G(s) = \frac{V(s)}{\theta(s)} = \frac{24}{s^2 + 24s + 10},
\]  

(1)

where \(V(s)\) is the input voltage applied to the motor, while \(\theta(s)\) is the rotor position in radians (rads) after applying a specific voltage. This value is used as feedback as shown in the schematic diagram in Figure 2.
The reference block represents the desired trajectory, while \( e \) and \( de \) are the error signal and its derivative, respectively, which serve as inputs for the fuzzy logic controller (FLC). This block sends a voltage signal to the motor block and its position is evaluated by the AADE so that the membership function parameters are changed to reduce the error.

The fuzzy system has three variables, error \( (e) \) and error change \( (c) \) are the inputs, and voltage \( (v) \) is the system output. Moreover, the variables have their own discourse universe range (Table 2) [37]. The configuration of the fuzzy system is established in 9 rules as shown in Table 3. Symbols NE, ZE, and PE are the linguistic terms for “negative error,” “zero error,” and “positive error”; NC, ZC, and PC are the linguistic terms for “negative change,” “zero change,” and

<table>
<thead>
<tr>
<th>Phalanx</th>
<th>Length of the prototype (mm)</th>
<th>Range of human hand movement</th>
<th>Range of prototype movements</th>
</tr>
</thead>
<tbody>
<tr>
<td>Distal (index and middle)</td>
<td>14</td>
<td>0–90’</td>
<td>0–85’</td>
</tr>
<tr>
<td>Middle (index and middle)</td>
<td>38.16</td>
<td>0–90’</td>
<td>0–75’</td>
</tr>
<tr>
<td>Proximal (index and middle)</td>
<td>41.89</td>
<td>0–90’</td>
<td>0–75’</td>
</tr>
</tbody>
</table>

**Table 1: Ranges for flexion/extension movements comparison.**

**Table 2: Ranges for the fuzzy system.**

<table>
<thead>
<tr>
<th>Variable</th>
<th>Discourse ranges</th>
</tr>
</thead>
<tbody>
<tr>
<td>Position error ( e )</td>
<td>([0, (\pi/2)])</td>
</tr>
<tr>
<td>Error change ( c )</td>
<td>([0, (\pi/6)])</td>
</tr>
<tr>
<td>Voltage ( v )</td>
<td>([-8.19, 8.19])</td>
</tr>
</tbody>
</table>
“positive chance”; and NV, ZV, and PV are the linguistic terms for “negative voltage,” “zero voltage,” and “positive voltage.”

The fuzzy function proposed for the control is Gaussian for ZV, ZE, and ZC. Sigmoid functions were used, as were Z shape membership functions for NE, NC, and NV, and for the rest of the variables, the S shape function was implemented. Three parameters are needed in order to define a Gaussian function, while 2 parameters are needed for sigmoid functions. The optimization problem is then defined by 21 parameters. The number of variables requires a considerable amount of computation time and the use of resources for the optimization algorithm. Therefore, the following methodology was implemented in order to reduce the number of variables.

2.3. Methodology. In [37] a method for reducing the number of parameters for a fuzzy set control through the use of symmetrical discourses was proposed. In basic terms, these variables can distribute several sets to the positive part of the discourse universe, and then reflect them all in the negative part. It is important to consider that not all the applications can be addressed with these variables, as it is preferable that every consequent action of the fuzzy system has the same magnitude in both positive and negative directions. In this case, the symmetrical discourses can be applied because the servomotors have the same range for the output in the positive and negative response.

For example, in Figure 3, different membership functions (MFs) are shown, all of which are symmetrical. The NULL or Zero set corresponding to the triangular function is located exactly in the middle of the discourse universe (red line) while the opening of the triangular function is varied, denoted as $d_1^e$. In order to achieve a symmetric function, the provided value is changed to negative and as a result, the complete function is obtained. The parameters of S function are defined by $d_1^c$ and $d_1^v$, and to obtain the Z function, the $S$ function.

Table 3: Rules set.

<table>
<thead>
<tr>
<th>Voltage</th>
<th>NC</th>
<th>ZC</th>
<th>PC</th>
</tr>
</thead>
<tbody>
<tr>
<td>NE</td>
<td>NV$^{(1)}$</td>
<td>NV$^{(2)}$</td>
<td>PV$^{(3)}$</td>
</tr>
<tr>
<td>ZE</td>
<td>NV$^{(4)}$</td>
<td>ZV$^{(5)}$</td>
<td>PV$^{(6)}$</td>
</tr>
<tr>
<td>PE</td>
<td>NV$^{(7)}$</td>
<td>PV$^{(8)}$</td>
<td>PV$^{(9)}$</td>
</tr>
</tbody>
</table>

Figure 3: Example of a symmetrical variables.
parameter functions are reflected. With this approach, the shape of the error sets can be defined with three parameters. This same approach is also applied for the change and voltage variables. When applying this method to the proposed problem, the number of parameters is reduced from 21 to 9.

In the methodology proposed in [37], GA was used as an optimization method, while in this paper the auto adaptive differential evolution proposed by [40] is used. This is because it tends to converge to the solution in fewer iterations (ten), and does not depend on the number of the population as with GA. A description of the algorithm is presented in the next section.

2.4. Autoadaptive Differential Evolution Algorithm (AADE). The AADE is a type of evolutionary algorithm used for multidimensional real-valued functions. This method does not require gradient information, which means that the optimization problem does not need to be differentiable. In addition, it has an efficient memory utilization as well as being effective and robust [36, 40–42]. The main feature of AADE is that parameters, such as \( F \) (factor scale) and \( C_r \) (crossover constant), show evidence of self-adaptation during an evolution phase, unlike GA or classical differential evolution (DE) algorithms, increasing its power and execution. Figure 4 summarizes the AADE operation flow chart. More specifically AADE’s basic strategy can be described as shown below.

2.4.1. Population. The population is a vector \( x \) defined as

\[
x = (x_1, x_2, \ldots, x_D),
\]

where \( D \) is the dimensionality of the function \( f \). The variables domains are defined by their lower and upper bounds: \( x_{j,\text{low}} \) and \( x_{j,\text{upp}} \); \( j \in \{1, \ldots, D\} \). The initial population is selected randomly between the lower \((x_{j,\text{low}})\) and upper \((x_{j,\text{upp}})\) bounds defined for each variable \( x_j \). The bounds are specified according to the user.

2.4.2. Mutation. In every iteration (generation) \( G \), DE uses the mutation operation for producing the donor vector. For each target vector \( x_i \), \( G \), \( i = 1, 2, 3, \ldots, \text{NP} \), a mutant vector

\[
U_{i,G+1} = \begin{cases} U_{j,G+1}, & \text{if (rand } b(j) \leq C_r) \text{ or } j = \text{rnbr}(i), \\ x_{j,G}, & \text{if (rand } b(j) > C_r) \text{ and } j \neq \text{rnbr}(i), \end{cases}
\]

where \( \text{rand } b(j) \) is the \( j \)-th evaluation of a uniform random number generator within the range \([0, 1]\), which has to be determined by the user. \( \text{rnbr}(i) \) is a randomly chosen index \( e1, 2, \ldots, D \), which ensures that \( U_{i,G+1} \) receives at least one parameter from \( U_{i,G+1} \).

2.4.3. Crossover. The crossover operator is used to increase the perturbed parameter vectors’ diversity. The trial vector

\[
U_{i,G+1} = (U_{1,i,G+1}, U_{2,i,G+1}, \ldots, U_{D,i,G+1}),
\]

is formed, where

\[
v_i = \{v_{1,i},v_{2,i}\} \text{ is generated according to a specific mutation scheme. The most widely used in differential evolution are}
\]

\[
\begin{align*}
\text{DE} & \left(\text{rand}/1v_i\right) = x_{r_1} + F(x_{r_2} + x_{r_3}), \\
\text{DE} & \left(\text{best}/1v_i\right) = x_{\text{best}} + F(x_{r_1} + x_{r_2}), \\
\text{DE} & \left(\text{rand}/2v_i\right) = x_{r_1} + F(x_{r_2} + x_{r_3}) + F(x_{r_4} + x_{r_5}),
\end{align*}
\]

where \( r_1, r_2, r_3, r_4 \) and \( r_5 \) are random integers within the range \([1, \text{NP}]\) and are different to index \( i \). \( r_i \neq r_j \neq r_k \neq r_4 \neq r_5 \). \( F \) is a real and constant factor, which controls the amplification of the differential variation.

2.4.4. Selection. The trial vector \( U_{i,G+1} \) is compared to the target vector \( x_i \) using the greedy criterion in order to decide whether or not it should become a member of generation \( G+1 \). If vector \( U_{i,G+1} \) yields a smaller cost function value than \( x_i \), then \( x_i \) is set to \( U_{i,G+1} \); otherwise, the old value \( x_i \) is retained.
2.4.5. F and Cr Value Adjust. The estimation of the F and Cr values depends on the following equations:

\[ F_{i+1,G} = F_{i+1,G} - \text{rand}_{i,j}[0,1] \cdot \left( \frac{f(U_{i,G}) - f(Y_{i,G})}{f(U_{i,G})} \right), \]

\[ Cr_{i+1,G} = Cr_{i+1,G} - \text{rand}_{i,j}[0,1] \cdot \left( \frac{f(U_{i,G}) - f(Y_{i,G})}{f(U_{i,G})} \right), \]

\[ F_{G+1} = \text{rand}_{i,j}[0,1] \cdot \left( \frac{\max - \min}{\max} \right), \]

\[ C_{G+1} = \text{rand}_{i,j}[0,1] \cdot \left( \frac{\max - \min}{\max} \right). \]

where \( F_{i+1,G} \) and \( Cr_{i+1,G} \) represent the estimation of the F and Cr of each vector. \( f(U_{i,G}) \) is the new vector and \( f(Y_{i,G}) \) is the current vector. \( F_{G+1} \) and \( C_{G+1} \) are the new F and Cr values for the next generation. In addition, the max and min are values obtained from an array that contains all \( F_{i+1,G} \) and \( Cr_{i+1,G} \).

The methodology implemented is as follows:

1. Start AADE algorithm
   a. Set the initial parameters.
   b. Initialize all NP individuals with random positions.
   c. Generate three and four individuals from the current population randomly based on the mutation used.
   d. Apply the mutation operator to form the donor vector using equation (6).
   e. Apply the crossover operator in order to increase diversity.
   f. Calculate the fitness of each individual.
   g. Calculate the new F and Cr values.

2. Once the best individual is found, every parameter must be assigned to its corresponding set.

3. Evaluate the fuzzy system.

2.5. Simulations and Results

2.5.1. Experimental Configuration. The numerical experiments were carried out on a Legion Y540-15IRH laptop with Core i7 9th gen processor, 16 GB RAM, and a clock speed of 2.6 GHz with turbo boost of up to 4.5 GHz.

Considering the range of flexion/extension movements generated by the human finger, a middle part of a sinusoidal trajectory was proposed as a reference for the position control. Due to the robotic finger design configuration, the amplitude of the function is 1.4835 rad (85°) for the proximal phalanx and 1.3962 rad (80°) for the medial phalanx. The equation of the mean squared error (MSE) is used in order to measure the performance of the fuzzy controller. The goal function is as follows:

\[ \min \left\{ \frac{1}{N} \sum_{k=1}^{N} [x(k) - y(k)]^2 + M_1 h_1 \right\}, \]

where \( h_1(x) = 0 \quad \rightarrow \quad \text{the values are in the range} \)

\( h_1(x) = 1 \quad \rightarrow \quad \text{otherwise}. \)

In equation (1), \( x(k) \) represents the value at time \( k \), \( y(k) \) is the value produced by the system at time \( k \), and \( N \) represents the total number of samples. \( M_1 \) is a constant of a very high value that penalize the goal function when the constraint fails.

The optimization problem is defined by

1. Design variables \( X = [d_1^d, d_2^d, d_3^d, d_4^d, d_5^d, d_6^d, d_7^d, d_8^d, d_9^d, d_{10}^d, d_{11}^d]. \)
(ii) Limits of the variables: \( d_1 \), \( d_2 \), \( d_3 \) \( \in \) \( [0, (\pi/2)] \); \( d_1^c \), \( d_2^c \), \( d_3^c \) \( \in \) \( [0, (\pi/6)] \); and \( d_1^v \), \( d_2^v \), \( d_3^v \) \( \in \) \( [-8.16, 8.16] \).

(iii) Parameters of the algorithm:

- Population size = 50.
- Generation limit = 150.
- \( F = 0.5 \).
- Crossover rate = 0.2.

**Table 4:** Parameters for set distribution with different mutation strategies.

<table>
<thead>
<tr>
<th>Mutation strategy</th>
<th>( d_1^e )</th>
<th>( d_2^e )</th>
<th>( d_3^e )</th>
<th>( d_1^c )</th>
<th>( d_2^c )</th>
<th>( d_3^c )</th>
<th>( d_1^v )</th>
<th>( d_2^v )</th>
<th>( d_3^v )</th>
</tr>
</thead>
<tbody>
<tr>
<td>DE/rand/1</td>
<td>0.5677</td>
<td>0.0873</td>
<td>0.0014</td>
<td>0.1883</td>
<td>0.3423</td>
<td>0.1870</td>
<td>0.9259</td>
<td>5.7279</td>
<td>7.6088</td>
</tr>
<tr>
<td>DE/best/1</td>
<td>0.7120</td>
<td>0.0957</td>
<td>0.0053</td>
<td>0.2691</td>
<td>0.4952</td>
<td>0.2781</td>
<td>3.6487</td>
<td>4.0871</td>
<td>6.5742</td>
</tr>
<tr>
<td>DE/rand/2</td>
<td>0.3860</td>
<td>0.0406</td>
<td>0.0180</td>
<td>0.2568</td>
<td>0.2537</td>
<td>0.3045</td>
<td>0.3248</td>
<td>6.003</td>
<td>6.6644</td>
</tr>
</tbody>
</table>

**Table 5:** Comparison of different mutations strategies in terms of error value.

<table>
<thead>
<tr>
<th>Mutation strategy</th>
<th>Fitness Pha1</th>
<th>Fitness Pha2</th>
</tr>
</thead>
<tbody>
<tr>
<td>DE/rand/1</td>
<td>3.1352E – 04</td>
<td>2.7109E – 04</td>
</tr>
<tr>
<td>DE/best/1</td>
<td>7.2511E – 04</td>
<td>6.4325E – 04</td>
</tr>
<tr>
<td>DE/rand/2</td>
<td>4.1357E – 04</td>
<td>4.0554E – 04</td>
</tr>
</tbody>
</table>

**Figure 6:** Membership function obtained for inputs: (a) error and (b) change.
The proposed optimization algorithm was implemented using Matlab software, while the parallel mode was used because it increases the convergence speed, thereby demonstrating another advantage of using this approach. The algorithm was iterated using the mutation strategies from equations (5)–(7), with 30 total iterations, and 10 per each mutation strategy. In order to evaluate the proposed fuzzy control, some simulations were carried out using the Simulink environment. For simulation purposes, a finger from the hand model presented in Figure 1 was exported to Simmechanics, and the control was implemented in two parts: one for the proximal phalanx and another for the middle phalanx. This is because the distal and middle phalanxes are connected through a link resulting in a subactuated mechanism. Only one finger is simulated because it is the same movement for the other fingers. The block diagram is shown in Figure 5.

3. Results and Discussion

Table 4 shows the best performance solutions for the finger control with the three used mutation strategies.

The fitness of each solution is shown in Table 5. Pha1 and Pha2 represent the proximal and middle phalanges, respectively. All solutions are acceptable because the error is very close to zero, although the best solution is presented by DE/rand/1, followed by DE/rand/2 and DE/best/1.

Using DE/rand/1 solution, the input and output values of the fuzzy system are defined. Figures 6 and 7 show the membership functions and Figure 8 shows the 3D surface viewer of the fuzzy logic controller to get a perspective of rules design.

The results of the virtual prototype and comparison between the desired trajectory and the trajectory by the control of each phalanx are shown in Figures 9 to 11. Figure 9 shows the virtual configuration of the robotic underactuated finger, while the position of the distal phalanx depends on the movement of the middle phalanx, which is transmitted by the $P$ bar.

Based on Figure 10, DTPP represents the desired trajectory of the proximal phalanx and OTPP is the obtained trajectory. The maximum rotation value of the proximal phalanx is 1.4835 rad due to the configuration of the mechanism. As can be seen in the finger’s final position, the base is represented by the $B$ block and the movement by the $A$ block. The time simulation is 3.2 s, and the OTPP converges to DTPP with an error of $3.1352E−04$ rad. This error is more visible in two intervals, from 1.3 to 2 and from 2.5 to 3.2. Between the trajectories, there are oscillations that cannot be appreciated because they are very small ($1E−03$ rad) and appear in the interval from 0 to 1.4 s.

Referring to the results of Figure 11, DTMP and OTMP are the desired and obtained trajectories of the medial phalanx. The maximum rotation value is 1.3089 rad due to the finger structure. Compared to the human finger range there is a difference of 16.66%, meaning that the finger is not able to grasp small cylindrical objects (diameter of less than 60 mm). The error between trajectories is $2.7109E−04$ rad.

According to analysis of its behavior, the controller presents a fast response due to the extremely small error tolerance (0.1 approx.), based on the membership function intersection (Figure 6(a)). Under the same conditions and modifying only the $d^{13}$ values, the system response speed decreases. On the other hand, after only modifying the parameter $d^{3}$ to values less than 0.2, the system does not present changes. It means that the motor does not change the direction of rotation and its velocity is the same under any condition. In this way, the values for $d^{1}$, $d^{2}$, and $d^{3}$ affect the response speed of the controller, the signal to change...
The rotation of the motor is defined by the parameters $d_1$, $d_2$, and $d_3$. The rest of the parameters define the motor rotation speed.

Table 6 shows a comparison between the present work and some results previously reported in the literature, and it can be observed that almost all of the finger models are underactuated. The most common technique for controlling the position of the robotic finger is the PID controller with some variants. A fuzzy controller is also used but the parameters are manually selected. Based on information in the table, in some cases, the finger dynamic model is presented in order to control the three joints. In [10, 12, 14], the authors report a step response with a steady-state error (SSE) of 0, although in [10] the time response is more than 10 sec. On the other hand, the numerical results presented in [12, 14] show settling times of less than 5 sec, but the control parameters are manually obtained. Tasar et al. [11] used a PID controller for a regulation problem of a full-actuated
A maximum overshoot of $14.902°$ was achieved for a reference of $90°$ for each actuated phalanx. Sinusoidal trajectories are proposed in [43] and different control techniques are presented. As in previous cases, the control parameters are tuned manually. The considered trajectory is a sinusoidal function with an amplitude range $[0–(−1)]$ rad in PID and SMC (sliding model control) cases, which is very similar to what is reported in this work but with a larger error. The main difference with our work is that the controller parameters are automatically adjusted, and the obtained error is the smallest in all the presented trajectories.

Regarding the optimization of the fuzzy control, other works have presented methods that, in some cases, optimize more than 10 variables [29]. Moreover, the most common algorithm is GA, although one of the main problems with this type of algorithm is that it is largely dependent on the population size. Moreover, the selection of the initial parameters represents a challenge because selecting the wrong values can lead to the algorithm not converging. In order to avoid this problem, the methods applied in this paper show that it is possible to reduce the number of parameters, considering that the fuzzy set is symmetrical. Instead of using a GA algorithm, it was decided to implement an AADE algorithm because the control parameters $F$ and $CR$ are automatic and the time convergence is smaller compared to GA. Another advantage is that the AADE has the potential to generate a fuzzy control with no previous knowledge of the system.

### 4. Conclusion

When removing the empiricism of a fuzzy controller, a nonlinear problem with several dimensions appears, which can be solved by focusing on the membership function to generate a desired control. In this paper, the auto adaptive differential evolution algorithm was implemented because of the limitations of other methods such as genetic algorithms, artificial bee colonies, or ant colonies. The chosen method performs a sophisticated local search through the recombination method, and the way in which the mutated method

---

**Figure 11:** Comparison of desired and obtained trajectory for medial phalanx.

**Table 6:** Comparison between different control strategies.

<table>
<thead>
<tr>
<th>Author</th>
<th>Control</th>
<th>Finger dynamic model</th>
<th>Trajectories</th>
<th>Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>Calderon et al. [10]</td>
<td>PI</td>
<td>No</td>
<td>Step 17 mm</td>
<td>SSE 0</td>
</tr>
<tr>
<td>Tasar et al. [11]</td>
<td>PID</td>
<td>Yes</td>
<td>Step (for three phalanxes)</td>
<td>SSE 0.255 deg (max)</td>
</tr>
<tr>
<td>Ghazali et al. [14]</td>
<td>Fuzzy-PID</td>
<td>No</td>
<td>Step 90 deg</td>
<td>SSE 0 deg</td>
</tr>
<tr>
<td>Ghazali et al. [14]</td>
<td>Fuzzy</td>
<td>No</td>
<td>Step 90 deg</td>
<td>SSE 0.12 deg</td>
</tr>
<tr>
<td>Raković et al. [12]</td>
<td>Fuzzy</td>
<td>No</td>
<td>Step 100 rad</td>
<td>SSE 0 rad</td>
</tr>
<tr>
<td>Jalani et al. [43]</td>
<td>PID</td>
<td>Yes</td>
<td>Sinusoidal $[0–(−1)]$ rad</td>
<td>0.0159 rad</td>
</tr>
<tr>
<td>Jalani et al. [43]</td>
<td>Adaptive</td>
<td>Yes</td>
<td>Sinusoidal $[0–4]$ rad</td>
<td>0.0791 rad</td>
</tr>
<tr>
<td>Jalani et al. [43]</td>
<td>SMC</td>
<td>Yes</td>
<td>Sinusoidal $[0–(−1)]$ rad</td>
<td>0.0167 rad</td>
</tr>
<tr>
<td>Our proposal</td>
<td>Optimized fuzzy</td>
<td>No</td>
<td>Sinusoidal $[0–1.483]$ rad</td>
<td>$3.1352E−04$ rad</td>
</tr>
</tbody>
</table>
vector is calculated means that the algorithm depends less on an initial position. Furthermore, another advantage is that $F$ and $C_r$ values are calculated automatically. Additionally, the use of the symmetrical discourses is different to other methods for finding the membership function mentioned in the introduction. Specific information about the system is neither not needed nor necessary to work with all the variables shown in the proposed problem. At the start of the process, it was composed of 21 variables, and before applying the method, the variables were reduced to 9. With the obtained vector, the MFs were generated to the fuzzy control. From the simulation, the obtained results proved that the control is able to perform flexion-extension movements following the proposed trajectory with an error of almost zero, despite not having a mathematical model of the robotic hand. It would be very difficult to achieve these results by selecting the controller parameters empirically.

The following step in this research is to prove experimentally the effectiveness of the proposed controller by carrying out tests on the physical prototype reported in [38]. In addition, future studies will focus on the optimization of membership functions to generate movement in the palm and doing fingers able to perform different kinds of grasps to evaluate the efficiency of this type of control scheme. Another interesting topic to be addressed in future works is the consideration of type-3 fuzzy logic systems (T3FLS) to improve the control system performance. T3FLS have been proposed to handle an increased number of uncertainties [44]. However, their main disadvantages are that the structure of the fuzzy sets has been assumed to be constant and the membership computation is complicated. Moreover, studies on the parameter optimization of T3FLS have been scarcely reported in literature [45]. Therefore, combining AADE algorithms with T3FLS could be a promising alternative to provide robustness and improve the fuzzy logic controller performance.

Data Availability

The data that support the findings of this study are available upon request to the corresponding author.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

The authors are grateful to the National Council of Science and Technology (CONACYT) and to the Project 621 from the Cathedra’s Program for supporting the research presented in this paper.

References

Mathematical Problems in Engineering

12


