A novel interval Fuzzy Logic Controller Type-2-PI Controller for Concise Semi-Active Ankle-Foot Prosthetic

E. G. SHEHATA[1], MARIEM Y. WILLIAM[2,*], A. A. HASSAN[1], KHALIL IBRAHIM[3,4]
[1]Faculty of Engineering,
Minya University,
EGYPT

[2]El Minya Higher Institute for Engineering and Technology,
Minya,
EGYPT

[3]Mechatronics Department,
Assiut University,
Assiut,
EGYPT

[4]Mechatronics Engineering Program,
Sphinx University,
Assiut,
EGYPT

Corresponding Author

Abstract: - The main goal of designing the semi-active ankle prosthetic model is to simulate the job of the lost limb. In the first, two PI controllers are designed to regulate motor speed and current. In the second, two FLCT1-PI are designed. FLCT1-PI is designed to tune online the gains of the PI-controller. In the third, FLCT2-PI controller is designed. FLCT2-PI is designed to tune online the gains of PI of the speed control only. The three stages of control are compared. The results reveal that a Concise semi-active ankle-foot prosthetic with a FLCT2-PI method is better than a FLCT1-PI.

Key-Words: - prosthetic, ankle, PI controller, fuzzy type-2 controller, MATLAB/Simulink, fuzzy logic control type-1-PI controller, fuzzy logic control type-2-PI controller.

Received: November 27, 2024. Revised: March 7, 2025. Accepted: April 26, 2025. Published: June 12, 2025.

1 Introduction

Transtibial amputees are becoming more common around the world because of numerous traffic and industrial accidents, as well as injuries sustained during daily sports activities, [1]. A variety of ankle prosthetics have been created to restore/assist below-ankle amputees' loss of gait function. The ankle joint's major job is to provide enough energy support for the body to move forward, [2]. The control's purpose is to steer the powered ankle prosthesis to the specified ankle angle. To adjust the progression deviation, the general control algorithm is to apply mechanical torque to a controlled ankle prosthetic. There are many types of controllers that can be applied to the Concise semi-active ankle prosthetic CSAAP.

The Conventional PI Controller Technique CPICT can be used in the CSAAP model because of the simplicity of the position and speed controls of the DC electric motor DCEM. It is important for the system to determine the proportional (P), and integral (I) parameters of CPICT. The signal constraint [3] method is used to determine these gain parameters. However, there are some problems, such as determining the maximum gain value for the given control system or finding the oscillation period in this method. Intelligent controllers are frequently used in many areas and provide very good results. These controllers have special calculation features for solving specific problems. The fuzzy logic controller type-1 (FLCT1) has been widely used in recent years. FLCT1 is described as a set of rules, that can be used to describe the action of complex systems that cannot be defined mathematically. FLCT1 has attracted great interest in numerous fields such as production techniques, decision-making, nonlinear approach, data analysis, modern information technology, and pattern recognition. Recently, one of the most remarkable research topics is the Fuzzy Logic Controller Type-2 (FLCT2) structure. Due to the difficulties of implementing this controller structure in recent years, academics and scientists have not shown enough interest. Because FLCT2s have better modeling of uncertainties than FLCT1s, and have better results in complex systems. In addition, FLCT2 ensures a robust and adaptive structure for high-performance control against system uncertainties and parameter changes, [4], [5]. So, the FLCT2 is used to control the CSAAP model in this work.

For the past two decades, research in the field of control has continuously inspired the use of FLCT2s design. Several researchers have demonstrated FLCT2s for various applications. For the mobile robot's route tracking issue, the authors of [6] presented an FLCT2-PID controller in a cascade control architecture. The control parameters of FLCT2 are optimized using a Big Bang–Big Crunch optimization methodology, which has low computational time and faster speed of convergence. The results confirm that the optimized FLCT2-PID controller performance is superior to optimized FLCT1-PID and PID controllers. The development of FLCT2-PID controllers for redundant robots was most recently presented in [7]. For various applications, like the inverted magnetic levitation system [8], AVR [9], etc., several researchers have investigated the simulation of FLCT2s. Finally, this thorough analysis of the existing literature demonstrated the variety of applications made possible and made it evident that FLCT2s consistently outperform their conventional counterparts in numerous applications. For the control of four benchmark linear and nonlinear plants, the authors [10] suggest an interval type-2 fractional order fuzzy PID IT2FO-FPID controller, which is an extension of fractional order calculus on FLCT2. Developing an FLCT2-PID controller using the proposed type-reduction method, which combines the conventional PID controller with the FLCT2 to improve the system performance compared with the FLCT1-PID controller, [3]. The speed of DCEM is controlled using FLCT2. DCEM is controlled using FLCT2 and CPTCT to indicate the dynamic performance of FLCT2, [4]. An optimal interval FLCT2 is designed for the speed control of DCEMs. An FLCT2 is implemented for the speed control of DCEM for the cases of series and shunt, [5].

In this work, the Concise semi-active ankle prosthetic CSAAP has been designed and developed using the CAD-CAM program. Mechanical design of the CSAAP model can simulate the motion of a non-amputee, but not exactly, because there is no control system. The CSAAP prototype is imported into ADAMS for dynamic analysis. The model of the CSAAP in ADAMS is then imported into MATLAB/Simulink for control system design. A DC motor is used to regulate the performance CSAAP model. The speed and current of the DC motor are regulated using two CPICTs. Then, for precision control, a Fuzzy logic controller type-1 is designed to tune online gains of PI-controllers (fuzzy logic control type-1-PI (FLCT1-PI) controller). And Fuzzy logic controller type-2 is designed to tune online gains of PI-controllers, fuzzy logic control type-2-PI (FLCT2-PI controller). A simulation model is designed to show the comparison between these controllers in MATLAB/Simulink. The paper is arranged as follows, section two displays the mechanical design of the CSAAP model, and section three explains the control design for the model. After that, the design of CPICT, FLCT1-PI, and FLCT2-PI controllers is presented. There is a comparison between the CPICT, FLCT1-PI, and FLCT2-PI. The results show that the FLCT2-PI controller has superior responses while the comparison to the CPICT, and FLCT1-PI controller. The final section displays the conclusions of this research.

2 Mechanical Design of the Ankle-Foot Prosthesis

The CSAAP model [11], [12], [13] consists of active and passive components that work together to generate total positive power at the PSAAFP joint during walking's stance stage. The Figure 1 system is displayed in a figure that was created with the help of a cad cam application. The CSAAP prototype has a carbon fiber foot, actuator, leaf series springs, and parallel springs with a cam profile mechanism in this design. The carbon fiber foot is utilized to attenuate shock during heel-strike. A DCEM, pulleys, a belt, and a screw bolt are the main components of the actuator. The CSAAP is actuated by a DCEM in response to the actuator's requests for torque, speed, and peak power. A pair of custom pulleys and a belt are used to transfer spin from the DCEM's output to the screw bolt. The DCEM's velocity is reduced by using the pulley-belt transmission. The pulley-belt transmission dissipates shock and vibrations from the outside world. Maximum accuracy and sensitivity are required in ankle joints, and a screw bolt nut is utilized to ensure that the artificial limbs continue to move effectively and efficiently. In terms of operation stability, dependability, and transmission efficiency, pulley drives utilized for rigid transmissions always perform well, [14]. In addition, the screw bolt nut transfers rotational to linear motion.

When the transmission fails to meet bandwidth requirements, the leaf series springs are used to increase the level of series elasticity, preventing the transmission from being destroyed during heel striking, [15], [16]. The nonlinear parallel springs system consists of a cam, a follower, and a spring element. The nonlinear parallel springs mechanism was created to simulate human ankle dorsiflexion stiffness, and the cam profile was designed in [15]. The cam shape has a specially designed ramp piece that allows the springs to be unloaded by twisting the ankle enough in the plantarflexion direction. On the device's top is a traditional pyramid adapter, which is used to connect the device to a patient's prosthetic socket. The CSAAP model is being developed. The model behaves like a non-amputee ankle after using the series and parallel springs and cam. There is a reduction in the amount of energy used. The PSAAFP model [17] meets the design requirements and intended angle-torque.

The CSAAP is modeled. The modeling of CSAAP consists of the motor, leaf series springs, and parallel springs. The leaf series springs are used to increase the level of series stiffness that adequately protects the transmission from damage during heel strike fails to satisfy bandwidth requirements.

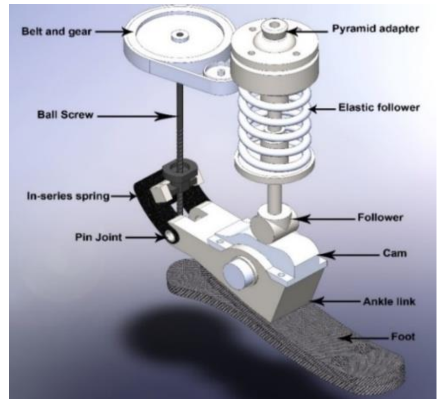

Fig. 1: The model of the CSAAP in CAD - CAM program

The parallel springs are used to reduce torque and power requirements to realize the reference torque $T_r\,(N.m)$, [18] the actuator provides the active torque T_{ac} (N.m),

$$T_{ac} = T_r - T_{kp}, \quad (1)$$

$T_{kp}\,(N.m)$ is the produced torque by the parallel spring. For linear passive springs, the torque T_{kp} while can be expressed in terms of the angular displacement for motorθ_{ac} and parallel spring's stiffness constant $K_p(N.m)$ as:

$$T_{kp} = -K_p\theta_{ac} - K_0, \quad (2)$$

where $K_0(N.m)$ is the preload of the springs and equal to 0 N.m. This expression can be generalized for a nonlinear nth degree polynomial spring to:

$$T_{kp} = -\sum_{i=0}^{m}\theta_{ac}^{i}\;K_{p_i} = -\Theta_m^T\,K_p, \quad (3)$$

where transpose is denoted by superscript T, Θ_m is a vector containing a geometric progression of $\theta_m\,(rad)$ up to power m:

$$\Theta_m = [\,1\;\;\theta_{ac}^{1}\;\theta_{ac}^{2}\;\;\dots\dots\dots\theta_{ac}^{m}]^T \quad (4)$$

and the vector K_p contains the $m+1$ nonlinear spring parameters:

$$K_p = \left[K_{p,0}\,,K_{p,1}\,,\dots\dots\dots,K_{p,n}\right]^T, \quad (5)$$

and the angular displacement of the motor can be determined from:

$$\theta_{ac} = \theta_r + T_r/K_s, \quad (6)$$

where $\theta_r(rad), K_s(N.m)$ are the required angular displacement and series springs stiffness constant respectively. To find the angular speed, the angular displacement must be differentiated with respect to time:

$$\dot{\theta_{ac}} = \dot{\theta_r} + \dot{T_r}/K_s, \quad (7)$$

Motor power is finally the product of motor torque T_{ac} and the derivative of the angular displacement:

$$P_{ac} = T_{ac}\dot{\theta_{ac}} \quad (watt),$$
$$P_{ac} = [T_r - T_{kp}][\dot{\theta_r} + \dot{T_r}/K_s], \tag{8}$$
$$P_{ac} = [T_r + \Theta_m^T K_p][\dot{\theta_r} + \dot{T_r}/K_s]$$

3 Control of Ankle-Foot Prosthetic

The control system [12], [13] for the PSAAFP was designed and implemented using MATLAB/SIMULINK and a DCEM. After the PSAAFP has been modelled using CAD-CAM, it must be regulated. A dynamic analysis must be done prior to control. Figure 2 depicts how the PSAAFP prototype is put into ADAMS for dynamic analysis. The most typical prototypes imported into ADAMS are ground, components, joints, and other prototypes. The prototype of the PSAAFP in ADAMS is imported into MATLAB/Simulink as shown in Figure 3, and the control simulation system's block diagrams are produced using the CPICT first, then the FLCT1-PI, and then the FLCT2-PI. The DCEM that is used in the CSAAP model, can be modelled by using the following equation, [19]:

$$V_{ac} = iR + L\frac{di}{dt} + E \tag{9}$$

where
V_{ac} is voltage applied in DCEM.
L is winding inductance.
R is winding Resistance.
E is back emf of the motor.

$$T_e = J\frac{d\omega}{dt} + B\omega + T_L \tag{10}$$

where
T_e is Electromagnetic torque in Nm.
J is Moment of inertia.
$\frac{d\omega}{dt}$ is Angular speed.
B is Friction coefficient.
T_L is Load torque.

4 PI-Controller Design

The CPICT [12], [13] are still the most widely used control structure and used to control the PSAAFP model. This is mainly because CPICT has simple control structures, affordable price, and effectiveness for linear systems. The CPICT 's primary role is to set a suitable proportional gain (KP) and integral gain (KI) for optimal control performance. CPICT parameters can be adjusted in a variety of ways; in this model, we'll apply the signal constraint, [3], [20].

The standard speed $S_r(t)$ is the beginning point for the CSAAP's control mechanism. The DCEM's feedback speed $S_m(t)$ is compared to the standard speed. The CPICT parameter is set to the difference between the reference and feedback speeds $e(t)$. This CPICT's output is reference current $u_r(t)$.

$$u_r(t) = k_p e(t) + k_i \int e(t)\, dt \tag{11}$$
$$e(t) = S_r(t) - S_m(t) \tag{12}$$

The DCEM's feedback current $u_m(t)$ is compared to the reference current. The difference in current between reference and feedback $ec(t)$ is incorporated into the second CPICT algorithm. The voltage $V(t)$ is the output of the second CPICT. This voltage is entered into the DCEM model. The DCEM's parameters can be found in [21]. Figure 4 shows a circuit of the control system of the virtual PSAAFP prototype produced in MATLAB/Simulink using the CPICT approach.

$$V(t) = k_p ec(t) + k_i \int ec(t)\, dt \tag{13}$$
$$ec(t) = u_r(t) - u_m(t) \tag{14}$$

Due to its linear structure, the CPICT is usually not effective if the system to be controlled has a high level of complexity, such as, time delay, high order, modeling nonlinearities, vague systems without precise mathematical models, and structural uncertainties. For these reasons, a CPICT is combined with a FLCT1 to achieve a better system performance over the CPICT. The FLCT1-PI is developed to improve the system performance rather than CPICT, [3].

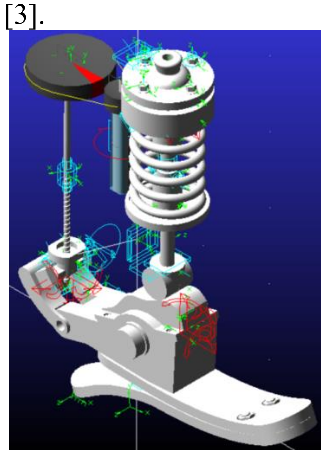

Fig. 2: the model of the CSAAP in the Adams program

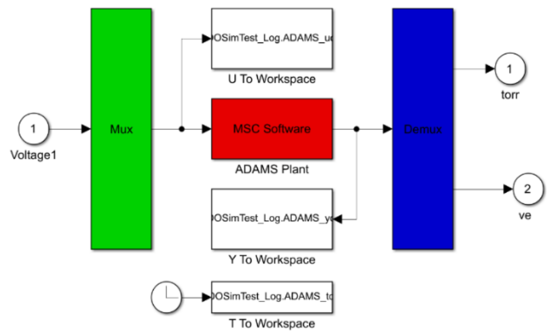

Fig. 3: Control plant generated by ADAMS

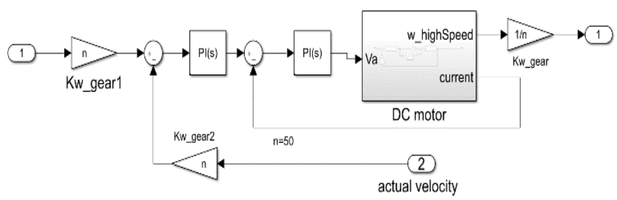

Fig. 4: the control system of CPICT algorithm

5 FLCT1-PI Controllers

The expression FLC [13] is done in human language. A FLC converts linguistic control strategies into FLC control strategies, and the fuzzy rules are constructed using the knowledge of an expert or a database. Figure 5 (Appendix) displays a MATLAB/Simulink of the FLCT1-PI. Two FLCT1-PIs are used in the control system. Two FLCT1-PIs are used in the control system. The variable inputs of the first FLCT1 are the error (e) and the change of the error (de) of the speed. The first FLCT1's variable output alters the CPICT's parameters, after which the CPICT determines the reference current. Second, the error (e) and the change of the error (de) of the current are the variable inputs of the second FLCT1. The variable output of the second FLCT1 is then processed using the CPICT to adjust the output voltage from the power source that supplies the DCEM. Due to their internal integration operation, FLCT1-PI controllers perform poorly for higher order processes during the transient period [3]. Despite the significant improvement in system performance with FLCT1s over their CPICT, it should also be noticed that they are usually not effective in those cases If the system that needs to be controlled contains system uncertainties because the normal FLCT1s can only handle a small amount of data uncertainty directly. The FLCT1's fundamental flaw is that it is unable to handle the linguistic and numerical uncertainties that are connected to dynamic, unstructured environments found in the actual world, [3], [10].

6 FLCT2 Systems

FLCT2 is generally applied to the system where the uncertainty is present in terms of nonlinearity between input and output. The FLCT2 can minimize the effect of uncertainties and disturbance, [10]. The FLCT2 sets are extremely helpful in situations when it is challenging to define an exact membership function for a fuzzy set since they can model such uncertainty due to the fact that their own membership functions are fuzzy. The general structure of FLCT2s is illustrated in Figure 6. As it is clear, the structure is almost like the structure of FLCT1s. Both the FLCT1 and FLCT2 have the same four components, which are a fuzzifier, a rule base, a fuzzy inference engine, and an output processor. Furthermore, unlike a FLCT1, the output processor generates a FLCT1output using the type-reducer or a crisp number using the defuzzifier. A FLCT2 also has IF–THEN rules, but type-2 sets make up its antecedent or consequent sets. When conditions are too ambiguous to precisely identify membership grades, FLCT2 can be utilized. They have various uses, notably in control systems. The FLCT2 is a special case of the FLCT2. They spread the uncertainty equally among all major memberships that are acceptable and are easier to work with than generic FLCT2. The FLCT2 has been applied to various fields with great success, [3], [9].

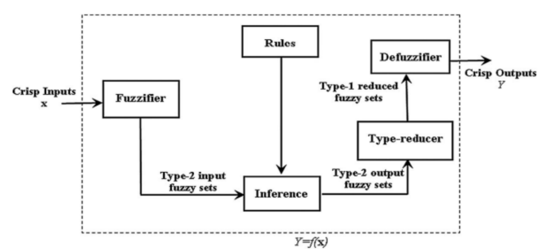

Fig. 6: General structure of FLCT2s

6.1 Structure of FLCT2

The controller structures of FLCT2 and FLCT1 are comparable. The defuzzification stage is the primary distinction between these controller topologies. Because the defuzzification process uses membership functions and because they differ structurally from one another, membership functions are divided into type-1 and type-2 categories. The type reduction procedure is one of the key characteristics that sets FLCT2 apart from FLCT1. Figure 7 also displays type-1 and type-2 triangular membership functions. The following is an expression for:

$$B = \left(x, \mu_B(x)\right)/\forall x \in X \tag{15}$$

where the membership level of the variable x in the B set, represented by $\mu_B(x)$ ranges from 0 to 1. Since each variable x has a membership level between 0 and 1, the uncertainty cannot be represented. Use of the type-2 membership function is necessary if the membership level of a variable is not fully known and cannot be calculated.

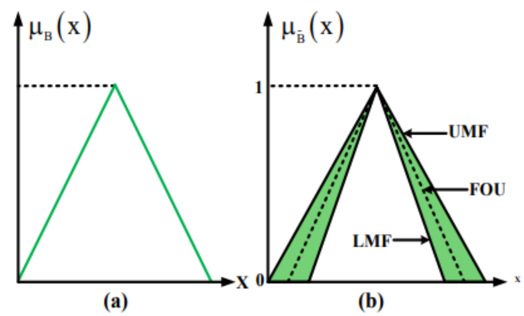

Fig. 7: Triangular membership functions a) FLCT1 b) FLCT2

The FLCT2 set seen in Figure 7(b) can be expressed by B, and the following equations can explain it.

$$\tilde{B} = \left\{(x,u), \left(\mu_{\tilde{B}}(x,u)\right)/\forall x \in X, \forall u \in J_x^u \subset [0,1]\right\} \tag{16}$$

$$\tilde{B} = \int_{x=X} \int_{u \in J_x} \mu_{\tilde{B}}(x,u)/(x,u)\, J_x \subseteq [0,1] \tag{17}$$

In this case, X is the input variable's domain, x is its value, u is an FLCT2 set's primary grade, J_x is its primary membership, and $\mu_{\tilde{B}}(x,u)$is its secondary membership function.

$$\tilde{B} = \int_{x=X} \int_{u \in J_x} 1/\mu_{\tilde{B}}(x,u)\, J_x \subseteq [0,1] \tag{18}$$

where $J_x \subseteq [0, 1]$ and ∬ denote the union of all valid x and u. The combination of all secondary sets after defuzzification can be represented as follows:

$$\tilde{B} = \int_{x=X} \left[\int_{u \in J_x} f_x(u)/u\right]/x\, J_x \subseteq [0,1] \tag{19}$$

6.2 Interval Type-2 Membership Functions

Upper and lower membership functions are the building blocks of an interval type-2 membership function. A classical type-1 membership function is comparable to the upper membership function (UMF). For all conceivable input values, the lower membership function (LMF) is smaller than or equal to the higher membership function. The footprint of uncertainty is the region between the UMF and LMF (FOU). The UMF, LMF, and FOU for a type-2 triangle membership function are depicted in the following diagram, Figure 7, respectively, [22]. For FOU [4], the following equation can be given.

$$FOU(\tilde{B}) = \cup_{\forall x \in X} J_x = \{(x,u): u \in J_x \subseteq [0,1]\} \tag{20}$$

The FOU of FLCT2 Figure 7 is bound using two type-1 membership functions. $\tilde{B}$ of UMF and LMF are shown by:

$$\bar{\mu}_{\tilde{B}}(x) = FOU(\tilde{B}), \forall x \in X, \tag{21}$$

$$\underline{\mu}_{\tilde{B}}(x) = FOU(\tilde{B}), \forall x \in X \tag{22}$$

Thus, J_x is an interval set, given by:

$$J_x = \left\{(x,u): u \in \left[\underline{\mu}_{\tilde{B}}(x), \bar{\mu}_{\tilde{B}}(x)\right]\right\} \tag{23}$$

So that FOU(Ã) in Eq. (20) can be given as:

$$FOU(\tilde{B}) = \cup_{\forall x \in X} \left[\underline{\mu}_{\tilde{B}}(x), \bar{\mu}_{\tilde{B}}(x)\right] \tag{24}$$

For continuous universe of discourse X, an embedded FLCT2 $\tilde{B}_e$ is:

$$\tilde{B}_e = \int_{x \in X} [1/u]/x, \; u \in J_x \tag{25}$$

Note that Eq. (25) means: $\tilde{B}_e : X \to \{u: 0 \le u \le 1\}$. The set $\tilde{B}_e$ is embedded in $\tilde{B}$ such that at each x it has only one secondary variable (i.e., one primary membership whose secondary grade equals 1). Examples of $\tilde{B}_e$ are $\bar{\mu}_{\tilde{B}}(x)$ and $1/\underline{\mu}_{\tilde{B}}(x), \forall x \in X$.

6.3 Type Reduction and Defuzzification

The aggregate FLCT2 set is first reduced to an interval FLCT1 set, which is a range with lower limit y_l and upper limit y_r, to determine the final crisp output value for the inference process. The centroid of the FLCT2 set is a frequent name for this interval FLCT1 set. The centroids of all the FLCT1 sets that are embedded in the FLCT2 set make up this centroid, in principle. Four built-in type-reduction techniques are supported by the Fuzzy Logic Toolbox program. The initialization strategies, presumptions, computational effectiveness, and terminating circumstances of these algorithms vary.

Crisp input and crisp output can be used to illustrate the input to output mapping, which is illustrated as $y = f(x)$, where function $f(\ )$ represents the fuzzy operations. You can identify this FLCT2 as FLCT2. The FLCT2 operates as follows: while only type-0 and type-1 fuzzification are feasible with FLCT1, type-0 and type-1 fuzzification is only possible with FLCT2. The rule base is represented, like in FLCT1, by a set of IF–THEN statements. Typically, an expert person or the results of a numerical analysis are used to supply the rule base. So, using N rules as a starting point, the generic rule structure of TSK FLCT2 is as follows:

R^n: IF x_1 is $\tilde{X}_1^n$ and….. x_i is $\tilde{X}_1^n$ ……and x_I is $\tilde{X}_I^n$,
THEN y is $\tilde{Y}^n$ $n = 1, 2, \ldots\ldots, \mathrm{N}$ (26)

In Eq. (26), $\tilde{X}_1^n\ (i = 1, 2, \ldots.I)$ are the antecedent membership functions *and* $\tilde{Y}^n = [\underline{y}^n, \overline{y}^n]$ are the consequent membership functions. This equation establishes a relationship between output space y ∈ Y and the input spaces, $x_1 \in X_1\ x \ldots.x\ x_I \in X_I$. Here $\underline{y}^n$ and $\overline{y}^n$ can be crisp consequences of linear functions as follow:

$$\underline{y}^n = \underline{a}_1^n x_1 + \cdots + \underline{a}_I^n x_I + \underline{b}^n \quad (27)$$

$$\overline{y}^n = \overline{a}_1^n x_1 + \cdots + \overline{a}_I^n x_I + \overline{b}^n \quad (28)$$

where the resultant coefficients for underline, $\underline{a}_1^n x_1 + \cdots + \underline{a}_I^n x_I + \underline{b}^n$ and $\overline{a}_1^n x_1 + \cdots + \overline{a}_I^n x_I + \overline{b}^n$ are the same (crisp numbers). The entire FLC is FLCT2 if either the antecedent or the consequent are modelled as FLCT2. In reality, the FLC is regarded as an FLCT2 if all of the antecedents and consequents are modelled as FLCT1s but at least one input is fuzzified as an FLCT2s. The inference engine maps the input FLCT2s to produce output FLCT2s using the rule basis.

For a crisp input *vector*$x' = (x_1' + x_2' + \cdots + x_I')$, the output calculation steps of the FLCT2 are as follow:

(1) The input FLCT2s can be defined in terms of membership interval of each x_i' on each X_i^n , i.e., $\left[\underline{\mu}_{\tilde{X}_i^n}(x_i'), \overline{\mu}_{\tilde{X}_i^n}(x_i')\right], i = 1,2, \ldots, I, n = 1,2, \ldots\ldots, N$

(2) The total firing interval of the n^th rule is:

$$F^n(x') = \left[\underline{f}^n(x'), \overline{f}^n(x')\right] \equiv \left[\underline{f}^n, \overline{f}^n\right], n = 1,2, \ldots, N \quad (29)$$

where both can be denoted as

$$\underline{f}^n(x') = \underline{\mu}_{\tilde{X}_1^n}(x_1') * \ldots.* \underline{\mu}_{\tilde{X}_I^n}(x_I') \quad (30)$$

$$\overline{f}^n(x') = \overline{\mu}_{\tilde{X}_1^n}(x_1') * \ldots.* \overline{\mu}_{\tilde{X}_I^n}(x_I') \quad (31)$$

where, product operation or minimum t-norm is denoted by *.

(3) TR and defuzzification are the two phases that make up the output processing block. The TR stage, which converts the fired rule consequent sets into FLCT1s, is crucial. There are several potential TR approaches. Each of these draw's inspiration from FLCT1's defuzzification techniques. $F^n(x')$ and the appropriate rule consequences are combined using the TR. The most popular technique is a centre of sets (COS) type reduction, which is provided by$Y_{cos}(x')$ as:

$$Y_{cos}(x') = \bigcup_{\substack{f^n \in f^n(x') \\ y^n \in y^n}} \frac{\sum_{n=1}^N y^n f^n}{\sum_{n=1}^N y^n} = [y_l, y_r] \quad (32)$$

In above Eq. (12), the two end points, y_l and y_r are defined as:

$$y_r = \frac{\sum_{n=1}^R \overline{y}^n \underline{f}^n + \sum_{n=R+1}^N \overline{y}^n \overline{f}^n}{\sum_{n=1}^R \underline{f}^n + \sum_{n=R+1}^N \overline{f}^n} \quad (33)$$

$$y_l = \frac{\sum_{n=1}^L \underline{y}^n \overline{f}^n + \sum_{n=L+1}^N \underline{y}^n \underline{f}^n}{\sum_{n=1}^L \overline{f}^n + \sum_{n=L+1}^N \underline{f}^n} \quad (34)$$

The iterative KM technique can be used to find the switching points in this case, which are L and R.

(4) The type of reduced set defuzzified to crisp output by only averaging the y_l and y_r points, i.e., crisp output, [3], [10].

$$\mathrm{y} = \frac{y_l + y_r}{2} \quad (35)$$

For the modified plant, the type-2 FLC exhibits more durable performance. A type-2 FLC's performance can be enhanced by employing a different: • FOU [22]; • Rule base; • Number of rules.

7 FLCT2-PI Controller

A superior control performance in terms of stability and robustness against system uncertainties and environmental disturbances is the main goal of the controller design. A straightforward upper-level intelligent controller and a lower-level classical controller make up the suggested control system. The development of the antecedent of fuzzy rules for control applications was typically preferred to the error (e) and modification of error (e) parameters. Therefore, the antecedent part of the rule-based in this proposed controller uses the error signal and the change in error signal. Typically, the CPICT is implemented as follows:

$$u(t) = k_p e(t) + k_i \int e(t)\, dt \tag{36}$$

$$e(t) = y_r(t) - y_m(t) \tag{37}$$

where k_p, k_i, are the proportional, and the integral, gains respectively. The letters u, ym, and yr, respectively, stand for the controller output, the process output, and the set point. Signal constraint is used in the CPICT to modify the values of k p and k i in Eq. (30). The performance of the traditional controller will be improved over a broad operating range by creating a rule based FLCT2 framework that allows these parameters to be changed online in response to changes in the process state without much operator involvement. The way the FLCT2 operates is as follows: the input interval FLCT2 sets are first created by fuzzifying the crisp input from the input sensors (FLCT2s). The inference engine and rule base are activated by the input FLCT2s to create the output FLCT2s. The FLCT2 rules will not change from the FLCT1, but the FLCT2s will represent the antecedents and/or consequences. A mapping from input FLCT2s to output FLCT2s is provided by the inference engine, which combines the fired rules. The kind of reducer then processes the FLCT2 outputs of the inference engine, combining the output sets and performing a centroid calculation to produce FLCT2s known as the type-reduced sets. The type-reduced sets (or approximate type-reduced sets) are defuzzified (by taking the average of the type-reduced or approximate type-reduced set) after the type reduction procedure to provide crisp outputs [3]. A MATLAB/Simulink of the FLCT2-PI control is shown in Figure 8. The error (e) and variation of the error (e) of the speed are the initial FLCT2's variable inputs. The variable output of the first FLCT2 adjusts the KP of the speed CPICT.

Figure 9, Figure 10 and Figure 11 show the initial FLCT2 and input variable memberships. For each input, five membership functions are chosen. One output as displayed in Figure 12 and five members made up the proportional gain. Negative Large (NLU), Negative Small (NSU), Zero (ZU), Positive Small (PSU), and Positive Large are the members of the inputs and outputs, respectively (PLU). Each input has five fuzzy memberships, thus, there are a total of 25 fuzzy rules in use. Table 1 shows these regulations for the first FLCT2. The error (e) and the change of the error (e) of the speed are the second FLCT2's variable inputs. The variable output of the second FLCT2 adjusts the KI of the speed CPICT. Figure 13, Figure 14 and Figure 15 display the second FLCT2 and input variable memberships. For each input, five membership functions are chosen. Integration gain had five memberships, and there is only one output that is displayed in Figure 16. Each input has five fuzzy memberships, thus, there are a total of 25 fuzzy rules in use. Table 2 lists these regulations. The CPICT then determines the reference motor current.

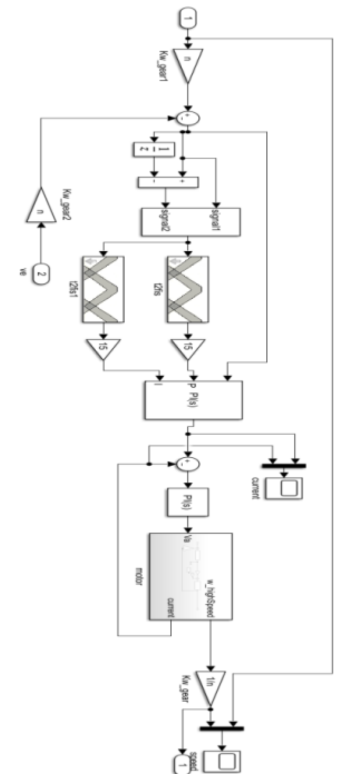

Fig. 8: The block diagram of the FLCT2-PI control

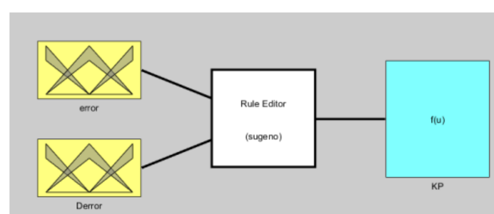

Fig. 9: Block FLCT2 for the first FLCT2

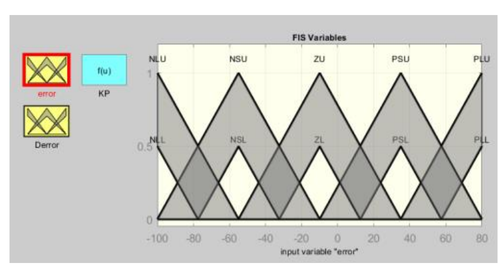

Fig. 10: Membership function of the speed error for the first FLCT2

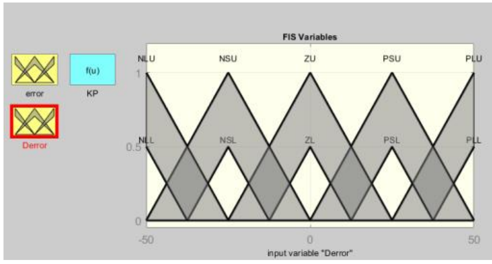

Fig. 11: Membership function of the change in the speed error for the first FLCT2

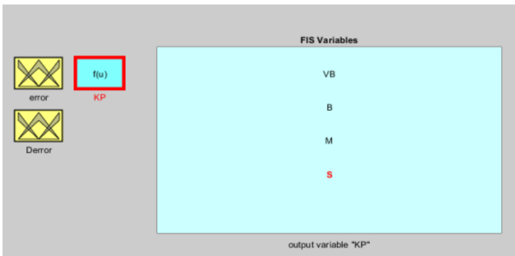

Fig. 12: Membership function of K_p of speed CPICT for the first FLCT2

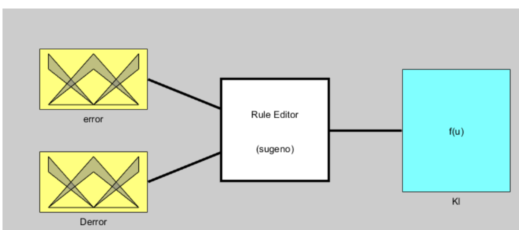

Fig. 13: Block FLCT2 for the second FLCT2

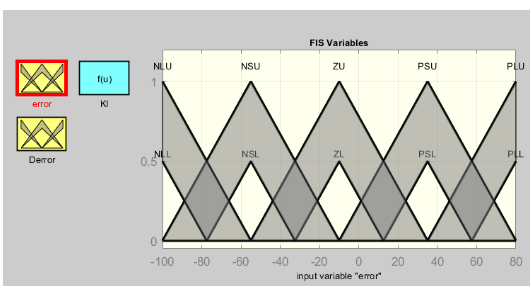

Fig. 14: Membership function of the speed error for the second FLCT2

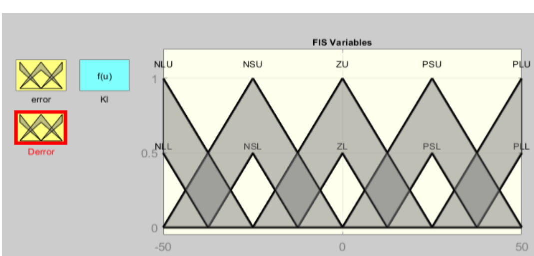

Fig. 15: Membership function of the change in the speed error for the second FLCT2

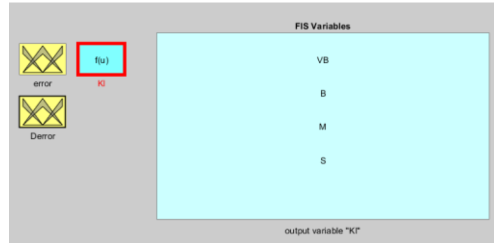

Fig. 16: Membership function of K_I of speed CPICT for the second FLCT2

Table 1. The rules applied in the CPICT for K_p.

De / e	NL	NS	Z	PS	PL
NL	M	B	VB	B	M
NS	S	M	B	M	S
Z	S	S	M	S	S
PS	S	M	B	M	S
PL	M	B	VB	B	M

Table 2. The rules applied in the CPICT for K_i.

De / e	NL	NS	Z	PS	PL
NL	B	B	VB	B	B
NS	M	M	B	M	M
Z	S	M	M	M	S
PS	M	M	B	M	M
PL	B	B	VB	B	B

8 Results and Discussion

This section compares the effectiveness of the CPICT, FLCT1-PI, and FLCT2-PI controllers. The reference speed and the DCEM's output speed while utilizing CPICT with fixed gains are depicted. When the FLCT1-PI and FLCT2-PI controller algorithms are employed. Figure 17 displays the response system of CPICT (Red), FLCT1-PI (Blue), and FLCT2-PI (Gray) controllers. The output of the first CPICT (Black) and the output current of the DCEM (Red) utilizing the CPICT with fixed gains, the output of the first FLCT1-PI (Green) and the output current of the DCEM (Blue), and the output of the first FLCT2-PI (burgundy) and the output current of the DCEM (Gray) are displayed in Figure 18. The findings indicate that FLCT1-PI performs marginally better than the single CPICT algorithm. Two loops are present in the system control. Every loop in the FLCT1-PI contains FLCT1-PI, whereas only one loop in the FLCT2-PI contains FLCT2-PI. As a result, FLCT2-PI's reaction system is a little better than FLCT1-PI's response system.

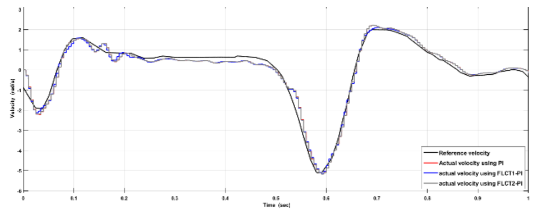

Fig. 17: System response using CPICT, FLCT1-PI, and FLCT2-PI

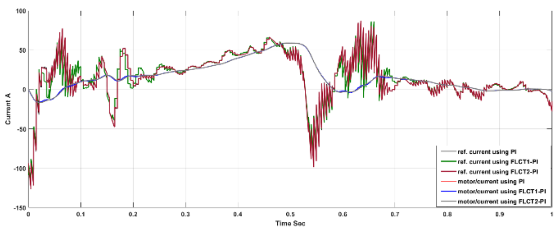

Fig. 18: Current of motor response using CPICT, FLCT1-PI, and FLCT2-PI

The PSAAP system is not constant. The weight of the amputee may be increased, or the amputee carries anything, so the load may be increased. The parameters of the DCEM may also be increased. In practice, the resistance and inductance of DCEM may be increased with increasing temperature. In the simulation, the load and the parameters of DCEM are increased by 20% and 50%. Figure 19 the reference speed (Black), the output speed of model using CPICT (Red) with fixed gains, the output speed of the model when FLCT1-PI (Blue), and the output speed of the model when FLCT2-PI (Gray) when the load and the parameters of DC motor are increased with 20%. Figure 20 the output of the first CPICT (black), the output current of the DCEM using CPICT (Red) with fixed gains, the output of the first FLCT1-PI (Green), and the output current of the DCEM when FLCT1-PI (Blue), and the output of the first FLCT2-PI (Orange), and the output current of the DCEM when FLCT2-PI (Gray), when the load and the parameters of DC motor are increased with 20%. Figure 21 shows the reference speed (Black), the output speed of model using CPICT (Red) with fixed gains, the output speed of the model when FLCT1-PI (Blue), and the output speed of the model when FLCT2-PI (Gray) when the load and the parameters of DCEM are increased with 50%. Figure 22 shows the output of the first CPICT (Black), the output current of the DCEM using (Red) with fixed gains, the output of the first FLCT1-PI (Green), and the output current of the DCEM when FLCT1-PI (Blue), and the output of the first FLCT2-PI (burgundy), and the output current of the DCEM when FLCT2-PI (Gray), when the load and the parameters of DCEM are increased with 50%. The results show that FLCT1-PI and FLCT2-PI controller algorithms can adjust the system when the load and the parameters of the DC motor are increased by 20% and 50%. The FLCT1-PI controller algorithm may be better than the FLCT2-PI controller algorithm because FLCT1 is applied to tune the parameters of two CPICT and FLCT2-PI is applied to tune the parameter of one CPICT.

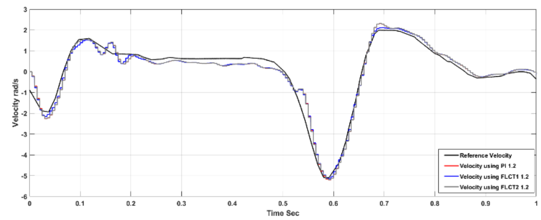

Fig. 19: System response using CPICT, FLCT1-PI, and FLCT2-PI when the parameters increase by 20%

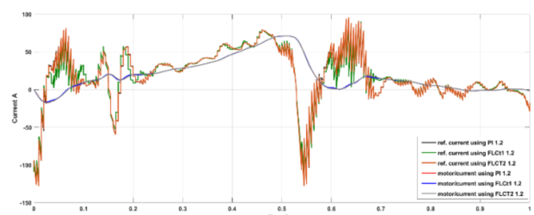

Fig. 20: Current of motor response using CPICT, FLCT1-PI, and FLCT2-PI when the parameters increase by 20%

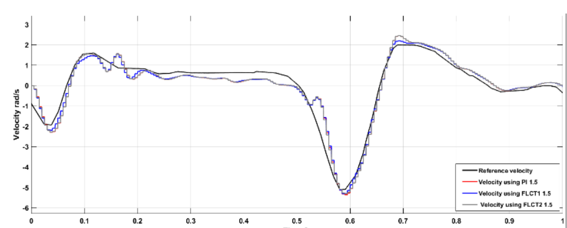

Fig. 21: System response using CPICT, FLCT1-PI, and FLCT2-PI when the parameters increase by 50%

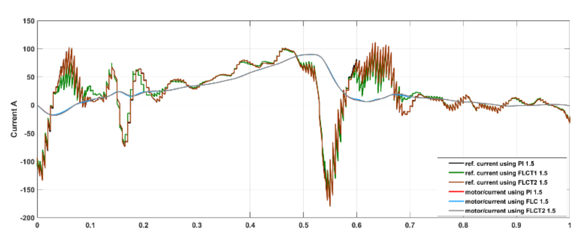

Fig. 22: Current of motor response using CPICT, FLCT1-PI, and FLCT2-PI when the parameters increase by 50%

9 Conclusions

The current CAD computer technologies give powerful, easy-to-use functionality that automates tasks and helps quickly define and validate the form, fit, and function of prosthetic design. In this paper, the mechanical design of the novel CSAAP is studied and reviewed. It is like the lower limb joint of the human body, and it can better meet the humanoid requirements of artificial legs. The virtual prototype of CSAAP is simulated based on the CPICT. Simulation results show that the CSAAP can follow the target trajectory well. This paper presents a FLCT1-PI and FLCT2-PI control approach of the human lower limb. The proposed strategy achieved good tracking performance compared to a healthy limb joint displacement, even in the presence of perturbation. Comparisons are made for the system with CPICT, FLCT1-PI, and FLCT2-PI controllers. The performance of the transtibial prosthetic limb using these controllers is also shown. The results show that implementing FLCT2-PI and FLCT1-PI controllers in the system gives better performance. The results reveal that a CSAAP with a FLCT2-PI and FLCT1-PI controller method outperforms a PICT alone. And in simulation, the load and the parameters of the DC motor are increased by 20% and 50%. The results show that FLCT2-PI and FLCT1-PI controller algorithms can be adjusted when the load and the parameters of the DC motor are increased by 20% and 50%. **In the future**, the FLCT2-PI control needs to be developed, and this model will be applied practically.

References:

[1] Minsu Chang, Kyoungsoon Kim, and Doyoung Jeon, "Research on Terrain Identification of the Smart Prosthetic Ankle by Fuzzy Logic", *IEEE Transactions on Neural Systems and Rehabilitation Engineering,* doi: 10.1109/TNSRE.2019.2933874, 2019.

[2] Kaiyang Yin, Muye Pang, Kui Xiang, Jing Chen, and Shenpei Zhou, "Fuzzy Iterative Learning Control Strategy for Powered Ankle Prosthesis", *International Journal of Intelligent Robotics and Applications,* Vol. 2, pp.122-131, 2018. https://doi.org/10.1007/s41315-018-0047-9.

[3] Mohammad El-Bardini, Ahmad M. El-Nagar, "Interval type-2 fuzzy PID controller for uncertain nonlinear inverted pendulum system", *ISA Transactions,* Vol. 53, pp.732-743, 2014. http://dx.doi.org/10.1016/j.isatra.2014.02.007.

[4] Hakan Acikgoz, "Speed Control of DC Motor Using Interval Type-2 Fuzzy Logic Controller", *International Journal of Intelligent Systems and Applications in Engineering,* Vol. 6(3), pp.197-202, 2018.

[5] Hossein Hassani and Jafar Zarei, "Interval Type-2 fuzzy logic controller design for the speed control of DC motors", *Systems Science & Control Engineering,* Vol. 3, pp. 266–273, 2015. http://dx.doi.org/10.1080/21642583.2015.1013644.

[6] Kumbasar T, Hagras H. Big bang-big crunch optimization based interval type-2 fuzzy PID cascade controller design strategy. *Inf. Sci.*, 2014, vol.282, pp.277-295. https://doi.org/10.1016/j.ins.2014.06.005.

[7] Kumar A, Kumar V. Evolving an interval type-2 fuzzy PID controller for the redundant robotic manipulator. *Expert Syst. Appl.,* 2016, vol.73, pp.161-177.

[8] Kumar A, Panda MK, Kundu S, Kumar V. Designing of an interval type-2 fuzzy logic controller for magnetic levitation system with reduced rule base. *Comput. Commun. Netw Technol.,* India 2012, pp.1–8.

[9] Panda MK, Pillai GN, Kumar V. Design of an interval type-2 fuzzy logic controller for automatic voltage regulator system. *Electr. Power Compon. Syst.*, 2011, vol.40, pp.219-235.

[10] Kumar A, Kumar V., "A novel interval type-2 fractional order fuzzy PID controller: Design, performance evaluation, and its optimal time domain tuning", *ISA Transactions*, 2017. http://dx.doi.org/10.1016/j.isatra.2017.03.022i.

[11] Mariem Y. William, Khalil Ibrahim, A. A. Hassan, E. G. Shehata, "Development and control of a semi active prosthetic ankle", *Journal of Advanced Engineering,* Vol. 42, No. 2, 2023.

[12] E. G. Shehata, Mariem Y. William, A. A. Hassan, Khalil Ibrahim, "Development and Control of Concise Semi-Active Ankle Prosthesis", *WSEAS Transactions on Systems and Control,* Vol. 18, pp.108-120, https://doi.org/10.37394/23203.2023.18.12.

[13] E. G. Shehata, Mariem Y. William, A. A. Hassan, Khalil Ibrahim, "Development of Powered Semi-Active Ankle-Foot Prosthetic with Fuzzy Logic-PI Controller", *Journal of Engineering Sciences*, Vol. 52, No. 1, Jan. 2024. doi: 10.21608/JESAUN.2023.234785.1257.

[14] Louis Flynn, Joost Geeroms, Rene Jimenez-Fabian, Sophie Heins, Bram Vanderborght, Marko Munih, Raffaele Molino Lova, Nicola Vitiello and Dirk Lefeber, "The Challenges and Achievements of Experimental Implementation of an Active Transfemoral Prosthesis Based on Biological Quasi-Stiffness: The CYBERLEGs Beta-Prosthesis", *Frontiers in Neurorobotics*, doi: 10.3389/fnbot.2018.00080, Dec 2018.

[15] Jonathan Realmuto, Glenn Klute, and Santosh Devasia, "Nonlinear Passive Cam-Based Springs for Powered Ankle Prostheses", *Journal of Medical Devices*, doi: 10.1115/1.4028653, 2015.

[16] Samuel K. Au, Hugh M. Herr, "Powered Ankle-Foot Prosthesis", *IEEE Robotics, Automation Magazine*, pp.52-59, 2008.

[17] Samuel K. Au, Jeff Weber, Hugh Herr, "Powered Ankle–Foot Prosthesis Improves Walking Metabolic Economy", *IEEE Transactions on Robotics*, Vol. 25, No.1, 2009.

[18] Grimmer M., Eslamy M., Gliech S., and Seyfarth A., "A Comparison of Parallel and Series Elastic Elements in An Actuator for Mimicking Human Ankle Joint in Walking and Running", *IEEE International Conference on Robotics and Automation*, pp.2463–2470, 2012.

[19] Hari Maghfiroh, Musyaffa' Ahmad, Agus Ramelan, and Feri Adriyanto, "Fuzzy-PID in BLDC Motor Speed Control Using MATLAB/Simulink", *Journal of Robotics and Control (JRC)*, Vol. 3, Issue. 1, January 2022, doi: 10.18196/jrc.v3i1.10964.

[20] Nassim Messaadi, and Abdelkader Amroun, "*Speed Control of DC Motor Using Fuzzy PID Controller*", Senior thesis, Department of Instrumentation and Automation, University of Science and Technology Houari Boumediene, 2021.

[21] Anupam Kumar n, Vijay Kumar, "A novel interval type-2 fractional order fuzzy PID controller: Design, performance evaluation, and its optimal time domain tuning", *ISA Transactions*, Vol.68, May 2017, pp.251-275. https://doi.org/10.1016/j.isatra.2017.03.022.

[22] Fuzzy Logic Toolbox™ User's Guide © COPYRIGHT 1995–2022 by The MathWorks, Inc., [Online]. https://www.mathworks.com/help/fuzzy/index.html (Accessed Date: October 8, 2024).

Contribution of Individual Authors to the Creation of a Scientific Article (Ghostwriting Policy)

The authors equally contributed in the present research, at all stages from the formulation of the problem to the final findings and solution.

Sources of Funding for Research Presented in a Scientific Article or Scientific Article Itself

No funding was received for conducting this study.

Conflict of Interest

The authors have no conflicts of interest to declare.

APPENDIX

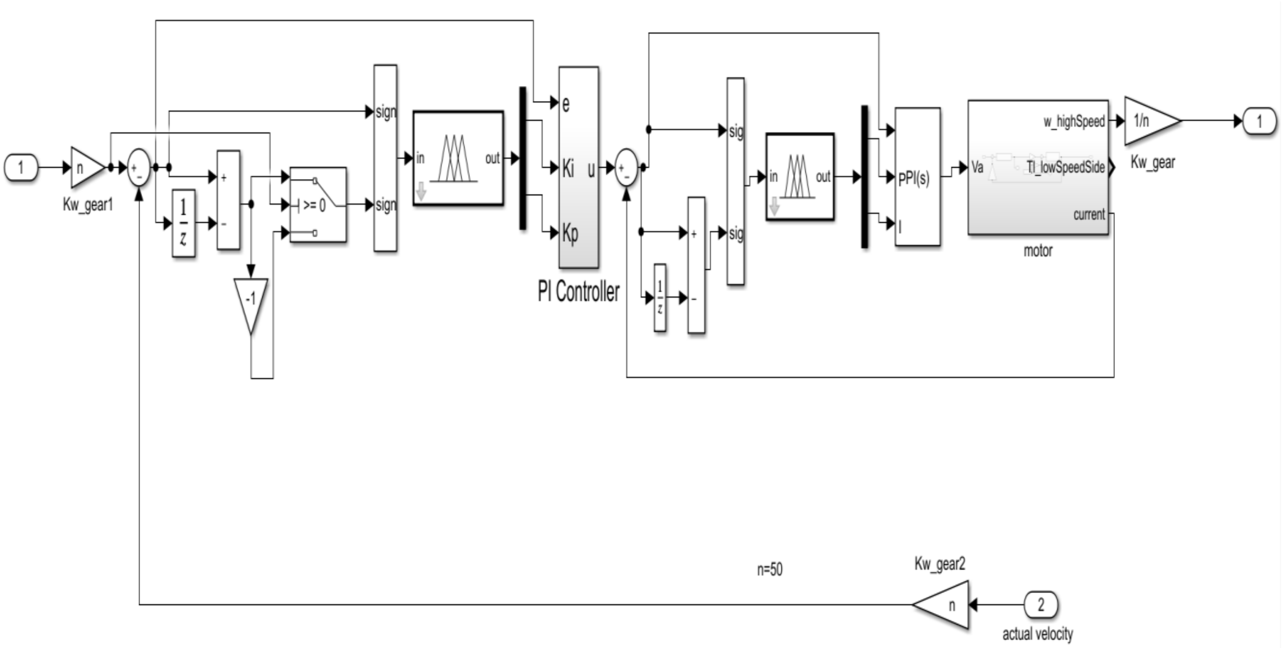

Fig. 5: The control system of the FLC1-PI control