IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received July 8, 2021, accepted July 27, 2021, date of publication July 30, 2021, date of current version August 6, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3101576

Offline ANN-PID Controller Tuning on a Multi-Joints Lower Limb Exoskeleton for Gait Rehabilitation

KARRAR H. AL-WAELI[1], RIZAUDDIN RAMLI[1], SALLEHUDDIN MOHAMED HARIS[1], (Member, IEEE), ZULIANI BINTI ZULKOFFLI[2], AND MOHAMMAD SOLEIMANI AMIRI[1]

[1]Department of Mechanical and Manufacturing Engineering, Faculty of Engineering and Built Environment, Universiti Kebangsaan Malaysia, Bangi 43600, Malaysia
[2]Department of Mechanical and Mechatronic Engineering, UCSI University, Kuala Lumpur 56000, Malaysia

Corresponding author: Rizauddin Ramli (rizauddin@ukm.edu.my)

This work was supported in part by the Ministry of Education under Grant ERGS/1/2012/TK01/UKM/02/2, and in part by Universiti Kebangsaan Malaysia under Grant DPK-2020-015.

ABSTRACT This paper presents Artificial Neural Network (ANN) as an optimization tool in tuning Proportional-Integral-Derivative (PID) controller's gain of a multi-joints Lower Limb Exoskeleton (LLE) for gait rehabilitation. The interest in wearable post-stroke and spinal cord injury rehabilitation devices such as LLE has been increasing due to the demand for assistive technologies for paralyze patients and to meet the concerns in the increasing number of ageing society. The dynamic of three degree of freedom LLE was determined using Euler-Lagrange equation, and PID parameters were initially tuned using the Ziegler-Nichols (ZN) method. The paper compares different ANN-based algorithms in tuning PID controller's gain for LLE applications. The method compared and evaluated with other methods and dynamic systems in the literature. ANN-based algorithms, Gradient Descent, Levenberg-Marquardt, and Scaled Conjugate Gradient, are utilized for PID tuning of each joint in the LLE model. The result shows faster convergence and improves step response characteristics for each controlled joint model. The overshoot values found to be 0.3126%, 0.6335%, and 0.2619% compared to the ZN method with 10.5582%, 15.1643%, and 11.8511% for hip, knee, and ankle joints, respectively. It can be ascertained that the PID controlled of LLE has been optimally tuned significantly by different ANN methods, which reduced its steady-state errors.

INDEX TERMS Proportional-integral-derivative, Lower Limb Exoskeleton, Artificial Neural Network, Euler-Lagrange.

I. INTRODUCTION

In recent decades, there has been an increase in individuals suffering from stroke and spinal cord injury. Accordingly, the demands for Lower Limb Exoskeleton (LLE) have increased substantially. LLE is a wearable robot, which augments the angular motion of a human's legs. The structure has two essential parts, mechanical and electrical parts. The mechanical part contains links and joints. In contrast, the electrical part represents the controller, actuators and sensors [1]–[3].

The advantage of LLE over other mechanisms such as wheelchairs, canes, crutches, knee walkers and seated scooters is the comfortability during mobility. However, the performance of LLEs is deficient and needs further improvement due to the complexity of the nonlinear behaviour in real-world applications.

Proportional-integral-derivative (PID) controller is the most crucial controller system used in industry due to its effectiveness in many different conditions [4], [5]. Also, the optimization methods have been developed to enhance the performance of control systems against nonlinear behaviours. Many different optimization methods are known today for their effectiveness in finding the most suitable solution to a problem, such as Artificial Neural Network (ANN) [6], Genetic Algorithm (GA) [7], Fuzzy Logic (FL) [8], Particle Swarm Optimization (PSO) [9] and others. ANN, which has many architectures such as Single-Layer Perceptron (SLP), Multi-Layer Perceptron (MLP), Support Vector

The associate editor coordinating the review of this manuscript and approving it for publication was Yongming Li.

Machine (SVM), Self-Organizing Feature Map (SOFM), and others, is considered one of the essential computing methods, which inspired by a basic unit of a biological brain of the neuron [10], [11].

Many researchers investigated LLE controllers to improve performance. Asl *et al.* [12] proposed a novel neural output controller for feedback trajectory tracking of n-link robotic exoskeletons. The controller is developed by defining auxiliary dynamics and utilizing an adaptive FF-ANN to compensate for the unknown nonlinear dynamics of the system. The performance of the controller was validated through simulations and experiments conducted on LLE. The study shows the neural network of the controller has assist-as-needed property, which contributed to the decrement of output when the user follow the desired trajectory in a rehabilitation task. Pongfai and Assawinchaichote [13] presented self-tuning of PID parameters using Artificial Intelligence (AI) algorithm for tuning the DC motor. The proposed approach combines the Neural Network and Genetic Algorithm (NN-GA). Also, the simulation results were analyzed by examining the transient responses and comparing the performance of self-tuning with NN and GA. Based on the simulation results, GA gives higher overshoot values, steady-state error, rise time and settling time than NN and NN-GA, with 21.708%, 0.04%, 1088 ms and 2802 ms, accordingly.

In addition, NN gives higher overshoot values, rise time and settling time than NN-GA, with 7.587%, 440 ms and 1629 ms, accordingly. NN-GA impressively gives the most improved transient response according to the criteria, with 6.879% overshoot, 0.020% steady-state error, 405 ms rise time and 1540 ms settling time. Lin and Liu [14] proposed an adaptive GA based self-tuning PID controller and compared traditional optimization methods. They claimed that the proposed controller is simple in structure and its computational task is smaller, so that the online adaptation is considered easy. Based on the simulation results, ZN gives higher peak amplitude values, overshoot, rise time and settling time than classical GA and adaptive GA, with 1.72, 72.000%, 0.41 s and 2.70 s, respectively. Ultimately, adaptive GA gives the most improved transient response according to peak amplitude, overshoot, rise time and settling time, with 1.06, 6.000%, 0.28 s and 0.75 s, respectively. The parameters obtained using adaptive GA provides better performance than those obtained by the ZN and classical GA methods.

Chiha *et al.* [15] developed a tuning method for PID parameters using multi-objectives Ant Colony Optimization (ACO). The design objective was to apply the multi-objectives ACO algorithm to tune the optimum solution of the PID parameters by minimizing the objective function. In all tested cases, the value of the maximum overshoot is quite small, nearly zero per cent and the values of the settling time, rise time, and integral of the squared error obtained by the multi-objective ACO method were less than those values by the other methods. Simulation results demonstrate that the tuning method using multi-objective ACO better controls system performance than the classic approach and the genetic algorithms.

Amiri *et al.* [16] proposed a method of tuning a PID controller for a 4-DoF LLE using a hybrid of GA and PSO. In the control system, the hybrid algorithm was applied to acquire the controller's parameters for each joint to minimize the trajectory error. In addition, a three-dimension model of the LLE was simulated to validate the proposed controller. They compared the results of the GA-PSO with pure GA and pure PSO based on statistics and found that the average error was minimized.

Wu *et al.* [17] aimed to assist stroke patients with a 3-DoF Lower Limb Rehabilitation Robot (LLRR) controlled by a robust adaptive sub-controller synthesized to make each joint track its designed training trajectory and overcome system uncertainties and reject the disturbances from the patient. Both joint and gait training were implemented to test controller performance with/without human reaction and the experiments show LLRR has better capability of trajectory tracking with less error.

Tanyildizi *et al.* [18] studied a new active PID control for 2-DoF LLE and a feedback controller using pressure-force sensors. They used a double-pendulum mechanism to demonstrate the human amplifier robot, which assists the healthy human low body movements. From the simulation results obtained for three motion scenarios and interpreted graphically, the LLE follow the human leg under different load conditions and successfully carry the loads without transferring them to the leg muscles. Gilbert *et al.* [19] created a PID controller according to the characteristics of the object model design of an LLRR and validated it using MATLAB/Simulink. The simulation results showed that LLE joints could track the desired joint ideal track position, proving the structure and the controller design are reasonable, with some guidance on rehabilitation for the LLRR. Uchegbu *et al.* [20] proposed a nonlinear control of stochastic differential equations to ANN matching; the model was validated, evaluated and compared with other existing controllers. They found good results by remodelling the PID controller with the neural network, which shows that the process industries' output loss drastically reduced to a minimum. Table 1 illustrates a literature survey of ANN in low-level controllers for different applications. The focus of this work is the optimization of low-level controllers for the LLE application.

This paper presents a comparison of ANN-based algorithms and manual-tuning of PID controllers. In self-tuning of the PID controller, the methods used the conventional ANN; this method does not manually tune the PID controller because there is no correlation between input data and target data. The current study in the research regarding the issue considered the inputs of ANN as step response characteristics of the closed-loop system, such as rise time (s), settling time (s) and overshoot (%). The proposed method gives more improved PID parameters, which later are used as initial values for adaptive controllers to obtain a better result and fast convergence.

In addition, the ANN-based algorithms, Gradient Descent (GD), Scaled Conjugate Gradient (SCG), and

TABLE 1. A literature survey of ANN in low-level controllers for different applications.

No	Reference	Year	Application	Controller	Type of tuning	Inputs to ANN
1	[21]	2012	Process control (liquid level)	PID-ANN	Self-tuning	Error signal
2	[22]	2012	Process control	PID-ANN-DEA	Self-tuning	Deviation, cumulation of deviation and variety of deviation
3	[23]	2013	Control of DC motor	PID-ANN	Self-tuning	Desired angle and actual angle
4	[8]	2014	Process control (continuous stirred tank reactor)	PID-ANN	Self-tuning	Proportional error, integral error and derivative error
5	[24]	2016	Control of underwater vehicles	PID-ANN	Self-tuning	Desired trajectories, actual trajectories, control signals and error signal
6	[20]	2016	Process control	PID-ANN	Self-tuning	Error signal and control signal
7	[13]	2017	Control of Brush DC motor	PID-NN-GA	Self-tuning	Angular speed deviation, voltage, and actual angular speed
8	[25]	2017	Passive optical networks	Adaptive-PID-ANN	Self-tuning	Three Discrete time errors
9	[12]	2018	Control of LLE (2DoF)	Adaptive-FF-ANN	Self-tuning	Angular trajectory
10	[6]	2019	Flight control of quadcopter	PID-ANN	Self-tuning	Reference signal, plant output and tracking error
11	Current study	2021	Control of LLE (3DoF)	PID-ANN	Manual-tuning	Rise time, settling time and overshoot

Self-tuning denotes that the controller parameters are continuously updated based on the measured error due to disturbances during system operation.
Manual-tuning denotes the controller parameters tuned before system operation, so they never change during system operation.

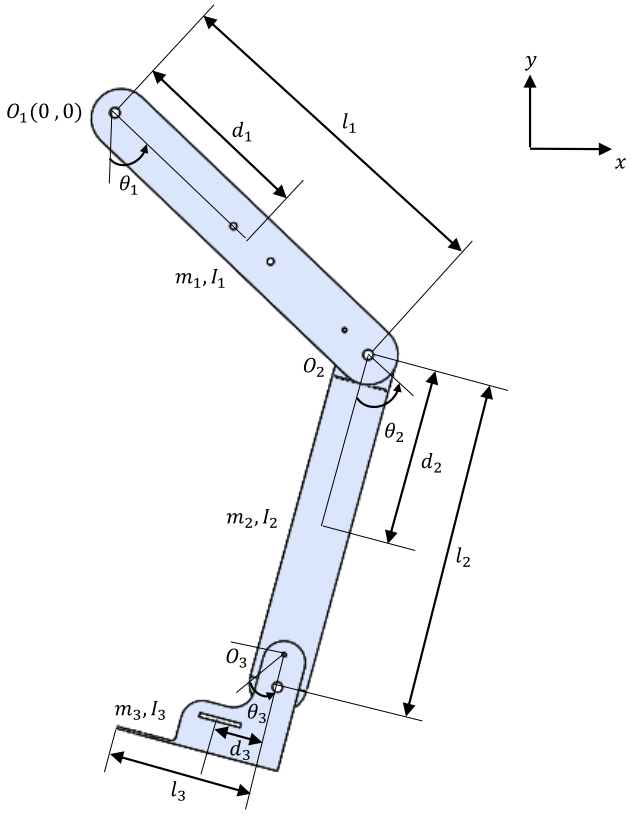


FIGURE 1. Physical diagram of proposed LLE model.

Levenberg-Marquardt (LM), are presented and compared. Other comparisons and experimental validation studies among the proposed and other methods and dynamic systems in the literature presented and compared.

II. MATHEMATICAL MODELING

A. MECHANICAL MODEL

Figure 1 illustrates a physical diagram of the proposed LLE model. The model accounted for the mass, rotational inertia and length of each link. The Range of Motion (ROM) of each actuated joint is limited for safety reasons due to the limitations in human's joints.

TABLE 2. Parameters of LLE.

Parameter	$i=1$	$i=2$	$i=3$
$m_i(kg)$	0.7955	0.5813	0.3158
$l_i(m)$	0.410	0.403	0.095
$d_i(m)$	0.1815	0.1416	0.0647
$I_i(kg.m^2)$	0.0421	0.0240	0.0032

Different methods can determine the dynamic equation of a multi-rigid-body system, such as Euler-Lagrange, Newton-Euler, and inverse pendulum. Among the methods, Euler-Lagrange equation is an energy-based method widely used to obtain the equations of motion of the LLE [17] shown as follows,

$$L = E_k - E_p \tag{1}$$

where E_k and E_p are the kinetic energy and potential energy, respectively. Now, integrating equation 1 to solve the following time-dependent partial differential equation, which is known as torque equation [26],

$$\tau_i = \frac{d}{dt}\left(\frac{\partial L}{\partial \dot{\theta}_i}\right) - \left(\frac{\partial L}{\partial \theta_i}\right) \tag{2}$$

where θ_i and $\dot{\theta}_i$ is the angle and angular velocity of each link in LLE. From equation 2, the state-space form of the dynamic model of LLE in a longitudinal plane is given by,

$$T_{ln} = M(\theta)\ddot{\theta} + C(\theta, \dot{\theta}) + G(\theta) \tag{3}$$

where T_{ln} is a 3-by-1 link torque vector; $M(\theta)$ is a 3-by-3 inertia matrix; $\ddot{\theta}$ is a 3-by-1 angular acceleration vector; $C(\theta, \dot{\theta})$ is a 3-by-1 centrifugal and Coriolis force vector; $G(\theta)$ is a 3-by-1 gravity vector. Table 2 represent the physical parameters of each link.

B. ELECTRIC MODEL

The LLE joint is actuated by DC motor which is an electromechanical machine that converts electrical energy into mechanical energy by taking direct currents to generate magnetic

fields, which power the movement of a rotor fixed inward a solid shaft. The generated torque and speed depend upon both the electrical input and the design of the motor. The mechanical part of the DC motor is represented by the rotary shaft, whereas the electrical part represented by resistance and inductance armatures. By applying Kirchhoff's voltage law [26], the description of the LLE model would be more realistic. However, the modelling of the DC motor is facilitated by neglecting the value of inductance since it is near zero [27], [28]. Accordingly, the electrical part of the DC motor is modelled by applying Ohm's law as follows,

$$U = iR \tag{4}$$

where U, R, and i are input voltage, resistance, and DC motor current, respectively. The output torque which rotates the shaft is proportioned to the current as follows,

$$T_m = K_m i \tag{5}$$

where K_m is the torque sensitivity. The dynamic equation of the shaft expressed as,

$$T_{sh} = J\ddot{\theta} + B\dot{\theta} \tag{6}$$

J and B represent the inertia of the shaft and friction of the DC motor, respectively.

C. DYNAMIC MODEL

The mechanical part of LLE is attached to the shaft of the DC motor. The required torque that rotates both the rotary shaft and link expressed as follows,

$$T_m = T_{ln} + T_{sh} \tag{7}$$

So, M, C and G in equation 3 expressed as follows,

$$M(\theta) = \begin{bmatrix} I_1 + d_1^2 m_1 + l_1^2(m_2 + m_3) + J & l_1 cos(\theta_1 - \theta_2)(d_2 m_2 + l_2 m_3) & d_3 m_3 l_1 cos(\theta_1 - \theta_3) \\ l_1 cos(\theta_1 - \theta_2)(d_2 m_2 + l_2 m_3) & I_2 + d_2^2 m_2 + l_2^2 m_3 + J & d_3 m_3 l_2 cos(\theta_2 - \theta_3) \\ d_3 m_3 l_1 cos(\theta_1 - \theta_3) & d_3 m_3 l_2 cos(\theta_2 - \theta_3) & I_3 + d_3^2 m_3 + J \end{bmatrix} \tag{8}$$

$$C(\theta, \dot{\theta}) = \begin{bmatrix} B\dot{\theta}_1 + l_1 sin(\theta_1 - \theta_2)\dot{\theta}_2^2(d_2 m_2 + l_2 m_3) \\ B\dot{\theta}_2 + d_3 m_3 l_2 sin(\theta_2 - \theta_3)\dot{\theta}_3^2 \\ B\dot{\theta}_3 - d_3 m_3 (l_1 sin(\theta_1 - \theta_3)\dot{\theta}_1^2 \\ \quad + d_3 m_3 l_1 sin(\theta_1 - \theta_3)\dot{\theta}_3^2 \\ \quad - l_1 sin(\theta_2 - \theta_1)\dot{\theta}_1^2(d_2 m_2 + l_2 m_3) \\ \quad + l_2 sin(\theta_2 - \theta_3)\dot{\theta}_2^2) \end{bmatrix} \tag{9}$$

$$G(\theta) = \begin{bmatrix} gsin(\theta_1)(l_1(m_2 + m_3) + d_1 m_1) \\ gsin(\theta_2)(d_2 m_2 + l_2 m_3) \\ gsin(\theta_3) d_3 m_3 \end{bmatrix} \tag{10}$$

The model of hip joint is given as follows,

$$G_1(s) = \frac{\theta_1(s)}{U_1(s)} = \frac{b}{a_{11}s^2 + a_{12}s + a_{13}} \tag{11}$$

TABLE 3. DC motor parameters.

parameter	Value
$R(\Omega)$	4
$J\ (Nm/rad\ s^{-2})$	5.5×10^{-6}
$B\ (Nm/rad\ s^{-1})$	0.003
$K_m(Nm/A)$	52

TABLE 4. Models for the three joints of LLE.

Joint	Model
hip	$G_1(s) = \frac{52}{0.8765s^2+0.012s+20.1}$
knee	$G_2(s) = \frac{52}{0.3478s^2+0.012s+8.224}$
ankle	$G_3(s) = \frac{52}{0.01811s^2+0.012s+0.8018}$

where,

$$b = K_m \tag{12}$$
$$a_{11} = R(I_1 + d_1^2 m_1 + l_1^2(m_2 + m_3) + J) \tag{13}$$
$$a_{12} = RB \tag{14}$$
$$a_{13} = R(gl_1(m_2 + m_3) + gd_1 m_1) \tag{15}$$

The model of knee joint is given as follows,

$$G_2(s) = \frac{\theta_2(s)}{U_2(s)} = \frac{b}{a_{21}s^2 + a_{22}s + a_{23}} \tag{16}$$

where,

$$a_{21} = R(I_2 + d_2^2 m_2 + l_2^2 m_3 + J) \tag{17}$$
$$a_{22} = RB \tag{18}$$
$$a_{23} = R(gd_2 m_2 + gl_2 m_3) \tag{19}$$

The model of ankle joint is given as follows,

$$G_3(s) = \frac{\theta_3(s)}{U_3(s)} = \frac{b}{a_{31}s^2 + a_{32}s + a_{33}} \tag{20}$$

where,

$$a_{31} = R(I_3 + d_3^2 m_3 + J) \tag{21}$$
$$a_{32} = RB \tag{22}$$
$$a_{33} = Rgd_3 m_3 \tag{23}$$

The DC motor parameters presented in Table 3. The models of hip, knee and ankle joints presented in Table 4. Figure 2 illustrates the block diagram of the system model.

III. CONTROL STRATEGY

The control strategy of the LLE is very crucial due to its ergonomics aspect. In previous studies by [29]–[31], designing an optimal control system for rehabilitation devices such as LLE will cause harm to the wearer. In addition, the wearer will not benefit due to the poor design of a control system. The LLE prototype consists of a control box connected to the reference frame attached to two legs; each has thigh, shank, and foot frames, as shown in Figure 3. The thigh frame is attached to the reference frame by the hip revolute joint, and also the shank frame is attached to the thigh frame by the knee revolute joint, and the foot frame is attached to the shank

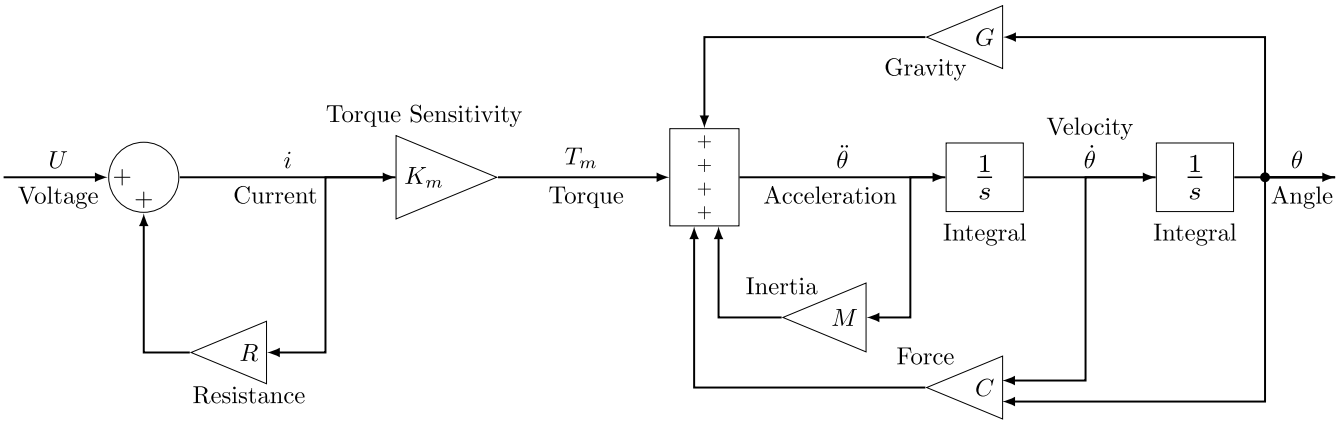


FIGURE 2. Block diagram of the system model.

frame by the ankle revolute joint. The utilized actuator in the LLE prototype is DC permanent magnet brushed geared motor. The motor and gearbox connected in parallel with the links in each leg. The gearbox used to increase the generated torque of the DC motor. The conceptual model of the experimental setup is shown in Figure 3(a). Figure 3(b) illustrates the components of LLE prototype.

The equipment used is an LLE prototype, Personal Computer (PC) equipped with a Windows 10 operating system and Intel core i7, Arduino Mega 2560 board with motor drivers, DC motors, gearbox and encoders. An incremental rotary encoder located at the gearbox's output shaft and the joint position information sent to the control system as feedback. Arduino Mega 2560 microcontroller used to communicate with the components of LLE and PC. The control system's input toward the microcontroller is Pulse Width Modulation (PWM), which bounded from 0 to 255. A motor driver utilized to control the voltage and movement direction of the DC motors by changing the positive and negative input voltage, which can tolerate a maximum of 12V. In addition, the direction data sent to the microcontroller; when the voltage to the plant is positive, it is 0, and it is 1 when the voltage to the plant is negative. The simulation performed in MATLAB/Toolbox.

A. PID CONTROLLER MODEL

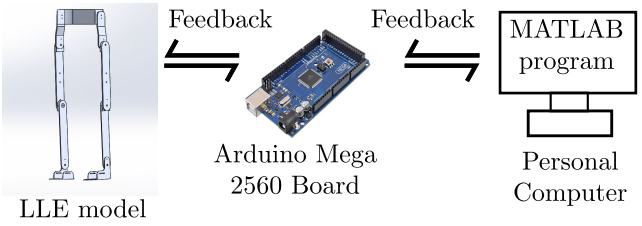
In this study, a PID controller that has been at the heart of control engineering practices is used [5]. PID controller consists of three constant parameters K_p, K_i and K_d to control the error trend of the angular trajectory of LLE [25], [26], [32], [33].

The PID model expressed as follows,

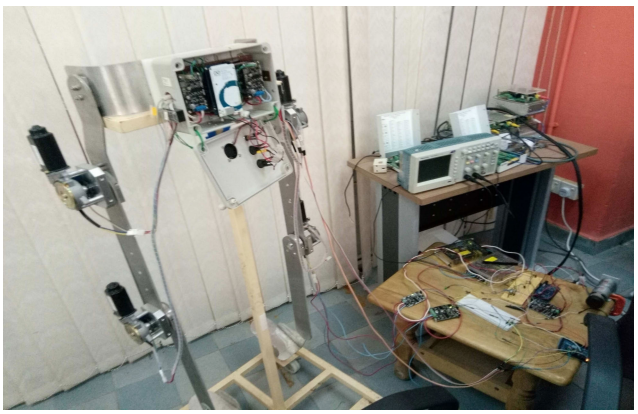
$$C_i(s) = \frac{U}{e} = K_p + K_i\frac{1}{s} + K_d s \tag{24}$$

where $C_i(s)$ represents the output of the PID controller of the hip, knee, and ankle joints, whereas $i = 1$, $i = 2$ and $i = 3$, respectively. Equation 24 simplified as,

$$C_i(s) = \frac{K_d s^2 + K_p s + K_i}{s} \tag{25}$$



(a) The integral steps of study setup



(b) LLE prototype

FIGURE 3. Experimental setup.

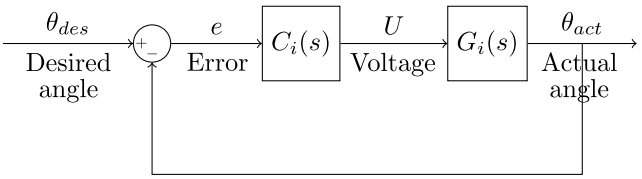


FIGURE 4. Closed-loop control system of each joint.

Figure 4 illustrates the closed-loop control system of each joint in LLE. Where θ_{des} and θ_{act} is the desired and actual angular trajectory; U is the input voltage to the DC motor; e is the steady-state error (SSE) of the closed-loop control

TABLE 5. PID controller parameters.

Joint	K_p	K_i	K_d
Hip	4.4960	8.9080	0.5672
Knee	2.4432	6.5962	0.2262
Ankle	0.1039	0.2893	0.0093

system, which is the difference between the actual and desired trajectory as follows,

$$e = \theta_{des} - \theta_{act} \tag{26}$$

The model function of the control system as follows,

$$\frac{\theta_{act}}{\theta_{des}} = \frac{C_i(s)G_i(s)}{1 + C_i(s)G_i(s)} \tag{27}$$

$G_i(s)$ represents the model of the hip, knee, and ankle joints, whereas $i = 1$, $i = 2$ and $i = 3$, respectively. The PID parameters can be tuned based on the model, either by trial-and-error or classical methods. In this paper, the PID parameters initially tuned using Ziegler-Nichols (ZN) method [26]. In Algorithm 2, K_i and K_d are set to zero, while K_p is set as K_u, which in turn is increased by a step value K_s. K_u a value chosen while observing a sustained oscillation; such parameter is called critical gain K_{cr}. After that, the critical period calculated between two excessive cycles. Based on K_{cr} and T_{cr}, the values of the PID parameters is calculated through the following equations,

$$K_p = 0.6 \times K_{cr} \tag{28}$$

$$K_i = \frac{1.2 \times K_{cr}}{T_{cr}} \tag{29}$$

$$K_d = \frac{3 \times K_{cr} \times T_{cr}}{40} \tag{30}$$

The limitations of the ZN method are as follows:

1) The PID tuning process depends only on two critical values, gain (K_{cr}) and period (T_{cr}), obtained during the observation of sustained oscillation.
2) The oscillation is observed by the user, which adds some inaccuracy to the final result.
3) The method cannot define the control objectives or closed-loop performance requirements.

Table 5 presents the PID parameters for the hip, knee and ankle joints. For the hip joint, the K_p, K_i, and K_d are 4.4960, 8.9080 and 0.5672, respectively. For the knee joint, the K_p, K_i, and K_d are 2.4432, 6.5962 and 0.2262, respectively. For the ankle joint, the K_p, K_i and K_d are 0.1039, 0.2893, and 0.0093, respectively.

B. NEURAL NETWORK MODEL

Optimization is the process of enhancing the standard way of finding the correct solution to a problem. In neural networks, it is very crucial to reach the minimum error with less time. Therefore, many algorithms have been developed for finding the global minimum of an objective function, e.g. Mean Squared Error (MSE), of a neural network. There are two stages in neural networks: the Forward Pass (FP) and Backward Pass (BP). In FP, the network passes input data from one layer to another until it reaches the output layer. This process involves taking the weighted sum of input data with a bias for each node and holding it in a transfer function. Usually, hidden nodes' transfer functions are different from the output nodes, and that depends on the nature of the problem. In BP, the network passes through the output nodes to calculate the error with the help of target data. This process involves updating the weights and biases using an iterative optimization algorithm. Three popular ANN-based algorithms, such as GD, LM and SCG, were chosen to be discussed and compared [34].

GD converges faster for larger datasets, as claimed by Moller [35], and it causes updates to the parameters more frequently. The updates to the parameters can be done either once for each sample or for all examples in the dataset. However, it raises the complexity of calculation for each learning iteration considerably since it has to perform a line search, e.g. global error derivative, to determine a suitable step size. Also, it often shows poor convergence as the minimization based on the linear approximation of Taylor expansion. In addition, due to the use of a constant step size, which is inefficient, in many cases, and causes lower robustness. While, LM has two possible options for the algorithm's direction at each iteration. Also, it is very efficient when training networks that have up to a few hundredweights. Although, the algorithm can be slow for enormous datasets, as claimed by Hagan [36]. While SCG avoids the line search for each learning iteration, thus avoiding the calculation's complexity, as claimed by Moller [35]. In addition, it does not include any user-dependent parameters required for the success of the algorithm. Although, it requires some conditions during calculation for the success of the algorithm. The GD 3, LM 4, and SCG 5 algorithms are presented in the appendix.

In the control system field, the conventional ANN method used to predict PID parameters, in self-tuning, by feeding desired signal, error signal, and actual signal as inputs to the neural network [6]. This method constitutes two issues when used in manual tuning, in which the PID parameters are kept constant during simulation time. So, the first issue is that the neural network will experience difficultly correlating between input and target data. Secondly, it will take time to process each input and target data sample due to their size for each training set.

As a solution, the proposed ANN method suggests that the step response characteristics are the inputs to the neural network. The chosen inputs are rise time (s), settling time (s) and overshoot (%). Figure 5 illustrates the neural network architecture for each joint model. The network consists of one input layer, two hidden layers and one output layer. The input layer consists of three nodes are rise time (T_r), settling time (T_s), and per cent overshoot (PO). Each hidden layer consists of 15 hidden nodes. Also, each hidden node processes its output using the tangent sigmoid function. The output layer consists of K_p, K_i and K_d, and each output node uses a linear function. Figure 6 and Algorithm 1 exhibit the flow chart

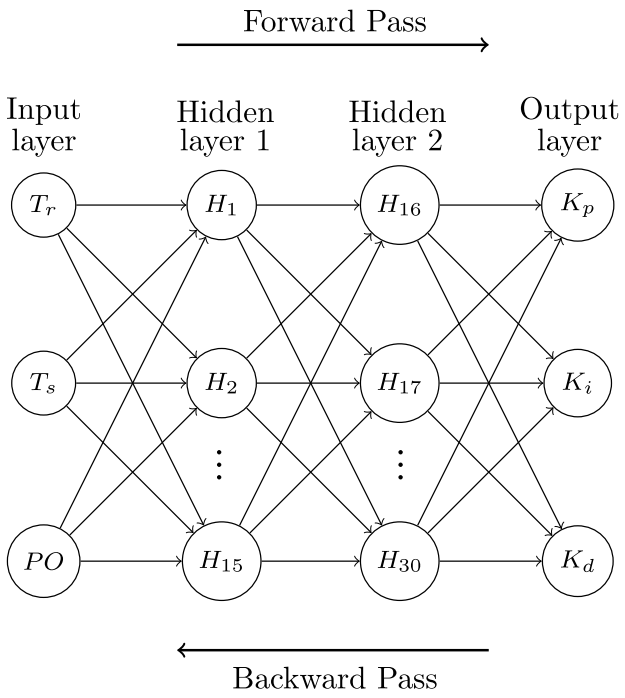


FIGURE 5. Proposed ANN architecture for each DoF of LLE.

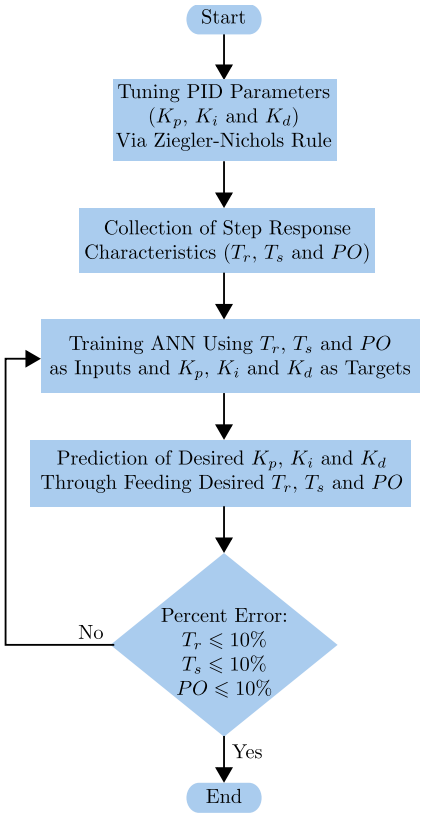


FIGURE 6. Flow chart of proposed ANN.

and pseudo code of proposed ANN method, respectively. Where K_{step} is a constant value for a step of K_p, K_i and K_d, and $rand[0, 1]$ is a positive random value between zero and one. The flow chart demonstrates the process of the proposed ANN for finding the optimal PID parameters. In addition, the flow chart corresponds to each joint in the LLE model. For example, it is applied first for the hip model to calculate the K_p, K_i and K_d. After that, it is repeated for the knee model and then for the ankle model. The steps presented in the flow chart described as follows,

1) The PID parameters for a joint model are tuned using the ZN method.
2) The step response characteristics of a control system collected in a spreadsheet.
3) The training of the neural network goes through the input data, T_r, T_s and PO, and target data, K_p, K_i and K_d.
4) The ANN model predicts the desired PID parameters for the control system by feeding the desired values of step response characteristics.
5) If the percent errors of T_r, T_s and OP are less than or equal to 10%, then the ANN model is ready to find the optimal PID parameters. Otherwise, it must be trained again with different settings and more training data.

The effect of changing the user-dependent parameters of ANN-based algorithms on the training outcome has experimented. It found that the error criterion has converged faster and more accurate. Also, a data set used to train the neural network, and then another data set was used to test it. The size of the input data and target data is 3000×3. Furthermore, the percent error calculated after testing the neural network, where the desired input data fed to the neural network after the training finished, but the actual step response characteristics retrieved after applying the new PID parameters to the system. The following equations represent the percent error for each input.

$$T_r^{error} = \frac{T_r^{actual} - T_r^{desired}}{T_r^{desired}} \times 100\% \quad (31)$$

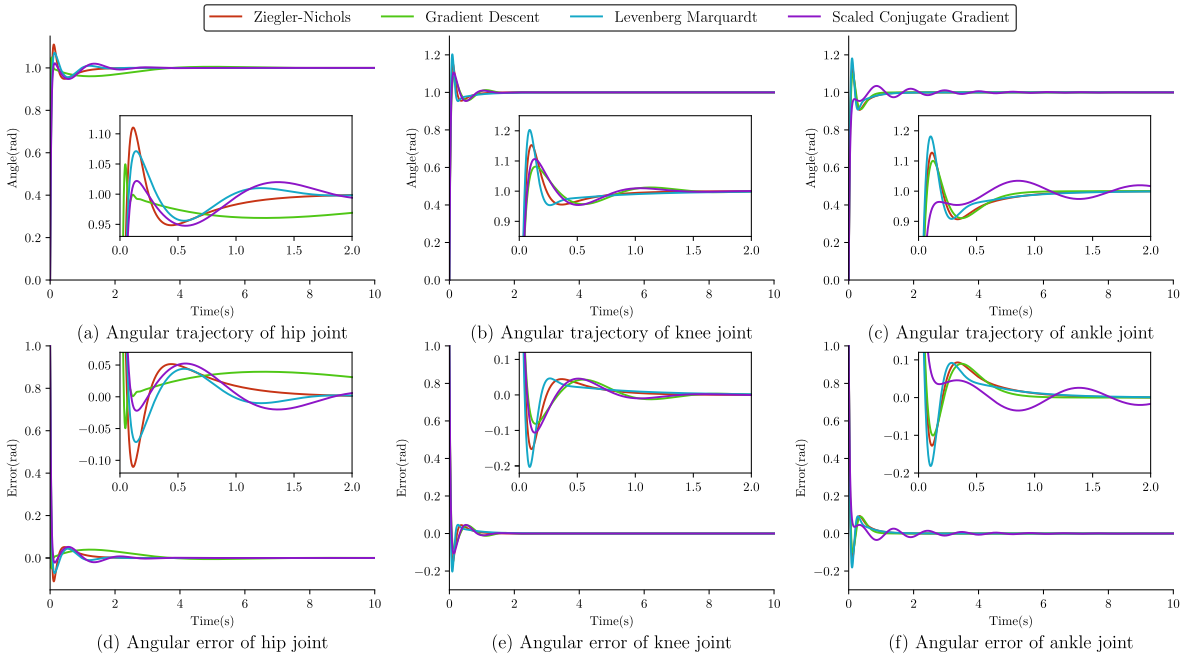
$$T_s^{error} = \frac{T_s^{actual} - T_s^{desired}}{T_s^{desired}} \times 100\% \quad (32)$$

$$PO^{error} = \frac{PO^{actual} - PO^{desired}}{PO^{desired}} \times 100\% \quad (33)$$

where T_r is the time required for the response curve to rise from 10% to 90% of its final value; T_s is the time required for the response curve to fall to within a range of 2% of its final value; PO is the value of the response curve measured after exceeding the steady-state value at the peak time [37].

IV. RESULTS AND DISCUSSIONS

In this paper, the objective is to find the optimum values of K_p, K_i and K_d, which gives the outstanding angle response in terms of rising time (s), settling time (s), overshoot (%), peak (rad) and peak time (s). Comparing the proposed ANN with the conventional one and the experimental validation for the proposed ANN with other researchers' studies is carried out in this section.



(a) Angular trajectory of hip joint (b) Angular trajectory of knee joint (c) Angular trajectory of ankle joint

(d) Angular error of hip joint (e) Angular error of knee joint (f) Angular error of ankle joint

FIGURE 7. Conventional ANN method - angular and error trajectories.

Algorithm 1 Pseudo Code of Proposed ANN

```
Start
Set G as joint model;
Set K_p = 0, K_i = 0 and K_d = 0; Set K_step = 1;
Set m = 3000 and n = 3;
Set m × n matrices for x, t and assign them to zero;
for i ← 1, m do
    K_p = rand[0, 1] × (K_p + K_step);
    K_i = rand[0, 1] × (K_i + K_step);
    K_d = rand[0, 1] × (K_d + K_step);
    Set C as PID model;
    Connect G and C models in closed-loop;
    return Vector of times t_out corresponding to the step
    responses in y_out;
    Obtain step response characteristics (T_r, T_s and PO)
    through t_out and y_out;
    Set T_r, T_s and PO to x_{i,n};
    Set K_p, K_i and K_d to t_{i,n};
end for
Training ANN model by feeding input x and target t;
Set x_des as new input to ANN model;
return K_p, K_i and K_d corresponding to x_des;
Repeat steps 10 − 13;
Calculate percent error for T_r, T_s and PO;
End
```

Figure 7 shows angular and error trajectories of the three joints with different control parameters tuned by ZN, GD, LM and SCG. Table 6 presents the results of the conventional ANN method for hip, knee and ankle joints.

The SCG obtained better response characteristics in the hip joint regarding overshoot and peak with 2.1319% and 1.0213 rad. However, LM obtained the fastest response regarding rising time and settling time with 0.0533 s and 0.8067 s. Also, the GD characteristics are almost equal to LM, but GD obtained the lowest overshoot compared to LM. Consequently, SCG characteristics came first, then GD, LM and ZN took the second, third, and fourth places, respectively. In the knee joint, the GD obtained better response characteristics regarding overshoot and peak with 8.0976% and 1.0810 rad, respectively. However, the LM obtained the fastest response regarding rising time and settling time with 0.0367 s and 0.5768 s, respectively. Consequently, GD characteristics came first, then SCG, ZN and LM took the second, third, and fourth places, respectively. The SCG obtained better response characteristics in the ankle joint regarding overshoot and peak with 3.4453% and 1.0344 rad, respectively. However, the LM obtained the fastest response in terms of rising time with 0.0460 s. Consequently, SCG characteristics came first, then GD, ZN and LM took the second, third, and fourth places, respectively.

The result of the conventional ANN on a manually-tune PID controller is as follows. The hip joint model was tuned optimally by the SCG algorithm with 1.1736, 8.9632 and 0.5672 for K_p, K_i and K_d. While, the knee joint was tuned optimally by GD algorithm with 0.9321, 6.7738 and 0.2262 for K_p, K_i and K_d, respectively. Whereas the ankle joint was tuned optimally by the SCG algorithm with 0.0070, 0.3311 and 0.0093 for K_p, K_i and K_d, respectively. However, LM obtained the best training performance, and GD obtained the worst training performance. So, the conventional ANN method results are unreliable because the training performance does not reflect prediction accuracy. There is no correlation between input and target data since each sample of target data is constant for each input data sample

TABLE 6. Conventional ANN method results (Desired angle, angular error, actual angle.)

Characteristics	Hip joint			
	ZN	GD	LM	SCG
Rise time (s)	0.0473	0.0576	0.0533	0.0580
Settling time (s)	0.9354	0.8315	0.8067	0.8924
Overshoot (%)	10.5582	3.7488	7.1099	2.1319
Peak (rad)	1.1056	1.0375	1.0711	1.0213
Peak time (s)	0.1003	0.1672	0.1338	0.1338
PID parameters				
K_p	4.4960	1.1371	2.2773	1.17363
K_i	8.9080	13.0566	13.5694	8.96325
K_d	0.5672	0.5667	0.5672	0.5672
Training Results				
Epoch		50000	20	5000
Time (min)		6.45	17.22	3.37
Mean Squared Error		6.2×10^{-26}	1.3×10^{-28}	7.8×10^{-25}
Characteristics	Knee joint			
	ZN	GD	LM	SCG
Rise time (s)	0.0431	0.0527	0.0367	0.0502
Settling time (s)	0.6463	0.7718	0.5768	0.7179
Overshoot (%)	15.1643	8.0976	19.6256	10.6304
Peak (rad)	1.1516	1.0810	1.1963	1.1063
Peak time (s)	0.1003	0.1338	0.1003	0.1338
PID parameters				
K_p	2.4432	0.9321	3.92945	1.2173
K_i	6.5962	6.7738	6.151945	8.0733
K_d	0.2262	0.2262	0.2262	0.2262
Training Results				
Epoch		50000	20	5000
Time (min)		6.45	16.24	3.35
Mean Squared Error		2.1×10^{-21}	2.5×10^{-29}	5.2×10^{-25}
Characteristics	Ankle joint			
	ZN	GD	LM	SCG
Rise time (s)	0.0514	0.0547	0.0460	0.0867
Settling time (s)	0.8214	0.7063	0.7972	1.4991
Overshoot (%)	11.8511	9.8396	18.0591	3.4453
Peak (rad)	1.1185	1.0984	1.1806	1.0344
Peak time (s)	0.1003	0.1338	0.1003	0.8696
PID parameters				
K_p	0.1039	0.079904	0.149102	0.007044
K_i	0.2893	0.3171	0.344711	0.3311
K_d	0.0093	0.0093	0.0093	0.0093
Training Results				
Epoch		50000	20	5000
Time (min)		6.52	18.12	3.36
Mean Squared Error		4.5×10^{-20}	5.1×10^{-31}	4.3×10^{-28}

(Manual-tuning). It is worthwhile to mention that there is a correlation between input and target data in the case of self-tuning.

Figure 8 presents angular and error trajectories of the three joints with different control parameters tuned by ZN, GD, LM and SCG. Table 7 presents the results of the proposed ANN method for hip, knee and ankle joints.

In the hip joint, the SCG obtained better response characteristics regarding rising time, overshoot and peak with 0.0267 s, 0.3126% and 1.0032 rad. SCG obtained a training time of 2.5 min, and the MSE is 3.7×10^{-28}. Also, the least accurate ANN-based algorithm is LM with 5.2×10^{-26}, and it is the slowest among them with 9.3 min. Nevertheless, the BP algorithms have almost the same system characteristics and PID parameters. In the knee joint, the GD obtained better response characteristics in terms of rising time, overshoot, peak and peak time with 0.0266 s, 0.6335%, 1.0063 rad and 0.0334 s, respectively. GD obtained a training time of 5.2 min, and the MSE is 8.1×10^{-27}. Moreover, the least accurate ANN-based algorithm is SCG with 9.6×10^{-25}, but it is the fastest among them with 2.3 min. Although, the BP algorithms have almost the same system characteristics and PID parameters. The GD obtained better response characteristics in the ankle joint regarding rising time, overshoot, peak and peak time with 0.0267 s, 0.2619%, 1.0026 rad, and 0.0334 s. GD obtained a training time of 5.3 min, and the MSE is 2.2×10^{-27}. In addition, the least accurate ANN-based algorithm is SCG with 6.1×10^{-25}, but it is the fastest among them with 2.6 min. However, the BP algorithms have almost the same system characteristics and PID parameters.

The result of the proposed ANN on a manually-tune PID controller is as follows. The hip joint model was tuned optimally by the SCG algorithm with 6.1919, 7.0092 and 4.3065 for K_p, K_i and K_d, respectively. While, the knee joint was tuned optimally by GD algorithm with 5.5106, 6.5755 and 2.0475 for K_p, K_i and K_d, respectively. The ankle joint was

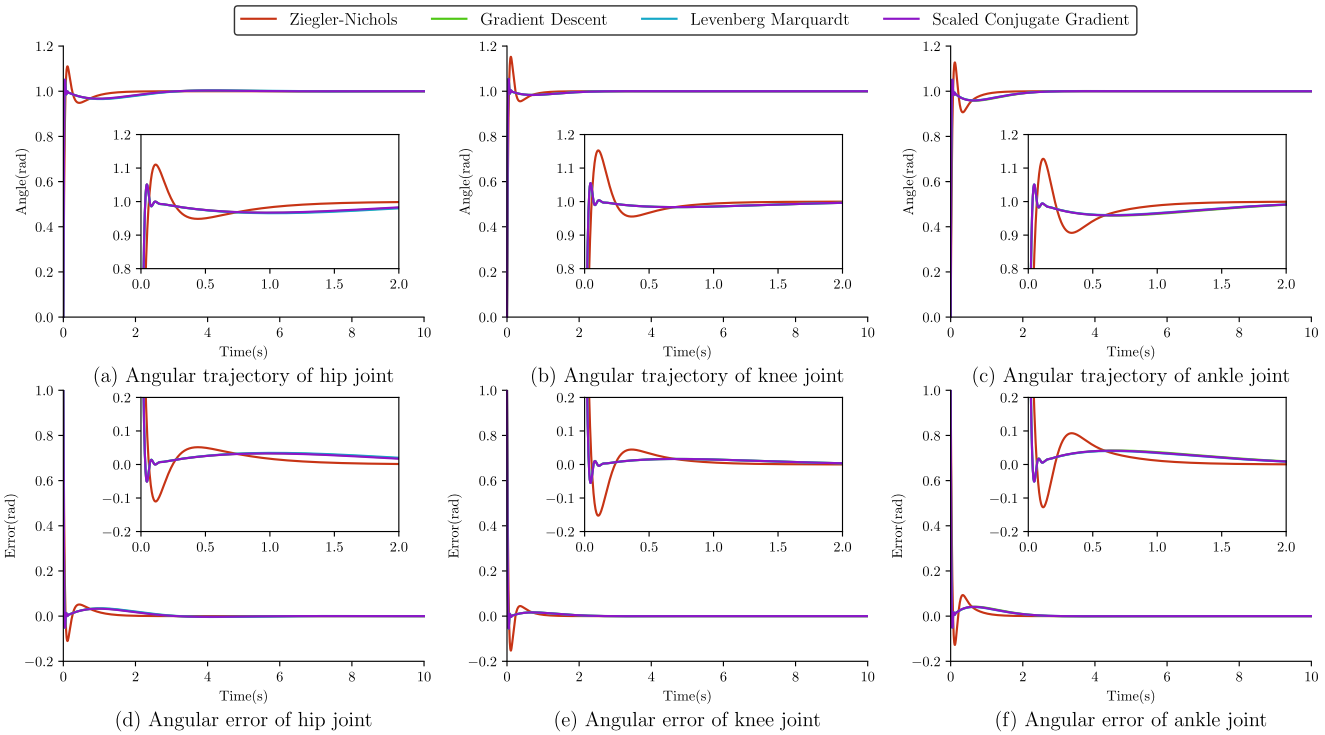


FIGURE 8. Proposed ANN method - angular and error trajectories.

tuned optimally by the GD algorithm with 0.2172, 0.2585 and 0.0866 for K_p, K_i and K_d, respectively. GD and SCG obtained the best training performance, and LM obtained the worst training performance due to a low number of epochs. In the proposed ANN method, the correlation between input and target data found since each input data sample is inconstant for each sample of target data (Manual-tuning).

It is crucial to validate the proposed ANN method by comparing it with other methods and dynamic systems published in the literature. Table 8 shows a comparison of the current study with Amiri's study [26]. The tuning of the hip joint model in Amiri's paper compared to the proposed ANN method. The proposed ANN method has an advantage in reducing overshoot and settling time, as shown in Figure 9 and the graphs in Amiri's paper.

Table 9 compares the current study with Kumar's study [38], where the two studies used ANN. However, in the current method, the inputs of ANN are selected to be rise time, settling time and overshoot, while in Kumar's method, the inputs of ANN are the error, accumulation of error, and rate of change of error. The tuning of the system model of Example C in Kumar's study compared to the proposed ANN method. The current method has an advantage in reducing overshoot and settling time, as shown in Figure 10 and the graphs in Kumar's study.

Figure 11(a) illustrates a comparison of the current method applied on the LLE model with other methods applied on different second-order models in the literature [8], [26],

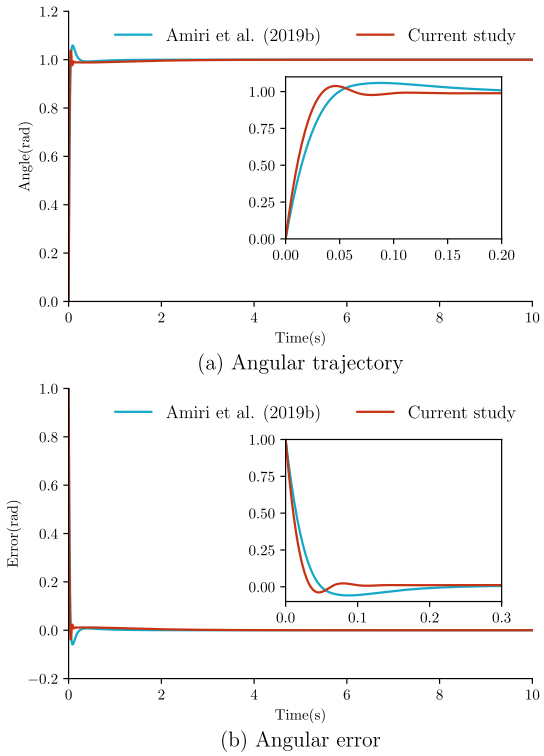


FIGURE 9. Comparison of angle (Top) and error (Bottom) curves with Amiri *et al.*

[38], [39] regarding rising time, settling time, overshoot, undershoot, and peak time. The current method obtained

TABLE 7. Proposed ANN method results (Rise time, settling time, overshoot.)

Characteristics	Hip joint			
	ZN	GD	LM	SCG
Rise time (s)	0.0473	0.0267	0.0267	0.0267
Settling time (s)	0.9354	1.9289	2.0059	1.8691
Overshoot (%)	10.5582	0.3569	0.3783	0.3126
Peak (rad)	1.1056	1.0036	1.0038	1.0032
Peak time (s)	0.1003	4.0803	4.1806	4.0468
PID parameters				
K_p	4.4960	5.9701	5.773315	6.1920
K_i	8.9080	6.83522	6.5006	7.0092
K_d	0.5672	4.3965	4.4088	4.3065
Training Results				
Epoch		10000	10	1000
Time (min)		5.1	9.3	2.5
Mean Squared Error		3.2×10^{-27}	5.2×10^{-26}	3.7×10^{-28}
Characteristics	Knee joint			
	ZN	GD	LM	SCG
Rise time (s)	0.0431	0.0266	0.0266	0.0266
Settling time (s)	0.6463	0.0326	0.0326	0.0326
Overshoot (%)	15.1643	0.6335	0.6483	0.6815
Peak (rad)	1.1516	1.0063	1.0065	1.0068
Peak time (s)	0.1003	0.0334	0.0334	0.0334
PID parameters				
K_p	2.4432	5.5106	5.51991	5.5699
K_i	6.5962	6.57554	6.5749	6.5754
K_d	0.2262	2.0475	2.0290	1.9945
Training Results				
Epoch		10000	10	1000
Time (min)		5.2	9.1	2.3
Mean Squared Error		8.1×10^{-27}	4.4×10^{-26}	9.6×10^{-25}
Characteristics	Ankle joint			
	ZN	GD	LM	SCG
Rise time (s)	0.0514	0.0267	0.0267	0.0267
Settling time (s)	0.8214	1.5610	1.4887	1.5087
Overshoot (%)	11.8511	0.2619	0.3812	0.3839
Peak (rad)	1.1185	1.0026	1.0038	1.0038
Peak time (s)	0.1003	0.0334	0.0334	0.0334
PID parameters				
K_p	0.1039	0.2172	0.2305	0.22495
K_i	0.2893	0.2585	0.2688	0.2619
K_d	0.0093	0.08661	0.08264	0.0808
Training Results				
Epoch		10000	10	1000
Time (min)		5.3	9.5	2.6
Mean Squared Error		2.2×10^{-27}	1.3×10^{-26}	6.1×10^{-25}

TABLE 8. Comparison of proposed ANN with IMRAC and MRAC methods.

	Current Study	Amiri et al. [26]	
Method Name	ANN	IMRAC	MRAC
Concept	Inputs to ANN are rise time, settling time and overshoot	PID parameters are initialized from ZN method	PID parameters are initialized from zero
Type of tuning	Manual-tuning	Self-tuning	Self-tuning
DoF of LLE	6	4	4
Electric model of DC motor	Ohm's law	Kirchhoff's law	Kirchhoff's law
Input signal	Constant unit step	Sinusoidal	Sinusoidal
Basis of comparison	Hip joint model in [26]		
Rise time (s)	0.0270	0.0362	N/A
Settling time (s)	0.0331	0.1674	N/A
Overshoot (%)	0	5.6081	N/A
Peak (rad)	0.9999	1.0561	N/A
K_p	10.7044	5.2522	N/A
K_i	7.4347	9.9557	N/A
K_d	4.0073	0.3898	N/A
Performance	High	Moderate	Low

the lowest rise time, and the other performance is credited compared to other methods. Figure 11(b) shows a comparison of peak (amplitude), where it is clear that Kumar's result [38] obtained the lowest value, and the current method is in the

TABLE 9. Comparison between different ANN methods.

Method Name	Current Study ANN	Kumar et al. [38] ANN
Concept	Inputs to ANN are rise time, settling time and overshoot	Inputs to ANN are error, accumulation of error and rate of change of error
Type of tuning	Manual-tuning	Self-tuning
System model	LLE 1. Hip model 2. Knee model 3. Ankle model	A. Second order. B. Second order with single parameter variation. C. Second order with two parameters variations. D. Armature DC motor with load.
Electric model of DC motor	Ohm's law	Kirchhoff's law
Input signal	Constant unit step	Inconstant unit step
Basis of comparison	The system model of example C in [38]	
Rise time (s)	0.0270	4.1993
Settling time (s)	3.0933	7.1418
Peak time (s)	4.7492	10
K_p	3.6828	0.00062797
K_i	4.1870	0.5119
K_d	3.6314	0.0001
Performance	High	Moderate

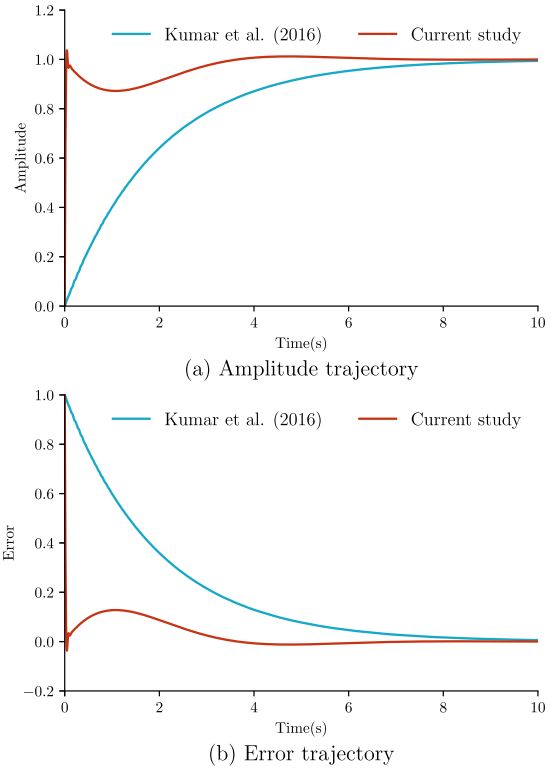


FIGURE 10. Comparison of amplitude (Top) and error (Bottom) curves with Kumar *et al.*

second place. Finally, the current method result is consistent compared to other methods in the literature with some credit.

The essential advantages of the proposed ANN-PID method compared to other methods are as follows:

1) The method can define the control objectives and closed-loop performance requirements.

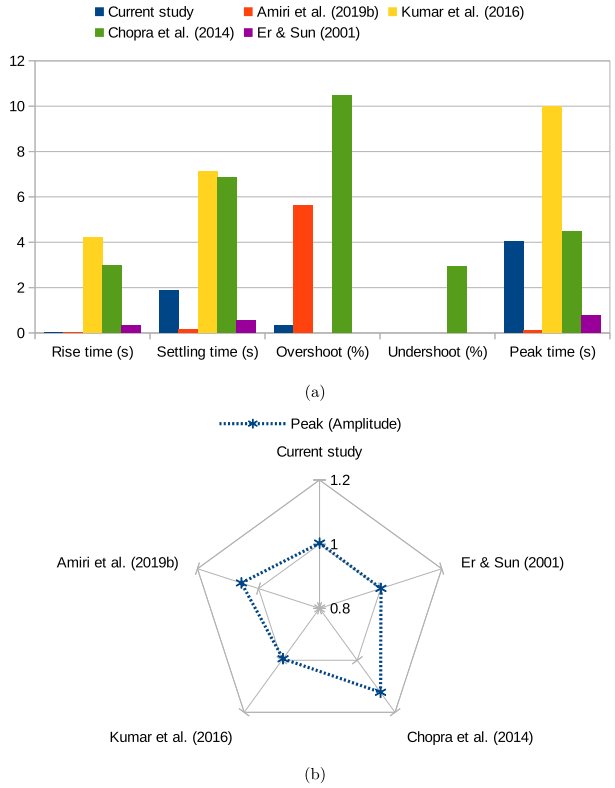


FIGURE 11. Comparison of current study with other studies, (a) System characteristics, (b) Peak (amplitude).

2) The ANN model can accept additional step response characteristics like peak, peak time and undershoot.
3) The method has several options for training the ANN model, like GD, LM and SCG.

The limitations of the method are as follows:

1) The PID tuning process depends only on the characteristics of the step response.
2) The necessity for using deep learning when the system model is very complicated.
3) The application of the proposed method is in the offline tuning process.

V. CONCLUSION

In this paper, a comparison study made between two different methodologies of ANN architecture. One method has been used commonly among researchers, and the other method proposed. The mechanical model and electric model determined by using the Euler-Lagrange equation and Ohm's law, respectively. Also, the LLE model determined by the electrical and physical features of the actual prototype. The PID parameters for the LLE model initially tuned by using the ZN method. The parameters of the tuned PID controller were optimized using ANN-based algorithms. The proposed ANN method proves the converged faster in terms of time and epoch, offered improved step response characteristics, and gave more reliable PID parameters.

Moreover, the proposed method was used and compared with other researchers' methods based on their system models. As a result, the proposed ANN method has proved its superiority among other methods, in literature, based on the best step response characteristics for different system models.

In future work, the comparison and analysis of the real-time control in the testing process for the actual LLE prototype will investigate.

APPENDIX
TUNING ALGORITHMS

Algorithm 2 ZN Algorithm [26]

$K_i \leftarrow 0$
$K_d \leftarrow 0$
$K_p \leftarrow K_u$
Require: K_u & K_s
while observing a sustained oscillations **do**
  $K_u = K_u + K_s$
end while
$K_u \leftarrow K_{cr}$
Obtain the critical period of sustained oscillations T_{cr}
Calculate K_p, K_i and K_d

Algorithm 3 GD Algorithm [35]

Choose initial weight vector $\tilde{w}_1$ and set $k = 1$.
Determine the gradient $E'(\tilde{w})$, as a search direction, and a step size α_k, so that $E(\tilde{w}_{k+1}) < E(\tilde{w}_k)$.
Update vector: $\tilde{w}_{k+1} = \tilde{w}_k - \alpha_k E'(\tilde{w})$.
if $E'(\tilde{w}_k) \neq 0$ **then**
  Set $k = k + 1$ and go to 2
else
  return $\tilde{w}_{k+1}$ as the desired minimum
end if

Algorithm 4 LM Algorithm [36]

Calculate the sum of squares of errors $V(\underline{x})$.
Compute the jacobian matrix $J(\underline{x})$ using Δ^M, $\underline{\delta}^k$, $\frac{\partial \hat{V}}{\partial w^k(i,j)}$ and $\frac{\partial \hat{V}}{\partial b^k(i)}$.
Obtain $\Delta\underline{x}$.
Calculate $V(\underline{x} + \Delta\underline{x})$.
while $V(\underline{x} + \Delta\underline{x}) \neq 0$ **do**
  if $V(\underline{x} + \Delta\underline{x}) < V(\underline{x})$ **then**
    Reduce μ by β, let $\underline{x} = \underline{x} + \Delta\underline{x}$
    return $V(\underline{x})$
  else
    Increase μ by β
    return $\Delta\underline{x}$
  end if
end while

Algorithm 5 SCG Algorithm [35]

Choose weight vector $\tilde{w}_1$ and scalars $0 < \sigma \leq 10^{-4}$, $0 < \lambda_1 \leq 10^{-6}$, $\bar{\lambda}_1 = 0$. Set $\tilde{p}_1 = \tilde{r}_1 = -E'(\tilde{w}_1)$, $k = 1$ and success = true.
if success = true **then**
  Calculate second order information:
  $\sigma_k = \sigma/|\tilde{p}_k|$,
  $\tilde{s}_k = (E'(\tilde{w}_k + \sigma_k \tilde{p}_k) - E'(\tilde{w}_k))/\sigma_k$,
  $\delta_k = \tilde{p}_k^T \tilde{s}_k$.
end if
Scale $\delta_k : \delta_k = \delta_k + (\lambda_k - \bar{\lambda}_1)|\tilde{p}_k|^2$.
if $\delta_k \leq 0$ **then**
  Make the Hessian matrix positive definite:
  $\bar{\lambda}_k = 2(\lambda_k - \delta_k/|\tilde{p}_k|^2)$,
  $\delta_k = -\delta_k + \lambda_k|\tilde{p}_k|^2$,
  $\lambda_k = \bar{\lambda}_k$.
end if
Calculate step size:
$\mu_k = \tilde{p}_k^T \tilde{r}_k$,
$\alpha_k = \mu_k/\delta_k$.
Calculate the comparison parameter:
$\Delta_k = 2\delta_k[E(\tilde{w}_k) - E(\tilde{w}_k + \alpha_k \tilde{p}_k)]/\mu_k^2$.
if $\Delta_k \geq 0$ **then**
  Successful reduction in error can be made:
  $\tilde{w}_{k+1} = \tilde{w}_k + \alpha_k \tilde{p}_k$,
  $\tilde{r}_{k+1} = -E'(\tilde{w}_{k+1})$,
  $\bar{\lambda}_k = 0$, success = true.
end if
if $k \bmod N = 0$ **then**
  Restart algorithm:
  $\tilde{p}_{k+1} = \tilde{r}_{k+1}$
else
  $\beta_k = (|\tilde{r}_{k+1}|^2 - \tilde{r}_{k+1}^T \tilde{r}_k)/\mu_k$,
  $\tilde{p}_{k+1} = \tilde{r}_{k+1} + \beta_k \tilde{p}_k$.
end if
if $\Delta_k \geq 0.75$ **then**
  Reduce the scale parameter:
  $\lambda_k = \frac{1}{4}\lambda_k$.
else
  $\tilde{\lambda}_k = \lambda_k$,
  success = false.
end if
if $\Delta_k < 0.25$ **then**
  Increase the scale parameter:
  $\lambda_k = \lambda_k + (\delta_k(1 - \Delta_k))/|\tilde{p}_k|^2)$.
end if
if the steepest descent direction $\tilde{r}_k \neq 0$ **then**
  Set $k = k + 1$ and go to 2
else
  return $\tilde{w}_{k+1}$ as the desired minimum.
end if

REFERENCES

[1] N. Aliman, R. Ramli, and S. M. Haris, "Design and development of lower limb exoskeletons: A survey," *Robot. Auton. Syst.*, vol. 95, pp. 102–116, Sep. 2017.

[2] W. Huo, S. Mohammed, J. C. Moreno, and Y. Amirat, "Lower limb wearable robots for assistance and rehabilitation: A state of the art," *IEEE Syst. J.*, vol. 10, no. 3, pp. 1068–1081, Sep. 2016.

[3] M. S. Amiri, R. Ramli, M. A. A. Tarmizi, and M. F. Ibrahim, "Simulation and control of a six degree of freedom lower limb exoskeleton," *J. Kejuruteraan*, vol. 32, no. 2, pp. 197–204, May 2020.

[4] M. R. Nimbalkar, M. S. Lavhate, M. V. Lawande, and M. N. Toradmal, "A review on significance of PID controller for speed control of DC motor," *Int. J. Recent Innov. Trends Comput. Commun.*, vol. 3, no. 4, pp. 2080–2082, 2015.

[5] H. B. Patel and S. N. Chaphekar, "Developments in PID controllers: Literature survey," *Int. J. Eng. Innov. Res.*, vol. 1, no. 5, pp. 425–430, 2012.

[6] S. Bari, S. S. Z. Hamdani, H. U. Khan, M. U. Rehman, and H. Khan, "Artificial neural network based self-tuned PID controller for flight control of quadcopter," in *Proc. Int. Conf. Eng. Emerg. Technol. (ICEET)*, Feb. 2019, pp. 1–5.

[7] M. S. Amiri, R. Ramli, and M. F. Ibrahim, "Genetically optimized parameter estimation of mathematical model for multi-joints hip–knee exoskeleton," *Robot. Auton. Syst.*, vol. 125, Mar. 2020, Art. no. 103425.

[8] V. Chopra, S. K. Singla, and L. Dewan, "Comparative analysis of tuning a PID controller using intelligent methods," *ACTA Polytechnica Hungarica*, vol. 11, no. 8, pp. 235–249, 2014.

[9] X.-S. Yang, "Particle swarm optimization," in *Nature-Inspired Optimization Algorithms*. Amsterdam, The Netherlands: Elsevier, 2014, ch. 7, pp. 99–110.

[10] S. Sumathi and S. Paneerselvam, *Computational Intelligence Paradigms: Theory & Applications Using MATLAB*. Boca Raton, FL, USA: CRC Press, 2010.

[11] A. Panbude and M. Sharma, "Implementation of neural network for PID controller," *Int. J. Comput. Appl.*, vol. 975, p. 8887, 2015.

[12] H. J. Asl, T. Narikiyo, and M. Kawanishi, "Neural network-based bounded control of robotic exoskeletons without velocity measurements," *Control Eng. Pract.*, vol. 80, pp. 94–104, Nov. 2018.

[13] J. Pongfai and W. Assawinchaichote, "Self-tuning PID parameters using NN-GA for brush DC motor control system," in *Proc. 14th Int. Conf. Electr. Eng./Electron., Comput., Telecommun. Inf. Technol. (ECTI-CON)*, Jun. 2017, pp. 111–114.

[14] G. Lin and G. Liu, "Tuning PID controller using adaptive genetic algorithms," in *Proc. 5th Int. Conf. Comput. Sci. Educ.*, Aug. 2010, pp. 519–523.

[15] I. Chiha, N. Liouane, and P. Borne, "Tuning PID controller using multiobjective ant colony optimization," *Appl. Comput. Intell. Soft Comput.*, vol. 2012, pp. 1–7, Jan. 2012.

[16] M. S. Amiri, R. Ramli, and M. F. Ibrahim, "Hybrid design of PID controller for four DoF lower limb exoskeleton," *Appl. Math. Model.*, vol. 72, pp. 17–27, Aug. 2019.

[17] J. Wu, J. Gao, R. Song, R. Li, Y. Li, and L. Jiang, "The design and control of a 3DOF lower limb rehabilitation robot," *Mechatronics*, vol. 33, pp. 13–22, Feb. 2016.

[18] A. K. Tanyildizi, O. Yakut, and B. Tasar, "Mathematical modeling and control of lower extremity exoskeleton," *Biomed. Res.*, vol. 29, no. 9, pp. 1947–1952, 2018.

[19] M. Gilbert, X. Zhang, and G. Yin, "Modeling and design on control system of lower limb rehabilitation exoskeleton robot," in *Proc. 13th Int. Conf. Ubiquitous Robots Ambient Intell. (URAI)*, Aug. 2016, pp. 348–352.

[20] C. Uchegbu, I. Eneh, M. Ekwuribe, and C. Ugwu, "Remoldelling of PID controller based on an artificial intelligency (neural network)," *Amer. J. Sci., Eng. Technol.*, vol. 1, no. 2, pp. 20–26, 2016.

[21] B. V. Murthy, Y. P. Kumar, and U. R. Kumari, "Application of neural networks in process control: Automatic/online tuning of pid controller gains for ±10% disturbance rejection," in *Proc. IEEE Int. Conf. Adv. Commun. Control Comput. Technol. (ICACCCT)*, Apr. 2012, pp. 348–352.

[22] W. Lu, J. H. Yang, and X. D. Liu, "The PID controller based on the artificial neural network and the differential evolution algorithm," *J. Comput.*, vol. 7, no. 10, pp. 2368–2375, Oct. 2012.

[23] M. Aamir, "On replacing PID controller with ANN controller for DC motor position control," 2013, vol. 2, no. 1, pp. 21–29, *arXiv:1312.0148*. [Online]. Available: http://arxiv.org/abs/1312.0148

[24] R. Hernández-Alvarado, L. G. García-Valdovinos, T. Salgado-Jiménez, A. Gómez-Espinosa, and F. Fonseca-Navarro, "Neural network-based self-tuning PID control for underwater vehicles," *Sensors*, vol. 16, no. 9, p. 1429, Sep. 2016.

[25] N. Merayo, D. Juárez, J. C. Aguado, I. de Miguel, R. J. Durán, P. Fernández, R. M. Lorenzo, and E. J. Abril, "PID controller based on a self-adaptive neural network to ensure QoS bandwidth requirements in passive optical networks," *J. Opt. Commun. Netw.*, vol. 9, no. 5, pp. 433–445, May 2017.

[26] M. S. Amiri, R. Ramli, and M. F. Ibrahim, "Initialized model reference adaptive control for lower limb exoskeleton," *IEEE Access*, vol. 7, pp. 167210–167220, 2019.

[27] N. Mohan, T. M. Undeland, and W. P. Robbins, *Power Electronics: Converters, Applications, and Design*. Hoboken, NJ, USA: Wiley, 2003.

[28] M. H. Rashid, *Power Electronic Circuits, Devices, and Applications*. Pensacola, FL, USA: Univ. West Florida, 2004.

[29] G. Si, W. Huang, G. Li, F. Xu, M. Chu, and J. Liu, "Research progress on compliant characteristics of lower extremity exoskeleton robots," *Sheng wu yi xue gong cheng xue za zhi=J. Biomed. Eng.=Shengwu yixue gongchengxue zazhi*, vol. 36, no. 1, pp. 157–163, 2019.

[30] W. Deng, I. Papavasileiou, Z. Qiao, W. Zhang, K.-Y. Lam, and S. Han, "Advances in automation technologies for lower extremity neurorehabilitation: A review and future challenges," *IEEE Rev. Biomed. Eng.*, vol. 11, pp. 289–305, 2018.

[31] M. Bošnak and I. Škrjanc, "Embedded control system for smart walking assistance device," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 25, no. 3, pp. 205–214, Mar. 2017.

[32] T.-Y. Wang and C.-D. Chang, "Hybrid fuzzy PID controller design for a mobile robot," in *Proc. IEEE Int. Conf. Appl. Syst. Invention (ICASI)*, Apr. 2018, pp. 650–653.

[33] W. S. Aboud, S. M. Haris, and Y. Yaacob, "Advances in the control of mechatronic suspension systems," *J. Zhejiang Univ., Sci. C*, vol. 15, no. 10, pp. 848–860, 2014.

[34] J.-Y. Jung, W. Heo, H. Yang, and H. Park, "A neural network-based gait phase classification method using sensors equipped on lower limb exoskeleton robots," *Sensors*, vol. 15, no. 11, pp. 27738–27759, 2015.

[35] M. F. Møller, "A scaled conjugate gradient algorithm for fast supervised learning," *Neural Netw.*, vol. 6, no. 4, pp. 525–533, Nov. 1993.

[36] M. T. Hagan and M. B. Menhaj, "Training feedforward networks with the Marquardt algorithm," *IEEE Trans. Neural Netw.*, vol. 5, no. 6, pp. 989–993, Nov. 1994.

[37] N. S. Nise, *Control Systems Engineering*. Hoboken, NJ, USA: Wiley, 2020.

[38] R. Kumar, S. Srivastava, and J. R. P. Gupta, "Artificial neural network based PID controller for online control of dynamical systems," in *Proc. IEEE 1st Int. Conf. Power Electron., Intell. Control Energy Syst. (ICPEICES)*, Jul. 2016, pp. 1–6.

[39] M. J. Er and Y. L. Sun, "Hybrid fuzzy proportional-integral plus conventional derivative control of linear and nonlinear systems," *IEEE Trans. Ind. Electron.*, vol. 48, no. 6, pp. 1109–1117, Dec. 2001.

KARRAR H. AL-WAELI received the bachelor's degree in mechanical and mechatronics engineering from Sohar University, Oman, in 2018, and the master's degree in mechanical and control systems engineering from the National University of Malaysia, Malaysia, in 2019, where he is currently pursuing the Ph.D. degree in mechanical and control systems engineering. His research interests include automation, robotics, control systems, optimization, and artificial intelligence.

RIZAUDDIN RAMLI received the bachelor's degree in mechanical engineering from Kyoto University, Japan, in 1997, and the master's degree in mechanical and system engineering and the Ph.D. degree in manufacturing systems from Gifu University, Japan, in 2005 and 2008, respectively. He is currently an Associate Professor with the Department of Mechanical and Materials, Faculty of Engineering and Built Environment, National University of Malaysia. His research interests include intelligent manufacturing systems, robotics, control, and artificial intelligence.

SALLEHUDDIN MOHAMED HARIS (Member, IEEE) received the B.Eng. degree in manufacturing systems engineering from the University of Leeds, U.K., in 1993, the M.Sc. degree in mechatronics from the University of London (King's College London), in 1996, and the Ph.D. degree in electronics and electrical engineering from the University of Southampton, U.K., in 2006. He is currently an Associate Professor with the Department of Mechanical and Manufacturing Engineering, Universiti Kebangsaan Malaysia. His fields of specialization are in control, mechatronics, and cyber-physical systems. His current research interests include autonomous robotic systems, magnetic gear systems, assistive technologies for people with disabilities, and computer algebra applications in control systems.

ZULIANI BINTI ZULKOFFLI received the master's degree in CAD/CAM field through the image processing method from Universiti Sains Malaysia, and the master's degree in material science in mechanical engineering from Universiti Kebangsaan Malaysia. She is currently a Lecturer with UCSI University, Kuala Lumpur. Her deep research interests include explore artificial intelligence in computer vision applications in wide deployment platforms, instrumentation, machine learning, information technology, and smart manufacturing.

MOHAMMAD SOLEIMANI AMIRI received the bachelor's degree in mechanical engineering from Qazvin Islamic Azad University, Iran, in 2011, the master's degree in mechanical and control system engineering from Ahrar Institute of Technology and Higher Education, Iran, and the Ph.D. degree in mechanical and control systems engineering from the National University of Malaysia, in 2020. He currently works as a Postdoctoral Researcher with the National University of Malaysia. His research interests include automation, optimization, robotics, and control systems.