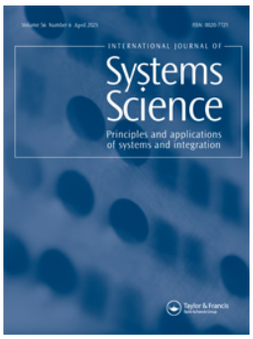


International Journal of Systems Science

ISSN: 0020-7721 (Print) 1464-5319 (Online) Journal homepage: www.tandfonline.com/journals/tsys20

Adaptive neuro-fuzzy inference system based active force control with iterative learning for trajectory tracking of a biped robot

Hanyi Huang, Adetokunbo Arogbonlo, Samson Yu, Lee Chung Kwek & Chee Peng Lim

To cite this article: Hanyi Huang, Adetokunbo Arogbonlo, Samson Yu, Lee Chung Kwek & Chee Peng Lim (2025) Adaptive neuro-fuzzy inference system based active force control with iterative learning for trajectory tracking of a biped robot, International Journal of Systems Science, 56:6, 1171-1188, DOI: 10.1080/00207721.2024.2420069

To link to this article: https://doi.org/10.1080/00207721.2024.2420069

Published online: 28 Oct 2024.

INTERNATIONAL JOURNAL OF SYSTEMS SCIENCE
2025, VOL. 56, NO. 6, 1171–1188
https://doi.org/10.1080/00207721.2024.2420069

Taylor & Francis
Taylor & Francis Group

Adaptive neuro-fuzzy inference system based active force control with iterative learning for trajectory tracking of a biped robot

Hanyi Huang[a], Adetokunbo Arogbonlo[b], Samson Yu[c], Lee Chung Kwek[d] and Chee Peng Lim[e]

[a]Institute for Intelligent Systems Research and Innovation, Deakin University, Waurn Ponds, Australia; [b]Swinburne Research Office, Swinburne University of Technology, Hawthorn, Australia; [c]School of Engineering, Deakin University, Waurn Ponds, Australia; [d]Faculty of Engineering and Technology, Multimedia University, Melaka, Malaysia ; [e]Department of Computing Technologies, Swinburne University of Technology, Hawthorn, Australia

ABSTRACT

We investigate the integration of the active force control (AFC) scheme and the adaptive neuro-fuzzy inference system (ANFIS) as an intelligent controller algorithm to address trajectory-tracking problems in robotic systems, The AFC-ANFIS model exploits iterative learning (IL) to improve the tracking performance based on its trained model. The ANFIS parameters are tuned using both particle swarm optimisation (PSO) and beetle antennae search (BAS) algorithms. The simulation results of two different robots, i.e. a five-link biped robot and a PUMA 560 robot arm, indicate that the proposed AFC-ANFIS controller performs well for trajectory tracking and disturbances rejection. The AFC-ANFIS performance is evaluated and compared with those from other controllers using the average tracking error (ATE) metric. The comparative results reveal that AFC-ANFIS offers a viable approach with a rapid training process to undertaking trajectory tracking and disturbance rejection tasks.

ARTICLE HISTORY
Received 24 April 2024
Accepted 17 October 2024

KEYWORDS
Active force control; adaptive neuro-fuzzy inference system; biped robot; inertia matrix; iterative learning

1. Introduction

Accurate and efficient trajectory tracking is an important requirement in many robotic applications, e.g. surgery robots (Li et al., 2015), autonomous underwater vehicles (Liu et al., 2019), and biped robots (Yang et al., 2022). In a minimally invasive procedure, surgical robots are required to perform every movement precisely to avoid risks (Li et al., 2015). For safe navigation, autonomous underwater vehicles should follow the pre-determined path to avoid obstacles or other unsafe marine conditions (Liu et al., 2019). As an indispensable part of automated manufacturing lines, industrial robots require high precision to complete tasks, such as spraying and welding (Li et al., 2021). Robots with a high degrees of freedom (DoF) are capable of complex movements that enable them to perform well with high flexibility. In industrial applications, 6 DoF robots such as PUMA 560 (Li et al., 2019), KUKA LBR IIWA (Khan et al., 2022) and ABB IRB 120 (Li et al., 2023), are ubiquitous because of their versatility. Similarly, a biped robot needs to maintain its balance by following a pre-defined path (Yang et al., 2022), because failure to achieve successful trajectory tracking can result in a fall of the robot. Hence trajectory control with high accuracy and satisfactory performance is essential for a biped robot to complete tasks successfully.

A biped robot is designed to replicate human walking ability. Research on biped robots helps study how humans maintain balance and develop control strategies for biped robots. These include minimising imbalance from compensatory movements (Huang et al., 2022), creating gait pattern generation methods for biped robots to walk omnidirectionally on inclined surfaces (Yu et al., 2016), and introducing a strategy that involves torso inclination to enable walking with unactuated ankles (Geng, 2014).

For the trajectory tracking problem, a wide range of control strategies have been proposed, such as proportional-integral-derivative (PID) controller (Zhao et al., 2019), model predictive control (MPC) (Baca et al., 2018) and force control (Ider, 2000). MPC is adaptable and capable of determining good solutions that adhere to both hard and flexible constraints.

CONTACT Hanyi Huang hanyi.huang@deakin.edu.au Institute for Intelligent Systems Research and Innovation, Deakin University, 75 Pigdons Rd, Waurn Ponds, VIC 3216, Australia

MPC has its limitations, including high computational burden and stringent model accuracy requirement (Razmi et al., 2022). The decisions of force control are based on feedback received during interactions with the environment. However, force control needs to cope with position control, as an accurate position of the object needs to be known, and the control object has to be rigid (Johansson et al., 2014). Additionally, neural network controllers have been proposed in the literature, including the zeroing neural network model (Zhang et al., 2024), recurrent neural network (Lu et al., 2019) and dual neural network model (Li et al., 2019). Despite their effectiveness, these neural network-based controllers operate on a 'black box' nature, lack insight into the function they approximate, and also require large amounts of data samples along with high computational costs.

The ability to negate disturbances is crucial during operations, as actual movement consistently encounters resistance. Various effective methods have been developed to address the challenge of disturbance rejection, such as disturbance observer (Sini & Ananthan, 2022), active disturbance rejection control (ADRC) (Chen et al., 2016), and sliding mode control (SMC) (Zhai & Li, 2022). However, each method comes with its drawbacks. As an example, a disturbance observer demands an accurate control model, which needs to be re-designed for different control objects. ADRC is criticised for its delayed response, inefficient use of model information and extensive effort for its design (Zhang et al., 2021). For traditional SMC, it necessitates high gains to achieve a quick convergence, leading to the potential for actuator saturation and serious chattering (Zhang et al., 2022).

This paper focuses on a method that excels in both trajectory tracking and disturbance rejection, known as Active Force Control (AFC) (Hewit & Burdess, 1981). Proposed by Hewit and Burdess (1981), AFC stands out for its innovative approach to combating resistance and maintaining the control robot in its predefined trajectory. AFC has been applied to a wide range of applications, starting from basic robotics, like planar parallel manipulators (Noshadi et al., 2012), to autonomous vehicles, such as underwater vehicles (Yuan et al., 2020), wheeled mobile robots (Ali et al., 2021), vehicle active suspension systems (Priyandoko et al., 2009), and unmanned aerial vehicles (UAVs) (Abdelmaksoud et al., 2020, 2021).

In AFC, the fundamental approach to compensating disturbances involves estimating the IN values. To achieve this, various learning algorithms have been utilised, including neural networks (NNs) (Priyandoko et al., 2009), fuzzy logic (FL) (Abdelmaksoud et al., 2020, 2021), crude approximation (CA) (Hassan et al., 2010, june), IL (Kwek et al., 2003), and reinforcement learning (RL) (Huang et al., 2024). However, these learning methods are normally integrated into the AFC loop for estimating IN values, which comes with limitations. It is noted that AFC-FL cannot entirely eliminate noise in constrained settings due to the overlapping ranges of linguistic variables (Ali et al., 2021). Moreover, as discussed in Kwek et al. (2003), AFC-CA requires an accurate dynamic model, making the IN tuning process lengthy. Additionally, though RL shows satisfactory results in minimising tracking errors in completing tasks, the training process is notably time-consuming. Hence, it is necessary to investigate a better learning technique for improving the ability to estimate the IN with an AFC-based controller.

AFC-IL represents a significant advancement, combining AFC with IL. According to (Kwek et al., 2003), AFC-IL could automatically perform and tune the inertia estimates without prior knowledge of the dynamic model of a biped robot. The achieved results were comparable to those of AFC-CA (which requires an exact dynamic model) with the added benefit of rapid execution. Nonetheless, the AFC-IL performance relies on manually adjustable parameters like learning parameters, impacting its outcomes. In order to enhance the performance without time-consuming tuning, we explore possibilities of integrating other machine learning algorithms with AFC-IL.

ANFIS emerges as a promising method that has been successfully applied to various fields, including gas tungsten arc welding (Liu & Zhang, 2015), flood prediction (Ghose et al., 2022), and power transformer fault diagnosis (Khan et al., 2015). Introduced by Jang (1993), ANFIS adapts itself using numerical input/output data sets to improve its performance by leveraging the benefits of both FL and NN models. ANFIS maps the input data into the desired outputs by utilising FL across an NN network of interconnected neurons with appropriate weights (Al-Hmouz et al., 2012). Specifically, ANFIS employs NN learning

methods to learn the parameters of a Fuzzy Inference System (FIS), i.e. tuning the related membership function (MF) parameters. The advantages of ANFIS include adaptive rule-changing capability, rapid convergence, ease of implementation, and ease of incorporating both linguistic and numeric knowledge for problem-solving with a wide scope of MF choices. Compared with FL and NN, ANFIS has simplified the parameters tuning process, and it can adjust some parameters during operation, but it still requires careful management of parameters, including the number of membership functions and epochs.

Consequently, we investigate a method using swarm intelligence to optimise the ANFIS parameters, namely the particle swarm optimisation (PSO) and beetle antennae search (BAS) algorithms. PSO is a swarm intelligent algorithm proposed by Kennedy and Eberhart (Kennedy & Eberhart, 1995) based on bird flocking or fish schooling behaviours. PSO is useful for function optimisation by iteratively improving a candidate solution using a collection of potential solutions (multiple particles) to form a swarm, where each particle represents a potential solution (Bansal, 2019). PSO has been utilised in many applications, including demand response applications in a home energy management system (Huang et al., 2018) and solar photovoltaic systems (Li et al., 2019). On the other hand, inspired by beetle foraging behaviours, BAS exploits a single agent to iteratively search the solution space. BAS has been used in optimising the gains of PID for control stability (Khan et al., 2022), and also for optimising trajectory generation and robot control tasks (Khan et al., 2021).

Hence, we design a new control method that integrates ANFIS with AFC-IL for trajectory tracking of robotic systems. To improve the performance of the ANFIS model, we use PSO and BAS for its parameters tuning process. The research motivations include: firstly, ANFIS requires a relatively small training data set; secondly, its training process is rapid; thirdly, it can adjust its parameters during training and the results change clearly after training; fourthly, its performance can be enhanced by increasing the number of MFs and epochs. However, improper increases in MFs can lead to overfitting. We use the AFC-IL outputs for ANFIS training and formulate an accurate AFC-ANFIS model for trajectory-tracking applications.

This paper introduces an AFC-ANFIS controller designed to enable a five-link biped robot to walk on a flat surface and climb stairs (as depicted in Figure 1) while counteracting various types of disturbances. A comparative analysis of the ATE is conducted to evaluate the performance in trajectory tracking between the AFC-ANFIS and AFC-IL controllers.

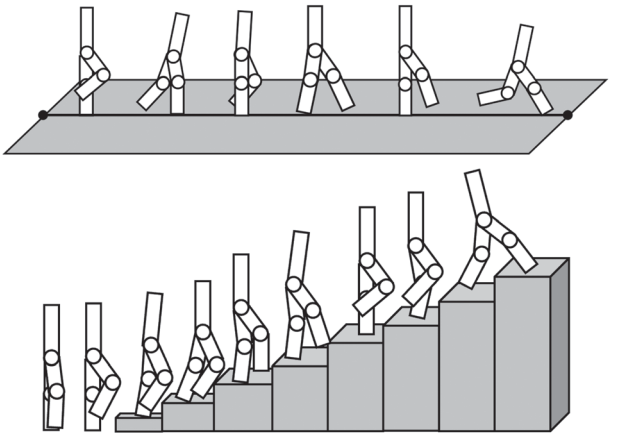
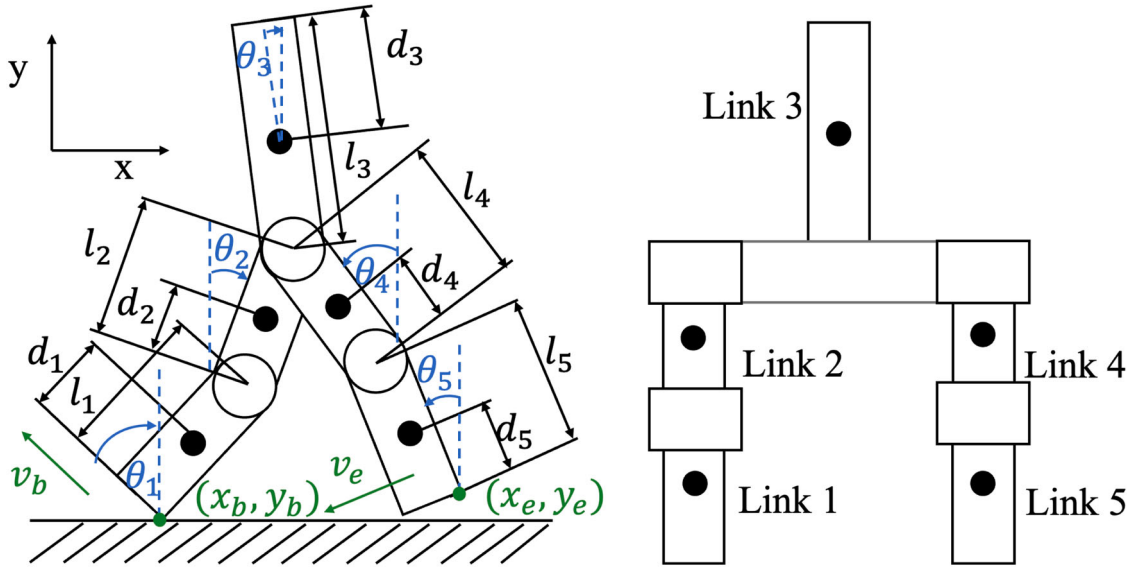


Figure 1. Biped robot walks and climbs stairs.

The main contributions of this paper are summarised as follows:

- A new AFC-ANFIS controller is proposed, which combines the ANFIS algorithm with an AFC-based controller to enhance efficiency and accuracy in control applications;
- Based on the AFC-ANFIS controller, we add an IL algorithm that allows the model to progress by itself through previously trained results.
- The AFC-ANFIS controller is applied to a five-link biped robot to complete two tasks, namely walking on a horizontal surface and climbing stairs.
- The proposed AFC-ANFIS controller is applied to a PUMA 560 robot arm for trajectory tracking, and the results are compared with those from a benchmark controller.

This paper is organised as follows. Section 2 describes the kinematics and dynamics model of the robotic systems, including a five-link biped robot and a PUMA 560 robot arm. In Section 3, the detailed AFC-ANFIS control strategy is explained. Section 4 explains the simulation environment, and training of the ANFIS model. Section 5 presents the simulation results of AFC-ANFIS with an extensive performance comparison study. Section 6 gives concluding remarks of this study.



(a) Sagittal plane (b) Coronal plane

Figure 2. Five-link biped robot. (a) Sagittal plane (b) Coronal plane.

Table 1. Robot parameters.

Symbols	Notations
i	Link number
l_i	Length of link
lc_i	Location of centre of mass
θ_i	Angle of link with respect to an axis
(x_b,y_b)	Position of the point of support
(x_e,y_e)	Position of the end effector
v_b	Velocity of the point of support
v_e	Velocity of the end effector

Table 2. Link parameters.

Link Number (i)	3	2,4	1,5
Mass m_i (kg)	14.790	5.280	2.230
Length l_i (m)	0.486	0.302	0.332
Location of centre of mass d_i (m)	0.282	0.236	0.189
Moment of inertia I_i (kg · m^2)	0.033	0.033	0.033

2. Robot model

2.1. Biped robot model

As shown in Figure 2, the biped robot comprises five links, including the torso (link 3) and the leg structure that encompasses the left leg (links 1 and 2) and the right leg (links 4 and 5). Links 1 and 5 represent the lower legs, while links 2 and 4 represent the upper legs. Since the biped robot does not have ankles and feet, the lower leg structure is considered sufficiently large enough to maintain its balance. The links are connected through four rotational joints (as shown in Figure 2(b)), powered by individual DC motors. The friction of these joints is neglected. The movement of the biped is restricted to the sagittal plane due to the planar rotation of the joints. The notations for the robot parameters are presented in Table 1.

The tracking error (TE_i) refers to the trajectory error in link angular positions. In this paper, the performance of various AFC-based control strategies is assessed using the ATE, the sum of absolute errors of the trajectories of four links divided by the number of links. Note that the calculation excludes the torso (link 3) tracking error, focussing on the leg structure for walking and stair-climbing tasks. For evaluation purposes, the total number of steps is denoted by K, calculated as $K = T_f/T_s$, where T_f is the finish time, and T_s is the sampling time, and j indicates the j^{th} step.

$$ATE = \frac{1}{K}\sum_{j=1}^{K}\frac{\sum_{i=1}^{4}|TE_i|}{4}. \quad (1)$$

2.1.1. Kinematic model

Adopted from Lum et al. (1999), the link specifications are detailed in Table 2. Figure 2(a) marks the centre of mass location (d_i) and illustrates the directions for the velocity of both the free end (v_e) and supporting end (v_b). Additionally, θ_i is the angle of link i with respect to the vertical surface, where $i = 1, 2, 3, 4, 5$ represents the number of each link.

As depicted in Figure 2(a), the robot's free end point coordinate is given as (x_e, y_e), and the supporting link's coordinates is (x_b, y_b). The relationship between

the link's supporting point and the free end point is described by Equations (2) and (3).

$$x_e = x_b + l_1 \sin(\theta_1) + l_2 \sin(\theta_2) + l_4 \sin(\theta_4) + l_5 \sin(\theta_5), \quad (2)$$

$$y_e = y_b + l_1 \cos(\theta_1) + l_2 \cos(\theta_2) - l_4 \cos(\theta_4) - l_5 \cos(\theta_5). \quad (3)$$

The kinematic equation for the biped robot is obtained as follows,

$$v_e = \begin{bmatrix} \dot{x}_e \\ \dot{y}_e \end{bmatrix} = \begin{bmatrix} l_1 \cos\theta_1 \\ -l_1 \sin\theta_1 \end{bmatrix} \dot{\theta}_1 + \begin{bmatrix} l_2 \cos\theta_2 \\ -l_2 \sin\theta_2 \end{bmatrix} \dot{\theta}_2 + \begin{bmatrix} l_4 \cos\theta_4 \\ l_4 \sin\theta_4 \end{bmatrix} \dot{\theta}_4 + \begin{bmatrix} l_5 \cos\theta_5 \\ l_5 \sin\theta_5 \end{bmatrix} \dot{\theta}_5. \quad (4)$$

2.1.2. Dynamic model

This research simplifies the biped robot's motion by focussing exclusively on the single-legged support phase, where one leg is grounded while the other is still in the air. Throughout the movement, it consistently maintains its single-leg support. Under these conditions, the robot's dynamics are characterised as follows,

$$D(\theta)\ddot{\theta} + h(\theta, \dot{\theta}) + G(\theta) = T_\theta. \quad (5)$$

In this context, T_θ represents the vector of driving torques in relation to θ_i, $[\theta_1\ \theta_2\ \theta_3\ \theta_4\ \theta_5]^T$, the vector of link angles. Additionally, $D(\theta)$ denotes a 5 × 5 symmetric positive-definite inertia matrix, while $h(\theta, \dot{\theta})$ is a 5 × 1 vector merging both Coriolis and centripetal torques. $G(\theta)$ is identified as a 5 × 1 vector of gravitational torques. The dynamic equation (Equation (5)), including $D(\theta)$, $h(\theta, \dot{\theta})$ and $G(\theta)$, are derived using the Lagrange equation. The comprehensive dynamics of the five-link biped robot are detailed in Tzafestas et al. (1996).

Links 1 and 2 within this model serve as the supporting leg. As the free leg, comprising links 4 and 5, reaches the ground, the supporting leg is promptly lifted. This transition from one supporting leg to the other occurs seamlessly at the end of each step. Given angle θ_1 at the point of ground contact is indirectly controlled through gravity, notation q is introduced in Kwek (2003) to describe the relative angular deviations of the relevant links.

$$q_0 = \theta_1,\ \ q_1 = \theta_1 - \theta_2,\ \ q_2 = \theta_2 - \theta_3, \quad q_3 = \theta_3 + \theta_4,\ \ q_4 = \theta_4 - \theta_5, \quad (6)$$

where $q_i,\ i = 0, 1, 2, 3, 4$ is a 5 × 1 vector.

Hence, the dynamic model is converted from Equations (5) to (7), which corresponds to q_i.

$$D_q(q)\ddot{q} + h(q, \dot{q}) + G(q) = T_q, \quad (7)$$

where $[q_1\ q_2\ q_3\ q_4]^T$ denotes the vector of link angles, while T_q represents the vector of driving torques relative to q_i. $D_q(q)$ refers to a 5 × 5 symmetric positive-definite inertia matrix. Additionally, $h(q, \dot{q})$ signifies a 5 × 1 matrix of Coriolis and centripetal torques, and $G(q)$ is a 5 × 1 matrix of gravitational torques. The motion equations are uniform for both legs, thus they apply to the support of either the left or right leg.

2.2. PUMA 560 robot arm

2.2.1. Kinematic model

As shown in Figure 3, the PUMA 560 robot arm consists of six joints. Three different joints control the three largest components, while the remaining three joints control the smaller end of the robotic arm.

The Denavit-Hartenberg (DH) parameters of the PUMA 560 robot arm are shown in Table 3. Using the DH parameters, we can obtain the transformation matrix of each joint (Craig, 2018; Spong & Vidyasagar, 2008), as follows.

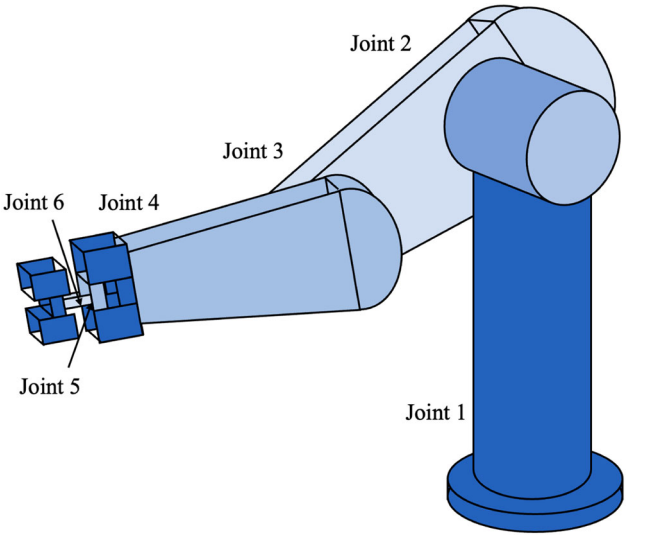


Figure 3. Schematic diagram of PUMA 560.

Table 3. DH parameters of PUMA 560.

Kinematics	θ (rad)	a (m)	d (m)	α (rad)
Joint 1	θ_1	0	0	0
Joint 2	θ_2	0	0	$\pi/2$
Joint 3	θ_3	0.4318	0.15005	0
Joint 4	θ_4	0.0203	0.4318	$-\pi/2$
Joint 5	θ_5	0	0	$\pi/2$
Joint 6	θ_6	0	0	$-\pi/2$

$$ {}^{0}T_n = \begin{bmatrix} R & p \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} & r_{13} & p_x \\ r_{21} & r_{22} & r_{23} & p_y \\ r_{31} & r_{32} & r_{33} & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{8} $$

$$ = \begin{bmatrix} \cos\theta_n & -\sin\theta_n \cos\alpha_n & \sin\theta_n \sin\alpha_n & a_n \cos\theta_n \\ \sin\theta_n & \cos\theta_n \cos\alpha_n & -\cos\theta_n \sin\alpha_n & a_n \sin\theta_n \\ 0 & \sin\alpha_n & r_{33} & d_n \\ 0 & 0 & 0 & 1 \end{bmatrix}, \tag{9} $$

where R is a 3 × 3 rotation matrix, p is a 3 × 1 column vector that stands for the position of the robot end effector, and n is the number of joints.

Finally, the position of the end effector of PUMA 560 using (p_x, p_y, p_z) from ${}^{0}T_6$ is as follows:

$$ {}^{0}T_6 = {}^{0}T_1\, {}^{1}T_2\, {}^{2}T_3\, {}^{3}T_4\, {}^{4}T_5\, {}^{5}T_6. \tag{10} $$

The inverse kinematics of the PUMA model have been derived in detail by Lee and Ziegler (1983). Using the inverse kinematic model, we can transfer the end-effector position and rotation to each joint angle.

2.2.2. Dynamic model

The comprehensive description of the dynamic model of the PUMA 560 robot can be found in Armstrong et al. (1986). Its dynamic model can be generally represented as follows:

$$ M(q)\ddot{q} + c(q, \dot{q}) + g(q) = T_q, \tag{11} $$

where T_q represents the 6 × 1 vector of torques, $M(q)$ refers to a 6 × 6 symmetric positive-definite inertia matrix. Additionally, $c(q, \dot{q})$ represents a 6 × 1 matrix of Coriolis and centripetal torques, and $g(q)$ is a 6 × 1 matrix of gravitational torques. $q = [\theta_1 \ldots, \theta_6]^T$ denotes the vector of joint angles.

3. The AFC-ANFIS controller

The primary aim of the AFC is to ensure stability by rejecting disturbances. Central to the AFC approach is IN estimation, integrated into the feed-forward loop with reference angular acceleration, disturbances and feedback angular acceleration. The reference angular acceleration is obtained via a PD controller. Systems feedback, including torques, angular positions, velocities and accelerations, is accurately measured through transducers. An illustrative schematic of the AFC-ANFIS controller for a five-link biped robot is depicted in Figure 4.

3.1. AFC structure

In accordance with Newton's second law for rotational dynamics, the torque (T) applied to an object is determined by multiplying the object's moment of inertia with its angular acceleration ($\ddot{q}$), as illustrated in Equation (12). Consequently, the object moves in the direction of the applied torque. Thus, the essence of

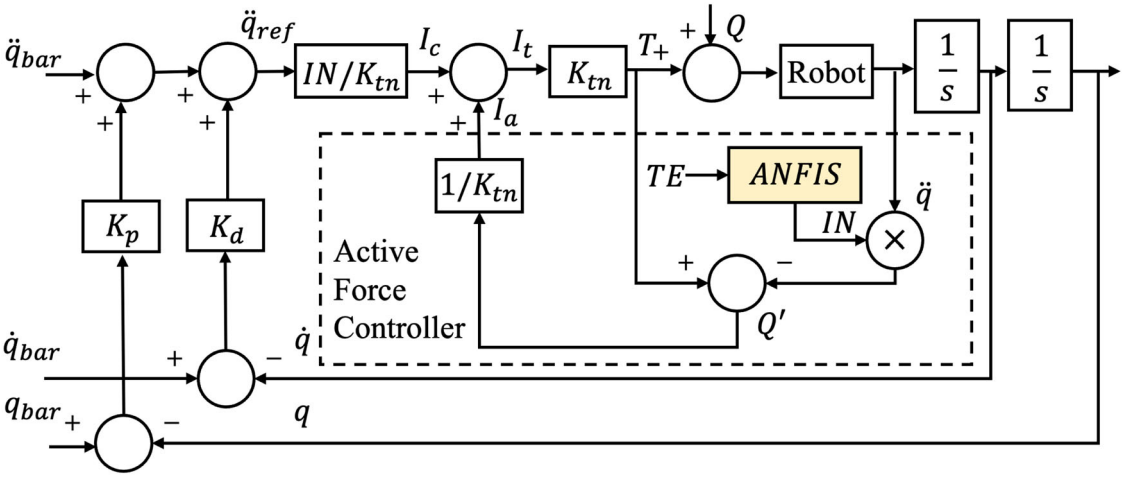


Figure 4. The schematic diagram of AFC-ANFIS scheme.

AFC revolves around the continuous computation and IN estimation within the control system.

$$T = IN \cdot \ddot{q}. \tag{12}$$

Considering the external disturbance Q introduced to the system, as depicted in Figure 4, we can derive

$$T + Q = IN \cdot \ddot{q}. \tag{13}$$

The estimated disturbance, arising from Coriolis, centrifugal, and frictional forces, can be expressed as follows,

$$Q' = IN \cdot \ddot{q} - T, \tag{14}$$

where Q' is calculated to counteract the actual disturbances Q, IN represents the estimated inertia matrix, $\ddot{q}$ denotes the measured angular acceleration, and T is the resultant output torque.

Upon calculating Q' and deriving the reference angular acceleration $\ddot{q}_{\text{ref}}$ from the PD module, an adjustable signal I_{a} is then supplied to the actuator system.

$$I_{\text{a}} = Q' \cdot \frac{1}{K_{\text{tn}}} \tag{15}$$

$$I_{\text{c}} = \ddot{q}_{\text{ref}} \cdot \frac{IN}{K_{\text{tn}}} \tag{16}$$

$$I_{\text{t}} = I_{\text{a}} + I_{\text{c}}, \tag{17}$$

where I_{a} is the compensating current vector representing the disturbance, I_{c} is the command current vector (which is the original control signal), and I_{t} is the armature current of the torque motor, while K_{tn} is the motor constant, provided by the motor maker.

Finally,

$$T = I_t \cdot K_{\text{tn}}, \tag{18}$$

where T is the applied control torque input to the robot dynamics.

The AFC modifies the control signal in response to the tracking error observed between the desired and actual angular positions and velocities. The tracking error is the input to ANFIS, and a PD controller adjusts the reference angular acceleration $\ddot{q}_{\text{ref}}$ according to the differences.

3.2. PD module

According to Yuan et al. (2020), AFC cannot operate independently, as it requires a complementary feedback controller in the outermost control loop for essential support. Therefore, a PD controller is utilised to determine the reference angular acceleration, ensuring an efficient operation of the AFC-based controller.

The reference angular acceleration can be determined using Equation (19),

$$\begin{aligned}\ddot{q}(t) = \ddot{q}_{bar}(t) + K_d\left(\dot{q}_{bar}(t) - \dot{q}(t)\right) \\ + K_p\left(q_{bar}(t) - q(t)\right).\end{aligned} \tag{19}$$

Equation (19) can be reformulated as follows,

$$\begin{aligned}\frac{d^2}{dt^2}(q_{bar} - q) + K_d\left(\dot{q}_{bar}(t) - \dot{q}(t)\right) \\ + K_p\left(q_{bar}(t) - q(t)\right) = 0,\end{aligned} \tag{20}$$

where K_d and K_p are gain matrices with the size of 5×5. Note that q_{bar} is the pre-set angular position, $\dot{q}_{bar}$ is the desired angular velocity, and $\ddot{q}_{bar}$ is the predetermined angular acceleration for link $i = 1, 2, 3, 4, 5$, which have the size of 5×1. In addition, q, $\dot{q}$ and $\ddot{q}$ represent the actual angular position, velocity, and acceleration with the size of 5×1.

To ensure that the link tracking error of ($TE = q_{bar} - q$) converges to zero, K_d and K_p need to be chosen such that the characteristic roots of Equation (20) possess negative real parts. Following Tzafestas et al. (1996), the PD gains are selected in a manner that results in a critically-damped closed-loop performance.

$$K_d = \text{diag}[2\lambda] \tag{21}$$

$$K_p = \text{diag}[\lambda^2], \tag{22}$$

where λ represents the desired bandwidth in the form of $\lambda = \left[\lambda_1, \lambda_2, \lambda_3, \lambda_4, \lambda_5\right]^{\mathsf{T}}$, with $\lambda_1 = \lambda_2 = \lambda_3 = \lambda_4 = \lambda_5$.

3.3. ANFIS module

3.3.1. General ANFIS structure

ANFIS is a 5-layer neuro-fuzzy model with supervised learning. The general ANFIS structure is depicted in Figure 5. A rectangular node indicates that its output depends on the node parameters, while a round node represents a fixed node. All nodes within the same layer share an identical activation function. The output of one layer serves as the input to the subsequent layer.

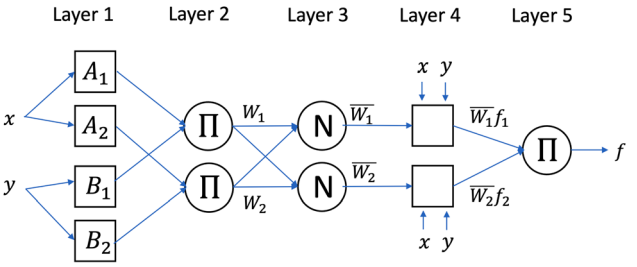


Figure 5. General ANFIS structure.

The mathematical functions in each layer are explained, as follows. The output of node j in layer l as $O_{l,j}$. Layer 1 consists of 4 nodes (A_1, A_2, B_1, B_2). Each node represents a fuzzy set, with x and y being its inputs. Its output is defined as follows,

$$O_{1,j} = \mu_{A_j}(x), \quad \text{for } j = 1, 2 \text{ or} \tag{23}$$

$$O_{1,j} = \mu_{B_{j-2}}(y), \quad \text{for } j = 3, 4 \tag{24}$$

The outputs of layer 1 are the membership values associated with the premise part. The MFs of A_j and B_j can be any suitable parameterised MFs, including triangular, trapezoidal, Gaussian, generalised Bell, and sigmoidal functions. As an example, A_j can be a generalised Bell function, as follows,

$$\mu_{A_j}(x) = \frac{1}{1 + \left(\frac{x - c_j}{a_j}\right)^{2b_j}}, \tag{25}$$

where a_j, b_j, c_j is the premise parameter set, which is updated during the training process.

Additionally, the number of MFs sets the number of nodes in layer 1. A greater number of MFs allows ANFIS to capture more complex relationships and patterns from the provided database, potentially enhancing the accuracy of the results. However, this increase in membership functions also heightens the risk of overfitting. Overfitting occurs when the model begins to interpret unnecessary noise as important, resulting in less reliable results.

For layer 2, the outputs are the products of the input, i.e.

$$O_{2,j} = w_j = \mu_{A_j}(x) \times \mu_{B_j}(y), \quad \text{for } j = 1, 2. \tag{26}$$

Layer 3 nodes perform the function of normalisation as follows,

$$O_{3,j} = \overline{w_j} = \frac{w_j}{w_1 + w_2}, \quad \text{for } j = 1, 2. \tag{27}$$

Nodes in layer 4 are adaptive with the following activation function,

$$O_{4,j} = \overline{w_j} f_j = \overline{w_j}(p_j x + q_j y + r_j), \quad \text{for } j = 1, 2 \tag{28}$$

where p_j, q_j, r_j is the consequent parameters set, which is modified during the training process.

Layer 5 calculates the total output by summing up all the incoming signals, i.e.

$$O_{5,j} = \sum_j \overline{w_j} f_j = \frac{\sum_j w_j f_j}{\sum_j w_j}. \tag{29}$$

During ANFIS training, it updates both premise and consequent parameters. In the forward pass, the inputs are forwarded to produce an output, keeping the premise parameters constant and optimising the consequent parameters using least squares estimation combined with backpropagation. Conversely, the backward pass refines the MF parameters through error feedback to reduce the prediction errors, adjusting the premise parameters via gradient descent with the consequent parameters fixed. This dual-direction process refines both FL and NN components for accurate modelling.

3.3.2. Iterative learning on ANFIS

In this study, we apply IL steps based on the original ANFIS structure and propose an effective and efficient method to search for a good solution, which ultimately gives better tuning results for ANFIS. The steps are illustrated as follows,

(1) Data collection: As ANFIS requires specific input and output data pairs for training, compute TE and their respective IN outputs derived from the AFC-IL results.
(2) Initial setup: Choose the initial FIS and ANFIS training parameters. Various training options, such as clustering methods, types and numbers of MFs, and epoch numbers, are established.
(3) Iteration Preparation: Set the iteration counter (n) to zero and establish the total number of iterations (m).
(4) Training: Train the n^{th} iteration of ANFIS with the predetermined settings.
(5) Evaluate and generating Data: Evaluate the n^{th} iteration of the ANFIS model and generate new input/output sets.

(6) Iteration advancement: Increment n by 1 and compare it to m. If n is lower than m, take the latest ANFIS-generated input and output sets as the training data and repeat the training process from step 4; otherwise, proceed to the final step.
(7) Completion: Finalise and save all tuned ANFIS models. Compare the tested results among all ANFIS models and determine the best fine-tuned ANFIS model.

3.3.3. Parameters search on ANFIS

PSO constitutes a swarm intelligence search algorithm. The swarm comprises a set of randomly generated potential solutions, and each particle is a potential solution. The particle finds the best solution by manipulating its velocity and position Bansal (2019). The velocity and position update equations are shown as follows,

$$v_{id}^{t+1} = v_{id}^{t} + c_1 r_1 (p_{gd}^{t} - x_{id}^{t}) \tag{30}$$

$$x_{id}^{t+1} = x_{id}^{t} + v_{id}^{t+1}, \tag{31}$$

where $d = 1, 2 \cdots, D$ represents the dimension of the search space, t is the time step, i is the number of i^{th} particle of the swam, v is the velocity and x is the position, p is the previously best-visited position, g is the index of the best particle in the swam. c_1 and c_2 are constants, which are cognitive and social scaling parameters, r_1 and r_2 are random numbers in the range $[0, 1]$ drawn from a uniform distribution.

On the other hand, the BAS algorithm is based on the beetle's natural tendency to find food using two antennae. It follows the one with the stronger smell. The beetle keeps repeating this process until reaches the food. The optimisation process of BAS can be described as follows:

$$x_r = x^t + d^t \overrightarrow{b} \tag{32}$$

$$x_l = x^t - d^t \overrightarrow{b} \tag{33}$$

$$d^t = \delta d^t, \tag{34}$$

where $\overrightarrow{b}$ is the random direction, x_r and x_l denote the position in the right-hand and left-hand sides, respectively, d^t is the step size at time t, and δ is a reduction factor. Steps for the PSO and BAS to optimise the ANFIS parameters can be described as follows:

(1) Initialisation: Initialise the ANFIS parameters. Note that numMF and numEp refer to the number of MFs and epochs. Confirm the boundaries of the numbers of MFs and epochs. For PSO, define the number of particles and the maximum iteration. For BAS, define the maximum iteration, step size and reduction factor.
(2) Before-start preparation: Initialise the random position and velocity of PSO, and direction of BAS.
(3) Parameters search: Use PSO or BAS to generate a set of ANFIS parameters (numMF and numEp).
(4) Testing: Integrate the AFC-ANFIS model using the ANFIS parameters generated and compute the performance metric (ATE).
(5) Update: Update the best performance and best position of both PSO and BAS. Generate the position and velocity of PSO, and update the direction and step size of BAS.
(6) Check: Check if the stopping threshold is met. If yes, go to the next step. If no, go back to the parameters search (Step 3).
(7) Completion: Finalise and save the best performance measurement, and record its corresponding number of MFs and epochs.

4. Simulation environment

Our tests are carried out on an Apple M2 computer with a 10-core CPU and a 16-core GPU. We use Matlab R2023a (Update 4) and Simulink with the FL toolbox. The simulated model in Simulink is shown in Figure 6, which contains the proposed AFC-ANFIS controller and other essential elements ranging from a PD controller and an AFC loop, a disturbance planner, and finally, a robot dynamics model that executes the task.

4.1. Controller setting

In this simulation, three controllers are utilised: the proposed AFC-ANFIS controller, as well as AFC-IL controller for performance comparison. All controllers employ an identical PD controller with fixed gains, and are evaluated on the same five-link biped robot. The motor constant (K_{tn}) for the biped robot remains the same across all three controllers, as it is derived from the actual datasheet.

The following learning rule is used in AFC-IL:

$$IN_{k+1} = IN_k + \left(\phi + \Gamma \frac{d}{dt}\right) TE_k, \tag{35}$$

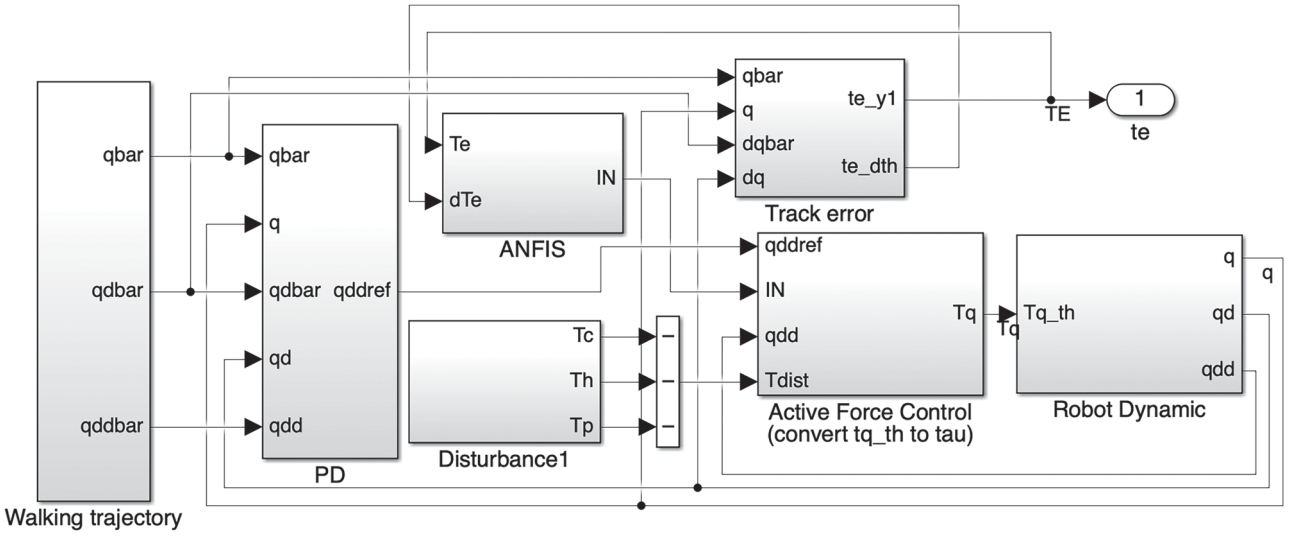


Figure 6. The proposed AFC-ANFIS method.

Table 4. Simulation settings.

Symbols	Values
Sampling time T_s	0.005 s
Finish time T_f	10 s
Solver	ODE45 Dormand–Prince
Step size	Variable
Number of sample data sets	2001
Clustering method	Grid partition
Output MF type	Linear

where ϕ and Γ are the learning parameters. The initial IN values required in the AFC-IL controller are $IN_1 = IN_2 = IN_3 = IN_4 = IN_5 = 1$.

4.2. Simulation setting

To enhance the performance in estimating IN values, ANFIS is trained based on 4 matrics data with the size of 2001 × 3. Because there are 4 links on completing tasks, and each input set has two inputs, including track error in angular positions (TE) and velocities ($\dot{TE}$), and one output IN. The training data sets have the form of $[(TE, \dot{TE}), IN]$. The number of rules in FIS (FR) can be determined by Equation (36).

$$FR = MF^n, \tag{36}$$

where FR is the total number of fuzzy rules, MF is the membership functions in the first layer of ANFIS, and n counts the number of inputs. In this case, as the n is given as 2, the total number of fuzzy rules is MF^2.

The simulation configuration is presented in Table 4.

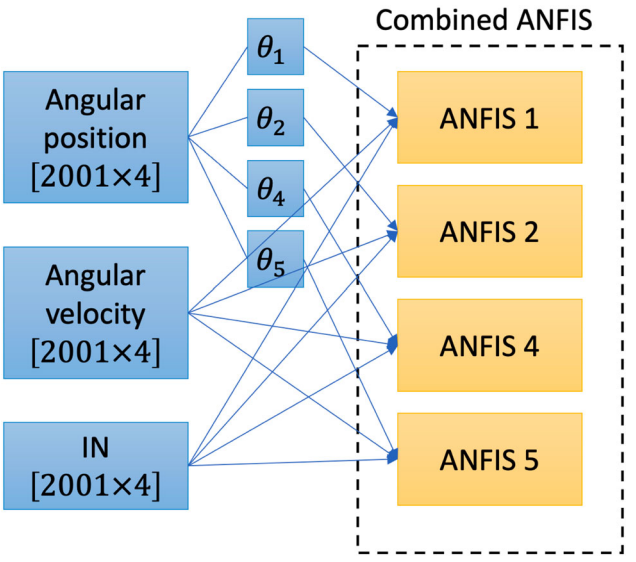


Figure 7. The training strategy for ANFIS.

4.3. ANFIS training

4.3.1. Iterative learning

As mentioned in Section 4.2, the fuzzy rules of the trained ANFIS model depend on the number of MFs and input number, with the rules expanding exponentially as the number of inputs increases. While adding more inputs can enhance performance, it also complicates the ANFIS model. For designing the controller for a five-link biped robot, the number of inputs is as many as 12. Therefore, as shown in Figure 7, we choose to separate the inputs and assign 4 ANFIS models to manage each link, instead of utilising a single, combined ANFIS model to process the complex

Table 5. Comparison in various MF type (walking and no disturbances).

MF type	MF number	Epochs	ATE (rad)	Duration (seconds)
Generalised Bell-shaped	5	100	0.0385	24.1982
Gaussian	5	100	0.0388	20.2139
Triangular	5	100	0.0394	25.3819
Trapezoidal	5	100	0.0394	21.3295

Table 6. Comparison in various MF number (no disturbances).

MF type	MF number	Epochs	ATE (rad)
	Walking		
Generalised bell-shaped	5	100	0.0385
Generalised bell-shaped	10	100	0.0399
Generalised bell-shaped	15	100	0.0388
Generalised bell-shaped	20	100	0.0387
Generalised bell-shaped	25	100	0.0402
	Climbing stairs		
Generalised bell-shaped	5	100	0.0823
Generalised bell-shaped	10	100	0.0828
Generalised bell-shaped	15	100	0.0820
Generalised bell-shaped	20	100	0.0814
Generalised bell-shaped	25	100	0.0820

inputs. All 4 ANFIS models adopt identical configurations.

In ANFIS training, it is important to observe the impact of changing MFs, epochs, and iterations. The choice and number of MFs are vital to the model performance and accuracy. In order to obtain satisfactory results, we explore a range of MFs, in terms of their quantity and types. The results are shown in Tables 5 and 6.

It can be obtained from Table 5, the generalised Bell function is the best option for this simulation. Although the Gaussian and trapezoidal functions have faster training time, the generalised Bell function yields the best ATE results.

Following Table 6, it is clear that the increment of MFs cannot guarantee an enhanced ANFIS performance. For stair-climbing tasks, identical results are observed when the number of MFs is set to 15 and 25, under fixed configurations of other parameters. In some cases, increasing the number of MFs can lead to a better approximation of complex relations based on the input data, enhancing the model prediction capability. More MFs allow for an intensive interpretation of input data sets and capture richer features. However, a significant risk associated with this strategy is overfitting, where the ANFIS model closely fits the given data and performs poorly on unseen data, and noise in data is over-interpreted.

This issue can arise not just from an excessive number of MFs but also from using too many training epochs. Hence, we adjust the epochs between 50 and 200 to evaluate the changes, as shown in Table 7.

Table 7. Comparison in various epoch numbers (no disturbances).

MF type	MF number	Epochs	ATE (rad)
	Walking		
Generalised bell-shaped	6	50	0.0389
Generalised bell-shaped	6	100	0.0387
	Climbing stairs		
Generalised bell-shaped	6	50	0.0822
Generalised bell-shaped	6	100	0.0826
Generalised bell-shaped	6	200	0.0829

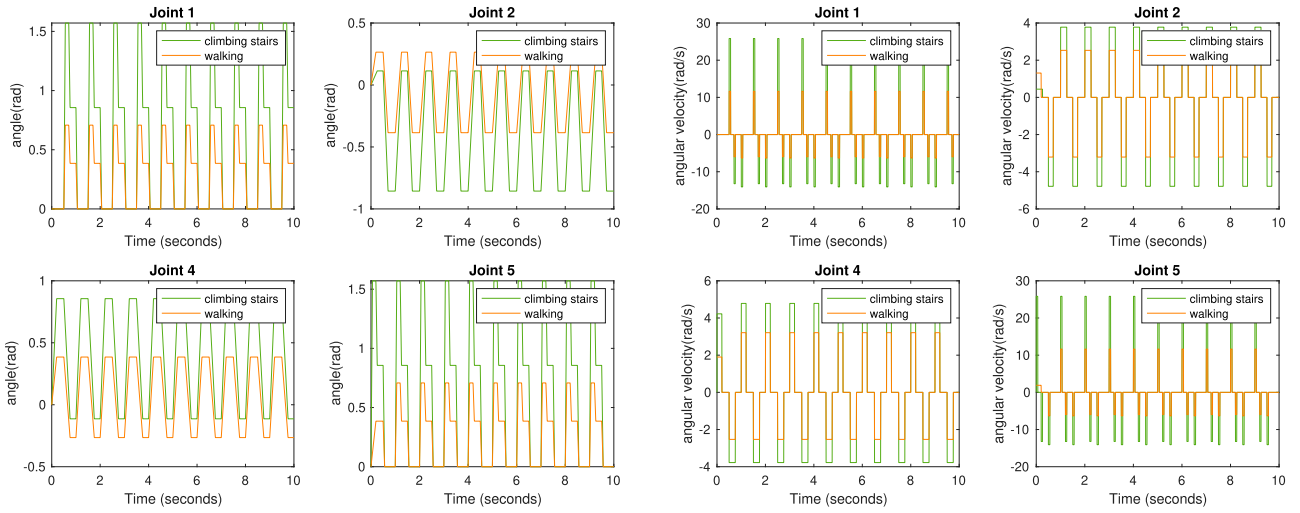
Table 8. Comparison in iterations (no disturbances).

ATE (rad)					
		Walking			
MF number	Epochs	IL1	IL2	IL3	IL4
5	100	0.0385	0.0386	0.0383	0.0383
6	100	0.0387	0.0389	0.0383	0.0382
IL5	IL6	IL7	IL8	IL9	IL10
0.0380	0.0386	0.0392	0.0391	0.0389	0.0391
0.0388	0.0385	0.0384	0.0388	0.0387	0.0385
		Climbing stairs			
MF number	Epochs	IL1	IL2	IL3	IL4
5	100	0.0823	0.0816	0.0816	0.0817
6	100	0.0826	0.0821	0.0823	0.0798
IL5	IL6	IL7	IL8	IL9	IL10
0.0816	0.0815	0.0810	0.0807	0.0827	0.0823
0.0815	0.0812	0.0819	0.0815	0.0818	0.081

From Table 7, when the biped robot is assigned to walk and the number of MFs is set to 6, increasing the number of epochs from 50 to 100 improves the performance. But when the biped robot is tasked with stair climbing with the same number of MFs, increasing the number of epochs from 50 to 100 and later to 200 only degrades the performance. The decline in performance could be related to overfitting.

Additionally, we set the number of iterations to 10 to observe changes in performance throughout the IL process of the generalised Bell MFs.

Table 8 indicates that IL enhances the ANFIS efficiency and outcomes. Employing a small number of MFs across multiple iterations refines model approximation without leading to overfitting. For both tasks, there is a substantial enhancement in effectiveness after several iterations as compared with that of the initial one. Similar to MFs and epochs, increasing the number of iterations does not necessarily enhance the performance.



(a) Angular position (b) Angular velocity

Figure 8. Reference trajectory. (a) Angular position (b) Angular velocity.

4.3.2. Parameters search

Compared with the IL method for tuning ANFIS whereby the numbers of MFs and epochs need to be set manually, the use of PSO and BAS for parameter search is more convenient. In the case of controlling the five-link biped robot, we set the upper and lower boundaries of MFs to 12 and 5, respectively, while the epoch runs from 90 to 110 with a step size of 5.

Using both PSO and BAS we can successfully obtain the results, whereby within the set ranges, the best performance for walking tasks is 0.03819 rad when the number of MFs is 5 and the number of epochs is 110. In the case of climbing stairs, the best performance is 0.08100 rad, with the number of MFs equals to 9 and the number of epochs equals to 105.

While with the results from IL-based ANFIS parameters are better than those from PSO and BAS, both parameters search methods offer a less time-consuming option, with slightly inferior performances.

5. Simulation results

5.1. Case 1: A five-link biped robot

5.1.1. Path setting

For a five-link biped robot to perform diverse tasks effectively, distinct settings of angular positions and velocities are given as shown in Figure 8. The desired link trajectory is derived from Lum et al. (1999).

Compared with walking, both changes in angular positions and accelerations are more significant in climbing stairs. Hence, it is more difficult for biped robots to climb stairs than walk, the tracking error in climbing stairs is expected to be greater than that from just walking.

Table 9. Controllers settings.

Symbols	Values
Motor constant K_{tn}	0.263
Proportional gain K_p	2500
Derivative gain K_d	100
Desired bandwidth λ_i, $i = 1, 2, 3, 4, 5$	50
AFC-IL	
Learning parameter ϕ	0.1
Learning parameter Γ	0.01
Initial IN IN_i, $i = 1, 2, 3, 4, 5$	1
AFC-ANFIS	
Input MF type	Generalised Bell
Number of input MFs	5-6
Epoch number	100

5.1.2. Numerical results

After the training of the ANFIS model using iterative learning, the best ANFIS models for both tasks are identified. The fine-tuned ANFIS model for the walking task requires 5 iterations with 5 generalised Bell-shaped MFs and 100 epochs. Its ATE result with no disturbances is 0.0380 rad. Conversely, the ANFIS model for climbing stairs delivers its optimal performance with ATE of 0.0798 rad (no disturbances), with 4 iterations, 6 generalised Bell-shaped MFs, and 100 epochs. The pre-set controller parameters and types are shown in Table 9. The AFC-IL parameter settings are derived from Kwek et al. (2003).

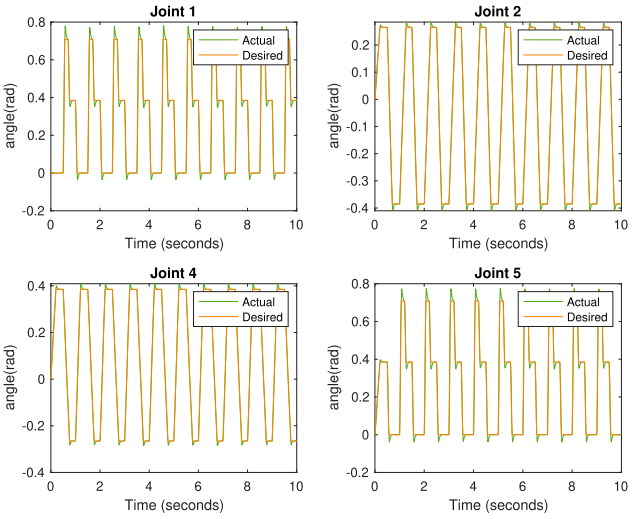


Figure 9. Angles of biped robot link in walking (no disturbances).

In order to evaluate the proposed AFC-ANFIS controller ability pertaining to trajectory tracking of a 5-link biped robot, we conduct several tests with different types of disturbances. These include an ideal condition where the disturbance is zero (T_0), disturbances are applied to links with a continuous torque T_c, a varying torque T_h, and a pulse torque T_p.

$$T_c = 200\,\text{N}, \tag{37}$$

$$T_h = 200\sin(10t)\,\text{N}, \tag{38}$$

$$T_p = 200\,\text{N},\ 10\%\ \text{of period of 0.5s}. \tag{39}$$

Figure 9 presents the angles of biped robot link in walking between actual and desired conditions with no disturbances. Figure 10 shows the comparison of tracking error in walking of both angular position and angular velocity between AFC-IL and AFC-ANFIS. Table 10 presents a performance comparison with the AFC-IL controller. From Table 10, AFC-ANFIS outperforms AFC-IL in all disturbance conditions in both walking and climbing tasks.

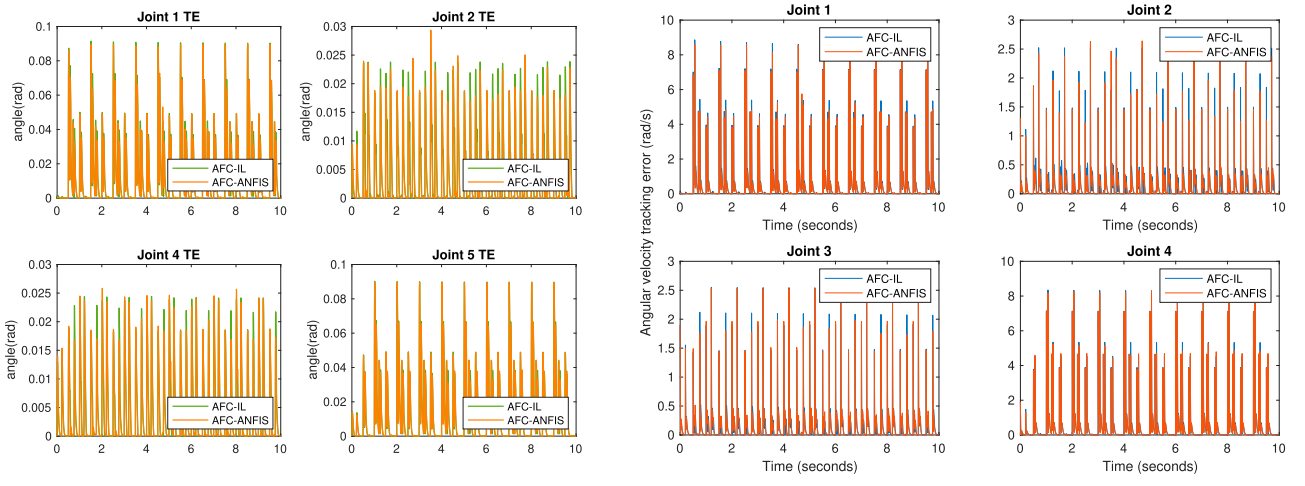
Table 10. Summary of results of AFC-ANFIS and AFC-IL.

Disturbance types	AFC-ANFIS ATE (rad)	AFC-IL ATE (rad)	Improvement (%)
	Walking		
T_0	0.0380	0.0396	4.1039
T_c	0.0380	0.0396	4.1234
T_h	0.0396	0.0407	2.6904
T_p	0.0382	0.0396	3.6591
	Climbing stairs		
T_0	0.0798	0.0832	4.0668
T_c	0.0808	0.0832	2.9016
T_h	0.0823	0.0839	1.9004
T_p	0.0809	0.0832	2.7885

5.2. Case 2: A PUMA 560 robot arm

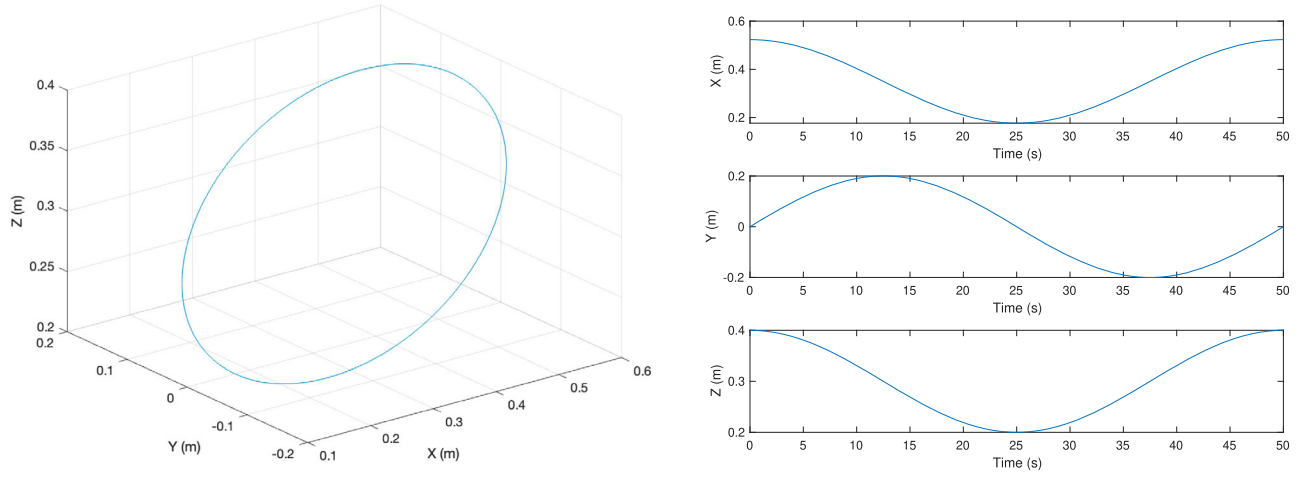
5.2.1. Path setting

In Li et al. (2019), model-free dual neural network (DNN) was proposed to utilise a PUMA 560 robot arm for tackling a trajectory tracking task. The predetermined trajectory is shown in Figure 11. It is a circle with a radius of 0.2 m and is inclined toward the x-axis at 30 degrees. The desired trajectory of the joint angles is shown in Figure 12.



(a) Angular position (b) Angular velocity

Figure 10. Comparison of tracking error in walking (no disturbances). (a) Angular position (b) Angular velocity.



(a) Three-dimension view (b) Trajectory in x,y and z axis

Figure 11. Reference trajectory. (a) Three-dimension view (b) Trajectory in x,y and z axis.

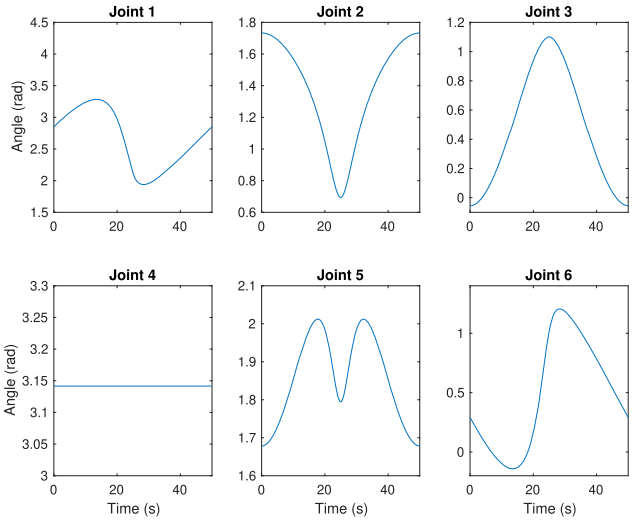


Figure 12. Desired trajectory of joints angles of PUMA 560 arm.

5.2.2. Numerical results

In this case study, both AFC-based controllers, namely AFC-ANFIS and AFC-IL, and DNN controllers are evaluated and compared. The controller settings are presented in Table 11.

According to Li et al. (2019), utilising the DNN controller on the PUMA 560 robot, the position errors for the x, y and z axes fluctuate within the range of 0.02 m. With the use of AFC-based controllers for the PUMA 560 robot to follow the same trajectory, the maximum position error is 0.0087 m from AFC-IL, and 0.0023 m from AFC-ANFIS as depicted in Figure 13(a,b). The tracking error of AFC-ANFIS is much smaller than AFC-IL because the AFC-IL controller does not have oscillations at the beginning. Hence, we can conclude

Table 11. Controller settings.

Symbols	Values
Proportional gain K_p	2025
Derivative gain K_d	90
Desired bandwidth λ_i, $i = 1, 2, 3, 4, 5$	45
AFC-IL	
Learning parameter ϕ	0.1
Learning parameter Γ	0.01
Initial IN IN_i, $i = 1, 2, 3, 4, 5$	1
AFC-ANFIS	
Input MF type	Generalised Bell
Number of input MFs	8
Epoch number	95

that, in this case study, all AFC-based controllers, including AFC-ANFIS, outperform the DNN controller in terms of trajectory tracking accuracy.

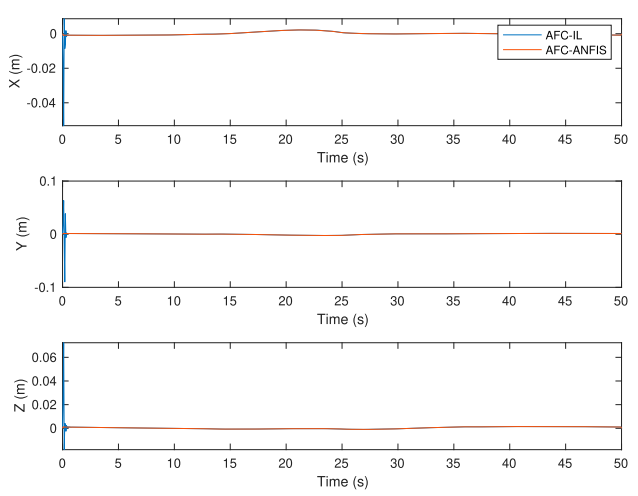
Figure 14 shows the joint angles of the PUMA 560 arm controlled by AFC-based controllers, depicting a good performance in following the predetermined trajectory. The trajectories are smooth. Figure 15 shows tracking trajectories in the x, y, and z axes. The actual trajectory is very close to the desired one, and their difference is shown in Figure 13.

In addition, we adjust the disturbances (continuous torque T_c, a varying torque T_h, and a pulse torque T_p.) applied to the joint of the PUMA 560 robot arm, as follows,

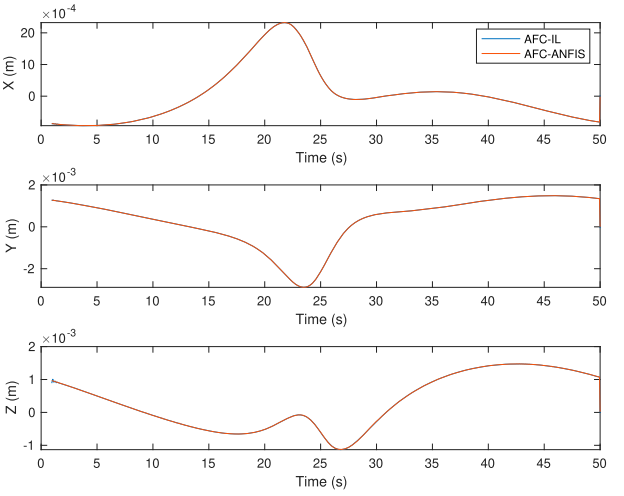
$$T_c = 100\,\text{N}, \tag{40}$$

$$T_h = 200\sin(10t)\,\text{N}, \tag{41}$$

$$T_p = 100\,\text{N},\ 10\%\ \text{of period of 0.5s}. \tag{42}$$



(a) Time history of position error



(b) Steady state history of position error

Figure 13. Comparison of track error of PUMA 560 arm. (a) Time history of position error (b) Steady state history of position error.

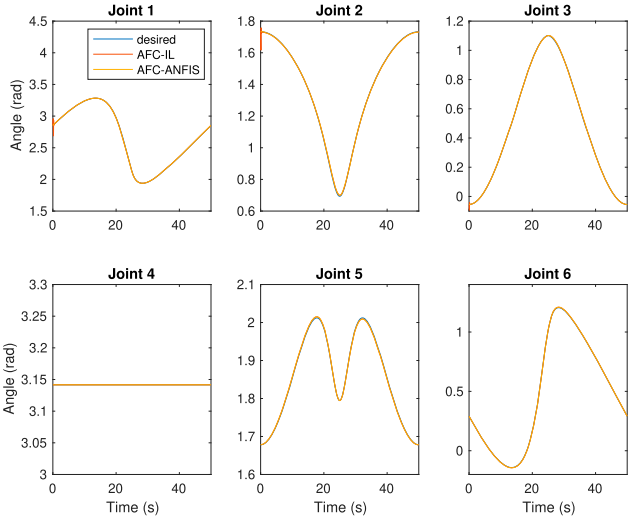


Figure 14. Angles of PUMA 560 arm (no disturbances).

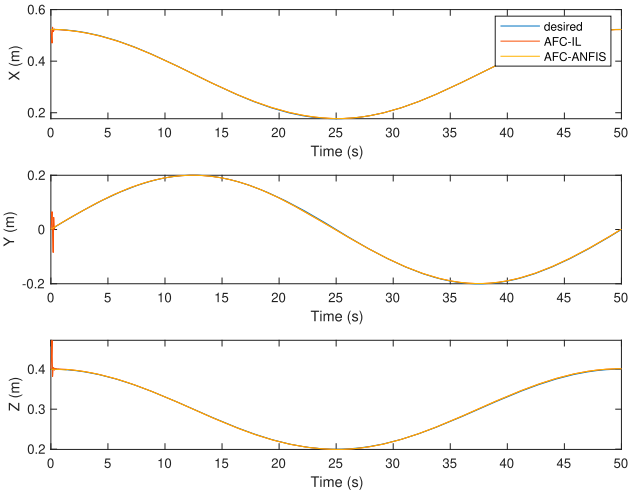


Figure 15. Comparison of translation position of PUMA 560 arm.

Table 12. Summary of results of AFC-ANFIS and AFC-IL.

Disturbance types	AFC-ANFIS ATE (rad)	AFC-IL ATE (rad)	Improvement (%)
T_0	0.0096	0.0103	7.0571
T_c	0.0096	0.0103	6.6023
T_h	0.0097	0.0114	14.6590
T_p	0.0096	0.0103	6.3346

Table 12 presents a performance comparison between AFC-ANFIS and AFC-IL controllers.

From the Table 12, the AFC-ANFIS outperforms the AFC-IL by 7.06%.

6. Conclusion

In this paper, we have designed an AFC-ANFIS controller to address the trajectory tracking problem. The developed controller has been evaluated using a five-link biped robot to perform both walking and climbing stairs and a PUMA 560 to follow a predetermined trajectory. Different scenarios with disturbances have been investigated, including ideal situations, disturbances with constant torques, varying torques, and pulse torques applied to each link. To tune the AFC-ANFIS controller, different ANFIS configurations have been tested, including the MF numbers and types as well as the epoch numbers. We have applied both PSO and BAS algorithms for AFC in ANFIS parameter tuning. The simulation on a biped robot has demonstrated that the proposed AFC-ANFIS controller outperforms AFC-IL in all scenarios. In the case study of

the PUMA 560 robot arm, the AFC-ANFIS controller outperforms the DNN controller.

The simulation results indicate that AFC-ANFIS performs well in disturbance rejection and trajectory tracking on two different types of robots. For further work, we aim to improve AFC-ANFIS from several perspectives. These include overfitting issues and a possible large number of inputs. The AFC-ANFIS performance can be improved to address other disturbances such as those caused by different terrains. In addition, type-2 MFs can be employed to enhance the adaptability and performance of the proposed AFC-ANFIS controller.

Author contribution

All authors have contributed to the following tasks: concept and design, data collection, analysis and interpretation of results, and manuscript preparation.

Disclosure statement

No potential conflict of interest was reported by the author(s).

Data availability statement

The datasets generated during the current study are not publicly available, but are available from the corresponding author at reasonable request.

References

Abdelmaksoud, S. I., Mailah, M., & Abdallah, A. M. (2020). Robust intelligent self-tuning active force control of a quadrotor with improved body jerk performance. *IEEE Access*, *8*, 150037–150050. https://doi.org/10.1109/Access.6287639

Abdelmaksoud, S. I., Mailah, M., & Abdallah, A. M. (2021). Practical real-time implementation of a disturbance rejection control scheme for a twin-rotor helicopter system using intelligent active force control. *IEEE Access*, *9*, 4886–4901. https://doi.org/10.1109/Access.6287639

Al-Hmouz, A., Shen, J., Al-Hmouz, R., & Yan, J. (2012, July). Modeling and simulation of an adaptive neuro-fuzzy inference system (ANFIS) for Mobile learning. *IEEE Transactions on Learning Technologies*, *5*(3), 226–237. https://doi.org/10.1109/TLT.2011.36

Ali, M. A., Radzak, M. S., Mailah, M., Yusoff, N., Razak, B. A., Karim, M. S. A., & Abdulghafor, R. (2021, November). A novel inertia moment estimation algorithm collaborated with active force control scheme for wheeled mobile robot control in constrained environments. *Expert Systems with Applications*, *183*, 115454. https://doi.org/10.1016/j.eswa.2021.115454

Armstrong, B., Khatib, O., & Burdick, J. (1986). *The explicit dynamic model and inertial parameters of the puma 560 arm* (Tech. Rep.). Stanford University: Stanford Artificial Intelligence Laboratory.

Baca, T., Hert, D., Loianno, G., Saska, M., & Kumar, V. (2018, December). Model predictive trajectory tracking and collision avoidance for reliable outdoor deployment of unmanned aerial vehicles. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE.

Bansal, J. C. (2019). *Evolutionary and swarm intelligence algorithms*. Springer International Publishing.

Chen, W. H., Yang, J., Guo, L., & Li, S. (2016, February). Disturbance-observer-based control and related methods–an overview. *IEEE Transactions on Industrial Electronics*, *63*(2), 1083–1095. https://doi.org/10.1109/TIE.2015.2478397

Craig, J. (2018). *Introduction to robotics: Mechanics and control*. Pearson.

Geng, T. (2014, July). Torso inclination enables faster walking in a planar biped robot with passive ankles. *IEEE Transactions on Robotics*, *30*(3), 753–758. https://doi.org/10.1109/TRO.2014.2298058

Ghose, D. K., Tanaya, K., Sahoo, A., & Kumar, U. (2022, March). Performance evaluation of hybrid ANFIS model for flood prediction. In *2022 8th International Conference on Advanced Computing and Communication Systems (ICACCS)*. IEEE.

Hassan, M. F., Mailah, M., Junid, R., & Alang, N. (2010, June). Vibration suppression of a handheld tool using intelligent active force control (AFC). In *Proceedings of the World Congress on Engineering*, Lecture Notes in Engineering and Computer Science (Vol. 2, pp. 1636–1641). IAENG.

Hewit, J., & Burdess, J. (1981, January). Fast dynamic decoupled control for robotics, using active force control. *Mechanism and Machine Theory*, *16*(5), 535–542. https://doi.org/10.1016/0094-114X(81)90025-2

Huang, H., Arogbonlo, A., Yu, S., & Kwek, L. C. (2024, April). Reinforcement learning integrated active force control for five-link biped robots. In *2024 IEEE International Systems Conference (SYSCON)*. IEEE.

Huang, Q., Dong, C., Yu, Z., Chen, X., Li, Q., Chen, H., & Liu, H. (2022, December). Resistant compliance control for biped robot inspired by humanlike behavior. *IEEE/ASME Transactions on Mechatronics*, *27*(5), 3463–3473. https://doi.org/10.1109/TMECH.2021.3139332

Huang, Y., Wang, L., Guo, W., Kang, Q., & Wu, Q. (2018, January). Chance constrained optimization in a home energy management system. *IEEE Transactions on Smart Grid*, *9*(1), 252–260. https://doi.org/10.1109/TSG.2016.2550031

Ider, S. (2000, March). Force and motion trajectory tracking control of flexible-joint robots. *Mechanism and Machine Theory*, *35*(3), 363–378. https://doi.org/10.1016/S0094-114X(99)00022-1

Jang, J. S. (1993). ANFIS: Adaptive-network-based fuzzy inference system. *IEEE Transactions on Systems, Man, and Cybernetics*, *23*(3), 665–685. https://doi.org/10.1109/21.256541

Johansson, R., Nilsson, K., & Robertsson, A. (2014). Force control. In *Handbook of manufacturing engineering and technology* (pp. 1–29). Springer London.

Kennedy, J., & Eberhart, R. (1995). Particle swarm optimization. In *Proceedings of ICNN'95 – International Conference on Neural Networks* (Vol. 4, pp. 1942–1948). IEEE.

Khan, A. T., Cao, X., & Li, S. (2022, April). Dual beetle antennae search system for optimal planning and robust control of 5-link biped robots. *Journal of Computational Science*, *60*, 101556. https://doi.org/10.1016/j.jocs.2022.101556

Khan, S. A., Equbal, M. D., & Islam, T. (2015, February). A comprehensive comparative study of DGA based transformer fault diagnosis using fuzzy logic and ANFIS models. *IEEE Transactions on Dielectrics and Electrical Insulation*, *22*(1), 590–596. https://doi.org/10.1109/TDEI.2014.004478

Khan, A. T., Li, S., & Cao, X. (2022, January). Human guided cooperative robotic agents in smart home using beetle antennae search. *Science China Information Sciences*, *65*(2), 122204. https://doi.org/10.1007/s11432-020-3073-5

Khan, A. T., Li, S., & Zhou, X. (2021, December). Trajectory optimization of 5-link biped robot using beetle antennae search. *IEEE Transactions on Circuits and Systems II: Express Briefs*, *68*(10), 3276–3280.

Kwek, L. (2003). *Intelligent active force control methods of a biped robot* [Unpublished doctoral dissertation]. University Science of Malaysia.

Kwek, L. C., Wong, E. K., Loo, C. K., & Rao, M. V. C. (2003). Application of active force control and iterative learning in a 5-link biped robot. *Journal of Intelligent and Robotic Systems*, *37*(2), 143–162. https://doi.org/10.1023/A:1024187206507

Lee, C., & Ziegler, M. (1983). *A geometric approach in solving the inverse kinematics of puma robots* (Tech. Rep.). The University of Michigan: Department of Electrical and Computer Engineering.

Li, Z., Feiling, J., Ren, H., & Yu, H. (2015, July). A novel tele-operated flexible surgical arm with optimal trajectory tracking aiming for minimally invasive neurosurgery. In *2015 IEEE 7th International Conference on Cybernetics and Intelligent Systems (CIS) and IEEE Conference on Robotics, Automation and Mechatronics (RAM)*. IEEE.

Li, Z., Li, S., Bamasag, O. O., Alhothali, A., & Luo, X. (2023, November). Diversified regularization enhanced training for effective manipulator calibration. *IEEE Transactions on Neural Networks and Learning Systems*, *34*(11), 8778–8790. https://doi.org/10.1109/TNNLS.2022.3153039

Li, Z., Li, S., & Luo, X. (2021, January). An overview of calibration technology of industrial robots. *IEEE/CAA Journal of Automatica Sinica*, *8*(1), 23–36. https://doi.org/10.1109/JAS.2020.1003381

Li, S., Shao, Z., & Guan, Y. (2019, May). A dynamic neural network approach for efficient control of manipulators. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, *49*(5), 932–941. https://doi.org/10.1109/TSMC.6221021

Li, H., Yang, D., Su, W., Lu, J., & Yu, X. (2019, January). An overall distribution particle swarm optimization MPPT algorithm for photovoltaic system under partial shading. *IEEE Transactions on Industrial Electronics*, *66*(1), 265–275. https://doi.org/10.1109/TIE.2018.2829668

Liu, Y., & Zhang, Y. (2015, June). Iterative local ANFIS-based human welder intelligence modeling and control in pipe GTAW process: A data-driven approach. *IEEE/ASME Transactions on Mechatronics*, *20*(3), 1079–1088. https://doi.org/10.1109/TMECH.2014.2363050

Liu, X., Zhang, M., & Rogers, E. (2019, December). Trajectory tracking control for autonomous underwater vehicles based on fuzzy re-planning of a local desired trajectory. *IEEE Transactions on Vehicular Technology*, *68*(12), 11657–11667. https://doi.org/10.1109/TVT.25

Lu, H., Jin, L., Luo, X., Liao, B., Guo, D., & Xiao, L. (2019, November). RNN for solving perturbed time-varying underdetermined linear system with double bound limits on residual errors and state variables. *IEEE Transactions on Industrial Informatics*, *15*(11), 5931–5942. https://doi.org/10.1109/TII.9424

Lum, H., Zribi, M., & Soh, Y. (1999, August). Planning and control of a biped robot. *International Journal of Engineering Science*, *37*(10), 1319–1349. https://doi.org/10.1016/S0020-7225(98)00118-9

Noshadi, A., Mailah, M., & Zolfagharian, A. (2012, June). Intelligent active force control of a 3-RRR parallel manipulator incorporating fuzzy resolved acceleration control. *Applied Mathematical Modelling*, *36*(6), 2370–2383. https://doi.org/10.1016/j.apm.2011.08.033

Priyandoko, G., Mailah, M., & Jamaluddin, H. (2009, April). Vehicle active suspension system using skyhook adaptive neuro active force control. *Mechanical Systems and Signal Processing*, *23*(3), 855–868. https://doi.org/10.1016/j.ymssp.2008.07.014

Razmi, D., Babayomi, O., Davari, A., Rahimi, T., Miao, Y., & Zhang, Z. (2022, August). Review of model predictive control of distributed energy resources in microgrids. *Symmetry*, *14*(8), 1735. https://doi.org/10.3390/sym14081735

Sini, S., & Ananthan, T. (2022, March). A disturbance observer based control for quadrotor aircraft subject to wind gusts. In *2022 IEEE International Conference on Signal Processing, Informatics, Communication and Energy Systems (SPICES)*. IEEE.

Spong, M., & Vidyasagar, M. (2008). *Robot dynamics and control*. Wiley India Pvt. Limited.

Tzafestas, S., Raibert, M., & Tzafestas, C. (1996, January). Robust sliding-mode control applied to a 5-link biped robot. *Journal of Intelligent and Robotic Systems*, *15*(1), 67–133. https://doi.org/10.1007/BF00435728

Yang, Y., Shi, J., Huang, S., Ge, Y., Cai, W., Li, Q., & Zhao, M. (2022, December). Balanced standing on one foot of biped robot based on three-particle model predictive control. *Biomimetics*, *7*(4), 244. https://doi.org/10.3390/biomimetics7040244

Yu, Z., Chen, X., Huang, Q., Zhang, W., Meng, L., Zhang, W., & Gao, J. (2016, July). Gait planning of omnidirectional walk on inclined ground for biped robots. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, *46*(7), 888–897. https://doi.org/10.1109/TSMC.2015.2487240

Yuan, S. Z., Mailah, M., & Howe Hing, T. (2020, December). Intelligent active force control of an underwater remotely operated vehicle using evolutionary computation technique. *Jurnal Mekanikal*, *43*(2), 11–25.

Zhai, J., & Li, Z. (2022, February). Fast-exponential sliding mode control of robotic manipulator with super-twisting method. *IEEE Transactions on Circuits and Systems II: Express Briefs*, *69*(2), 489–493.

Zhang, B., Li, S., Chen, X., & Mao, Y. (2024). A novel zeroing neural model for solving dynamic matrix moore-penrose inverse and its application to visual servoing control of manipulator. *IEEE Transactions on Instrumentation and Measurement*, *73*, 1–13.

Zhang, B., Nie, K., Chen, X., & Mao, Y. (2022, January). Development of sliding mode controller based on internal model controller for higher precision electro-optical tracking system. *Actuators*, *11*(1), 16. https://doi.org/10.3390/act11010016

Zhang, Y., Zhang, P., Ran, C., & Zhang, X. (2021, April). An active disturbance rejection control design method based on disturbance separation. *Journal of Physics: Conference Series*, *1848*(1), 012127.

Zhao, B., Wang, H., Li, Q., Li, J., & Zhao, Y. (2019, June). PID trajectory tracking control of autonomous ground vehicle based on genetic algorithm. In *2019 Chinese Control and Decision Conference (CCDC)*. IEEE.