Robust PID Control for Upper Limb Exoskeleton Robot based on the Beetle Antennae Search Algorithm

Fenggang Liu, Lang Rao, Zhaoyun He, Lie Yu and Lei Ding

Abstract—**This paper presents a computational model for one kind of two-degree-of-freedom (2-DOF) upper limb exoskeleton robot. This robot consists of two connected body segments to imitate the upper shoulder and elbow joints of human hand. Two sets of direct current (DC) motors are severally used to drive the robotic shoulder and elbow joints. The dynamic of the DC motor is accurately developed to establish the relationship between the input current and the actuated force/torque. The PID controller is considered as a standard in technical and industrial applications, and selected to control the whole system in this paper. Therefore, the tuning of PID gains is extremely important, while utilizing optimization algorithms to reduce the tracking error is one approach of tuning PID gains. This paper presented the improved beetle antennae search (IBAS) algorithm to optimize the PID gains. The original BAS algorithm can achieve a wide search range, cost low time complexity and gain high search accuracy. Moreover, the proposed IBAS algorithm can realize more extensive search scope and more precise search compared with the original BAS algorithm. Finally, the IBAS algorithm is compared with the original BAS, PSO and GA algorithms on the basis of the optimization results. The comparison results demonstrated that the IBAS algorithm gains superior performance in addressing PID tuning problems.**

Index Terms—**Upper limb exoskeleton robot, Direct current motor, PID controller, Beetle antennae search.**

I. Introduction

The upper limb exoskeleton robot has recently attracted considerable interests due to its potential applications in the fields of medicine, industry and military [1-2]. Such devices can provide additional assistance to military soldiers engaged in heavy lifting tasks. The purpose of such robots is usually to enable the robotic limb to move in unison with the human limb. Moreover, the human limb exerts the amount of force as less as possible when lifting a load, while the robotic limb provides the amount as bigger as possible. As this kind of robot is a wearable device attached closely with the human limbs, the controller design must consider the cooperation between the human and robot movements.

Many studies have been made to establish the dynamic equation between the wearer and the robot in order to figure out the inherent difficulties associated with mathematical modelling. The most prevalent methods for developing the dynamic relationship between force/torque and position are the Newton-Euler and Lagrange equations [6-7]. Moreover, as the number of degrees of freedom (DOF) increases, the computational cost of creating the dynamic model also dramatically increases.

The PID controller is widely used due to its simple design and ease of implementation [8]. Its performance is mainly determined by three parameters, such as the proportional gain, the integral gain, and the differential gain [9-10]. It is important to select appropriate gains for the PID controller. In the literature, numerous methods for determining the PID gains have been established. The most conventional method is the Ziegler-Nichols (Z-N) method [11]. However, this Z-N method would cause large overshoot and control oscillation. Therefore, the artificial intelligence methods, such as the genetic algorithm (GA) [12-13], particle swarm optimization (PSO) [14-15], whale optimization algorithm (WOA) [16-17], and others, are increasingly commonly utilized to realize the PID gains adjustment. This study employs a similar approach to the Reference [18] using the beetle antennae search (BAS) method to tune the PID gains. The BAS algorithm can achieve effective global optimization, and has been applied in several scientific domains, including machine learning [19-20], robotics [21], engineering [22], and finance [23]. Additionally, the BAS algorithm has undergone several modifications, such as binary [24] and semi-integer [25] versions, to more effectively address different problems.

This paper presents a dynamic model of two degrees of freedom (2-DOF) upper limb exoskeleton robot driven by direct current (DC) motors. The dynamic model is established and explained how the torque produced by the robot and human to control the robot precisely. The PID controller is used to control the whole system, and an improved BAS algorithm (IBAS) is presented to adjust the PID gains. Specifically, the iterative chaotic map with infinite collapses (ICMIC) is to obtain a random value, which

Manuscript received January 9, 2024; revised October 17, 2024.

This work was supported by the National Natural Science Foundation of China "Research on motion pattern recognition of exoskeleton robot based on curve similarity model" (62106178).

Fenggang Liu is an Associate Professor at the School of Artificial Intelligence, Wuchang University of Technology, Wuhan, 430221, China. (e-mail: 34834779@qq.com).

Lang Rao is an Associate Professor at the School of Artificial Intelligence, Wuchang University of Technology, Wuhan, 430221, China. (e-mail: 413564609@qq.com).

Zhaoyun He is a Lecturer at the School of Artificial Intelligence, Wuchang University of Technology, Wuhan, 430221, China. (e-mail: 1007256364@qq.com).

Lie Yu is an Associate Professor at the School of Electronic and Electrical Engineering, Wuhan Textile University, Wuhan, 430223, China. (corresponding author to provide phone: +86 18607155647; e-mail: lyu@wtu.edu.cn).

Lei Ding is an Associate Professor at the School of Computer Science and Artificial Intelligence, Wuhan Textile University, Wuhan, 430223, China. (e-mail: lding@wtu.edu.cn).

can realize more uniform traversal and accelerate the convergence. A novel method is proposed to compute the step size which can implement more extensive search scope in the early stages, and more precise search in the later stages. Finally, the control performance based on IBAS algorithm is compared with those based on the other optimization algorithms.

II. Dynamics of the Upper Limb Exoskeleton

The schematic of 2-DOF upper limb exoskeleton robot is depicted in Fig. 1. This robot is employed to provide supplementary physical assistance to the upper limb for the purpose of lifting heavy loads. In particular, the physical assistance is provided only in the sagittal plane, with two joints, such as the shoulder and elbow parts, actuated by direct current (DC) motors.

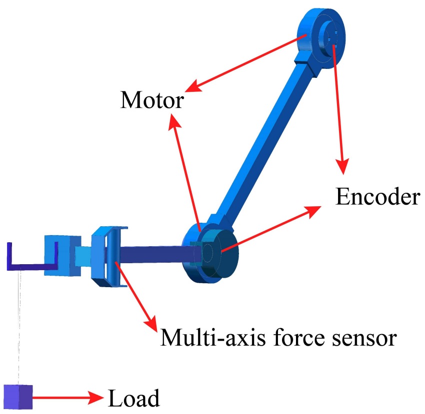

Fig. 1. Schematic of 2-DOF upper limb exoskeleton robot.

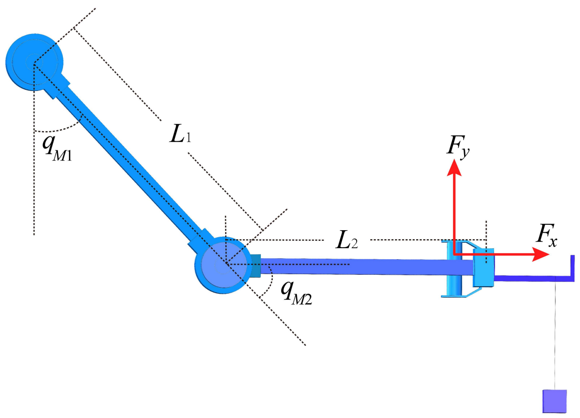

Fig. 2. Geometry sizes of upper limb exoskeleton robot.

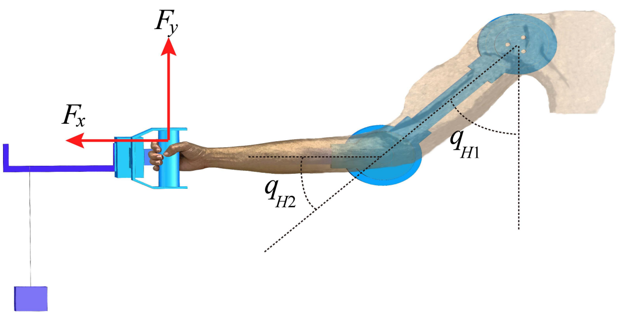

Fig. 3. Schematic of human rotary angles.

Two encoders are positioned inside the joints to collect the angles and velocities. The control strategy is to synchronize the robot with the human operator, while the robot provides more power and the human offers less. A multi-axis force sensor is mounted on the wrist as the end effector to detect the contact force between the human and robot. The force detected in operational space must be transformed into a torque in joint space, which can be described as follows.

$$T = J^T F \tag{1}$$

where T is the torque in the joint space, and J is the Jacob matrix. F is the force in the operational space, and monitored by the multi-axis force sensor. As this study referred to two joint controls, the Equation (1) could be specified below.

$$T = [T_s \quad T_e]^T \tag{2}$$

$$J = \begin{bmatrix} L_1\cos(q_{H1}) + L_2\cos(q_{H1}+q_{H2}) & L_2\cos(q_{H1}+q_{H2}) \\ L_1\sin(q_{H1}) + L_2\sin(q_{H1}+q_{H2}) & L_2\sin(q_{H1}+q_{H2}) \end{bmatrix} \tag{3}$$

$$F = [F_x \quad F_y]^T \tag{4}$$

where T_s is the torque of shoulder joint, and T_e is the torque of elbow joint. J is a 2×2 Jacob matrix. L_1 is the forearm length, and L_2 is the upper arm length. q_{H1} is the human shoulder angle, and q_{H2} is the human elbow angle. Actually, F_x and F_y are obtained through the multi-axis force sensor, as shown in Fig. 2 and Fig. 3.

For simulation, the multi-axis force sensor is considered as a spring model whose force is in direct proportion to its deformation. In this paper, the deformation can be regarded as the error between the human and robot positions. If this error is closed to zero, the robot synchronizes well with the human such that the values of F_x and F_y approach around zero. If this error is significantly large, the human drives the robot to move with the result that the F_x and F_y possess big values. Therefore, F can be described as follows.

$$\begin{cases} F = k_f L \\ L = [L_x \quad L_y]^T \end{cases} \tag{5}$$

where k_f is the elastic coefficient, and L is the error between the human and robot positions. L_x and L_y can be specified below.

$$\begin{cases} L_x = [L_1\cos(q_{H1}) + L_2\cos(q_{H1}+q_{H2})] \\ -[L_1\cos(q_{M1}) + L_2\cos(q_{M1}+q_{M2})] \\ L_y = [L_1\sin(q_{H1}) + L_2\sin(q_{H1}+q_{H2})] \\ -[L_1\sin(q_{M1}) + L_2\sin(q_{M1}+q_{M2})] \end{cases} \tag{6}$$

where q_{M1} is the robot shoulder angle, and q_{M2} is the robot elbow angle. The robot joint angles are measured by the encoders, while the joints are driven by the DC motor. The dynamic of DC motor actuating the joint can be expressed through the Lagrange equation in the following.

$$\begin{cases} T_L + T = M(q_M)\ddot{q}_M + C(q_M, \dot{q}_M) + G(q_M) \\ T_L = \begin{bmatrix} T_{L1} & T_{L2} \end{bmatrix}^T \\ q_M = \begin{bmatrix} q_{M1} & q_{M2} \end{bmatrix}^T \end{cases} \tag{7}$$

where T_{L1} is the driving torque generated by the DC motor to drive the forearm, and T_{L2} is the driving torque generated by the DC motor to drive the upper arm. M is a 2×2 matrix about inertia torque, C is a 2×2 matrix about the torque produced by the centripetal force, and G is a 2×1 matrix about the torque caused by gravity. Specifically, the elements of matrix of M, C and G can be written below.

$$M = \begin{bmatrix} M_{11} & M_{12} \\ M_{21} & M_{22} \end{bmatrix} \tag{8}$$

$$C = \begin{bmatrix} C_{11} & C_{12} \\ C_{21} & C_{22} \end{bmatrix} \begin{bmatrix} \dot{q}_{M1}^2 \\ \dot{q}_{M2}^2 \end{bmatrix} + \begin{bmatrix} C_{13} & C_{14} \\ C_{23} & C_{24} \end{bmatrix} \begin{bmatrix} \dot{q}_{M1}\dot{q}_{M2} \\ \dot{q}_{M2}\dot{q}_{M1} \end{bmatrix} \tag{9}$$

$$G = \begin{bmatrix} G_1 \\ G_2 \end{bmatrix} \tag{10}$$

$$M_{11} = m_1 L_{g1}^2 + m_2(L_1^2 + L_{g1}^2) + 2m_2 L_{g2} L_1 \cos(q_{M2}) + I_1 + I_2 \tag{11}$$

$$M_{12} = M_{21} = m_2 L_{g2}^2 + m_2 L_{g2} L_1 \cos(q_{M2}) + I_2 \tag{12}$$

$$M_{22} = m_2 L_{g2}^2 + I_2 \tag{13}$$

$$C_{11} = C_{22} = C_{23} = C_{24} = 0 \tag{14}$$

$$C_{12} = C_{13} = C_{14} = -m_2 L_{g2} L_1 \cos(q_{M2}) \tag{15}$$

$$C_{21} = m_2 L_{g2} L_1 \cos(q_{M2}) \tag{16}$$

$$G_1 = m_2 g L_{g2} \sin(q_{M1} + q_{M2}) + m_1 g L_{g1} \sin(q_{M1}) + m_2 g L_1 \sin(q_{M1}) \tag{17}$$

$$G_2 = m_2 g L_{g2} \sin(q_{M1} + q_{M2}) \tag{18}$$

where m_1 is the mass of forearm, and m_2 is the mass of upper arm. L_1 is the length of forearm, and L_2 is the length of upper arm. L_{g1} is the centre of gravity position of forearm, and L_{g2} is the centre of gravity position of upper arm. I_1 is the inertia of forearm, and I_2 is the inertia of upper arm. When the driving torques are imposed on the forearm and upper arm, the robot joints would produce the angles, velocities and accelerations. The Equation (7) can be rewritten in the following.

$$\ddot{q}_M = M^{-1}(T_L + T - C - G) \tag{19}$$

where M^{-1} is the inverse matrix of M, and the expression of M^{-1} can be written as:

$$M^{-1} = \begin{bmatrix} H_{R11} & H_{R12} \\ H_{R21} & H_{R22} \end{bmatrix} \tag{20}$$

where the elements of matrix M^{-1} can be described as follows.

$$H_{R11} = \frac{4(\mathrm{m_2 L_2^2} + 4\mathrm{I_2})}{H_{note}} \tag{21}$$

$$H_{R12} = \frac{-4(4\mathrm{I_2} + 2\mathrm{m_2 L_1 L_2}\cos(q_{\mathrm{M2}}) + \mathrm{m_2 L_2^2})}{H_{note}} \tag{22}$$

$$H_{R21} = \frac{-4(4\mathrm{I_2} + 2\mathrm{m_2 L_1 L_2}\cos(q_{\mathrm{M2}}) + \mathrm{m_2 L_2^2})}{H_{note}} \tag{23}$$

$$H_{R22} = \frac{4(4\mathrm{I_1} + 4\mathrm{m_2 L_1^2} + 4\mathrm{m_2 L_1 L_2}\cos(\mathrm{q_{M2}}))}{H_{note}} + \frac{4(\mathrm{m_2 L_2^2} + \mathrm{m_1 L_1^2} + 4\mathrm{I_2})}{H_{note}} \tag{24}$$

$$\begin{aligned} H_{note} &= 4\mathrm{I_1 m_2 L_2^2} + 16\mathrm{I_1 I_2} + 4\mathrm{m_2^2 L_1^2 L_2^2} + 16\mathrm{m_2 L_1^2 I_2} \\ &+ \mathrm{m_1 L_1^2 m_2 L_2^2} + 4\mathrm{m_1 L_1^2 I_2} - 4\mathrm{m_2^2 L_1^2 L_2^2}(\cos(q_{\mathrm{M2}}))^2 \end{aligned} \tag{25}$$

The DC motors selected in this system are permanent magnet synchronous motors, which are driven by DC voltage. The current commutation of DC motors is achieved by solid-state switches, while the commutation instant is determined by the rotor position, which is detected by an encoder mounted inside the robot joint. Consequently, the torque from the DC motor can be calculated as follows.

$$T_e = J_\Omega \ddot{q}_M + R_\Omega \dot{q}_M + T_L \tag{26}$$

where T_e is the electromagnetic torque, J_Ω is the moment of inertia of DC motor, q is the rotor position, R_Ω is the damping coefficient, and T_L is the load torque. In this paper, the electromagnetic torque is considered to be proportional to the input current, which can be described as:

$$T_e = K_t i \tag{27}$$

where K_t is the motor torque constant and i is the input direct current. For DC motor, the relationship between the current (i.e., i) and the voltage (i.e., u) can be written as

$$u = R_A i + L_A \frac{di}{dt} \tag{28}$$

where R_A is the armature winding resistance, and L_A is the armature inductance. Substituting the Equations (13) and (14) into Equation (12), it can be deduced that

$$\begin{cases} T_{L1} = K_t i_1 - J\ddot{q}_{M1} - R_\Omega \dot{q}_{M1} \\ \dfrac{di_1}{dt} = \dfrac{u_1}{L_A} - \dfrac{R_A}{L_A} i_1 \end{cases} \quad (29)$$

$$\begin{cases} T_{L2} = K_t i_2 - J\ddot{q}_{M2} - R_\Omega \dot{q}_{M2} \\ \dfrac{di_2}{dt} = \dfrac{u_2}{L_A} - \dfrac{R_A}{L_A} i_2 \end{cases} \quad (30)$$

where i_1 and u_1 are the input direct current and voltage into the DC motor mounted on the upper arm, while i_2 and u_2 are the input direct current and voltage into the DC motor mounted on the forearm.

III. Controller Design

A. Design of Traditional PID Controller

The human rotary angles (q_{H1} and q_{H2}) are the system inputs, while the currents (i_1 and i_2) are the system outputs. The control aim of this work is to make the robot position synchronize the human position. In order to implement this aim, a closed-loop control strategy is designed which is shown in Fig. 4.

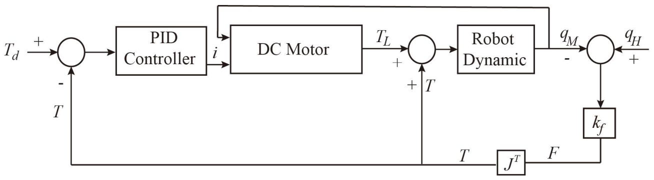

Fig. 4. The block diagram of the normal PID feedback control system.

As shown in Fig. 4, T_d is the desired torque, while the control aim is to minimize the actual force/torque (T). Therefore, T_d should be set to be zero.

$$T_{\mathrm{err}} = T_d - T = 0 - T \quad (31)$$

where T has been constructed in Eq. (1)-(6). T_{err} is the error between the desired and actual torques. Then, the PID controller is selected and designed to implement this control aim.

$$i = K_p T_{err} + K_i \int T_{err} dt + K_d \frac{dT_{err}}{dt} \quad (32)$$

where K_p, K_i and K_d are the proportional, integral and differential gains, respectively. The output is the current i which controls the DC motor to generate the driving torque T_L according to Equation (29). Then, the driving torque T_L from the DC motor and the joint torque T would together drive the robot arm to move.

B. Design of BAS-PID Controller

The BAS algorithm is a global method that is both rapid and straightforward to be implemented, allowing for the optimization of the aforementioned parameters. The concept was originally inspired by the behaviour of beetles searching for food and detecting paths. The control strategy of the BAS algorithm optimizing the PID gains is pictured in Fig. 5. Through the BAS method, the best PID gains are obtained. The fitness function of BAS concerns the synchronization of human-machine positions. In summary, the fitness function can be described in detail as follows.

$$\min \ f = \frac{1}{N} \sum_{i=1}^{N} (q_{Mi} - q_{Hi})^2 \quad (33)$$

The outcome of Equation (33) demonstrated that the optimal synchronization performance is achieved when the fitness function value is minimized.

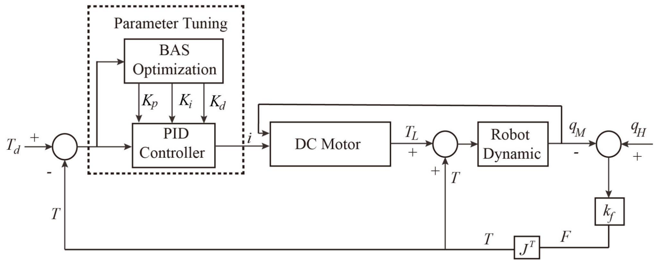

Fig. 5. The block diagram of the BAS-PID feedback control system.

For the BAS algorithm, the searching process is divided into four steps, such as parameter initialization, direction updating, position updating, and other parameter updating. Firstly, some basic parameters should be initialized. Specifically, the dimensions of the search space is indicated as n, the initial step size is described as δ_0, the maximum number of iterations is written as K, the initial search distance is defined as d_0, and the fitness function is expressed as $f()$. The second step is to update the directions of the two antennae, which can be described as follows.

$$\begin{cases} x_r = x^i + d^i b \\ x_l = x^i - d^i b \end{cases} \quad (34)$$

where x^i is the optimization target at the *i-th* iteration. x_r and x_l are severally the positions of the right and left antennae, d^i represents the searching distance at the *i-th* iteration, and b is a unit vector. In this paper, x^i and b can be specified as:

$$\begin{cases} x^i = [K_p^i \quad K_i^i \quad K_d^i] \\ b = \dfrac{rnd(n,1)}{\| rnd(n,1) \|} \end{cases} \quad (35)$$

where K_p^i, K_i^i and K_d^i are the optimized PID gains at the *i-th* iteration. $rnd()$ is a function to produce a random value ranging from [0, 1], which obeys the uniform distribution. The third step is to establish the iterative model and update the position of the beetle according to the fitness difference.

$$x^{i+1} = \begin{cases} x^i - \delta^i \cdot b, f(x_r^i) > f(x_l^i) \\ x^i + \delta^i \cdot b, f(x_r^i) \le f(x_l^i) \end{cases} \quad (36)$$

where δ^i indicates the step size at the *i-th* iteration, which is used to determine the convergence speed. The final step is to update the other parameters, such as δ^i and d^i.

$$\begin{cases} d^{i+1} = c_1 d^i + c_2 \\ \delta^{i+1} = c_3 \delta^i \end{cases} \tag{37}$$

where c_1, c_2 and c_3 are constant coefficients to realize the updating in Equation (37), and their settings are listed as c_1=c_3=0.95, and c_2=0.01. The whole optimization process is to repeat the steps 2~4 until the iteration i increases and reaches K. It is important to set the initial conditions of the related inputs and parameters. The upper and low limits of the inputs should be given, and we define that UB and LB are the upper and low limits of x^i, respectively. Thus, the initial conditions could be expressed as follows.

$$\begin{cases} x^0 = LB \\ d^0 = \dfrac{UB + LB}{2} \\ \delta^0 = \dfrac{UB + LB}{2} \end{cases} \tag{38}$$

where LB and UB are the 1×3 matrix according to the definition of x^i in Equation (35). The length of LB and UB must stay in step with x^i.

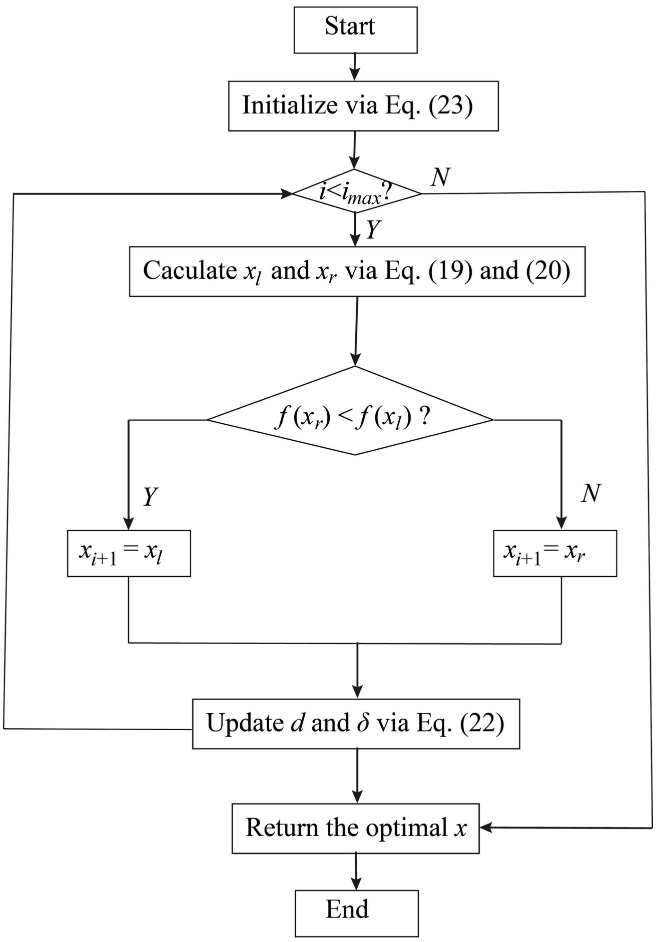

Fig. 6. The flowchart of the BAS algorithm

The entire process of the BAS algorithm optimizing the PID gains is depicted in Fig. 6. In this diagram, the symbol i_{max} indicates the maximum iteration. The $f(x)$ is calculated in the event that the entire control system is operated with PID parameters in x defined in Equation (36), and the output is the fitness defined in Equation (33). Concurrently, the values of d and δ should be updated in order to adjust the input. Finally, the optimal gains are obtained.

C. *Design of IBAS-PID Controller*

In order to enhance the performance of the BAS algorithm, the improved BAS algorithm (IBAS) is developed by incorporating the following enhancements. In Equation (35), the random function is selected to generate the value for b. However, the iterative chaotic map with infinite collapses (ICMIC) method is used to produce the b parameter. The ICMIC mapping is a chaotic model with an infinite number of mapping folds, offering the benefits of uniform traversal and rapid convergence. The mathematical expression of ICMIC mapping b value can be described as follows.

$$b^{i+1} = \sin\left(\frac{a_x \pi}{b^i}\right) \tag{39}$$

where a_x is a fixed parameter, and has a range extending from zero to positive infinity. Meanwhile, the step size δ^i in Equation (37) is rewritten in the following.

$$\delta^i = \frac{(\delta_{\max} - \delta_{\min})(i - i_{\max})}{1 - i_{\max}} + \delta_{\min} \tag{40}$$

where δ_{max} and δ_{min} restrict the range of δ value assigned to the novel step factor. The δ value is decreasing which could acquire more extensive search scope in the early stages, and implement more precise search in the later stages.

IV. RESULTS

A. *Parameter Selection*

We established the dynamics of robotic arm and DC motor with many variables and parameters. Some of the parameters are selected to be optimized, while the others should be determined ahead, which are listed in Table I. For traditional PID controller, the Z-N method is chosen to acquire the PID gains. As a result, the best parameters are figured out that K_p=30.5, K_i=1.2, and K_d=0.98 for shoulder joint, and K_p=25.6, K_i=1.26, and K_d=0.118 for elbow joint. The LB and UB are given that LB = [0.01 0.01 0.01] and UB = [100 30 2]. For the IBAS algorithm, a_x is given to be 5.8, while the δ_{max} and δ_{min} are set to be 5 and 0.1, respectively.

TABLE I.
SELECTION OF SYSTEM PARAMETERS

Name	Symbol	Unit	value
Forearm length	L_1	m	0.28
Ipper arm length	L_2	m	0.20
Mass of forearm	m_1	kg	0.25
Mass of upper arm	m_2	kg	0.35
Inertia of forearm	I_1	$kg.m^2$	0.0044
Inertia of upper arm	I_2	$kg.m^2$	0.0013
Centre of gravity position of forearm	L_{g1}	m	0.14
Centre of gravity position of upper arm	L_{g2}	m	0.11
Moment of inertia	J_Ω	$N.m$	0.0125
Damping coefficient	R_Ω	Ω	1.19
Motor torque constant	K_t	-	1.5
Armature inductance	R_A	H	1.19
Armature winding resistance	L_A	Ω	0.0112
Elastic coefficient	k_f	Kg.m^{-3}	1850

B. Comparison Results

The experiments were carried out to prove the superiority of the proposed IBAS algorithm by comparing it with other optimization algorithms, such as PSO, GA and the original BAS. The PID gains after optimization are described in Table II. The maximum iteration is set to be 100, and the fitness results of four methods are shown in Fig. 7. The performance evaluation is inversely proportional to the fitness value. After 100 iterations, the proposed IBAS-PID achieved the lowest fitness value, while the PSO-PID acquired the highest value. Additionally, the IBAS-PID requires the fewest number of iterations (approximately 29 times) to reach a stable value, while the PSO-PID requires approximately 73 iterations for stabilization. The proposed IBAS-PID shows only a little better performance than the original BAS-PID which takes 46 iterations to reach the stable goal. However, the GA-PID takes only 10 iterations for stabilization, but gains a relatively high fitness value.

TABLE II
SELECTIONS OF PID GAINS

Parts	Methods	K_p	K_i	K_d
Shoulder Joints	Normal-PID	30.53	1.24	0.98
	GA-PID	40.2	12.6	1.16
	PSO-PID	45.6	18.9	1.13
	BAS-PID	44.96	19.68	1.15
	IBAS-PID	45.35	19.28	1.17
Elbow Joints	Normal-PID	25.7	1.26	0.12
	GA-PID	35.6	8.95	0.18
	PSO-PID	34.9	9.48	0.19
	BAS-PID	35.64	8.94	0.18
	IBAS-PID	36.58	9.38	0.19

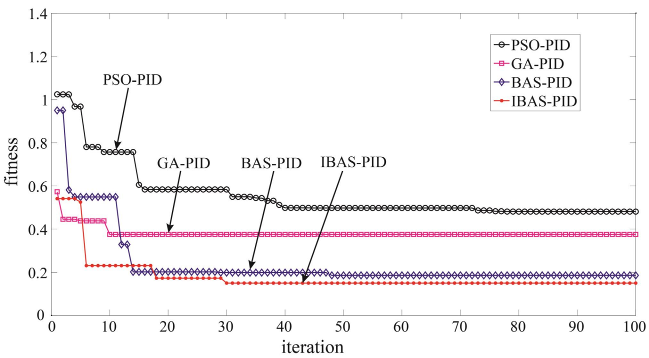

Fig. 7. Fitness of four methods after 100 iterations.

As pictured in Fig. 8, the step response is selected to evaluate the control performance. Specifically, the proposed IBAS-PID algorithm gains the lowest rise time, while the normal-PID algorithm acquires the biggest rise time. On the other side, the GA-PID and normal-PID algorithm achieve zero overshoot, while the BAS-PID algorithm realizes the biggest overshoot. As the shoulder and elbow joints are linked and affected, the parameters would show different results when they are optimized. As illustrated in Fig. 9, all the algorithms exhibit significant overshoots. The BAS-PID algorithm acquires the biggest overshoot, while the PSO-PID algorithm achieves the least overshoot. On the other side, the proposed IBAS-PID algorithm gains the least rise time, while the normal-PID algorithm obtains the biggest rise time.The specific quantitative results are depicted in Table III in terms of the rise time, overshoot, mean absolute error (MAE) and root mean square error (RMSE).

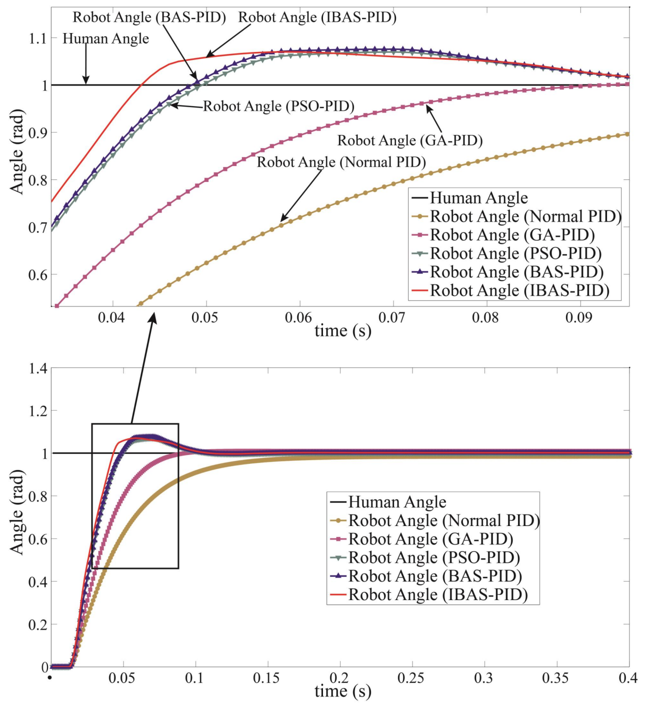

Fig. 8. Step responses of five methods for shoulder joint, including normal-PID, GA-PID, PSO-PID, BAS-PID and IBAS-PID controllers.

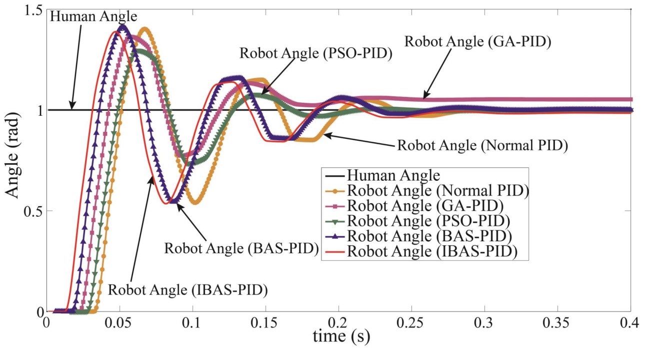

Fig. 9. Step responses of five methods for elbow joint, including normal-PID, GA-PID, PSO-PID, BAS-PID and IBAS-PID controllers.

TABLE III
QUANTITATIVE COMPARISON OF STEP RESPONSE

Parts	Methods	Over-shoot	Rise time (s)	MAE (rad)	RMSE (rad)
Shoulder Joints	Normal-PID	0	0.13	0.0297	0.0704
	GA-PID	0.01%	0.078	0.0292	0.0695
	PSO-PID	8.64%	0.046	0.0293	0.0699
	BAS-PID	9.15%	0.045	0.0256	0.0611
	IBAS-PID	8.26%	0.041	0.0248	0.0607
Elbow Joints	Normal-PID	41%	0.048	0.0504	0.1176
	GA-PID	38%	0.038	0.0416	0.0955
	PSO-PID	34%	0.042	0.0450	0.1082
	BAS-PID	42%	0.032	0.0354	0.0932
	IBAS-PID	39%	0.021	0.0338	0.0897

As described in Table III, the normal-PID controller exhibits zero overshoot, but gains the greatest rise time (0.13s), MAE (0.0297 rad) and RMSE (0.0704 rad) for the shoulder joint. For the shoulder joint, the BAS-PID controller acquires the largest overshoot compared with the other optimization algorithms. Meanwhile, the proposed IBAS-PID controller gains the least rise time (0.041 s), MAE (0.248 rad) and RMSE (0.0607 rad). As to the elbow joints, the BAS-PID controller shows the largest overshoot. Except from the metric of overshoot, the proposed IBAS-PID controller gains the least values (i.e., rise time is 0.021 s, MAE is 0.0338 rad and RMSE is 0.0897 rad) in other three metrics. The experimental results show that the proposed method gains the most optimal performance for the robot control compared with other methods.

Subsequently, the tracking effect of sinusoidal signal tracking was also tested. As depicted in Fig. 10, all the methods shows satisfactory tracking performance. Initially, the robot's shoulder joint angles exhibited considerable variation and exhibited a tendency to oscillate around the human angles. Subsequently, the robot angles exhibited a high degree of correlation with the human angles. Upon commencement of the second tracking cycle, the robot angles exhibited a slight divergence from the human angles. Upon amplifying this graph, it becomes evident that the robot angle from the IBAS-PID control follows the human angle the most closely.

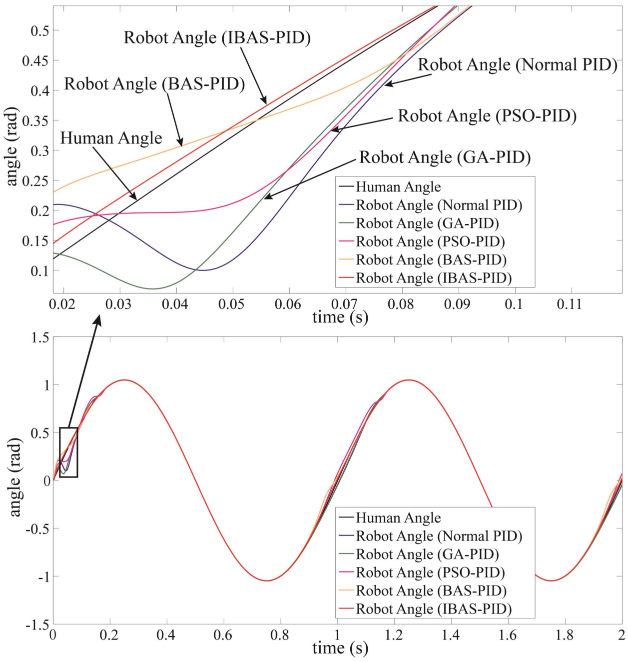

Fig. 10. Comparison of sinusoidal response for shoulder joint

Figure 11 demonstrates the tracking performance for the elbow joint. From the amplifying part, the robot angle from the proposed IBAS-PID control is the closest to the human angle, and shows the most optimal control performance. However, at most of the time, the robot angle tracks the human angle closely for all methods. Obviously, it is hard to evaluate the control performance precisely. The specific and quantitative results are figured out and reported in Table IV which includes the metrics of MAE, RMSE and time delay.

As described in Table IV, the normal-PID controller gains the worst performance in all metrics (i.e., MAE is 0.011 rad and RMSE is 0.025 rad, and time delay is 2 ms for the shoulder joint, while MAE is 0.028 rad and RMSE is 0.067 rad, and time delay is 0.9 ms for the elbow joint). Meanwhile, the proposed IBAS-PID control possesses the best control performances in all metrics (i.e., MAE is 0.009 rad and RMSE is 0.021 rad, and time delay is 0.61 ms for the shoulder joint, while MAE is 0.023 rad and RMSE is 0.052 rad, and time delay is 0.55 ms for the elbow joint). The comparison results show that the proposed IBAS-PID control gains the best performance in the optimization process of tuning PID gains.

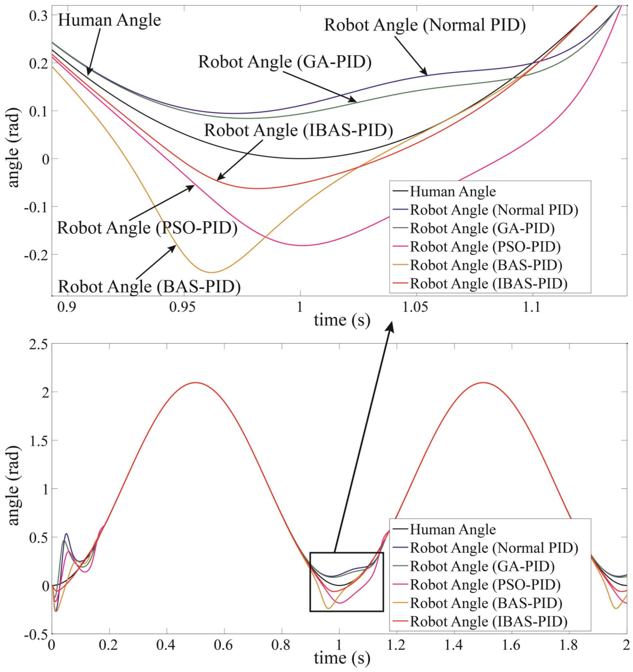

Fig. 11. Comparison of sinusoidal response for elbow joint

TABLE IV
QUANTITATIVE COMPARISON OF SINUSOIDAL TRACKING

Parts	Metric	MAE (rad)	RMSE (rad)	Time Delay (ms)
Shoulder Joints	Normal-PID	0.011	0.025	2.00
	GA-PID	0.010	0.022	0.70
	PSO-PID	0.013	0.031	0.72
	BAS-PID	0.010	0.021	0.63
	IBAS-PID	0.009	0.021	0.61
Elbow Joints	Normal-PID	0.028	0.067	0.90
	GA-PID	0.025	0.058	0.68
	PSO-PID	0.028	0.068	0.76
	BAS-PID	0.024	0.054	0.57
	IBAS-PID	0.023	0.052	0.55

V. CONCLUSION

A mathematical and computational model of the 2-DOF upper exoskeleton robot, driven by direct current motors, has been successfully constructed, simulated, and tested. The PID control is selected to control the whole system, and its gains are optimized through the proposed IBAS algorithm. Specifically, the ICMIC method is used to generate the random value, which would achieve uniform traversal and accelerate the convergence. A novel step size is proposed to realize more extensive search scope in the early stages, and more precise search in the later stages. The control performance is tested in terms of step response and sinusoidal tracking. The results show that the proposed IBAS-PID control gains the most optimal performance in terms of MAE, RMSE, rise time, and time delay in the control strategy.

REFERENCES

[1] A. J. Young and D. P. Ferris, "State of the art and future directions for lower limb robotic exoskeletons," *IEEE Trans Neural Syst Rehabil Eng*, vol. 25, no. 2, pp. 171–182, 2017.

[2] N. Masud, P. Mattsson, C. Smith and M. Isaksson, "On stability and performance of disturbance observer-based-dynamic load torque compensator for assistive exoskeleton: A hybrid approach," *Mechatronics*, vol. 69, pp. 102373, 2020.
[3] G. W. Zhang, J. Wang, P. Yang and S. J. Guo, "A learning control scheme for upper-limb exoskeleton via adaptive sliding mode technique," *Mechatronics*, vol. 86, pp. 102832, 2022.
[4] H. H. Harith, M. F. Mohd and S. N. Sowat, "A preliminary investigation on upper limb exoskeleton assistance for simulated agricultural tasks," *Applied Ergonomics*, vol. 95, pp. 103455, 2021
[5] B. Brahmi, M. Saad, C. Ochoa-Luna, M. H. Rahman and A. Brahmi, "Adaptive tracking control of an exoskeleton robot with uncertain dynamics based on estimated time-delay control," *IEEE/ASME Transaction on Mechatronics*, vol. 23, no. 2, pp. 575-285, 2018.
[6] Z. J. Li, Y. Kang, Z. Y. Xiao and W. G. Song, "Human-robot coordination control of robotic exoskeletons by skill transfers," *IEEE Trans Ind Electron*, vol. 64, no. 6, pp. 5171–5181, 2017.
[7] N. Masud, C. Smith and M. Isaksson, "Disturbance observer based dynamic load torque compensator for assistive exoskeletons," *Mechatronics*, vol. 54, pp. 78–93, 2018.
[8] Jian Zhang, Lie Yu, and Lei Ding, "Velocity feedback control of Swing phase for 2-DoF robotic leg driven by electro-hydraulic servo system," *Engineering Letters*, vol. 24, no. 4, pp. 378-383, 2016.
[9] Y. L. Zhang, L. J. Zhang and Z. L. Dong, "An MEA-Tuning Method for Design of the PID Controller," *Mathematical Problems in Engineering*, pp. 1-11, 2019.
[10] X. Yang, X. Chen, R. Xia and Z. Qian, "Wireless Sensor Network Congestion Control Based on Standard Particle Swarm Optimization and Single Neuron PID," *Sensors*, vol. 8, pp. 1265, 2018.
[11] J. G. Ziegler and N. B. Nichols "Optimum Settings for Automatic Controllers," *ASME. J. Dyn. Sys*., *Meas*., *Control*, vol. 115, no. 2, pp. 220–222, 1993.
[12] B. Zhao, H. Wang, Q. Li, J. Li and Y. Zhao, "PID Trajectory Tracking Control of Autonomous Ground Vehicle Based on Genetic Algorithm," *Chinese Control And Decision Conference*, pp. 3677-3682, 2019.
[13] G. G. Chen, F. Qin, H. Y. Long, X. J. Zeng, P. Kang, and J. M. Zhang, "Fuzzy PID Controller Optimized by Improved Gravitational Search Algorithm for Load Frequency Control in Multi-area Power System," *IAENG International Journal of Computer Science*, vol. 49, no.1, pp. 125-139, 2022.
[14] N. Qin, and X. L. Meng, "High-speed Train Rescheduling Based on a New Kind of Particle Optimization Algorithm," *Engineering Letters*, vol. 31, no.2, pp. 640-647, 2023.
[15] Y. Hou, C. X. Wang, W. C. Dong, and L. X. Dang, "An Improved Particle Swarm Optimization Algorithm for the Distribution of Fresh Products," *Engineering Letters*, vol. 31, no. 2, pp. 494-503, 2023
[16] F. S. Gharehchopogh and H. Gholizadeh, "A comprehensive survey: Whale Optimization Algorithm and its applications," *Swarm and Evolutionary Computation*, vol. 48, pp. 1-24, 2019.
[17] Q. V. Pham, S. Mirjalili, N. Kumar, M. Alazab and W. J. Hwang, "Whale Optimization Algorithm With Applications to Resource Allocation in Wireless Networks," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 4, pp. 4285-4297, 2020.
[18] Y. Fan, J. Shao and G. Sun, "Optimized PID controller based on beetle antennae search algorithm for electro-hydraulic position servo control system," *Sensors*, vol. 19, no. 12, pp. 2727, 2019.
[19] T. E. Simos, V. N. Katsikis and S. D. Mourtas, "Multi-input bio-inspired weights and structure determination neuronet with applications in European central bank publications," *Math. Comput. Simulation*, vol. 193, pp. 451–465, 2022.
[20] T. E. Simos, S. D. Mourtas and V. N. Katsikis, "Time-varying black-litterman portfolio optimization using a bio-inspired approach and neuronets," *Appl. Soft Comput*, vol. 112, pp. 107767, 2021.
[21] Y. Cheng, C. Li, S. Li, Z. Li, "Motion planning of redundant manipulator with variable joint velocity limit based on beetle antennae search algorithm," *IEEE Access*, vol. 8, pp. 138788–138799, 2020.
[22] X. Li, H. Jiang, M. Niu, R. Wang, "An enhanced selective ensemble deep learning method for rolling bearing fault diagnosis with beetle antennae search algorithm," *Mech. Syst. Signal Process*, vol.142, pp. 106752, 2020.
[23] H. Y. Long, Y. Q. He, Y. S. Xu, C. You, D. Y. Zeng, and H. Lu, "Optimal Allocation Research of Distribution Network with DGs and SCs by Improved Sand Cat Swarm Optimization Algorithm," *IAENG International Journal of Computer Science*, vol. 50, no. 2, pp. 646-661, 2023.
[24] H. X. Guan, B. Yang, H. R. Wang, D. Wu, B. X. Zhao, J. B. Liu, and T. Wu, "Multiple Faults Diagnosis of Distribution Network Lines Based on Convolution Neural Network with Fuzzy Optimization," *IAENG International Journal of Computer Science*, vol. 47, no. 3, pp. 567-571, 2020.
[25] V. N. Katsikis and S. D. Mourtas, "Diversification of time-varying tangency portfolio under nonlinear constraints through semi-integer beetle antennae search algorithm," *Applied Math*, vol. 1, no. 1, pp. 63–73, 2021.

BIBLIOGRAPHY

Fenggang Liu is an associate professor at Wuchang University of Technology in Wuhan city of China. His research interests mainly include the development of robot control and service robot.

Lang Rao is an associate professor at Wuchang University of Technology in Wuhan city of China. His research interests mainly include the development of robot control and human gait pattern recognition.

Zhaoyun He is a teacher at Wuchang University of Technology in Wuhan city of China. His research interests mainly include the development of robot control and human motion recognition.

Lie Yu is an associate professor at Wuhan Textile University in Wuhan city of China. He received his BS Degree from the Xidian University in 2009, MS Degree from the Wuhan University of Science and Technology in 2011, and PhD Degree from the Wuhan University of Technology in 2016. His research interests mainly include the development of intelligent control and rehabilitation robots.

Lei Ding is an associate professor at Wuhan Textile University in Wuhan city of China. His current research interests include robust control, biomedical science and optical coherence tomography.