Entropy-Weighted Simulated Annealing optimisation of Human-simulated Multi-mode PD-PI Control for Biped Robots

Xingyang Liu
School of Electrical Engineering
Southwest Jiaotong University
Chengdu, China
xiaoxingyang@126.com

Ferrante Neri, Daniel Cyrus
School of Computer Science and Electronic Engineering
University of Surrey
Guildford, United Kingdom
{f.neri,d.cyrus}@surrey.ac.uk

Haina Rong
School of Electrical Engineering
Southwest Jiaotong University
Chengdu, China
ronghaina@126.com

Gexiang Zhang
School of Control Eng.
Chengdu University of Information Technology
Chengdu, China
zhgxdylan@126.com

***Abstract*—Facing the challenges of biped robot walking control, this study introduces an innovative multi-mode PD-PI (Proportional-Differential Proportional-Integral) controller. The design of this controller is inspired by human-simulated intelligent control, aiming to enhance the accuracy and stability of joint motion. The setting of the parameters of the controller results in a complex multi-objective optimisation problem. To effectively tune the controller, this study incorporates a combination of entropy weight method and simulated annealing algorithm. Simulation experiments performed on the traditional PD controller and on the proposed PD-PI demonstrate that the optimised PD-PI controller significantly improves upon the traditional PD performance in terms of control precision and joint stability. These improvements highlight its potential advantages in advancing the gait of robots.**

***Index terms*—** Human-simulated intelligent control, biped robots, multi-mode control, entropy weight method, simulated annealing

I. Introduction

In the field of biped robots, the Proportional-Derivative (PD) algorithm is widely favored for its simplicity and effectiveness [1–3]. However, this method has limitations in terms of control precision, and tuning parameters for optimising performance poses a challenge. To address these issues, human-simulated intelligent control (HSIC) algorithm was proposed to enhance existing (Proportional-integral-differential) PID and PD control systems [4–6]. This algorithm employs the concept of multi-mode control, allowing it to flexibly choose the most suitable control method based on real-time error feedback. Such flexibility not only enables the control system to respond more accurately to different operating conditions but also contributes to improving overall control precision.

HSIC, by mimicking human decision-making and control strategies, can effectively enhance the performance of traditional PD control systems. The distinguishing feature of such control methods lies in their multi-mode capability, allowing flexible switching of different control strategies based on the system's feedback state to optimise overall control effectiveness. Researchers can design specific control algorithms for different operating states according to the specific requirements of practical applications. This not only provides a high degree of design freedom but also enhances system adaptability and flexibility. Building on these advantages, this study proposes a novel multi-mode PD-PI controller that combines the multi-mode control philosophy of HSIC with the robustness of traditional PD and PI control. However, parameter tuning for PD and PI control typically requires extensive experience and expertise, especially in multi-mode PD-PI control systems where different parameters for different modes make manual tuning a complex and time-consuming task.

In practical applications, the optimal control of robots often involves the optimisation of multiple simultaneous and conflicting objectives, necessitating the identification of a suitable trade-off. Furthermore, these multi-objective optimisation problems are often computationally expensive, making the use of complex algorithms impractical, for example, those employing large populations of candidate solutions. A population based multi-objective approach such as the non-dominated sorting genetic algorithm - II (NSGA-II) would often not be a viable option since handling a population and checking the dominance would result into an excessive computational cost of the optimisation process, see [7].

Based on these considerations, we propose in this study the application of the simulated annealing (SA) algorithm. SA is a popular algorithm for global optimisation used in many engineering applications [8–11]. SA perturbs a single solution and allows the acceptance of a worse solution with a probability that decreases over time. This logic enables the exploration of multimodal fitness landscapes. For multi-objective optimisation, scalarization methodsmethods that transform the multi-objective optimisation problem into a single-objective problem through the linear combination of the objective function valuesare often ineffective. This is because such methods require an a priori human-driven decision on the importance of each objective, resulting in an arbitrary modification of search and objective spaces. To address this issue, we employ the entropy weight method [12–15]. Through normalising the objective functions and calculating entropy values, it automatically derives weights, reducing the bias of subjectively setting

979-8-3503-5409-6/24/$31.00 ©2024 IEEE
DOI 10.1109/CAI59869.2024.00104

weights and better reflecting the trade-off relationships among objectives in multi-objective problems. Therefore, entropy weight simulated annealing (EWSA) algorithm is used in this paper to optimise the parameters of PD-PI controller.

The main contributions of this paper are:

1) Based on HSIC, a multi-mode PD-PI controller is designed for the motion of biped robots, aiming at improving control accuracy and joint stability.
2) The entropy weight method is integrated into SA to address the multi-objective and multi-parameter optimisation problem of the controller. The entropy weight method can determine the weight coefficient adaptively based on the importance of each objective. This adaptability ensures that the relative importance of multiple optimisation objectives is effectively considered, leading to further improvements in the performance of the controller.

The remainder of this article is organised in the following way. Section II introduces the dynamic model of a biped robot. Section III describes the proposed PD-PI controller. Section IV provides the details of the proposed optimisation algorithm and its application to the tuning of the PD-PI controller. Section V presents and comments on the numerical results of this study. Section VI provides the conclusion to this study.

II. Dynamic Framework of a Biped Robot

In this section, we introduce the dynamic model for a biped robot characterized by an n-degree of freedom (n-DOF) system. The dynamics of the robot are captured using a hybrid model, which encompasses both the continuous dynamics of a single-leg support phase (SSP) and the discrete dynamics associated with impact events. In this model, one degree of freedom is removed from the robot's model. If we consider gravity, the dynamic equation of a joint can be formulated exploiting Lagrange's method [1]:

$$D(q)\ddot{q} + C(q,\dot{q})\dot{q} + G(q) = \tau \tag{1}$$

where $\tau \in R^n$ is the torque applied to each robot's joint; $q, \dot{q}, \ddot{q} \in R^n$ represent position, velocity, and acceleration of the joint (they are vectors as each joint is in a 3D space); $D(q) \in R^{n\times n}$ and $C(q,\dot{q}) \in R^{n\times n}$ denote the inertia matrix and the Coriolis and centripetal forces; at last $G(q)$ formalises the effect of gravity on the robot.

In the Lagrangian framework, a discontinuity is observed in the model when the leg of the robot makes contact with the ground. Consider a function H that represents the vertical distance between the end of the robot's swinging leg and the ground. The set $S = \{(q,\dot{q}) \mid H(q) = 0, \bigtriangledown H(q)\dot{q} < 0\}$ defines the state $(q,\dot{q})$ at the moment of ground contact by the swinging leg.

Drawing from the work presented in [1], the model describing the impact is:

$$\begin{cases} q(t^+) = q(t^-) \\ \dot{q}(t^+) = Z(q(t^-))\dot{q}(t^-), \end{cases} (q(t^-),\dot{q}(t^-)) \in S \tag{2}$$

In this context, (-, +) represent the states immediately before and after the impact, respectively, and Z is the transition matrix.

The robot's entire dynamic model is thus constituted by the combination of equations (1) and (2).

III. Human-simulated PD-PI controller

In control engineering some standard controllers include proportionate-derivative (PD), proportional-integral (PI) and proportional-integral-derivative (PID). PD controllers are suitable to control the present state and rate of change of a system. PI controllers are appropriate when it's necessary to eliminate any steady-state error in the system. PID controllers are versatile and suitable, being able to address both present and past errors as well as predict future behavior. Furthermore, a PD-PI controller is a hybrid or combination controller that incorporates elements of both PD and PI controllers. With respect to the PID, the PD-PI controller benefits from the fast dynamic response offered by the PD. In the biped robot control, the PID response would not be fast enough to guarantee a high dynamic performance. For this reason we propose here the use of a PD-PI controller and we propose the human-simulated implementation illustrated in the PD-PI controller in Fig. 1 and Table I.

The proposed controller employs PI control when the error state is ψ_1 and switches to PD control when the error state is ψ_2 or ψ_3. In situations with large errors, PD control is capable of achieving a rapid response. For smaller errors, the integral term in PI control aids in eliminating steady-state errors, thereby providing more accurate control. Additionally, at lower error speeds, the integral term contributes to smooth control, preventing excessive oscillations and enhancing the joint stability.

IV. Entropy Weigh Simulated Annealing Algorithm: Optimisation Approach and Applications

In this section, the multi-mode PD-PI controller is applied to the joint control of the biped robot BHR-6s, and the entropy weighted simulated annealing (EWSA) algorithm is employed to optimise the parameters of this controller.

A. Entropy weight method

The entropy weight method is a multi-criterion decision method used to determine the weight coefficients of multiple criteria [16]. The basic idea is to determine the relative importance of each criterion according to its information entropy, so as to obtain the weight coefficient.

Specifically, the entropy weight method first calculates the information entropy of each criterion, which is an indicator to measure the degree of information dispersion within the criterion. The larger the entropy value, the more dispersed the information within the criterion, the smaller the weight. On the contrary, the smaller the entropy value, the more concentrated the information inside the criterion and the greater the weight.

TABLE I
HUMAN-SIMULATED PD-PI CONTROL ALGORITHM

Feature Condition	Control Algorithm	Remark
$\psi_1 : \lvert e_q \rvert \leq \delta_1 \bigcap \lvert \dot{e}_q \rvert \leq \delta_1$	$K_p e_q + K_i \int e_q dt$	PI
$\psi_2 : e_q > \delta_1 \bigcup (0 \leq e_q \leq \delta_1 \cap \lvert \dot{e}_q \rvert > \delta_1)$	$K_p e_q + K_d \dot{e}_q$	PD
$\psi_3 : e_q < -\delta_1 \bigcup (-\delta_1 \leq e_q \leq 0 \cap \lvert \dot{e}_q \rvert > \delta_2)$	$K_p e_q + K_d \dot{e}_q$	PD

Then, the weight coefficient is calculated according to the information entropy of each criterion. The calculation process of the weight coefficient is to divide the information entropy of each criterion by the sum of the information entropy of all criteria to obtain the relative weight of each criterion.

The workflow for determining weights using the entropy weight method is as follows:

1) Input the original multi-objective evaluation data.
2) Normalise the data.
   To eliminate the effects of data dimensions and numerical magnitude, the original data must first be standardized. In eq. (3), y_{ij} represents the original data value of the ith evaluation object on the jth evaluation criterion.

$$y_{ij} = \frac{y_{ij} - min\{y_{i1}, y_{i2}, ..., y_{in}\}}{max\{y_{i1}, y_{i2}, ..., y_{in}\} - min\{y_{i1}, y_{i2}, .., y_{in}\}} \tag{3}$$

3) Calculate the entropy of each objective.
   For each evaluation criterion, compute its entropy value e_j. Here, $k = 1/lnn$, where n is the number of evaluation objects.

$$e_j = -k \sum_{i=1}^{n} y_{ij} ln y_{ij} \tag{4}$$

4) Compute the divergence redundancy based on the entropy values.
   To obtain the weight of each evaluation criterion, it's necessary to first compute its divergence redundancy.

$$d_j = 1 - e_j \tag{5}$$

5) Determine the weight of each objective using the divergence redundancy.
   Lastly, determine the weights ω_j of each evaluation criterion based on their divergence redundancy. Here, r represents the number of evaluation criteria.

$$\omega_j = \frac{d_j}{\sum_{j-1}^{r} d_j} \tag{6}$$

B. Simulated annealing algorithm

SA is a single solution heuristic optimisation algorithm used in both combinatorial and continuous spaces. Its fundamental concept is inspired by the annealing process of solids, simulating the random movement of particles in a solid by gradually lowering the temperature to eventually reach a stable state. This is a probability-based stochastic optimisation algorithm that, during the process of searching for the global optimum, not only accepts better solutions but also, based on the Metropolis criterion, probabilistically accepts worse solutions, as illustrated in the equation below.

$$P = \begin{cases} 1 & f(i+1) \leq f(i) \\ exp\left(-\frac{f(i+1)-f(i)}{T_i}\right) & f(i+1) > f(i) \end{cases} \tag{7}$$

where, $f(i)$ is the function value of generation i, and T_i is the current temperature. As the algorithm approaches the global optimal, the probability P gradually approaches 0, and accepting the poor solution makes it possible for the algorithm to jump out of the local optimal and search for the global optimal solution.

The working steps are as follows:

1) Choose an initial solution s and set a high initial temperature T_0.

2) For each temperature, that is, T_i, a new solution s' is generated from the current solution s (add a small random disturbance), and it is decided whether to accept the new solution according to equation (7).

3) Lower the temperature T in some way, and the method of this paper is as follows:

$$T_{i+1} = \alpha \times T_i \qquad (0 < \alpha < 1) \tag{8}$$

4) Iterate until the termination condition is reached, such as when the temperature falls below a preset threshold, or a certain number of iterations have been completed, and then output the optimal solution s.

C. Human-simulated PD-PI controller parameter optimisation

In this section, entropy weight method is combined with SA, and it is applied to the parameter optimisation process of a biped robot (see Figure 1). This paper focuses on a biped robot with four active joints, each corresponding to a controller. Among these four controllers, the parameters to be optimised are $K_p : (k_{p1}, k_{p2}, k_{p3}, k_{p4})$, $K_i : (k_{i1}, k_{i2}, k_{i3}, k_{i4})$, $K_d : (k_{d1}, k_{d2}, k_{d3}, k_{d4})$, and the threshold value δ_1. To evaluate the performance of the controller, we employ a multi-objective fitness function, which encompasses two key metrics: joint angle error and the stability of the joint (based on joint angular velocity error). The weight parameters for these two metrics are ω_1 and ω_2, respectively.

Figure 2 illustrates the process of optimising the parameters of a PD-PI controller for a biped robot using the EWSA algorithm. Initially, the parameters to be optimised, including K_p, K_i, K_d and δ_1, are extracted from the robot's controller.

$$\mathbf{x} = (k_{p1}, k_{p2}, k_{p3}, k_{p4}, k_{i1}, k_{i2}, k_{i3}, k_{i4}, k_{d1}, k_{d2}, k_{d3}, k_{d4}, \delta_1) \tag{9}$$

$$x_i' = x_i + \Delta x_i \quad (i = 1, 2, ..., 13) \tag{10}$$

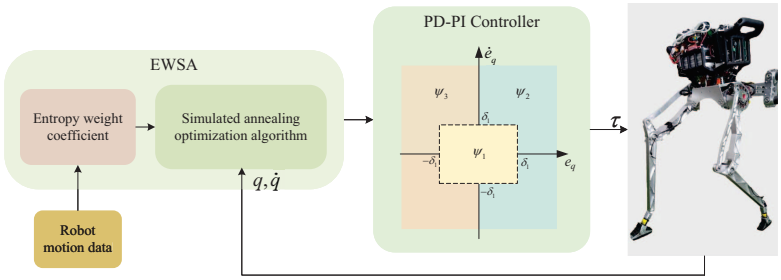

Fig. 1. Overall scheme of the proposed robotic control system comprising EWSA, the PD-PI controller and the biped robot.

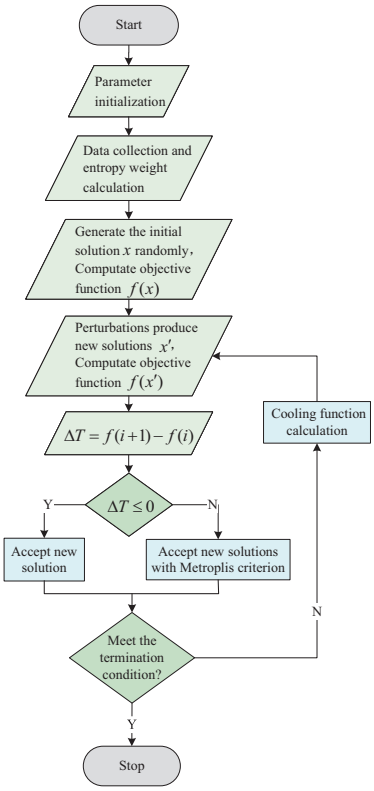

Fig. 2. EWSA to optimise the parameters of PD-PI controller.

where, x_i represents each parameter in $\mathbf{x}$, Δx_i is a small random perturbation, and x_i' denotes each parameter after adding the perturbation.

$$\Delta x_i = r_i \times m_d \times \frac{T}{T_0} \tag{11}$$

in this case, r_i is a number randomly drawn from a uniform distribution, typically within the range $[-1, 1]$. m_d is the maximum amplitude of the perturbation, a predetermined parameter that determines the maximum range of perturbation at the highest temperature. T is the current temperature, while T_0 is the initial temperature.

The entropy weight method is then introduced, it takes motion data from the robot and, through normalizing the objective function and computing entropy values, automatically derives weight parameters. This reduces subjective biases in weight assignment.

Next, the multi-objective optimisation problem is transformed into a single-objective problem (see eq.(14)), considering both joint angle error and joint stability as key indicators and incorporating entropy weight parameters into the objective function. Through the SA algorithm, the system searches for the optimal solution in the solution space, continuously updating parameter values to maximize the objective function. After a series of iterative optimisations, the obtained optimised parameters are applied to the PD-PI controller of the biped robot, aiming to enhance its gait control performance.

$$f_1 = \sum_{i=1}^{4} (q_i - q_{ri})^2 \tag{12}$$

$$f_2 = \sum_{i=1}^{4} \dot{q}_i^2 \tag{13}$$

where, $fitness_1$ is the joint error fitness, q_i is the actual joint angle, q_{ri} is the target joint angle. $fitness_2$ is the joint stability fitness, $\dot{q}_i$ is the actual joint angular velocity, and the target joint angular velocity is zero.

$$f = \omega_1 \times f_1 + \omega_2 \times f_2 \tag{14}$$

$fitness$ is the normalized fitness, and $\omega_1 + \omega_2 = 1$. Algorithm 1 highlights the functioning of the optimiser.

Algorithm 1 Functioning of the optimiser to determine the best control parameters.

Require: Initial base solution $\mathbf{x} = (k_{p1}, k_{p2}, k_{p3}, k_{p4}, k_{i1}, k_{i2}, k_{i3}, k_{i4}, k_{d1}, k_{d2}, k_{d3}, k_{d4}, \delta_1)$, parameters T_0, m_d, α.
Run a simulator of the biped robot for a given task with the control parameters $\mathbf{x}$. Collect the simulation data.
With the collected simulation data calculate f_1 and f_2 according to eq. (12) and (13)
Use entropy weight method described in Section IV-A to determine ω_1 and ω_2
Calculate f according to eq. (14)
while budget conditions are not satisfied **do**
    Perturb each element of $\mathbf{x}$ according to eq.s (10) and (11)
    Select the new base solution $\mathbf{x}$ according to eq. (7)
    Update the temperature according to eq. (8)
end while
Output: The best individual $\mathbf{x}$.

V. Numerical Results

To verify the effectiveness of the proposed method, we constructed a numerical simulation model based on the dynamics model of the biped robot BHR-6s in the Matlab environment. The model parameters are shown in Table II. We applied the EWSA method to optimise the parameters of the PD and PD-PI controllers and conducted a comparative analysis with PD and PD-PI. In the experiments, we set the target velocity of the robot to be 0.4 m/s, with a duration of 4 s, and a target displacement of 1.6 m. Our optimisation strategy targeted two main objectives: minimizing the error in joint angles and ensuring the stability of joint motions (i.e., minimizing the error in joint angular velocities).

TABLE II
Parameter Settings for BHR-6s Simulation

Biped	Mass (kg)	Length (m)	Inertia $(kg \cdot m^2)$
Torso	5.580	0.257	0.043320
Thigh	0.548	0.401	0.000680
Shank	0.813	0.300	0.000034

After the optimisation process, we obtained the optimal parameters for both controllers, as shown in Table III. At the same time, the weight coefficients obtained through the optimisation process are $\omega_1 = 0.67$ and $\omega_2 = 0.33$, respectively.

We apply the four sets of optimised parameters to the walking control of the robot. Through simulation tests, we obtain the average value and standard deviation of the joint angle error for each parameter set, as well as quantitative indicators representing stability, as shown in Table IV. From the results, it is evident that the method proposed in this study outperforms its competitor in these two key indicators. This not only demonstrates the improvement in control accuracy but also emphasizes its high level of stability.

TABLE III
Optimisation of Controllers Parameters of Different Methods

Methods	Parameter	Joint 1	Joint 2	Joint 3	Joint 4
PD	K_p	950	950	250	180
	K_d	95	95	25	18
PD-PI	K_p	500	500	300	150
	K_i	37	37	21	7
	K_d	50	50	30	15
EWSA-PD	δ_1	1.0	1.2	1.8	1.8
	K_p	330	180	363	199
	K_d	33	18	25	17
EWSA-PD-PI	K_p	450	240	400	220
	K_i	37	17	32	16
	K_d	45	24	40	20
	δ_1	1.2	1.8	2.3	2.6

Fig. 3 presents the joint angle control tracking curves for the PD, PD-PI, EWSA-PD, and EWSA-PD-PI methods. A comparison reveals that PD-PI demonstrates superior control precision compared to PD. After applying EWSA optimisation, both methods exhibit varying degrees of improvement, with EWSA-PD-PI showing the best performance. On the other hand, in Fig. 4, angular velocity curves representing joint stability are provided. Comparative analysis indicates that

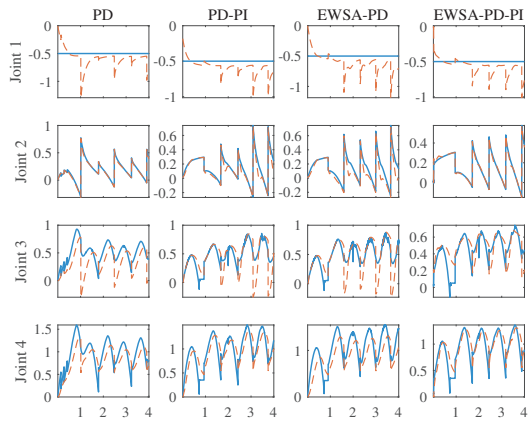

Fig. 3. The trajectories of joint angle under different methods.

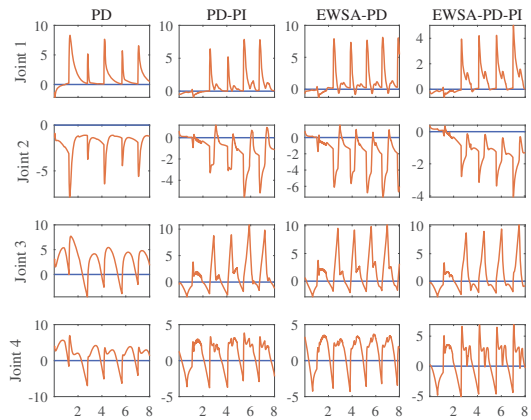

Fig. 4. Joint angular velocity trajectories under different methods.

the angular velocity variation is the smallest for the EWSA-PD-PI method, further emphasizing its superiority in terms of joint stability. Figs. 5 and 6 show the comparison between the centroid velocity and displacement of the robot. From the figures, it is clear that the EWSA-PD-PI method exhibits more stable velocity control and more accurate displacement control, while the orther control methods exhibit significant velocity oscillations leading to displacement far from the

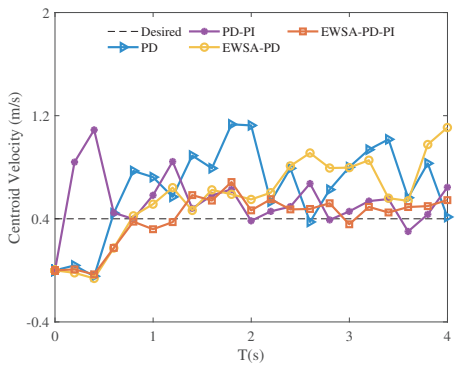

Fig. 5. The velocity of the center of mass under different methods.

TABLE IV
JOINT ANGLE ERROR AND STANDARD DEVIATION WITH DIFFERENT OPTIMISATION CONTROLLERS

		Mean				Standard Deviation			
		Joint 1	Joint 2	Joint 3	Joint 4	Joint 1	Joint 2	Joint 3	Joint 4
Joint error	PD	-0.1156	-0.0094	-0.1447	-0.1886	0.1440	0.0575	1.1984	0.2463
	PD-PI	-0.1011	-0.0057	-0.1423	-0.1047	0.1234	0.0110	0.2768	0.2235
	EWSA-PD	0.1152	0.0644	0.0252	0.1445	0.1365	0.0157	0.2573	0.1861
	EWSA-PD-PI	0.0075	0.0034	-0.0122	-0.0205	0.0935	0.0445	0.2155	0.2089
Joint stability	PD	0.9282	-1.5990	1.8112	1.2725	2.8172	2.6486	4.0014	3.3784
	PD-PI	0.6836	-1.2603	1.3609	1.0054	1.8431	1.8673	2.3510	2.2884
	EWSA-PD	0.8189	-1.6492	1.8627	1.4692	2.3523	2.3274	3.1808	2.2455
	EWSA-PD-PI	0.5932	-1.2153	1.2363	0.8956	1.6436	1.0431	1.5663	1.7905

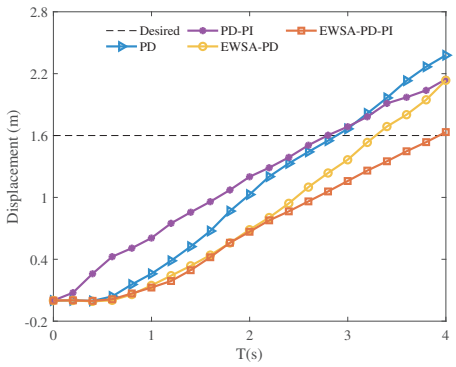

Fig. 6. The displacement under different methods.

target position. In conclusion, through this series of simulation experiments, we have demonstrated the excellent performance of the EWSA-PD-PI controller in ensuring the joint stability and accuracy of the robot's walking.

VI. CONCLUSION

In this paper, a human-simulated multi-mode PD-PI controller for biped robots was designed based on HSIC, aiming to improve the joint accuracy and stability of traditional PD algorithms. By introducing a multi-mode control strategy, we can better adapt to different control states, thereby achieving superior control performance. Additionally, to optimise the controller parameters, we combine the entropy weight method with SA to solve the complex optimisation problem with multiple objectives and parameters. The entropy weight method can adaptively determine the weight coefficients based on the importance of each objective, while the SA can globally search the parameter space to find the optimal solution. Through this combination optimisation method, we can obtain more reasonable and optimised controller parameters. Finally, through a series of tests and simulation experiments, we verify the superiority of EWSA-PD-PI, demonstrating the effectiveness of our proposed control method in improving the joint stability and accuracy of biped robot walking.

ACKNOWLEDGEMENTS

This work was supported by NSFC (61972324), Sichuan Science and Technology Program (2023NSFSC1985), Chengdu University of Information Technology (KYTD202212), Beijing Advanced Innovation Center for Intelligent Robots and Systems (2019IRS14), and Jiangsu Distinguished Professor programme.

REFERENCES

[1] S. Kolathaya, "Local stability of pd controlled bipedal walking robots," *Automatica*, vol. 114, p. 108841, 2020.

[2] K. Shishir, "Pd tracking for a class of underactuated robotic systems with kinetic symmetry," *IEEE Control Systems Letters*, vol. 5, no. 3, pp. 809–814, 2020.

[3] Z. Qiu, C. He, and X. Zhang, "Multi-agent cooperative structural vibration control of three coupled flexible beams based on value decomposition network," *Engineering Applications of Artificial Intelligence*, vol. 114, p. 105002, 2022.

[4] L. N. Z. Qijian, "A self tuning algorithm capable of simulating human control strategy," *Journal of Chongqing University (Natural Science Edition)*, vol. 1, pp. 135–145, 1985.

[5] J. Chenglin and Z. Xiaoming, "Application of humanoid predictive control in biped robot walking control," *sensor and microsystem*, vol. 33, no. 3, pp. 157–160, 2014.

[6] R. Colom, S. Karama, R. E. Jung, and R. J. Haier, "Human intelligence and brain networks," *Dialogues in clinical neuroscience*, 2022.

[7] V. Roostapour, A. Neumann, and F. Neumann, "Single- and multi-objective evolutionary algorithms for the knapsack problem with dynamically changing constraints," *Theoretical Computer Science*, vol. 924, pp. 129–147, 2022.

[8] C. Li, F. You, T. Yao, J. Wang, W. Shi, J. Peng, and S. He, "Simulated annealing particle swarm optimization for high-efficiency power amplifier design," *IEEE Transactions on Microwave Theory and Techniques*, vol. 69, no. 5, pp. 2494–2505, 2021.

[9] S. A. Davari, V. Nekoukar, C. Garcia, and J. Rodriguez, "Online weighting factor optimization by simplified simulated annealing for finite set predictive control," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 1, pp. 31–40, 2021.

[10] M. Lin, C. Yan, J. Meng, W. Wang, and J. Wu, "Lithium-ion batteries health prognosis via differential thermal capacity with simulated annealing and support vector regression," *Energy*, vol. 250, p. 123829, 2022.

[11] Y. Zhou, W. Xu, Z.-H. Fu, and M. Zhou, "Multi-neighborhood simulated annealing-based iterated local search for colored traveling salesman problems," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 9, pp. 16 072–16 082, 2022.

[12] H. Li, W. Wang, L. Fan, Q. Li, and X. Chen, "A novel hybrid mcdm model for machine tool selection using fuzzy dematel, entropy weighting and later defuzzification vikor," *Applied Soft Computing*, vol. 91, p. 106207, 2020.

[13] M. A. Alao, T. R. Ayodele, A. Ogunjuyigbe, and O. Popoola, "Multi-criteria decision based waste to energy technology selection using entropy-weighted topsis technique: The case study of lagos, nigeria," *Energy*, vol. 201, p. 117675, 2020.

[14] J. Zhang, Q. Zhang, L. Wu, and J. Zhang, "Identifying influential nodes in complex networks based on multiple local attributes and information entropy," *Entropy*, vol. 24, no. 2, p. 293, 2022.

[15] J. Wang, X. Ma, Z. Xu, and J. Zhan, "Regret theory-based three-way decision model in hesitant fuzzy environments and its application to medical decision," *IEEE Transactions on Fuzzy Systems*, vol. 30, no. 12, pp. 5361–5375, 2022.

[16] "Revealing the benefits of entropy weights method for multi-objective optimization in machining operations: A critical review," *Journal of Materials Research and Technology*, vol. 10, pp. 1471–1492, 2021.