Received: 7 November 2023 | *Revised: 22 December 2023* | *Accepted: 28 December 2023*
DOI: 10.37917/ijeee.21.2.20 *Early View* | *December 2025*

Open Access

Iraqi Journal for Electrical and Electronic Engineering
Original Article

IJEEE University of Basrah College of Engineering

Design a Stable an Intelligent Controller for an Eight-legged Robot

Fatima H. Khayoun*, Ammar A. Aldair
Electrical Engineering Department, University of Basrah, Iraq

Correspondance
*Fatima H. Khayoun
Electrical Engineering Department
University of Basrah, Basrah, Iraq
Email: fatimaahameed98@gmail.com

Abstract
At recent days, the robot performs many tasks on behalf of humans or in support of humans. Among the most prominent benefits of robots for humans are removing the risk factor from humans, completing routine tasks for humans, saving a lot of time and effort, and mastering work. This paper presents a model of an eight-legged robot equipped with an intelligent controller that was simulated using MATLAB. The designed structure contains 24 controllers, three for each leg, to provide flexibility in movement and rotation. Proportional Integral Derivative (PID) controller has been used in this work , each leg contains three PIDs. A particle swarm optimization algorithm (PSO) was used to adjust the parameters of the PID controller (K_p , K_i and K_d). The structure of eight legs robot with controller is implemented using Simscape Multibody in the MATLAB program, where the movement of the eight-legged robot is visualized and analyzed without the need for complex analysis associated with a mathematical model. The simulation results were conducted in a three-dimensional environment and were presented in two scenarios . The first was implementing and simulating the robot without using a controller, which leads to the robot falling at the starting point. The second was when a PID controllers are used with the system, where better movement was obtained. Finally, the robustness of the controller was verified by changing the load that the robot bears.

Keywords
Eight-Legged Robot, PID Controller, Particle Swarm Optimization (PSO).

I. Introduction

In the last decades, many types of robots were developed. Starting with a simple mobile robot that can be readily driven by two powered wheels and progressing to advanced structures that use four wheels to drive the motion of a mobile robot. When four motorized wheels are used, at least four controllers are required to control the movement of such mobile robots [1]. The legged robots are needed to navigate rugged and uneven terrain for probing and reconnaissance because wheeled vehicles lack the necessary additional mobility [2]. Over the last decade, the robotics community has made tremendous efforts to improve robot capabilities in order to fulfill the growing demands of burgeoning application domains. Robots began to work in shared spaces with human users, gaining access to previously prohibited contexts such as public places, collaborative industrial settings, and residences. This new generation of collaborative robots must demonstrate its interaction capabilities with humans and the environment in order to attain high levels of reliability, safety. As a result, performance evaluation has become increasingly important in numerous fields of robotics. Last years have seen the introduction of extremely performant generations of legged robots with amazing biomimetic skills in unstructured natural environments. While robotic locomotion on flat surfaces has received extensive attention in the scientific literature, little efforts have been made to systematically test locomotion abilities in less-than-ideal conditions [3]. In a tough environment, a legged robot can jump and pass hurdles safely and smoothly.

The modeling system of a walking robot that has many legs contains a high degree of nonlinearity and uncertainty. As a result, a strong controller is necessary to steer the walking robot's mobility. The cost of multiple-leg walking robots is undeniably high. As a result, before buying a physical robot, a dependable controller must be appropriately constructed and its performance thoroughly analyzed. Furthermore, simulation tools enable researchers to evaluate and estimate a robot's overall performance and optimize the path planning of its process. For these reasons, using a simulator program is helpful because it can save time and money [4]. There are many programs that have been used to simulate the movement of robots such as Microsoft robotics studio (MSRS) and NET framework 4.5 that used by authors in reference [5]. Another study used SolidWorks and SimScape Multibody packing to model and investigate the performance of a platform that included parallel robots and a developed controller [6]. Other researchers designed a control method, visualized the outcomes, and analyzed simulation findings using simulation tools such as 20-sim Mathematica, Dymola/Modelica, Matlab/Simulink, and others [7–11].

The SimScape Multibody toolbox in MATLAB is used in this work to simulate and display the motion of an eight-legged walking robot using a proposed controller. The PID approach is utilized to design the walking robot's controller system. The particle swarm optimization (PSO) algorithm is used to find the best PID controller parameters. By varying the carried weight of the walking robot, the effectiveness and robustness of the developed controller are investigated. The acquired simulation results demonstrate the validity of the suggested robot controller, and overall performance has been enhanced.

II. Modeling The Eight-Legged Robot by Using SIMSCAPE Multibody Toolbox

Simulating robots with a controller is considered a very important issue to study the strength and effectiveness of the controller to drive the robot under any disturbances, as researchers used many simulation programs to conduct this problem [12]. The Simscape Multibody toolbox used to simulate a three-dimension robotics and construction equipment and it includes a variety of libraries and Simulink blocks that can be used to design robots with any architecture, including mobile robots, manipulators with various numbers of joints, and walking robots [13]. It also includes simulation and control interfaces that help users in obtaining necessary simulation results and displaying actual robot motion. Fig. 1 shown below indicates the eight-legs robot in simscape multibody in Matlab. This robot consists from torso with eight legs and each leg contain three joints (hip, knee and ankle).

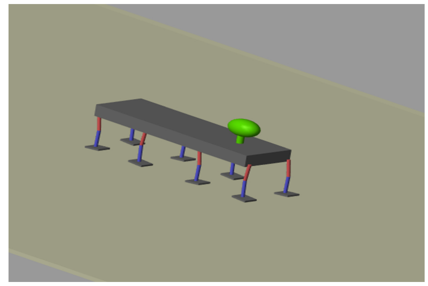


Fig. 1. visual display of eight-legs robot in simscape multibody in Matlab

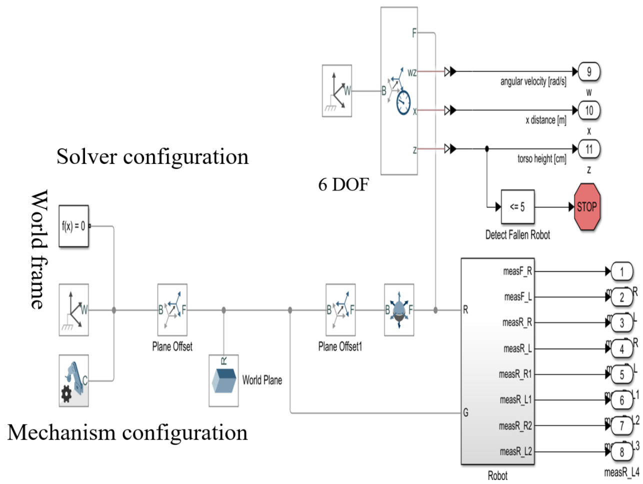


Fig. 2. the main structure of eight legs walking robot

The Fig. 1 shows the visual display of the eight-legged robot built by simscape multibody toolbox in MATLAB.

Fig. 2 shows the main components of the eight-legged robot that are linked together. Begin with the block of the world frame which is connected with the solver configuration and mechanism configuration blocks.

Fig. 3 shows the robot sub-system that is shown in Fig. 2 that contains eight legs.

As shown above in Fig. 4 the subsystem of leg in Fig. 3 that contains three joints (hip, knee and ankle) and each joint contains one PID controller with torque. It also contains the upper and lower legs and foot.

Fig.5 show the joint subsystem in Fig. 4 that contains a joint that is controlled by PID controller . In addition, there is a revolute joint that used to constrain the movement of two random frames connected to the joint's base and follower frames to a single, pure rotation around a shared axis.

Fig. 6 show the PID controller used to control the movement of the joints of an eight-legged robot.

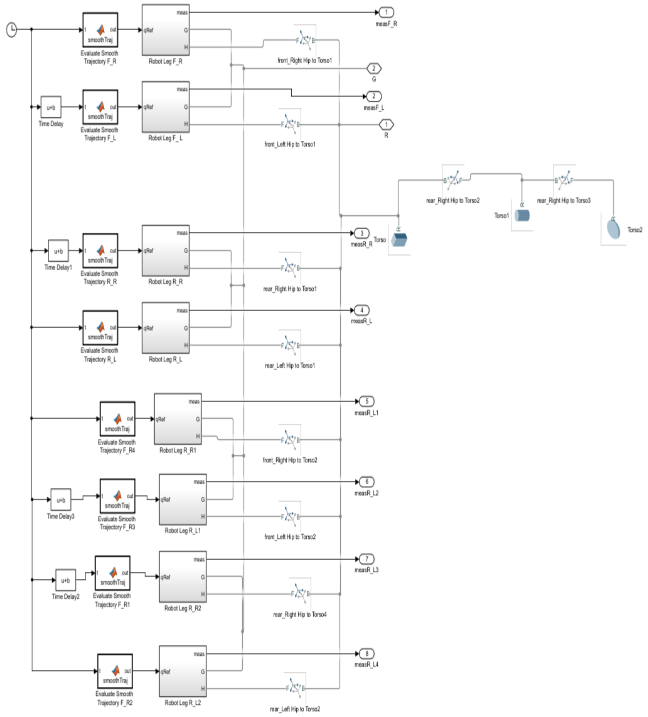
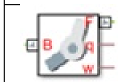
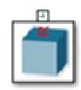


Fig. 3. robot sub-system that contain eight legs

Table I shown below contains the blocks used in building the eight-legged robot and a description of each of them. These blocks are found in the MATLAB Simscape Multibody Toolbox.

TABLE I.
SIMSCAPE BLOCKS

blocks	name	describtion
	Solver Configuration	establishes the simulation's setup parameters
	World Frame	construction of the mechanicalmodel's reference point. Global Frame.
	Mechanism Configuration	Setting up the initial mechanical and simulation parameters
	Rotational Joint	To interpret motion at angles between the actuators and the fixed base, a rotational joint is used.
	Solid Block	The solid blocks offer solid characteristics.
	PS-Simulink	PS-Simulink Converter that converts the input physical signal into a simulink output signal or vice versa

III. CONTROLLER SCHEME

In this part, the control system for the eight-legged robot was designed and implemented. The robot consists from eight legs and each leg consists of three joints, and each joint is controlled by one PID controller. Thus, the robot contains 24 PID controllers in total. The control system for one leg is introduced and the same issue will be applied to the other legs. Each leg contains revolve joint. The block diagram of the revolute joint has two inputs: B (is the port that connects the current link to the prior one) and the second input t (it denotes the control signal provided to the torque). The output of the revolve joint are: F (is the point of contact between the current link and the next one). The w and q are the speed and angle of a current link, respectively. The revolve joint shown in Fig. 7.

In this paper, two scenarios are studied for the robot, the first scenario is an open-loop system and the second scenario is a closed-loop system with a PID controller and the simulation results between the two scenarios are compared:

A. *Open Loop System*

In this case, the eight legs robot is built without controller and it is tested to see if it walked stably at a straight line. The results show that the robot fell at the beginning of the simulation and lost its motion stability as shown in Fig. 8.

Fig. 9 shows that the robot falls when the simulation begins. This is due to the lack of a controller in the joints of the legs, which helps in straight and stable movement. Therefore, in the second scenario, the PID controllers are used to control the movement of the eight-legged robot.

B. *Closed-loop System with PID Controller*

The PID controller is an important part of the process industry's control loop [14]. Due to its straightforward structure, simplicity of design, and low implementation costs, traditional proportional plus integral plus derivative (PID) controllers continue to be the most extensively used approach in industry for many control applications [15]. The abbreviation PID stands for the first letters of words of proportional-integral-

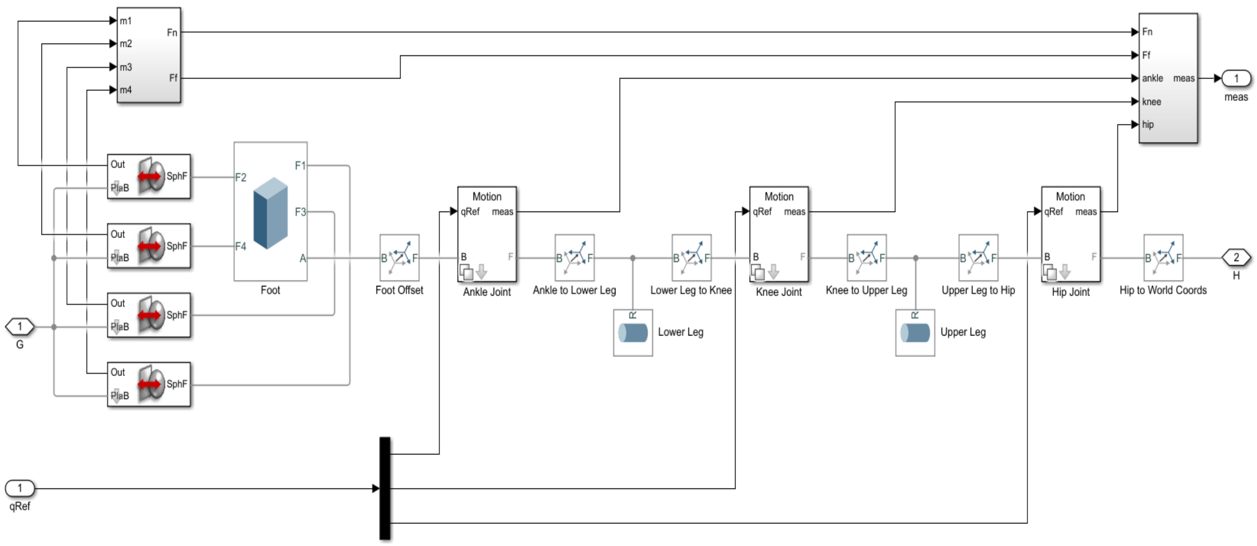


Fig. 4. One leg of eight legs robot (leg subsystem)

TABLE II.
SHOWS THE ROBOT'S DIMENSIONS.

Parameter's Name	Dimensions (cm)
Torso x, y and z	5,10 and 7
Upper leg length	10
Upper leg radius	1.25
Lower leg length	10
Lower leg radius	0.75
Foot x, y and z	8,6 and 10

derivative and these parameters can be adjusted appropriately to improve plant performance, reduce overshoot, eliminate steady-state error, and encourage system stability [16]. Choosing the appropriate PID gains is the basic controller's main problem. If fixed gains are applied, the controller might not provide the required control performance when plant features and operating conditions change. As a result, tuning is necessary to make sure the controller can handle changes in the plant [17]. In order to obtain the best performance of the PID controller, the parameters of the PID (KP, KI and KD) must be tuned. There are many optimization algorithms that perform tuning to improve the performance of the controller, including these algorithms: Genetic Algorithm (GA), Particle Swarm Optimization (PSO), Artificial Bee Colony (ABC) and Ant Colony Optimization (ACO) [18]. The main equation for the output of a PID controller is:

$U(t) = K_p e(t) + K_i \int e(t)dt + K_d \dot{e}(t)$

where e(t) represents the difference in error between reference and actual values. The actual value is the true output response of the plant, while the reference value is the signal that is delivered based on the desired input of the plant. The block diagram of the PID controller can be shown in figure(9).

In this paper, we used the PSO optimization algorithm to adjust the parameters of the PID controller used in the eight-legged robot. Russell and James first presented the well-known PSO algorithm in 1995, drawing inspiration from the social behavior of fish and birds. Individuals relate to particles in PSO, which is regarded to be an adaptive algorithm based on a social-psychological metaphor. PSO adjusts by stochastically going back to previously successful regions [19].Position update and velocity update are the two primary operators in PSO. The PSO basic design is easy to excite and just requires a few parameter adjustments. The behavior of the PSO mechanism, which is based on particles, is designed to mimic the success of nearby particles. The objective is to locate the best path across a variety of search spaces [20].

Where, denotes the position of the particle and stands for the velocity of the particle. The following equations represent the updated PSO parameters:

$$V_i(t+1) = \omega * V_i(t) + C_1 r_1(t) * [P_{besti} - P_i] + C_2 r_2(t) * [g_{besti} - P_i] \quad (1)$$

$$P_i(t+1) = P_i(t) + V_i(t+1) \quad (2)$$

Where r_1 , r_2 are random numbers selected from the interval [0,1]. ω is inertia weight. $P_besti = P_best1 P_best2 \ldots\ldots P_bestd$

is the personal optimal solution of the particle
$g_{besti} = g_{best1}g_{best2}\cdot\cdots\cdot g_{bestd}$ is the global optimal solution of the generation particle.

Updates to the particle position are made using Equation 3, whereas updates to the particle velocity are made using Equation 2. The fitness function determines the personal best (P_best), which denotes the best value of each particle in the search space, followed by the global best (g_{bestd}), which denotes the best value of the particle across the board for the swarm. The loop is repeated until the best solution is found. Each particle will utilize the updated velocity to update its position after the initial update of the particle's velocity by (P_best) and (g_{bestd}). The iterations will continue till the maximum number has been reached. In order to create the optimal performance with the least amount of error, the plant uses the PID controller's obtained values. The PSO technique calls for clearly defined parameters for the swarm number, maximum iterations, and absolute permitted error. The PSO parameters should be carefully chosen in order to obtain rapid convergence. A great deal of expertise and understanding of system behavior are required for the skill of selecting the appropriate settings [21]. The objective function that used to improve the gait of the walking robot is:

$$objective\ function(OF) = \sum_{t=0}^{T} e^2(t) \tag{3}$$

Where T is the maximum number of iterations. To achieve the best control performance at nominal operating circumstances, the stochastic algorithm can be used to tune the gains of PID controllers. PSO is used to adjust the K_p, K_i, and K_d PID gains and parameters. The algorithm can be seen below.

The PSO algorithms initialization parameters used in this work can be seen in the table III shown below:

Table IV shows the best parameters found from the particle swarm optimization (PSO) algorithm.

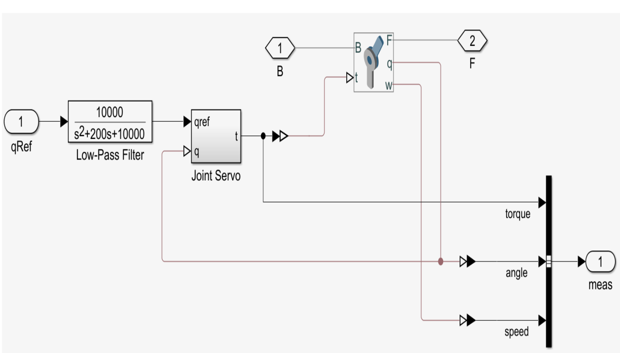


Fig. 5. joint subsystem of eight-legged robot

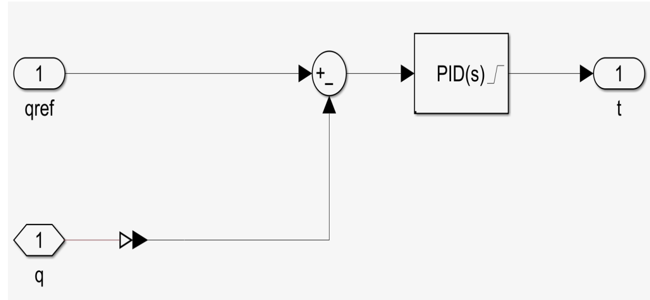


Fig. 6. PID controller system

TABLE III.
THE PSO ALGORITHMS INITIALIZATION PARAMETERS USED IN THIS WORK

Initialization parameter	PSO
Number of iterations	50
Size of populations	50
Inertia weight	0.8
Personal learning factor	2
Global learning factor	2
Number of variables	3
Lower bound of variables	0
Upper bound of variables	100

IV. SIMULATION RESULTS AND DISCUSSION

In this part, simulation results utilizing the Simscape analysis approach show how the system functions without the requirement for a challenging mathematical equation model of an eight-legged robot. This kind of analysis decreased the need for the physical system to be developed in real life and expensively tested with various control approaches. The proposed eight-legged robot model's performance and robustness under the influence of road disturbances are investigated through simulation simulations. There are two scenarios are examined: In the first, the robot moves in an open-loop system without

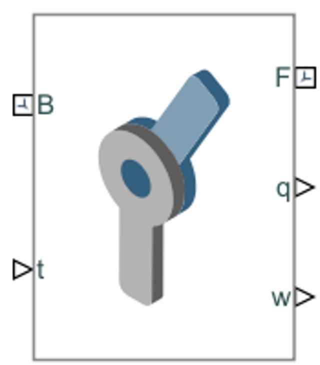


Fig. 7. Revolute joint

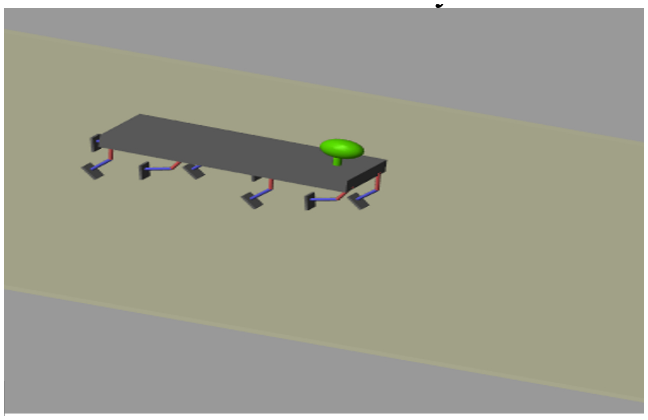


Fig. 8. The eight-legs robot with open-loop system controller

any controllers. The second system employs closed-loop PID controls and The third, increasing weights to assess the stability of the controller.

A. *Open-loop System*

In this scenario, there is no controller. On the road, the robot should be able to move vertically and complete the motion without falling. Sadly, the robot is immobile in this scenario because it lost equilibrium and dropped to the ground as soon as the simulation began. The robot falling is shown in the Fig. 8.

B. *Cosed-loop Using PID Controller*

In this case, we will test the robot with the PID controller present, but without putting weight on the robot. Fig. 11 shows the simulation results of an eight-legged robot without weight.

It is difficult to demonstrate the functionality of each joint because a walking robot has four legs, each with three links. As a result, just the front right leg's performance is shown in

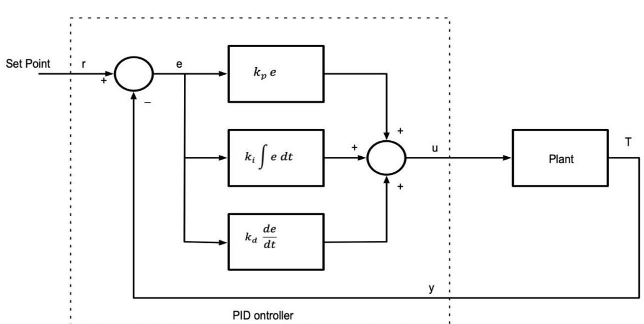


Fig. 9. The block diagram of the PID controller

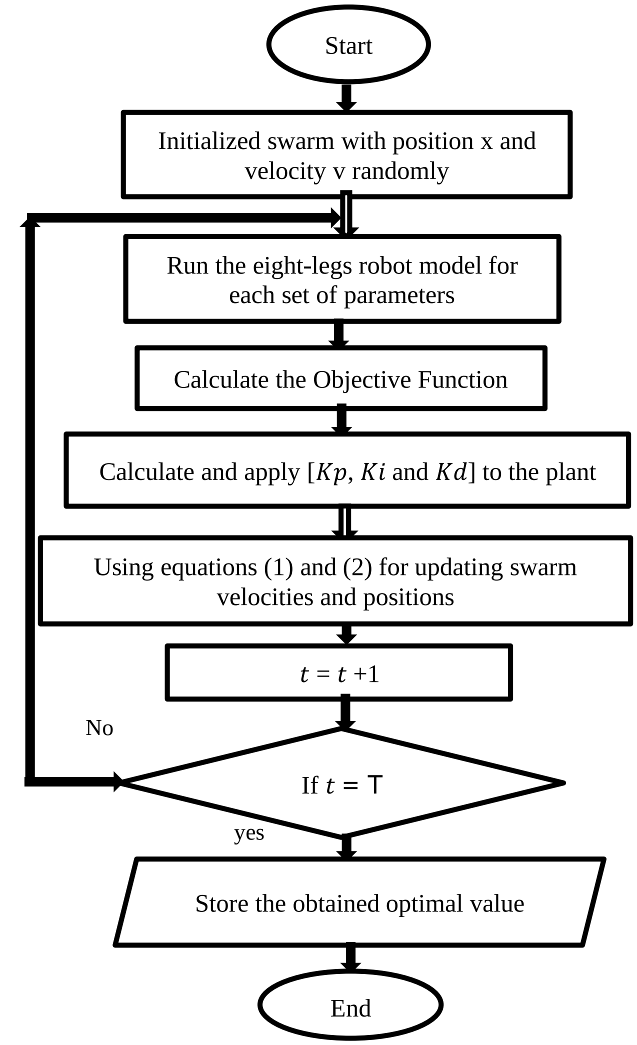


Fig. 10. Reference angle and actual angle comparison for ankle joint.

this piece. The Torques of each joint of front right leg of the robot are shown in Fig. 12.

Additionally, make a comparison between the reference angle with the actual angle. Fig. 13 , Fig. 14 and Fig. 15 depict the angle of ankle, the angle of knee and the angle of hip respectively.

C. *Robustness Test of Proposed PID Controller*

Two scenarios of disturbances are applied to the controlled system in order to investigate the robustness and efficacy of the designed controller. In the first scenario, the walking robot carries an extra kilogram of weight on its back (1kg) as shown in Fig. 16.

TABLE IV.
THE BEST PARAMETERS FOUND FROM THE PARTICLE SWARM OPTIMIZATION (PSO) ALGORITHM

PID	Kp	Ki	Kd
Controller 1	5.5269	21.8716	-0.06424
Controller 2	22.2173	90.1853	-0.26839
Controller 3	30.28177	158.32704	-0.57732
Controller 4	5.0143	14.7205	0.14882
Controller 5	47.28025	313.82798	-0.48928
Controller 6	28.7516	119.2822	0.4294
Controller 7	31.94132	162.0567	-0.3828
Controller 8	34.1137	168.7684	-0.2241
Controller 9	66.6964	19.65366	-0.2301
Controller 10	12.8901	165.2114	0.19982
Controller 11	22.7589	400.792	-0.03485
Controller 12	18.9221	350.6995	0.04146
Controller 13	11.48341	161.99661	0.06266
Controller 14	19.9210	251.9785	0.0304
Controller 15	25.2465	590.8900	-0.2182
Controller 16	45.7930	375.752	-0.6269
Controller 17	36.3697	189.1284	-0.15095
Controller 18	41.2373	233.0145	0.1947
Controller 19	39.61914	242.7642	-0.3423
Controller 20	17.45206	67.0916	-0.32989
Controller 21	12.39902	40.3531	-0.08427
Controller 22	26.5328	102.2719	0.5455
Controller 23	23.4975	220.2998	-0.0915
Controller 24	15.98097	159.2325	0.2113

Fig. 17 depicts the right front leg's PID controller efforts for the ankle, hip, and knee when 1 kg is applied. Fig. 18 , Fig. 19 and Fig. 20 show the comparison between the reference and actual angles for (hip,knee and ankle) of the front right leg leg when 1kg is added.

In the second scenario, the walking robot carries weight on its back (3kg).

Fig. 22 depicts the right front leg's PID controller efforts for the ankle, hip, and knee when 3 kg is applied.

Fig. 23 , Fig. 24 and Fig. 25show the comparison between the reference and actual angles for (hip,knee and ankle) of the front right leg leg when 3kg is added.

According to the case two results mentioned above, the robot can walk steadily and can reach the target without falling when it is carrying a 3-kg load on its torso and using a closed-loop control system with PID controllers.

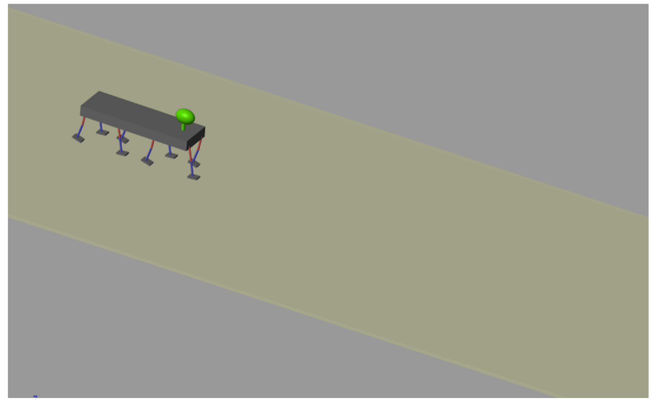


Fig. 11. Walking robot with no load at initial position

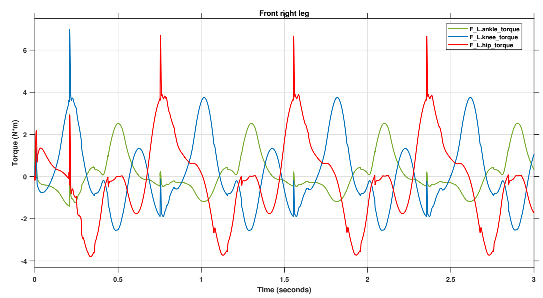


Fig. 12. Front right leg control efforts of the walking robot with no load

V. CONCLUSIONS

We used two ways to control the eight-legs robot. The first way open-loop system, as we observed that the robot falls at the beginning of the simulation. The second method The PSO algorithm is used to modify the PID controllers' best parameters so they can be used with the eight-legged walking robot. Twenty-four PID controllers are created for each of the twenty-four joints in the walking robot under study. The walking robot is implemented using the Simscape toolbox, and the proposed controllers are built using an optimization approach using the simulation toolbox. According to the simulation results, the suggested controllers successfully directed the walking robot to adopt the desired gaits. By placing additional weights on the back of the walking robot, the proposed controllers' resilience has been evaluated. There is an addition of two separate weights, 1 kg and 3 kg. The controlled system in both situations has been steady and has flown the desired gaits while rejecting disruption. As a result, the overall designed system shows that it can be used effectively and suitably to operate the walking robot. In this work, it was proven that the PID-PSO method is an effective and fast way to improve the controller and is better than other traditional

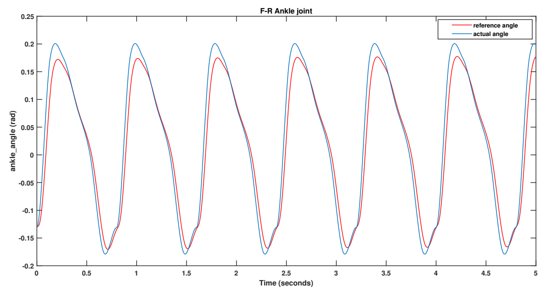


Fig. 13. Reference angle and actual angle comparison for ankle joint.

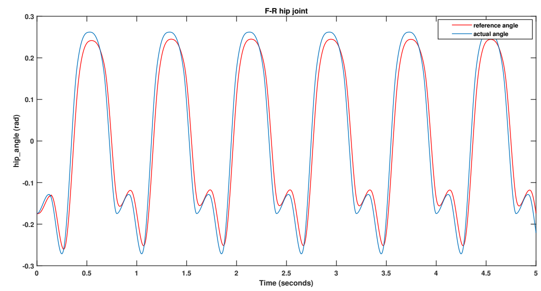


Fig. 15. Reference angle and actual angle comparison for ankle joint.

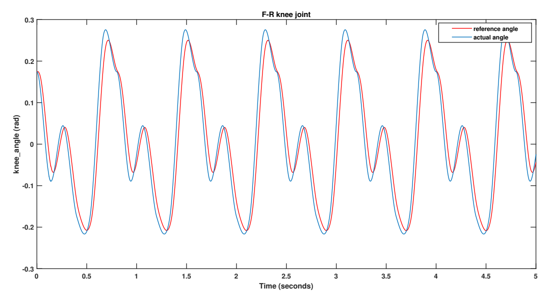


Fig. 14. Reference angle and actual angle comparison for knee joint.

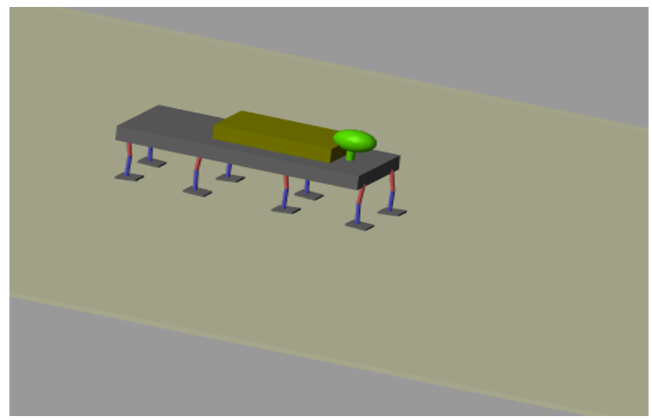


Fig. 16. visual display of eight legs walking robot in the Mechanics Explorer display with 1 Kg.

methods used in previous studies. Also, the robot became more stable when walking, walked a longer distance within a certain time, and carried larger weights.

Conflict of Interest

The authors have no conflict of relevant interest to this article.

References

[1] A. A. Aldair, A. Al-Mayyahi, and W. Wang, “Design of a stable an intelligent controller for a quadruped robot,” *Journal of Electrical Engineering & Technology*, vol. 15, no. 2, pp. 817–832, 2020.

[2] B. L. Youliang, “Control of an 8-legged, 24 dof, mechatronic robot,”

[3] A. Torres-Pardo, D. Pinto-Fernández, M. Garabini, F. Angelini, D. Rodriguez-Cianca, S. Massardi, J. Tornero, J. C. Moreno, and D. Torricelli, “Legged locomotion over irregular terrains: State of the art of human and robot performance,” *Bioinspiration & Biomimetics*, vol. 17, no. 6, p. 061002, 2022.

[4] A. A. Aldair, A. Al-Mayyahi, and B. H. Jasim, “Control of eight-leg walking robot using fuzzy technique based on simscape multibody toolbox,” in *IOP Conference Series: Materials Science and Engineering*, vol. 745, p. 012015, IOP Publishing, 2020.

[5] A. A. Aldair, A. Al-Mayyahi, and B. H. Jasim, “Control of eight-leg walking robot using fuzzy technique based on simscape multibody toolbox,” in *IOP Conference Series: Materials Science and Engineering*, vol. 745, p. 012015, IOP Publishing, 2020.

[6] J. Olaya, N. Pintor, O. F. Avilés, and J. Chaparro, “Analysis of 3 rps robotic platform motion in simscape and matlab gui environment,” *International Journal of Applied Engineering Research*, vol. 12, no. 8, pp. 1460–1468, 2017.

[7] B. N. AbdulSamed, A. A. Aldair, and A. Al-Mayyahi, “Robust trajectory tracking control and obstacles avoidance algorithm for quadrotor unmanned aerial vehicle,” *Journal of Electrical Engineering & Technology*, vol. 15, no. 2, pp. 855–868, 2020.

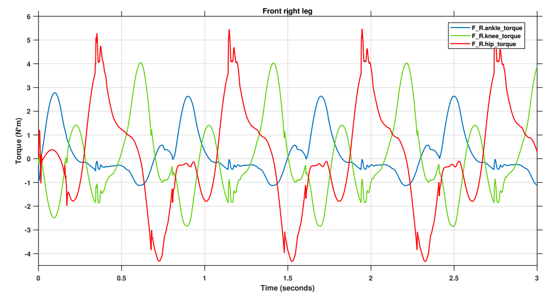


Fig. 17. PID controller effects for ankle, hip and knee when 1Kg is added.

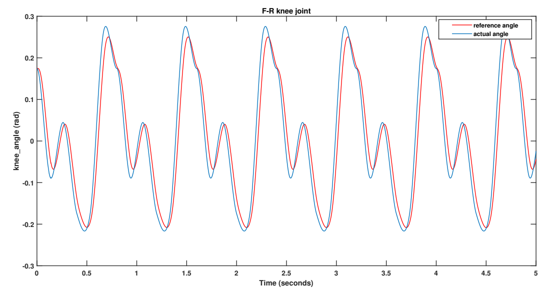


Fig. 19. Reference angle and actual angle comparison for knee joint.

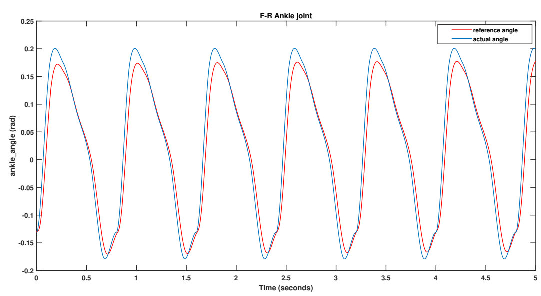


Fig. 18. Reference angle and actual angle comparison for ankle joint.

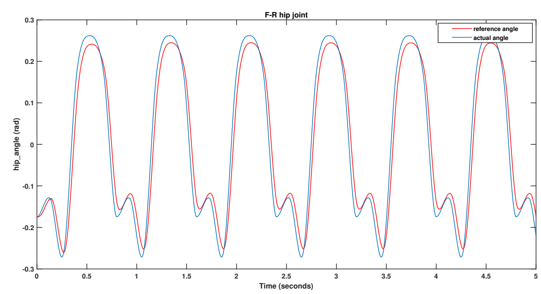


Fig. 20. Reference angle and actual angle comparison for hip joint.

[8] A. A. Aldair, A. Al-Mayyahi, and B. H. Jasim, "Control of eight-leg walking robot using fuzzy technique based on simscape multibody toolbox," in *IOP Conference Series: Materials Science and Engineering*, vol. 745, p. 012015, IOP Publishing, 2020.

[9] H. Liao, M. Noguchi, T. Maruyama, Y. Muragaki, E. Kobayashi, H. Iseki, and I. Sakuma, "Medical image analysis," *Science Direct, Tokyo, Japan*, 2013.

[10] C. C. Aranha, S. R. Carvalho, and L. M. GONCalvez, "Cambio: realistic three dimensional simulation of humanoids based on computer vision and robotics," in *Proceedings. XV Brazilian Symposium on Computer Graphics and Image Processing*, pp. 388–395, IEEE, 2002.

[11] N. A. M. Johari, H. Haron, and A. S. M. Jaya, *Robotic modeling and simulation of palletizer robot using Workspace5*. IEEE, 2007.

[12] T. Kubela, A. Pochyly, and V. Singule, "Advanced tools for multi-body simulation and design of control structures applied in robotic system development," *Solid state phenomena*, vol. 164, pp. 387–391, 2010.

[13] S. Mohapatra, R. Srivastava, and R. Khera, "Implementation of a two wheel self-balanced robot using matlab simscape multibody," in *2019 Second International Conference on Advanced Computational and Communication Paradigms (ICACCP)*, pp. 1–3, IEEE, 2019.

[14] G. K. McMillan, "Industrial applications of pid control," *PID control in the third millennium: Lessons learned and new approaches*, pp. 415–461, 2012.

[15] V. Kumar, B. Nakra, and A. Mittal, "A review on classical and fuzzy pid controllers," *International Journal of Intelligent Control and Systems*, vol. 16, no. 3, pp. 170–181, 2011.

[16] A. Issa and A. Aldair, "Learning the quadruped robot by reinforcement learning (rl).," *Iraqi Journal for Electrical & Electronic Engineering*, vol. 18, no. 2, 2022.

[17] R. S. Ali, A. A. Aldair, and A. K. Almousawi, "Design an optimal pid controller using artificial bee colony and genetic algorithm for autonomous mobile robot," *International Journal of Computer Applications*, vol. 100, no. 16, pp. 8–16, 2014.

[18] A. A. Aldair, A. Al-Mayyahi, and W. Wang, "Design of a stable an intelligent controller for a quadruped robot,"

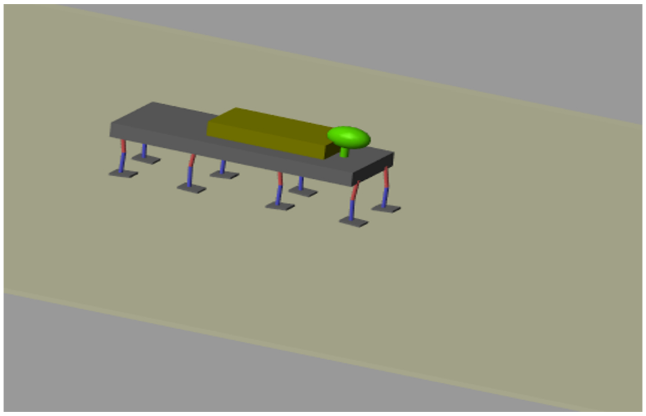


Fig. 21. Visual display of eight legs walking robot in the Mechanics Explorer display with 3 Kg.

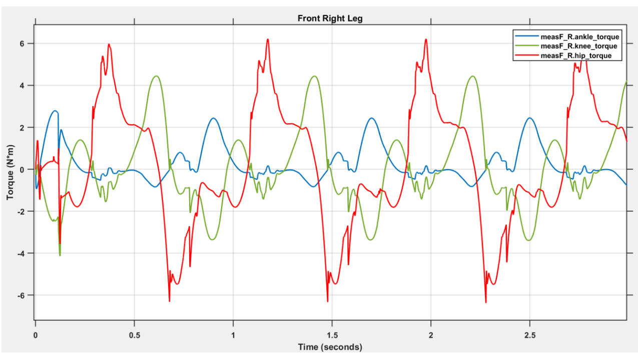


Fig. 22. PID controller effects for ankle, hip and knee when 3Kg is added.

Journal of Electrical Engineering & Technology, vol. 15, no. 2, pp. 817–832, 2020.

[19] D. Wang, D. Tan, and L. Liu, "Particle swarm optimization algorithm: an overview," *Soft computing*, vol. 22, no. 2, pp. 387–408, 2018.

[20] G.-R. Yu, Y.-K. Leu, and H.-T. Huang, "Pso-based fuzzy control of a self-balancing two-wheeled robot," in *2017 Joint 17th World Congress of International Fuzzy Systems Association and 9th International Conference on Soft Computing and Intelligent Systems (IFSA-SCIS)*, pp. 1–5, IEEE, 2017.

[21] I. K. Ibraheem and G. A. Ibraheem, "Motion control of an autonomous mobile robot using modified particle swarm optimization based fractional order pid controller," *Eng. Technol. J*, vol. 34, no. 13, pp. 2406–2419, 2016.

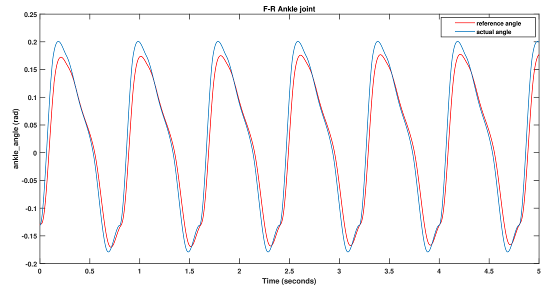


Fig. 23. Reference angle and actual angle comparison for ankle joint.

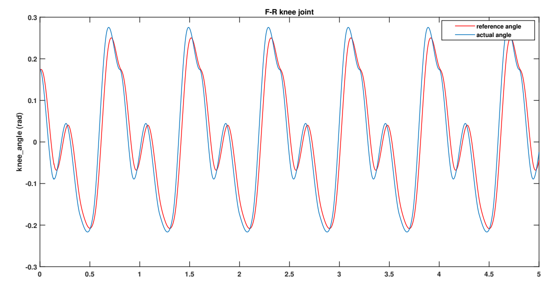


Fig. 24. Reference angle and actual angle comparison for knee joint.

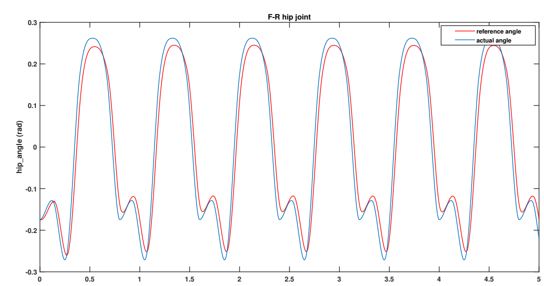


Fig. 25. Reference angle and actual angle comparison for hip joint.