

# Using Genetic Algorithm to Learn Gaits for an Eight-Legged Robot

Fatima H. Khayoun\*<sup>1</sup>, Ammar A. Aldair<sup>1</sup>, Ali A. Issa<sup>2</sup><sup>1</sup>Electrical Engineering Department, University of Basrah, Iraq<sup>2</sup>Medical Device Engineering, National University of Science and Technology, Dhi-Qar, Iraq

Correspondance

\*Fatima H. Khayoun

Electrical Engineering Department, University of Basrah, Iraq

Email: fatimaahameed98@gmail.com

## Abstract

Legged robots offer several benefits over standard wheeled vehicles when operating in tough and unstructured terrain. These benefits include increased speed, improved fuel efficiency, increased mobility, improved isolation from uneven terrain, and reduced environmental harm. This paper presents the modeling of an eight-legged robot that was simulated using Simscape Multibody toolbox in MATLAB, where the robot consists of eight legs, and each leg contains three links, and each link contains a PID controller, meaning it contains a total of 24 controllers. This controller was used to control the robot's gait and make it more stable. To obtain the optimal and most stable gait for the robot and to travel a longer distance, an optimization algorithm should be used, so that in this paper the genetic algorithm (GA) is used to obtain those points. To test the robustness of the proposed controllers, different weights are added (1 kg and 3 kg) as a load to the body of the legged robot, the obtained results show the efficiency of the proposed controllers.

## Keywords

**Eight-legged Robot; Simscape Multibody Toolbox; Optimization; Genetic Algorithm.**

## I. INTRODUCTION

Legged robotic systems are appealing alternatives to tracked/wheeled robots for applications in rough terrain and crowded situations. Their ability to choose contact places with the environment allows them to overcome hurdles comparable to their leg length. With such skills, legged robots may one day rescue people in jungles and mountains, climb stairs to carry payloads on construction sites, check uncontrolled underground passages, and explore distant planets. Legged systems have the capacity to do any physical activity that humans and animals are capable to archived [1]. Walking robots, such as bipeds, tripods, quadrupeds, hexapods, and octopods, have been extensively researched in the literature, with numerous instances of both design and technical solutions. In general, a legged robot is a complicated mechatronic system.

with several legs coupled to a main body (trunk) that serves as a support structure. When the robot legs are controlled autonomously, it can walk about its environment and ex-

ecute intended activities. As a result, multi-legged walking machines are an important class of robot in engineering applications. Because many animal species use legged mobility, such robots are inspired by the anatomy and physiology of various creatures [2]. Indeed, the natural world has long served as a source of inspiration and solutions for researchers studying legged robotics. Prior studies on biological walking machines have pinpointed specific evolutionary hotspots that led to the remarkable adaptability of natural walking mechanisms. For instance, the terrestrial crab's remarkable agility is mostly ascribed to its abdomen's natural reduction during evolution, which allowed for the possibility of lateral walking. The ability to move extremely quickly has been made possible by this naturally occurring improvement in the crab skeleton, which allows neighboring limbs to move freely without worrying about interfering with one another. The design and development of a walking robot may benefit from tuning and optimization techniques that can mimic this process of "im-



This is an open-access article under the terms of the Creative Commons Attribution License, which permits use, distribution, and reproduction in any medium, provided the original work is properly cited.  
©2026 The Authors.

Published by Iraqi Journal for Electrical and Electronic Engineering | College of Engineering, University of Basrah.

provement by natural evolution” as they may lead to solutions that would not have been achievable to arrive at manually. The use of genetic algorithms to enhance the mechanisms governing a legged robot’s stepping sequence will be covered in this paper. To do this, probabilistic search techniques based on natural selection principles and genetic algorithms (GAs) are applied. This paper demonstrates that a legged robot’s walking system can achieve favorable gait behavior by utilizing GA optimization. Through the analysis of these GA-derived solutions, optimal behavior patterns can be discovered, providing invaluable knowledge that can be utilized to either update the control mechanisms or enhance future robot designs. The use of GAs for this purpose is examined in this paper, with special attention to the situations where applying GAs results in walking motions that are optimally stable under normal circumstances and in the event that one leg becomes inoperable. The research was carried out using an eight-legged robot [3]. In 2016 the gaits that GA produced were tested on a simulated hexapod robot. Simulations have demonstrated that a hexapod’s developed gait allows it to move with one leg broken. This method will improve autonomous robots’ dependability and efficiency in human-hostile environments [4]. In 2018 a biped robot joint trajectory model was presented, in order to walk straight, utilizing polynomial equations for the joints in the support leg and Truncated Fourier (TFS) Series equations for the joints in the swing leg in the frontal and sagittal planes. Evolutionary techniques employing four unique genetic algorithms (GA-1 to GA-4) with varying crossover step implementations are utilized to optimize equation parameters and attain optimal walking motion speed and performance. The parent selection and crossover step components of these four GAs are different. We examine a sophisticated comparison feature between the best of two generations (parent and child) in GA-4 after a primary evaluation to ensure that the following generation is better off than the previous one. The GA-4 model has the best performance and reaches the necessary fitness value, according to tests of the algorithms conducted on the Darwin humanoid robot in the Webots simulated environment [5].

The authors suggested modeling a quadruped robot in 2019 using three moveable links for each leg. This will present the robot’s many advantages for easy and fluid movement. This design’s architecture has been implemented for each individual link and then integrated to create the entire system. The latter was tested without the use of a controller, and it was found that the robot was incapable of moving at all. With twelve controllers for each servomotor in each link, they consequently suggested two innovative controllers: PID and fuzzy controllers [6].

In 2019 the authors used a hybridized fuzzy-genetic con-

troller to conduct a navigational analysis of a humanoid robot in a challenging environment. The generated controller was created using a two-step hybridization basis, in which the fuzzy controller’s intermediate output was once more supplied as one of the inputs to the GA controller along with other standard inputs. The designed controller has been tested on a simulation platform, and the outcomes of the simulation have been confirmed in an experimental configuration set up in a lab. When the outcomes from the experimental and simulation platforms were compared, there was little to no discrepancy in the percentage of mistakes between both [7]. In 2020 the authors proposed an eight-legged robot with fuzzy controller. Since the walking robot under study has twenty-four joints, each joint has twenty-four fuzzy controllers built into it. The walking robot is implemented using the Simscape toolbox, and the suggested controllers with the optimization technique are built using the simulation toolbox. The simulation results collected demonstrate that the walking robot has been successfully steered to follow the intended gaits by the proposed controllers. The walking robot gets additional weights added to its back to evaluate the robustness of the suggested controllers. One kilogram and two kilograms are added as separate weights. In both cases, the controlled system remained stable and flowed smoothly despite disturbances [8]. In 2022, the authors suggested a quadruped robot constructed with SimScape-Multibody. The robot was tested without a controller, however the results indicate that it does not complete its gait to the final goal. They provide two learning methods: PID controller technique and reinforcement learning (RL). The robustness of the suggested controllers was assessed by adding additional weight to the quadruped robots. One kilogram and two kg are added, respectively. Despite rejection of the disturbance, the regulated system remained steady and operated correctly in every instance. Consequently, the entire system under consideration exhibits practicality and effectiveness in controlling a quadruped robot. Furthermore, there is a benefit to the control system with PID controllers compared to the control design using DDPG reinforcement learning [9].

The research in 2022 proposes a novel method for mobile robots’ smooth path planning that combines Bezier curves with genetic operators. Initially, the Bezier curve’s control points were identified by genetic operations. Then, the planning path was made smoother and more consistent by utilizing the Bezier curve’s smoothing properties, which decreased the amount of energy lost during robot movement. Secondly, in order to guarantee the robot’s safe and effective mobility, the safety distance was introduced to the fitness function and could be dynamically modified based on the distance between the path and the obstacle. Lastly, the simulation results demon-

strated that the suggested approach was successful in locating the ideal path, which was safer, smoother, and shorter than the paths found by conventional algorithms. The simulation time was extended, nevertheless. This occurred as a result of the enhanced algorithm's increased focus on solution quality during the path planning phase [10].

The authors in 2021 provides a novel chaotic map that is based on the degree of freedom of each joint of the robot and is used to pick the desired path coordination for each joint. Additionally, an ideal intelligent control system-based approach for a 3DOF robot is proposed for the rehabilitation of knee and hip joints. Artificial Neural Networks compute the angles and velocities of the knee and hip joints after the Genetic Algorithm optimizes them to take the intended path. The findings of the trial demonstrate that the method being described is capable of autonomously imitating the physiotherapist's actions at every stage and for every patient [11].

A 10-DoF humanoid robot's quasi-dynamic walking has been optimized using the GA in 2022. The lower half of a human being, which consists of two feet, two legs, two thighs, and one waist, is represented by the robot.

There are two knees, two ankles, and two hips among the joints. To meet the goal of improving the robot's walking performance in terms of high forward displacement and low deviation, a novel fitness function is developed. Another consideration is the robot's stability, or how it moves without tumbling over. The first one is acquired through forward kinematics, which involves visualizing the necessary joint synchronization to enable the robot to make forward step movements. The first person was the source of the other members of the original population. Only four GA cycles were carried out because of the humanoid robot's durability problem. There can be up to two progeny produced in each cycle. Only the six most physically fit people move on to the following cycle. Following the ultimate round, the population's highest fitness score rises by 111.48%, from 12.02 to 25.42 points. While the deviation angle decreases by 56.75% from 25.39° to 10.75°, the walking distance increases by 26.12% from 25.33 to 31.94 [12].

In 2023 The authors presented a MATLAB-simulated model of an eight-legged robot with an intelligent controller. 24 controls total—three for each leg of the structure—are included in the design to provide for flexibility in rotation and movement. In this study, a proportional integral derivative (PID) controller—three PIDs for each leg—has been used. Particle swarm optimization (PSO) was employed to modify the PID controller's parameters ( $K_p$ ,  $K_i$ , and  $K_d$ ). Simscape Multibody is used in the MATLAB software to create the structure of an eight-leg robot with a controller. This allows for the visualization and study of the robot's movement without the requirement for intricate analysis linked to a mathemati-

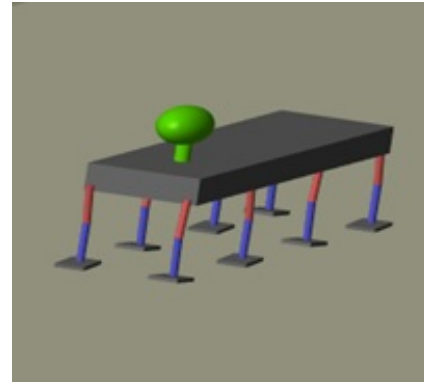


Fig. 1. Visual display of the eight-legged robot

cal model. Two scenarios were created using the simulation findings, which were carried out in a three-dimensional setting. The robot falls at the starting place in the first scenario, which involves implementing and simulating the robot without a controller. Better mobility was achieved in the second instance when PID controllers were incorporated into the system. Lastly, by varying the load that the robot carries, the resilience of the controller was confirmed [13].

## II. MODELING THE EIGHT-LEGGED ROBOT USING SIMSCAPE MULTIBODY TOOLBOX

The physical model of eight legs of robot is very complex, so that the obtaining the mathematical equation to simulate the structure of the robot is not an easy job. Although it is relatively simple to obtain an expression for the dynamics of a single leg of the eight-legged robot, the simulation becomes more complex due to the eight legs total of twenty-four degrees of freedom. For this reason, the MATLAB tool Simscape Multibody is used, which has the benefit of allowing us to run simulations with blocks that simulate joints and links (rotational or prismatic). Therefore, the simulate of the dynamic model can be performed without needing for obtaining mathematical equation, as each block's parameters are configured independently. These parameters include the robot's geometry, masses, center of mass, and inertia matrix. To accomplish that, we work in the following manner: we implement a leg first, and then we turn it into a block to make handling and replication for the robot's simulation easier. MATLAB provides three associated PID control blocks for each leg. Fig. 1 shown below indicates the visual display of the eight-legged robot in Simscape multibody in MATLAB [14].

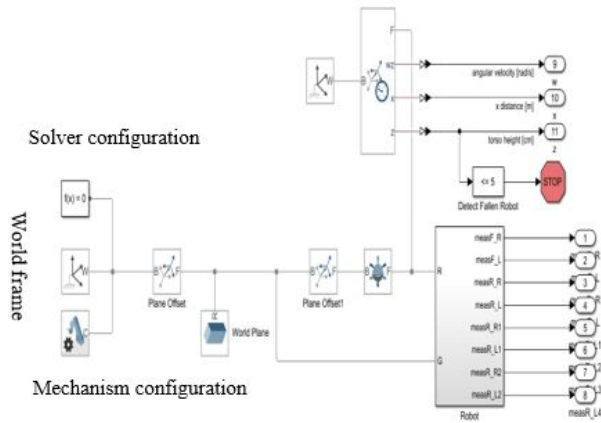


Fig. 2. The main structure of eight legs walking robot

Fig. 2 shows the main components of the eight-legged robot, which consists of a World frame, a Solver Configuration and a Mechanism Configuration linked to robot subsystem. The World frame represents the ground on which the robot walks.

Fig. 3 shows the content of the robot subsystem, which consists of eight legs that are connected to the torso and also contains the robot's neck and head, which are also connected to the torso.

In Fig. 4 the leg contains three joints (hip, knee and ankle) and each joint contains PID controller which shown in Fig. 5.

### III. CONTROLLER SCHEME

In this section, we suggest a global optimization approach for improving or obtaining the best eight-legged robot's gait. The reason for using a global optimization method for this model is that the walking problem is extremely nonlinear, which means that optimization will yield few local minimums so we use the genetic algorithm GA method to design the control

TABLE I.  
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

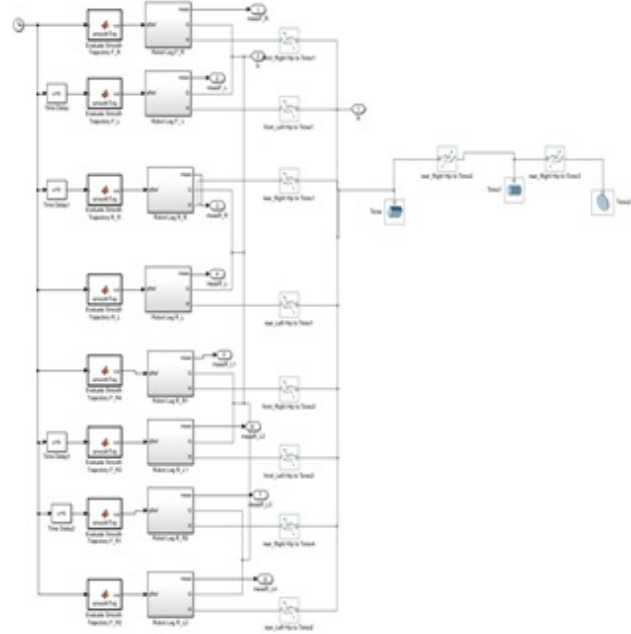


Fig. 3. Robot Subsystem

structure for the eight-legged robot.

#### A. Genetic Algorithm

The GA is a meta-heuristic that falls within the broad category of evolutionary algorithms in computational mathematics and informatics. It is driven by the evolutionary process. These algorithms, which concentrate on bio-inspired operators like selection, convergence, or mutations, are widely employed to produce excellent solutions to optimize and search problems [15]. In 1992, J.H. Holland proposed GA. Chromosome representation, fitness selection, and biologically inspired operators are the fundamental components of GA. Additionally, Holland presented a brand-new component called Inversion, which is frequently applied in GA deployments. The chromosomes typically accept data in binary string format. Every locus, or precise location on a chromosome, contains two potential alleles, or different variations of the same gene: 0 and 1. Considered as points in the solution space are chromosomes. Genetic operators are used to process them by repeatedly replenishing their population. Every chromosome in the population is given a value by means of the fitness function. The operators that draw inspiration from biology include selection, mutation, and crossover. Chromosomes are chosen for additional processing based on their fitness value throughout the selection phase. In crossover operator, the subsequences across chromosomes are altered by selecting a random locus to produce offspring. Certain chromosomal

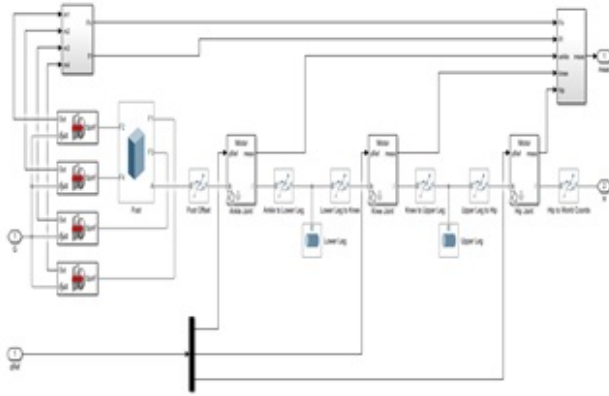


Fig. 4. one leg of eight legs robot (leg subsystem)

segments will be randomly flipped during a mutation based on likelihood [16]. The operations perform in GA are [17]:

**1) Selection (Encoding of a Chromosome)**

A chromosome will be organized to store data pertaining to the personification of the solution. Binary string formats are a commonly used encoding format. It is possible to plot each chromosome using a binary string. Every bit that the string accommodates is likewise accountable for carrying part of the solution's standards or elements.

**2) Crossover**

Crossover procedure can begin as soon as the chosen coding is verified to be utilized. A new offspring is created when a portion of the parent chromosomes' genes undergo crossover function. The most straightforward way to do this is to randomly choose a crossover point that is between the first parent point and this point.

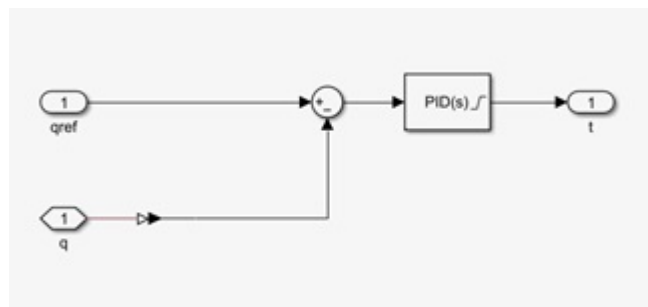


Fig. 5. PID controller

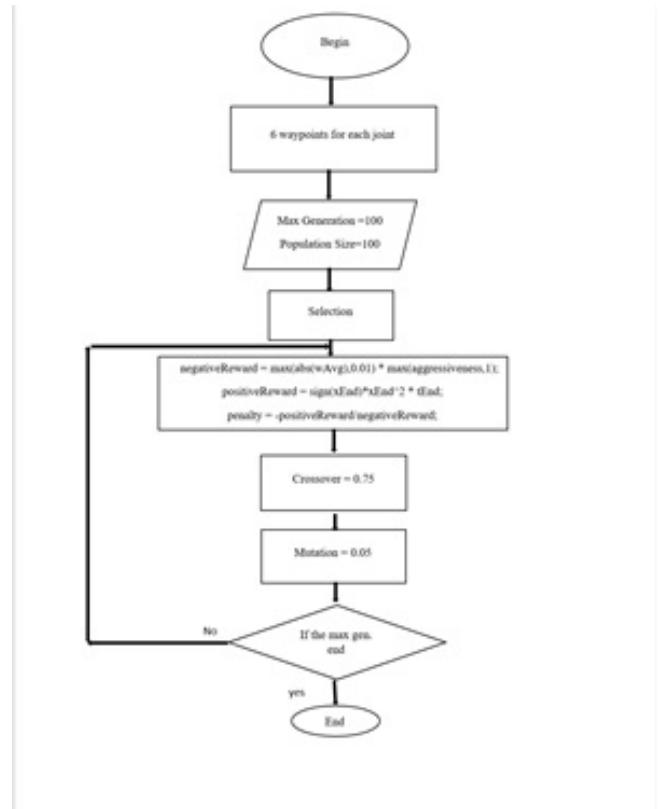


Fig. 6. Flowchart of genetic algorithm

**3) Mutation**

The following stage is called mutation after the crossing is completed. In order to prevent every solution in the population from falling into a local optimum of the solved problem, mutation is done on purpose. Crossover produces offspring that are randomly altered by mutation. We can randomly choose which bits in binary encoding to change from 0 to 1 or 1 to 0. This paper describes how an algorithm for controlling a walking robot's limb stepping sequences can be made better through the use of genetic algorithms, or genetically inspired learning strategies, as an evolutionary design tool. The paper provides simulated results and gives a specific scenario of determining the best walking gaits for an eight-legged robot. The genetic algorithm process can be summarized in flowchart shown in Fig. 6.

**B. Genetic algorithm inputs assumptions of eight-legs robot**

We made several assumptions in this part to allow us to control the model. Among these assumptions are the waypoints, as we assumed the waypoints are 6 that we must walk on. In addition, the left leg mimics the right leg accurately, but with a delay time of half a second. Since we have six waypoints

and three joints for each leg, then we have 18 optimization parameters (6 waypoints \* 3 joints). This means that each of the three leg joints (hip, knee, and ankle) has six waypoints that are repeated over a sequence. We also assume the maximum generation equals 100.

### C. Optimization output of an eight-legged robot

A cost function, which is separated into good and bad costs, is typically used to generate the outputs. It indicates how well the optimization problem works for a specific set of input parameters. In the walking robot system, two good costs (positive reward) are taken into account. The first is the end distance travelled ( $X_{end}$ ), which indicates how far the eight-legged robots can move in the allotted simulation period. The second positive cost is that the robot did not collapse ( $t_{end}$ ), allowing us to conduct the simulation for the full duration. The positive reward, or good costs, are represented by Eq. 1.

$$PositiveReward = X_{end} * t_{end} \quad (1)$$

Two negative costs (negative reward) are also expected. The first is the robot's average angular velocity ( $W_{avg}$ ), which indicates that a penalty will be imposed Eq. 2 can be used to express the bad cost (negative reward).

$$NegativeReward = W_{avg} * Aggress \quad (2)$$

Additionally, Eq. 3 can be used to express the final reward, or cost function.

$$Finalreward = \frac{- positiveReward}{negativeReward} \quad (3)$$

## IV. SIMULATION RESULTS

The simulation results of the genetic algorithm GA optimization method were shown in this section. The cost function for an eight-legged robot, as determined via the genetic algorithm GA optimization approach, is simulated and presented in Fig. 7. The GA results in Fig. 7 show that the cost function has a shape resembling a staircase.

The following Figures Fig. 8, Fig. 9 and Fig. 10 shown below show the results of the joints angles of the front right leg of the eight-legged robot from the optimization method (genetic algorithm).

In addition, Fig. 11 show the control efforts for the front right leg of the robot.

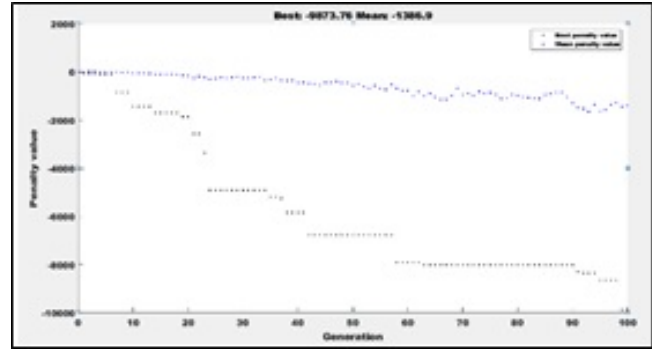


Fig. 7. Cost function result of GA of robot

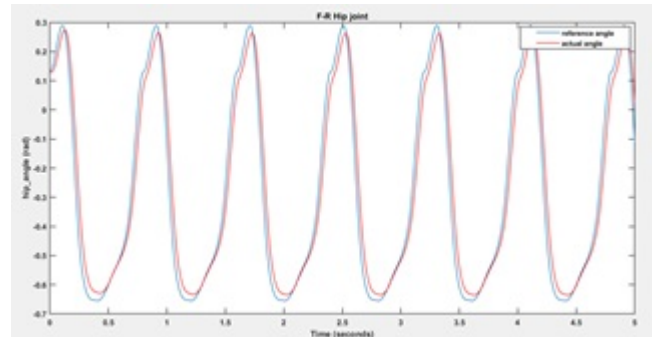


Fig. 8. Referene angle and actual angle comparison for hip joint.

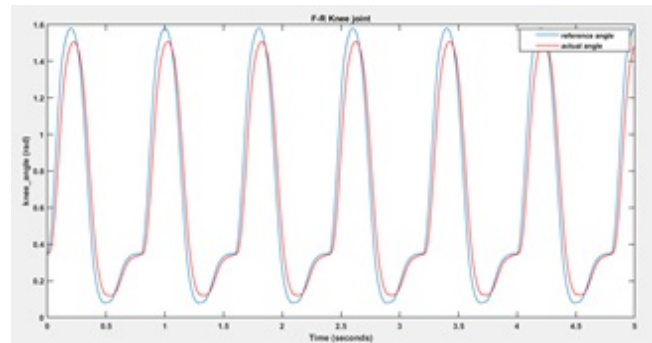


Fig. 9. Referene angle and actual angle comparison for knee joint.

To further test the stability and effectiveness of a proposed controller, two disturbance scenarios are added to the controlled system: the first scenario involves adding a 1-kg load to the robot, while the second scenario involves adding a 3-kg load.

In the first scenario, the walking robot's torso was given an additional 1 kg of weight.

The following Figures Fig. 12, Fig. 13 and Fig. 14 shown below show the results of the joints angles of the front right leg of the eight-legged robot from the optimization method

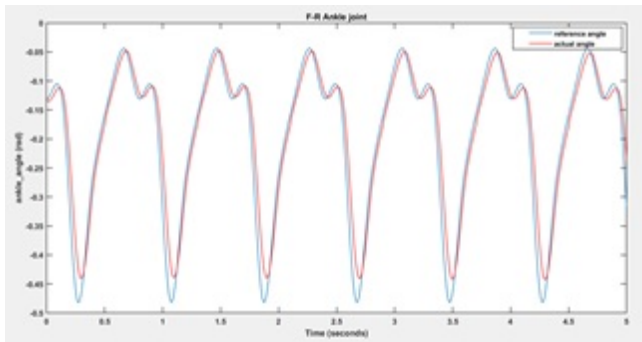


Fig. 10. Referene angle and actual angle comparison for ankle joint.

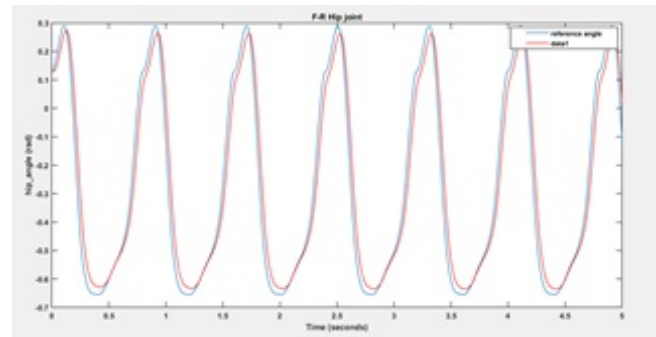


Fig. 12. Referene angle and actual angle comparison for hip joint after adding 1 kg.

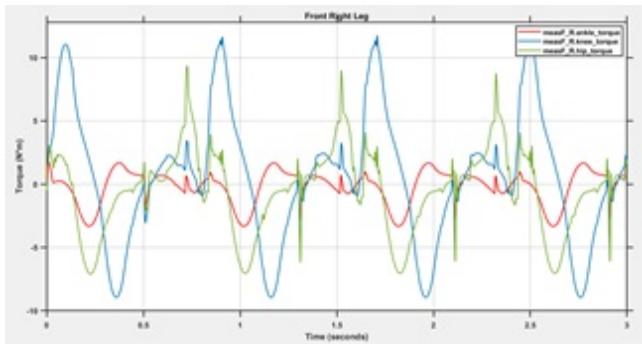


Fig. 11. The joints torque of the front right leg of an eight-legged robot without load.

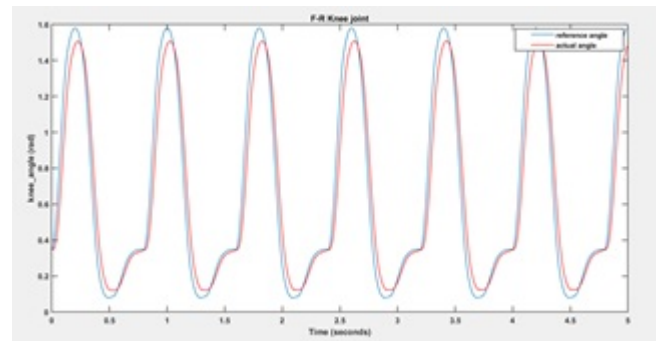


Fig. 13. Referene angle and actual angle comparison for knee joint after adding 1 kg.

(genetic algorithm) after adding load (1 kg).

In addition, Fig. 15 show the control efforts for the front right leg of the robot after adding 1 kg.

In the second scenario, the walking robot's torso was given an additional 3 kg of weight.

The following figures Fig. 16, Fig. 17 and Fig. 18 shown below show the results of the joints angles of the front right leg of the eight-legged robot from the optimization method (genetic algorithm) after adding load (3 kg).

In addition, Fig. 19 show the control efforts for the front right leg of the robot after adding 3 kg.

## V. CONCLUSION

In this paper, genetic algorithm was used to improve the eight-legged robot's gait and make it more stable and to cover a longer distance. The eight-legged robot was designed using Simscape Multibody toolbox in MATLAB. Then we tested the strength of the controller by loading the robot with a weight of 1 kg and 3 kg. The results showed that the genetic algorithm

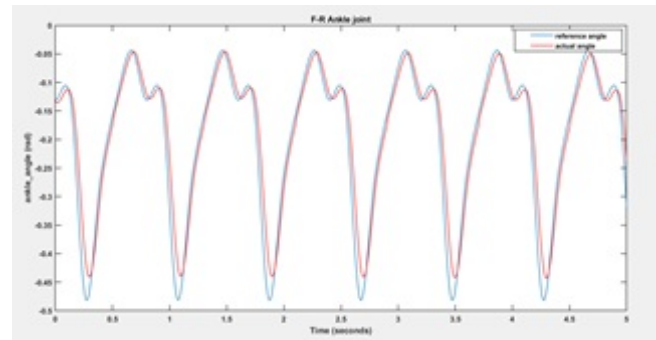


Fig. 14. Referene angle and actual angle comparison for hip joint after adding 1 kg.

can be used effectively and appropriately to control the robot's gait.

## CONFLICT OF INTEREST

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

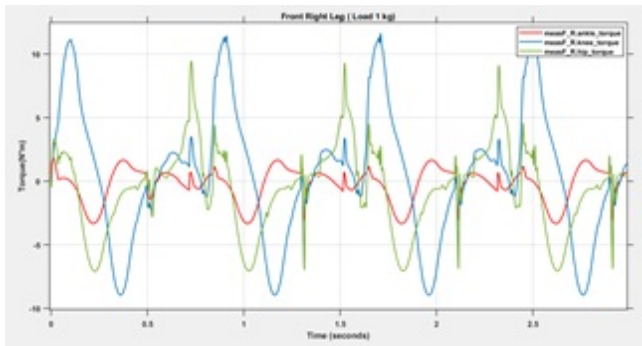


Fig. 15. The joints torque of the front right leg of an eight-legged robot after adding load 1kg.

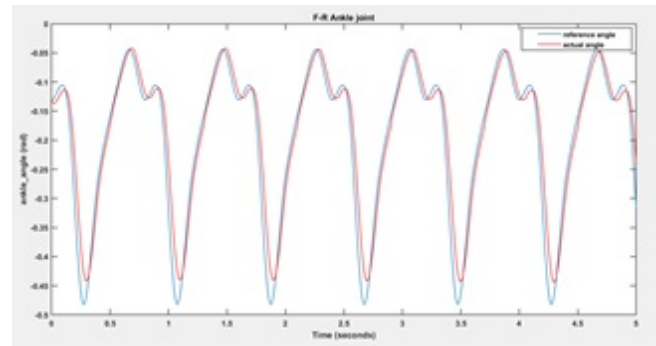


Fig. 18. Referene angle and actual angle comparison for ankle joint after adding 3 kg.

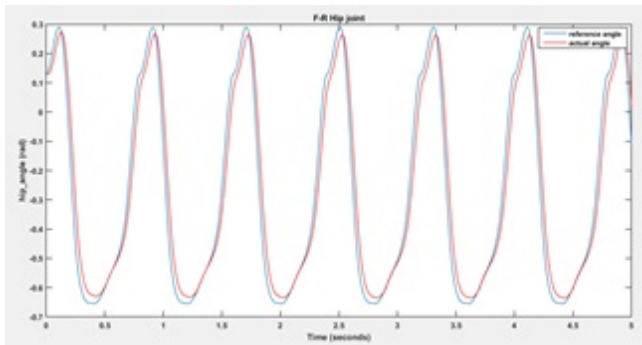


Fig. 16. Referene angle and actual angle comparison for hip joint after adding 3 kg.

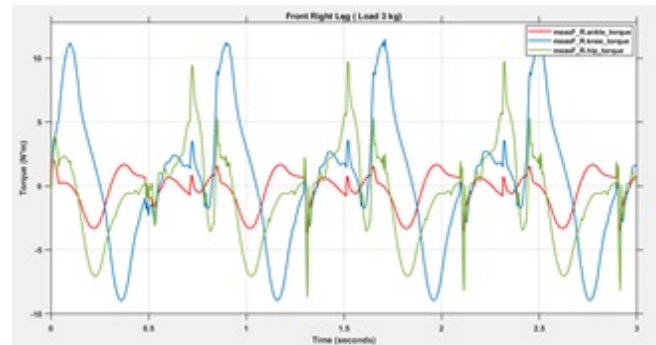


Fig. 19. The joints torque of the front right leg of an eight-legged robot after adding load 3kg.

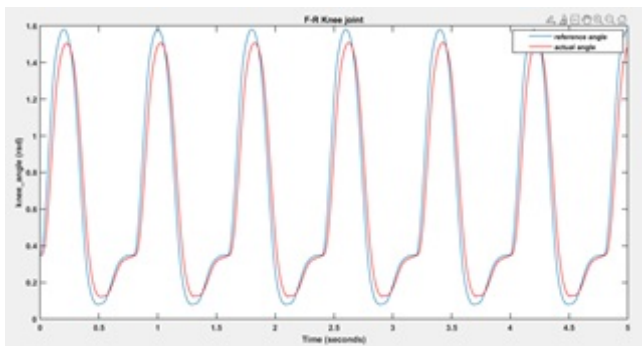


Fig. 17. Referene angle and actual angle comparison for knee joint after adding 3 kg.

## REFERENCES

- [1] J. Hwangbo, J. Lee, A. Dosovitskiy, D. Bellicoso, V. Tsounis, V. Koltun, and M. Hutter, "Learning agile and dynamic motor skills for legged robots," *Science Robotics*, vol. 4, no. 26, p. eaau5872, 2019.
- [2] D. Grzelczyk and J. Awrejcewicz, "Modeling and control of an eight-legged walking robot driven by different gait generators," *International Journal of Structural Stability and Dynamics*, vol. 19, no. 05, p. 1941009, 2019.
- [3] B. L. Luk, S. Galt, and S. Galt, "Using genetic algorithms to establish efficient walking gaits for an eight-legged robot," *International Journal of Systems Science*, vol. 32, no. 6, pp. 703–713, 2001.
- [4] A. Manglik, K. Gupta, and S. Bhanot, "Adaptive gait generation for hexapod robot using genetic algorithm," in *2016 IEEE 1st International Conference on Power Electronics, Intelligent Control and Energy Systems (ICPEICES)*, pp. 1–6, IEEE, 2016.
- [5] M. Salehi, M. Azarkaman, and M. Aghaabbasloo, "Optimized joint trajectory model with customized genetic algorithm for biped robot walk," *Journal of Computer & Robotics*, vol. 11, no. 1, pp. 21–29, 2018.
- [6] 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.

- [7] A. K. Rath, D. R. Parhi, H. C. Das, P. B. Kumar, M. K. Muni, and K. Salony, "Path optimization for navigation of a humanoid robot using hybridized fuzzy-genetic algorithm," *International journal of intelligent unmanned systems*, vol. 7, no. 3, pp. 112–119, 2019.
- [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] A. Issa and A. Aldair, "Learning the quadruped robot by reinforcement learning (rl)," *Iraqi Journal for Electrical & Electronic Engineering*, vol. 18, no. 2, 2022.
- [10] J. Ma, Y. Liu, S. Zang, and L. Wang, "Robot path planning based on genetic algorithm fused with continuous bezier optimization," *Computational intelligence and neuroscience*, vol. 2020, no. 1, p. 9813040, 2020.
- [11] W. A. Azar and P. S. Nazar, "An optimized and chaotic intelligent system for a 3dof rehabilitation robot for lower limbs based on neural network and genetic algorithm," *Biomedical Signal Processing and Control*, vol. 69, p. 102864, 2021.
- [12] E. Sitompul and M. Y. Baihaqi, "Quasi-dynamic walking optimization of humanoid robot using genetic algorithm," *Int. J. Artif. Intell.*, vol. 20, pp. 24–45, 2022.
- [13] F. H. Khayoun and A. A. Aldair, "Design of a stable an intelligent controller for an eight-legged robot," *Iraqi Journal for Electrical & Electronic Engineering*, 2023.
- [14] C. Urrea, L. Valenzuela, and J. Kern, "Design, simulation, and control of a hexapod robot in simscape multibody," *Applications from Engineering with MATLAB Concepts*, pp. 126–137, 2016.
- [15] T. Alam, S. Qamar, A. Dixit, and M. Benaida, "Genetic algorithm: Reviews, implementations, and applications," *arXiv preprint arXiv:2007.12673*, 2020.
- [16] S. Katoch, S. S. Chauhan, and V. Kumar, "A review on genetic algorithm: past, present, and future," *Multimedia tools and applications*, vol. 80, pp. 8091–8126, 2021.
- [17] A. Lambora, K. Gupta, and K. Chopra, "Genetic algorithm-a literature review," in *2019 international conference on machine learning, big data, cloud and parallel computing (COMITCon)*, pp. 380–384, IEEE, 2019.