autonomous-obstacle-avoidance-for-a-hexapod-robot-using-proximity-sensors

Article · June 2024

CITATIONS
0

READS
80

4 authors, including:

Ahmed Benyoucef
Université 20 août 1955-Skikda
3 PUBLICATIONS **2** CITATIONS
SEE PROFILE

Ali Amrane
Université 20 août 1955-Skikda
3 PUBLICATIONS **0** CITATIONS
SEE PROFILE

Zennir Youcef
Université 20 août 1955-Skikda
121 PUBLICATIONS **450** CITATIONS
SEE PROFILE

International Journal of Automation and Safety (IJAS)
Vol 02, N° 01, June, 2024, pp. 01-06

Journal homepage: https://www.asjp.cerist.dz/en/PresentationRevue/944

Autonomous Obstacle Avoidance for a Hexapod Robot Using Proximity Sensors

Ahmed Benyoucef[1*], Ali Amrane[2], Youcef Zennir[3] , Ammar Belatreche[4]

[1,3] Automatic Laboratory of skikda (LAS). University of Skikda, Skikda, Algeria
[2] LRPCSI Laboratory. University of Skikda, Skikda, Algeria
[4] Department of Computer and Information Sciences, Northumbria University, UK

Corresponding Author Email: a.benyoucef@univ-skikda.dz

Received: March 28th, 2024
Accepted: April 08th, 2024
Published: June 30th 2024

Keywords:
Hexapod, Control, Locomotion, Algorithm, Proximity sensor, CoppeliaSim, Simulation

ABSTRACT

The hexapod walking robot serves as a versatile mobile platform, adept at navigating challenging terrains owing to its stable leg-based locomotion. Demonstrating high stability in both static and dynamic states, it effectively traverses terrains with obstacles. In this paper, we propose a method for obstacle avoidance utilizing proximity sensors, integrated with an Intelligent Walking Algorithm (IWA) for locomotion control, and a tripod gait algorithm for walking. Our simulations were conducted using the CoppeliaSim simulator with the Python interface. The results were highly satisfactory, as the robot consistently avoided obstacles with a remarkable combination of stability and precision.

1. INTRODUCTION

The hexapod robot is esteemed within the realm of mobile robotics for its exceptional stability, manifesting itself admirably in both stationary and dynamic states. However, the intricacies involved in controlling the locomotion of this robot breed substantial complexity, primarily stemming from its elaborate mechanical design, featuring six legs, each endowed with two degrees of freedom or more. Researchers such as John Euler Chamorro Fuertes et al [1] and Joana Coelho et al [2] have immersed themselves in the realm of hexapod locomotion control, utilizing Touch-Pressure Sensors to navigate the intricate terrains. Conversely, Maged M. Abou Elyazed et al [3] have focused on refining walking gaits, particularly within hexapod robots like the Phantom II, to glide effortlessly along predefined paths. Additionally, Karlisa Priandana et al [4] and Xingji Duan [5] have delved into leg control methodologies, exploring the nuances of Simple Geometrical Tripod-Gait and Inverse Kinematics approaches.

Moreover, Mănoiu et al [6] have harnessed the inherent dynamics of the robot's legs to exert precise control over individual limbs of the hexapod, while A. Benyoucef and Y. Zennir [7] have employed Genetic Algorithms to sculpt adaptive locomotion strategies. Conversely, Qory Hidayati et al [8] have directed their efforts towards crafting an Intelligent Control System tailored for tasks such as Fire-Extinguishing and Obstacle Avoidance.

In the scope of our study, we pivot our focus towards the intricate orchestration of the robot's locomotion, leveraging a PID controller in tandem with our innovative Intelligent Walking Algorithm (IWA). This algorithm, fortified by the utilization of proximity sensors, steers the robot through a labyrinth of obstacles, finely regulating its locomotion. Our investigations were conducted through meticulously crafted simulations on the CoppeliaSim platform, culminating in promising and efficacious results. Witnessing the robot navigate through a myriad of tasks and challenges with unwavering success underscores the efficacy of our approach.

Our paper is organized as follows: Section 1 introduces the topic, while Section 2 elaborates on the structure of the hexapod robot and its kinematics model. Section 3 discusses the method of robot control and navigation. Section 4 presents simulation results and ensuing discussions. Finally, we conclude this paper in the last section.

2. HEXAPOD ROBOT

The hexapod robot's locomotion capabilities are remarkable due to its six legs, each boasting three degrees of freedom, totaling an impressive 18 degrees of freedom. With this level of flexibility, the robot can execute various movements seamlessly.

Firstly, it can walk forward and backward with ease, adjusting the positioning and movement of its legs to propel itself in the desired direction. This forward and backward motion is complemented by the ability to turn left and right, enabling the robot to navigate through complex environments with agility and precision.

Furthermore, the hexapod robot exhibits dynamic body movements that enhance its locomotion capabilities. It can perform yaw, pitch, and roll motions, allowing it to adapt to uneven terrain and maintain stability while in motion. These body movements not only contribute to the robot's overall stability but also facilitate efficient navigation through challenging landscapes.

In addition to the basic locomotion patterns, the hexapod robot possesses the capability to move laterally to the right and left without the need for turning. This lateral movement enhances its manoeuvrability, enabling it to sidestep obstacles or adjust its position when navigating tight spaces.

Overall, the hexapod robot's locomotion capabilities are a testament to its advanced design and engineering, allowing it to traverse diverse terrains and environments with remarkable agility and versatility.

A. *The leg of the robot:*

Each leg of the hexapod robot is a sophisticated mechanism designed for versatility and precision in movement. Comprising three degrees of freedom, the leg is capable of executing a wide range of motions to facilitate locomotion and interaction with the environment.

The three angles of joint movement within the leg are denoted as follows:

θ1 (Coxa Joint): This angle represents the movement of the coxa joint, which serves as the connection point between the leg and the body of the robot. The coxa joint allows the leg to pivot horizontally, enabling adjustments in the leg's orientation relative to the robot's body.

θ2 (Femur Joint): The femur joint governs the movement of the leg's femur segment. By adjusting the angle of θ2, the leg can extend or retract, facilitating forward and backward movements during walking or manoeuvring.

θ3 (Tibia Joint): The tibia joint controls the movement of the leg's tibia segment, which is responsible for vertical movement. By altering the angle of θ3, the leg can lift or lower, allowing the robot to navigate uneven terrain or adjust its height as needed.

These three angles work in concert to enable the leg to adapt to various environmental conditions and execute complex motions with precision. Whether traversing rough terrain, climbing obstacles, or interacting with objects, the leg's multifaceted design provides the hexapod robot with the agility and dexterity required for effective locomotion and manipulation tasks.

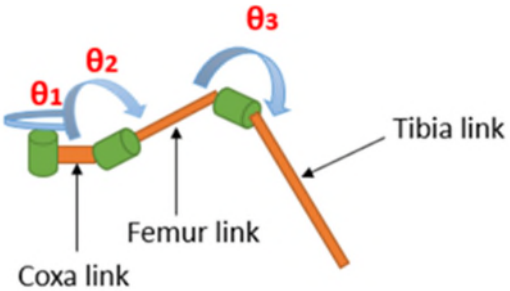

Figure. 1 Leg of Hexapod Robot

B. *Kinematics of hexapod robot:*

Forward kinematics refers to the process of determining the position and orientation of the end-effector of a robot given the joint angles. In the case of a hexapod robot leg, forward kinematics is used to calculate the Cartesian coordinates (x, y, z) of the end-effector, which is typically the tip of the leg, based on the angles of its joints.

To perform forward kinematics, Denavit-Hartenberg parameters are commonly utilized. These parameters define the geometric relationship between adjacent links in a robotic arm or leg. They include parameters such as link length, link twist, joint angle, and joint offset.

Table 1 presents the Denavit-Hartenberg parameters used for the forward kinematics of the hexapod robot leg. These parameters are crucial for establishing the transformation matrices between adjacent links, ultimately allowing us to compute the position and orientation of the end-effector in Cartesian space.

Once the Denavit-Hartenberg parameters are established, the forward kinematics equations can be derived using matrix transformations. These equations take into

account the joint angles of the leg and apply successive transformations to determine the position and orientation of the end-effector relative to a chosen reference frame.

By implementing forward kinematics, we can accurately determine the spatial coordinates of the end-effector of the hexapod robot leg, enabling precise control and coordination of its movements in various tasks and environments.

Table 1. Denavit-Hartenberg parameters

Links	αi	ai	di	θi
1	90	Lc	0	θ1
2	0	Lf	0	θ2
3	0	Lt	0	θ3

Where the description of the parameters is illustrated in Table2.

Table 2. Description of the parameters

Parameters	Description
α_i	the distance from Zi to Zi+1 measured along Xi
a_i	the angle from Zi to Zi+1 measured about Xi.
d_i	the distance from Xi-1 to Xi measured along Zi.
θ_i	the angle from Xi-1 to Xi measured about Zi.
θ1	Coxa angle
θ2	Femur angle
θ3	Tibia angle
Lc	Coxa Link length
Lf	Femur Link length
Lt	Tibia Link length

the transformation matrix of one link of the leg is as follows:

$$T_i^{i-1}(\theta_i, \alpha_i, d_i, a_i) = \begin{bmatrix} \cos\theta_i & -\sin\theta_i \cos\alpha_i & \sin\theta_i \cos\alpha_i & \alpha_i \cos\theta_i \\ \sin\theta_i & \cos\theta_i \cos\alpha_i & -\cos\theta_i \sin\alpha_i & \alpha_i \sin\theta_i \\ 0 & \sin\theta_i & \cos\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (1)$$

The transformation matrix is the multiplication of three matrices of transformations:

$$T_3^0 = T_1^0 . T_2^1 . T_3^2 \quad (2)$$

The leg tip's Cartesian position and orientation are established through forward kinematics in this study, employing the Denavit-Hartenberg method [9], [10], as illustrated in equation 3.

$$\begin{cases} x = \cos\theta_1(L_c + L_f \cos\theta_2 + L_t \cos(\theta_2 - \theta_3)) \\ y = \sin\theta_1(L_c + L_f \cos\theta_2 + L_t \cos(\theta_2 - \theta_3)) \\ z = L_f \sin\theta_2 + L_t \sin(\theta_2 - \theta_3)) \end{cases} \quad (3)$$

C. *Inverse kinematics :*

Inverse kinematics is employed to determine the angles of the three links within the leg, as it allows us to infer the joint angles necessary to achieve a desired position of the end-effector. In the case of the hexapod robot, each leg consists of multiple links and spinning joints, making inverse kinematics crucial for controlling its movements effectively. To solve the inverse kinematics problem [11], various constraints need to be considered. These constraints may include factors such as the desired position of the end-effector, the limitations of the robot's mechanical structure, and any environmental obstacles that must be avoided during movement. By incorporating these constraints into the inverse kinematics calculations, we can accurately determine the joint angles required to achieve the desired leg position and movement.

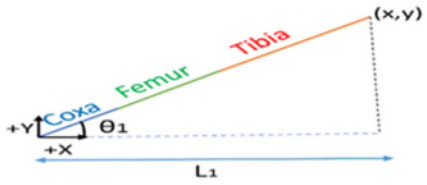

Figure. 2 Triangle to obtain coxa angle (θ1)

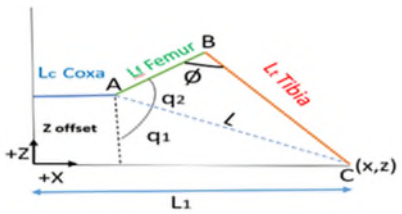

Figure. 3 Triangles to obtain Femur and Tibia angles (θ2, θ3)

From Figure 1, we can determine the angle θ1, which represents the motion of the coxa joint. Subsequently, from Figure 2, we derive equations (4) through (6) to calculate the angles θ2 and θ3, corresponding to the movement of the femur and tibia joints, respectively. These equations are presented as follows:

$$\theta_1 = \tan^{-1}\left(\frac{x}{y}\right) \quad (4)$$

$$\theta_2 = q_1 + q_2 \quad (5)$$

$$\theta_3 = \emptyset - 180 \quad (6)$$

Where:

$$q_1 = \tan^{-1}\left(\frac{Z_{offset}}{L_1 - L_c}\right) \quad (7)$$

$$q_1 = \cos^{-1}\left(\frac{\sqrt{L_f^2 + L^2 - L_t^2}}{2L_f - L}\right) \quad (8)$$

$$L = \sqrt{{Z_{offset}}^2 + (L_1 - L_c)^2} \quad (9)$$

$$\emptyset = \cos^{-1}\frac{\sqrt{L_f^2 + L_t^2 + L^2}}{2L_f L_t} \quad (10)$$

3. CONTROLLING AND NAVIGATION

A. *Controlling robot locomotion:*

We controlled the hexapod robot to walk forward with a constant linear velocity (VR(t)), navigating around obstacles in its path using our walking algorithm equipped with proximity sensors. We obtained the Euler angles of the robot's leg joints through its inverse kinematics (K-1 (Θ)), enabling us to implement a PID controller on each joint [12], Claudio Urrea et al [13] used classic optimization and A.Benyoucef et Y.zennir [7] used heiristic method of optimization . This ensured consistent trajectory tracking for each leg's end-effector, whether the robot was moving forward or turning. The tuning process for the PID controller is illustrated in Figure 4.

Where Xd and Yd represent the desired trajectory of each leg, and Θijd (for i = 1, 2,..., 6) denotes the desired Euler angles for the ith leg and jth joint (where j = 1, 2, 3 representing the coxa, femur, and tibia joints, respectively). ε(t) denotes the error in the position of the Euler angle, as presented in Equations 11 and 12. PID (Lij) serves as the controller for each leg joint.

$$fij = \int |\varepsilon_{i,j}(t)| dt \quad (11)$$

$$\varepsilon ij(t) = |\theta ijd(t) - \theta ij(t)| \quad (12)$$

For optimal robot walking, we must minimize the function f.

B. *Intelligent Walking Algorithm (IWA):*

The navigation of the robot relies on proximity sensors for obstacle avoidance and object detection [14], as well as for detecting tilt, calculating distance, and sensing contact with objects [15]. Sensors play a crucial role in enabling robots to navigate various terrains effectively. Therefore, in our IWA algorithm, the robot moves forward while employing an obstacle avoidance strategy based on proximity sensors to measure the distance between the robot and obstacles, regardless of their position (front, right, or left). The flowchart of the Intelligent Walking Algorithm, depicted in Figure 5, illustrates how the robot navigates around obstacles using proximity sensors.

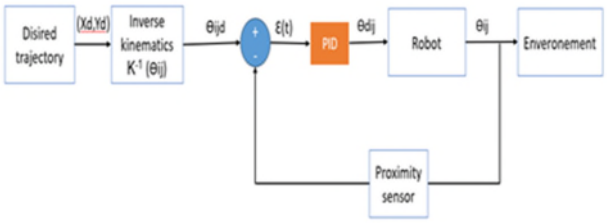

Figure. 4 Tuning loop of the legs angles (θij)

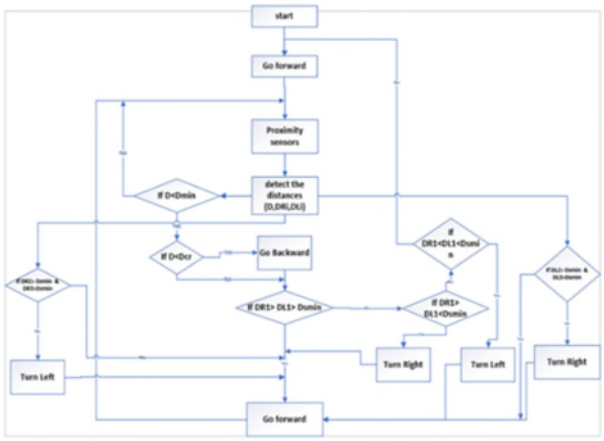

Figure. 5 flowchart of Intelligent Walking Algorithm (IWA)

"D" represents the distance between the robot and the obstacle, while "Dmin" denotes the minimum acceptable distance between them. In the event of a front obstacle, where "D" falls below "Dmin", the robot halts and assesses the best course of action to avoid the obstacle. Initially, it calculates the distances between the left and right sides of the robot concerning the nearest obstacle. If one side is closer, the robot steers towards the opposite direction. In cases where there are no body obstacles on either side, the robot can turn left or right according to our desired trajectory. However, if the distance "D" is less than the critical distance, "Dcr", the robot stops and reverses its direction to enable turning left or right, as collision with the obstacle impedes turning. Moreover, when an obstacle is behind the robot, it may not be detected by front sensors. In such instances, the middle and rear sensors detect it, and distances are calculated on both sides (DR1, DR2, DR3 for the right side and DL1, DL2, DL3 for the left side) to compare them with the minimum side distance, Dsmin.

4. SIMULATION RESULTS AND DISCUSSION

Our simulation was conducted using the CoppeliaSim simulator with a Python interface. We utilized the PyQt5 library to create a command interface. Through this Python interface, we could initiate the walking of the robot and display the distances between the robot and obstacles, as illustrated in Figures 6 and 7.

Our robot is equipped with seven proximity sensors: one in the front for detecting front obstacles, and three on each side distributed across each leg on the body frame for detecting side obstacles, as depicted in Figures 6 and 8.

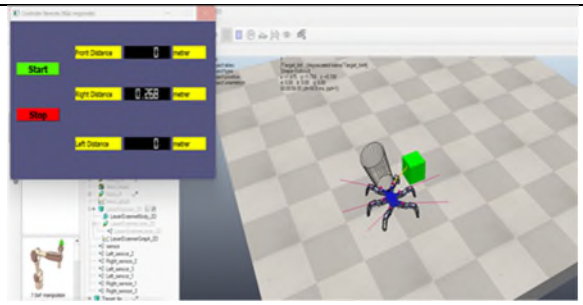

Figure. 6 Python interface with simulation environment.

Figure. 7 python interface board

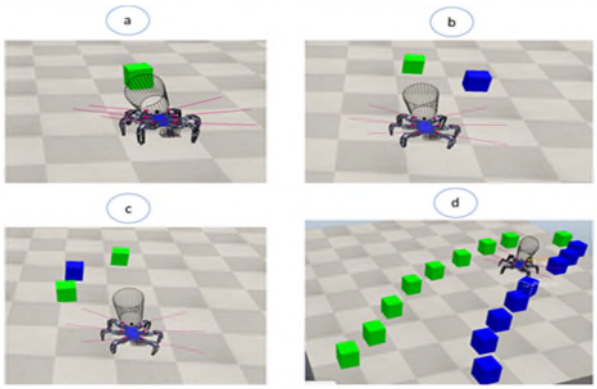

Figure. 8 3D simulation of the hexapod robot with obstacles (a): with front obstacle, (b) with front and right obstacles, (c) with front and left obstacles, (d) with multi-obstacles in various sides.

Figure 8 is a description of the 3D simulation scenarios of the hexapod robot with obstacles:

(a) Simulation with a front obstacle: The hexapod robot encounters a single obstacle directly in front of it. It navigates its path while avoiding collision with this obstacle.

(b) Simulation with front and right obstacles: In this scenario, the hexapod robot faces obstacles both in front and to its right side. It must maneuver to avoid collision with both obstacles simultaneously.

(c) Simulation with front and left obstacles: Similar to scenario (b), but the obstacles are located in front and to the left side of the hexapod robot. It adjusts its trajectory to bypass these obstacles safely.

(d) Simulation with multi-obstacles in various sides: This scenario presents the hexapod robot dealing with multiple obstacles distributed across various sides, including the front, right, and left sides. The robot must

navigate through these obstacles efficiently while maintaining its course.

In Figure 9, the trajectories followed by the robot when it detects obstacles are depicted. The blue curve represents the body trajectory (center of gravity) of the robot, while the asterisks (*) indicate the center of mass of obstacles (the box).

In accordance with the IWA principles, when the robot detects an obstacle on one side, it will turn to the other side if there is no obstacle present, as shown in Figure 9 (b and c). In the scenario where the robot detects only a front obstacle, leaving two empty sides (without obstacles), we programmed it to turn left to maintain our trajectory, as depicted in Figure 9 (a). Figure 9(d) illustrates a challenge where the robot encounters multiple obstacles, which could be located in the front, right, left, or all sides simultaneously.

In summary a description of the trajectories of the robot when avoiding obstacles in the specified scenarios:

(a) Trajectory with a front obstacle: The robot adjusts its trajectory to circumvent the obstacle in front of it while continuing its forward movement. It may veer slightly to the left or right, depending on the available space to navigate around the obstacle.

(b) Trajectory with front and right obstacles: In this scenario, the robot maneuvers to avoid collisions with both the obstacle in front of it and the obstacle to its right side. It may execute a combination of forward, backward, and lateral movements to bypass the obstacles safely.

(c) Trajectory with front and left obstacles: Similar to scenario (b), but the robot avoids obstacles located in front and to its left side. It adjusts its trajectory accordingly to avoid collisions with these obstacles while maintaining its intended direction of movement.

(d) Trajectory with multi-obstacles in various sides: In this complex scenario, the robot dynamically navigates through multiple obstacles distributed across different sides. It employs a combination of movements, including turning, side-stepping, and adjusting its elevation, to evade obstacles and continue its path towards the target destination.

The positions and velocities of the joint angles are depicted in Figures 10 and 11, respectively. Figure 10 illustrates the positions of the joint angles (coxa, femur, and tibia) of a single leg when the robot is walking forward or backward and when it is turning left or right. The blue curve represents the femur angle position, the red curve represents the coxa angle position, and the green curve represents the tibia angle position. Correspondingly, in Figure 10, the velocities of these angles are depicted using the same color scheme.

In Figure 12, we utilized a background color because there are multiple different colors present, and if we were to use a white background, some colors may not be clearly visible.

For reference in figure 12, the legs are named from the right side of the robot to the left side as f1 to f6. Each foot is associated with a specific color curve, as follows:

Red: Foot 1
Blue: Foot 2
Yellow: Foot 3
Green: Foot 4
Pink: Foot 5
White: Foot 6

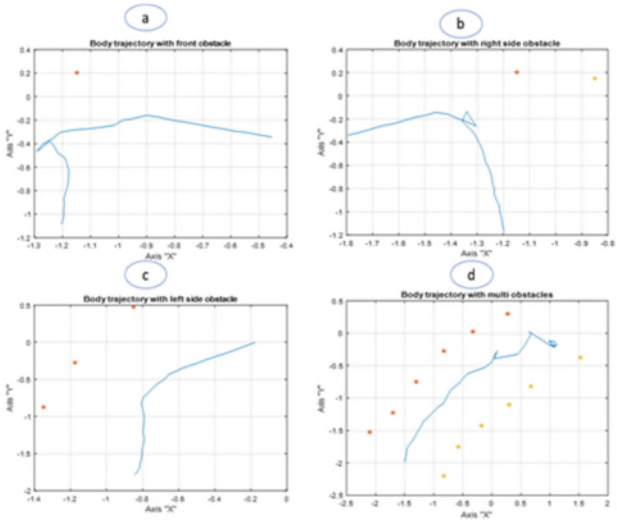

Figure. 9 The Trajectory of the Robot When Avoiding Obstacles, (a) with front obstacle, (b) with front and right obstacles, (c) with front and left obstacles, (d) with multi-obstacles in various sides.

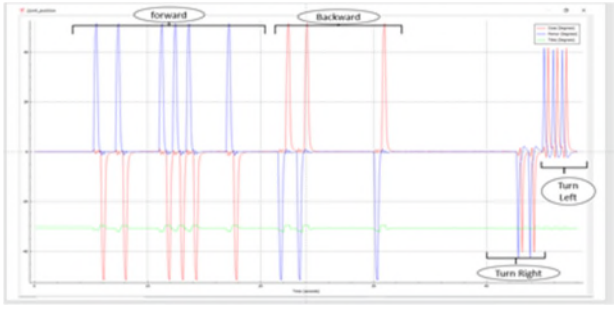

Figure. 10 Joint Angles Position of One Leg with Various Locomotions of the Robot

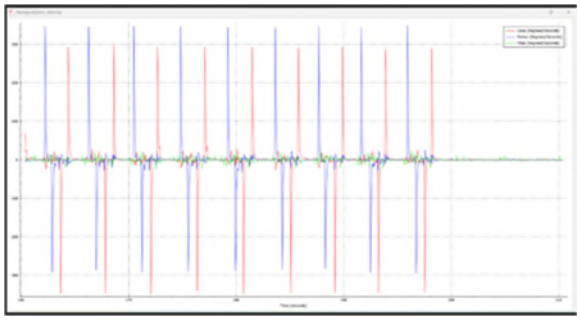

Figure. 11 Joint Velocities of a Single Leg During Multi-Obstacle Avoidance by the Robot

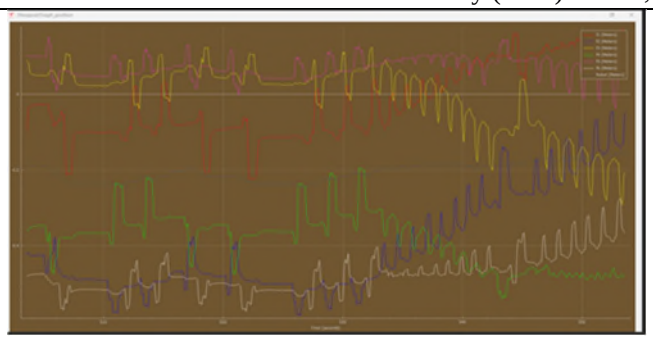

Figure. 12 Position of End-Effector of the Six Legs of the Robot on the X-Axis

Discussion: The results of the simulation were highly effective, as the robot demonstrated a robust response to our implemented walking algorithm (IWA). It successfully completed all assigned tasks and adeptly avoided obstacles placed at various positions relative to the robot. Notably, the robot maintained high stability and exhibited precise movements, as depicted in Figure 9. Additionally, the joint angles closely followed their desired trajectories, as illustrated in Figure 8. These findings underscore the efficacy of our algorithm, particularly in conjunction with proximity sensors, for navigation and obstacle avoidance in hexapod robots.

5. CONCLUSION

In our research, we focused on controlling the locomotion of the robot using PID controllers for each angle and implementing a navigation system for obstacle avoidance. Leveraging a specific walking algorithm driven by proximity sensors yielded promising results. However, challenges arose in confined spaces with obstacles positioned simultaneously in front, right, and left of the robot, leading to prolonged obstacle avoidance times. Notably, we did not explore obstacle avoidance while the robot walked backward, a consideration for future investigation. Moving forward, our future work will address this challenge and incorporate additional sensors alongside artificial intelligence techniques to enhance navigation in uneven terrains.

REFERENCES

[1] J. E. C. Fuertes, J. J. M. Arciniegas, P. B. G. De Castro, and O. A. V. Alban, 'Locomotion Control of PhantomX Hexapod Robot with Touch-Pressure Sensor and RoboComp', in 2020 IEEE Colombian Conference on Applications of Computational Intelligence (IEEE ColCACI 2020), Cali, Colombia: IEEE, Aug. 2020, pp. 1–6. doi: 10.1109/ColCACI50549.2020.9247874.

[2] J. Coelho, R. Sá, T. Ribeiro, and F. Ribeiro, 'Study of the locomotion of a hexapod using CoppeliaSim and ROS', International Conference on Computers and Automation (CompAuto), 007-09 Sep2021, Paris,France, p.6

[3] M. M. A. Elyazed and A. Y. AbdelHamid, 'ADAPTIVE WALKING GAIT FOR HEXAPOD PHANTOM_|| ROBOT TO FOLLOW A SMOOTHED PRE-DEFINED PATH', vol. 7, no. 6, 2019.

[4] K. Priandana, A. Buono, and Wulandari, 'Hexapod leg coordination using simple geometrical tripod-gait and inverse kinematics approach', in 2017 International Conference on Advanced Computer Science and Information Systems (ICACSIS), Bali: IEEE, Oct. 2017, pp. 35–40. doi: 10.1109 / ICACSIS. 2017.8355009.

[5] 'Tripod Gaits Planning and Kinematics Analysis of a Hexapod Robot.pdf'.

[6] M. – O. Sorin and M. Niţulescu, 'Hexapod Robot Leg Dynamic Simulation and Experimental Control using Matlab', IFAC Proc. Vol., vol. 45, no. 6, pp. 895–899, May 2012, doi: 10.3182 / 20120523-3-RO-2023. 00335.

[7] A. Benyoucef and Y. Zennir, 'Enhancing Hexapod Robot Locomotion Control Through PID Controller Optimization Using Genetic Algorithm', in 2023 IEEE 11th International Conference on Systems and Control (ICSC), Sousse, Tunisia: IEEE, Dec. 2023, pp. 556–561. doi: 10.1109/ICSC58660.2023.10449774.

[8] Q. Hidayati, F. Z. Rachman, and N. Yanti, 'Intelligent Control System of Fire-Extinguishing and Obstacle-Avoiding Hexapod Robot', Kinet. Game Technol. Inf. Syst. Comput. Netw. Comput. Electron. Control, pp. 1–10, Oct. 2017, doi: 10.22219/kinetik.v3i1.470.

[9] M. Silva, R. Barbosa, and T. Castro, 'Multi-legged Walking Robot Modelling in MATLAB/Simmechanics TM and Its Simulation', in 2013 8th EUROSIM Congress on Modelling and Simulation, Cardiff, United Kingdom: IEEE, Sep. 2013, pp. 226–231. doi: 10.1109/EUROSIM.2013.50.

[10] P. Ramdya et al., 'Climbing favours the tripod gait over alternative faster insect gaits', Nat. Commun., vol. 8, no. 1, pp. 1–11, Feb. 2017, doi: 10.1038 / ncomms 14494.

[11] S. I. Beaber, A. S. Zaghloul, M. A. Kamel, and W. M. Hussein, 'Dynamic Modeling and Control of the Hexapod Robot Using Matlab SimMechanics', in Volume 4A: Dynamics, Vibration, and Control, Pittsburgh, Pennsylvania, USA: American Society of Mechanical Engineers, Nov. 2018, p. V04AT06A036. doi: 10.1115/IMECE2018-88226.

[12] M. – O. Sorin and M. Niţulescu, 'Hexapod Robot Leg Dynamic Simulation and Experimental Control using Matlab', IFAC Proc. Vol., vol. 45, no. 6, pp. 895–899, May 2012, doi: 10.3182/20120523-3-RO-2023.00335.

[13] C. Urrea, L. Valenzuela, and J. Kern, 'Design, Simulation, and Control of a Hexapod Robot in Simscape Multibody', in Applications from Engineering with MATLAB Concepts, J. Valdman, Ed., InTech, 2016. doi: 10.5772/63388.

[14] S. J. Moon, J. Kim, H. Yim, Y. Kim, and H. R. Choi, 'Real-Time Obstacle Avoidance Using Dual-Type Proximity Sensor for Safe Human-Robot Interaction', IEEE Robot. Autom. Lett., vol. 6, no. 4, pp. 8021–8028, Oct. 2021, doi: 10.1109/LRA.2021.3102318.

[15] K. Koyama, M. Shimojo, T. Senoo, and M. Ishikawa, 'High-Speed High-Precision Proximity Sensor for Detection of Tilt, Distance, and Contact', IEEE Robot. Autom. Lett., vol. 3, no. 4, pp. 3224–3231, Oct. 2018, doi: 10.1109/LRA.2018.2850975.