Koc. J. Sci. Eng., 7(2): (2024) 166-178 https://doi.org/10.34088/kojose.1440839

Kocaeli University

Kocaeli Journal of Science and Engineering

http://dergipark.org.tr/kojose

Kinematics-Based Motion and Balance Control of a Six-Legged Spider Robot

Enes VARDAR [1,*], Kenan IŞIK [2], H. Metin ERTUNÇ [3]

[1] *Department of Mechatronic Engineering, Kocaeli University, Kocaeli, 41001, Turkey,* ***ORCID:*** *0000-0002-8884-9679*
[2] *Department of Mechatronic Engineering, Karabük University, Karabük, 78050, Turkey,* ***ORCID:*** *0000-0002-0973-5180*
[3] *Department of Mechatronic Engineering, Kocaeli University, Kocaeli, 41001, Turkey,* ***ORCID:*** *0000-0003-1874-3104*

Article Info

Research paper

Received : *February 21, 2024*
Accepted : *May 23, 2024*

Keywords

Hexapod,
Kinematics,
Spider Robot,
Simulation

Abstract

In this study, the kinematics of a hexapod spider robot with 18 joints, consisting of 3 joints per leg, were modeled. The aim was to provide a theoretical framework for algorithms enabling the robot to walk, change direction, and control its body coordinate system. A novel and parametric approach was taken by creating a model instead of a table listing joint positions for various scenarios typically used in spider robot motion trajectories. Initially, the kinematic model of the hexapod robot was established using the D-H method. Subsequently, algorithms derived from kinematic equations were tested in a simulation environment to enable walking, rotation, and movement within the body's coordinate system. The simulation visualized movement trajectories without relying on mathematical libraries or specific programming languages, ensuring flexibility across different environments. Results from simulations and experimental tests demonstrated realistic movement capabilities. The software, validated in the simulation environment, was successfully implemented on a physical spider robot, leading to effective operation.

1. Introduction

Numerous research studies have been conducted for decades on multi-legged walking robots created to mimic the structure and movement control of limbs in insects and arthropods. Among multi-legged robots, hexapod robots have been a popular choice for a wide range of tasks [1] as they are less affected by environmental conditions than wheeled robots [2]. With the advancement of robot technology, robotic applications have expanded beyond industrial applications and gained popularity in other sectors such as service, medicine, and cleaning [3]. Possible areas of application of hexapod robots include volcanic exploration, search and rescue applications, detection of land mines, sample collection in locations that are not easily reachable, and life search in extraterrestrial explorations. Many of these tasks are dangerous and are often conducted in challenging environments that are hazardous to human beings [4]. When a hexapod robot moves, each joint can be in various joint positions. In some applications, the necessary joint angles for all the joints are pre-calculated for every possible leg configuration during walking and these calculated joint angles are stored as a table.

These pre-calculated joint angles are later used during walking [5-6]. In such applications, when the dimensions of the limbs of the robot change, these tables need to be recreated. Creating these tables is a laborious task. Additionally, when the robot is moved based on this table, the positioning resolution of the legs is limited to the size of the table, and the mobility of the robot is therefore restricted by the table. There are many studies other than the tabular solution.

For example, Sun et al. experimentally studied the walking and turning abilities of a six-legged spider robot [7]. In their study, forward and inverse kinematic equations were used to implement the robot's movement algorithms, and D-H parameters were employed to derive the kinematic equations. They planned the movement trajectories of the robot's toes using cosine, sine, and straight-line functions and executed the movements associated with these trajectories.

Thilderkvist and Svensson conducted a thesis

* Corresponding Author: enesvrdr0@gmail.com

utilizing a model-based design for controlling a spider robot [8]. They implemented enhancements to alter the movement, walking pattern, rotation, and body height of the spider robot. Additionally, their thesis encompassed a successful balancing mode. In their experimental findings, they managed to maintain the main body in an upright position despite changes in the ground angle of the robot.

Yamagan conducted a kinematic analysis of a six-legged spider robot in his thesis and subsequently simulated it in the MATLAB environment to facilitate real-time system implementation [9]. This study serves as a preliminary investigation for hardware simulation of actual applications involving six-legged robots and rapid prototyping applications. The D-H approach, commonly employed in analyzing forward and inverse kinematics of designed systems, was utilized. Additionally, the study mentioned the use of the Lagrange-Euler equation for solving robot dynamics.

Ekelund conducted a thesis focused on enabling a spider robot to walk and maintain balance [10]. PID control was employed to ensure system stability. He tested the implemented algorithms for walking and balance maintenance in both simulation and real-world environments, examining the experimental results obtained from the actual system.

Erkol investigated real-time kinematic control of multi-jointed robots [11], with a particular emphasis on minimizing the processing load associated with kinematic calculations. To achieve this, they developed FPGA-based hardware. Utilizing this hardware, kinematic calculations were performed, enabling motion control of a six-legged spider robot. D-H parameters were employed to facilitate the kinematic calculations. Trajectories were generated, and the robot's motion was coordinated by referencing each foot base relative to the coordinate system corresponding to the leg connected to the body.

In this study, the forward and inverse kinematic equations of a 6-legged spider hexapod robot were derived. Using these equations, motion algorithms for the robot have been developed and initially used in a simulation environment for verification of motion patterns and then applied to a real hexapod robot. Using the derived equations, the points of contact of each leg with the ground were controlled with developed motion algorithms, allowing the spider robot to perform realistic movements such as walking and rotating. Since the algorithms do not use any table, high mobility was achieved using the robot's kinematic model. Furthermore, by deriving the kinematic model, a foundation was provided for the angular control of the robot's body, enabling the development of algorithms for the robot to maintain balance on surfaces with changing slopes.

2. Hexapod Robot

In the classification of multi-legged robots, the hexapod robot is defined as a robot with six legs for mobility. The CAD drawing of the robot used in this study is shown in Figure 1. Each leg has three degrees of freedom.

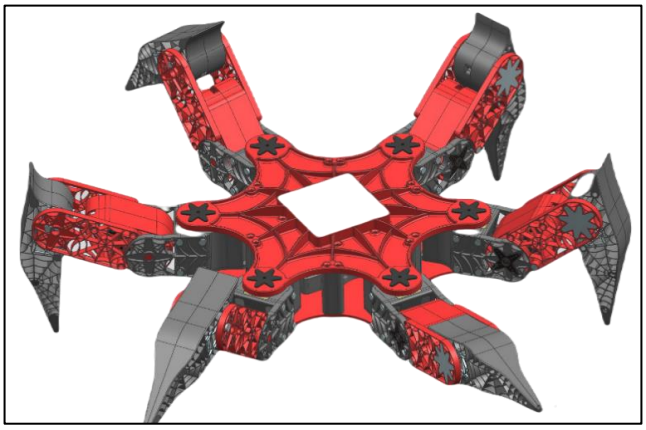

Figure 1. Hexapod Robot

The rotation directions of each joint are shown in Figure 2. All the joints in the legs can rotate around the Z-axis. As shown in Figure 2, since each leg has 3 joints, there are a total of 18 joints on the robot. In addition to the coordinate systems placed on these joints, a coordinate system has also been placed at the center of the robot's body for use in kinematic calculations. Additionally, there are imaginary points at the base of each foot. The positions of these imaginary points are determined by solving the kinematic equations of the legs with respect to the coordinate system at the point where the leg is connected to the body. The robot is moved by bringing these imaginary points to the desired positions.

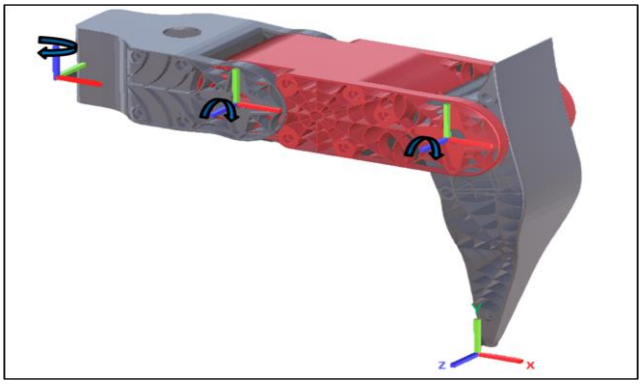

Figure 2. Axes located on each leg

3. Kinematic

In this section, the forward and inverse kinematic solutions of the developed spider robot are presented. The

algorithms used for the robot's walking and various movements are developed using the fundamental kinematic equations derived in this section.

3.1. Forward Kinematics

To obtain the kinematics of the robot under consideration, the modified Denavit-Hartenberg (D-H) parameters method was adopted and applied to one of the legs of the spider robot. This robot consists of a symmetrical structure composed of six identical legs, each with three degrees of freedom. As shown in Figure 2, each leg has 3 coordinate systems corresponding to each joint and an additional coordinate system holding the position of the contact point of the leg with the ground. In these coordinate systems, the blue vector represents the Z-axis, the green vector represents the Y-axis, and the red vector represents the X-axis.

The second coordinate system is obtained by rotating the first coordinate system around the X-axis by -90 degrees. The third coordinate system is a translated version of the second coordinate system. The first three coordinate systems lie on the same line.

While the first joint allows the leg to rotate around the robot's body, the second and the third joints change the vertical and lateral position of the leg's contact point with the ground. The fourth frame is an auxiliary frame which is used to calculate the position of the foot.

Figure 3 shows the locations and orientations of the frames that will be used in the kinematic analysis of the robot. The structure shown in Figure 3 is different from the one shown in Figure 2.

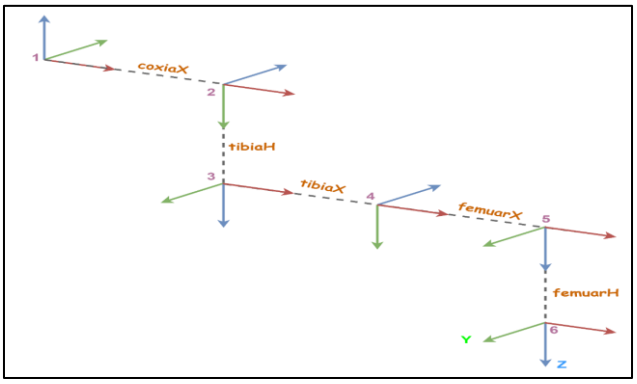

Figure 3. General coordinate system structure

This is because the system aims to be parametric, making it applicable to spider robots with different structures when needed.

Enhancing flexibility provides a significant advantage during both the software development process and its subsequent use, offering users a broader array of options [12]. In this study, the structure illustrated in Figure 3 was employed to ensure the software's compatibility with robots featuring varying lengths or kinematics. 1st, 2nd and 4th frames represent the rotating joints of the leg. As shown in the figure, the spider robot is defined by entering the length values of $coxia_X$, $tibia_H$, $tibia_X$, $femuar_X$ve $femuar_H$as parameters according to the mechanical structure of the robot. These parameter names are the names used to physically identify the bit spider [13].

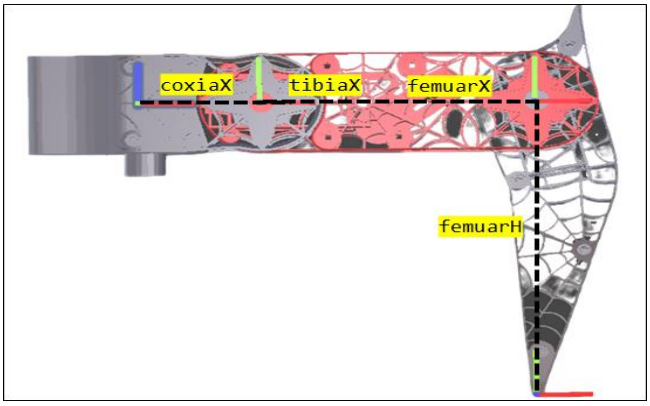

Figure 4. Length parameters on the leg

On the robot leg used in this study, as shown in Figure 4, the $tibia_H$ value is zero, and the $coxia_X$ and $tibia_X$ are side by side. The parameters obtained by applying the D-H method to these joints are shown in Table 1.

Tablo 1. D-H variables

i	α_{i-1}	a_{i-1}	d_i	θ_i
1	0	0	0	θ_1
2	-90	$coxia_X$	0	θ_2
3	-90	0	$tibia_H$	0
4	90	$tibia_X$	0	θ_3
5	-90	$femuar_X$	0	0
6	0	0	$femuar_H$	0

After the determination of the D-H parameters of the robot, the template given in Equation (1) is used for calculating the transformation matrices in forward kinematics analysis [14].

$$ {}^{i-1}_{i}T = \begin{bmatrix} cos\theta_i & -sin\theta_i & 0 & a_{i-1} \\ sin\theta_i cos\propto_{i-1} & cos\theta_i cos\propto_{i-1} & -sin\propto_{i-1} & -sin\propto_{i-1} d_i \\ sin\theta_i sin\propto_{i-1} & cos\theta_i sin\propto_{i-1} & cos\propto_{i-1} & cos\propto_{i-1} d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (1) $$

Using the D-H parameters method and the parameters in Table 1, the transformation matrices for each frame were found as given in the equations below.

$$ {}^{0}_{1}T = \begin{bmatrix} cos\theta_1 & -sin\theta_1 & 0 & 0 \\ sin\theta_1 & cos\theta_1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2) $$

$$ {}^{1}_{2}T = \begin{bmatrix} cos\theta_2 & -sin\theta_2 & 0 & coxia_X \\ 0 & 0 & 1 & 0 \\ -sin\theta_2 & -cos\theta_2 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3) $$

$$ {}^{2}_{3}T = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & tibia_H \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4) $$

$$ {}^{3}_{4}T = \begin{bmatrix} cos\theta_3 & -sin\theta_3 & 0 & tibia_X \\ 0 & 0 & -1 & 0 \\ sin\theta_3 & cos\theta_3 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (5) $$

$$ {}^{4}_{5}T = \begin{bmatrix} 1 & 0 & 0 & femuar_X \\ 0 & 0 & 1 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (6) $$

$$ {}^{5}_{6}T = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & femuar_H \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (7) $$

The transformation matrix ${}^{0}_{6}T$, which provides the position and orientation of the robot's foot with respect to the frame on the robot body, is obtained by multiplying the transformation matrices given by Equations (2) - (7) as shown in Equation (8).

$$ {}^{0}_{6}T = {}^{0}_{1}T\ {}^{1}_{2}T\ {}^{2}_{3}T\ {}^{3}_{4}T\ {}^{4}_{5}T\ {}^{5}_{6}T \quad (8) $$

The position of the foot is given by,

$$ p_x = cos\theta_1(coxia_X + a - b + c - d) \quad (9) $$

$$ p_y = sin\theta_1(coxia_X + a - b + c - d) \quad (10) $$

$$ p_z = -a - b - tibia_H cos\theta_2 - tibia_X - sin\theta_2 \quad (11) $$

where

$$ a = femuar_X cos\theta_{2+3} \quad (12) $$

$$ b = femuar_H sin\theta_{2+3} \quad (13) $$

$$ c = tibia_X cos\theta_2 \quad (14) $$

$$ d = tibia_H sin\theta_2 \quad (15) $$

3.2. Inverse Kinematics

Inverse kinematics is the process of determining the required joint angles for bringing the end effector to a desired position/orientation in Cartesian space [14]. Unlike the problem posed by forward kinematics, the procedure for obtaining these equations of joint angles is highly dependent on the configuration of the robot. This complicates the process significantly, as acquiring these equations systematically poses considerable challenges. In such cases, inverse kinematics is obtained with geometric evaluations based on the shape of the leg [14]. Therefore, to derive the inverse kinematic equations of the system, Equation (16) is obtained from the forward kinematic equations.

$$ {}^{0}_{1}T^{-1}\ {}^{0}_{6}T = {}^{1}_{2}T\ {}^{2}_{3}T\ {}^{3}_{4}T\ {}^{4}_{5}T\ {}^{5}_{6}T \quad (1) $$

The product on the left-hand side of Equation (16) is named S_L and is given by Equation (17). The result of the product on the right-hand side is named S_R and is represented by Equation (18).

$$ \begin{bmatrix} \dots & \dots & \dots & p_x cos\theta_1 cos\theta_2 - p_z sin\theta_2 - cos\theta_2 coxia_X + p_y cos\theta_2 sin\theta_1 \\ \dots & \dots & \dots & sin\theta_2 coxia_X - p_z cos\theta_2 - p_x cos\theta_1 sin\theta_2 - p_y sin\theta_1 sin\theta_2 \\ \dots & \dots & \dots & p_y cos\theta_1 - p_x sin\theta_1 \\ \dots & \dots & \dots & 1 \end{bmatrix} \quad (2) $$

$$ \begin{bmatrix} \dots & \dots & \dots & tibia_X + femuar_X cos\theta_3 - femuar_H sin\theta_3 \\ \dots & \dots & \dots & tibia_H + femuar_H cos\theta_3 - femuar_X sin\theta_3 \\ \dots & \dots & \dots & 0 \\ \dots & \dots & \dots & 1 \end{bmatrix} \quad (3) $$

By equating Equations (17) and (18), Equations (21), (24), and (25) are obtained.

$$ S_{L1} = p_x cos\theta_1 cos\theta_2 - p_z sin\theta_2 - cos\theta_2 coxia_X + p_y cos\theta_2 sin\theta_1 \quad (4) $$

$$S_{R1} = tibia_X + femuar_X cos\theta_3 - femuar_H sin\theta_3 \quad (5)$$

$$S_{L1} = S_{R1} \quad (6)$$

$$S_{L2} = sin\theta_2 coxia_X - p_z cos\theta_2 - p_x cos\theta_1 sin\theta_2 - p_y sin\theta_1 sin\theta_2 \quad (7)$$

$$S_{R2} = tibia_H + femuar_H cos\theta_3 - femuar_X sin\theta_3 \quad (8)$$

$$S_{L2} = S_{R2} \quad (9)$$

$$0 = p_y cos\theta_1 - p_x sin\theta_1 \quad (10)$$

By using Equation (10) the angular value of θ_1 is found as in Equation (11).

$$\theta_1 = tan\frac{p_y}{p_x} \quad (11)$$

By using the representation in Equation (27), Equations (13) and (14) were obtained from Equations (19) and (20).

$$a = p_x cos\theta_1 - coxia_X + p_y sin\theta_1 \quad (12)$$

$$S_{L1} = cos\theta_2 a - p_z sin\theta_2 \quad (13)$$

$$S_{L2} = -sin\theta_2 a - p_z cos\theta_2 \quad (14)$$

By taking the sum of the squares of the equalities $S_{L1} = S_{R1}$ and $S_{L2} = S_{R2}$ and using the following representations, the Equation (32) was obtained.

$$b = (2tibia_H a + 2p_z tibia_X) \quad (15)$$

$$c = (-2tibia_X a + 2p_z tibia_H) \quad (16)$$

$$d = sin\theta_2 b + cos\theta_2 c \quad (17)$$

By using Equation (10) the angular value of θ_2 is found as in Equation (11).

$$\theta_2 = atan2(b,c) \pm atan2((b^2 + c^2 - d^2)^{1/2}, d) \quad (18)$$

θ_3 is found as in Equation (37) by solving the equality $S_{L2} = S_{R2}$ using the following representations.

$$a = femuar_X \quad (19)$$

$$b = femuar_H \quad (20)$$

$$c = sin\theta_2 coxia_X - p_z cos\theta_2 - p_x cos\theta_1 sin\theta_2 - p_y sin\theta_1 sin\theta_2 - tibia_H \quad (21)$$

$$\theta_3 = atan2(a,b) \pm atan2((a^2 + b^2 - c^2)^{1/2}, c) \quad (22)$$

Tests performed on the robot with both solutions of equations 33 and 36 in the Unity 3D simulation environment have shown the correct results of the positive solutions.

4. Algorithms

In this study, by systematically relocating the coordinate frames on the robot's body, the kinematic relationships between each leg and the body were defined, enabling the robot's body movements to be realized in a coordinated and natural manner. Additionally, by establishing the relationship between the leg and the body, walking movements were made in harmony with the rotation of the coordinate system on the body, allowing the robot to adapt to changes in rotation. Furthermore, this relationship provided the robot with capabilities such as maintaining balance on inclined surfaces and direction stabilization. For the spider robot to perform various movements smoothly, balanced, and in a controlled manner, the legs need to work in synchrony with each other. While the legs in each group work in sync, the legs outside the moving group maintain the foot base position in their initial location before starting the movement. Typically, in previous studies, robot legs were divided into two groups and moved accordingly. In this way, the three non-moving legs remain on the ground while the other three legs remain in the air, allowing the spider robot to move without falling over.

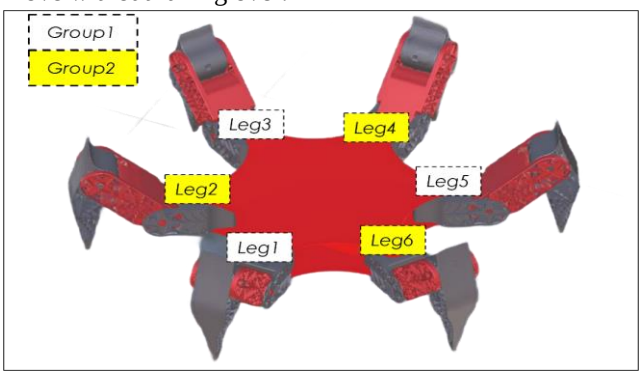

Figure 5. Leg groups on the robot

For this purpose, the six feet of the spider robot were divided into two groups, as shown in Figure 5, and moved sequentially. For kinematic calculations, imaginary coordinate systems were placed at the attachment points of the legs to the body and in the space where the robot moves, as shown in Figure 6. The coordinate systems at the attachment points were named Leg1 through Leg6. Additionally, an imaginary coordinate system outside the robot, named Origin Cs, was created as a reference for the coordinate systems on the robot. The main goal of robot control is to move the foot base position along the desired trajectory.

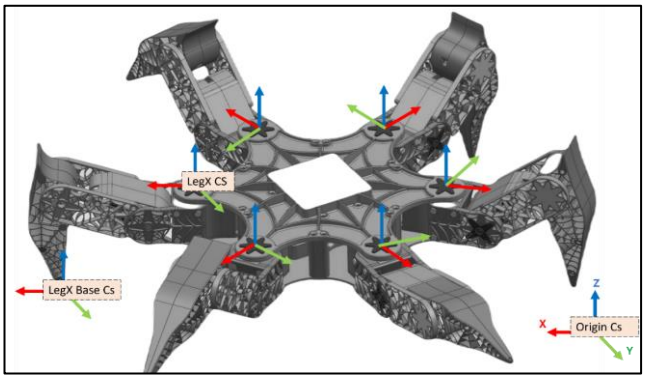

Figure 6. Coordinate systems on the robot

To achieve this, the angles θ_1, θ_2 and θ_3 of the joints, necessary for moving the foot base along the desired trajectory, need to be calculated. These angle values are found by inverse kinematics solution of the foot base position with respect to LegX. Additionally, there is a coordinate system called Body on the body of the robot, containing the position information of the robot's body.

The position of the foot base with respect to the origin coordinate system is determined by Equation (38).

$$ {}^{O}_{BASE}P = P_{(X_{OB},Y_{OB},Z_{OB})} \tag{38} $$

The position of the point where the leg is attached to the body (LegX) with respect to the origin coordinate system is calculated by Equation (39).

$$ {}^{O}_{LX}P = P_{(X_{OL},Y_{OL},Z_{OL})} \tag{39} $$

A rotation matrix describes the orientation of a coordinate system {B} concerning another coordinate system {A}. When this rotation matrix is known, it is possible to define the position of a point defined in coordinate system B with respect to coordinate system A [12]. The position of the foot base with respect to the coordinate system where the foot is attached to the body is calculated by Equation (40).

$$ {}^{LX}_{BASE}P = {}^{O}_{LX}R^{-1}\ {}^{O}_{BASE}P \tag{40} $$

The position of the foot base with respect to the origin coordinate system is calculated by Equation (41).

$$ {}^{O}_{B}P = {}^{O}_{LX}R\ {}^{LX}_{BASE}P \tag{41} $$

Each ${}^{O}_{LX}P$ value is calculated using the transformations between the coordinate system in the robot's body and the origin coordinate system.

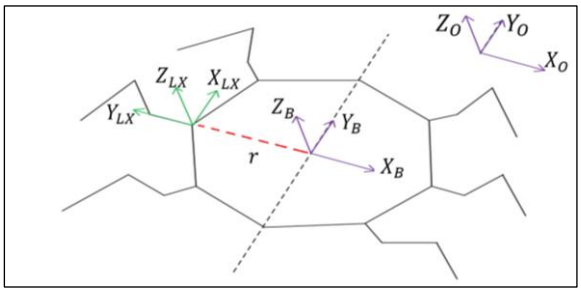

Figure 7. LX, Body, and origin coordinate systems

As shown in Figure 7, the LX coordinate system is calculated with Equation (42) with respect to the origin coordinate system. The value *r*, as shown in the figure, is expressed by ${}^{B}_{LX}P$. This value represents the position of the LX coordinate system with respect to the body. A similar approach has been used in the kinematic calculations and transformations of a spider robot climbing on a wall using a vacuum in literature [15]. Establishing the connection of the LX coordinate system with the robot's body causes the legs to adapt to different orientations of the robot's body.

$$ {}^{O}_{LX}P = {}^{O}_{B}R\ {}^{B}_{LX}P + {}^{O}_{B}P \tag{42} $$

4.1. Walking Algorithm

As mentioned in the previous sections, in the walking algorithm, robot legs are divided into two different groups and moved. Movements are carried out in an orderly and sequential manner.

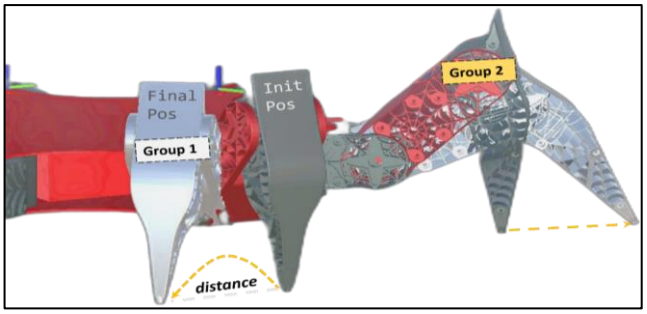

Figure 8. Movement pattern of legs in different groups

Figure 8 illustrates the movements of the legs in two different groups of the robot during walking. The legs shown in gray indicate the position of the robot before starting the movement, while the white ones show the position of the legs after one step of walking. As shown in Figure 8, one group of legs remains on the ground to prevent the robot from tipping over during the robot's movement, while the other group of legs moves to make the robot move forward. At the same time, the legs in contact with the ground move the body forward by the desired distance.

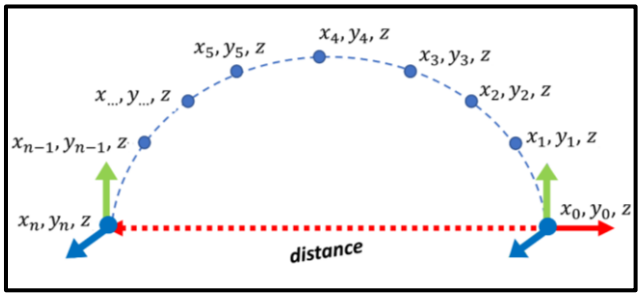

Figure 9. Trajectory of a leg in Group 1

Figure 9 shows the trajectory of a leg in Group 1. This trajectory represents the path followed by the foot base of a leg in the LegX coordinate system. Here, the leg base is moved in the X-axis of the LegX coordinate system by the desired distance the robot needs to cover. For example, suppose the robot needs to move a distance *K*. In each time sample *t* during the movement, the foot base is moved by a distance x_1 on the X-axis and $\pm y_1$ on the Y-axis. If half of the distance to be traveled has been completed, the foot base is moved downward by a distance $-y_1$ in the Y-axis.

When the robot needs to move a distance K in the X-axis of the origin coordinate system, the following steps are executed sequentially. The foot base position of each leg is known with respect to the origin coordinate system before these steps are executed.

1. The position of the Body coordinate system on the body is moved by a distance x_1 in the X-axis with respect to the origin coordinate system using Equation (43).

$$ {}^{O}_{BODY}P \;=\; {}^{O}_{BODY}P(x + x_1, y\ , z) \tag{43}$$

2. The coordinate system on the foot base of each leg in Group 1 is moved by a distance x_1 in the X-axis with respect to the origin coordinate system using Equation (44).

$$ {}^{0}_{BASE}P \;=\; {}^{O}_{BASE}P(x + x_1, y\ , z) \tag{44}$$

3. The coordinate system on the foot base of each leg in Group 1 is moved in the Y-axis with respect to the origin coordinate system using Equation (45).

3.1 If half of the targeted distance has not been completed in the X-axis, the foot base is moved by a distance $+y_1$ in the Y-axis.

3.2 If half of the targeted distance has been completed in the X-axis, the foot base is moved by a distance $-y_1$ in the Y-axis.

$$ {}^{0}_{BASE}P \;=\; {}^{O}_{BASE}P(x, y \pm y_1\ , z) \tag{45}$$

4. After the first three steps are executed, the inverse kinematics of the system is solved. The following steps are sequentially processed for each leg group.
5. The rotation matrix ${}^{0}_{LX}R$ of the point where the foot is attached to the body concerning the origin coordinate system is calculated.
6. The rotation matrix ${}^{0}_{BODY}R$ of the coordinate system on the body for the origin coordinate system is calculated.
7. Using the position of the coordinate system on the body with respect to the origin coordinate system ${}^{0}_{BODY}P$ and the position of the coordinate system to which a leg is attached with respect to the body's coordinate system ${}^{BODY}_{LX}P$, the position value of the coordinate system where the leg is connected to the body is calculated with respect to the origin coordinate system using Equation (46).

$$ {}^{O}_{LX}P \;=\; {}^{O}_{BODY}R\,{}^{BODY}_{LX}P + {}^{O}_{BODY}P \tag{46}$$

8. Using Equation (46) and the rotation matrix ${}^{0}_{LX}R$, the transformation matrix of the coordinate system where a leg is connected to the body with respect to the origin coordinate system is calculated as follows with Equation (47).

$$ {}^{O}_{LX}T = \begin{bmatrix} {}^{0}_{LX}R & {}^{O}_{LX}P \\ 0\ 0\ 0 & 1 \end{bmatrix} \tag{47}$$

9. If the inverse of Equation (47) is taken, the transformation matrix in Equation (48) is obtained. By means of this matrix, any position value known with respect to the origin coordinate system can be calculated in the LegX coordinate system.

$$ {}^{LX}_{O}T \;=\; {}^{O}_{LX}T^{-1} \tag{48}$$

10. Using Equation (48), the position information for each foot base known with respect to the origin coordinate system is calculated in the LegX coordinate systems.

$$ {}^{LX}_{BASE}P \;=\; {}^{LX}_{O}T\ {}^{O}_{BASE}P \tag{49}$$

11. Using the position information in Equation (49), inverse kinematic equations are solved in order to calculate the angles θ_1, θ_2 and θ_3.
12. If the distance to be traveled has been covered, the group with stationary foot bases is changed, and the algorithm proceeds to the first step. If the distance has not been covered, the first four steps are repeated sequentially.

Table 2. Transition table for the walking algorithm

T1.1	state = walking
T1.2	unconditional
T1.3	0 ≤ stepX and stepX ≤ k/2
T1.4	k/2 ≤ stepX and stepX ≤ k
T1.5	unconditional
T1.6	unconditional
T1.7	unconditional
T1.8	state ≠ stop or stepX ≠ k
T1.9	state = stop and stepX = k
T1.10	Feet group = Group 2
T1.11	Feet group = Group 1
T1.12	unconditional

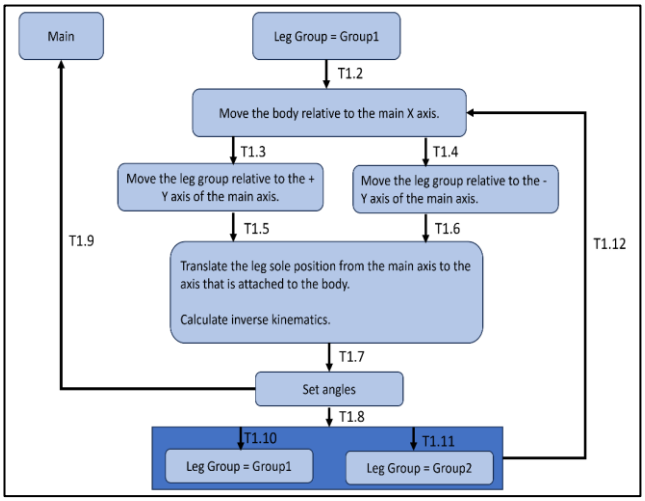

Figure 10. State machine for the walking algorithm

Figure 10 depicts the state machine of the walking algorithm. Table 2 provides the transition conditions in this state machine. In the first step, the foot group to move is configured, as shown in the figure. Subsequently, the body is moved. In the next step, the relevant group leg is moved up or down. If a stop command is given, the algorithm waits for the completion of the relevant movement, as indicated in the table. It is crucial for the robot to be in the initial position before starting any movement. The algorithms are designed to complete and terminate a movement and are developed to be run from a specific initial position. Unconditional transitions are shown in Table 2, created to move on to the next operation after any process is completed.

4.2. Body Movement Algorithm

The body movement algorithm is developed to allow the robot to rotate around itself. With this developed algorithm, the robot can change its walking direction.

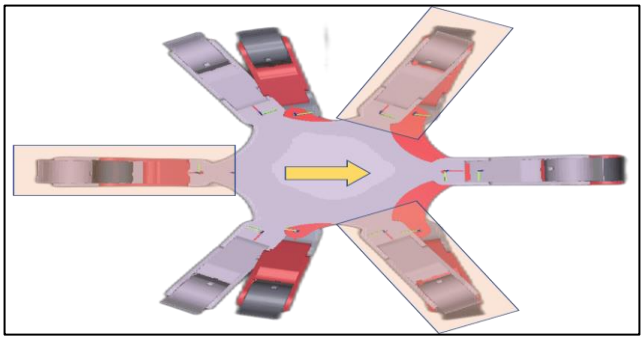

Figure 11. Top view of the robot

As shown in Figure 11, the foot bases of the legs in the yellow group maintain their positions. When the body is moved, these legs move as shown in the figure above to preserve their foot base positions. As shown in the figure, the coordinate systems located on the axis where each leg is connected to the body move along with the body. Movement occurs through the different actions of the two groups of legs, similar to the walking algorithm. While one group of legs tries to maintain the point where it touches the ground before the turning motion, the other group of legs performs the turning step along with the body. As the coordinate system on the axis where each leg is connected to the body, as shown in the figure, moves with the body, motion occurs through the group of legs that either preserves or does not preserve the place it steps on.

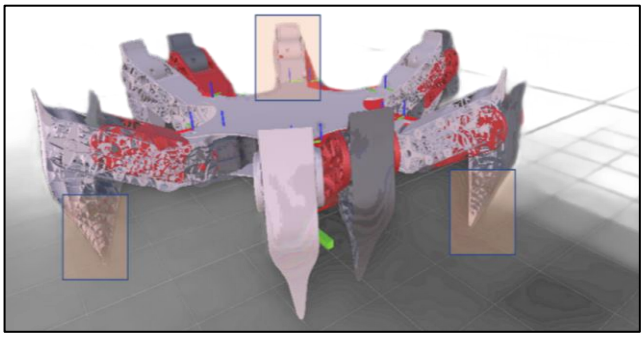

Figure 12. Side view of the robot

As shown in Figure 12, one group of legs preserves their location, while the other legs are in motion. This way, the body moves forward a certain distance, and the robot is made to walk. It is relatively easier to move the body of the robot only in the *x*, *y*, or *z*-axis. In this case, before moving the body, the position where each foot will step with respect to the origin coordinate system, ${}^{O}_{BASE}P$, is memorized. Then, the coordinate systems where each leg is connected to the body are moved by a distance of *k* concerning the origin coordinate system. The new position values of the coordinate systems are also denoted by ${}^{O}_{LX}P$.

The position of the LegX coordinate system ${}^{O}_{LXORG}P$, is calculated for the origin coordinate system. With the found position and rotation matrix, the position of the foot base is calculated in the new LegX coordinate system using Equation (50).

$$ {}^{LX}_{BASE}P = {}^{LX}_{O}R\,{}^{O}_{BASE}P + {}^{O}_{LXORG}P \tag{50}$$

The position information calculated with Equation (50) is solved with the equations in the inverse kinematics section to find the values of θ_1, θ_2 and θ_3.

Figure 13. Negative movement along the Y-axis

When the calculated values of θ_1, θ_2 and θ_3 are set to the axes, the robot performs the movements shown in Figure 13. The calculated values of θ_1, θ_2 and θ_3 perform the movements shown in Figure 13 when a desired axis is moved. As shown in Figure 13, when only the body of the robot is desired to move, all feet touch the ground and make a pushing movement. This pushing movement occurs by giving the angle values obtained from the inverse kinematics solutions, due to the shift of coordinate systems, to the axes in the legs.

4.3. Balancing Algorithm

In this section, the studies conducted for the robot to maintain balance on an inclined surface are explained. To allow the spider robot to stay balanced on an inclined surface, the pitch and roll angle changes on the body, as given in Figure 14, are calculated.

As shown in Figure 14, the pitch angle change corresponds to the rotation of the robot body around the X-axis, the roll angle change is around the Y-axis, and the yaw angle change is around the Z-axis.

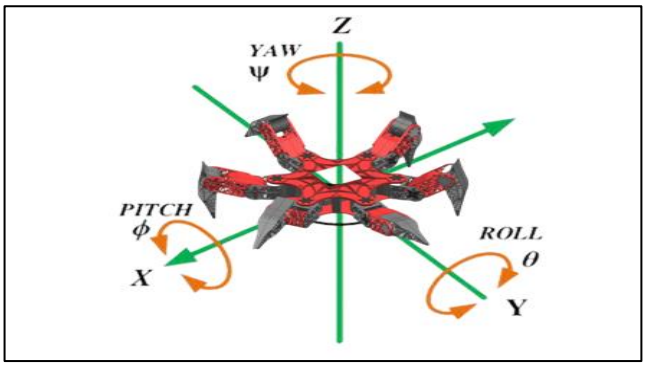

Figure 14. Roll, pitch, and yaw rotation axes on the spider robot

To calculate roll, pitch, and yaw angle changes, an MPU6050 sensor was used. The angular accelerometer values obtained from the sensor were used to calculate roll and pitch angle changes using the Equations (51) and (52) as follows [16]. The values A_x, A_y and A_z used in the equations represent the gravitational accelerations. The respective A value indicates the acceleration affected by gravity along the relevant axis of the device. The measurements provide the accelerations with the unit g.

$$pitch = tan^{-1}\left(\frac{Ax}{\sqrt{Ay^2+Az^2}}\right) \tag{51}$$

$$roll = tan^{-1}\left(\frac{Ay}{\sqrt{Ax^2+Az^2}}\right) \tag{52}$$

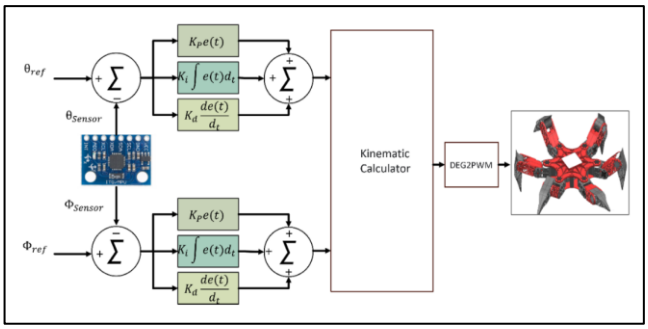

Figure 15. Balancing algorithm

The calculated angle values were used to keep the robot balanced using the algorithm shown in Figure 15. The roll and pitch angle values from the sensor and the reference angle values for the robot to stay used in a PID algorithm and then kinematic transformations were calculated.

5. Experimental Results

The developed algorithms were executed, and the trajectories for the foot bases were examined.

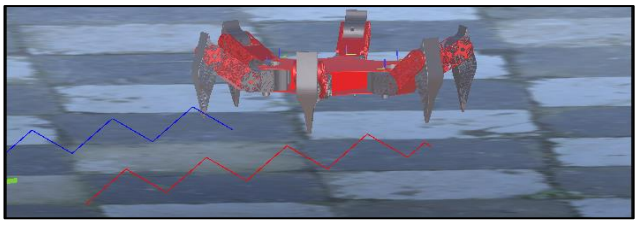

Figure 16. Walking trajectory of two different groups of legs in the Unity 3D environment

During the execution of the walking algorithms, the positions of the foot bases in two different groups are shown in Figure 16 for retrospective analysis. It can be observed that while one group of legs performs a stepping motion as desired in the walking algorithm, the legs in the other group maintain their position.

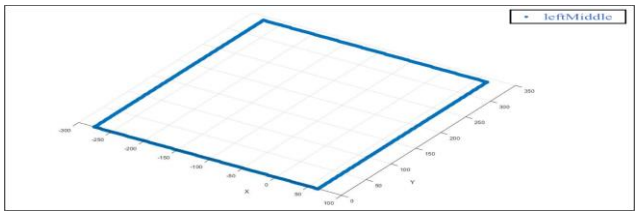

Figure 17. Position change of a leg's foot base during walking

During the test of the robot's walking algorithm, the robot was moved forward by 300 cm, to its right by 340 cm, then backward by 300 cm, and to its left by 340 cm. The positional changes in the X and Y axes of a leg's foot base were obtained as shown in Figure 17. As depicted in the figure, the foot base successfully returned to the initial position.

Figure 18. Implementation of the designed spider robot

A control algorithm with parametrically high maneuverability was successfully developed for the motion and control of the spider robot without creating any pre-calculated table. Furthermore, the algorithm was developed in the Unity 3D environment, a 3D game engine, and simulated. The code, whose accuracy and functionality were tested during the simulation, was transferred to an ARM-based processor, resulting in the realization of the spider robot shown in Figure 18. It was observed that the implemented spider robot successfully completed the motions tested during the simulation.

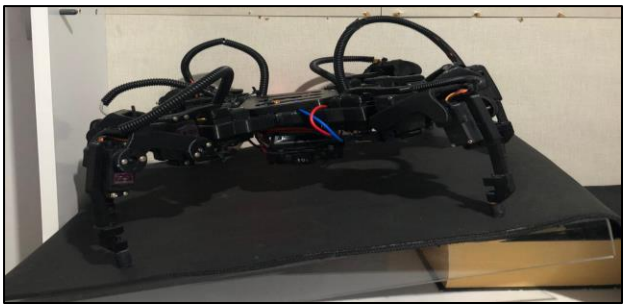

Figure 19. Test setup for the balancing algorithm

To test the effectiveness of the balancing algorithm, the robot was placed on an inclined surface as shown in Figure 19, and then the balancing mode was activated through a remote interface, and the results were observed. During the experiments, the K_p coefficient was initially changed, followed by the K_d coefficient, and finally, the K_i coefficient was changed. After each coefficient change, the test was repeated from the beginning. By altering the PID controller coefficients in this manner, it was possible to examine the robot's ability to maintain balance on an inclined surface. In this experiment, where the effect of PID coefficients, especially to the roll angle due to the slope of the tested surface, was observed, the roll angle change around the y-axis on the body of the robot was monitored.

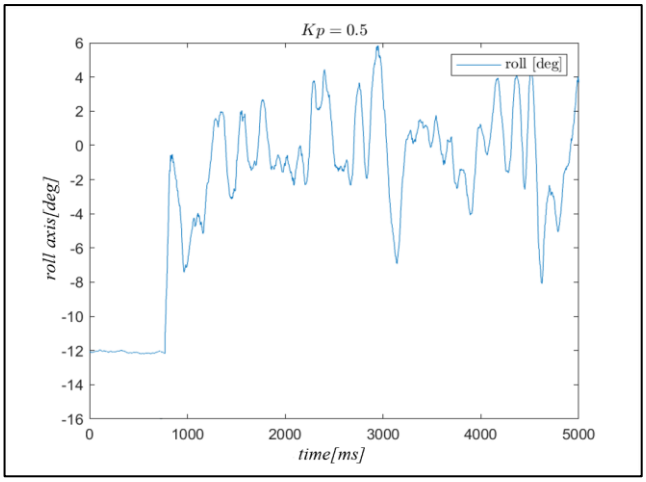

Figure 20. Roll angle change for K_p = 0.5

In the created test setup, the initial observation was

made by setting the K_p coefficient to 0.5. The roll angle change on the Y-axis in the robot's body, as shown in Figure 20, was obtained. The robot's effort to keep its body in a horizontal position was observed, however the robot could not stay completely horizontal and entered oscillations. As seen in the figure, while the robot attempted to keep its body parallel to the ground, oscillations in the roll angle between −6 and +6 degrees were observed.

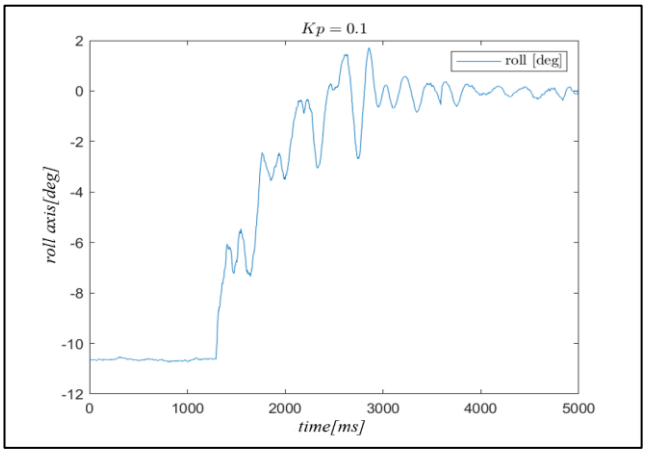

Figure 21. Roll angle change for K_P = 0.1

In the next step, the K_p coefficient was set to 0.1 to examine the robot's response. The roll angle change on the Y-axis in the robot's body was obtained as shown in Figure 21. The robot's effort to keep its body in a level position was observed. Reducing the K_p value resulted in a reduction in oscillations. With a K_p value of 0.1, it was observed that the robot's body came to a level position, and the steady-state error varied between −0.4 and +0.4 degrees.

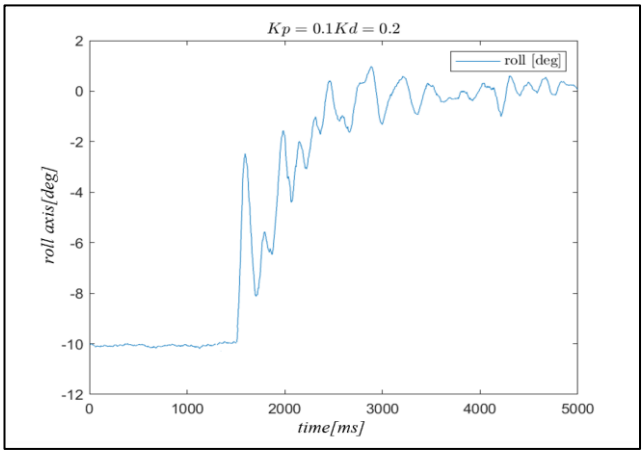

Figure 22. Roll angle change for K_p = 0.1 and K_d = 0.2

Following the K_p value, the roll angle change was recorded for K_d coefficient of 0.2, as shown in Figure 22. As in other tests, the robot's effort to keep its body in a parallel position is notable. Incorporating the K_d parameter effectively mitigated oscillations and reduced the steady-state error to 0.1 degrees.

In the next step, the response of the balance algorithm with the addition of the K_i coefficient is observed, as shown in Figure 23.

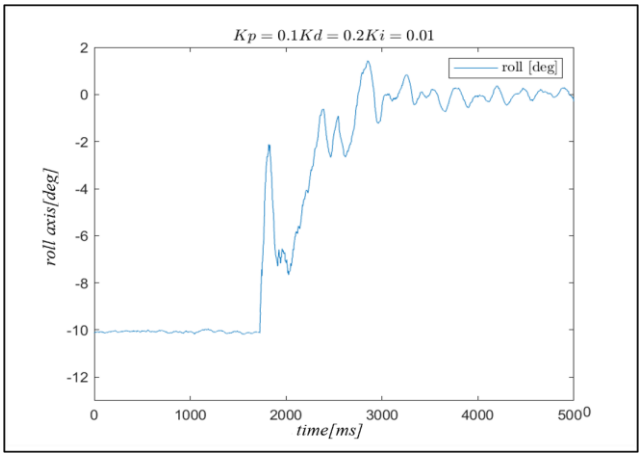

Figure 23. Roll angle change for K_p = 0.1, K_d = 0.2 K_i = 0.01

It was determined that the K_i parameter had little effect on the system. It was observed that the steady-state error remains in the range of 0.1 degrees.

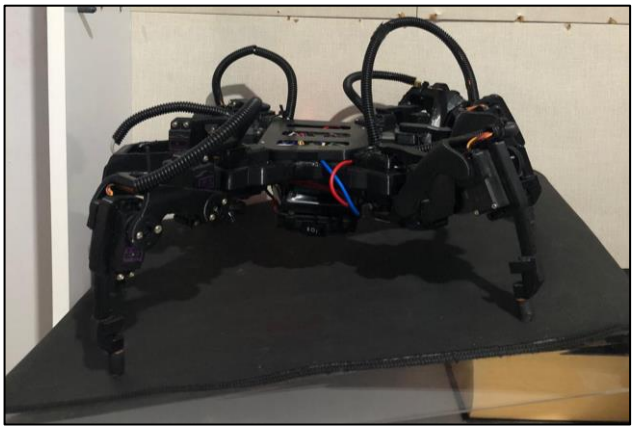

Figure 24. Effect of the balance algorithm on the test setup

Figure 24 shows the robot keeping the robot's body parallel to the ground on an inclined surface. The robot solved a kinematic model to keep its body parallel to the ground on an inclined surface by moving its legs to the corresponding position.

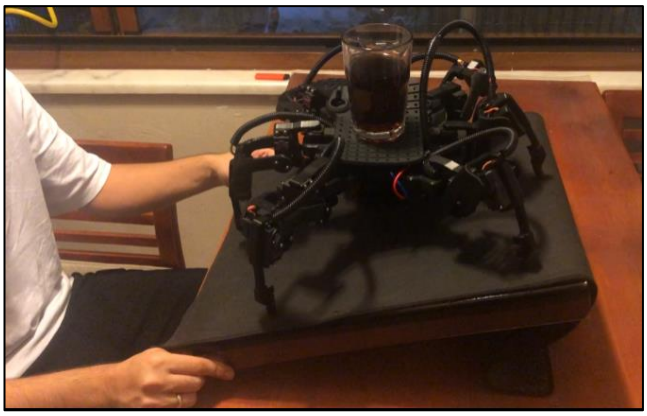

Figure 25. Test of the balance algorithm, condition 1

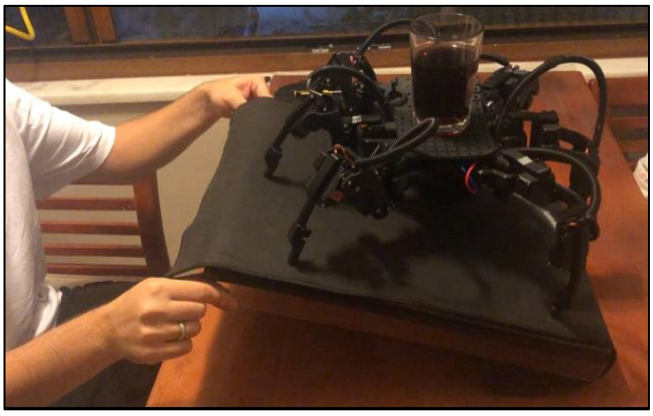

Figure 26. Test of the balance algorithm, condition 2

To test the system's balance on a platform, a test setup was prepared as shown in Figures 25 and 26. In the test setup, a cup filled with liquid was placed on the robot. Then, the platform was moved as shown in the figures, and the condition of the cup was observed. Due to the balance algorithm, the robot kept its body parallel to the ground, preventing the cup from falling and spilling its contents.

6. Conclusion

In this study, algorithms for walking, turning, and balance control on inclined surfaces for a hexapod spider robot were developed in virtual environment allowed for the rapid testing of many scenarios, and the impact on the robot was observed in a short time. To facilitate the transfer of code from a virtual environment to a real system with minimal changes, the code developed in the virtual environment did not use libraries specific to the development environment. Developments were made in Unity 3D without using any ready-made mathematical libraries. After the developed algorithms were transferred to a real spider robot, walking and turning algorithms were tested on the system. Based on the tests conducted, similar movements were observed in both the virtual and real environments. In the study, a PID-based algorithm was developed for the robot to maintain balance on inclined surfaces. The response of the robot on inclined surfaces was observed using different PID coefficients, and suitable coefficients were determined. It was observed that K_p and K_d coefficients were more effective on the system. With high coefficient values, the system was observed to oscillate and could not find the balance position. With the determined PID coefficients, the robot successfully maintained balance when tested on an inclined surface with a cup of water placed on it, without spilling the water. Information from the IMU sensor on the robot indicated that the robot-maintained balance with a precision of 0.1 degrees on the *roll* and *pitch* axes of its body. Further development of the robot is planned with new features to be added. With the addition of sensors to the robot, the goal is to determine the robot's position, follow desired trajectories on a map, and navigate around obstacles encountered in its path.

Declaration of Ethical Standards

The author of this article declares that the materials and methods used in this study do not require ethical committee permission and/or legal-special permission.

Conflict of Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

This work was conducted in the Sensor Laboratory of the Mechatronics Engineering Department at Kocaeli University.

References

[1] Urvaev I., Spirkin A., Bazykin S, 2022. Kinematic Control of The Hexapod Robot. IEEE 23rd International Conference of Young Professionals in Electron Devices And Materials, Altai, Russian Federation, 30-31 June 2022.

[2] Tedeschi F., Carbone G., 2014. Design Issues for Hexapod Walking Robots. Robotics, 3(2), 181-206. DOI:10.3390/robotics3020181

[3] Sastry, Naveen K., 2016. Design and development of a bio-inspired hexapod robot for search and rescue operations. Journal of Field Robotics, 33, pp. 452-475.

[4] Roth J., 2019. Trajectory Regulation for Walking Multipod Robots. International Journal on Advances in Systems and Measurements, 12, pp. 265-278.

[5] Quadruped Robot using Arduino | Spider Robot https://www.electronicshub.org/quadruped-robot-using-arduino/ 31/03/2023

[6] Spider Robot using Arduino https://www.flyrobo.in/blog/spider-robot-arduino 31/03/2023

[7] Sun J., Ren J., Wang B., Chen D., 2017. Hexapod Robot Kinematics Modeling and Tripod Gait Design Based on the Foot End Trajectory. IEEE International Conference on Robotics and Biomimetics, Macau, Macao, 12-13 December 2022.

[8] Thilderkvist, D., Svensson S., 2015. Motion Control of Hexapod Robot Using Model Based Design. Master Thesis, Lund University, Department of Automatic Control, Lund.

[9] Yamağan I., 2013. Altı Bacaklı Bir Robot için Dinamik Simülatör Tasarımı. Yüksek Lisans Tezi, Fırat Üniversitesi, Fen Bilimleri Enstitüsü, Karabük, 334623.

[10] Ekelund J., 2018. Balancing and Locomotion of a Hexapod Robot Traversing Uneven Terrain. Master Thesis, Lund University, Department of Automatic Control, Lund.

[11] Erkol H., 2015. Robot Kinematik Denklemlerinin FPGA ile Çözülmesi ve Çok Eklemli Bir Robota Uygulanması. Doktora Tezi, Karabük Üniversitesi, Fen Bilimleri Enstitüsü, Karabük, 405868.

[12] Martin R., 2008. Clean Code a Handbook of Agile Software Craftsmanship (1st ed.). Boston: Pearson.

[13] Jocqué R., Schoeman A., Sophia A. 2001. African Spiders an Identification Manual (1st ed.). United Kingdom: Pemberley Natural History Books BA.

[14] Bingül Z., Küçük S., 2006. Robot Kinematiği. İstanbul: Birsen Yayınevi.

[15] Xu S., He Bin., Hu H. 2019. Research on Kinematics and Stability of a Bionic WallClimbing Hexapod Robot. Hindawi Applied Bionics and Biomechanics, 1(1), 1-17, DOI:10.1155/2019/6146214

[16] https://www.analog.com/en/app-notes/an-1057.html,