Kinematics analysis of a quadruped robot: Simulation and Evaluation

Conference Paper · March 2022

DOI: 10.1109/ICIPRob54042.2022.9798744

8 authors, including:

Md Hasibur Rahman
Independent University

Md Fayed Al Monir
Independent University

Saadia Binte Alam
Independent University

Mohammad Shidujaman
Independent University

Kinematics analysis of a quadruped robot: Simulation and Evaluation

Md. Hasibur Rahman
Department of Electrical and Electronic Engineering
IUBAT
Dhaka, Bangladesh
errorhasib@gmail.com

Md. Mazharul Islam
Department of Electrical and Electronic Engineering
IUBAT
Dhaka, Bangladesh
mazharulislam629@gmail.com

Md. Fayed Al Monir
Department of Mechanical Engineering
IUBAT
Dhaka, Bangladesh
suadfayed28@gmail.com

Saadia Binte Alam
Department of Electrical and Electronic Engineering
IUBAT
Dhaka, Bangladesh
saadiabinte@gmail.com

Ruzbelt
Department of Electrical and Electronic Engineering
IUBAT
Dhaka, Bangladesh
ruzbelt123@gmail.com

Md. Mijanur Rahman
Department of Electrical and Electronic Engineering
IUBAT
Dhaka, Bangladesh
mijanur.bseee37@gmail.com

Mohammad Shidujaman
Department of Electrical and Electronic Engineering
AIUB
Dhaka, Bangladesh
shantothu@gmail.com

Rubaiyat Islam
Sifchain Finance
USA
rubaiyat.islam1982@gmail.com

Abstract— ***Kinematics analysis is essential for finding the robot manipulator's joint motions. Using kinematics, we can determine how the robot's arms will move and the possible direction of that arm's end-effector. Kinematics analysis is split into two parts forward Kinematics and Inverse Kinematics. Forward Kinematics measure the positions and orientations of the robot's end-effector, and we can calculate the joint angles by using inverse kinematics. This study provided mathematical derivation for forward and Inverse kinematics to determine our robots' positions and joint angles briefly. To implement different walking motions in a quadruped robot, firstly, we need to analyze the single-leg structure. In this study, we analyzed and derived the forward and inverse kinematics of the quadruped robots and verified their correlation. We used python and Pybullet physics engines to analyze our robot leg motions for mathematical programming.***

Keywords — Forward Kinematics, Inverse Kinematics, Pybullet, End effector, Position, Orientations, Joint angles.

I. INTRODUCTION

The Quadruped robot is a Bio-inspired Robot. Stair climbing algorithms develop a biologically inspired four-leg quadruped robot with the help of an Ultrasonic Distance sensor [1]. Quadruped robot is getting more concentration for its locomotion control. A quadruped robot consists of four legs, and each leg has a coxa, thigh, and tibia. Both thigh and tibia are connected through a knee joint [2]. For trajectory analysis and gait implementation, kinematics analysis is very much critical. Foot state estimation can be done by using forward and inverse kinematics studies. Translational gait planning is one of the most acceptable locomotion studies Xuedong Chin did in his research [3]. The translational crawl gait called TITAN-VIII was also designed by solving the forward and inverse kinematic problems [4]. To overcome irregular terrain, an adaptive locomotion control is proposed, consisting of a workspace trajectory generator and a motion generator [2]. It helps to generate the trajectories in real-time by checking its ground condition. It has mainly four legs for walking; every leg has a certain degree of freedom which is possible to manipulate through Electrical signals. Nowadays, quadruped robot is highly researched for solving the problem of rough terrain. It is difficult to go over rough terrain by wheel robot. In a legged robot, it is very much important to apply a verified kinematics model. A good kinematic model is required for dynamic stability analysis and path tracking. The two types of kinematics are forward and inverse kinematics. The joint variables are expressed in the forward kinematic analysis to identify the location of the robot's body. The location of the body is given in inverse kinematic analysis to identify the joint variables required to transport the body to the desired location. Solving inverse kinematics is computationally expensive and generally takes a very long time in the real-time control of manipulators [5]. A research group from Selçuk University showed the inverse and forward kinematics of their quadruped robot [6]. Denavit Hartenberg conventions are used to derive their forward kinematics [6], [7]. Denavit Hartenberg's method that uses four parameters is the most common method for describing the robot kinematics, and these parameters ai-1, αi-1, di, and θi are the link length, link twist, link offset, and joint angle, respectively [5]. The Denavit-Hartenberg (D-H) formalism and inverse kinematics are used to plan afoot WT, which has the advantages of low mechanical shock, smooth movement, and a streamlined trajectory [8]. Xuewen Rong derives the inverse kinematics equations using forward kinematics [7]. A new center of gravity trajectory planning algorithm for walking was proposed by Hyunkoo Park by using inverse kinematics [9]. In their proposed quadruped robot, each leg has seven joints, and the whole system has 28 degrees of freedom, which is a spider-type robot. In the design of the quadruped robot for all cases, kinematics is used to determine the angles of all the leg joints [2], [10], [11], [12]. Xiangqi Zeng's model is based on kinematics. All the leg structure is designed with the help of kinematics. Moreover, they find a fixed point for inverse kinematics for movement. Many researchers use two joints on their quadruped robots. If we see the biological leg structure of animals like a cat, it has three joints. Three joints give an extra advantage; it increases the running speed. The robot body's degrees of freedom (DOFs) are divided into two categories: the major DOFs,

978-1-6654-0771- 7/22/$31.00 ©2022 IEEE

which are required for walking, and maintaining the secondary DOFs. The motions of the major DOFs can be realized by discovering a suitable kinematic resolution of the motions of the minor DOFs [13]. There have many techniques for solving the inverse kinematics problems such as the Gradient Projection method [14] (GP), Weighted Least-Norm method [15] (WLN), and Extended Jacobian matrix [16] (EJ) method. The extended Jacobian matrix (EJ) method is used to solve the inverse kinematics and the Time-Pose control method to control the robot [17]. The inverse kinematics problem was also solved by a modified improved Jacobian pseudoinverse (mIJP) algorithm [9]. Based on inverse kinematics using the Jacobian of the whole body, gait planning was designed for a robot modeled on a gecko [18]. To track a planned trajectory for the Center of Mass a method for whole-body control of legged robots is developed [19]. Before implementing a robot algorithm, researchers need a safe environment for testing. The simulator is the safest environment to test the result of the newly implemented algorithm and the most trusted software RoboDK is the best simulator now for forward and inverse kinematics [20]. So, for implementing different types of gaits like walking, creep, transitional crawl, we need to find our robot's kinematics, and establish the forward and inverse kinematics to get the end effector positions and joint angles. In this study, our main objective is to derive forward and inverse kinematics and verify their co-relation.

II. Methodology

A. Coordinate System of a quadruped robot

A quadruped robot consists of four legs with a rigid body. Each leg has 3 degrees of freedom. All the links are connected by each other by revolute joints. There are three joints: the coxa joint, femur joint, and tibia joint. There are several walking systems in a quadruped robot, which makes this quadruped robot unique and animalistic such as walking, running, pace, canter, gallop, creep, and trot. To create a different gait in a quadruped robot, we need to find the forward and inverse kinematics of the robot. To find the forward kinematics, we need to solve the frame-to-frame rotation and translation matrices. By using forward kinematics, we can achieve the end-effector positions. Inverse kinematics gives us the different joint angles of the quadruped robot leg. If we follow figure 1, we can see links L_1, L_2, and L_3. We can see there is a total four number of frames. 0th frame X_0, Y_0, Z_0.1st frame X_1, Y_1, Z_1. 2nd frame X_2, Y_2, Z_2. 3rd frame X_3, Y_3, Z_3, and the Last frame is the fourth frame which is the end effector X_4, Y_4, Z_4. To proceed, we must determine the forward kinematics by referring to Figure 1. From the 0th to the fourth frame, we must perform multiple rotations and translations in sequence.

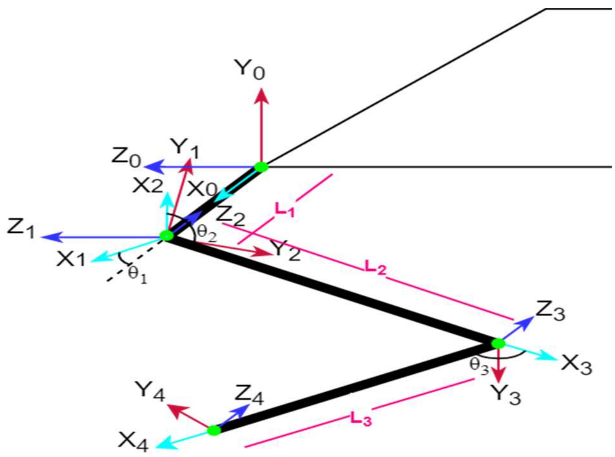


Fig. 1. Quadruped robot single-leg coordinate systems. It consists of 3 links and 3 revolute joints, which are L_1, L_2, L_3, and J_1, J_2, J_3.

B. Transformation matrix for each joint of a leg

We will have two matrices from the 0th frame to the first frame (according to Table 2). From figure 2, we can see that about the Z_0 axis; there will be a rotation θ_1. For that, we will have a rotational matrix about the Z_0 axis.

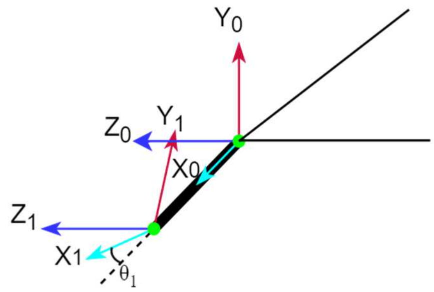


Fig. 2. Transforming 0th frame to 1st frame. This figure is used to find out the rotational and translation matrix for frame 0 to frame 1 where, X_0, Y_0, Z_0 is frame 0 and X_1, Y_1, Z_1 frame 1.

$$R_z = \begin{bmatrix} \cos(\theta_1) & -\sin(\theta_1) & 0 & 0 \\ \sin(\theta_1) & \cos(\theta_1) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$T_x = \begin{bmatrix} 1 & 0 & 0 & L_1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

We will also have a translational matrix along the X_0 axis. Now the transformation matrix from 0th frame to 1st frame will be $T_0^1 = R_z * T_x$

$$T_0^1 = \begin{bmatrix} \cos(\theta_1) & -\sin(\theta_1) & 0 & L_1\cos(\theta_1) \\ \sin(\theta_1) & \cos(\theta_1) & 0 & L_1\sin(\theta_1) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

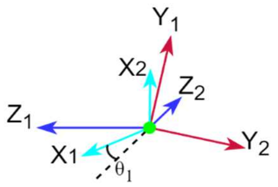


Fig. 3. Transforming 1st frame to 2nd frame. This figure is used to find out the rotational and translation matrix for frames 1 to frame 2. Where X_1, Y_1, Z_1 are frame 1 and X_2, Y_2, Z_2 are frame 2.

We are transforming the first frame into the second frame. This figure is used to find the rotational and translation matrix for frames 1 to 2. X_1, Y_1, Z_1 are frame one, and X_2, Y_2, Z_2 are frame 2. We will have two rotational matrices (according to the D-H convention). In figure 3, we can see that, about the Z_1 axis, there will be a 90º anti-clockwise rotation. For that reason, our rotational matrix about the Z_1 axis will be Rz. We can also see that along the X_1 axis; there will be another 90º clockwise rotations. For that reason, our rotational matrix about the X_1 axis will be Rx.

$$R_z = \begin{bmatrix} \cos(\theta_1) & -\sin(\theta_1) & 0 & 0 \\ \sin(\theta_1) & \cos(\theta_1) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$R_x = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(-pi/2) & -\sin(-pi/2) & 0 \\ 0 & \sin(-pi/2) & \cos(-pi/2) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Now the transformation matrix from the first frame to the second frame will be $T_1^2 = R_z * R_x$

$$T_1^2 = \begin{bmatrix} 0 & 0 & -1 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

We will have two matrices from the second frame to the third frame (according to the D-H parameter). From the figure, we can see that about the Z_2 axis; there will be a rotation θ_2. For that reason, we will have a rotational matrix about the Z_2 axis.

$$R_z = \begin{bmatrix} \cos(\theta_1) & -\sin(\theta_1) & 0 & 0 \\ \sin(\theta_1) & \cos(\theta_1) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

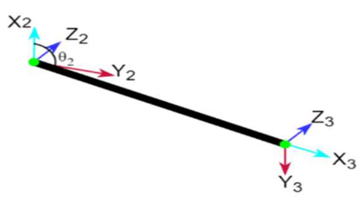


Fig. 4. Transforming 2nd frame to 3rd frame. This figure is used to find out the rotational and translation matrix for frames 2 to frame 3. Where X_2, Y_2, Z_2 are frame 2 and X_3, Y_3, Z_3 frame 3.

We will also have a translational matrix along the negative X_2 axis.

$$T_x = \begin{bmatrix} 1 & 0 & 0 & -L_2 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Now the transformation matrix from the second frame to the third frame will be $T_2^3 = R_z * T_x$

$$T_2^3 = \begin{bmatrix} \cos(\theta_2) & -\sin(\theta_2) & 0 & -L_2\cos(\theta_2) \\ \sin(\theta_2) & \cos(\theta_2) & 0 & -L_2\sin(\theta_2) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

We will have two matrices from the third frame to the fourth frame (according to the D-H parameter). From the figure, we can see that about the Z_2 axis; there will be a rotation θ_3. For that, we will have a rotational matrix about the Z_3 axis.

$$R_z = \begin{bmatrix} \cos(\theta_1) & -\sin(\theta_1) & 0 & 0 \\ \sin(\theta_1) & \cos(\theta_1) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

We will also have a translational matrix along the X_3 axis.

$$T_x = \begin{bmatrix} 1 & 0 & 0 & -L_3 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

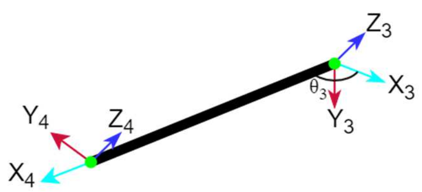


Fig. 5. Transforming 3rd frame to 4th frame. This figure is used to find out the rotational and translation matrix for frame 3 to frame 4 where, X_3, Y_3, Z_3 iactionme 3 and X_4, Y_4, Z_4 frame 4.

Now the transformation matrix from the third frame to the fourth frame will be $T_3^4 = R_z * T_x$

$$T_3^4 = \begin{bmatrix} \cos(\theta_3) & -\sin(\theta_3) & 0 & -L_3\cos(\theta_3) \\ \sin(\theta_3) & \cos(\theta_3) & 0 & -L_3\sin(\theta_3) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Now the final transformation matrix from 0th frame to the fourth frame will be $T_0^4 = T_0^1 * T_1^2 * T_2^3 * T_3^4$

$$T_0^4 = \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{bmatrix}$$

By multiplying each frame of the transformation matrix, we get the final transformation matrix from the 0th frame to the 4th frame. From this transformation matrix, we get the end-effector position of a single leg. By using this end effector value, we can analyze our inverse kinematics for a single leg. In Table I, we organize the full matrix multiplications.

TABLE I. ALL VALUES OF T_0^4 MATRIX AFTER MULTIPLICATION

a_{11}	sin(θ_1) * sin(θ_2) * sin(θ_3) - cos(θ_2) * cos(θ_3) * sin(θ_1)
a_{12}	cos(θ_2) * sin(θ_1) * sin(θ_3) + cos(θ_3) * sin(θ_1) * sin(θ_2)
a_{13}	-cos(θ_1)
a_{14}	L_1*cos(θ_1) + L_2*cos(θ_2) * sin(θ_1) + L_3*cos(θ_2) * cos(θ_3) * sin(θ_1) - L_3 * sin(θ_1) * sin(θ_2) * sin(θ_3)
a_{21}	cos(θ_1) * cos(θ_2) * cos(θ_3) - cos(θ_1) * sin(θ_2) * sin(θ_3)
a_{22}	- cos(θ_1) * cos(θ_2) * sin(θ_3) - cos(θ_1) * cos(θ_3) * sin(θ_2)
a_{23}	-sin(θ_1)
a_{24}	L_1 * sin(θ_1) - L_2 * cos(θ_1) * cos(θ_2) - L_3* cos(θ_1) * cos(θ_2) * cos(θ_3) + L_3 * cos(θ_1) * sin(θ_2) * sin(θ_3)
a_{31}	- cos(θ_2) * sin(θ_3) - cos(θ_3) * sin(θ_2)
a_{32}	sin(θ_2) * sin(θ_3) - cos(θ_2) * cos(θ_3)
a_{33}	0
a_{34}	L_2 * sin(θ_2) + L_3 * cos(θ_2) * sin(θ_3) + L_3 * cos(θ_3) * sin(θ_2)
a_{41}	0
a_{42}	0
a_{43}	0
a_{44}	1

From the DH Parameter table, we can also find out the transformation matrix. DH parameter works on x and z-axis only. In this table, α_{i-1} and a_{i-1} represent the angle of z_{i-1} to z_i about x_{i-1} axis and frame length from z_{i-1} to z_i along the x_{i-1} axis respectively. Also, d_i and θ_i represent the frame offset that is measured from x_{i-1} to x_i along z_i and angle from x_{i-1} to x_i about the z_i axis respectively. Table II contains the DH parameters of the robot leg.

TABLE II. DH PARAMETER

The frame (i)	a_{i-1}	α_{i-1}	d_i	θ_i
0-1	L_1	0	0	θ_1
1-2	0	$-\pi/2$	0	$\pi/2$
2-3	L_2	0	0	θ_2
3-4	L_3	0	0	θ_3

In table III there is a list of parameters that are used to find the forward and inverse kinematics equations.

TABLE III. LIST OF PARAMETERS

The Length of Coxa	$L_1 = 20$mm
The length of the Femur	L2 = 100mm
The length of Tibia	L3 = 117mm
The coordinate system of leg	X_0, Y_0, Z_0
The coordinate system of Coxa	X_1, Y_1, Z_1
The coordinate system of Femur	X_2, Y_2, Z_2
The coordinate system of Tibia	X_3, Y_3, Z_3
The coordinate system of end-effector	X_4, Y_4, Z_4
The Angle of Coxa Joint	θ_1
The Angle of Femur Joint	θ_2
The Angle of Tibia Joint	θ_3

C. *Inverse Kinematics Analysis*

After finishing forward kinematics, it is very much important to find the inverse kinematics to get the desired joint angles. From forward kinematics, we have found the end effectors of the leg by giving the angle values. In inverse kinematics, we use the end effectors value to determine the joint angles.

In figure 6, we have found the θ_1 by using the front view of our leg. From figure 6, we can see two right-angle triangles ABC & ADC from which we have derived our desired equation for θ_1.

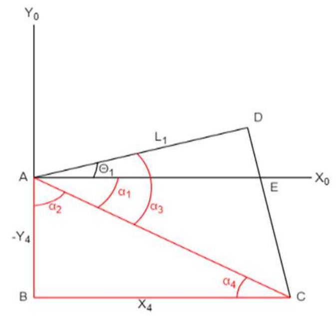


Fig. 6. Derivation of θ_1.

$$\theta_1 = \arctan 2\left(\left(\sqrt{x_4^2 + y_4^2 - L_1^2}\right), L_1\right) - \arctan 2(-y^4, x^4) \quad (1)$$

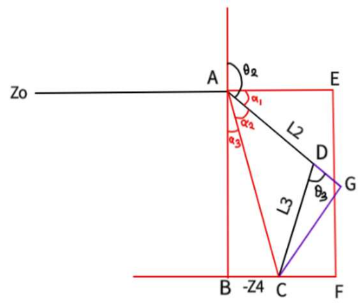


Fig. 7. Derivation of θ_2.

In this figure 7, we have found θ_2 by using the side view of our leg. From the figure 7, we can see two right-angle triangles ABC, CDG and one obtuse triangle ADC from which we have determined our desired equation for θ_2.

$$\theta_2 = -arctan2((L_3 \sin(\theta_3)), (L_2 + L_3 \cos(\theta_3))) - arctan2\left(-z_4, \sqrt{x_4^2 + y_4^2 - L_1^2}\right) \quad (2)$$

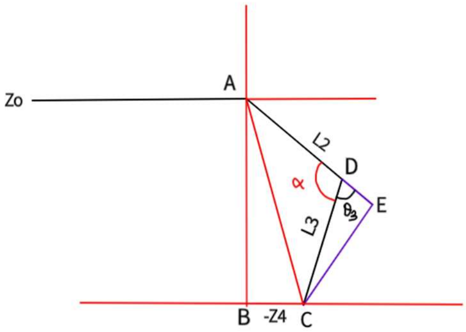


Fig. 8. Derivation of θ_3.

In this figure, we have found θ_2 by using the side view of our leg. From the figure, we can see one right angle triangle ABC and one obtuse triangle ADC from which we have determined our desired equation for θ_3.

$$\theta_3 = \pi - \arccos\left(\frac{L_2^2+L_3^2-x_4^2-y_4^2+L_1^2-Z_4^2}{2L_2L_3}\right) \quad (3)$$

III. Simulation & Experimental Result

A. Verification of co-relation between forward and inverse kinematics

After deriving the Forward and inverse kinematics, it is time to verify their co-relation. For this verification, we used mathematical coding and analyzed the robot's angle and end-effector position using the Pybullet physics engine. We have taken some pre-defined angle values in radian for forward kinematics for this verification. We are taking $\theta_1 = 0.2rad$, $\theta_2 = 0.9rad$, $\theta_3 = 0.4rad$. After putting this value on the end-effector.

- $x_4 = a_{14} = L_1*\cos(\theta_1) + L_2*\cos(\theta_2) * \sin(\theta_1) + L_3*\cos(\theta_2) * \cos(\theta_3) * \sin(\theta_1) - L_3 * \sin(\theta_1) * \sin(\theta_2) * \sin(\theta_3)$
- $y_4 = a_{24} = L_1 * \sin(\theta_1) - L_2 * \cos(\theta_1) * \cos(\theta_2) - L3* \cos(\theta_1) * \cos(\theta_2) * \cos(\theta_3) + L3 * \cos(\theta_1) * \sin(\theta_2) * \sin(\theta_3)$
- $z_4 = a_{34} = L_2 * \sin(\theta_2) + L_3 * \cos(\theta_2) * \sin(\theta_3) + L_3 * \cos(\theta_3) * \sin(\theta_2)$

We got $x_4 = 0.0381$, $y_4 = -0.0876$, $z_4 = 0.1910$. Now for verification of Inverse kinematics' we are going to put these values on equation 1, 2 and 3. After computing Inverse kinematics we got $\theta_1 = 0.199$, $\theta_2 = 0.89$, $\theta_3 = 0.40$. Comparing both side value, it seems similar and accurate. Now we can apply our equation to real robots for manipulating their links and joints. we can locate the foot position in 3D space.

B. Pybullet implement analysis

We will create a reference position of our robot leg where; our robot's leg will get some offset value compared to figure 1. Coxa has 0 rad, the femur has 2.0 rad, and the tibia has 0.52 rad offset comparing figure 1. Table IV contains the information on simulation and evaluation.

TABLE IV. Simulated Result in different angles.

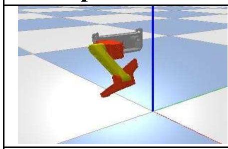
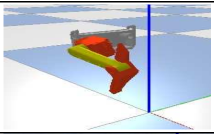
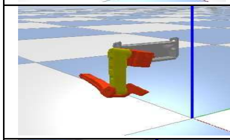
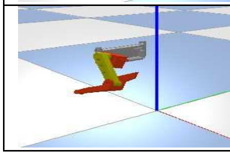
Links & Joints position	End effector	Angles
	$X_4 = 0.02$ $Y_4 = -0.217$ $Z_4 = 0$	$\theta_1 = 0$ rad $\theta_2 = 0$ rad $\theta_3 = 0$ rad
	$X_4 = 0.07$ $Y_4 = -0.17$ $Z_4 = 0.10$	$\theta_1 = 0.3$rad $\theta_2 = 0.4$ rad $\theta_3 = 0.2$ rad
	$X_4 = -0.03$ $Y_4 = -0.17$ $Z_4 = -0.12$	$\theta_1 = -0.3$rad $\theta_2 = -0.7$ rad $\theta_3 = 0.2$ rad
	$X_4 = -0.06$ $Y_4 = -0.20$ $Z_4 = -0.02$	$\theta_1 = -0.4$rad $\theta_2 = -0.15$ rad $\theta_3 = 0.03$ rad

Fig. 9. Solid works model. This model is used to create URDF mesh files in Pybullet physics engine.

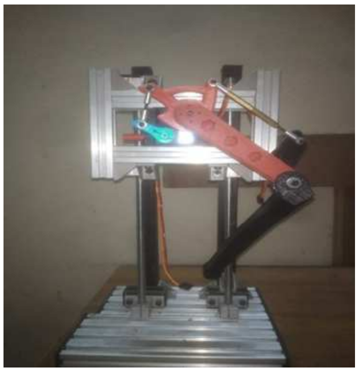


Fig. 10. Real 3D printed model for Future Testing purpose.

Following figure 9 is the SolidWorks model. We use this model to create a universal robot description format for the pybullet physics engine. We are working on full quadruped robots. Figure 10 is a real 3d printed model for advance study.

IV. Discussion

We have derived the forward kinematics and inverse kinematics for a single leg of a quadruped robot in kinematic analysis. For forward kinematic analysis, we have developed a diagram (Figure 1) of a single leg of a quadruped robot, where we have the length of coxa, femur, and tibia as translation parameters and joint angles as rotational parameters. The diagram shows four different frames, such as the 0^{th} frame, first frame, the second frame, and third frame, from which we have derived four different transformation matrices depending on the rotation and translation of each frame. We used DH parameters for each frame. After getting transformation matrices for all joints, we derived the final matrix with the matrix multiplication method. We have obtained the end effector values for a single leg from the final transformation matrix.

After completing the forward kinematic analysis, we derived the inverse kinematic equation for the joint angles θ_1, θ_2, θ_3 of a single leg of the quadruped robot. For θ_1, we developed a diagram (figure 6) using our quadruped robot's front view. We used the Pythagorean theorem and trigonometric functions for the right triangle to derive the equation for $\theta 1$. For θ_2, we developed another diagram (figure 7) from the side view of our quadruped robot. Here we implemented the projection value from the diagram of θ_1 and used the trigonometric functions and cosine formula to derive the equation for θ_2. For θ_3, we also developed a diagram (figure 8) from the side view of our quadruped robot. We used the Pythagorean theorem and cosine formula to derive the equation for θ_3.

After completing both forward and inverse kinematic analysis, we verified the correlation between forward and inverse kinematic equations. We used mathematical coding in python and Pybullet physics engines to visually observe the joint angles and end-effectors positions. We developed a forward and inverse kinematic algorithm where we can put many different angles, verified by the inverse kinematic equations, and gives us the same angle values we input previously. Now we can take the end effector wherever we want by using this algorithm.

V. Conclusion

In the development of a quadruped robot, the most important thing is the kinematic analysis of its motion. We have worked with a single leg of our quadruped robot in this study. We have developed two parts of the kinematic analysis, 1. forward kinematic analysis and 2. inverse kinematic analysis. In forward kinematic analysis, our main goal was to determine the end effector values of a single leg. To meet our goal, we have developed our transformation matrices using DH parameters according to our leg position and orientation. Then, through matrix multiplication, we have determined the end effector values of a single leg. After completion of our forward kinematics, we developed the inverse kinematic equations for each joint angle of a single leg. In inverse kinematics, our main concern was to get desired joint angle by using the end effector values that we got from our forward kinematic analysis. After deriving the forward and inverse kinematics, we verified their correlation. In this verification procedure, we used mathematical coding in python and Pybullet physics engines. We developed an algorithm in python where we input joint angle values to get the end effector values, which we use in inverse kinematic equations to get the desired joint angles that match with our forward kinematic analysis. Currently, the quadruped robot is one of the most advanced robots in the robotics industry. The main advantage of a quadruped robot over a conventional wheel robot is its ability to overcome complex terrain. To meet our goal in this paper, we derived our kinematic equation for our quadruped robot, which defines the position, orientation, and movement of the quadruped robot.

References

[1] Sarmah, A. N., Boruah, A., Kalita, D., Neog, D., & Paul, S. (2018). A bio-inspired implementation of walking and stair climbing on a quadruped robot. *Procedia computer science*, *143*, 671-677.

[2] Liu, Chengju, Qijun Chen, and Danwei Wang. "CPG-inspired workspace trajectory generation and adaptive locomotion control for quadruped robots." *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)* 41.3 (2011): 867-880.

[3] Chen, Xuedong, et al. "Translational crawl and path tracking of a quadruped robot." *Journal of Robotic Systems* 19.12 (2002): 569-584.

[4] Chen, X., Watanabe, K., Kiguchi, K., & Izumi, K. (2000). Real-time kinematics on the translational crawl motion of a quadruped robot. *Journal of Intelligent and Robotic Systems*, *29*(2), 111-131.

[5] Kucuk, S., & Bingul, Z. (2006). *Robot kinematics: Forward and inverse kinematics*. INTECH Open Access Publisher.

[6] Sen, M. A., Bakircioglu, V., & Kalyoncu, M. (2017). Inverse kinematic analysis of a quadruped robot. *International journal of scientific & technology research*, *6*(9), 285-289.

[7] Rong, X. W., Li, Y. B., Ruan, J. H., & Song, H. J. (2010). Kinematics analysis and simulation of a quadruped robot. In *Applied Mechanics and Materials* (Vol. 26, pp. 517-522). Trans Tech Publications Ltd.

[8] Zeng, Y., Li, J., Yang, S. X., & Ren, E. (2018). A bio-inspired control strategy for locomotion of a quadruped robot. *Applied Sciences*, *8*(1), 56.

[9] Park, H., Kwak, B., & Bae, J. (2018). Inverse kinematics analysis and COG trajectory planning algorithms for stable walking of a quadruped robot with redundant DOFs. *Journal of Bionic Engineering*, *15*(4), 610-622.

[10] Li, Junmin, et al. "Gait planning and stability control of a quadruped robot." *Computational intelligence and neuroscience* 2016 (2016).

[11] Zhang, Shuaishuai, et al. "Generation of a continuous free gait for quadruped robot over rough terrains." *Advanced Robotics* 33.2 (2019): 74-89.

[12] Zhang, Jiaqi, et al. "Trot gait design and CPG method for a quadruped robot." *Journal of Bionic Engineering* 11.1 (2014): 18-25.

[13] Chen, X., Gao, F., Qi, C., Tian, X., & Wei, L. (2015). Kinematic analysis and motion planning of a quadruped robot with partially faulty actuators. *Mechanism and Machine Theory*, *94*, 64-79.

[14] Liegeois, A. (1977). Automatic supervisory control of the configuration and behavior of multibody mechanisms. *IEEE transactions on systems, man, and cybernetics*, *7*(12), 868-871.

[15] Chan, T. F., & Dubey, R. V. (1995). A weighted least-norm solution-based scheme for avoiding joint limits for redundant joint manipulators. *IEEE Transactions on Robotics and Automation*, *11*(2), 286-292.

[16] Baillieul, J. (1985, March). Kinematic programming alternatives for redundant manipulators. In *Proceedings. 1985 IEEE International Conference on Robotics and Automation* (Vol. 2, pp. 722-728). IEEE.

[17] RunBin, C., YangZheng, C., Lin, L., Jian, W., & Xu, M. H. (2013). Inverse kinematics of a new quadruped robot control method. *International Journal of advanced robotic systems*, *10*(1), 46.

[18] Son, D., Jeon, D., Nam, W. C., Chang, D., Seo, T., & Kim, J. (2010). Gait planning based on kinematics for a quadruped gecko model with redundancy. *Robotics and Autonomous Systems*, *58*(5), 648-656.

[19] Farshidian, F., Jelavić, E., Winkler, A. W., & Buchli, J. (2017, September). Robust whole-body motion control of legged robots. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (pp. 4589-4596). IEEE.

[20] Chakraborty, S., & Aithal, P. S. (2021). Forward and Inverse Kinematics Demonstration using RoboDK and C. *International Journal of Applied Engineering and Management Letters (IJAEML)*, *5*(1), 97-105.