iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 8 **Issue:** IV **Month of publication:** April 2020

DOI:

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.429
Volume 8 Issue IV Apr 2020- Available at www.ijraset.com

Simulation of CANINE: An Autonomous Quadruped Walking Robot in MATLAB – Simscape Multibody

Mrs. Priyanka Deshmukh[1], Mr. Sumit Kumbhar[2], Mr. Prathamesh Kadam[3], Mr. Satyam Koul[4], Mr. Mohammad Javed Khan[5], Mr. Imran Khan[6]

[1]*Assistant Professor, Dept. of Mechanical Engineering, MCT's Rajiv Gandhi Institute of Technology, Mumbai, Maharashtra, India.*
[2,3,4,5]*B.E. Student, Dept. of Mechanical Engineering, MCT's Rajiv Gandhi Institute of Technology, Mumbai, Maharashtra, India.*
[6]*B.E. Student, Dept. of Computer Engineering, MCT's Rajiv Gandhi Institute of Technology, Mumbai, Maharashtra, India.*

Abstract: ***With the emerging technology, it has been made practically possible to create and analyze any product or process virtually, without actually manufacturing or performing it in the real-life replicating scenarios. Simulation is an approximate imitation of the operation of a process or system, as it may behave in the real world. To achieve this, different variables are used to imitate various factors that may influence the system practically. The paper describes the simulation of CANINE: An Autonomous Quadruped Walking Robot in MATLAB Simscape. The gait mechanism of the quadruped robot is simulated in Simscape Multibody.***

Keywords: Autonomous, Simulation, Quadruped Walking Robot, MATLAB, Inverse Kinematics, Gait, CAD Model, Joints.

I. INTRODUCTION

Simulation is done for analyzing & visualizing the outputs or performance of any specific machine. For doing so, we need a well-defined model of that particular system which represents its key characteristics, such as its behavior, functions and abstract or physical properties. The model represents the system itself, whereas the simulation represents its operation over time. Simulation can be used to show the eventual real effects of alternative conditions and courses of action. It's also used when the real system cannot be engaged, because it may not be accessible, or it may be dangerous or unacceptable to engage, or it is being designed but not built, or it may simply not exist.

Fig. 1. CANINE: An Autonomous Quadruped Walking Robot

The robot has to perform certain functions which include scanning its surrounding space and looking for a vacant region where it can proceed to walk. While walking forward it continues to scan its front for any obstacles at a particular distance. If it detects any obstruction in front of it, then it checks its left and right side for the same. If either of it is free, then it turns and starts moving in that particular direction. If there are obstructions in both the sides as well, then the Robot will start walking backwards while continuously scanning both left and right sides. While moving backwards, if it detects that any of the sides are free, then it turns and starts moving forward in that particular side. Thus, the Robot can autonomously sense the surroundings and can traverse on its own without colliding with any of the nearby obstacles.

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.429
Volume 8 Issue IV Apr 2020- Available at www.ijraset.com

II. SIMULATION

The walking and turning motions of the Robot are simulated in Simscape Multibody in MATLAB. The constructed Simscape block diagram resembling the Robot is summarized in Fig. no. 2. For the simulation of the Robot, first the CAD models of the respective components are exported from the modelling software's such as Solidworks, Autodesk Inventor, etc. in the form of '.xml' file format

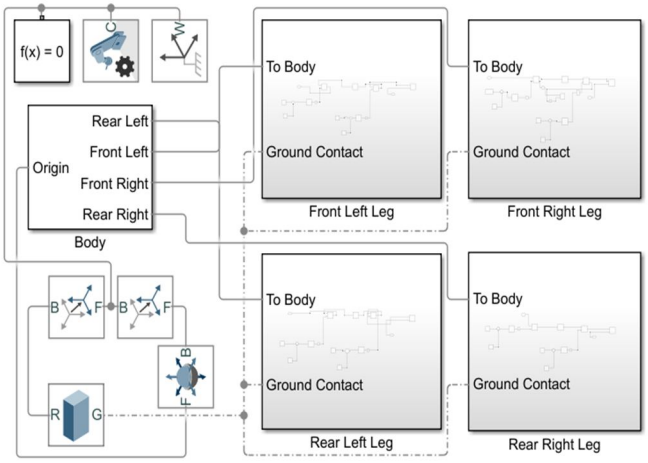


Fig. 2. Simscape model representing the simulation.

The file is then opened in MATLAB-Simscape and the block diagram is generated automatically according to the constraints and assembly of that particular CAD model. The mates and constraints which we define while modelling the part are converted into the corresponding Simscape Multibody blocks which have the same functionality in the MATLAB environment.

Along with that a Datafile is created in '.m' format which consists of the detailed information about the geometry of the CAD model such as its dimensions or its solid properties like weight, material, density, volume, etc.

Another way of importing the CAD models into MATLAB is by including the 'File Solid' block in the Simscape model and then selecting the respective part in 'Step' or 'Stl' file format. But by following this method we need to further define the mates and constraints of various individual imported parts with each other by using various Simscape Multibody blocks.

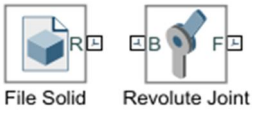


Fig. 3. Simscape blocks representing various joints.

The simulation process starts by connecting the three parent blocks in Simscape viz. Mechanism Configuration, World and Solver Configuration.

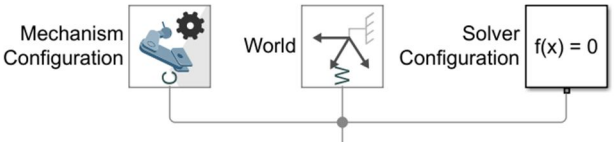


Fig. 4. Simscape parent blocks.

Mechanical Configuration sets mechanical and simulation parameters that apply to an entire machine, the target machine to which the block is connected and also, the uniform gravity for the entire mechanism can be specified. World provides access to the world or ground frame, a unique motionless, orthogonal, right-handed coordinate frame predefined in any mechanical model. World frame is the ground of all frame networks in a mechanical model. Solver Configuration defines solver settings to use for simulation.

The main elements of the simulation like the Body and the Links of the Robot are connected to each other and also to the parent blocks. The frames are created on the simulation elements according to the requirements.

The frames create separate coordinate axes which can be aligned and transformed depending upon the need. A custom frame is fully defined when its origin and axes are too. Of these, the axes require the most care. We must specify two axes, one primary and one secondary. The primary axis defines the plane on which the other axes must lie. The secondary axis is merely the projection of a selected direction axis or geometry feature on that plane.

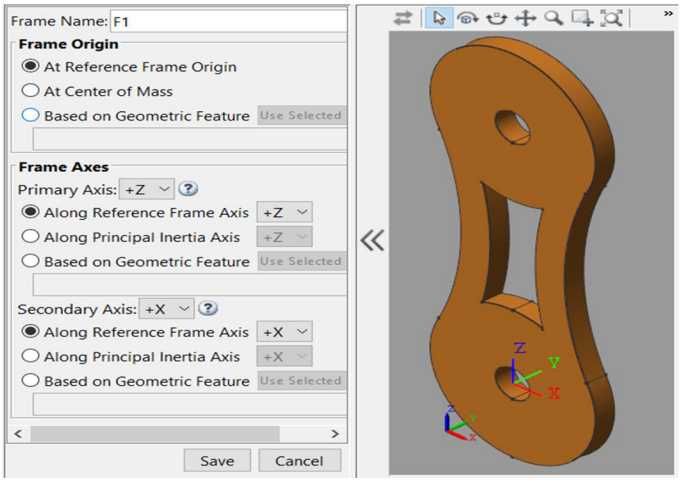


Fig. 5. Frame creation window

The frames are used as a reference while assembling multiple parts, defining various joints or while creating contact surfaces. We can create any number of frames on a particular part.

The Rigid Transform block is used to define a fixed 3D rigid transformation between two frames. Two components independently specify the translational and rotational parts of the transformation. Different translations and rotations can be freely combined. Various contact surface blocks can be included in the Simscape model to define the contact surface between different bodies. The dimensions and properties of the contact surfaces such as the **contact stiffness**, **contact damping** and **friction** can be specified in those blocks.

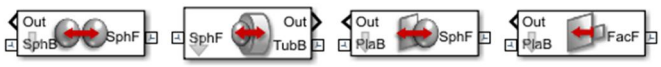


Fig. 6. Simscape blocks representing various contact surfaces

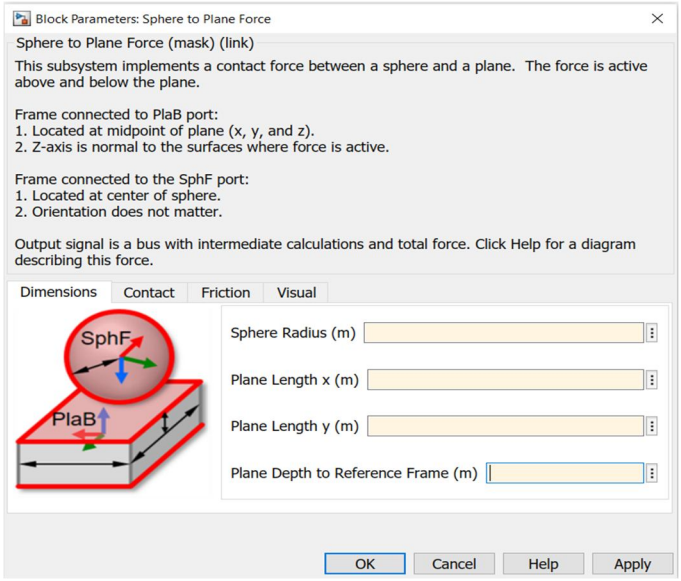


Fig. 7. Contact force block in Simscape

The contact between the end portion of the legs which is going to come in contact with the ground and the ground surface is defined using the Spatial Contact Force block and connecting it to a Cylindrical solid block resembling the contact surface.

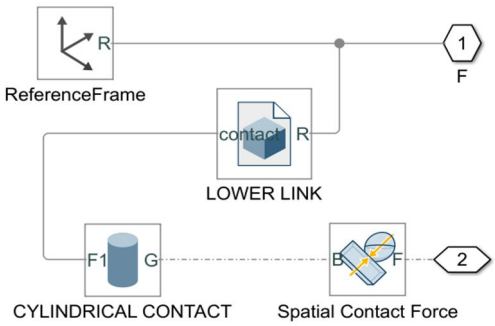


Fig. 8. Simscape block diagram representing the contact.

Defining the required contact surfaces between all the elements along with specifying the contact properties is a very important and crucial aspect of the simulation. And once that's done, the Robot is positioned at the required location and orientation using the Rigid Transform block. Once the placement is completed, the '6-Degrees of Freedom' block is added to the Simscape model. This joint has three translational and three rotational degrees of freedom represented by three prismatic primitives' axes along a set of mutually orthogonal axes, plus a spherical primitive.

This joint allows unconstrained 3-D translation and rotation. The follower origin first translates relative to the base frame. The follower frame then rotates freely, with the follower origin as the pivot. This block enables free movement of the elements in the simulation environment under the natural forces like acceleration due to gravity.

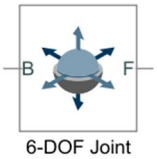


Fig. 9. Simscape block representing the 6-DOF joint.

The Revolute joint is used to define a Pin Joint between two parts. This joint has one rotational degree of freedom represented by one revolute primitive. The joint constrains the origins of the two frames to be coincident and the z-axes of the base and follower frames to be coincident, while the follower x-axis and y-axis can rotate around **the z-axis**. It is used while assembling the links of the legs with the body of the Robot.

III. INPUT

Once the required joints between all the parts of the elements in the simulation are defined the inputs for the actual working of the mechanisms are to be specified. The variables declared in the MATLAB script should be assigned with their respective values.

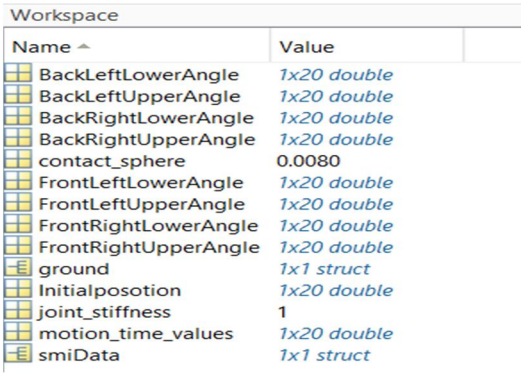


Fig.10. Populated Workspace

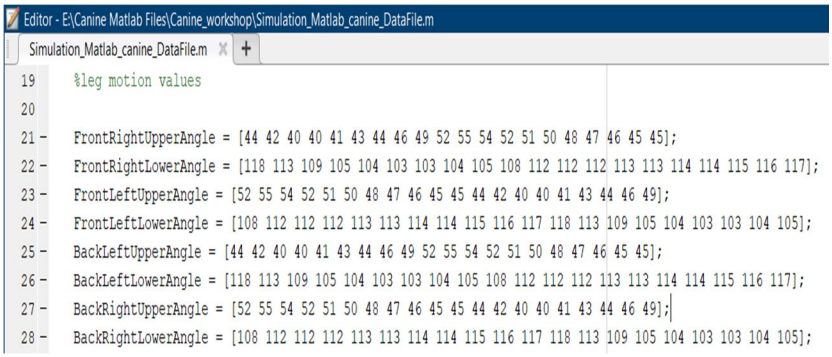


Fig. 11. Data in MATLAB Script.

The MATLAB script i.e. the Datafile is then loaded and the Workspace is then populated with the data given by us in the form of matrices.

The inputs are first converted to Physical Signals by using the **'Simulink-PS Converter'**. Similarly, while recording the outputs in the Scope the output signals are converted to Simulink signals by using **'PS-Simulink Converter'**. Now, the input to the links is given in terms of the angle and time values. Once the values of the angles are determined, they are fed to the respective variables in the MATLAB script. The angles are specified in degrees. The number of angles depends on the level of accuracy. More number of angles would result into more accurate tracing of the trajectory. The angles are determined by using the *'Inverse Kinematics'*

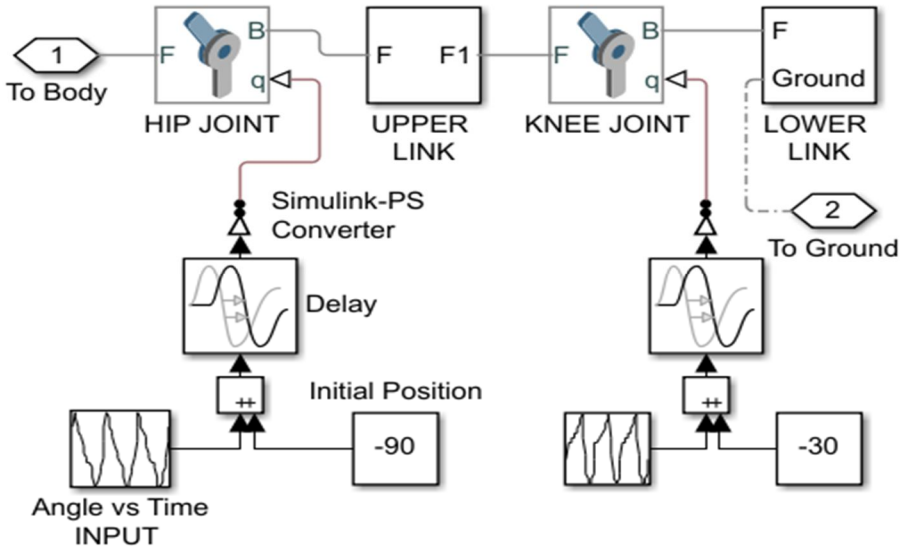


Fig. 12. Simscape diagram representing input structure.

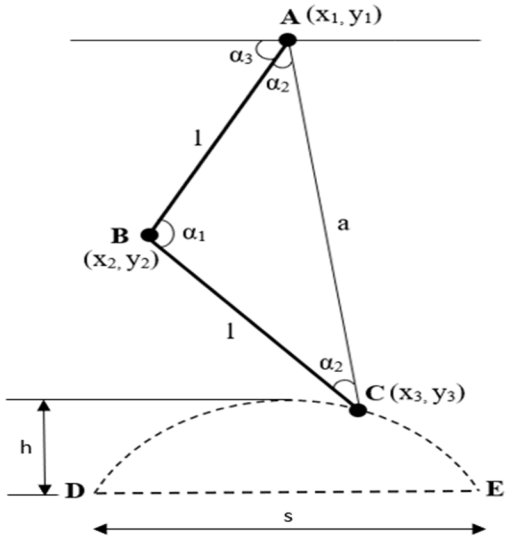
principle. The angles are then combined with the constant initial angle block if required by using the *Sum* block. The constant initial angle is determined on the basis of the initial adjustment of the links with respect to the body. An additional *Delay* block can be provided if necessary. Delay is given in seconds. Then the input signal is given to both the *Revolute Joint* blocks resembling the Hip Joint between the body and the upper link & Knee Joint between the upper link and lower link. The input signal is passed through the *Simulink-PS Converter*.

Now to find out the required angles by Inverse Kinematics, the code is written in the MATLAB script. Input parameters such as the Lift of the legs, Span of the step, Number of points and the coordinates of few fixed points are provided as the input.

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.429
Volume 8 Issue IV Apr 2020- Available at www.ijraset.com

IV. INVERSE KINEMATICS

It is used to calculate the angles through which the upper and lower links are supposed to rotate so that the end of the leg traces the predefined trajectory for walking motion. The dotted figure represents the trajectory.



(Construction: As the length of both the links is equal, join points A and C to form an isosceles triangle ABC)

Fig. 13. Linkage diagram representing links with the trajectory.

We already know the position of point A which is located on the body, so we have $\mathbf{(x_1, y_1)}$. Now the trajectory will be divided into 'n' number of points depending upon the level of accuracy for path tracing. So, as we have a predefined trajectory, we have the coordinates of each point on the trajectory. Thus, we have the location of point C i.e. $\mathbf{(x_3, y_3)}$. We also have to set the span of the step (s) i.e. length DE. This is the distance through which the end of the leg would be in contact with the ground. The lift of the leg (h) is also to be decided. These two factors will together decide the shape of the trajectory. The required angles are determined as follows:

Length of the link = l
Hip Joint = Point A, coordinates (x_1, y_1)
Knee Joint = Point B, coordinates (x_2, y_2)
Leg Contact Point = Point C, coordinates (x_3, y_3)
Angle between Upper & Lower link = α_1
Angle between links and line segment AC = α_2
Angle between Upper link & Body horizontal = α_3
Now, by *Distance formula* we get,

$$a = [(x_1 - x_3)^2 + (y_1 - y_3)^2]^{1/2}$$

By *cosine rule* we have,

$$a^2 = l^2 + l^2 - 2*l*l*\cos \alpha_1$$
$$a^2 = 2l^2 - 2l^2\cos \alpha_1$$
$$2l^2\cos \alpha_1 = 2l^2 - a^2$$
$$\cos \alpha_1 = (2l^2 - a^2)/2l^2$$
$$\alpha_1 = \cos^{-1}[(2l^2 - a^2)/2l^2]$$

We know,

$$\alpha_1 + 2\alpha_2 = 180$$
$$2\alpha_2 = 180 - \alpha_1$$
$$2\alpha_2 = 180 - \cos^{-1}(1 - a^2/2l^2)$$
$$\alpha_2 = 90 - 0.5*\cos^{-1}(1 - a^2/2l^2)$$

Now, the slope of line segment AC is given as

$$m = (y_1 - y_3)/(x_1 - x_3)$$

Also, $m = \tan\theta$

(where θ is the inclination w.r.t. the horizontal)

$\theta = \tan^{-1}m$

$\alpha_3 = \theta - \alpha_2$

$\alpha_3 = \tan^{-1}m - \alpha_2$

$\alpha_3 = \tan^{-}1[(y_1 - y_3)/(x_1 - x_3)] - \alpha_2$

$\alpha_3 = \tan^{-}1[(y_1 - y_3)/(x_1 - x_3)] - 90 - 0.5*\cos^{-1}(1- a^2/2l^2)$

The calculated values of the angles can now be used as input in the simulation process. But the angles alone cannot lead us to the expected motion of the links. The time instances for each of the angle values are also supposed to be given to the system. That is done by using the 'Repeating Table' Simscape block. The output of a repeating sequence are numbers specified in a table of time-value pairs where the values of time should be monotonically increasing.

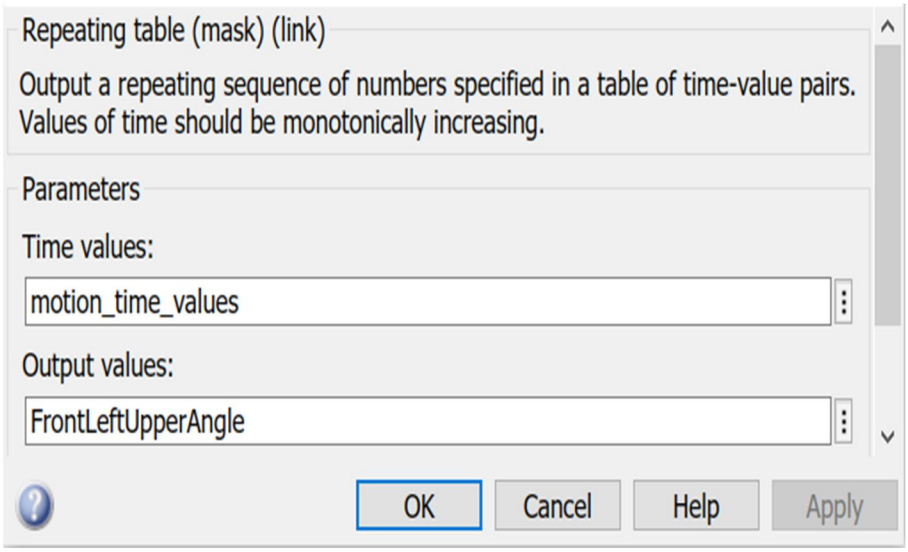


Fig. 14. Repeating Table block in Simscape.

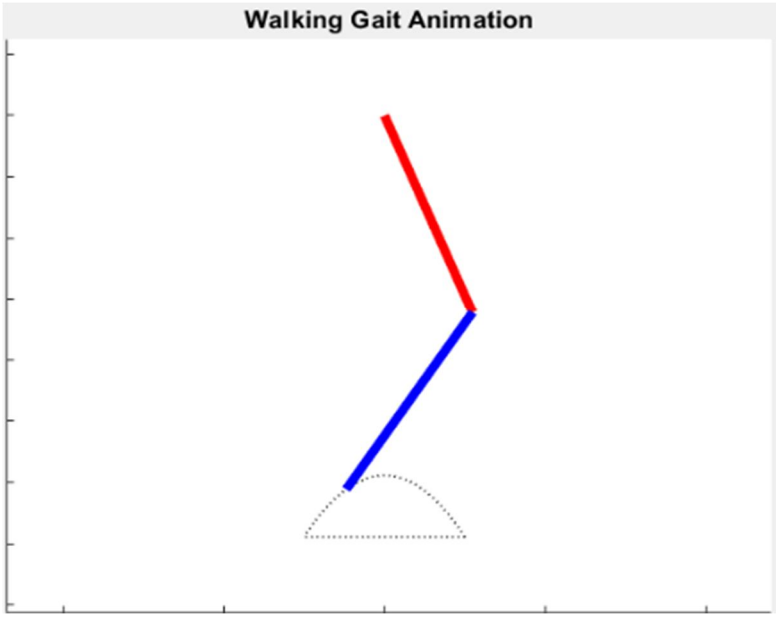


Fig. 15. Links tracing the predefined trajectory.

V. OUTPUT

The output is recorded by using a Transform Sensor block and placing it at the end of the link or the contact point. The transform sensor shows the variation in the position of that point with respect to a fixed point on the body and a graph is plotted representing the trajectory. The visualization of the simulation is done in the *Mechanics Explorer* window. The stability of the walking mechanism can be analyzed by observing the animation and it can be optimized by varying various contact and weight parameters. Output is also taken out from the Revolute joint and graphs are plotted representing the variations in the Hip & Knee Angles with respect to time.

Fig. 16. Mechanics Explorer window in Simscape.

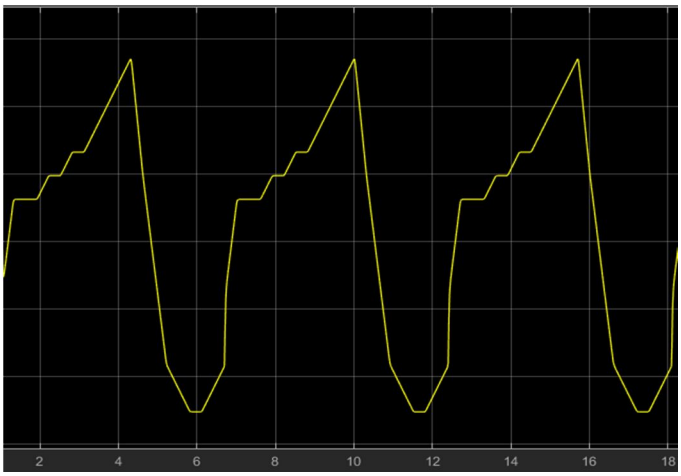


Fig 17. Variation in Hip Angle with respect to time

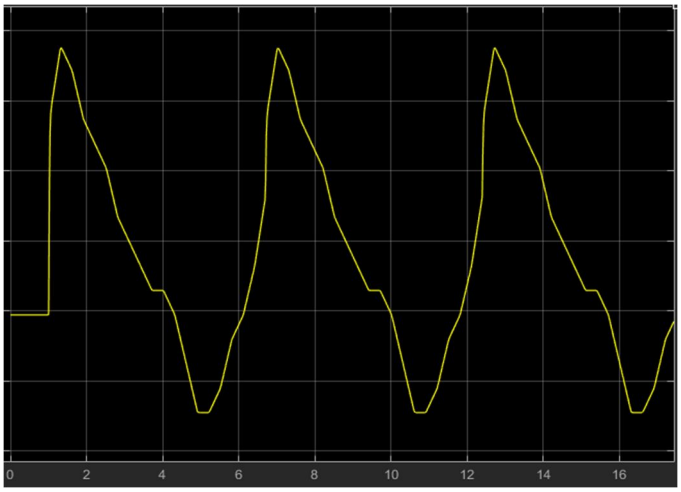
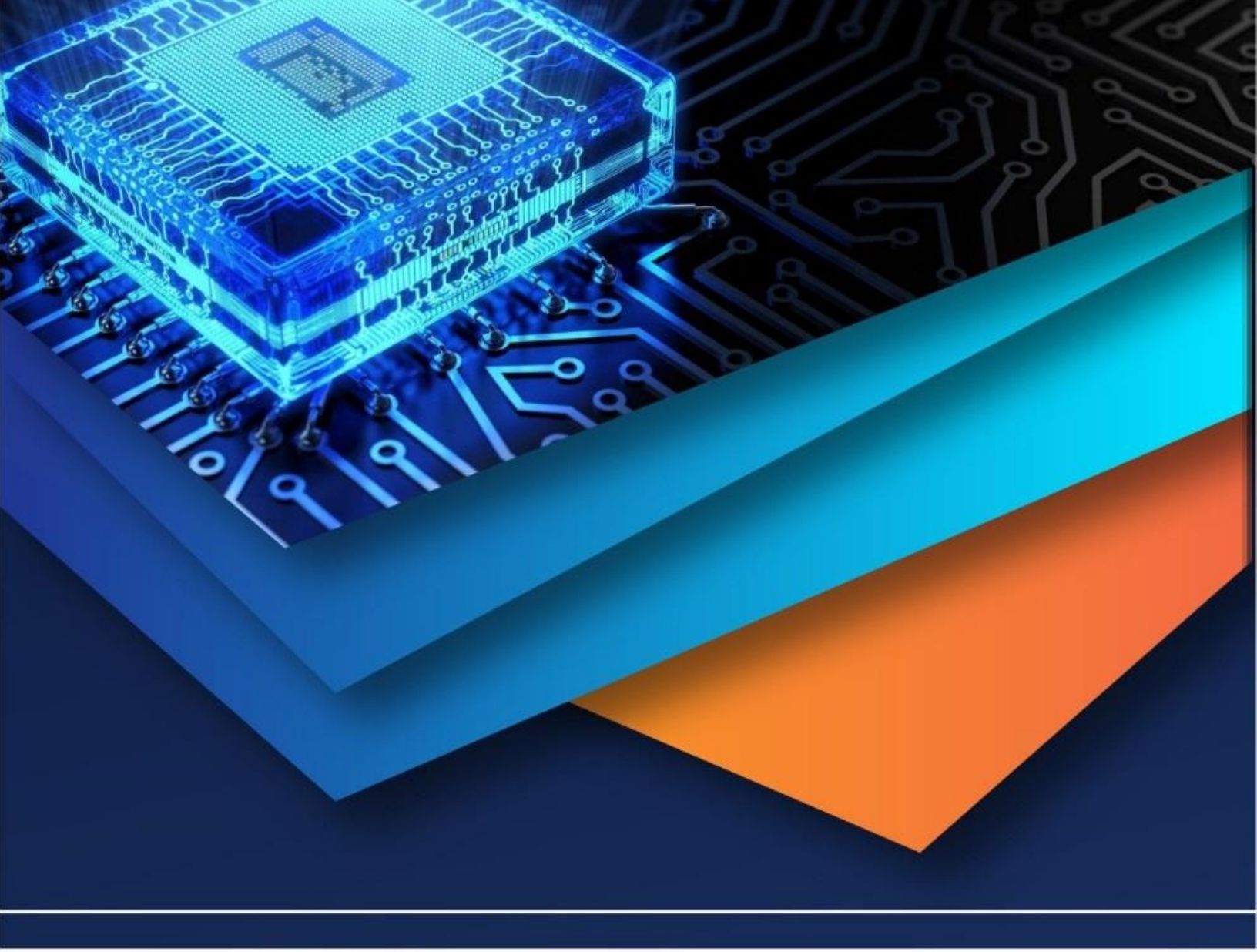


Fig. 18 Variation in knee Angle with respect to time

REFERENCES

[1] A. Frank, "Automatic Control System for Legged Locomotion Machines," USC Rept., p. 273, 1968.
[2] R. B. McGhee and G. I. Iswandhi, "Adaptive Locomotion of a Multilegged Robot Over Rough Terrain," IEEE Trans. Syst., Man, Cybern., vol. SMC-9, no. 4, pp. 176- 182, 1979.
[3] C. A. Klein and R. L. Briggs, "Use of Active Compliance in the Control of Legged Vehicles," IEEE Trans. Sysr., Man, Cybern., vol. SMC-10, no. 7, 1980.
[4] S. Hirose, "A Study of Design and Control of a Quadruped Walking Vehicle." Inr. J. Robotics Res., vol. 3, no. 2, pp. 113-133, summer 1984.
[5] M. Kaneko, M. Abe, and K. Tanie, "A Hexapod Walking Machine with Decoupled Freedoms," IEEE J. Roborics Autom., vol. RA-1, no. 4, pp. 183-190, 1985.
[6] T.-T. Lee and C.-L. Shih, "A Study of the Gait Control of a Quadruped Walking Vehicle," IEEE J. Roborics Autom., vol. RA2, no. 2, pp. 61-69, 1986.
[7] H. Miura, I. Shimoyama, M. Mitsuishi, and H. Kimura, ''Dynamic Walking of Quadruped Robot Proc. 2nd ISRR, pp. 317-324, 1984.
[8] S. Kajita and A. Kobayashi, "Dynamic Walk Control of a Biped Robot with Potential Energy Conserving Orbit," Trans. SICE (in Japanese), vol. 23, no. 3, pp. 75-81, 1987.
[9] K. J. Waldron and R. B. McGhee, "The ISRR, pp. 317-324, 1984. 18 IEEE Control Systems Magazim Adaptive Suspension Vehicle," IEEE Contr. Sysr. Mag., vol. 6, no. 6, pp. 7-12, 1986.
[10] C.-K. Tsai, H.-C. Wong, and D. E. Orin, "Modified Hybrid Control for an Electrohydraulic Robot Leg," IEEE Contr. Syst. Mag., vol. 7, no. 4, pp. 12-18, 1987.



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY