Modelling and Simulation of Quadruped Robot using Fusion 360 and Robotic Operating System (ROS)

[1]PP Jain, [2]HP Patolia, [3]JV Sanandiya

[1]M. Tech (Machine Design) Student, [2]Professor, Mechanical Engineering Department, B.V.M., V.V.Nagar, [3]Assistant Professor, Mechanical Engineering Department, B.V.M., V.V.Nagar
[1]Mechanical Engineering Department,
[1]Birla Vishvakarma Mahavidyalaya, V.V. Nagar, Anand, India.

Abstract : A 12 DoF Quadruped Robot model is made using Fusion 360. The model is then exported into ROS by converting into ROS executable by using Fusion2URDF plugin available. Then later the simulation is done on ROS Melodic running on Ubuntu 18.04LTS. Later, Trot gait is used in simulation. After simulation is done on Rviz the same is being implemented on Servo Motor connected with Arduino and then interfacing it with ROS. Different graphs like joint angle, velocity is being plotted using Plot Juggler.

Index Term-Ros, Quadruped Robot, Fusion 360

I. INTRODUCTION

Quadruped Robots are being a subject of research and are gaining a wide popularity in industries. They have a wide variety of applications like using for inspection in industry, defence, rescue missions and terrain application etc. They have many advantages over wheeled robots. The Kinematics is related to the motion of robot from a reference point to desired position with no consideration of force and or other factors. The Dynamic study of robot gives the joint reaction forces and moments needed to execute a motion. There are different types of joint actuations are used like hydraulic or electric depending upon the application. Electric joints are widely used since they are more accurate and faster to react to the situation and gives a better motion to robot.

II. LITERATURE REVIEW

M. S. Maradkar and P. V. Manivannan have tried to make a dynamic model and simulation of four legged jumping robots with compliant legs. Bounding gait and Sagittal plane have been considered for the simulation. Passive dynamics have been implemented for energy saving. Different state variables have been gained for examination. To control forward velocity control strategies have been implemented. **X. Zeng, S. Zhang, Y. Fu, H. Zhou, X. Li and H. Zhang** have developed a single legged platform for quadruped robots using the motivation of high-speed locomotion. The leg has three joints just like an actual quadruped animal designed to be lightweight and low having a low inertia structure. FEA has been implemented to get the balance between the strength of a robot leg and the load during the locomotion. The gait trajectory implemented is trot gait as an actual quadruped animal uses while traversing at high speed. The walk direction is arranged for swing and position stage; accordingly, the robot can keep its solidness with movable directions while following a particular step design. Particularly for the swing stage, the proposed direction can limit the greatest speed increase of legs and guarantee the congruity of position, speed, and speed increase. At that point, in view of the kinematics investigation, the proposed direction is contrasted and the direction of Bezier curve for the force utilization. **W. Wang, Y. Zhang, Y. Yang and X. Shao** proposed a control method for trajectory planning and posture adjustment for quadruped robot's obstacle striding. The suggested mixed parabola plans of travelling paths over an obstacle in Cartesian Coordinate system. The restraints on accelerations, velocities, and jerks at waypoints are incorporated to generate the time-efficient smooth cubic splices joint trajectories by nonlinear optimization technique. The arranged directions augment the consistence also, adaptability of joint development. To assure the static stability for the pitch-pitch type quadruped robots, a posture adjustment strategy grounded so the potential energy is examined to regulate the trajectory of COG. The experiment resulted the feasibility of the of the planned method. **Z. Shauaishuai, Y. Li, B. Li and X. Rong** proposed COG trajectory generates method and the gait planning method so that the robot can walk through the rough terrain in a condition that the stability margin of the robot is as equal to the minimum stability margin. Gait planning is the primary concern for the quadruped robot especially when it is traversing through rough terrain. The author has focused on the scenario where the robot is traversing using static walking gait pattern. The author has presented a trajectory generator based on COG which has been characterized by continuous velocity and acceleration profiles based on the foot placement of robot. **Y. N. Andrew and Z. J. Kolter** consider the task of planning smooth trajectories for robot motion. They present a method for cubic spline optimization; this technique lets us simultaneously plan optimal task-space trajectories and fit cubic spines to the trajectories, while obeying many of the same constraints imposed by a typical motion planning algorithm. The algorithm uses convex optimization methods to efficiently plan task-space trajectories, while obeying collision constraints, kinematic feasibility and avoiding contact. **S. Liu, S. Yu and H. L** investigated about quadruped robot optimal feet force distribution. They then formulated a reduced dynamic model of motion/force for quadruped robot. Assuming an external wrench on quadruped robot, a neural network is adopted to minimize the optimized objective. **M. S. Maradkar and P. V. Manivannan** have analysed a quadruped robot with nine link closed chain leg mechanism, each leg having two degrees of freedom. Considering all the physical parameters like, height, stroke length etc. a support polygon has been

developed. Based on this data inverse kinematics is done and the data obtained is further used for dynamic analysis. The dynamic results obtained then used for finite element analysis (FEA) of the robot component.

III. METHODOLOGY

The first step in simulation is to make a Quadruped Robot model. The model is made in Fusion 360. The Quadruped Robot has total of 12 Degrees of Freedom, each leg having 3 Degrees of Freedom. The reference dimensions are taken from previously available Robokit and are further tweaked as per the requirement.

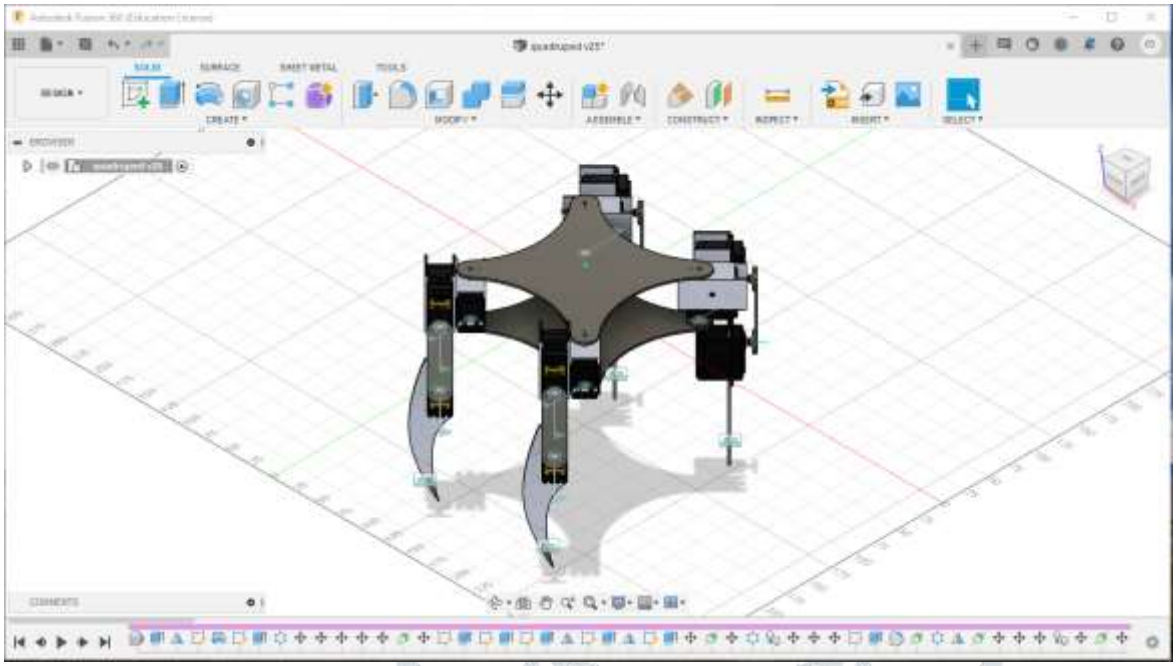

Figure 1 CAD Model in Fusion 360

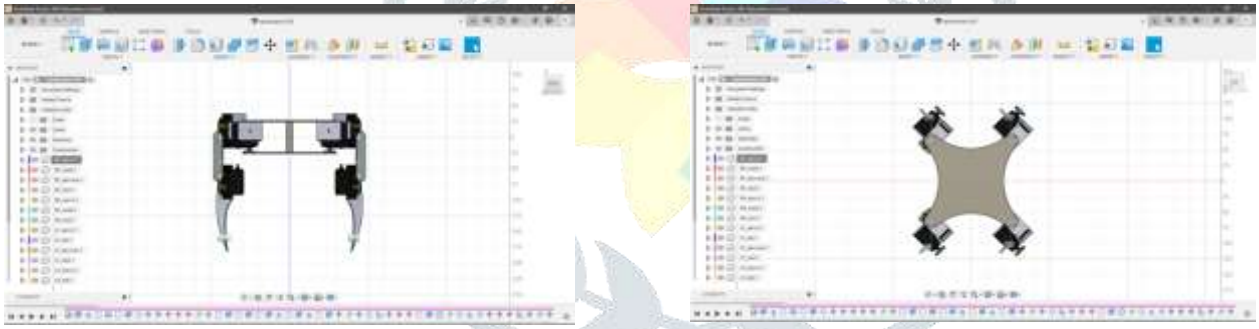

Figure 2 CAD model

The Table 1 shows the DH parameters of the Quadruped Robot.

Table 1 DH parameters

Link i	Link Length a_i (m)	Link Twist α_i (deg)	Joint Offset d_i (m)	Joint Angle θ_i (deg)
1	0.25	90	0.15	θ_1
2	0.25	0	0	θ_2
3	0.15	0	0	θ_3

The 12 Degree of Freedom robot is then exported into ROS executable using a plugin available on GitHub called Fusion2URDF. After making the model of Quadruped Robot we need to click on Add-Ins option then click on Scripts and Add-Ins, then we will be directed to a pop-up and in that window, we need to click on URDF_Exporter.

The Robot Operating System (ROS) is a flexible framework for writing robot software. It is a collection of libraries, tools, and conventions that aim towards making task easy and creating complex and robust behaviour types of robotic platforms. The biggest advantage of ROS is that it is an opensource software and has become the most popular and widely choose software for simulation

of robots. Being an opensource software gives it an advantage of having a biggest community of people working and readily available packages required for simulation of robots.

The Figure 3 Flow ChartFigure 3**Error! Reference source not found.** shows the flow chart of all the command executed in Linux Terminal in order to run the simulation in ROS. Later all the codes are being explained with the outcome after executing the code in Linux Terminal. The manly includes "roslaunch" which is a command used to launch the multiple rosnodes at once compared to "rosrun" which is used to run a node directly which is available inside a package.

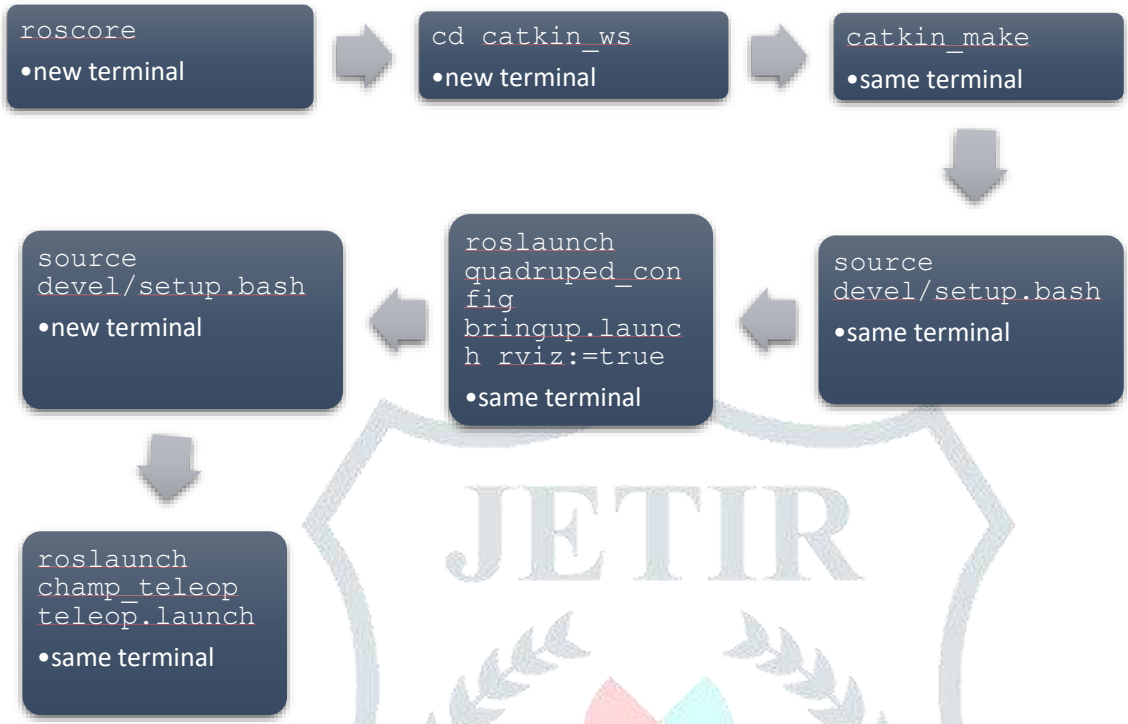

Figure 3 Flow Chart

- **roscore** (new terminal)

This will launch ROS Master which is a server that tracks IP addresses of all active nodes.

Now that we've launched ROS Master we will now open a new tab and then navigate to our ROS workspace. For that we will type the following command. The "cd" command is used to change the directory of the working folder. Here we have directed for home to catkin_ws which is where we have stored all the simulation data and respective launch files.

- **cd catkin_ws** (new terminal)

Before starting any work, we need to compile the whole workspace of ROS and make it executable. For this we will type the following command.

- **catkin_make** (same terminal)

The simulation of Quadruped Robot is done in ROS (Robotic Operating System). The simulator used is called RViz and then the certain Joint Values have been obtained.

After that we need to source our terminal. The reason to source the terminal is it will tell shell where to find the ROS executable program.

- **source devel/setup.bash** (same terminal)

Now to run the simulation the following command is used to run the simulation in Rviz. The command is in four different parts. The first part is roslaunch which will execute multiple nodes, then "quadruped_config" is the package where all the simulation files exist, the then later command "bringup.launch" is the package to be executed, and at last there is "rviz:=true" command is used to launch the simulation in Rviz simulator.

- **roslaunch quadruped_config bringup.launch rviz:=true** (same terminal)

Now a new terminal is opened and is sourced again to make shell find the ROS executable files using the same command

- **source devel/setup.bash** (new terminal)

Now, in new terminal, to move the robot from one point to another we will use a teleop keyboard which sends command to simulator using keyboard

- **roslaunch champ_teleop teleop.launch** (same terminal)

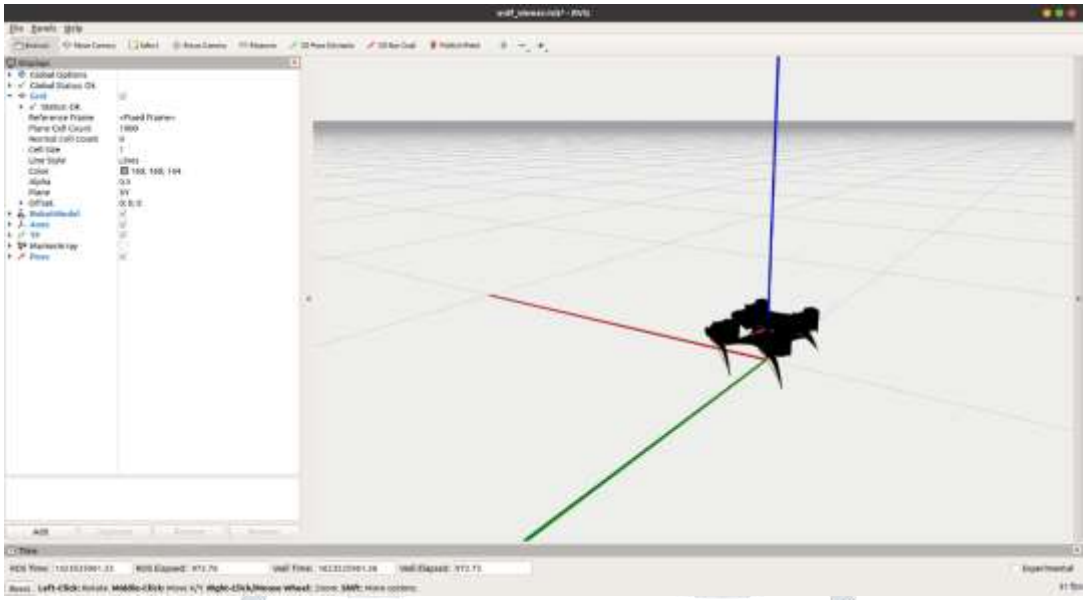

Figure 4 Rviz

After the simulation starts there are different graphs which can be plotted. The following graphs are being plotted using Plot Juggler, a free to use to available on GitHub.

The Figure 5 shows the velocity in Y axis and on X axis we have time series.

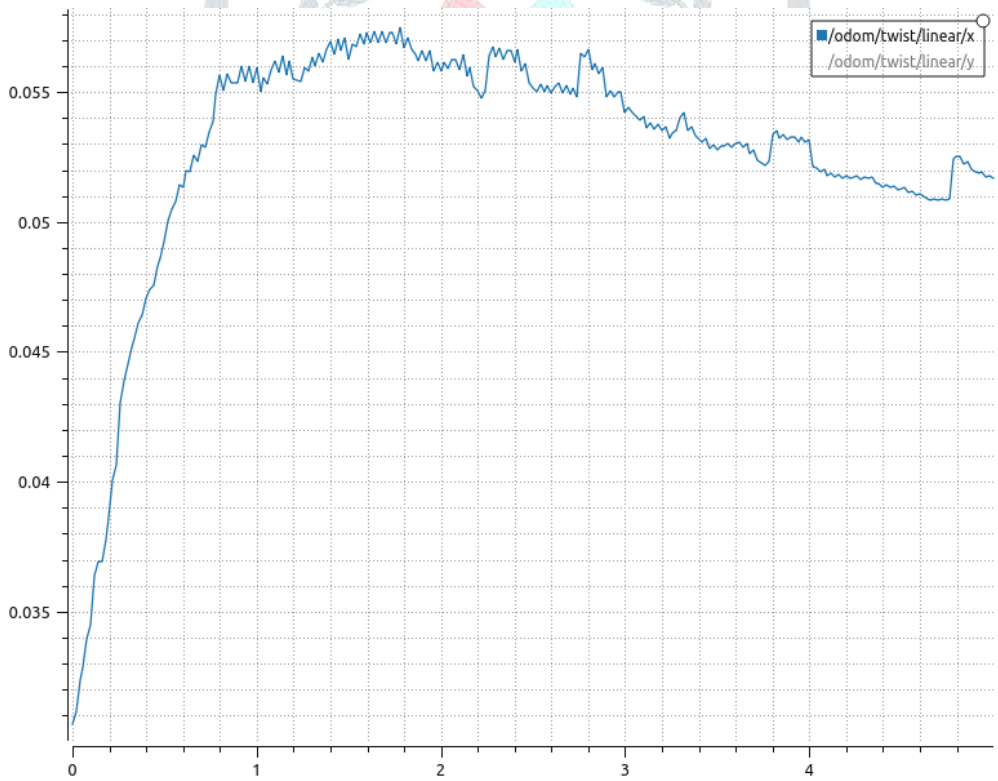

Figure 5 Time v/s Velocity

Now that we have plotted the velocity profile, "plot juggler" also helps in plotting joint state of each joint of the quadruped robot. The values of joint state are in radian shown on Y axis and time stamp on X axis. Below shows figures are the joint state of each joint.

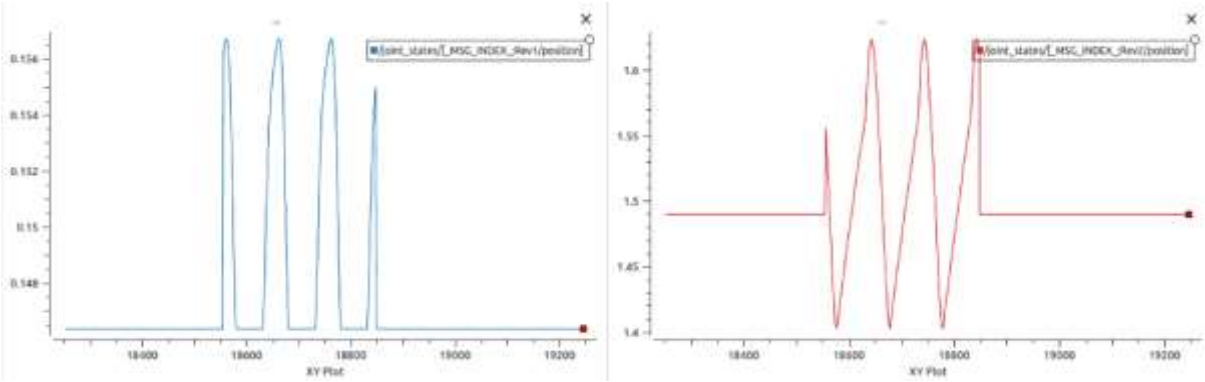

Figure 6 Joint State 1

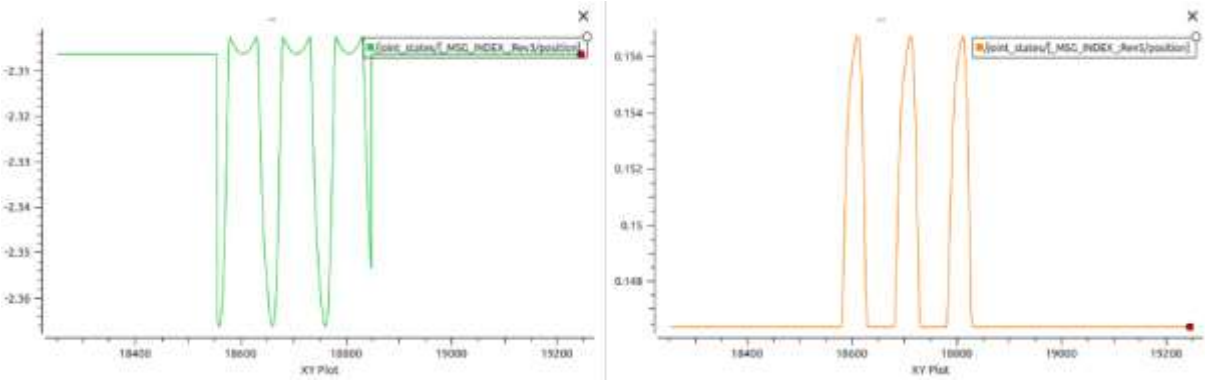

Figure 7 Joint State 2

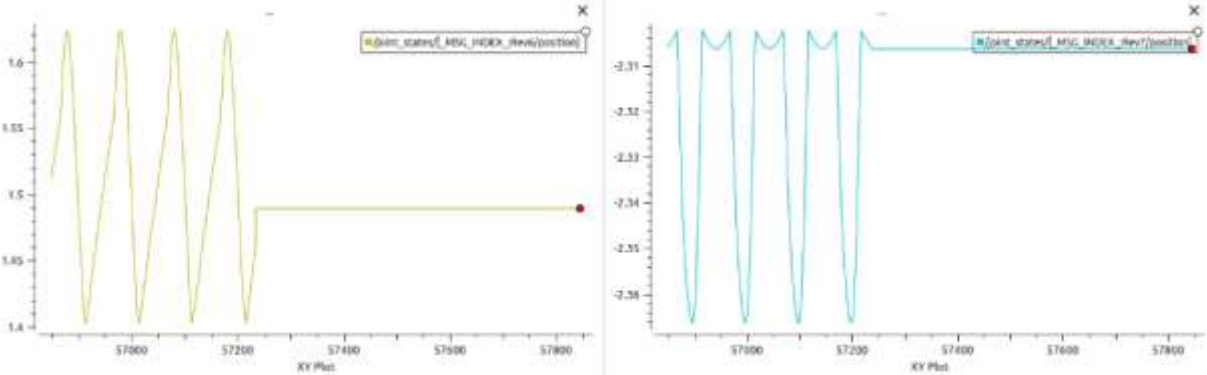

Figure 8 Joint State 3

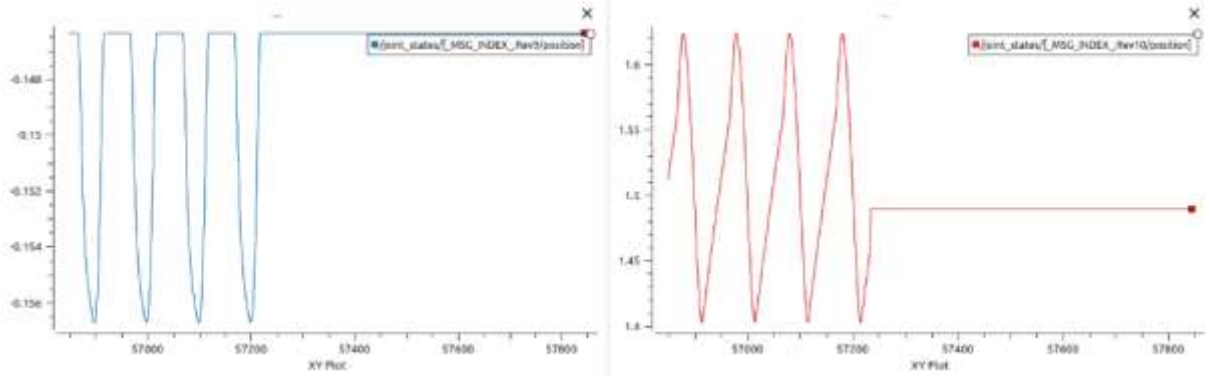

Figure 9 Joint State 4

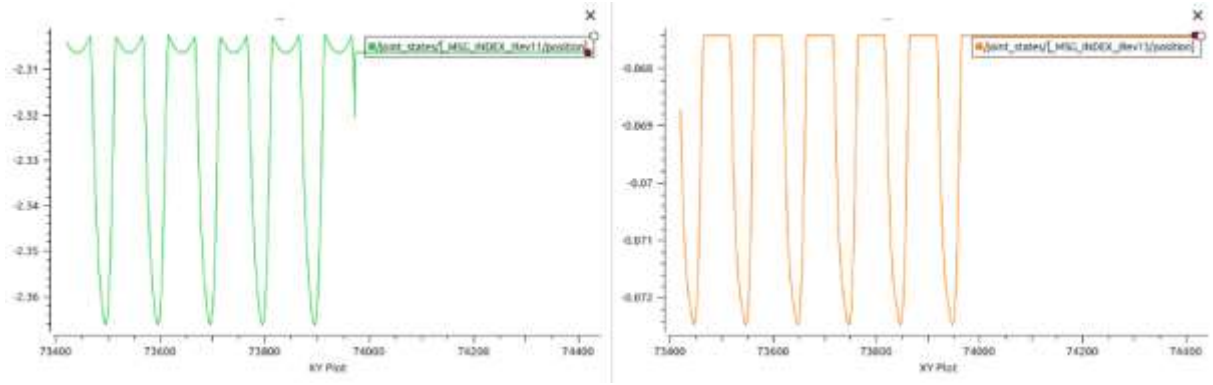

Figure 10 Joint State 5

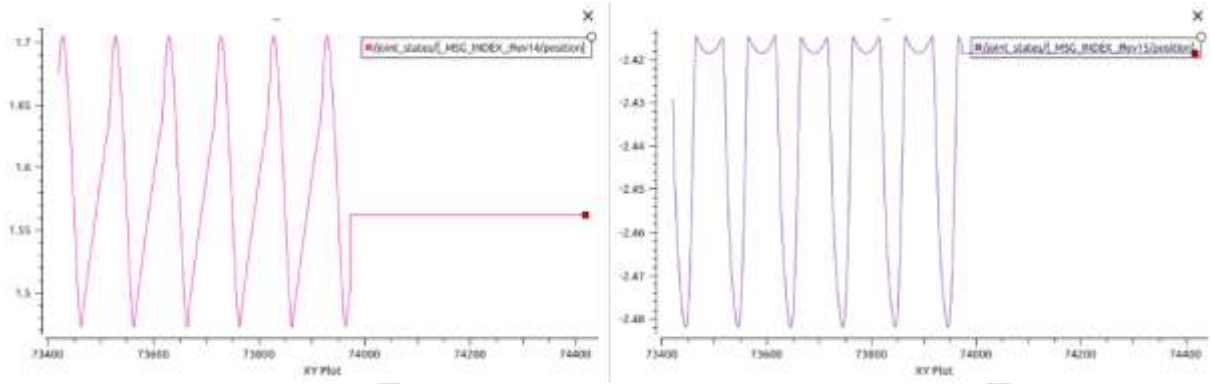

Figure 11 Joint State 6

Plog Juggler also helps in plotting the global position of robot in coordinate system in X axis measuring axis and Y axis having a time stamp. The following Figure 12Figure 12 shows the distance in X direction and the Figure 13Figure 13 shows the distance in Y direction.

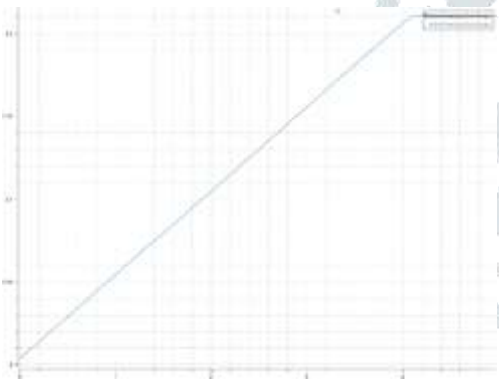

Figure 12 Global Coordinate X

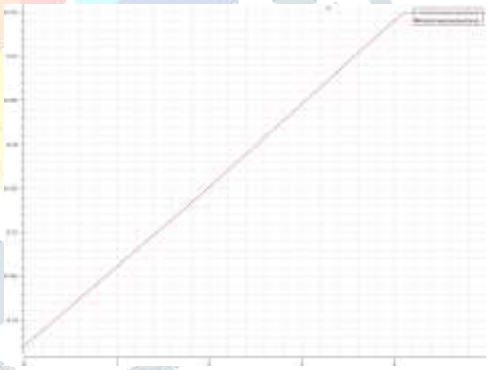

Figure 13 Global Coordinate Y

The below shown Figure 14**Error! Reference source not found.** depicts the curvature traced by the Quadruped Robot when moving in a circular direction. The motion is captured when the robot is given a command to traverse in a right direction using keyboard.

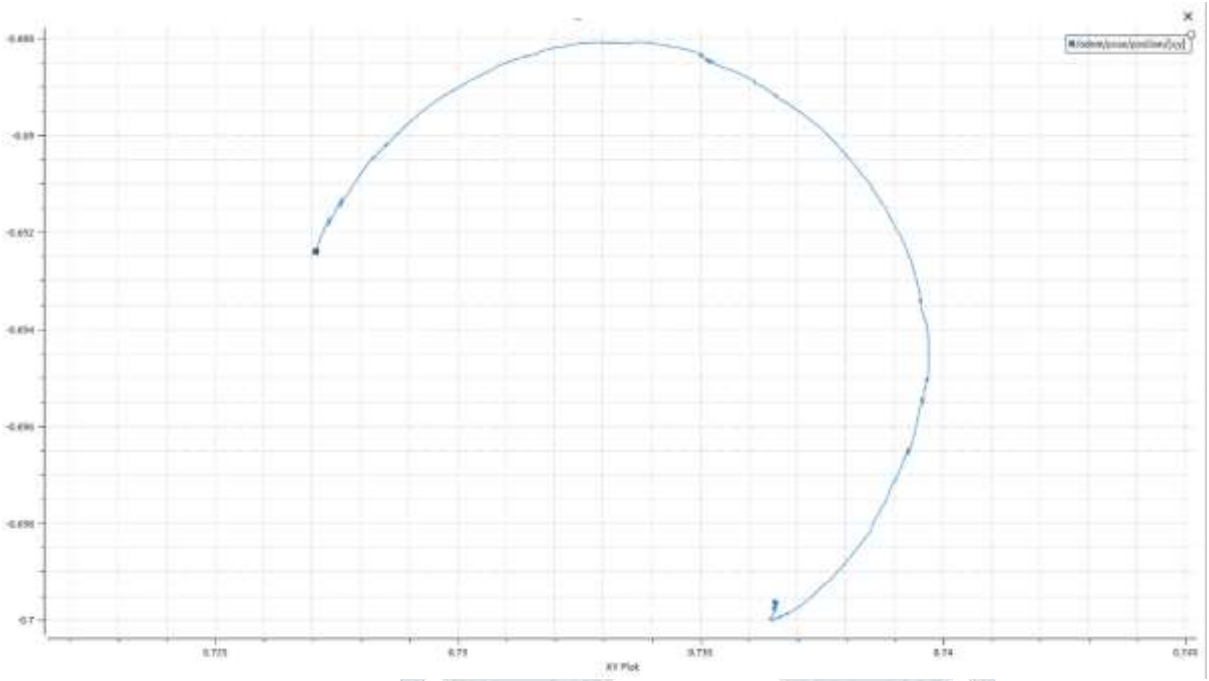

Figure 14 Circular Path

When the simulation is in process, we also need a trajectory to be followed by the leg of Quadruped Robot. The following Figure 15 show the “D” shaped trajectory followed by the Quadruped Robot.

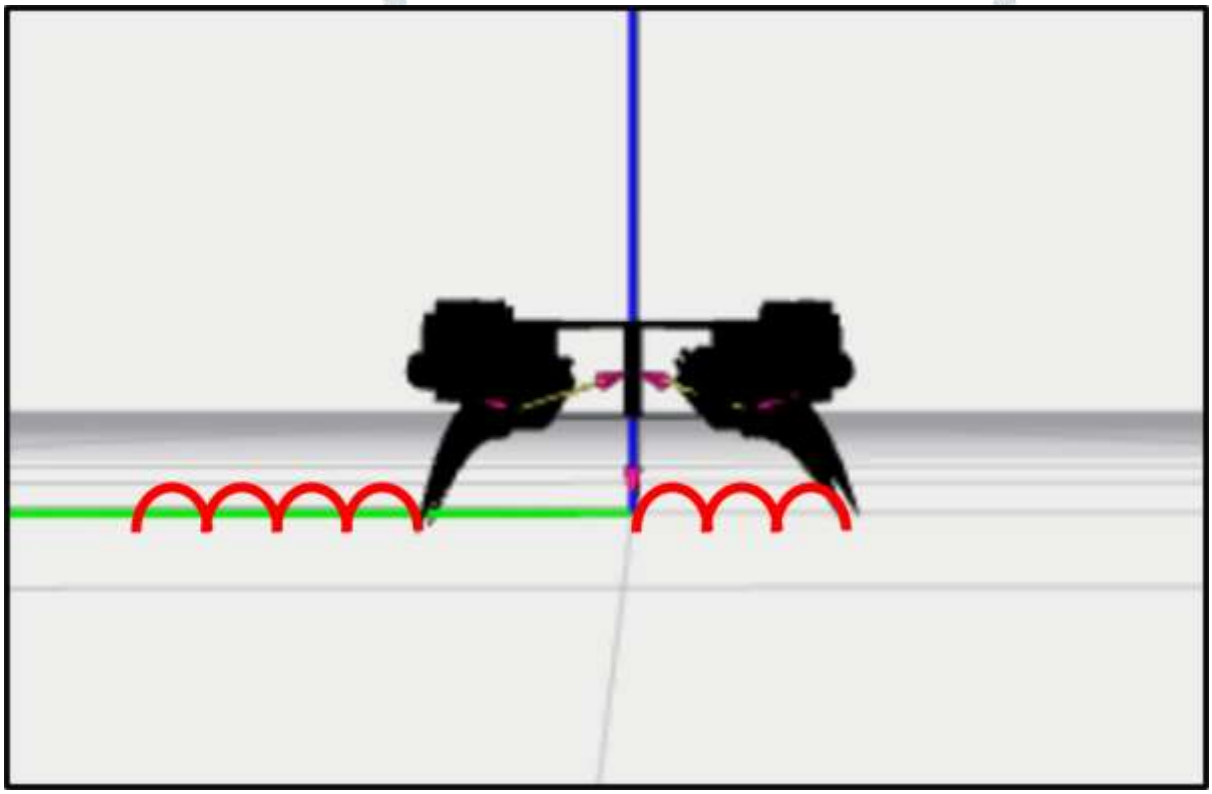

Figure 15 "D" Shaped Trajectory

Now, after the simulation is being done, a trial is being carried out on a servo motor. The servo motor is connected with Arduino Uno on PWM pin 9 as shown in Figure 16. Then the Arduino is connected with laptop and then to interface ROS with Arduino we established a serial link between them.

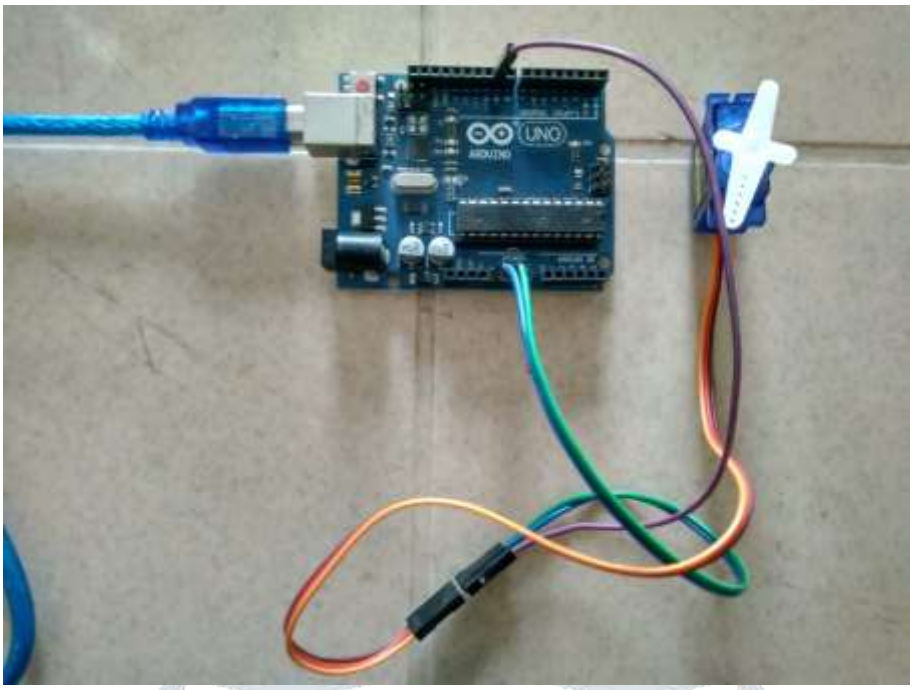

Figure 16 Arduino Setup

After establishing the serial link, we can rerun the simulation and then control the servo from the joint state publisher GUI. As we change the angle on joint state publisher GUI, we can see the change in angle of servo motor.

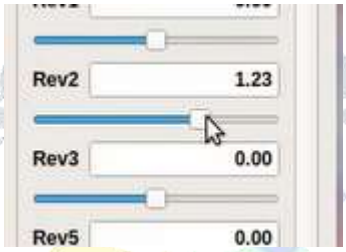

Figure 17 Joint State Publisher GUI

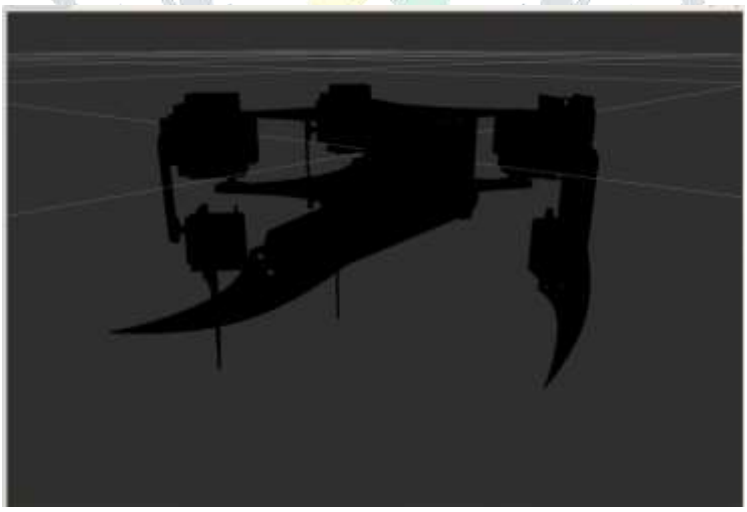

Figure 18 Rviz

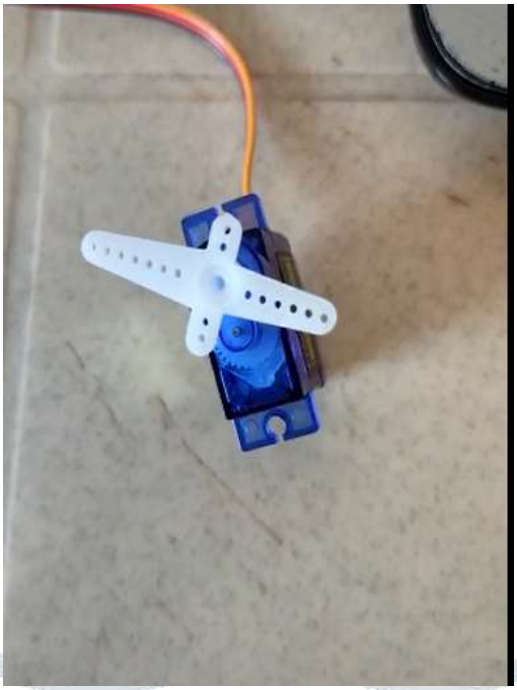

Figure 19 Servo Motor

IV. CONCLUSION

Quadruped Robot are more adaptable to terrain changes by changing foot position. By doing this simulation we get the velocity and joint state profile during trot gait. Different graphs can be generated for different type of gait pattern. Using the simulation and the graph values servo motor is being controlled via. interfacing ROS with Arduino. The "D" shaped leg trajectory is generated during the simulation. Doing the Simulation in ROS helps us to use more opensource resources for simulation. Creating a CAD model in Fusion 360 makes it much easier to work with ROS.

V. FUTURE SCOPE

- The simulation can be extensively implemented on a complete hardware model and the motion can be made much more precise while it is being deployed in real world environment.
- Sensors like ultrasonic, LiDar can be implemented sensing the environment around the robot. Also, use of camera vision can also be implemented for robot perception and better accuracy for obstacle avoidance.
- Using the sensors and coupling it with a much more powerful embedded computers like Jetson Nano the robot can be made autonomous.

VI. REFERENCES

[1] X. Zeng, S. Zhang, Y. Fu, H. Zhou, X. Li and H. Zhang, "Leg Trajectory Planning for Quadruped Robots with High-Speed Trot Gait," *MDPI,* 2019.

[2] W. Wang, Y. Zhang, Y. Yang and X. Shao, "Trajectory Planning and Posture Adjustment of a Quadruped Robot for Striding Gait," *IEEE,* 2011.

[3] Z. Shauaishuai, Y. Li, B. Li and X. Rong, "A Composite COG Trajectory Planning Method for the Quadruped Robot Walking on Rough Terrain," *International Journal of Control and Automation,* 2015.

[4] U. A. Moin, I. S. Rafiqul and R. A. Atiqur, "Development of an 8 DOF Quadurped Robot and Implementation of Inverse Kinematics Using DH convention," *IEEE,* 2018.

[5] M. S. Maradkar and P. V. Manivannan, *International Conference on Robotics: Current Trends and Future Challenges,* 2016.

[6] S. Liu, S. Yu and h. Li, "Dynamical Balance Optimization and Control of Quadruped Robots Under Perturbing External Force," *Internation Conference of Intelligent Control and Automation Science,* 2013.

[7] J. Lentin, Learning Robotics Using Python, Aluva: Packt Publishing Ltd, 2015.

[8] L. Joseph, Robot Operating Sysytem for Absoulte Begineers, Aluva: Apress, 2018.

[9] P. González de Santos, E. Gracia and J. Estremera, Quadrupedal Locomotion, Springer, 2006.

[10] K. K. Ganesh and M. P. Pushparaj, "Dynamic modelling & simulation of a four legged jumping robot with compliant legs," *Elsevier,* 2012.

[11] J. J. Craig, Introduction to Robotics, Pearson Education, 2004.

[12] Y. N. Andrew and Z. K. J., "Task-Space Trajectories via Cubic Spline Optimization," *IEEE International Conference on Robotics and Automation,* pp. 1675-1682, 2009.

[13] T. Kitamura, "Fusion2URDF," *GitHub repository,* 2020.