ENPM662 - Introduction to Robot Modeling

Project 2 Report

Quadruped Spiderbot Crawler

Authors

Aditya Chaugule (119280539)

Satish Vennapu (118306759)

Date: 12/07/2022

Abstract

The focus of this project is to understand and implement the core concepts of robot modelling course such as forward kinematics, inverse kinematics, velocity kinematics and design and simulation of the robot. For the scope of this project, the equations of forward kinematics, inverse kinematics, and velocity kinematics are manually calculated for a 8-DOF spider configuration robot. Furthermore, the spider-inspired robot model is designed in SolidWorks and simulation of the model and kinematic equations are performed in ROS, Rviz, and Gazebo environments.

We designed a spider structure-based robot which addresses the concern of exploring and searching in uncharted territories and assists in managing disasters, search and rescue operations in a space-constrained and hazardous environment.

Project Goals

- Understand and implement the core concepts of robot modelling. Robot modelling consists of forward kinematics, inverse kinematics, velocity kinematics and simulation in ROS.
- Design a Robot Model using SolidWorks and export the URDF.
- Calculations of equations manually for the Robot's kinematics, particularly forward and inverse kinematics.
- Validation of all the equations by simulating the robot in the ROS/Gazebo environment.
- Design and simulate a robot which can work in real-time to explore uncharted territories and move swiftly avoiding collisions with obstacles.

Figure – Quadruped Spiderbot Crawler

Table of Contents

Introduction and Organization

In nature and modern society, remote spaces and situations are inaccessible or highly dangerous for humans, such as planetary surfaces, construction sites, mining zones, disaster struck areas and anti-terrorism action situations. For the progress of human society, and the development of science and technology, however, it is essential to involve in long-range research on the circumstances mentioned above in order to seek feasible solutions. What these all conditions have in common are dynamic, uneven and irregular terrains, making the application of wheeled robots and tracked robots limited. Legged robots are more suitable for these circumstances due to their terrain adaptation ability. Therefore, research on legged robots has become a hot topic in the field of robotics. Thus, we modeled and simulated the Spider bot.

The report is organized into sections, the first section describes a brief context introduction and how the report is organized. After this introductory section, the next section reveals the reason for choosing the problem statement and our interest in choosing this type of robot. Section three explains the CAD robot model specifying the geometrical aspects and the dimensional aspects. It provides a brief idea of the topics such as the robot’s sensors, motors, joints, and materials which are critical in the design. Some pictures of the robot design are attached in this section. Section four summarizes the appropriateness of the design of the Spider Bot in solving multiple problems and doing different tasks better than robots of a similar kind. Section five provides an account of the robot's possible scope of study and the part of the robot’s functions that will be studied in this model. Section six presents scope appropriateness which briefly summarizes the scope of the studies' significance. Section seven mentions the scope of achievement of this project performed. Section eight presents the assumptions made to design the Spider Bot ground drone model. The next section discusses the approach taken to complete this project. After defining the steps involved, the next two sections mention the forward kinematics and inverse kinematics of the robot. Section eleven reviews the model validation and validates the equations. The twelfth section involves the validation of the equations by performing a simulation in a physics environment. Section thirteen explains the challenges involved during the implementation phase. The next three sections include future work that can be performed, conclusions, and acknowledgment. Finally, the last two sections mention references and citations.

Motivation

The natural topography comprises of harsh environmental conditions with a variety of dynamic terrain scenarios. Legged mobile robots which use artificial limbs for locomotion, help maneuver the robot in inclement conditions.

Our intent is to build a robot that will address the present concern of exploring and searching in uncharted territories and task-based configuration domains, in immensely dangerous and hazardous conditions. The robot model of our four-legged robot can operate in inaccessible regions and move swiftly in any surrounding that a conventional robot can't move. The robot helps assist in monitoring convoluted scenarios without the need for human presence. The current design lacks modularity and has redundant DOFs in each leg which increases design complexity and production scalability. Although the robot is designed to require reduced the number of total linkages compared to commercially available robots providing agile motion with significantly low power consumption.

We designed a robot to attempt to solve the difficulties faced by workers working in dangerous and inaccessible conditions. Autonomous robots are an integral part of "Industry 4.0", as they can complete tasks with intelligence, persistence, and precision with minimal human input. Therefore, understanding the robot model and workspace is essential for efficient working. The motivation for our project is to understand robot kinematics and practically implement a customized robot model in ROS and simulate it in the Gazebo environment for hazardous environments. A nature-inspired spider-configuration robot was selected owing to its flexible configuration design and stability.

Applications

The spiderbot can effectively traverse and conduct survey operations during catastrophe missions, such as collapsed buildings after an earthquake. Owing to its mobility, small size, and high-definition imaging, the spider can avoid various obstacles and enter difficult-to-reach locations to search for trapped victims. Replacing humans in dangerous missions (for example, sweeping and neutralizing minefields) is another potential application area. These challenges are met by a highly mobile walking scheme. Typically a modular design is implemented with advanced control laws to extend the application to adapt the environment suitably and in dynamic conditions

Robot Description

The concept robot is designed with four legs (spider-legged design), each of which will include a hip joint and a knee joint with two degrees of freedom (DOFs), allowing the robot to move laterally and longitudinally. Any desired motion can be produced by all legs working together properly. Four legs will have benefits like better agility, sturdy support, and optimum balance on uneven surfaces. To fulfill the objective of exploring, rescuing, and searching in a hazardous environment (mining), we have implemented a camera and LiDAR on the top surface of the spider-bot.

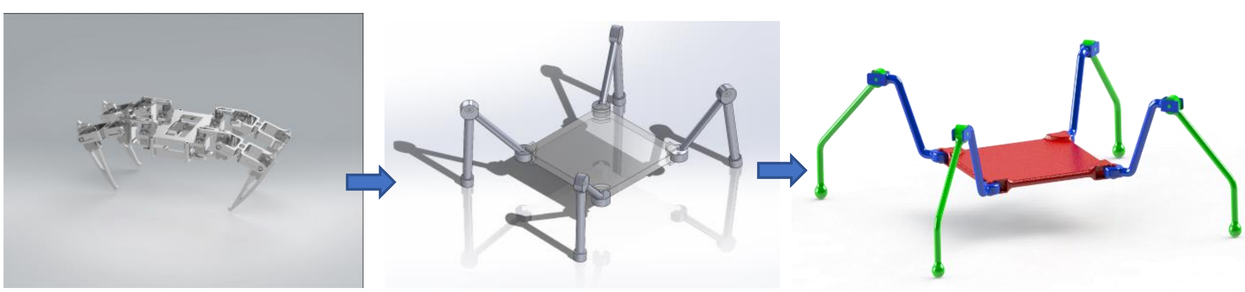


Figure: Design of robot CAD model from (a) inspired design (b) Initial design (c) Final Design

Spider Bot Specifications	
Weight	2 kg(without the arm) and 3 kg (with arm)
Dimensions of the Robot	200 x 200 x 10 mm (without the arm) 505 x 505 x 187 mm (with arms)
Material	Aluminum 1060 alloy
Degrees of Freedom of the Bot	8 DOF
Applications	Working in Hazardous Environment, Survey and Reconnaissance
Embedded Vision	Camera on the face of the robot (1280 x 800 pixels)
Motors	8 Servo motors driving the legs
Sensors	Lidar Sensor
IP Classification	IP50 rating
Power Requirement	Standard power outlet (120V, 6 amps)
Battery	6-7 hours (12V)
Operating Software and Onboard computer	Processor - Intel Core i7, 3.4 GHz Cortex-A7 Memory - 4GB RAM (microSD slot) Wireless - Bluetooth Networking — 10/100 Ethernet port
Robot subsystem	Joystick controller pad

Table: Robot Specifications

Robot Appropriateness for the task

We have constructed an all-terrain robot that can travel in any direction while maintaining proper balance. The four-legged framework with adequate gait scheduling gives the entire structure flexibility, maneuverability, and size-restricted capabilities. The robot has a camera and LiDAR module, which can do a variety of things including capture images, recognize objects, plan paths, etc. The robot is actually designed to navigate and investigate risky environments like mines, places with low oxygen levels, fossil fuel extracting wells, and areas exposed to radiation. The robot will be most effective for these tasks if it has all-terrain movement, agility, maneuverability, and balance. The proposed robot will be ideally equipped and ready to operate there and do the required task, regardless of how low the oxygen is, how poisonous the place is, how uneven the territory is, or how much radiation is released in that location.

Scope Description:

Goals achieved:

1. The designed and simulated spiderbot can walk in any direction based on the controlled input while observing and monitoring its environment through a camera and LiDAR modeule.

2. Knee joints, and hip joints result in a more restrictive workspace albeit the goal to reduce number of linkages and achieve a stable gait was achieved.

3. Additional capability of the robot that transforms the legs into paddling feet to swim in aquatic and sub-surface conditions.

DH Parameter Table

To demonstrate the kinematic analysis of the robot linkages, using selective variable convention set, The **Denavit–Hartenberg parameters** set were selected. The DH parameter table was constructed for the linkages of the robot.

Link	a	α	d	θ
Gazebo – Base	0	0	h	0
Base – Link 0	w/ √2	0	0	π/4
Link 0 – Link 1	a	π/2	a	Θ_1
	0	0	0	Θ_2
Link 1 - End Point	a	-π/2	0	0
	0	0	-b	-π/4

Table: DH parameter set of Front Right Arms of the robot

Where,

w = width of base frame = 200 mm

a = length dimension = 100 mm

b = length dimension = 175 mm

Coordinate frames in the DH table were assigned similarly in the Solidworks CAD package to enable further analysis in gazebo environment. The orientation of the gazebo reference origin and Solidworks origin were noted and a base coordinate transformation was incorporated. The coordinate frames comply the convention of assigning frames –

- x-direction: points to the forward motion of the robot
- z-direction: points along the gravity vector direction

```python
# DH Parameter table

z_z = w/sym.sqrt(2)

dh_tablestatic = sym.Array([0, 0, h, 0])

dh_tablefr = sym.Array([[z_z, 0, 0, sym.pi/4],
                        [a, sym.pi/2, a, theta1],
                        [0, 0, 0, theta2],
                        [a, -sym.pi/2, 0, 0],
                        [0, 0, -b, -sym.pi/4]])

dh_tablefl = sym.Array([[z_z, 0, 0, -sym.pi/4],
                        [a, sym.pi/2, a, theta3],
                        [0, 0, 0, theta4],
                        [a, -sym.pi/2, 0, 0],
                        [0, 0, -b, sym.pi/4]])

dh_tablerl = sym.Array([[z_z, 0, 0, 5*sym.pi/4],
                        [a, sym.pi/2, a, theta5],
                        [0, 0, 0, theta6],
                        [a, -sym.pi/2, 0, 0],
                        [0, 0, -b, -5*sym.pi/4]])

dh_tablerr = sym.Array([[z_z, 0, 0, 7*sym.pi/4],
                        [a, sym.pi/2, a, theta7],
                        [0, 0, 0, theta8],
                        [a, -sym.pi/2, 0, 0],
                        [0, 0, -b, -7*sym.pi/4]])
```

Figure: The DH table formulated for every arm – i.e. a) Front Right b) Front Left c). Rear Right and d) Rear Left Legs

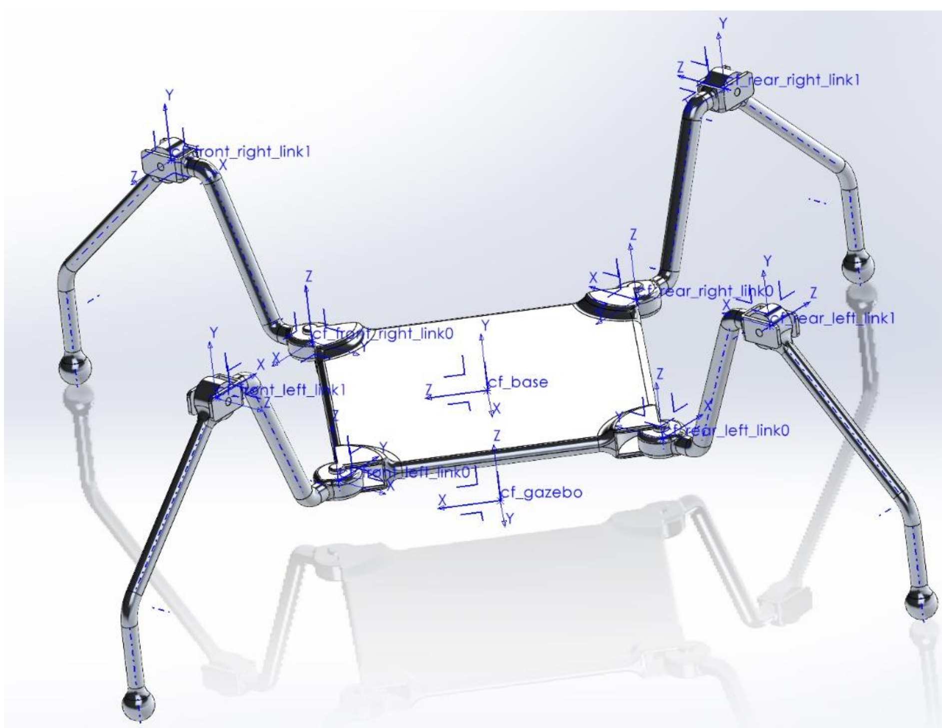


Figure: Assigned coordinate frames mapped in Solidworks to obtain link transformation for URDF export

Linkage Transformations

A transformation matrix was tabulated using the DH parameter table for every link. Transformation matrix are given as –

$$A = \begin{bmatrix} \cos(\theta) & -\sin(theta) & 0 & 0 \\ \sin(\alpha) & \cos(\theta) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & a \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(\alpha) & -\sin(\alpha) & 0 \\ 0 & \sin(\alpha) & \cos(\alpha) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Successive transformations are evaluated for net transformation calculations. This gives the total transformation matrix for a robot arm.

$$T_{gazebo\ origin}^{base\ origin} = A_1$$

$$T_{link\ 0}^{gazebo\ origin} = T_{base\ origin}^{gazebo\ origin} . T_{link\ 0}^{base\ origin} = A_1 . A_2$$

$$T_{link\ 1}^{gazebo\ origin} = T_{base\ origin}^{gazebo\ origin} . T_{link\ 0}^{base\ origin} . T_{link1}^{link0} = A_1 . A_2 . A_3$$

$$T_{arm\ end\ point}^{gazebo\ origin} = T_{base\ origin}^{gazebo\ origin} . T_{link\ 0}^{base\ origin} . T_{link1}^{link0} . T_{arm\ end\ point}^{link\ 1} = A_1 . A_2 . A_3 . A_4$$

Further, total transformation matrices are calculated for every arm. The setup of the total transformation matrix for each arm is shown in the image below.

```
T_static0 = get_tf(dh_tablestatic[0],  dh_tablestatic[1],   dh_tablestatic[2],  dh_tablestatic[3])

T_fr = sym.eye(4)
T_fl = sym.eye(4)
T_rl = sym.eye(4)
T_rr = sym.eye(4)

for i in range(5):
    T_fr *= get_tf(dh_tablefr[i][0],  dh_tablefr[i][1],   dh_tablefr[i][2],  dh_tablefr[i][3])

for i in range(5):
    T_fl *= get_tf(dh_tablefl[i][0],  dh_tablefl[i][1],   dh_tablefl[i][2],  dh_tablefl[i][3])

for i in range(5):
    T_rl *= get_tf(dh_tablerl[i][0],  dh_tablerl[i][1],   dh_tablerl[i][2],  dh_tablerl[i][3])

for i in range(5):
    T_rr *= get_tf(dh_tablerr[i][0],  dh_tablerr[i][1],   dh_tablerr[i][2],  dh_tablerr[i][3])
```

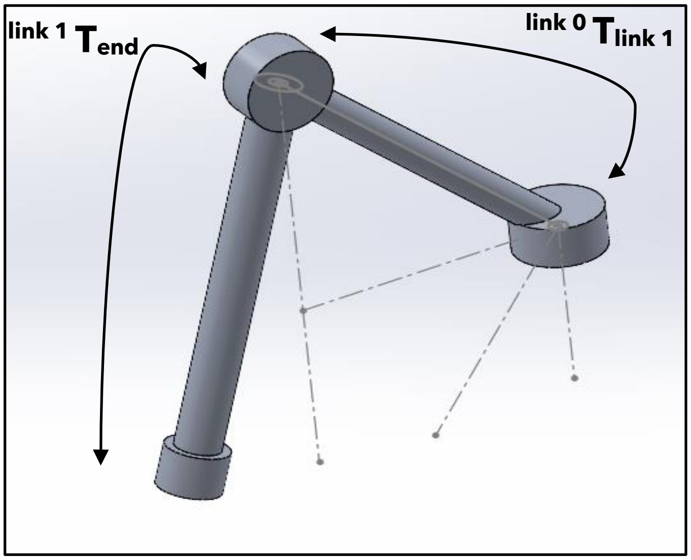


Figure: Transformation between linakges

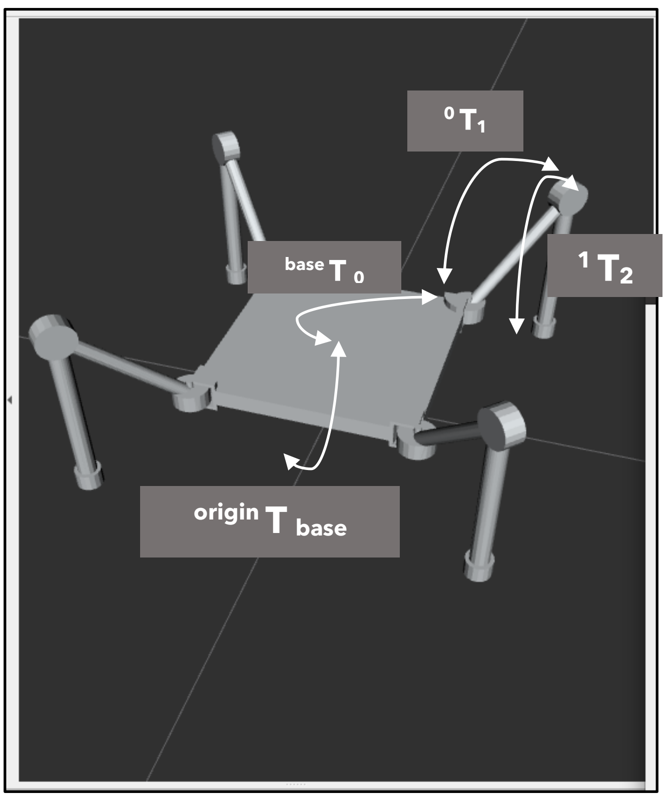


Figure: Transformation between linkages to base of the robot

The figure above shows the transformation relative to each link to the robot. The transformation at each instant is updated and further the forward kinematics relations are realized.

Forward Kinematics

The forward kinematics of the robot were tabulated to determine the position and potentially the velocity of the end point of the leg, given the joint angles and angular velocities. The linkage transformations were used in sequence to obtain the forward kinematics of the robot. Since the robot has 8 DOF, a combination of values for all joint angles determine the orientation configuration of the robot.

Forward Kinematics relations for all four legs are given below –

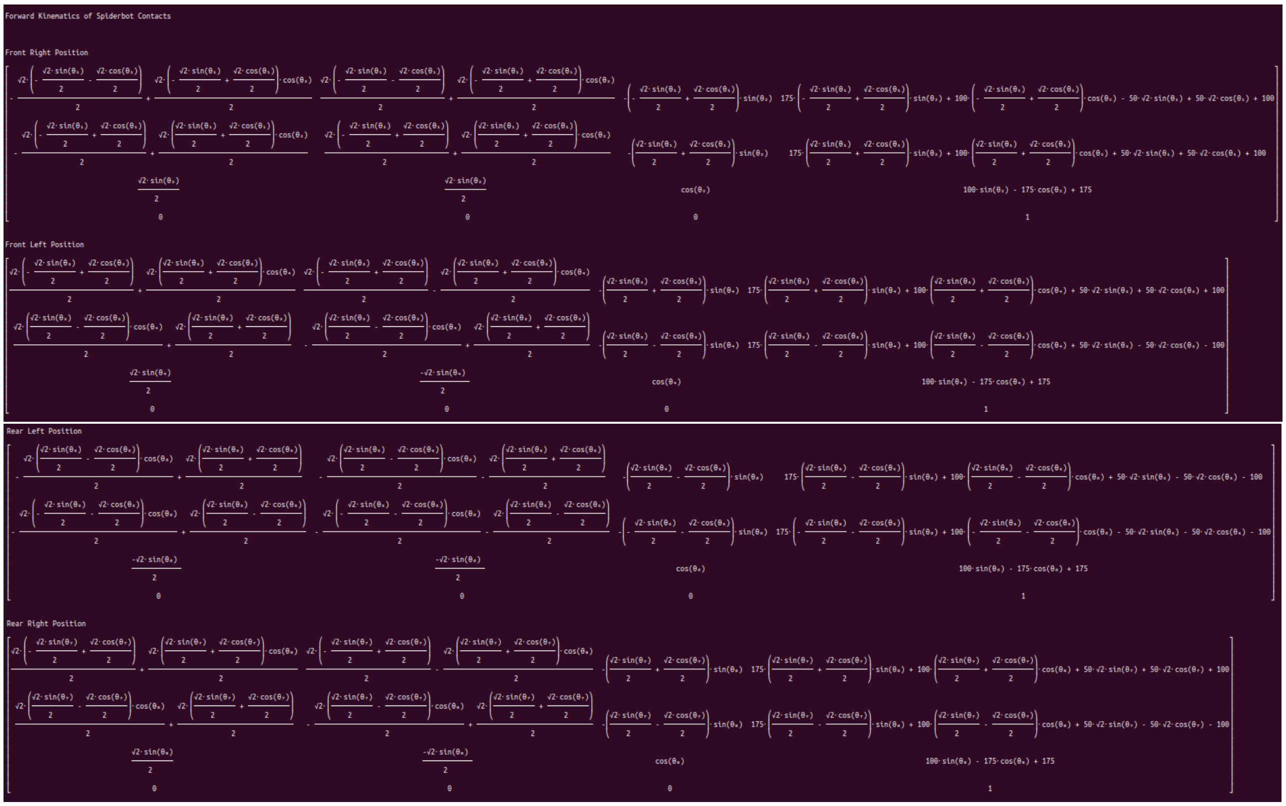


Figure: Forward Kinematics equations for Spiderbot

For initial pose, all θ are taken a 0 (zero). The transformation matrices for this pose give the forward kinematics relationships of the end point of the spiderbot leg which is given below.

Forward Kinematics of Spiderbot Contacts

Front Left Position

$$\begin{bmatrix} 1 & 0 & 0 & 100 + 100\cdot\sqrt{2} \\ 0 & 1 & 0 & -100\cdot\sqrt{2} - 100 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Front Right Position

$$\begin{bmatrix} 1 & 0 & 0 & 100 + 100\cdot\sqrt{2} \\ 0 & 1 & 0 & 100 + 100\cdot\sqrt{2} \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Rear Left Position

$$\begin{bmatrix} 1 & 0 & 0 & -100\cdot\sqrt{2} - 100 \\ 0 & 1 & 0 & -100\cdot\sqrt{2} - 100 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Rear Right Position

$$\begin{bmatrix} 1 & 0 & 0 & 100 + 100\cdot\sqrt{2} \\ 0 & 1 & 0 & -100\cdot\sqrt{2} - 100 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Forward Kinematics Validation

Rviz visualization tool was initially used to observe the joint states using joint state controllers to verify the angular displacements compliance. Further, The joint configurations were validated gazebo simulation environment.

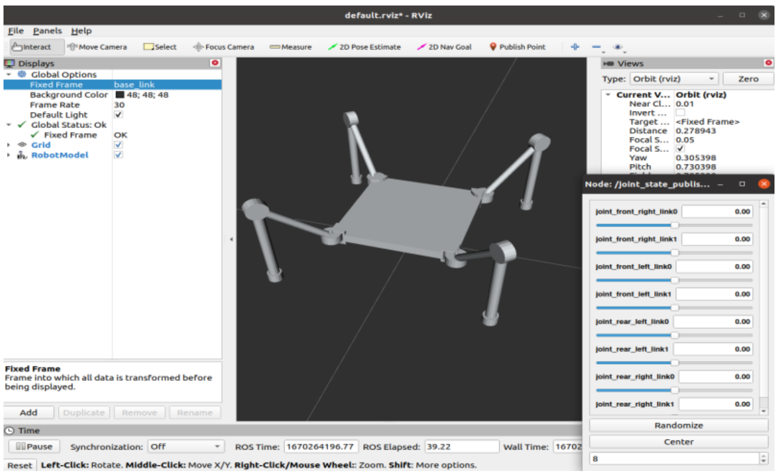
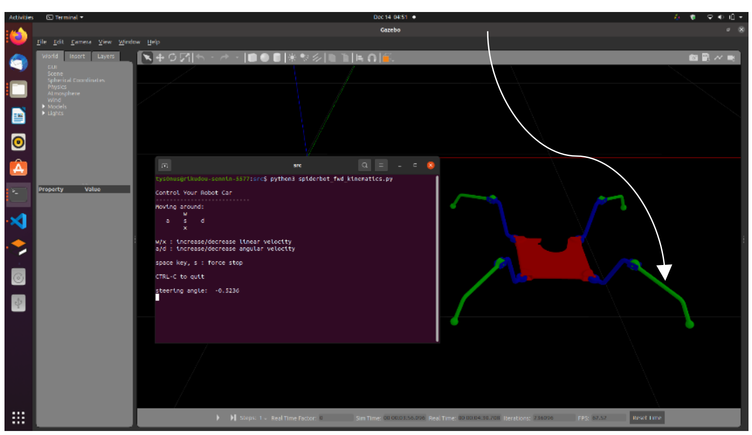
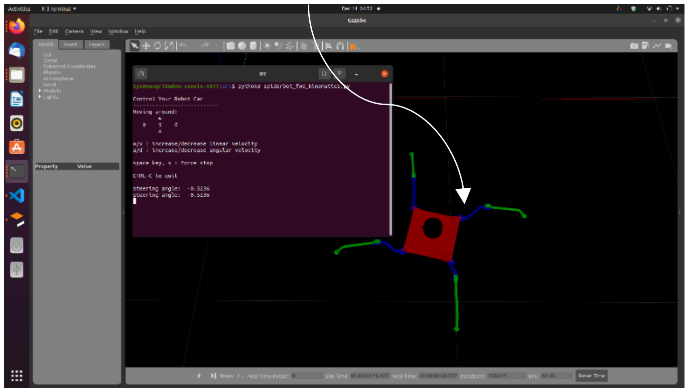


Figure – The joint state controllers were manually checked using the controller slider

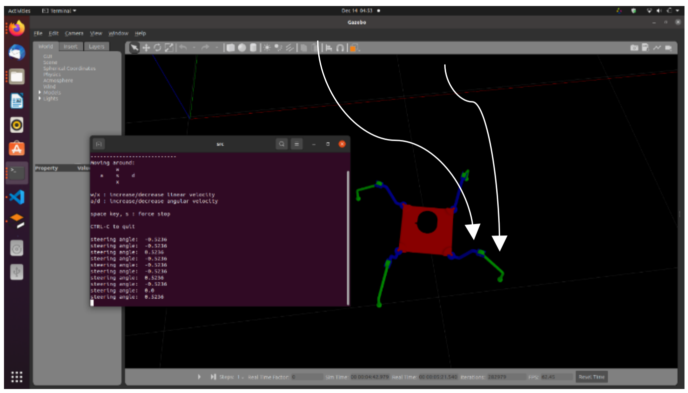
For the final version of the robot, the forward kinematics validation is accomplished for known position constraints and orientations.



Pose 1: Front Right Link 1 is given a rotational displacement



Pose 2: *Front Right Link 0 is given a rotational displacement*



Pose 3: *Front Right Link 0 and Front Right Link 1 is given a rotational displacement*

Inverse Kinematics

The spiderbot has 2 linkages per arm, hence the relationship between the joint angles and robot configuration can be determined geometrically. The angles are determined as –

$$\theta_1 = \text{atan2}\left(\frac{y}{x}\right) - \frac{\pi}{4}$$

$$\theta_2 = \text{atan2}\left(\frac{z}{\sqrt{(x^2 + y^2) - len(link_0)}}\right)$$

Although this solution is feasible for verification purposes only. The the combined motion of all linkage joints produce many solutions for the same configuration. Therefore, a numerical equation solution is optimal to identify the inverse kinematics, where –

$$\Delta x_{k+1} = J_p^{+}(x_k).\Delta\theta_k$$

Where J is the Moore-Penrose pseudo-inverse of the Jacobian matrix. Hence, the Jacobian matrix is computed as –

$$J = \begin{bmatrix} \dfrac{\partial O_n}{\partial \theta_1} & \dfrac{\partial O_n}{\partial \theta_2} \\ z_{i-1} & z_{i-1} \end{bmatrix}$$

The Jacobian matrix was setup using a python script and evaluated as follows –

```
T_fr_tot = T_static0*T_fr
T_fl_tot = T_static0*T_fl
T_rl_tot = T_static0*T_rl
T_rr_tot = T_static0*T_rr

Tl0= sym.eye(4)
for i in range(2):
    Tl0 *= get_tf(dh_tablefr[i][0],  dh_tablefr[i][1],   dh_tablefr[i][2],  dh_tablefr[i][3])

z0 = sym.Matrix([0, 0, 1])
z1 = T_static0[0:3, 2]
z2 = Tl0[0:3, 2]
z2 = T_fr_tot[0:3, 2]


O1 = T_static0[0:3, 3]
O2 = Tl0[0:3, 3]
O3 = T_fr_tot[0:3, 3]

################Setting up Jacobian Matrix
J = sym.Matrix([[sym.diff(O3,theta1), sym.diff(O3,theta2)],[z1, z2]])
print("\n\nInverse Kinematics of Spiderbot Contacts\n")
sym.pprint(J)
```

Figure: *Jacobian Matrix Setup in Python Script*

```
Inverse Kinematics of Spiderbot Contacts
```

$$\begin{bmatrix} 175\cdot\left(-\frac{\sqrt{2}\cdot\sin(\theta_1)}{2} - \frac{\sqrt{2}\cdot\cos(\theta_1)}{2}\right)\cdot\sin(\theta_2) + 100\cdot\left(-\frac{\sqrt{2}\cdot\sin(\theta_1)}{2} - \frac{\sqrt{2}\cdot\cos(\theta_1)}{2}\right)\cdot\cos(\theta_2) - 50\cdot\sqrt{2}\cdot\sin(\theta_1) - 50\cdot\sqrt{2}\cdot\cos(\theta_1) & -100\cdot\left(-\frac{\sqrt{2}\cdot\sin(\theta_1)}{2} + \frac{\sqrt{2}\cdot\cos(\theta_1)}{2}\right)\cdot\sin(\theta_2) + 175\cdot\left(-\frac{\sqrt{2}\cdot\sin(\theta_1)}{2} + \frac{\sqrt{2}\cdot\cos(\theta_1)}{2}\right)\cdot\cos(\theta_2) \\ 175\cdot\left(-\frac{\sqrt{2}\cdot\sin(\theta_1)}{2} + \frac{\sqrt{2}\cdot\cos(\theta_1)}{2}\right)\cdot\sin(\theta_2) + 100\cdot\left(-\frac{\sqrt{2}\cdot\sin(\theta_1)}{2} + \frac{\sqrt{2}\cdot\cos(\theta_1)}{2}\right)\cdot\cos(\theta_2) - 50\cdot\sqrt{2}\cdot\sin(\theta_1) + 50\cdot\sqrt{2}\cdot\cos(\theta_1) & -100\cdot\left(\frac{\sqrt{2}\cdot\sin(\theta_1)}{2} + \frac{\sqrt{2}\cdot\cos(\theta_1)}{2}\right)\cdot\sin(\theta_2) + 175\cdot\left(\frac{\sqrt{2}\cdot\sin(\theta_1)}{2} + \frac{\sqrt{2}\cdot\cos(\theta_1)}{2}\right)\cdot\cos(\theta_2) \\ 0 & 175\cdot\sin(\theta_2) + 100\cdot\cos(\theta_2) \\ 0 & -\left(-\frac{\sqrt{2}\cdot\sin(\theta_1)}{2} + \frac{\sqrt{2}\cdot\cos(\theta_1)}{2}\right)\cdot\sin(\theta_2) \\ 0 & -\left(\frac{\sqrt{2}\cdot\sin(\theta_1)}{2} + \frac{\sqrt{2}\cdot\cos(\theta_1)}{2}\right)\cdot\sin(\theta_2) \\ 1 & \cos(\theta_2) \end{bmatrix}$$

Figure: *Jacobian Matrix Evaluation*

Workspace Study

The workspace of the Spiderbot is symmetric for all of the four arm linkages. Hence a workspace plot for only one arm is shown below. Since each arm has two linkages, the workspace is plotted for the variation of angles theta 1 and theta 2 within a constraint set of -30°≤ θ_1 ≤ 30° and -30°≤ θ_2 ≤ 30°.

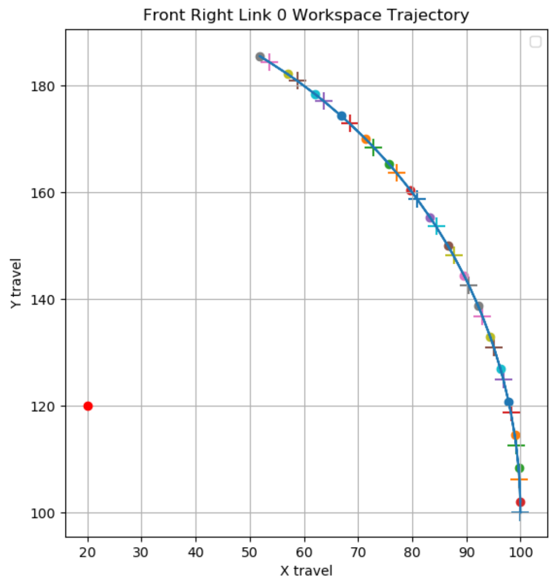


Figure: Front Right Hip Joint Workspace Trajectory

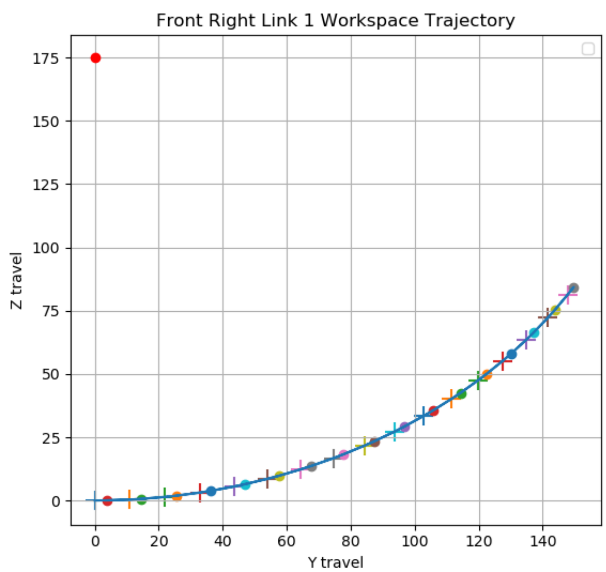


Figure: Front Right Knee Joint Workspace Trajectory

A live updated workspace is plotted for variation of angles: [Click here]

Assumptions

The spider robot is designed using a 4-legged configuration. Each leg is characterized using a 2-arm mechanism through a revolute joint. Robot movement is limited to the custom generated gazebo world scaled for the robot interaction with included objects. The robot maneuvers in a dynamic terrain environment in the gazebo world where physics is enabled. The soil adhesion and sinking as assumed to be non-existant for demonstration purposes.

Robot Model design assumptions:

- All the joints are considered as pure revolute joints.
- All links are assumed to be rigid in construction.
- The weight of the robot is opined as low as possible using aluminum and polymer materials.
- Any internal joint-friction and external disturbances are not taken into account.
- Robot self-collision is not considered, only collision with external obstacles is taken into account for this scope of the project.
- The path of the arm or the robot is just one solution among all the other solutions it can have, this may or may not be the optimal solution.

Control method

The controllers were defined for the spiderbot using the ros controllers. The joint state controller was predefined while exporting the urdf from Solidworks. Since the robot is composed of 8 constraint driven DOF's, effort controllers for each joint actuator were defined. The effort controllers command a desired torque effort to the spiderbot hardware interface transmission. The effort controller uses a joint position controller which receives a position input and sends an effort output using a PID controller. The PID gains for the controllers were identified using trial and error method for which the robot stabilized in its orientation optimally. Hence the desired output of the actuator was satisfactorily calibrated using a PID controller.

For Rviz visualization, the joint state controllers were manipulated for spiderbot assuming the joints to be continuous. This allowed for flexible verification of the robot configuration and initialization controller check for given coordinate frame reference orientation.

For Gazebo simulation, the originally defined revolute joint definitions were incorporated. Teleop Node is implemented using a python script to publish joint commands on prompt of the user. The orientation of each joint angle changes with the appropriate input published on the topic. For each joint, a transmission was defined in the urdf and a virtual controller was incorporated. The teleop node publishes messages to the controller topics defined in the launch file and the spiderbot subscribes to the topic. The schematic is shown below -

```
/spiderbot/controller_front_right_link0/command
/spiderbot/controller_front_right_link0/pid/parameter_descriptions
/spiderbot/controller_front_right_link0/pid/parameter_updates
/spiderbot/controller_front_right_link0/state
/spiderbot/controller_front_right_link1/command
/spiderbot/controller_front_right_link1/pid/parameter_descriptions
/spiderbot/controller_front_right_link1/pid/parameter_updates
/spiderbot/controller_front_right_link1/state
```

Figure: Active controller topics to which the teleop node publishes the message

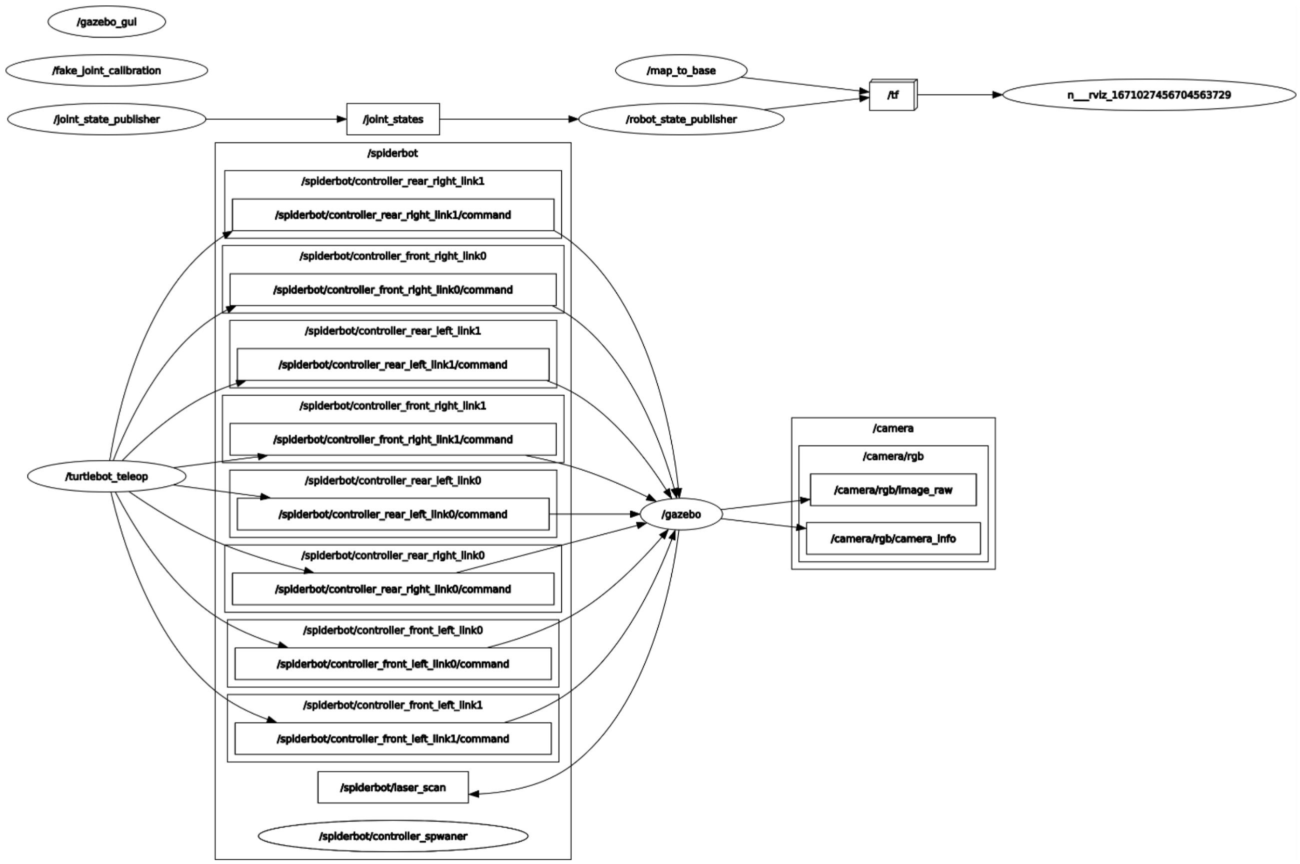
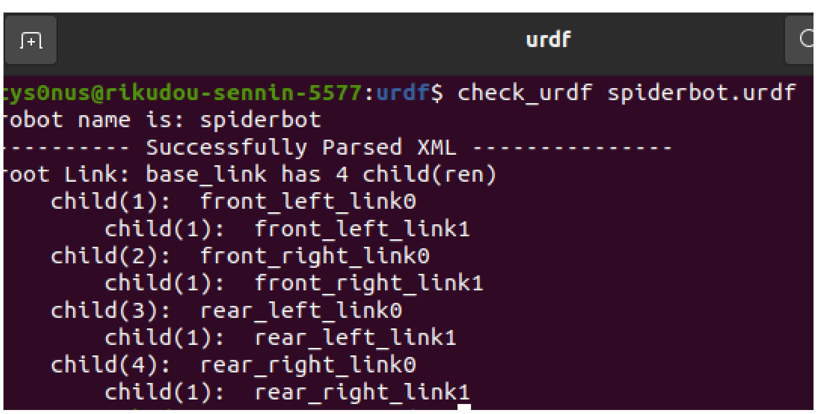


Figure: ROS computation graph

Also, an open loop controller was defined using a python script which publishes messages to the controller command topics using a Publisher Node to emulate gait mechanism for Spiderbot.

Gazebo and RViz visualization

Prior to Gazebo simulation the ROS package was setup to include all the necessary dependencies. The urdf exported from Solidworks was verified using check_urdf command, as follows -



```
urdf
:ys0nus@rikudou-sennin-5577:urdf$ check_urdf spiderbot.urdf
obot name is: spiderbot
---------- Successfully Parsed XML ---------------
oot Link: base_link has 4 child(ren)
    child(1):  front_left_link0
        child(1):  front_left_link1
    child(2):  front_right_link0
        child(1):  front_right_link1
    child(3):  rear_left_link0
        child(1):  rear_left_link1
    child(4):  rear_right_link0
        child(1):  rear_right_link1
```

Figure: URDF check for all linkage definitions and child-parent relations

After creation of urdf, a simple transmission was added to the Spiderbot joints using an Effort Joint interface. The actuators were defined for adding controllers for robot joint motions. A PID controller was selected owing to its quick response time and robost feedbcack and the definitions of Effort Joint controllers were listed in the yaml file.

A camera and LiDAR interface was incorporated through xacro - macros for ROS. A launch file was created to launch a sequence of nodes in the gazebo environment with definition of transform data and the controllers. Finally, the simulation was run the ROS gazebo environment with Rviz visualization using a teleop node.

The robot performs satisfactorily to emulate the modelled motions as per the given controlled input. The images below show the RViz visualization of the gazebo environment where the robot performas tasks such as monitoring and surveying. The camera output shows the objects placed in their vicinity with a true field of view (FoV) of 1.39 degrees. Simultaneously, the LiDAR scanner shows the lazer scan of the environment.

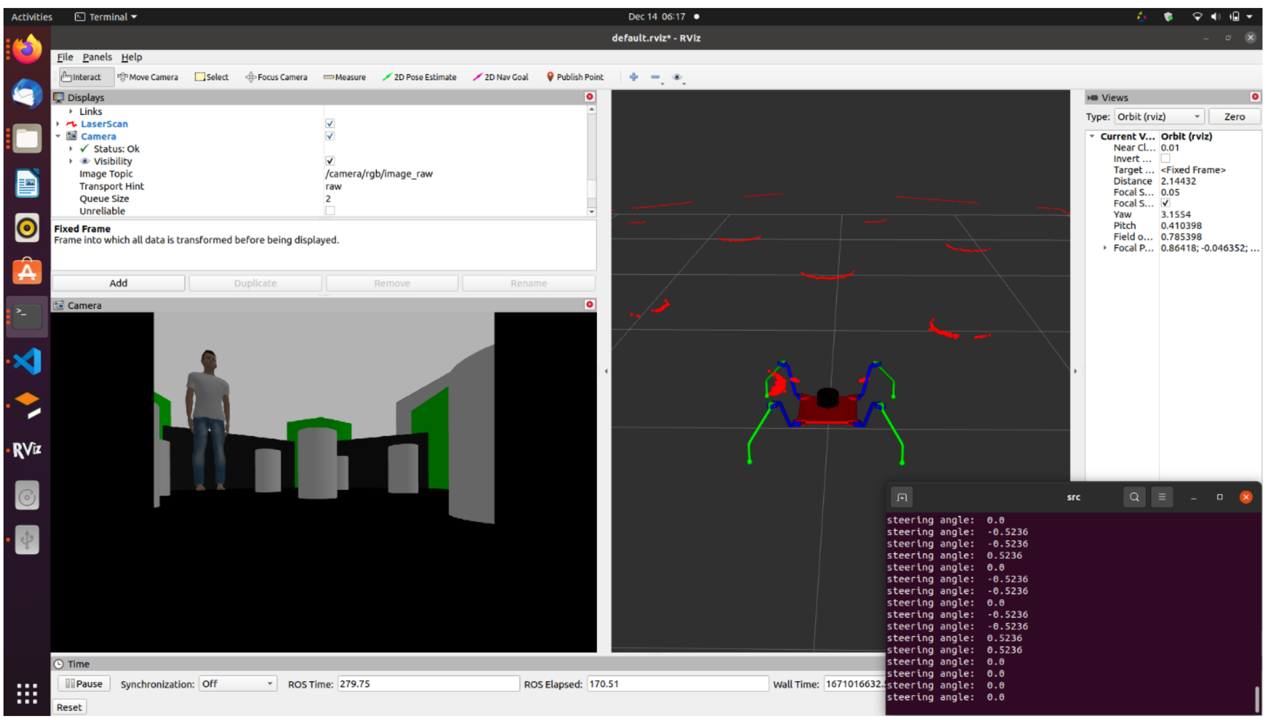


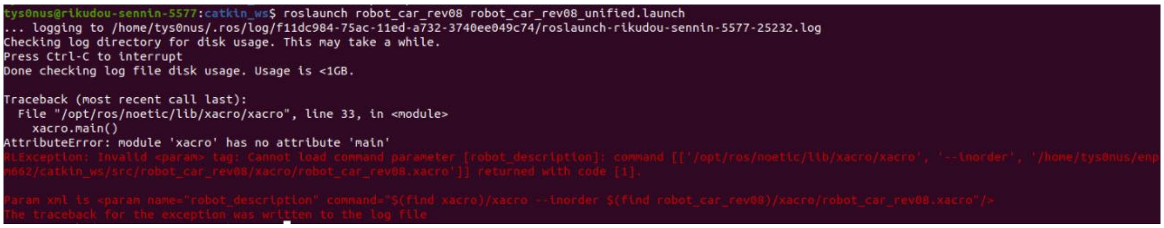
Figure: Rviz visualization of the gazebo environment

Simulation Video: [Click here]

Challenges Faced

During the project, we faced three major challenges –

- Unable to build a unified xacro file due to ROS package incongruencies. Due to xacro package not found problem getting resolved at an end stage of the project, the additional scope of adding a manipulator arm on Spiderbot was not implemented.



- Abnormal Linkage movement, the robot's legs not moving as expected.
- Presence of multiple solutions of the Inverse Kinematics.

To overcome the challenge of properly moving the links, we had to adjust the PID values of the controller and the mass value of the links. Also the URDF was checked multiple time to ensure that any interference was not present. Adjusting these values properly resulted in the expected motion of the manipulator links. For the legs part, we had to use the sleep command in the rospy library so that each link would get enough time to get to the desired position before the next command is executed and thus this resulted in the appropriate motion of the legs.

A significant issue was validating the Inverse Kinematics solution with that of the Forward Kinematics due to the presence of multiple solutions of the Inverse Kinematics. To overcome this, we had to manually run an Inverse Kinematics script for each of the possible answer patterns to obtain the desired solution and reject the non-feasible solutions.

Scope of Future Works:

- Modelling Joints as a Virtual Spring to enable step feasibility control over dynamic terrains
- Designing smaller robots (multi-agent systems) to work in coordination with each other by replacing one big robot with six to ten small powerful robots reducing the investment cost.
- Enhancing the robot's capability by mimicking the spider movements such as rolling from the hills, and folding legs to swim in hazardous chemicals which was implemented in the project as a small demonstration.
- Placing an arm manipulator on the top of the robot, so that it can perform lots of tasks like pick and place, cleaning the chemicals from hazardous environments, etc.

Conclusions:

- The project helped us delve into modelling aspects and integrate the learning with ROS for understanding robot development fundamentals
- This project improved our skills in SolidWorks and in exporting URDF and integrating with ROS.
- Through this project we acquired in-depth knowledge about robot forward, inverse and velocity kinematics and learned the modified Denavit-Hartenberg Spong convention.
- Calculated and implemented robot forward and inverse kinematic equations.
- Learned the new concepts of bio-inspired robots and their functionality.
- Learned the simulation of the Robot in the ROS, Rviz, and Gazebo environment.
- Learned to add sensors and controllers and perform teleop operation and programming in python.

Acknowledgement:

We would like to thank Dr. Reza Monfaredi, Mr. Adarsh Malapaka, and Mr. Pawan Matripragada for their continuous support and feedback.

Citations:

- Tetsuta Takamura, Yuki Shimura, Kazuhiro Motegi **"**Development of Simulator and Analysis of Walking for Hexapod Robots" Proceedings of the IEEE 2019.

- Tolga Karakurt, Akif Durdu, and Nihat Yilmaz, "Design of Six-Legged Spider Robot and Evolving Walking Algorithms" International Journal of Machine Learning and Computing, Vol. 5, No. 2, April 2015
- 3. Ioannis Tsitsimpelis, C. James Taylor, BarryLennox, Malcolm J.Joyce, "A review of ground-based robotic systems for the characterization of nuclear environments", Progress in Nuclear Energy Volume 111, March 2019, Pages 109-124
- 4. Jianbo Sun, Jie Ren, Yinglian Jin, Binrui Wang, Dijian Chen, "Hexapod Robot Kinematics Modeling and Tripod Gait Design Based on the Foot End Trajectory"
- 5. Cavas, Mehmet & Ahmad, Muhammad Baballe. (2019). "A REVIEW ON SPIDER ROBOTIC SYSTEM", International Journal of New Computer Architectures and their Applications. 9. 19-24. 10.17781/P002604.
- 6. Servet Soyguder and Hasan Alli, "Design and prototype of a six-legged walking insect robot", Industrial Robot: An International Journal34/5 (2007) 412 – 422.