Introduction to Robot Modeling (ENPM662) Final-Project Report
Krabot Ground Drone (six-legged bot)

Authors: Ajinkya Parwekar (117030389)

Karan Sutradhar (117037672)

Date : 12/09/2020

Abstract:

The main focus of this project is to understand and implement the core concepts of robot modelling. Robot modelling consists of forward kinematics, inverse kinematics, velocity kinematics and trajectory planning to avoid collisions for the robot and the manipulator arm. For this scope of this project the equations of forward kinematics, inverse kinematics, velocity kinematics are manually calculated for a 6-DOF manipulator arm (arm of the Krabot). Additionally a robot model is designed in Solidworks and simulation of the equations are performed in ROS and Gazebo environment.

We have designed a bio-inspired robot (Krabot) to pick objects, place the same objects on a dumpster and move swiftly avoiding collisions with the obstacles.

Project Goals:

- Understand and implement the core concepts of robot modelling. Robot modelling consists of forward kinematics, inverse kinematics, velocity kinematics and trajectory planning.
- Design a Robot Model using Solidworks and exporting the URDF.
- Calculations of equations manually for kinematics of the Robot specifically forward kinematics, inverse kinematics, velocity kinematics.
- Validation of all the equations in the ROS/Gazebo environment.
- Design and Simulate a Robot which can work in real time to pick objects, place the same objects on a dumpster and move swiftly avoiding collisions with the obstacles.

Table of Contents

Introduction and Organization:

The word “robot” is precisely 100 years old this year. It was coined by the Czech writer Karel Capek, in a science fiction play (Rossum's Universal Robots) that set the template for a century's machine dreams and nightmares. The recent developments in the field of robotics have progressed from science fiction fantasy to world-bound reality. Robots are in the industry for a while consistently substituting humans to do mundane, hazardous tasks with higher precision.

These technologies deal with automated machines that can take the place of humans in dangerous environments or manufacturing processes, or resemble humans in appearance, behavior, and/or cognition. Thus, we modelled and simulated the Krabot.

This report is organized into seventeen sections, the first section describes a brief context introduction and describes how the report is organized. After this introductory section, the next section describes the reason for choosing the problem statement and our interest in choosing this type of Robot. Section three describes the Robot model specifying the geometrical aspects and the dimensional aspects. It gives a brief idea of the robot’s sensors, motors, joints, and materials which are critical in the design. Some pictures of the robot design are added in this section. Section four summarizes the aptness of the design of the Krabot 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 which 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 Krabot ground drone model. After defining the model assumptions the next two sections mention the inverse kinematics and forward kinematics of the robot and the robot manipulator. The next section reviews the model validation and validates the equations. The twelfth section validates the equations and performs simulation in a physics environment. Section ten explains the validation of implementation. Finally, section eleven gives references and citations.

Motivation:

On 26th April 1986, a nuclear incident shocked the world. The Chernobyl nuclear incident which is considered the biggest nuclear disaster to date involved huge capital ($2 billion) in manpower and equipment, cleaning the most dangerous areas of the plant roof (nuclear debris). This is just one case out of the other hundreds of nuclear accidents (according to Google, data from 2014) which made humans understand the importance of automation in technology. There are several plant areas on numerous sites where manual work cannot be undertaken owing to challenging radiological and conventional safety environments. There is a need for the remote capability for dismantling/deconstruction of industrial plants, size reduction, and waste segregation to enable decommissioning of these areas.

We designed a robot which will be an attempt to solve the apprehension of people working in dangerous and hard to access 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 inevitable for efficient working. The motivation of the project is to understand robot kinematics and practically implement a self-designed robot model in ROS and simulate in the Gazebo environment for hazardous environments. Also as we both are extremely interested in bio-inspired robotics, thus we plan to model a spider looking robot.

Robot Description:

The Krabot ground drone can operate in a high-level radiation nuclear environment. This six-legged bot can move very swiftly in any surroundings that a traditional robot can't move (eg: all-terrain). Moreover, this is not the only task this robot can perform. Such a robot can handle many other problems like working independently or in multi-agent groups to solve problems in dangerous environments.

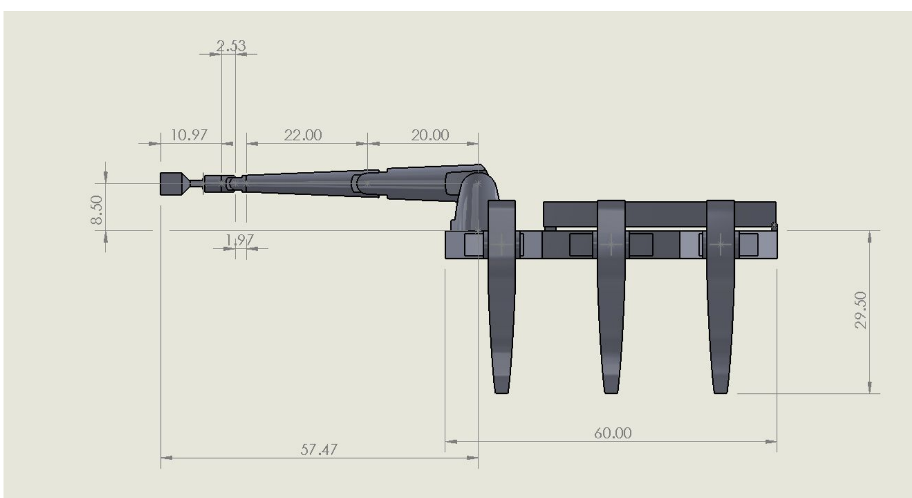

Fig 1

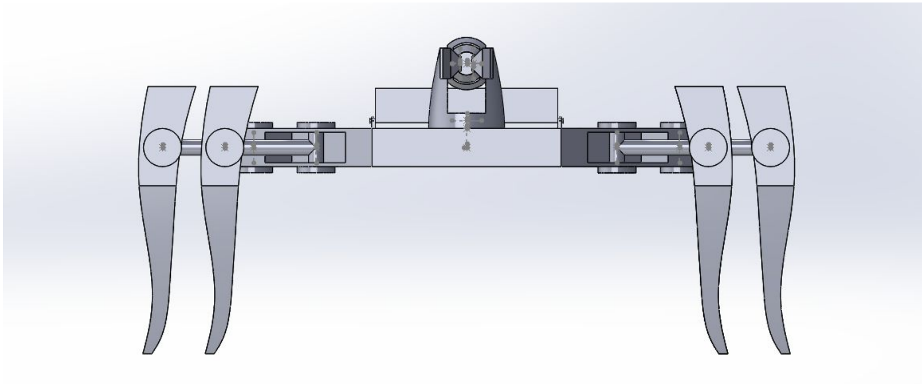

Fig 2

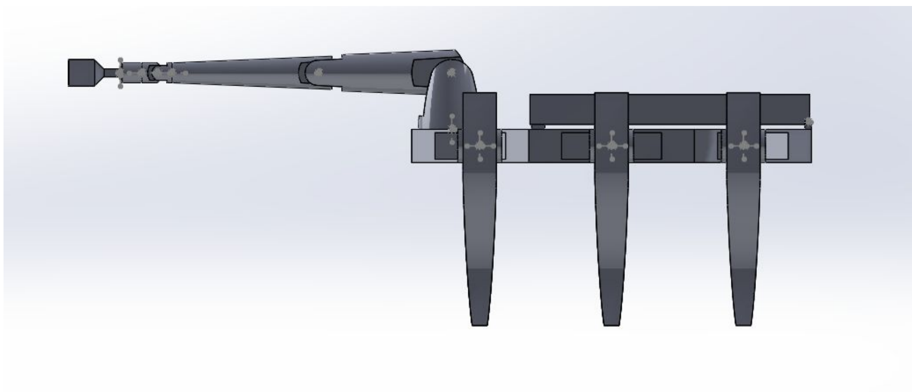

Fig 3

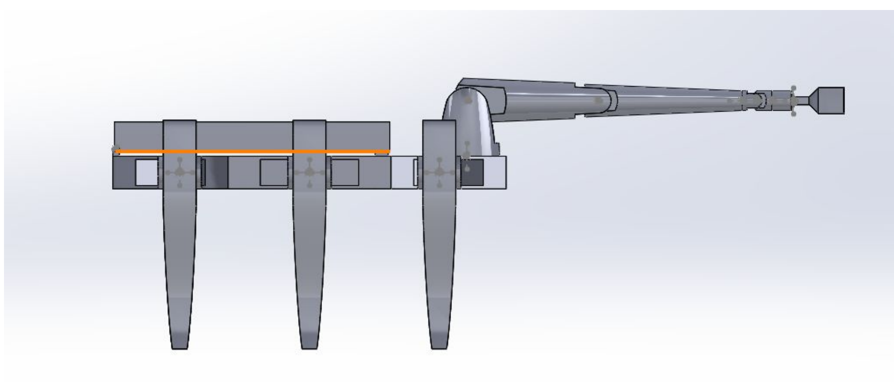

Fig 4

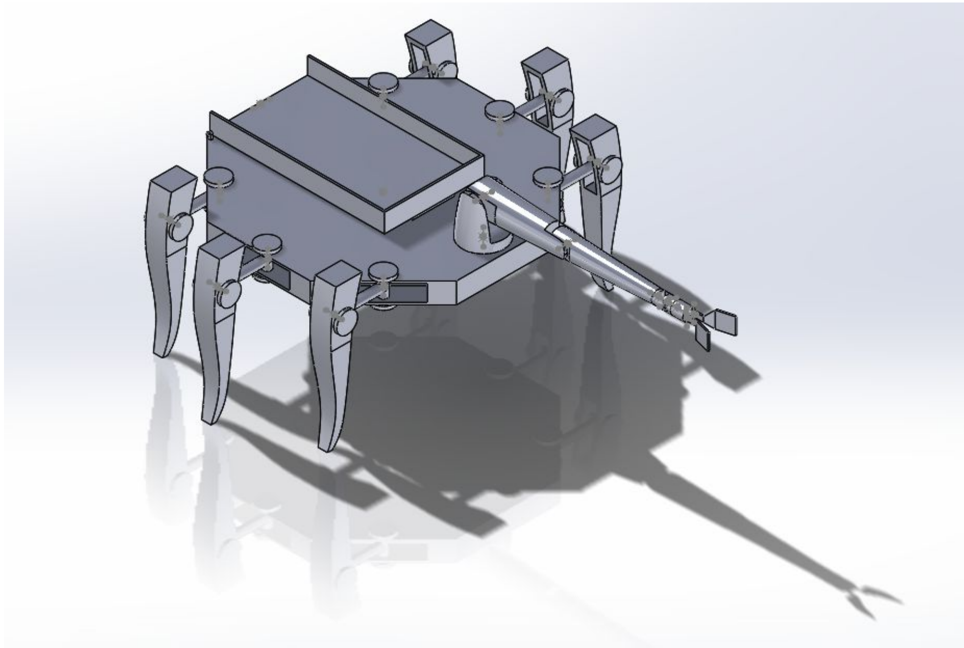

Fig 5

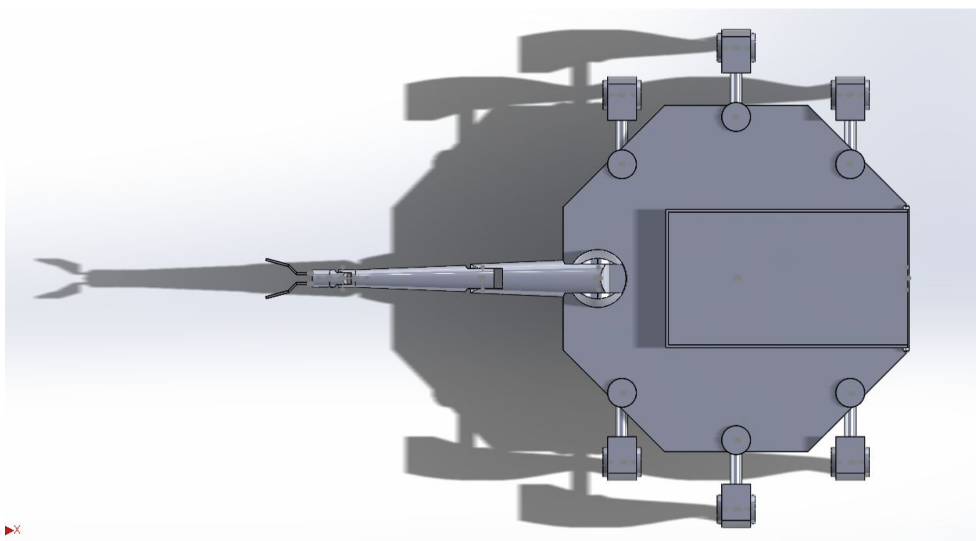

Fig 6

Krabot Theoretical Specifications	
Weight	155 lbs (70 kg) (winout the arm) and 185 lbs (with arm)
Dimensions of the Robot	70 x 60 x 36 inches (without the arm) 70 x 60 x 55 inches (with arm)
Material	Lead coated - Aluminium
Degrees of Freedom of the Bot	12 dof

Degrees of Freedom of the Manipulator Arm	6 dof
Maximum reach	1210 mm of the arm
Payload	7 lb (3.175 kg)
Applications	Material Handling, Working in Hazardous Environment, Material Removal, etc.
Embedded Vision	Camera on the face of the robot (1280 x 800 pixels)
Motors	6 Servo motors driving the legs
Sensors	Gravity, gyroscope, digital compass, pressure, temperature, Force sensors embedded at each joint (stan- dard), plus sonar, accelerometers and range-finding sensors, Lidar
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	Bluetooth controller pad

Table 1: Robot Specifications

Robot Appropriateness for the task:

- There are several reasons for selecting Krabot for the task:
- Ease of device management across its life-cycle. Therefore, easy to build, deploy, maintain, and decontaminate.
- Several industrial robots are capable of doing the pick and place or moving with wheels in the industry, but they can’t move swiftly in all-terrains/ hazardous environments. This can move fast on every kind of surface and can pick and place easily.
- Krabot can be easily installed in a human-occupied environment or hazardous environment. Other Industrial robots are difficult to incorporate in extreme environments.
- Minimal intervention required to deploy and Radiation tolerance.
- Flexible to the surroundings and the environment it moves.
- Ability to use in complex and congested spaces and visualization of hard to access areas.

Scope Description:

- To understand and calculate Forward and Inverse kinematics for any real-world robot
- To learn simulated environments and model the real world in ROS - Gazebo & Rviz.
- To understand link/joint parent-child relationships in CAD/Solidworks and how to model them.
- To understand ROS and its functionalities like “MoveIt”, “Teleop”.
- To do and learn trajectory planning for the manipulator arm and path planning for the Krabot.
- To pick and place objects avoiding obstacles in free space.
- To validate and verify calculated Forward and Inverse kinematics solutions using a developed simulated environment.
- To understand the integration of sensors, controllers in a simulated environment.
- To understand grasping analysis of a manipulator's arm in a real-world scenario.

Scope Appropriateness:

This project made us understand the basic modeling of any robot in the real world. We had a brief understanding of the Kinematic and Dynamic principles. We took this specific spyder-robot to understand rigid body transformation, Inverse Kinematics and Forward Kinematics, contact modeling, and grasping. Modeling this robot made us learn the links and joints mechanism in actual modeling scenarios. Out of other things, this project made us learn the grasping mechanism, trajectory, and path planning. Also, this project made us sure we learn to integrate sensors, controllers in ROS-Gazebo environments. This study covers all the essentials required to model a robot to work in the real world. This project will help all the industries working toward a robot which can work in extreme conditions and hazardous environments.

Scope of Achievement:

- We have achieved the understanding and implementation of the core concepts of robot modelling. Robot modelling consists of forward kinematics, inverse kinematics, velocity kinematics and trajectory planning.
- We have achieved the ability to design a Robot Model using Solidworks and exporting the URDF.
- We have achieved the calculations of equations manually for kinematics of the Robot specifically forward kinematics, inverse kinematics, velocity kinematics.

- We have achieved the validation of all the equations in the ROS/Gazebo environment.
- We have achieved the designing and Simulating of the Robot which can work in real time to pick objects, place the same objects on a dumpster and move swiftly avoiding collisions with the obstacles.

Model Assumptions:

The design of our robot will have 6 legs (like a web-spider). Unlike traditional legs having 3 joints, each for hip, knee, and elbow, we had each leg with 1 hip joint only, which will have 1 DOF and will provide lateral motion to the leg. To give the desired motion to the robot, all 6 legs will be used in proper combination (with the help of cam and follower mechanism). A robot with 6 legs will come with its advantages like the proper balance on uneven surfaces, increased stability, and robust support. To complete the task of cleaning the nuclear debris or pick & place other things, attached a manipulator arm having 6-DOF on the front end of the robot. It will consist of three revolute joints and its end-effector will be a standard 3-DOF (pitch, roll, and yaw motions) that can do user-specific tasks. This structure will give us a 3-DOF manipulator arm.

Some other design/ simulation assumptions:

- All the joints and objects are considered to be rigid.
- The friction and the other external disturbances are not taken into account.
- Robot self-collision is not considered, only collision with the external obstacles are 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.
- The path planning module is not used. Thus, the path planning is performed using forward kinematics by selection of appropriate way points.

Approach to performing the work:

The initial stage of the project focuses on analyzing the specification of our designed model Krabot to understand joint variables to formulate forward and inverse kinematic equations. The forward and inverse kinematic equations are formulated by using the DH parameters concept. In the next stage, we intend to model and design the robot in CAD/Solidworks/Inventor. After the model is designed we will generate the URDF and export it to a gazebo world. We will integrate sensors, controllers, lasers, transmission, and lidar into our robot. We have to modify the URDF for extra links and joints. After that, we will make changes to other files like the launch file, xacro file, etc. We will use “MoveIt” to move the robot using “Teleop”. We used a manual hit and trial method to manipulate the arm of the robot for pick and place operations, after planning the trajectory for the arm manipulator. In the final stage, we would carry out validation of manually calculated forward and inverse kinematic. Validation, the process would be visual in ROS where our robot would pick an object from one spot and place it on another spot after traveling some distance avoiding collision with obstacles. If we have enough time we intend to do a velocity and Jacobian analysis for the Krabot.

Forward Kinematics:

The figure below shows the manipulator configuration in base position with all the dimensions in inches.

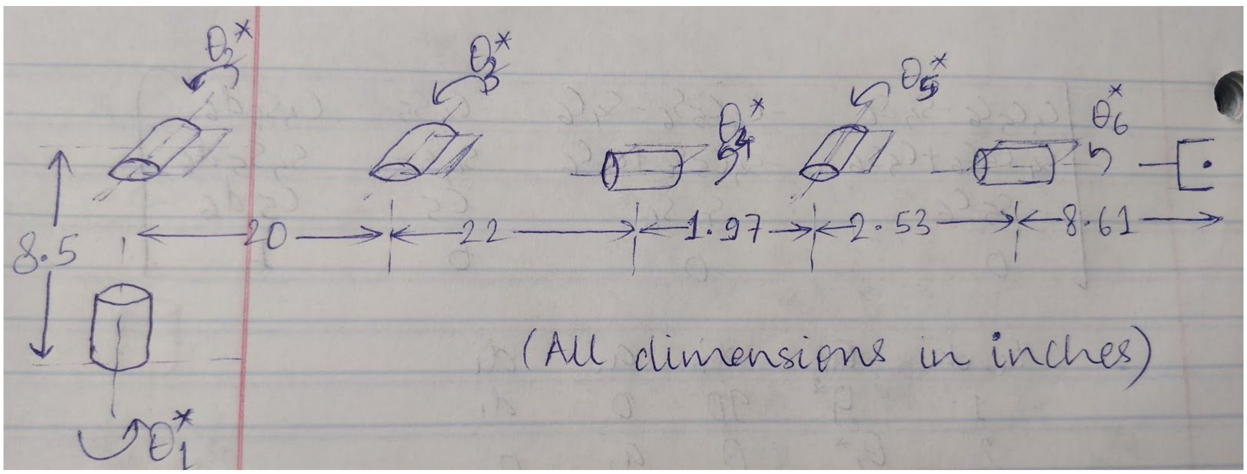

Figure 7: Manipulator base configuration

The manipulator we selected is a standard 6DOF articulated manipulator with 3 revolute joints in articulated configuration for the arm and 3 revolute joints in standard spherical configuration for the wrist. The position of manipulator w.r.t. to the robot's global origin is given in the figure below.

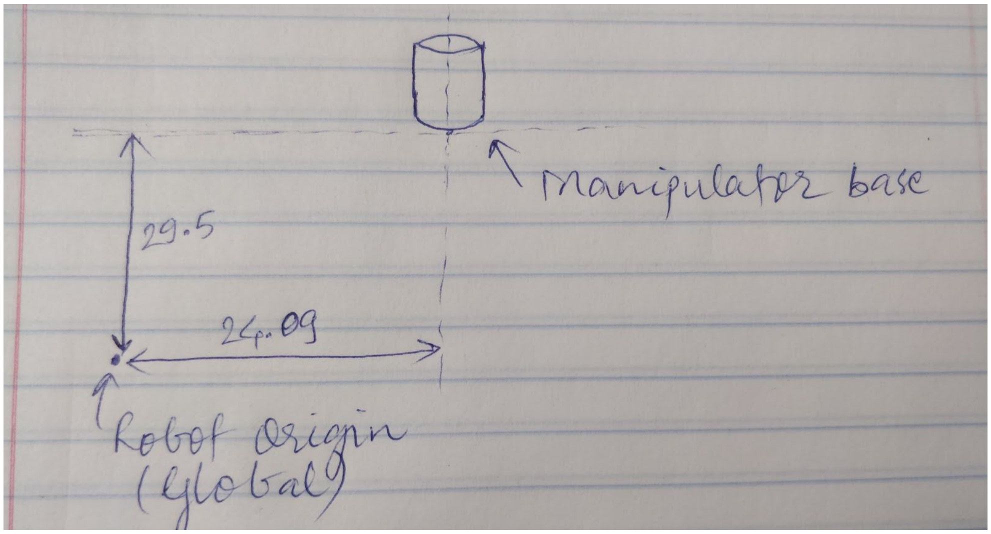

Figure 8: Manipulator position w.r.t. robot's global origin

From the figure, the DH parameters table for the arm can be formulated as:

Link	θ_i	α_i	a_i	d_i
1	θ_1^*	90	0	a_1
2	θ_2^*	0	a_2	0
3	$\theta_3^* + 90$	90	0	0
4	θ_4^*	90	0	$a_3 + a_4$
5	θ_5^*	-90	0	0
6	θ_6^*	0	0	$a_5 + a_6$

Table 2: DH parameters for the arm

The standard homogeneous transformation matrix from frame $i-1$ to frame i using DH method is given by:

$$H_{i-1}^{i} = \begin{bmatrix} c\theta_i & -s\theta_i c\alpha_i & s\theta_i s\alpha_i & a_i c\theta_i \\ s\theta_i & c\theta_i c\alpha_i & -c\theta_i s\alpha_i & a_i s\theta_i \\ 0 & s\alpha_i & c\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

After obtaining all the transformation matrices using above matrix format and multiplying them one by one, we obtain the final transformation matrix H_6^0 as:

$$H_6^0 = \begin{bmatrix} r_{11} & r_{12} & r_{13} & d_x \\ r_{21} & r_{22} & r_{23} & d_y \\ r_{31} & r_{32} & r_{33} & d_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$r_{11} = -c_1c_2s_3c_4c_5c_6 - c_1s_2c_3c_4c_5c_6 + s_1s_4c_5c_6 + c_1c_2s_3s_5c_6 + c_1s_2c_3s_5c_6 + s_1c_4s_6 + c_1c_2s_3s_4s_6 + c_1s_2c_3s_4s_6$$
$$r_{12} = -c_1c_2s_3c_4c_5s_6 - c_1s_2c_3c_4c_5s_6 + s_4s_5c_6 - c_1c_2s_3s_5s_6 - c_1s_2c_3s_5c_6 + s_1c_4c_6 + c_1c_2s_3s_4c_6 + c_1s_2c_3s_4c_6$$
$$r_{13} = c_1c_2s_3c_4s_5 + c_1s_2c_3c_4s_5 + s_1s_4s_5 + c_1c_2s_3c_5 + c_1s_2c_3c_5$$
$$r_{21} = -s_1c_2s_3c_4c_5c_6 - s_1c_2s_3c_4c_5c_6 - c_1s_4c_5c_6 + s_1c_2s_3s_5c_6 + s_1c_2s_3s_5c_6 + c_1c_4s_6 + s_1c_2s_3s_4s_6 + s_1c_2s_3s_4s_6$$
$$r_{22} = s_1c_2s_3c_4c_5s_6 + s_1c_2s_3c_4c_5s_6 + c_1s_4c_5s_6 - s_1c_2s_3s_5s_6 - s_1c_2s_3s_5s_6 - c_1c_4c_6 + s_1c_2s_3s_4c_6 + s_1c_2s_3s_4c_6$$
$$r_{23} = s_1c_2s_3c_4s_5 + s_1c_2s_3c_4s_5 + c_1s_4s_5 + s_1c_2s_3c_5 + s_1c_2s_3c_5$$
$$r_{31} = c_2c_3c_4c_5c_6 - s_2s_3c_4c_5c_6 - c_2c_3s_5c_6 + s_2s_3s_5c_6 - c_2c_3s_4s_6 + s_2s_3s_4s_6$$
$$r_{32} = c_2c_3c_4c_5s_6 - s_2s_3c_4c_5s_6 - c_2c_3s_5s_6 + s_2s_3s_5s_6 - c_2c_3s_4c_6 + s_2s_3s_4c_6$$
$$r_{33} = c_2c_3c_4s_5 - s_2s_3c_4s_5 - c_2c_3c_5 + s_2s_3c_5$$
$$d_x = a_2c_1c_2 + (a_3 + a_4)(c_1c_2s_3 + c_1s_2c_3) + (a_5 + a_6)(c_1c_2s_3c_4s_5 + c_1s_2c_3c_4s_5 + s_1s_4s_5 + c_1c_2s_3c_5 + c_1s_2c_3c_5)$$
$$d_y = a_2s_1c_2 + (a_3 + a_4)(s_1c_2s_3 + s_1s_2c_3) + (a_5 + a_6)(s_1c_2s_3c_4s_5 + s_1c_2s_3c_4s_5 + c_1s_4s_5 + s_1c_2s_3c_5 + s_1c_2s_3c_5)$$
$$d_z = a_1 + a_2s_2 - (a_3 + a_4)(c_2c_3 - s_2s_3) + (a_5 + a_6)(c_2c_3c_4s_5 - s_2s_3c_4s_5 - c_2c_3c_5 + s_2s_3c_5)$$

Where $c_i = cos(\theta_i)$ and $s_i = sin(\theta_i)$.
After providing the corresponding joint angles, the end effector position and orientation can be found out.

Inverse Kinematics:

As we have used a standard spherical wrist in our manipulator, we have the advantage of using the concept of Kinematic Decoupling for formulating the Inverse Kinematics. Thus we split the manipulator into two parts as arm and wrist and then calculate the parameters for them seperately.

The figure below shows the assigned coordinate frames for the articulated arm of the manipulator.

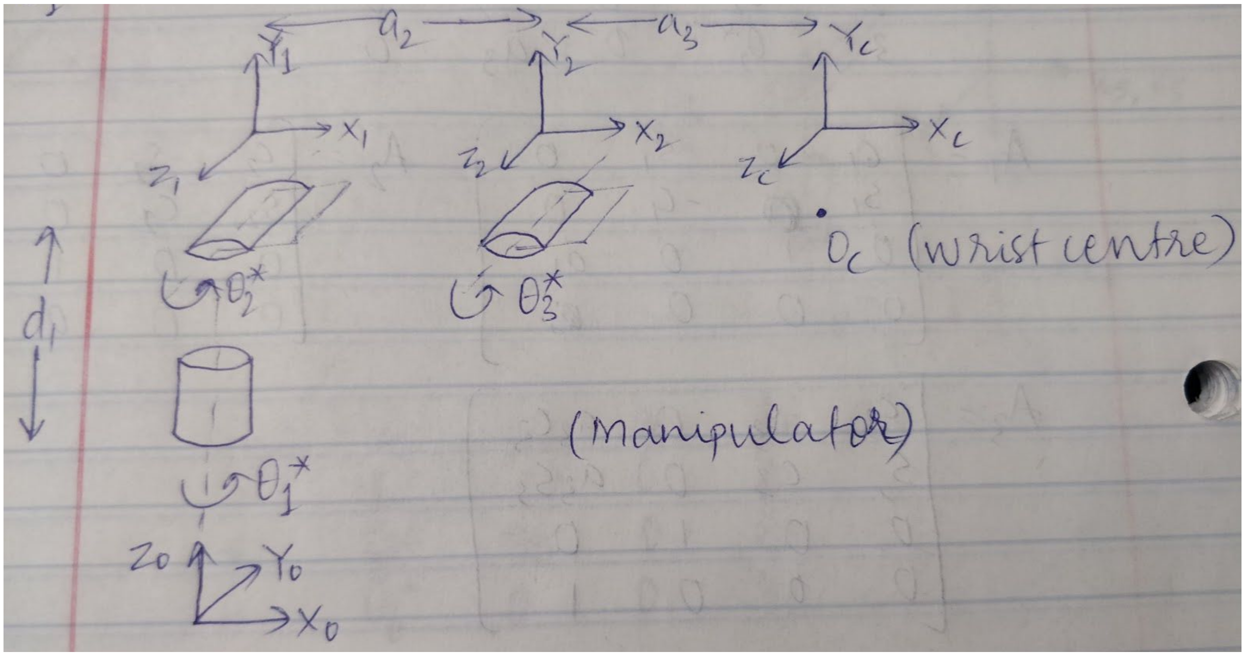

Figure 9: Base position for the manipulator arm alone

From the figure, the DH parameters table for the arm can be formulated as:

Link	θ_i	α_i	a_i	d_i
1	θ_1^*	90	0	d_1
2	θ_2^*	0	a_2	0
3	θ_3^*	0	a_3	0

Table 3: DH parameters for the manipulator arm

Using the above DH table, the homogeneous transformation matrices for the articulated arm can be found as:

$$A_1 = \begin{bmatrix} c_1 & 0 & s_1 & 0 \\ s_1 & 0 & -c_1 & 0 \\ 0 & 1 & 0 & d_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} A_2 = \begin{bmatrix} c_2 & -s_2 & 0 & a_2c_2 \\ s_2 & c_2 & 0 & a_2s_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} A_3 = \begin{bmatrix} c_3 & -s_3 & 0 & a_3c_3 \\ s_3 & c_3 & 0 & a_3s_3 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

In these matrices, the matrix formed by the elements a_{11} to a_{33} denote the corresponding rotation matrix of that link. Thus, by multiplying those matrices, we can get the matrix R_3^0 as:

$$R_3^0 = \begin{bmatrix} c_1c_{23} & -c_1s_{23} & s_1 \\ s_1c_{23} & -s_1s_{23} & -c_1 \\ s_{23} & c_{23} & 0 \end{bmatrix}$$

The figure below shows the assigned coordinate frames for the spherical wrist of the manipulator:

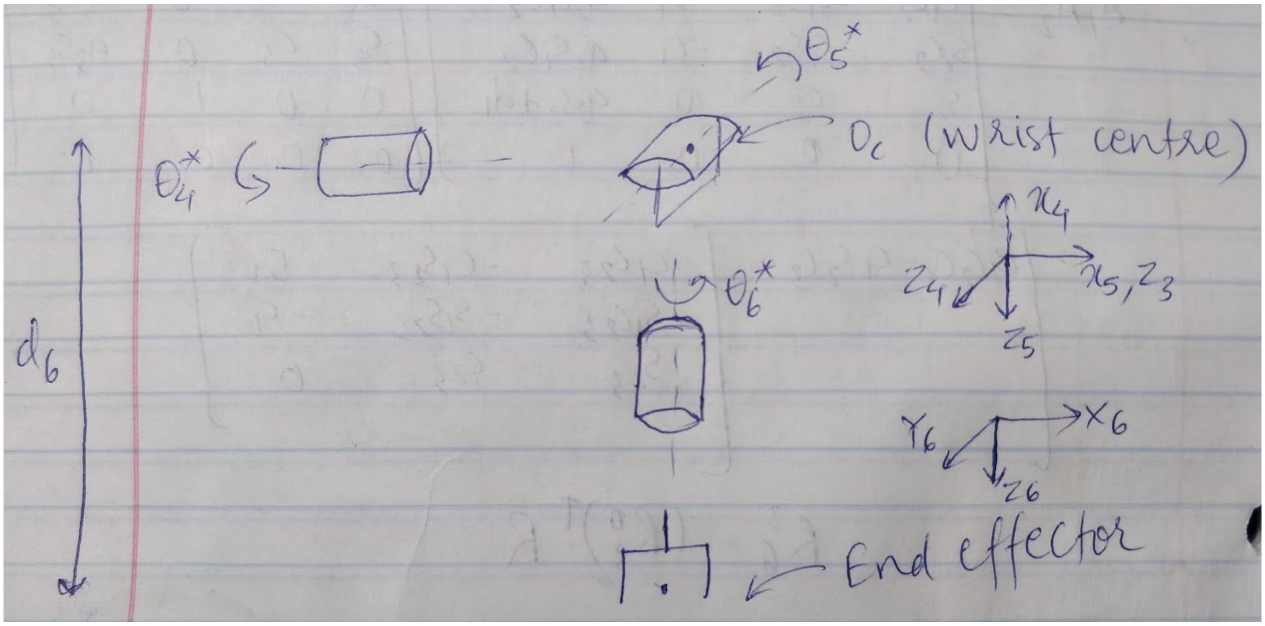

Figure 10: Base configuration for manipulator wrist alone

From the figure, the DH parameters table for the wrist can be formulated as:

Link	θ_i	α_i	a_i	d_i
4	θ_4^*	-90	0	0
5	θ_5^*	90	0	0
6	θ_6^*	0	0	d_6

Table 4: DH parameters for manipulator wrist

Using the above DH table, the homogeneous transformation matrices for the wrist of the manipulator can be found as:

$$A_4 = \begin{bmatrix} c_4 & 0 & -s_4 & 0 \\ s_4 & 0 & c_4 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} A_5 = \begin{bmatrix} c_5 & 0 & s_5 & 0 \\ s_5 & 0 & -c_5 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} A_6 = \begin{bmatrix} c_6 & -s_6 & 0 & 0 \\ s_6 & c_6 & 0 & 0 \\ 0 & 0 & 1 & d_6 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

By multiplying these matrices, the homogeneous transformation matrix R_6^3 can be found as:

$$R_6^3 = \begin{bmatrix} c_4c_5c_6 - s_4s_6 & -c_4c_5s_6 - s_4c_6 & c_4s_5 & c_4s_5d_6 \\ s_4c_5c_6 + c_4s_6 & -s_4c_5s_6 + c_4c_6 & s_4s_5 & s_4s_5d_6 \\ -s_5c_6 & s_5s_6 & c_5 & c_5d_6 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Here, the R matrix formed by the elements a_{11} to a_{33} gives the orientation of the wrist tip w.r.t. the wrist centre and the matrix formed by the elements a_{14}, a_{24}, a_{34} gives the position of the end effector w.r.t. the wrist centre.

Now, assuming a position matrix of end effector as O_d and orientation matrix as R_d; where

$$O_d = \begin{bmatrix} x_d \\ y_d \\ z_d \end{bmatrix} R_d = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix}$$

The tip of the end effector is at a distance of d_6 from the wrist center (along z_6 axis). Since the distance is along z_6 axis, only the third column of the R_d would be taken into account, which maps the z_6 axis. Therefore, we have the equation: $O_d = O_c + d_6 R_d [0\ 0\ 1]^T$

i.e. $O_c = O_d - d_6 R_d [0\ 0\ 1]^T$

$$\begin{bmatrix} x_c \\ y_c \\ z_c \end{bmatrix} = \begin{bmatrix} x_d \\ y_d \\ z_d \end{bmatrix} - d_6 \begin{bmatrix} r_{13} \\ r_{23} \\ r_{33} \end{bmatrix}$$

i.e. the wrist centre coordinates can be given as:

$$\begin{bmatrix} x_c \\ y_c \\ z_c \end{bmatrix} = \begin{bmatrix} x_d - d_6 r_{13} \\ y_d - d_6 r_{23} \\ z_d - d_6 r_{33} \end{bmatrix}$$

To find the values of θ_1 , θ_2 and θ_3 , we use the geometrical approach. The figures below show our manipulator configuration as seen from Top and Front respectively.

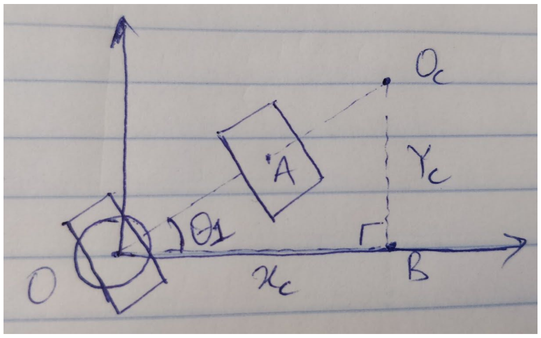

Figure 11: Top View

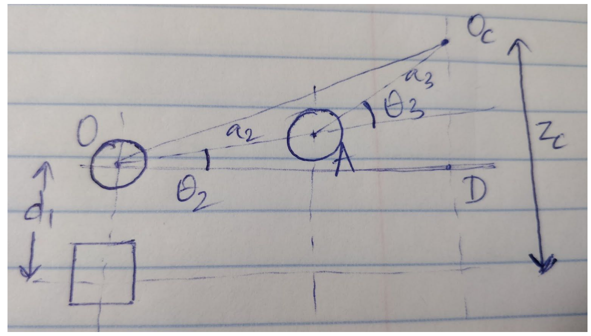

Figure 12: Front View

From these figures, we can formulate some equations as:

$$x_c = [a_2cos(\theta_2) + a_3cos(\theta_2+\theta_3)]cos(\theta_1) \text{ and } y_c = [a_2cos(\theta_2) + a_3cos(\theta_2+\theta_3)]sin(\theta_1)$$

i.e. $tan(\theta_1) = y_c/x_c$

$$\theta_1 = tan^{-1}(y_c/x_c)$$

Also, $z_c = a_1 + a_2sin(\theta_2) + a_3sin(\theta_2+\theta_3)$

Using these equations and also by using the law of cosines, we can find the values of θ_2 and θ_3 as:

$$\theta_2 = tan^{-1}(s/r) - tan^{-1}(a_3s_3/a_2 + a_3c_3)$$

Where $s = z_c - d_1$ and $r = \sqrt{x_c^2 + y_c^2 - d^2}$

$$\theta_3 = tan^{-1}(\sqrt{1-D^2}/D)$$

Where $D = (x_c^2 + y_c^2 - d^2 + (z_c - d_1)^2 - a_2^2 - a_3^2)/(2a_2a_3)$

In the previous equation, we had found out the orientation of the end effector w.r.t. the wrist centre as R_6^3 in terms of θ_4, θ_5 and θ_6 which are unknown parameters at present. Also, we had found out the orientation matrix for the manipulator arm as R_3^0. Now we have the equation:

$$R_d = R_3^0R_6^3$$

pre-multiplying both sides by $(R_3^0)^{-1}$ we get

$$(R_3^0)^{-1}R_d = (R_3^0)^{-1}R_3^0R_6^3 \text{ i.e. } (R_3^0)^TR_d = R_6^3$$

The three equations given by the third column in the above matrix are given by:

$$c_4s_5 = c_1c_{23}r_{13} + s_1c_{23}r_{23} + s_{23}r_{33}$$
$$s_4s_5 = -c_1s_{23}r_{13} - s_1s_{23}r_{23} + c_{23}r_{33}$$
$$c_5 = s_1r_{13} - c_1r_{23}$$

Therefore, $\theta_5 = tan^{-1}(\sqrt{1-(s_1r_{13}-c_1r_{23})^2}/s_1r_{13} - c_1r_{23})$

Also, $\theta_4 = tan^{-1}(- c_1 s_{23} r_{13} - s_1 s_{23} r_{23} + c_{23} r_{33} / c_1 c_{23} r_{13} + s_1 c_{23} r_{23} + s_{23} r_{33})$

And $\theta_6 = tan^{-1}(s_1 r_{12} - c_1 r_{22} / - s_1 r_{11} + c_1 r_{21})$

Given the end effector position and orientation, the joint angles of the manipulator can be found.

Solution Validation:

Validation for the project was done through simulation using ROS in a Gazebo environment. After we calculated forward and inverse kinematic solutions, we created two separate scripts in python for the validation part. The forward Kinematics script would generate the end effector transformation matrix with all the end effector orientation matrix elements and the end effector position elements w.r.t. to the base of the manipulator, given all the link lengths and joint angle states. Similarly, the inverse Kinematics script would output all the joint angle states from manipulator base link to the wrist's last link, given the end effector position and orientation values. We placed an object (which would be considered as the nuclear debris part) in gazebo world at a known location w.r.t. the robots global origin frame. Then we input those values into the IK script to calculate all the joint angles. Then we fed those joint angles in the FK script to check whether the IK solution was correct or not.

The python code for FK and IK scripts are given below:

```
import numpy as np

th1 = 0.81, th2 = 0, th3 = 1.22, th4 = 0, th5 = -0.89, th6 = 0

a1 = 8.5, a2 = 20, a3 = 22, a4 = 1.97, a5 = 2.53, a6 = 8.61

c1 = np.cos(th1), c2 = np.cos(th2), c3 = np.cos(th3)
c4 = np.cos(th4), c5 = np.cos(th5), c6 = np.cos(th6)

s1 = np.sin(th1), s2 = np.sin(th2), s3 = np.sin(th3)
s4 = np.sin(th4), s5 = np.sin(th5), s6 = np.sin(th6)

a = -c1*c2*s3 - c1*s2*c3, b = -s1*c2*s3 - s1*s2*c3, c = c2*c3 - s2*s3

r11 = c5*c6*(a*c4 + s1*s4) - a*s5*c6 - a*s4*s6 + s1*c4*s6
r12 = c5*s6*(a*c4 + s1*s4) + a*s5*s6 - a*s4*c6 + s1*c4*c6
r13 = -s5*(a*c4 + s1*s4) - a*c5
r21 = c5*c6*(c4*b - c1*s4) - b*s5*c6 - s6*(s4*b + c1*c4)
r22 = -c5*s6*(c4*b - c1*s4) + b*s5*s6 - c6*(s4*b + c1*c4)
r23 = -s5*(c4*b - c1*s4) - b*c5
r31 = c*c4*c5*c6 - c*s5*c6 - c*s4*s6
r32 = -c*c4*c5*s6 + c*s5*s6 - c*s4*c6
r33 = -c*c4*s5 - c*c5
dx = (a5+a6)*r13 + a2*c1*c2 - a*(a3+a4)
dy = (a5+a6)*r23 + a2*s1*c2 - b*(a3+a4)
dz = (a5+a6)*r33 + a1 + a2*s2 -c*(a3+a4)
```

Figure 13: Forward Kinematics Python script

```python
import numpy as np

th1 = 0, th2 = 0, th3 = 0, th4 = 0, th5 = 0, th6 = 0

a1 = 8.5, a2 = 20, a3 = 22, a4 = 1.97, a5 = 2.53, a6 = 8.61

r11 = 1, r12 = 0, r13 = 0
r21 = 0, r22 = 1, r23 = 0
r31 = 0, r32 = 0, r33 = 1

dx = 30, dy = -20, dz = -50

th1 = np.arctan2(dy, dx)

s = dz - a1, r = np.sqrt(dx*dx + dy*dy - a2*a2)
D = (dx*dx + dy*dy - a2*a2 + s*s - a2*a2 - a3*a3)/(2*a2*a3)

th3 = np.arctan2(np.sqrt(1-D*D), D)

s1 = np.sin(th1), c1 = np.cos(th1), s3 = np.sin(th3), c3 = np.cos(th3)

th2 = np.arctan2(s, r) - np.tan(((a3*s3)/a2) + a3*c3)

th5 = np.arctan2(np.sqrt(1-np.square(s1*r13 - c1*r13)), s1*r13 - c1*r13)

s23 = np.sin(th2+th3), c23 = np.cos(th2+th3)

th4 = np.arctan2(-c1*s23*r13 - s1*s23*r33 + c23*r33,
    c1*c23*r13 + s1*c23*r23 + s23*r33)

th6 = np.arctan2(s1*r12 - c1*r22, -s1*r11 + c1*r21)
```

Figure 14: Inverse Kinematics Python Script

Simulations:

We did the simulations for two different purposes. First was for the pick and place operation of the manipulator and the second was for giving motion to the legs of the robot. For the first part, after successfully validating the FK and IK solution, we fed those joint angle values to the joint state controllers of the manipulator one at a time to execute the pick and place operation. Immediately after starting the simulation, the manipulator would get into its base position and then start moving as per commands. After picking the object, we did not actually do proper trajectory planning for the end effector to travel till the dumping tray, as it is out of the scope of this project; instead we provided a bunch of way points from its object picking position till its object dropping position for smooth motion. The following figures show some snaps of the simulation of pick and place operation.

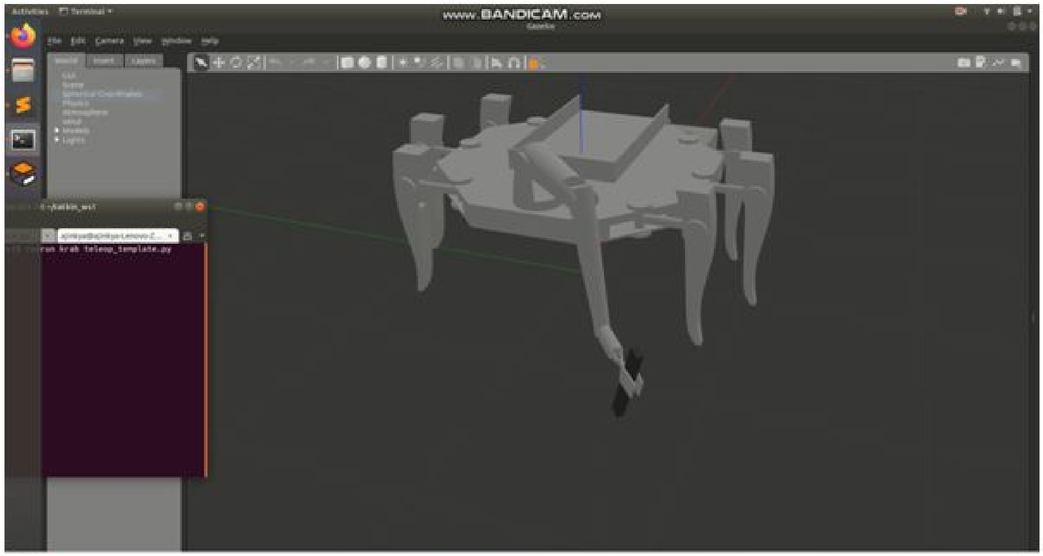

Figure 15: Manipulator picking up the object

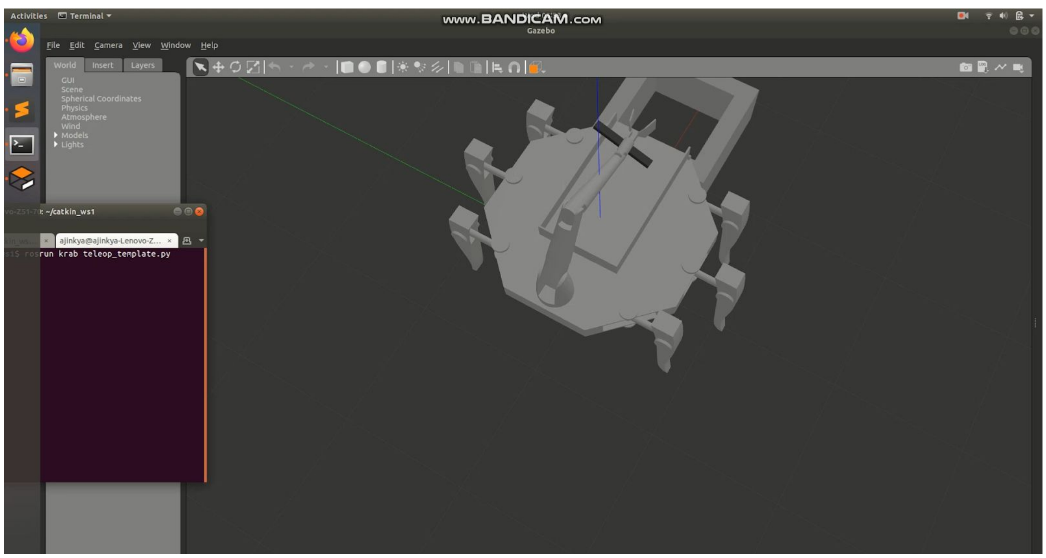

Figure 16: Manipulator dropping the object in the dumping tray

For the second part, we only focused on giving motion to the legs of the robot and thus did not provide any joint angle values to the manipulators' controllers. For the legs motion, we provided joint angle values sequentially to the leg first (to lift it), then to the connecting link (to move the leg forward) and then again to the leg in reverse direction (to put the leg back on ground). Feeding these commands sequentially one by one to all the legs would result in the robot walking in the desired direction. The following figure shows a snap of the simulation of legs motion.

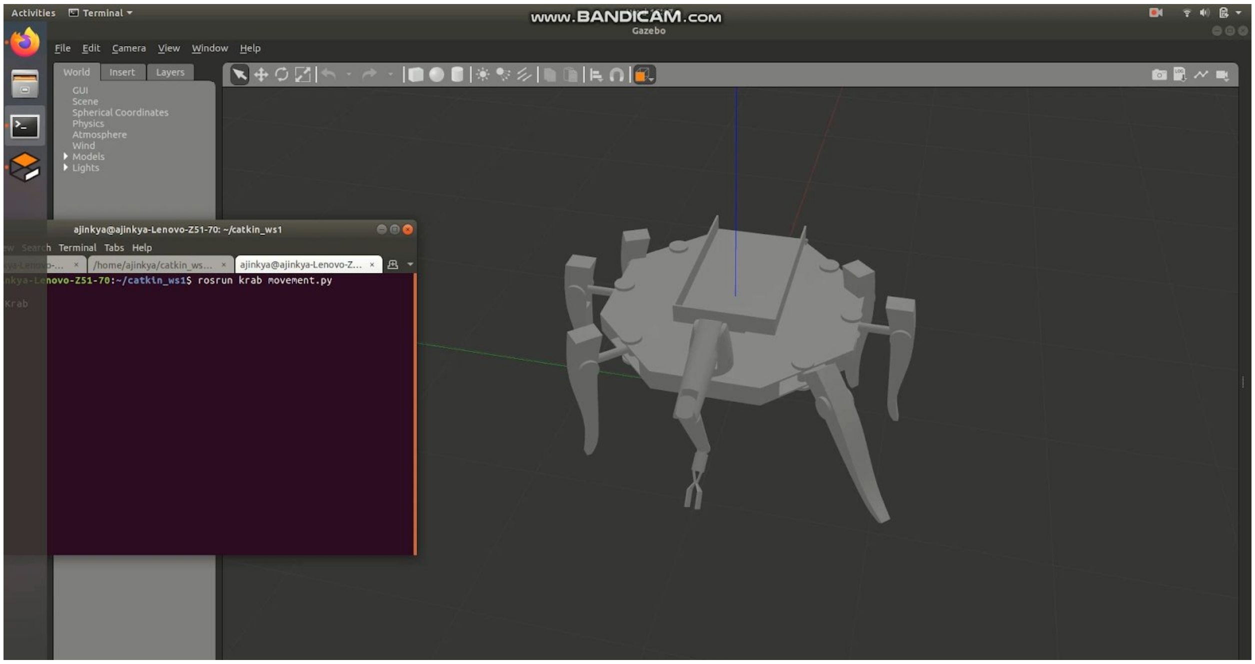

Figure 17: Manipulator legs motion

Challenges:

During the project, we faced three major challenges like manipulator links abnormal movement, robots legs not moving as expected and presence of multiple solutions of the Inverse Kinematics. To overcome the challenge of properly moving the manipulator links, we had to adjust the mass value of the links and their velocity of motion. 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 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 proper motion of the legs.

The biggest 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 Inverse Kinematics script for each of the possible answer pattern to obtain the desired solution and reject the non feasible solutions.

Scope of Future Works:

- Spherical joints on the hips and on the wrist of the arm manipulator.
- Jamming wrists with lead based soft material for easy lifting purpose.
- Knee joints and hip joints, so that overall DOF is more for the robot for easy movements.
- Designing smaller robots (multi-agent system), working in coordination with each other replacing one big robot with six to eight small powerful robots reducing the investment

cost.

- Designing a similar robot with ability to fold legs to swim in hazardous chemicals.
- Placing another arm manipulator on the top of the robot, so that it can perform lots of other tasks like bomb diffusion etc.

Conclusions:

- The project provided valuable understanding of the present trend in industry and how robots work collaboratively with humans.
- This project honed our skills in Solidworks and exporting URDF.
- Through this project we were able to gain in depth knowledge about robot kinematics and learned the modified DH convention.
- We worked on trajectory planning to avoid obstacles for the robot and able to achieve the goal to pick and place objects.
- Implemented and calculated robot kinematic equations.
- Learned the new concepts of bio-inspired robots and their functionality.
- Learned the simulation of the Robot in the ROS 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 and Mr. Vishnuu Appaya Dhanabalan for their continuous support and feedback.

Citations:

- 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
- Ioannis Tsitsimpelis, C. JamesTaylor, 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
- 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.

- Servet Soyguder and Hasan Alli, “Design and prototype of a six-legged walking insect robot”, Industrial Robot: An International Journal34/5 (2007) 412 – 422.