ENPM662 Project 2 Report

FIRE FIGHTING ROBOT

Authors: Alvina Alex (118414553), Rishabh Singh (117511208)

Project Mentors: Dr. Monfaredi, Ajinkya and Karan

FIRE FIGHTING ROBOT

1. Introduction 3

2. Application 4

3. Robot Type 5

4. DOF's and Dimensions 6

4.1. DOF's 6

4.2. Dimensions 7

5. Finite Element Analysis 8

5.1. Mobile Robot 8

5.1.1. Body 8

5.1.2. Steering Knuckle 9

5.2. 3-R Arm 10

5.2.1. Link-1 10

5.2.2. Link 2 11

5.2.3. Link 3/End effector nozzle 11

6. Kinematic Analysis 12

6.1. DH Parameters 12

6.2. Forward Kinematics 13

6.3. Inverse Kinematics 17

6.4. Workspace Study 21

7. Assumptions 22

7.1. Modelling Assumptions 22

7.2. Simulated Environment Assumptions 22

8. Control Method 22

9. Gazebo and Rviz visualisation Results: 24

9.1. Laid back goals: 24

9.2. Ambitious Goals: 25

10. Problems faced 26

11. Lessons learned 27

12. Conclusion 28

13. Future Work **28**

14. References **29**

15. Supplemental Material **29**

16. Contributions **29**

1. Introduction

Fig 1. A newspaper clipping as our motivation (**Source:** FirefighterCloseCalls.com)

Fires propagate at an exponential speed which makes it utterly necessary to be quick in action. Since structures like houses and buildings tend to collapse , it is tough to reach the source of the disaster to extinguish it. The significant factors required to focus on these situations are speed, robustness, the safety of firefighters, and communication. [5]

The USA is known to have one of the highest fire death rates in the industrialized world during the past decade. According to the 2019 reports, the local fire departments all over the country have responded to an estimate of 1,291,500 fires, which have caused an estimated 3,704 civilian deaths, 16,600 civilian injuries, and $14.8 billion in direct property damage [1]. To worsen matters, career firefighters have experienced an average of 20,650 fire-ground injuries each year from 2015 through 2019 and have lost 140 firefighters while on the job in 2020. The adversity does not end, firefighters, who survive, experience long term consequences from primary fire ground injuries like strains or sprains (27 percent), smoke inhalation (18 percent), the pain only(12 percent), thermal burns (7 percent), and cuts or lacerations (5 percent) [2]. The above facts are quite disappointing considering the technological advancements of this era.

UNIVERSITY OF MARYLAND

In this paper, we propose a compact and portable emergency responder robot that assists firemen in fighting high-rise fires, especially in highly dangerous environments where it is not safe for people to enter, called the Fire-fighting Robot. The Fire-fighting robot is custom-designed to be modular, fire-resistant, and robust. Operated by remote belly-pack controllers, users are provided a real-time video feed allowing them to traverse hazardous terrain and push obstacles from their path while withstanding extreme elements. It utilizes a lidar to get depth knowledge of its surroundings in real-time. The robot is also programmed to protect itself by keeping a specified distance from the center of hazard [3].

This robot will reduce the risk of injury for firefighters and possible victims and decrease the monetary losses which increase considerably as fire duration increases.

2. Application

Fig 2. Existing robots doing the job (**Source:** www.robots.ieee.org)

UNIVERSITY OF MARYLAND

Certain tasks are in account when creating robotic firefighting systems. These tasks include evaluating and finding flames, conducting search and rescue operations, monitoring hazardous variables, and fire management and suppression [5].
Numerous lives can be saved by employing robots in hazardous situations. These include the lives of those affected by the disaster and the people working to save them such as firefighters. This product can be very useful in times of emergency when it is very dangerous for humans such as the retrieval of hazardous materials.

The fire-fighting robot is designed to mitigate life-threatening situations with tools required for fire suppression, situational awareness, and intelligence gathering. It can identify the location of hazards and operate remotely, giving continuous real-time feedback to the operator. It will sense the fire and spray fire-suppression liquids and chemicals onto that direction until the fire subsides. [3]

3. Robot Type

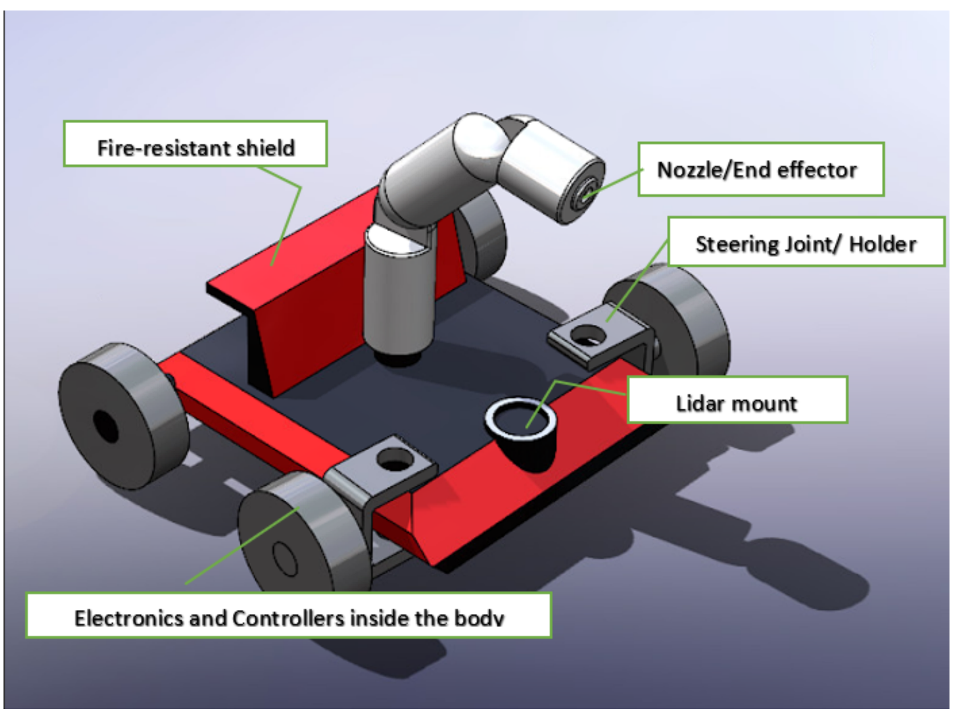

Fig 3. Simplified model of our fire-fighting bot

The proposed model is a ground vehicle with a rear drive transmission. The front two wheels are used to steer the robot. A 3-R spatial arm, which is attached to the body of the mobile robot, carries a nozzle at its end effector. The joint attached to the robot's base, controls the yaw of the nozzle, while the other two revolute joints contribute to the pitching and positioning of the nozzle. The 3-R arm will aid in accessing elevated areas, consequently increasing the workspace.

Inspired by the Thermite Robot, created by Howe x Howe, the design incorporates a high-temperature resistant and fire protection shield that is used to protect the controllers and back of the robot. The protective shield is made from intumescent materials that expand and become denser when exposed to fire or intense heat [3]. One of the main goals of creating a fire-fighting robot is its capability to access areas where it is unsafe for firefighters to enter. The robot can be remotely controlled by an operator at a safe distance using lidar, which is placed in a cavity in front of the robot. Even though the robot is controlled remotely, the robot is programmed to protect itself by keeping a 2-meter distance from the center of the hazard.

Lidar stands for Light Detecting And Ranging which is a remote sensing method that utilizes laser pulses to measure the ranges of an object. A recent discovery by the researchers at the National Institute of Standards and Technology has used the Lidar system to image three-dimensional objects melting in flames [4]. They concluded that this method offers a precise, safe, and compact way to measure structures as they collapse in a fire. The ultimate goal of the robot is to extinguish fires without the intervention of human firefighters with a pressure nozzle on a hydraulic arm.

4. DOF's and Dimensions

4.1. DOF's

The robot can be majorly divided into two categories based on its functions:

1. The mobile rear-wheel drive

2. The 3-R spatial arm

The mobile rear-wheel drive has 2 degrees of freedom. It travels in the x-y plane which is essentially the ground surface.
The 3-R spatial arm has 3 degrees of freedom. The joint attached to the robot's base controls the yaw of the nozzle, while the other two revolute joints contribute to the pitching and positioning of the nozzle.

4.2. Dimensions

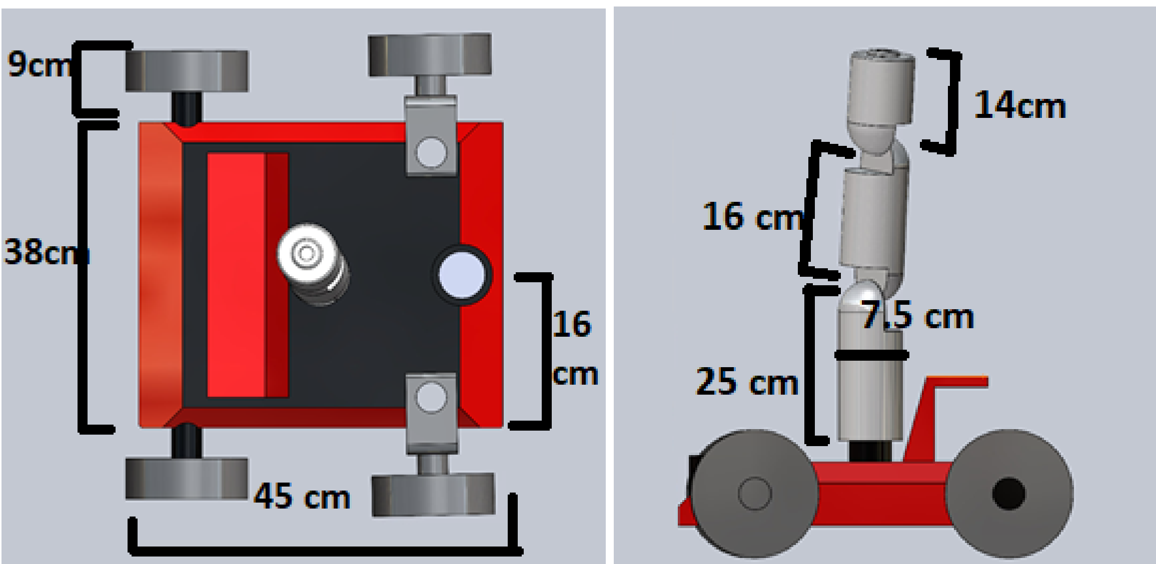

Fig 4. Annotations for the bot. All dimensions are in centimeters.

UNIVERSITY OF MARYLAND

5. Finite Element Analysis

5.1. Mobile Robot

5.1.1. Body

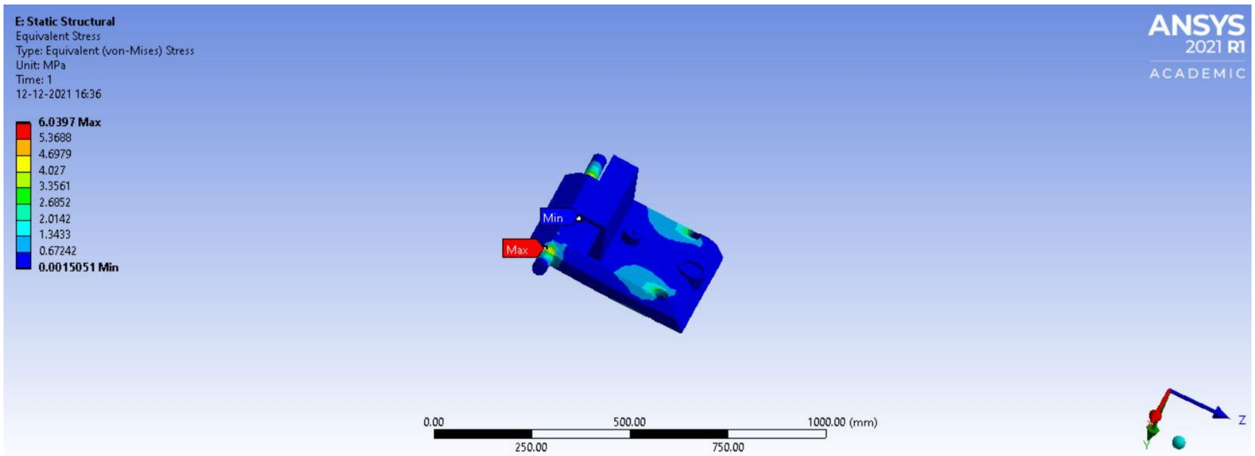

Fig 5. Body drop test stress analysis (Von-mises criterion)

This stress test shows the yield during the drop test of the robot from 4 m. It can be seen that the maximum stress is around 6 MPa. Since, we are not gonna carry loads more than 15 kg and also the max speed is around 8 km/h, drop test qualifies for the worst case. Basically 4 times the load of the vehicle is applied at the hard points.

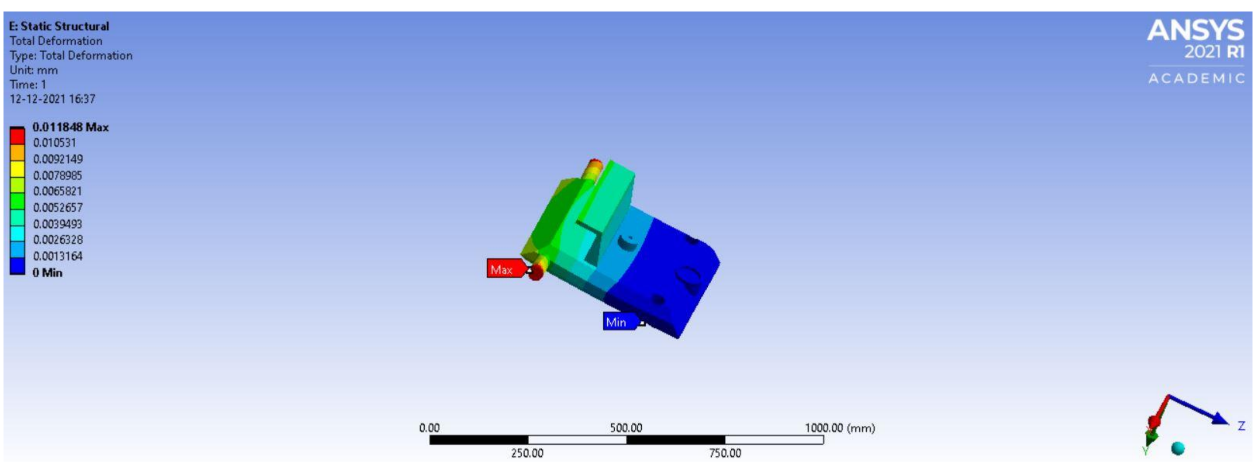

Fig 6. Body drop test total deformation

UNIVERSITY OF MARYLAND

5.1.2. Steering Knuckle

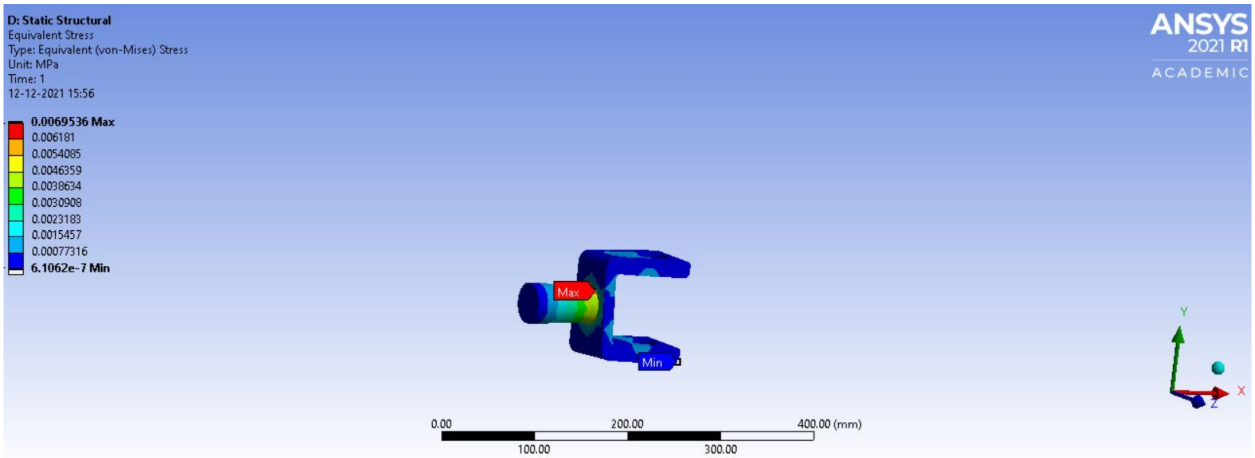

Fig 7. Steering knuckle test stress analysis (Von-mises criterion)

We have considered the worst situation when the wheels are stuck and the motor is still trying to rotate it using its full stall torque capacity which is around 25 N-m.

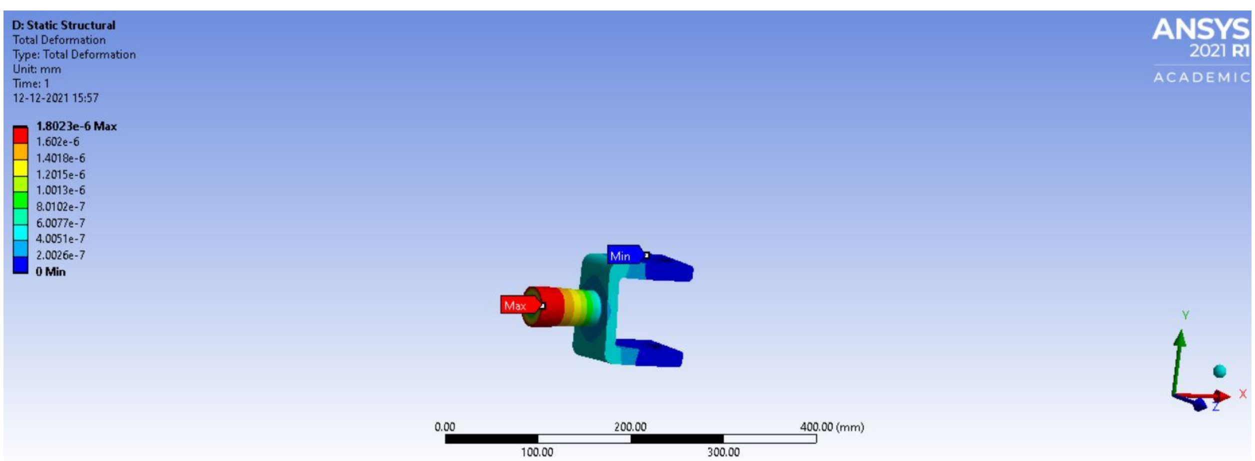

Fig 8. Steering knuckle test total deformation

5.2. 3-R Arm

5.2.1. Link-1

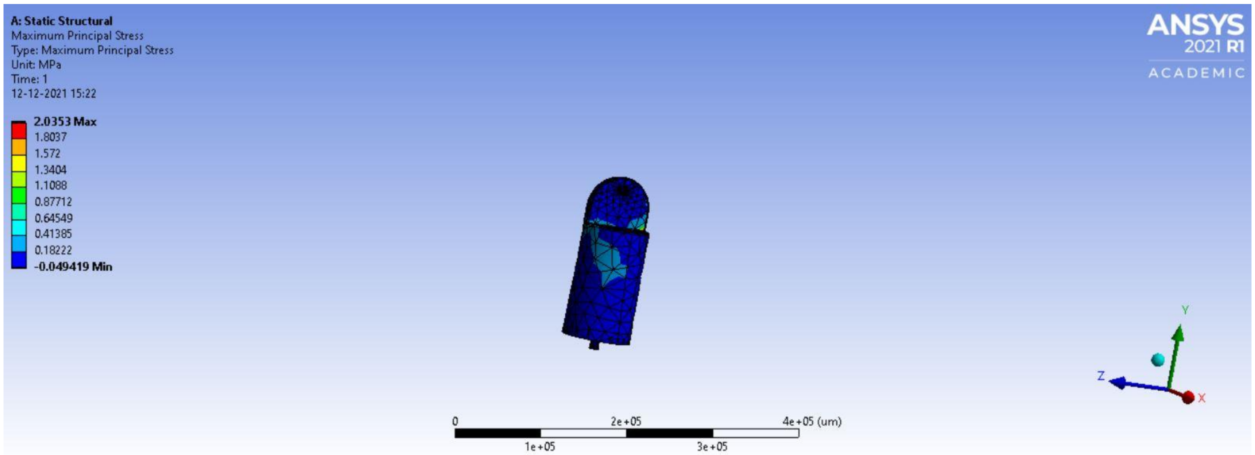

Fig 9. Link-1 test maximum principle stress analysis

Here we have considered the worst situation when the links are stuck and the motor is still trying to rotate it using its full stall torque capacity which is around 20 N-m.

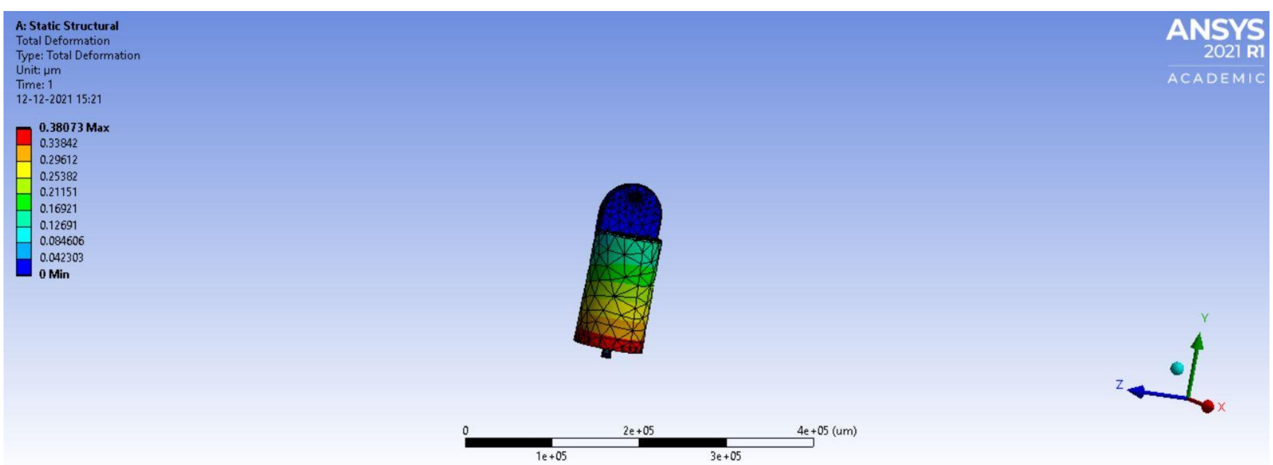

Fig 10. Link-1 total deformation

5.2.2. Link 2

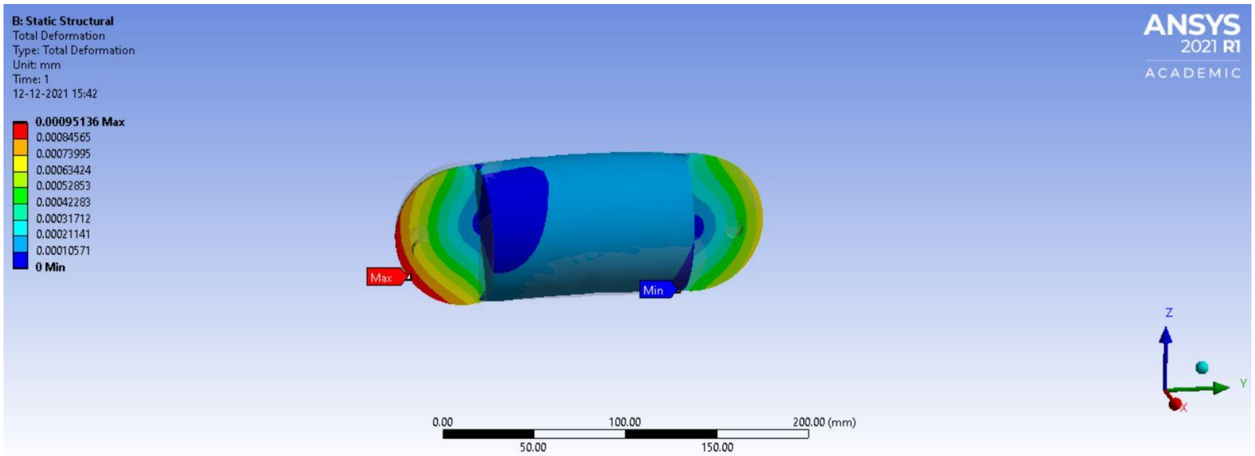

Fig 11. Link-2 total deformation

The method of analysis is the same as what we have described in the above sections. The motor stall torque is assumed to be around 18 N-m.

5.2.3. Link 3/End effector nozzle

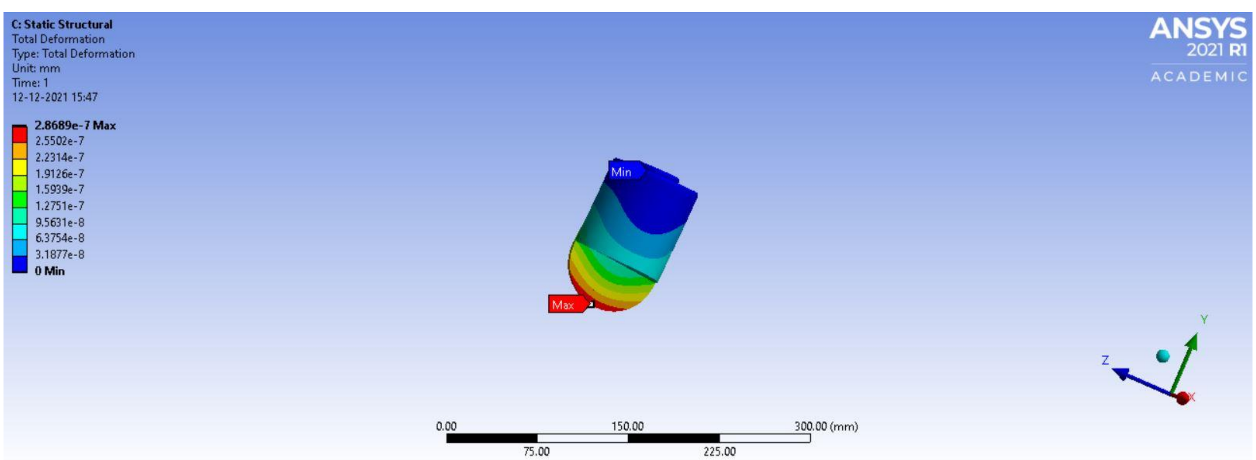

Fig 12. Link-3 total deformation

UNIVERSITY OF MARYLAND

The method of analysis is the same as what we have described in the above sections. The motor stall torque is assumed to be around 10 N-m.

In all the cases, we can see that the total deformation is in microns, so as a conclusion the design of the arms, chassis and steering knuckle as well as the selection of material gives us enough FOS (Factor of Safety) to make this Robot commercial with the life of around 2-3 years and 1 lacs of kilometers in the terms of strength. There can be other operational hitches like motors, electrical wirings etc. in between which require further validation but are out of scope for this project's study.

6. Kinematic Analysis

6.1. DH Parameters

We are having a 3-R manipulator, hence we derived the DH table for the arms from the base location with angles for the joint from the base location as θ_1, θ_2 and θ_3. For arm link lengths d = 25.4 cm, e = 16 cm and f = 16 cm

DH Table:

θ	α	d	a
θ_1	90 deg	25.4 cm	0
θ_2	0	0	16
θ_3	0	0	16

Table 1. DH parameter table

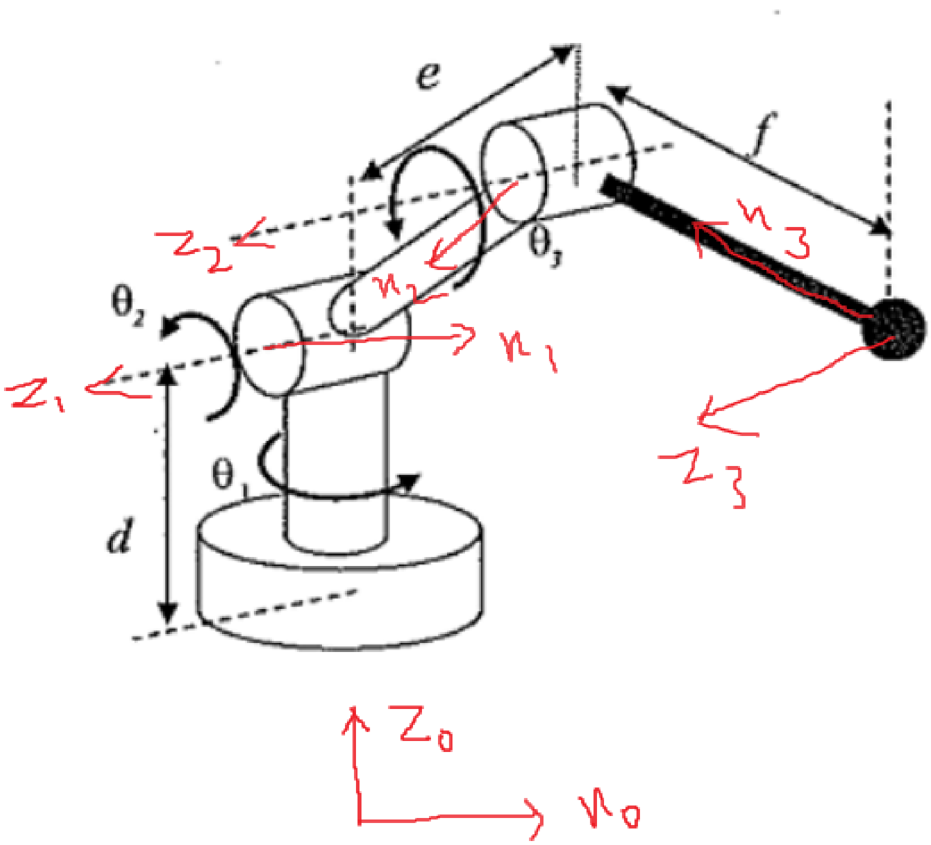

Fig. 13. 3R Manipulator with considered axes diagram

6.2. Forward Kinematics

Transformation matrix for the end effector position is as given below:

UNIVERSITY OF MARYLAND

T_final =

$$
\begin{bmatrix}
-\sin(\theta_2)\sin(\theta_3)\cos(\theta_1)+\cos(\theta_1)\cos(\theta_2)\cos(\theta_3) & -\sin(\theta_2)\cos(\theta_1)\cos(\theta_3)-\sin(\theta_3)\cos(\theta_1)\cos(\theta_2) & \sin(\theta_1) & -10\sin(\theta_2)\sin(\theta_3)\cos(\theta_1)+10\cos(\theta_1)\cos(\theta_2)\cos(\theta_3)+16\cos(\theta_1)\cos(\theta_2) \\
-\sin(\theta_1)\sin(\theta_2)\sin(\theta_3)+\sin(\theta_1)\cos(\theta_2)\cos(\theta_3) & -\sin(\theta_1)\sin(\theta_2)\cos(\theta_3)-\sin(\theta_1)\sin(\theta_3)\cos(\theta_2) & -\cos(\theta_1) & -10\sin(\theta_1)\sin(\theta_2)\sin(\theta_3)+10\sin(\theta_1)\cos(\theta_2)\cos(\theta_3)+16\sin(\theta_1)\cos(\theta_2) \\
\sin(\theta_2)\cos(\theta_3)+\sin(\theta_3)\cos(\theta_2) & -\sin(\theta_2)\sin(\theta_3)+\cos(\theta_2)\cos(\theta_3) & 0 & 10\sin(\theta_2)\cos(\theta_3)+16\sin(\theta_2)+10\sin(\theta_3)\cos(\theta_2)+25.6 \\
0 & 0 & 0 & 1
\end{bmatrix}
$$

Essentially, the forward kinematics is given by the transformation matrix as shown above which is respect to the mobile robot platform base. We are trying to validate the forward kinematics in Gazebo itself by taking the arms to known angles. For first validation, we have kept all the θ to be zero. Which will give the end effector coordinates as, x = 32 cm (approx), y = 0 and z = 25.4 cm (approx). This can be verified by seeing the image below which has been taken from the video we displayed during the class presentation. The x, y and z coordinates are with respect to the ground bot rather than the world or gazebo. The forward direction for the bot is considered as the x axis.

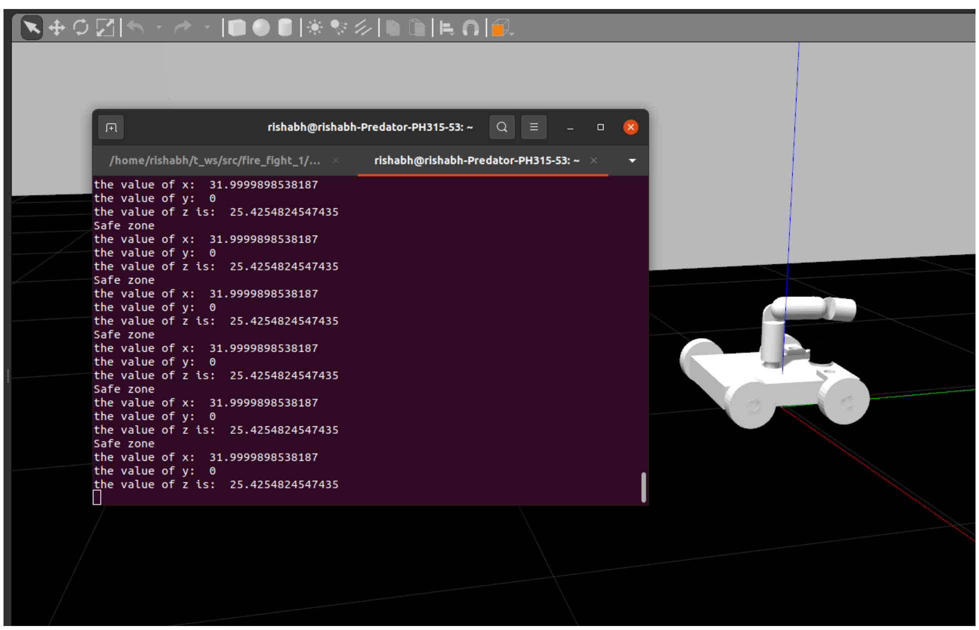

Fig. 14 Forward Kinematics Validation I (Home position, θ_1, θ_2 and θ_3= 0)

The image shown above is at the manipulator's home position, and the values are being displayed using the end effector coordinates taken in parametric form from the final transformation matrix as shown above (T_final) and using the same formula in the teleop python script which apparently works like encoder for us.

Another validation we did is by changing the θ_1 to 90 degrees and as expected the output came out is as follows:

x = 0, y = 32 cm (approx) and z = 25.4 cm (approx) as shown in the image below:

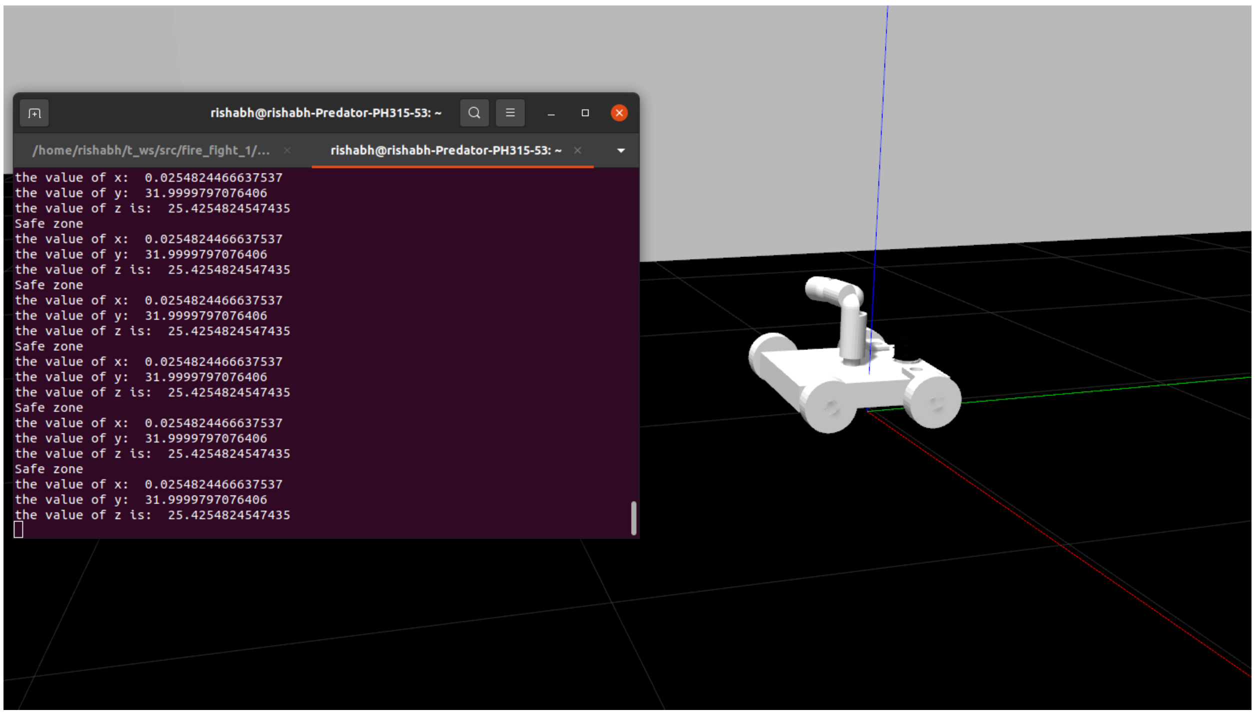

Fig. 15 Forward Kinematics Validation II ($\theta_1 = 90\,deg$, $\theta_2 = 0$ and $\theta_3 = 0$)

The last validation we did is by changing the θ_2 to 90 degrees and keeping other angles as zero and as expected the output came out is as follows:

x = 0, y = 0 and z = 57.4 cm (approx) as shown in the image below (launching pose for the robot in gazebo):

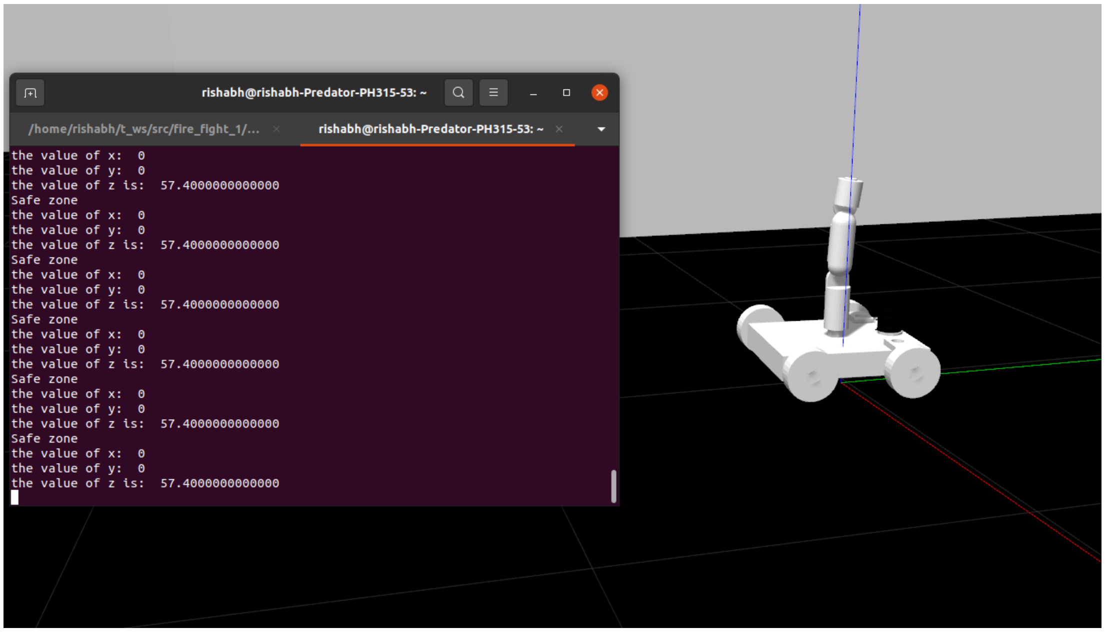

Fig. 16 Forward Kinematics Validation II ($\theta_1 = 0\,deg$, $\theta_2 = 90\,deg$ and $\theta_3 = 0$)

Similarly, we tried proving the forward kinematics using the transformation matrix directly as a python code and below are the outputs:

6.2.1 Forward Kinematics Validation I (Home position, θ_1, θ_2 and $\theta_3 = 0$)

```
end_final
```

$$\begin{bmatrix} -16\sin(\theta_2)\sin(\theta_3)\cos(\theta_1) + 16\cos(\theta_1)\cos(\theta_2)\cos(\theta_3) + 16\cos(\theta_1)\cos(\theta_2) \\ -16\sin(\theta_1)\sin(\theta_2)\sin(\theta_3) + 16\sin(\theta_1)\cos(\theta_2)\cos(\theta_3) + 16\sin(\theta_1)\cos(\theta_2) \\ 16\sin(\theta_2)\cos(\theta_3) + 16\sin(\theta_2) + 16\sin(\theta_3)\cos(\theta_2) + 25.4 \end{bmatrix}$$

```
[34] end_final.subs({theta1:0, theta2:0, theta3:0})
```

$$\begin{bmatrix} 32 \\ 0 \\ 25.4 \end{bmatrix}$$

6.2.2 Forward Kinematics Validation II ($\theta_1 = 90\,deg$, $\theta_2 = 0$ and $\theta_3 = 0$)

```
[3]  end_final
```

$$\begin{bmatrix} -16\sin(\theta_2)\sin(\theta_3)\cos(\theta_1) + 16\cos(\theta_1)\cos(\theta_2)\cos(\theta_3) + 16\cos(\theta_1)\cos(\theta_2) \\ -16\sin(\theta_1)\sin(\theta_2)\sin(\theta_3) + 16\sin(\theta_1)\cos(\theta_2)\cos(\theta_3) + 16\sin(\theta_1)\cos(\theta_2) \\ 16\sin(\theta_2)\cos(\theta_3) + 16\sin(\theta_2) + 16\sin(\theta_3)\cos(\theta_2) + 25.4 \end{bmatrix}$$

```
end_final.subs({theta1:sym.pi/2, theta2:0, theta3:0})
```

$$\begin{bmatrix} 0 \\ 32 \\ 25.4 \end{bmatrix}$$

6.2.3 Forward Kinematics Validation II ($\theta_1 = 0\,deg$, $\theta_2 = 90\,deg$ and $\theta_3 = 0$)

```
[3]  end_final
```

$$\begin{bmatrix} -16\sin(\theta_2)\sin(\theta_3)\cos(\theta_1) + 16\cos(\theta_1)\cos(\theta_2)\cos(\theta_3) + 16\cos(\theta_1)\cos(\theta_2) \\ -16\sin(\theta_1)\sin(\theta_2)\sin(\theta_3) + 16\sin(\theta_1)\cos(\theta_2)\cos(\theta_3) + 16\sin(\theta_1)\cos(\theta_2) \\ 16\sin(\theta_2)\cos(\theta_3) + 16\sin(\theta_2) + 16\sin(\theta_3)\cos(\theta_2) + 25.4 \end{bmatrix}$$

```
[36] end_final.subs({theta1:0, theta2:sym.pi/2, theta3:0})
```

$$\begin{bmatrix} 0 \\ 0 \\ 57.4 \end{bmatrix}$$

6.3. Inverse Kinematics

In this section we tried to plot the circle using our end effector on the x-y plane. We will be submitting the notebook with all the codes in it but still we would like to demonstrate what we did.

So, we tried to take the base configuration avoiding all kinds of singularity which meant taking $\theta_1 = 0\,deg$, $\theta_2 = 90\,deg$ and $\theta_3 =- 90\,deg$, as displayed in the figure below:

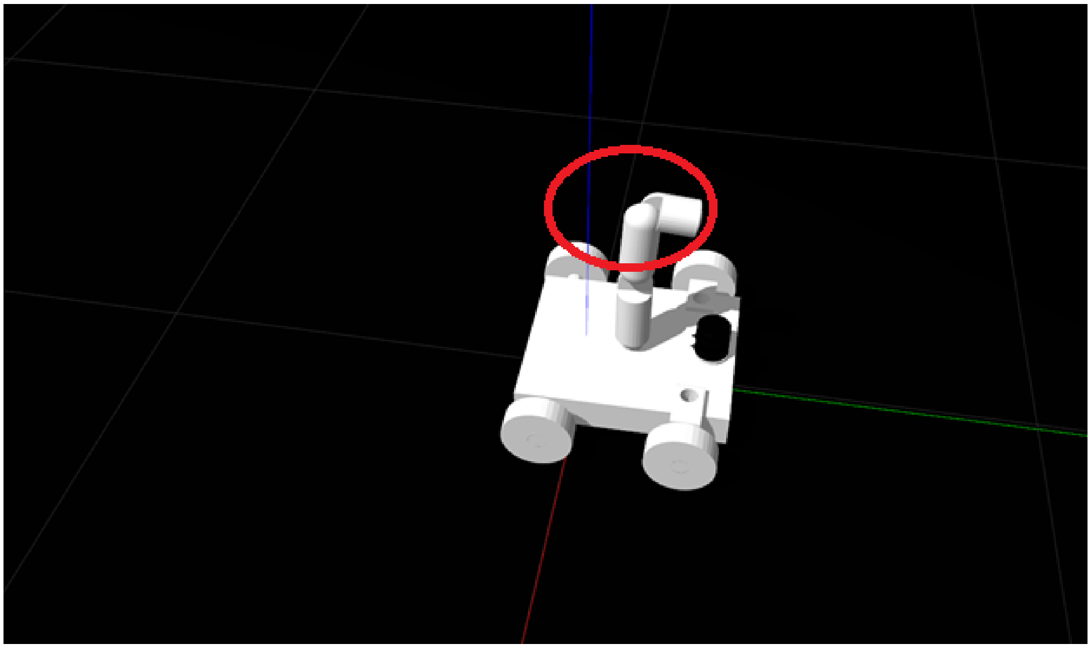

Fig. 17 The circle been plotted for the IK validation

The manipulator jacobians (J1, J2 and J3) for our velocity kinematics are as shown below:

$$J1 = \begin{bmatrix} 16\sin(\theta_1)\sin(\theta_2)\sin(\theta_3) - 16\sin(\theta_1)\cos(\theta_2)\cos(\theta_3) - 16\sin(\theta_1)\cos(\theta_2) \\ -16\sin(\theta_2)\sin(\theta_3)\cos(\theta_1) + 16\cos(\theta_1)\cos(\theta_2)\cos(\theta_3) + 16\cos(\theta_1)\cos(\theta_2) \\ 0 \\ \sin(\theta_1) \\ -\cos(\theta_1) \\ 0 \end{bmatrix}$$

$$J2 = \begin{bmatrix} -16\sin(\theta_2)\cos(\theta_1)\cos(\theta_3) - 16\sin(\theta_2)\cos(\theta_1) - 16\sin(\theta_3)\cos(\theta_1)\cos(\theta_2) \\ -16\sin(\theta_1)\sin(\theta_2)\cos(\theta_3) - 16\sin(\theta_1)\sin(\theta_2) - 16\sin(\theta_1)\sin(\theta_3)\cos(\theta_2) \\ -16\sin(\theta_2)\sin(\theta_3) + 16\cos(\theta_2)\cos(\theta_3) + 16\cos(\theta_2) \\ \sin(\theta_1) \\ -\cos(\theta_1) \\ 0 \end{bmatrix}$$

$$J3 = \begin{bmatrix} -16\sin(\theta_2)\cos(\theta_1)\cos(\theta_3) - 16\sin(\theta_3)\cos(\theta_1)\cos(\theta_2) \\ -16\sin(\theta_1)\sin(\theta_2)\cos(\theta_3) - 16\sin(\theta_1)\sin(\theta_3)\cos(\theta_2) \\ -16\sin(\theta_2)\sin(\theta_3) + 16\cos(\theta_2)\cos(\theta_3) \\ \sin(\theta_1) \\ -\cos(\theta_1) \\ 0 \end{bmatrix}$$

Now, using the inverse velocity kinematics we tried to find the joint angles required to plot a circle on this plane in 200 secs. End effector position, end effector velocity and instantaneous joint velocities for each point on the circle can be seen in the notebook (python script) file attached in the submission.
The equations for plotting the circle is defined as below:

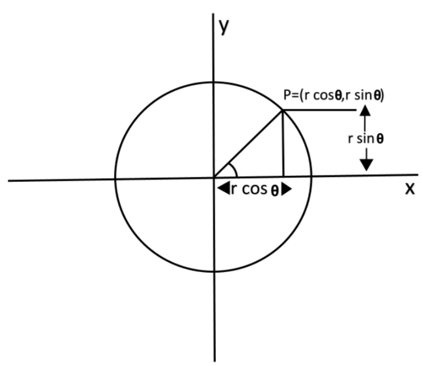

Fig. 18 The circle plotted on x-y plane

$$x\_end = 16 \times cos(\theta) \text{ cm}$$
$$y\_end = 16 \times sin(\theta) \text{ cm}$$
$$z\_end = 41.4 \text{ cm}$$

Using these equations we have divided the circle into 40 points, calculated the instantaneous velocity, then joint velocities, the jacobian at each joint velocity, and finally re-plotted the circle in order to validate our inverse kinematics. The output we got is as follows:

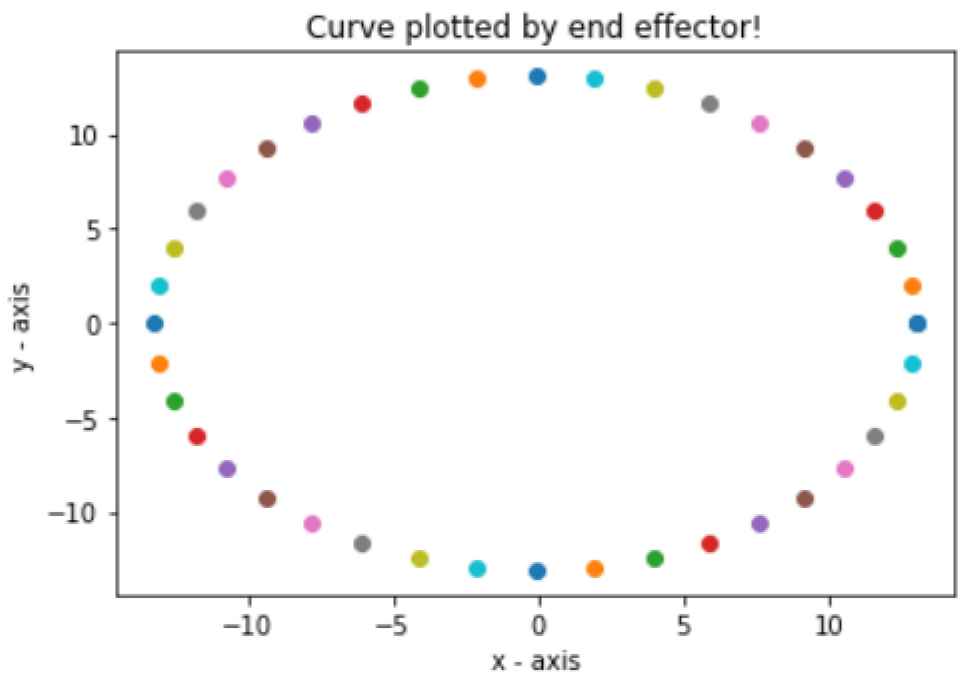

Fig. 19 The circle plot validation

It can be observed that a circle in the x-y plane with diameter of about 32 cm is plotted, which validates our jacobians and joint velocities for each point, since we are able to trace it back.

Below are the end effector velocities at different points, it can be seen that the z is constantly zero as we are plotting the circle in the x-y plane:

End effector velocities at each point

```
[48] end_new_dot
```

$$\begin{bmatrix} -0.03 & -0.1 & -0.16 & -0.21 & -0.27 & -0.31 & -0.35 & -0.38 & -0.4 & -0.41 & -0.41 & -0.4 & -0.38 & -0.35 & -0.31 \\ 0.41 & 0.4 & 0.38 & 0.35 & 0.31 & 0.27 & 0.21 & 0.16 & 0.1 & 0.03 & -0.03 & -0.1 & -0.16 & -0.21 & -0.27 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

The whole process is available in the attached python script/notebook.

6.4. Workspace Study

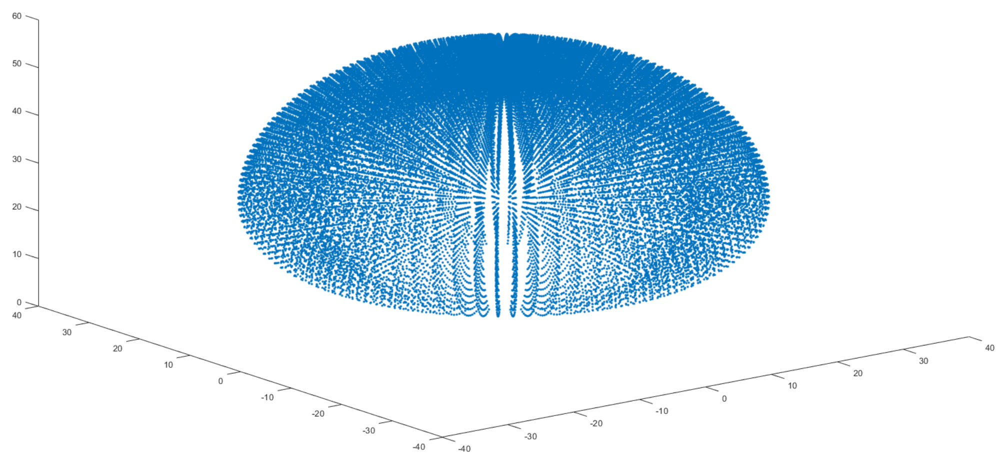

Fig. 20 Workspace Study

The workspace contains all the positions the end effector can reach. As observed from the 3D plot graph, the workspace is a hollow hemisphere. From the top view we can observe the centre to the longest distance, 30 cm, is a sum of the lengths of link 2 and link 3 i.e 16+14=30 cm.

The base link (link 1) has a length of 25 cm, $-\Pi < \theta_1 < \Pi$.

The link 2 has a length of 16 cm, $-\Pi/2 < \theta_2 < \Pi/2$.

The link 3(end effector) has a length of 14 cm, $-\ \Pi/2\ < \theta_3 < \ \Pi/2$.

The plot is obtained by iterating the θ_1, θ_2,θ_3 for the above mentioned limits in the forward kinematics equation in MATLAB.

Note: Code is attached in the folder.

7. Assumptions

7.1. Modelling Assumptions

1. Links and Joints are assumed to be rigid.
2. Factors like friction, joint clearance, backlash are not considered..
3. The robot is not modelled to each specific material.

7.2. Simulated Environment Assumptions

1. The CAD design is simplified when exported to Gazebo environment for faster computations
2. Since fire cannot be depicted in the environment, pillars/walls are assumed to be the center of fire.
3. For proof of concept, 2D-LIDAR is used instead of other external sensors.

8. Control Method

An open-loop control method is employed to regulate the movement of the joint controllers in the Robot. In this method, the output of the system has no role in the effect of the control action.[6]

For the traversal part of the robot, Position Joint controllers are used to steer the front wheels of the robot. The latter takes in position as an input to align the front wheels.

The rear wheels of the robot are powered by motors using velocity joint controllers, making it a rear-wheel drive. Velocity joint controllers are used to give velocity for the rotation of the rear wheels.
The 3-R spatial arm of the robot is regulated using Position Joint controllers at each of the revolute joints. The end effector position is achieved by combining motion of the previous links.

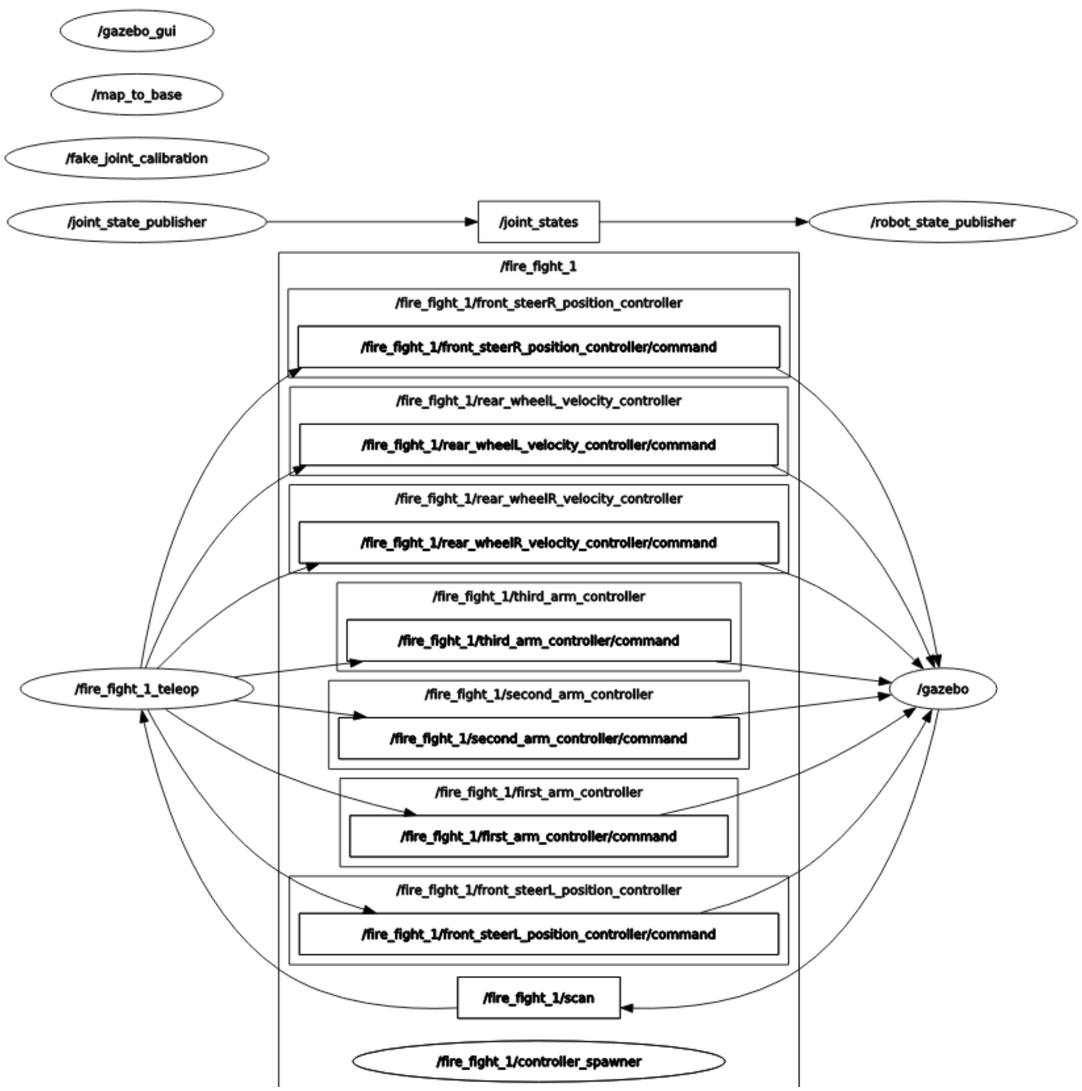

Fig. 19 RQT Graph for all nodes and topics

9. Gazebo and Rviz visualisation Results:

In this section we will first go through whatever we proposed as our laidback and ambitious goals and then we'll talk about how we demonstrated it using Rviz and Gazebo.

9.1. Laid back goals:

A. Designing a compact and robust robot:

As per our proposal we tried to make the bot as small as compact as possible to solve the purpose and be maneuverable inside the buildings which can have sharp turns or narrow paths. The track width of the robot is around 44 cm and the wheelbase of the robot is also around 45 cm. The materials of the Robot for keeping the robustness are assumed to be 1020 DOM steel and 7075 Aluminium. The CAE section gives out the results for von-mises stresses and total deformation with work conditions like, the robot drop test from a height of 4 m which gives out the deformation as close as 0.1 mm which gives sufficient FOS to go ahead with the same design.

B. Aesthetics, and control with minimal time delay and slip:

In project 1 we faced issues like lagging and the bot was moving without any input which was not good for aesthetics. This time we tried to check everything by giving clearances as well as taking the real time feedback from the loop. With utilizing minimum sleep and refresh rate as well as sufficient acceleration in our algorithm we tried to teleop the robot with minimum delay and slip. Also our safety mechanism is fast and efficient because of the way we have utilized the callback function in the code.

C. Designing a 3-R arm for the nozzle:

Though not too fancy, we designed everything including the arms from scratch and defined all the controllers manually, according to need, instead of using MoveIt. So, basically all the joints in our robot are either manually controllable using the telop node or else using a publisher script.

D. Controlling the arm with a minimal positional error:

This issue is again taken care of by our efficient algorithm and the real time feedback which acts like an encoder and allows you to take the end effector to your desired location with a least count of around 0.1 cm.

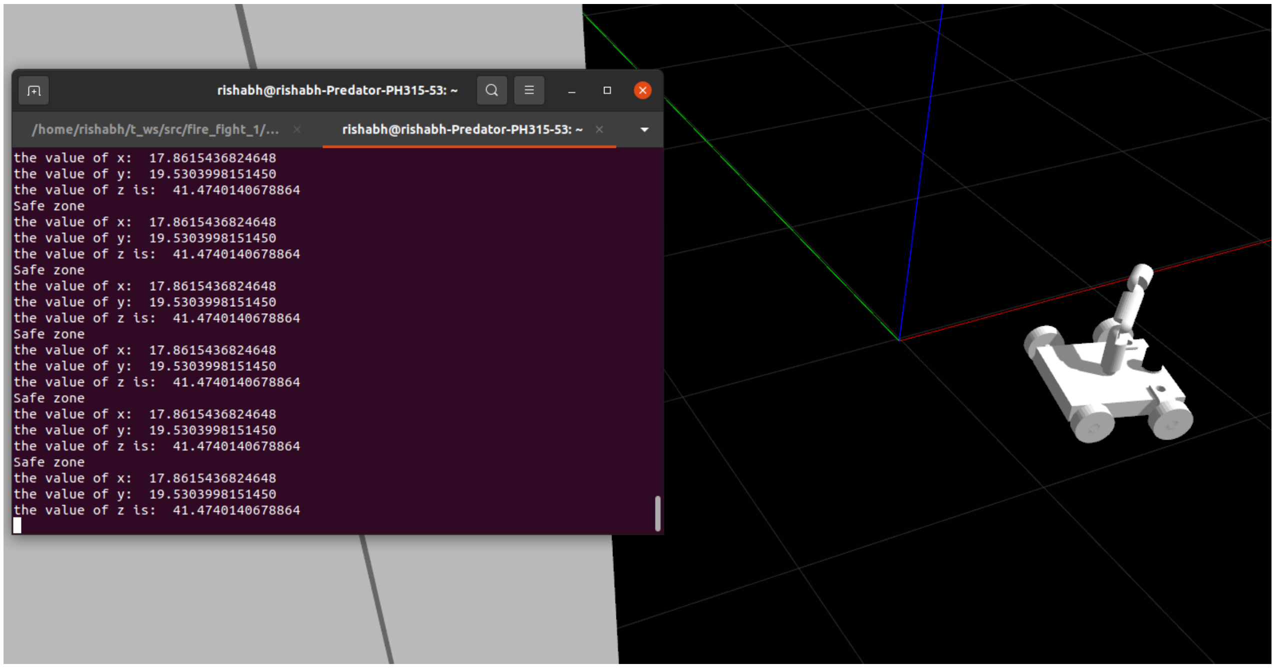

Fig. 20 Real-time feedback from the robot

9.2. Ambitious Goals:

A. **Programming the robot to maintain a distance from the proximity of the fire, automatically**
B. **Programming the robot to provide continuous feedback when the temperature is too high.**
C. **Autonomous traversal of the robot.**
D. **Continuous communication is available with the operator.**

These are the achievements which we are really proud of. We assumed the walls to be the fire and as a safety element the bot won't let you drive it near the walls. In addition to this, it will act autonomously, traverses back to the 2m zone, manipulates the nozzle towards the fire, extinguishes it for couple of seconds (this can be done using a sensor feedback but we didn't have time to simulate the real fire as such), gives the feedback to the user and also give backs the control to the user to let him/her drive it to the new location. Continuous real time feedback is being given to the operator for safe zone, unsafe zone (too close), extinguishing in progress and fire extinguished.

We have subscribed to the scan topic created by the lidar node and for our purpose we have modified the FOV and sampling rate. Rviz will be showing only 8-10 points in front during the detection process as can be seen in the videos below. We have shown the validation at two-three different locations for the confirmation. The same can be verified by launching the package and by trying to operate the bot near the walls.

The validation, gazebo and rviz visualization can be seen in the given video links (watch in 2X to save time):

Video1:
https://drive.google.com/file/d/1eNeBHBc03--l9YuwkIQNe0cwVSJH29Yp/view?usp=sharing

Video2:
https://drive.google.com/file/d/1zyi0A6W0nWwWyqcLmRIcf816NRYv7Zmw/view?usp=sharing

10. Problems faced

- **Collision free design for manipulator:**
Multiple attempts of designing to create a functional robot with minimal collisions.

- **P gain error while launching gazebo:**
The solution mentioned did not apply in our case.

- **The robot toppled due to the mass of the manipulator and gazebo physics:**

When the robot was introduced in the gazebo environment, the initial configuration of the robot was toppled. Later, it was known that this was because the manipulator was heavier than the robot body.

● **Defining boundary conditions for Finite Element Analysis**

● **Incorporating arm manipulator control code in teleop:**
While writing the teleop code for the manipulator, we faced many challenges like creating new keys for arm link movements.

● **Understanding the use of callback function and rosspin():**
The teleop code for the robot was malfunctioning due to misuse of rosspin() and rosspinOnce() for callback function.

● **Difficulty in code implementation for automated intelligent fire-safety mechanism:**
While configuring the ranges from laserscan topic to program an fire-safety mechanism, the sampling rate of the lidar points had to be reduced to isolate the area of scan.

11. Lessons learned

- Analyze the kinematics and dynamics of building a robot.
- Evaluating the Inverse and Forward Kinematics
- Creating a design and modelling it in Solidworks.
- Correct configuration of reference geometry for smooth simulation in Gazebo
- To design simulation environments and implement it in ROS- Gazebo
- To incorporate the sensor outputs in Rviz with the Robot model in Gazebo
- Working of a 3-R arm to a defined position with minimal error.
- Integrating sensor data to control the robot in an environment.
- To understand ROS libraries and how to implement them.
- To study the dexterity and workspace of the robot.

12. Conclusion

The vision of this project is to reduce human presence in highly hazardous environments like fire. The robot can access these areas, mitigate the effects of fire, and transfer data to the central control stations at a much safer and faster rate.The robot is intelligently programmed to protect itself while being controlled remotely. The benefits reaped are not limited to speed but reduces the loss factor involved in fire.

It can be concluded that building and programming firefighting robots is an accomplishable task. Though we only used LIDAR, due to proof of concept reasons, various sensors and cameras can be integrated into the robot for solving purposes like human detection and then carry them back safely using a follower robot which receives the coordinates of the place where the victims are, from the leader robot which is technically being remotely operated.

The robot is designed keeping in mind features like modularity, robustness. The ROS environment is used to materialise our POC (proof of concept) in a well defined manner before real world implementation. Extending this project to RWUP (Real World Usage Profile) could be a challenge but after doing this project we can conclude that, given the proper resources it is achievable. Firefighting is no more a risky and dangerous career !!

13. Future Work

Improvements to automation is a never ending process. In the future, we hope to program the robot to identify a human and send feedback to the operator, which could be very beneficial to the rescue process. We would also like to extend the current semi-automated robot to be completely autonomous in path planning and traversal. Failsafe mechanisms and sensors should be incorporated for reliable operation in such extreme conditions. Thermal sensors can be incorporated to give feedback to the nozzle to continue spraying until the fire is extinguished.

14. References

1. https://www.nfpa.org/News-and-Research/Data-research-and-tools/US-Fire-Problem/Fire-loss-in-the-United-States)
2. (https://www.usfa.fema.gov/data/statistics/#tab-4)
3. https://www.howeandhowe.com/civil/thermite
4. https://www.nist.gov/news-events/news/2018/08/nist-shows-laser-ranging-can-see-3d-objects-melting-fires
5. https://safetymanagement.eku.edu/blog/the-use-of-robotics-in-firefighting/
6. https://www.electronics-tutorials.ws/systems/open-loop-system.html

15. Supplemental Material

https://drive.google.com/drive/folders/1-olWPS6E5dEb62dAt1q-vj1033zLsgyW?usp=sharing

16. Contributions

Alvina Alex

1. Design and CAD modelling of the robot
2. URDF export
3. Forward Kinematics Analysis
4. Workspace Analysis
5. Gazebo and Rviz Visualization
6. Report Writing on the parts done by each
7. Presentation

Rishabh Singh

1. ROS Teleop and Automation Script
2. Move it Analysis
3. Ansys Analysis
4. Inverse Kinematics Analysis
5. Gazebo and Rviz Visualization
6. Report Writing on the part done by each
7. Presentation