ENPM662 - Project 2 Report

Robotic Arm with Enhanced Reachable Workspace

December 10, 2021

Authors

Rigved Kulkarni (UID: 117358124)

Alec Lahr (UID: 114750709)

Table of Contents

1 Introduction

Automation is arguably the most important aspect of modern human society. It enables humans to live more relaxed and content lives by lifting away menial and repetitive work burdens. Robots used in automation are also far more precise than human workers, empowering us to create sophisticated technologies not possible by human hands. These technologies, like computers and vehicles, are the very foundations of life in the 21st century.

Automation can be found in nearly every industry from transportation to information technology to agriculture to, perhaps most significantly, manufacturing. By reducing production costs, increasing product quality, and increasing factory reliability and predictability, automated manufacturing processes have become a core component of the global economy. Examples of machines included in these automated manufacturing processes include CNC machines, foundry casting machines, painting machines, plastic forming machines, and many more. All of the machines listed above are self-contained. That is, they do one specific job to a workpiece given to them; though, they are incapable of inserting and removing the workpiece and passing it down the line to the next machine. Moreover, some CNC machines are incapable of changing their own tools. Historically, the task of moving parts between machines and performing tool changes on machines has been for humans. Though, in more recent years, robots have become affordable enough and usable enough to enable this menial task to be automated too.

In addition to transferring parts between machines and performing tool changes, humans have also traditionally done jobs that require high workspace flexibility like welding or tightening bolts in difficult-to-reach places.

There are countless versions of robot arms on the market for use in industrial manufacturing applications - each with unique strength, speed, weight, and precision. However, the reachable workspace for nearly all of these arms is the same: a semi-sphere around and above the robot.

This study aims to develop a novel robot arm capable of reaching below its workspace uninhibited. Upon completion, the authors will present a full CAD rendering of the design as well as derive forward and inverse kinematics for the robot.

This report is a proposal for work on the novel robot arm concept. Section 2 of this report will talk about the author's motivation for choosing this problem and why it is interesting. Section 3 will describe the geometry of the robot as well as any sensors, joints, and special materials present. Section 4 will explain why the design of the robot is appropriate for tasks in industrial automation applications requiring a large reachable workspace. Sections 5 and 6 will describe which parts of the robot's functions will be

explored in this study then justify the significance of this scope. Section 7 will briefly cover what will not be explored in this study by listing all assumptions made. Sections 8 and 9 will describe an implementation plan and set progress goals on a timeline. Finally, section 10 details how the results of the study will be validated.

2 Application

The global industrial automation field has been rapidly growing for many years and is expected to continue growing. Currently, the industry is worth approximately 200 billion dollars and has an expected compounded annual growth rate of 8% until 2026, at which point it will be worth 300 billion dollars [1]. Contributing to this growth is the increased use of robotic arms for cost-efficient, high-volume, low-risk production processes.

A significant gap in the industrial robot arm market is robot arms with a fully spherical reachable workspace - as opposed to the common semi-sphere robot workspaces.

Robot arms that are capable of reaching far below their base have many advantages. First, they reduce wasted space in the factory. With traditional robot arms (TRAs), tools and workpieces were restricted to being placed above and around the arm. Often, this meant making space for a whole large station to work from. A spherically reaching robot arm (SRRA) on the other hand, enables factory designers to save floor space by putting the work area below the machine. Second, consider the example of a robot welding the frame of a car on an automated assembly line. TRAs are unable to reach beneath the car without it being significantly raised. Often, the tradeoff here is needing two robots, one to weld the top of the car and one to weld the bottom of the car. With its greater maneuverability, an SRRA is capable of performing both of these tasks without needing the vehicle to be raised. In this example, having a single SRRA instead of two TRAs would significantly reduce factory startup costs.

SRRAs have the potential to reduce the cost to build a factory by using floor space more efficiently and performing the same job as two TRAs. In an industry that continues to grow rapidly, even a small reduction in cost could save millions of dollars.

3 Robot Description

The proposed robot is a 6-DoF robot arm. The first three joints move can be used to position the end-effector in 3-D space while the final three joints act as a spherical wrist to angle the end-effector. Together, these 6-DoF can reach an infinite number of positions and angles relative to the base of the robot. While the specific motors selected are not relevant to this study, all motors controlling the proposed robot will be high-performance stepper motors with precise, consistent, strong, and fast actuation.

Figure 1 shows the robot in multiple configurations with the right image highlighting the reachability advantages of the C-shaped link. Figure 2 shows the robot in its home position and each link reference frame used to create the DH table. Table 1 shows the resulting DH table.

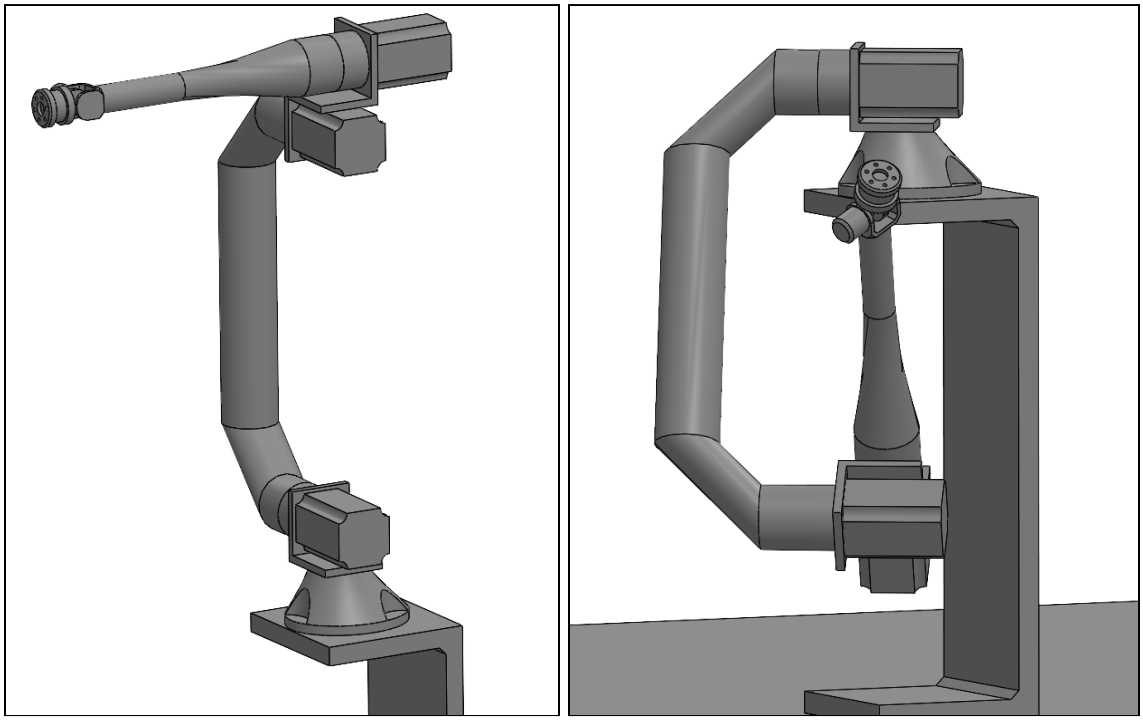


Figure 1: An isometric view of the robot in multiple configurations

Table 1: DH parameters

	α	a	d	Θ
0-1	-90	0	0	Θ_1
1-2	0	0	0	Θ_2-90
2-3	0	a_3	0	0
3-d	-90	a_d	0	Θ_3
d-4	0	0	d_4	0
4-5	90	0	0	Θ_4
5-6	-90	0	0	Θ_5
6-7	0	0	d_7	Θ_6

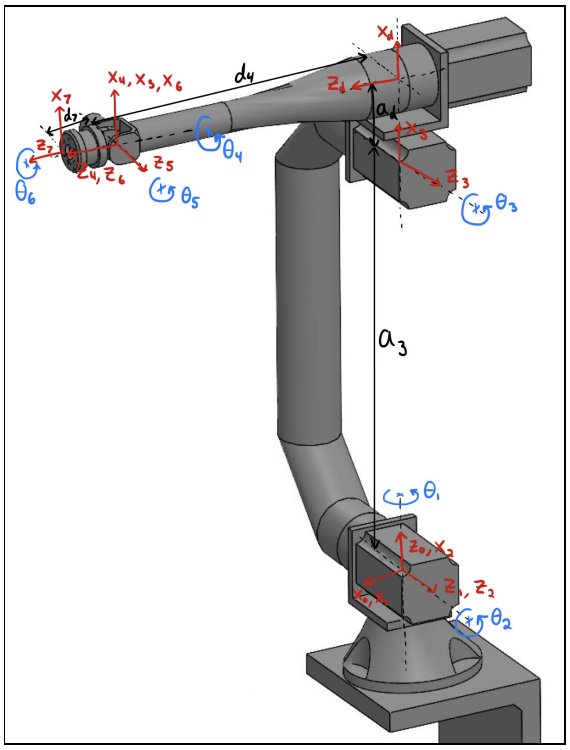


Figure 2 (left): Robot with coordinate frames, Θ definitions, and dimensions

For this robot design, a_3=975mm, a_d=155.9mm, d_4=744.92mm, and d_7=144.98mm.

Figure 3 below shows a composite image of the robot in multiple configurations which shows the reachable workspace of the robot. Notice in the side view (left) that the workspace is the entire circle, while most other robot arms are only partial circles.

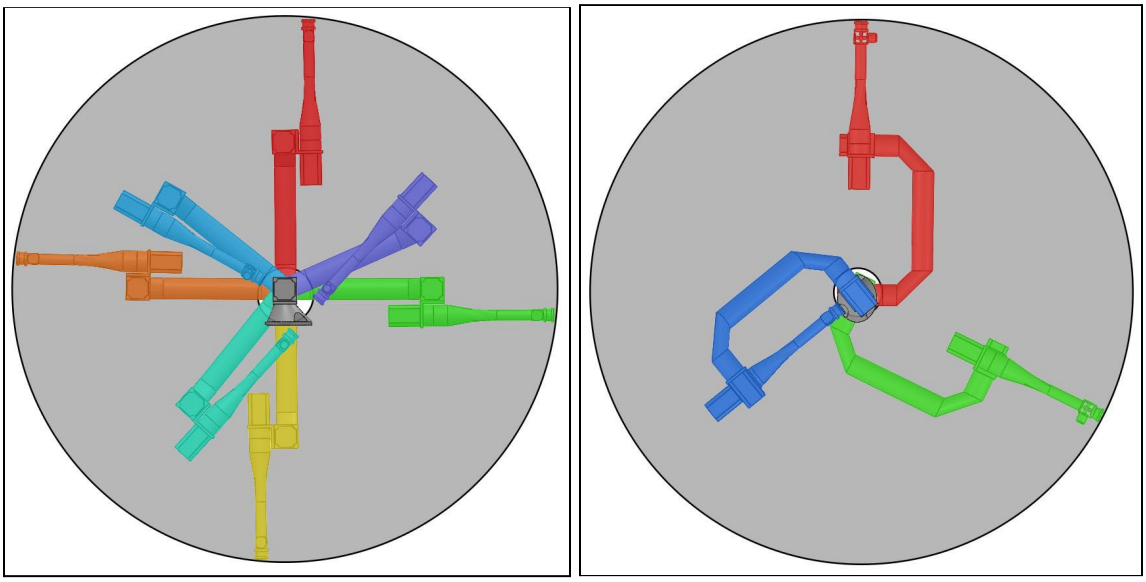


Figure 3: Robots reachable workspace shown in gray. Robot configurations shown in different colors. Side view (left) and top view (right).

4 Forward Kinematics

Using the DH parameters in table 1 and the Sprog convention transformation matrix formula, the transformation matrices between coordinate frames are given as follows:

$$T01=\begin{bmatrix} \cos(\Theta_1) & 0 & -\sin(\Theta_1) & 0 \\ \sin(\Theta_1) & 0 & \cos(\Theta_1) & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad T12=\begin{bmatrix} \sin(\Theta_2) & \cos(\Theta_2) & 0 & 0 \\ -\cos(\Theta_2) & \sin(\Theta_2) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$T23=\begin{bmatrix} 1 & 0 & 0 & a_3 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad T3d=\begin{bmatrix} \cos(\Theta_3) & 0 & -\sin(\Theta_3) & ad\cos(\Theta_3) \\ \sin(\Theta_3) & 0 & \cos(\Theta_3) & ad\sin(\Theta_3) \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$Td4=\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_4 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad T45=\begin{bmatrix} \cos(\Theta_4) & 0 & \sin(\Theta_4) & 0 \\ \sin(\Theta_4) & 0 & -\cos(\Theta_4) & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$T56=\begin{bmatrix} \cos(\Theta_5) & 0 & -\sin(\Theta_5) & 0 \\ \sin(\Theta_5) & 0 & \cos(\Theta_5) & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad T67=\begin{bmatrix} \cos(\Theta_6) & -\sin(\Theta_6) & 0 & 0 \\ \sin(\Theta_6) & \cos(\Theta_6) & 0 & 0 \\ 0 & 0 & 1 & d_7 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Multiplying each of these matrices together, we obtain:

T07=T01*T12*T23*T3d*Td4*T45*T56*T67

The full T07 matrix is too large to reasonably show in this report; therefore, find the included Python code to generate the T07 matrix.

To validate the forward kinematics equations, the T07 matrix will be evaluated for two known robot configurations. First, consider the home position (figure 2) where all Θ=0. The numerical solution to the T07 matrix is:

$$T07=\begin{bmatrix} 0 & 0 & 1 & d_4+d_7 \\ 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & a_3+ad \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Looking at figure 2, we can see by inspection that this T07 matrix is correct. X_0 becomes Z_7 and Z_0 becomes X_7. The origin of frame 7 is a distance d_4+d_7 from frame 0 in the X direction, a distance of 0 in the Y direction, and a distance of a_3+a_d in the Z direction.

Next, consider the configuration shown in figure 4 given by:
$\Theta_1=90$ $\Theta_2=-90$ $\Theta_3=0$ $\Theta_4=180$ $\Theta_5=0$ $\Theta_6=0$

The python code gives the T07 as:

$$T07=\begin{bmatrix} 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & -a_3 - ad \\ 0 & 0 & 1 & d_4 + d_7 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Looking at figure 4, we can see by inspection that this T07 matrix is correct. X_0 becomes $-Y_7$ and Z_0 stays Z_7. The origin of frame 7 is a distance of 0 in the X direction, a distance $-a_3-a_d$ from frame 0 in the Y direction, and a distance of d_4+d_7 in the Z direction.

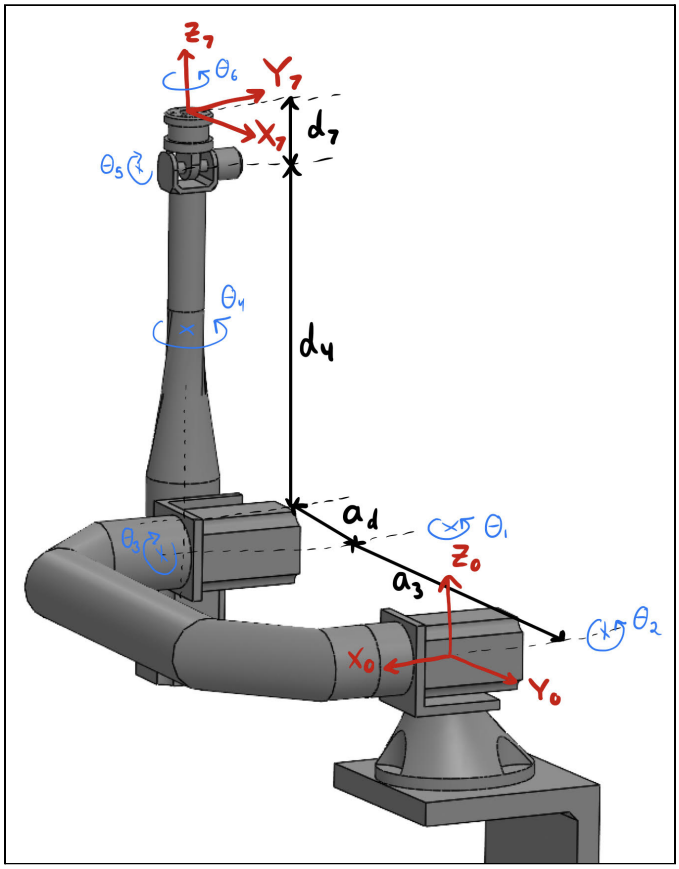


Figure 4: An alternate robot configuration used for forward kinematics validation

5 Inverse Kinematics

This robot was strategically developed to have a spherical wrist so the position and rotation components of the goal pose can be separated. Joints 1, 2, and 3 will be used to achieve the desired end-effector position while joints 4, 5, and 6 will be used to achieve the desired end-effector rotation. The following steps use this separability principle to derive the joint angles $\Theta_{1\text{-}6}$ that bring the end effector to the goal.

Let O be the position vector of the goal pose and R be the rotation matrix of the goal pose. Next, let O_C^0 be the position of the wrist center.

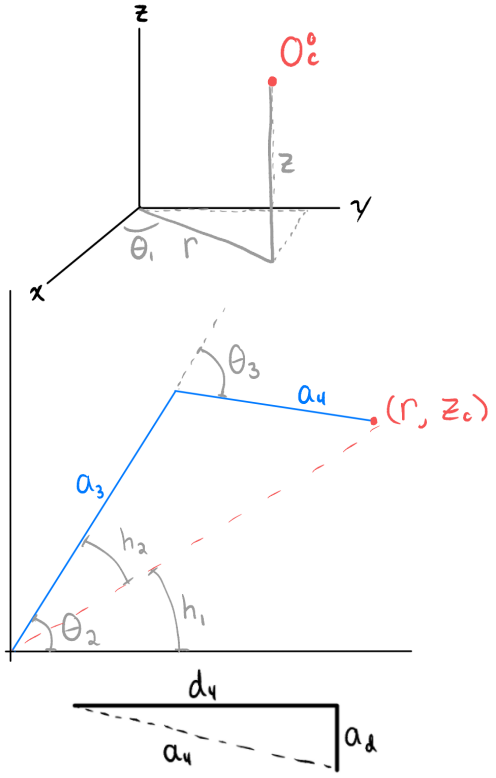
Then $O = O_C^0 + d_7 * R\ [0\ \ 0\ \ 1]^T$

$O = [O_X\ \ O_Y\ \ O_Z]^T$ and $O_C^0 = [X_C\ \ Y_C\ \ Z_C]^T = [(O_X - d_7 r_{13})\ \ (O_Y - d_7 r_{23})\ \ (O_Z - d_7 r_{33})]^T$

Find $\Theta_{1\text{-}3}$ given the intermediate goal O_C^0:
$\Theta_1 = \text{atan2}(X_C, Y_C)$
$r^2 = X_C^2 + Y_C^2$

From the law of cosines,
$\cos\Theta_C = -(r^2 + Z_C^2 - a_3^2 - a_4^2) / (2a_3a_4)$
$\Theta_3^* = \pi - \text{acos}(-(r^2 + Z_C^2 - a_3^2 - a_4^2) / (2a_3a_4))$
Θ_3^* is for the a_4 simplification, to convert it to the geometry described in figure 2, find Θ_3
$\Theta_3 = \pm(\Theta_h - \Theta_3^*)$
+ elbow up
- elbow down
$\Theta_h = \text{atan}(d_4 / a_d)$

$\Theta_2 = h_1 \pm h_2$
+ elbow up
- elbow down
$h_1 = \text{atan2}(r, Z_C)$
$h_2 = \text{atan}(a_4 \sin\Theta_3^* / (a_3 + a_4\cos\Theta_3^*))$
$a_4 = \text{sqrt}(d_4^2 + a_d^2)$



Find $\Theta_{4\text{-}6}$ that reaches O from O_C^0:
$R_4^0 R_7^4 = R \rightarrow R_7^4 = R_4^{0\,-1} R$
Represent with Euler angles $\Theta_4 = \Phi$, $\Theta_5 = \Theta$, $\Theta_6 = \Psi$

$$R_7^4 = \begin{bmatrix} C\Phi C\Theta C\Psi - S\Phi S\Psi & -S\Phi C\Psi - C\Phi C\Theta S\Psi & S\Phi S\Theta \\ S\Phi C\Theta C\Psi - C\Phi S\Psi & C\Phi C\Psi - S\Phi C\Theta S\Psi & S\Phi S\Theta \\ -S\Theta C\Psi & S\Theta S\Psi & C\Theta \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix}$$

$\cos\Theta = r_{33} \rightarrow \Theta = \Theta_5 = \text{atan}(\text{sqrt}(1 - r_{33}^2) / r_{33})$

$\tan\Phi = \sin\Phi\sin\Theta / (\cos\Phi\sin\Theta) = r_{23} / r_{13} \rightarrow \Phi = \Theta_4 = \text{atan2}(r_{13}, r_{23})$

$\tan\Psi = \sin\Theta \sin\Psi / -(-\sin\Theta\cos\Psi) = r_{32} / -r_{31} \rightarrow \Psi = \Theta_6 = \text{atan2}(-r_{31}, r_{32})$

To validate the inverse kinematic equations derived above, we will randomly choose a goal pose, find the joint angles with inverse kinematics, then verify the joint angles by plugging them into the forward kinematics equations. If the inverse kinematics are correct, the starting pose should equal the resulting pose from the forward kinematics.

Choose the goal pose:

$$T07 = \begin{bmatrix} -0.2999 & -0.8466 & 0.4397 & 794.1 \\ -0.9532 & 0.2849 & -0.1015 & 715.6 \\ -0.0393 & -0.4495 & -0.8924 & -1497 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Position joints

$[X_C \ Y_C \ Z_C]^T =$

$[(794.1 - 144.98*0.4397) \ (715.6 - 144.98*-0.1015) \ (-1497 - 144.98*-0.8924)]^T$

$[X_C \ Y_C \ Z_C]^T = [730.35 \ 730.31 \ -1367.6]^T$

Θ_1 = atan2(730.35, 730.31) = 0.7854

$r = \text{sqrt}(730.25^2 + 730.31^2) = 1032.84$

$a_4 = 761.06$

$\Theta_3^* = \pi - \text{acos}(- (1032.84^2 + -1367.6^2 - 975^2 - 761.06^2) / (2*975*761.06)) = 0.3236$

$\Theta_h = \text{atan}(744.92/155.9) = 1.364$

Θ_3 = - (1.364 - 0.3236) = -1.0404

$h_1 = \text{atan2}(1032.84, -1367.6) = 2.495$

$h2 = \text{atan2}(975 + 761.06*\cos(0.3236), 761.06*\sin(0.3236)) = 0.1417$

Θ_2 = 2.495 - 0.1417 = 2.353

Rotation joints

$$R = \begin{bmatrix} -0.2999 & -0.8466 & 0.4397 \\ -0.9532 & 0.2849 & -0.1015 \\ -0.0393 & -0.4495 & -0.8924 \end{bmatrix}$$

Using forward kinematics on the found $\Theta_1, \Theta_2, \Theta_3$, we obtain $R_4^{0\,-1}$

$$R_4^{0\,-1} = \begin{bmatrix} 0.6837 & 0.6837 & 0.2553 \\ 0.7071 & -0.7071 & 0 \\ 0.1806 & 0.1806 & -0.9669 \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix}$$

Multiplying gives us R_7^4

$$R_7^4 = \begin{bmatrix} -0.8667 & -0.4988 & 0.0033 \\ 0.4619 & -0.8001 & 0.3829 \\ -0.1882 & 0.3332 & 0.9287 \end{bmatrix}$$

Θ_5 = atan(sqrt(1 - r_{33}^2) / r_{33}) = -0.3921

Θ_4 = atan2(r_{13}, r_{23}) = 1.562

Θ_6 = atan2(-r_{31}, r_{32}) = 1.0566

Plugging these Θ values back into the forward kinematics equation for T_7^0 we validate:

$$T_7^0 = \begin{bmatrix} -0.299647275126202 & -0.846870403668105 & 0.439342724874855 & 793.828921581312 \\ -0.953228913200246 & 0.284857396372023 & -0.101049011728122 & 715.485609541849 \\ -0.0395746073703833 & -0.44907324917342 & -0.892618097132423 & -1497.05540641283 \\ 0 & 0 & 0 & 1.0 \end{bmatrix}$$

This result is very close to the goal pose. The slight inaccuracies are attributed to rounding errors. Therefore, the derived inverse kinematics equations are validated.

6 Robot Appropriateness

The task of automated industrial manufacturing requires great precision and dexterity combined with a moderate amount of strength and speed. Some solutions made specifically for certain processes may be greater in all of these categories, though they aren't as generalizable as a robot arm. The true advantage of high DoF arms is their generalizability, the ability to be used in many situations with little modification, and thus can be purchased and set up in bulk to lower cost. While being generalizable, robot arms can also be exceptionally precise and boast impressive strength and speed. For these reasons, an arm is the best design for this task.

In order to expand the reachable workspace below the base of the robot, joint 2 must be allowed to turn a full 360°. This cannot happen on traditional robot arms because link 2 would intersect with the base. To overcome this, link 2 can be redesigned to be a C-shape. The cutout in the C allows the link to pass "through" the base.

An alternate solution could be increasing the number of degrees of freedom of the robot. Though, this solution comes with severe disadvantages. Namely, higher cost to have more motors, higher mathematical complexity for inverse kinematics computations, and decreased precision due to compounded joint angle errors.

With this design analysis in mind, the authors believe a C-shaped link 2 is not only appropriate but the best choice to solve this problem.

7 Model Assumptions

The SRRA is a 6-DoF arm with 6 revolute joints, 3 for positioning and 3 for rotating the end-effector. The differentiating factor is link 2 which enables the arm to reach almost an entire sphere as opposed to traditional robotic arms which have a semi-sphere reachable space. The following assumptions will be considered for the scope of this study:

1. All the joints and links of the arm will be rigid.
2. The friction between the subsequent joints is ignored.
3. During some configuration movement, the arm may collide with itself. This action will be ignored for the scope of this study. The collision with any other external object will be taken into account.
4. As we intend to program the robot to reach any given reachable endpoint. The path to reach the endpoint will not be optimized as a part of the fallback goal. Further work may be done on the trajectory planning as an ambitious goal

8 Control Method

The SRRA consists of 6 joints. As all the joints of the arm are revolute, we would need precise angle control for each joint to ensure an accurate end-effector position. To achieve this we would be using a Position Controller.

Position Controllers set a particular position of the joint depending on the input signal to the controller. The controller internally uses a PID controller for precise movement of the joints. Each joint has a complete 360^{o} range, hence any input to the controller in this range will rotate the joint to the desired location. Note that the axis of rotation is specific to each joint hence the zero position for each joint is with respect to that axis. You can also refer to the axis assignment for computing the DH parameters to identify the zero positions for each joint.

As the inbuilt controllers come equipped with PID control, no external control loop is programmatically applied to the system. The gains for each controller are manually tuned to reduce the jitters while moving.

9 Gazebo Simulation

There are two parts of the Gazebo Simulation to validate the working of the SRRA.

1. Forward Kinematics Validation

   Using the DH parameters we have the Transformation Matrix for end-effector position with respect to the base. Thus for a given set of joint angles, we can compute the end-effector position using the Transformation matrix. The same joint angles are given to the controllers in gazebo and matched with the output of the transformation matrix. The validation is verified for two configurations:

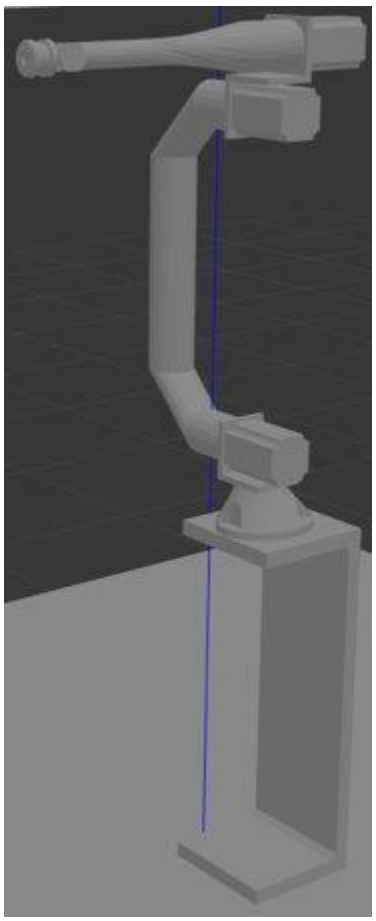


Figure 5a: Home Position Forward Kinematics Validation

theta1:0, theta2:0, theta 3:0, theta 4:0, theta 5:0, theta 6:0

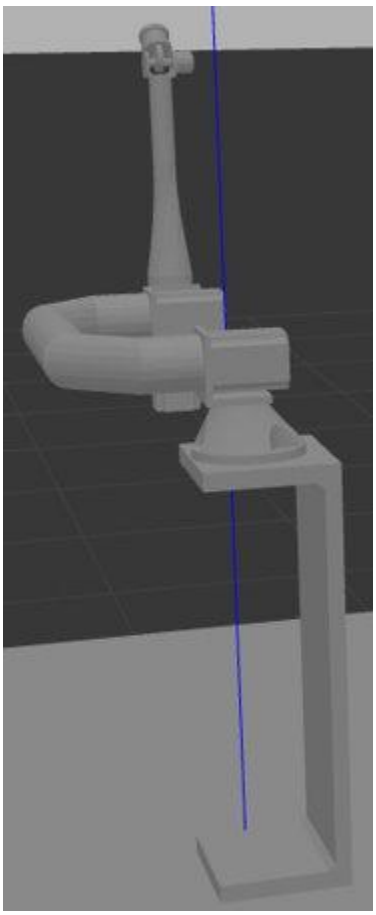
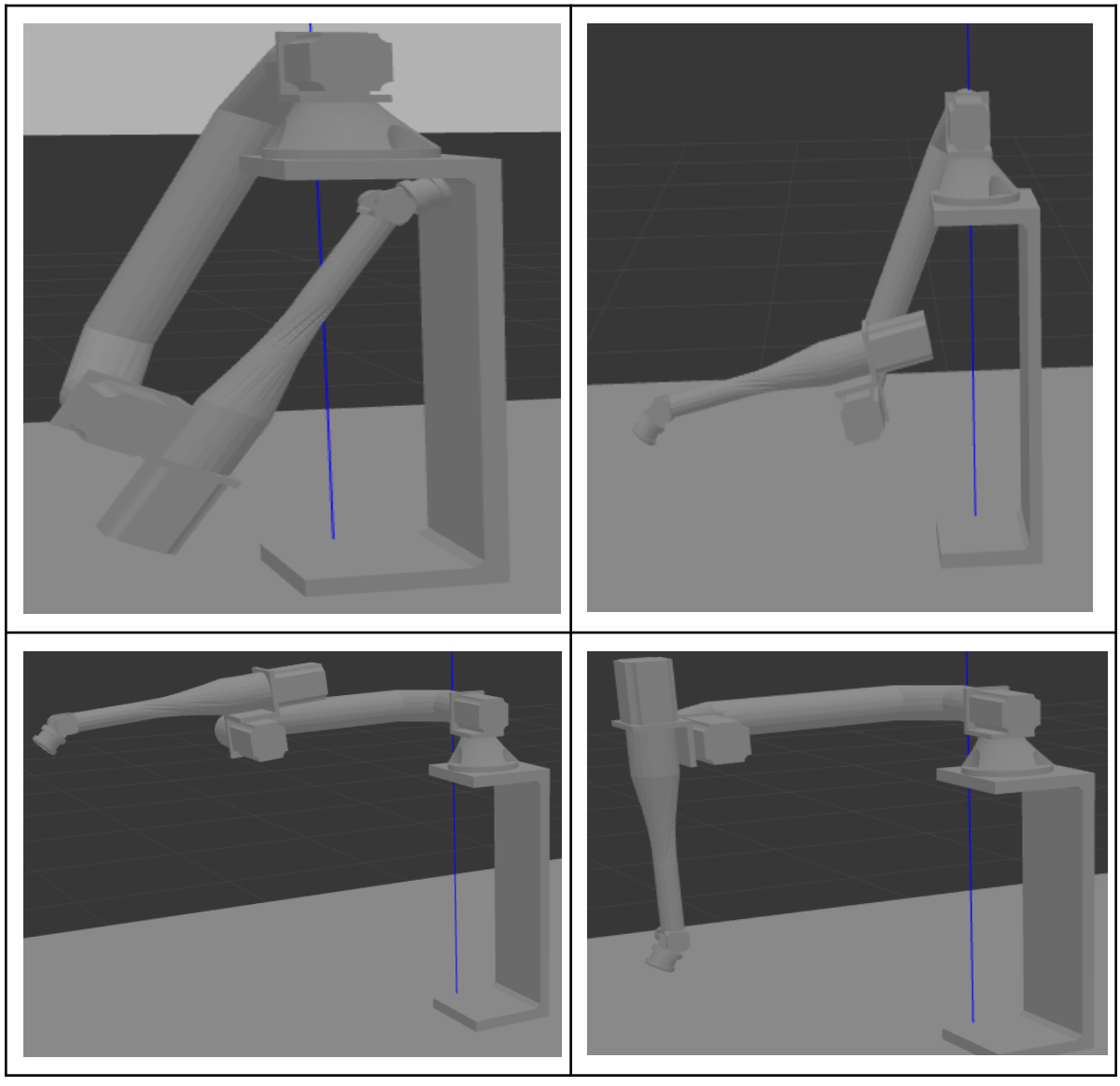


Figure 5b: Alternate Position Forward Kinematics Validation

theta1:pi/2, theta2:-pi/2, theta 3:0, theta 4:pi, theta 5:0, theta 6:0

2. Increased Reachable WorkSpace Validation

   Using the second link as a ‘C-shaped’ link increases the reachable workspace of the arm. It enables the arm to easily reach places below its mounting point. The SRRA covers the workspace of a traditional robot arm along with its increased coverage of volume. Following are some configurations that showcase this capability of the SRRA:



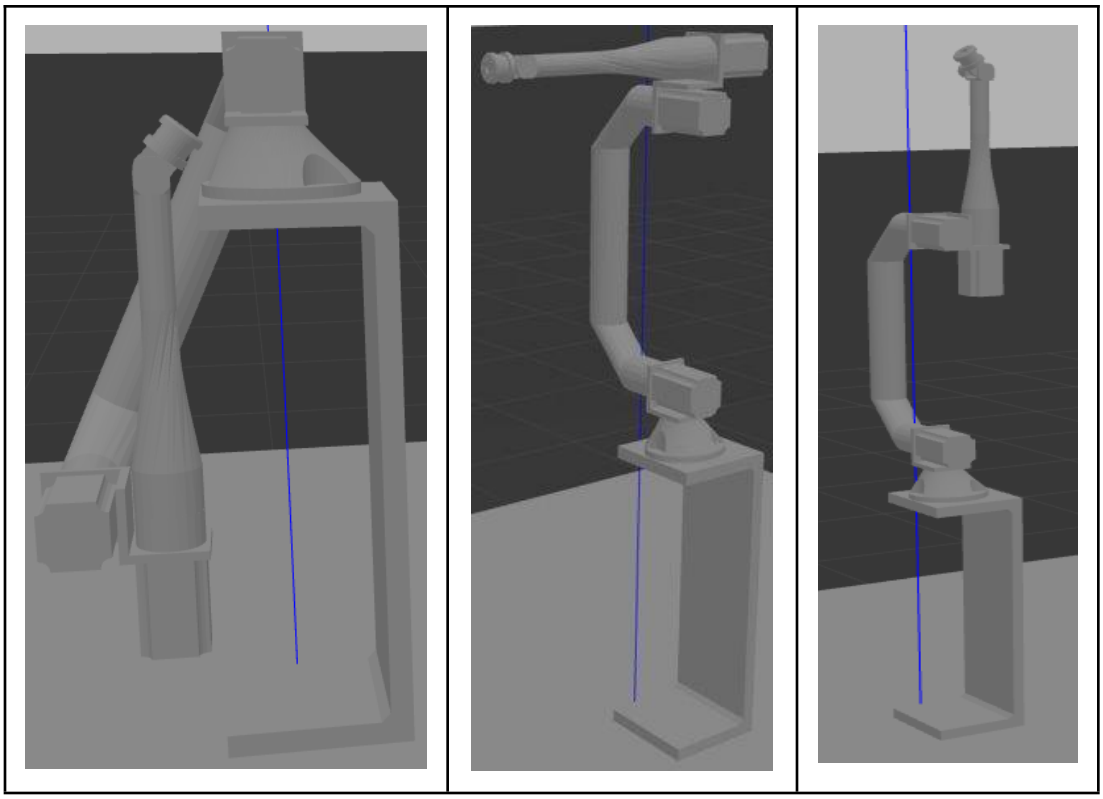


Figure 6. Different Possible Configurations of the SSRA.
Note that for each configuration given above the arm can cover a 360º rotational area

The link to the video showcasing the entirety of the workspace: Reachability

10 Ambitious Goal: Path Planning

During simulation, it is observed that while transitioning between states the robot may take a path that may lead to self-collision or an unnecessarily long path to reach the end goal. This would make the robot useless in a real-world scenario. Hence, the authors have implemented a collision-free optimal path generator between any start and end goal.

To achieve this we are using Moveit. Moveit is a motion planning framework that can be used for navigation and control as well. It is widely used in the industry for simulation and analysis. It provides an efficient way to test the robot in multiple scenarios and understand the fault points in the system.

In this project, we are using Moveit to demonstrate the path planning capability of Moveit for the custom robot arm. The configuration for Moveit needs a correct URDF and a list of available moveable joints. Depending on which it generates the workspace for the robot. It expects three inputs from the user: the start position, the end position, and the path planning algorithm. There are multiple path planning algorithms we can use with Moveit. In this demonstration, we are using RRTStar to generate a path. RRTStar generates a less optimal path as compared to Djikstras or AStar but has a lesser computation time in comparison making it one of the most widely used algorithms.

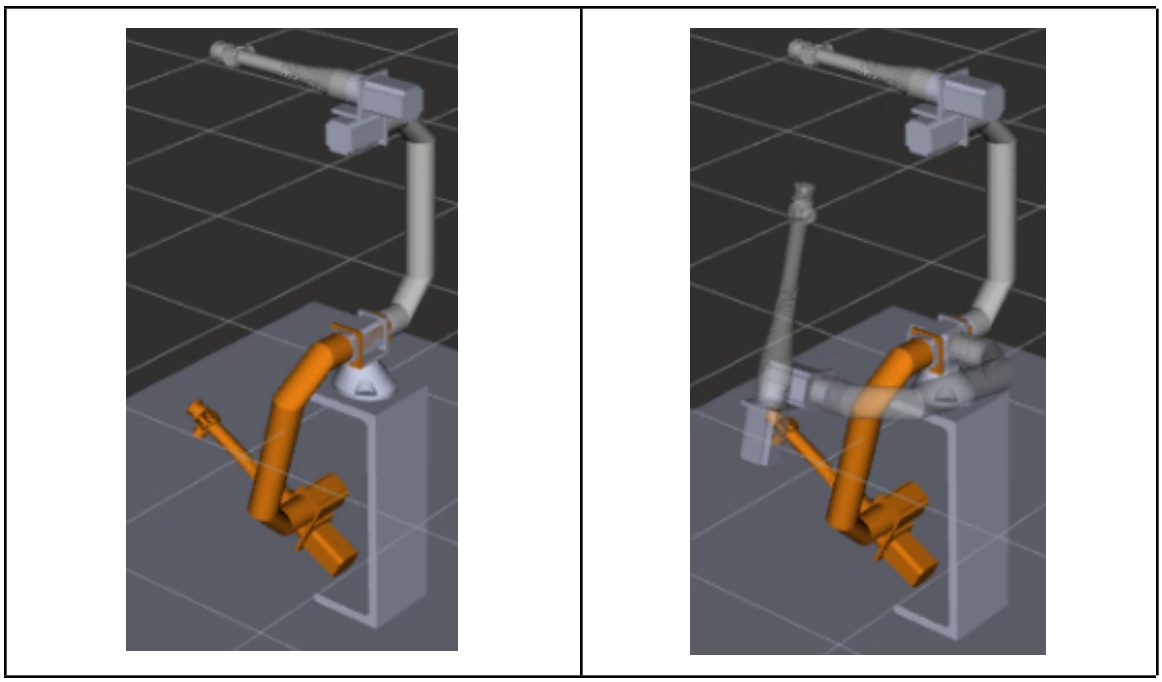


Figure 6. Path Planning. (Left) Start and end goal. (Right) Intermediate state of the path

The link to the video showcasing the path planning capabilities: Path Planning

11 Problems

1. Choosing robot dimensions to avoid self-collision while minimizing weight
2. Robot toppling due to small base and higher position of the center of mass.
3. Unclean reexport to URDF messed up joint origins
4. Precise Control for individual joint angles

12 Lessons Learned

1. Decrease the weight of the robot arm to allow the controllers to control the arms. We can also increase the torques of the controller by adjusting the mechanical reductions or other parameters in the configuration.
2. As it is not easy to fix the base to ground in the gazebo, increase the surface area of the base of the robot so that the arm does not topple.
3. When URDF is generated for the first time it assigns origins at each of the joints. If we change the configuration of the assembly in Solidworks and export the URDF again it uses the previously assigned origins which may not match with the actual joint positions. Hence, clear the previous joint origins created while exporting the URDF.
4. Use Position controllers for revolution joints as they provide angular precision.

13 Conclusions

1. Effort controllers perform better with high torque applications and velocity controllers perform better where speed is of priority.
2. Position controllers provide precise angular movement for revolute joints
3. Spherically Reaching Robotic Arm benefits:
    a. Increased reachable workspace
    b. Increase in the number of configurations for the same end position
    c. Increase in the usable area in a factory line-up setup
    d. Easily configurable to be used for multiple use cases
4. Path planning library benefits:
    a. Smoother transition between configurations
    b. Collision free optimal path generation

14 Future Work

1. Optimize the weight distribution of the links to reduce the overall weight of the arm
2. Simulate the arm to perform specific tasks like pick and place or drilling operations.
3. Implement a closed-loop controller using the joint state data to reduce oscillation of the arm during movement.
4. Integrate the path planning with the navigation to simulate the actual movement of the arm following the path

15 Links

GitHub Repo: https://github.com/RigvedRK/ENPM662-Project2

Arm Reachability: https://youtu.be/Oz01eVTqIAw

Path Planning: https://youtu.be/_ZpoJo-5P70

References

[1] Statista. (2020). *Size of the global industrial automation market from 2019 to 2026*. https://www.statista.com/statistics/1219772/industrial-automation-market-size-worldwide/

[2] Mark W. Spong, Seth Hutchinson, and M. Vidyasagar John Wiley & Sons (2006), Robot Modeling and Control, ISBN-10 0-471-64990-2

[3] http://wiki.ros.org/

[4] https://answers.gazebosim.org/

[5] https://answers.ros.org/