ENPM662: INTRO TO ROBOT MODELLING

Term Project Report on:

Forward and Inverse Kinematic Modelling of a Mobile Manipulator: WALLE

Submitted to:

Dr. Chad Kessens
University of Maryland, College Park

Submitted by:

Abhinav Modi
116346849

Contents

List of Figures

Chapter 1

Introduction

A **Mobile Manipulator** is a robot system in which a robotic manipulator arm is mounted on a mobile platform. These systems incorporte the best of the two worlds. They are fast in locomotion and possess the high level manipulation abilities at the same time.

The mobile platform offers "unlimited" workspace to the manipulator. The extra degrees of freedom provide the user with more choices but also, sometimes increase the challenges for effective system operation.

Today, mobile manipulation robots are a major research area in the field of robotics. Either autonomous or teleoperated, these type of robots are currently being developed for use in multitude of environments: home-care and health-care, military operations, to collect samples in the extreme environment of an active volcano site, or even other planets. Furthermore, these are easy to manufacture and can be developed using off-the-shelf components.

1.1 Motivation

A recent study has shown that the nurses spend about 30-35% of their time "fetching and gathering" general supplies for the hospital rooms. This makes them less available for the immediate help of the patients. A mobile manipulator can be used here to pre-fetch these supplies for the nurses and even get them to the rooms where nurses can just use them.

1.2 Aim

The main idea of this project is to model the Kinematics of a reduced DOF(Degree of Freedom) model of the robot: **WALL-E** from the original Disney PIXAR movie: **WALL-E**. The objective is to perform a pick and lift operation using both the arms together on a mobile base. This will enable the robot to pick a box full of supplies, kept at fixed distance which is more than one arms reach. As a further development to this project, a path planning algorithm can be implemented which will allow the robot to navigate through the corridors of a hospital environment, avoiding collision with potential obstacles to reach the destination.

Chapter 2

Model Description

As shown in the movie, the robot WALL-E has two 10 DOF arms which are attached to a mobile platform. The mobility is provided using a set of tracks on each side as can be seen in the figure, in which two wheels are actuated in each set.

As the name suggests: Waste Allocation Load Lifter Earth-class, WALL-E was used to transport trash from one place to another using different methods due to its high dexterity. For the scope of this project certain changes have been made to reduce the degrees of freedom. The major structural changes are listed as following:

As shown in the movie

- 3 slider mechanisms to move around the shoulder joint as a whole.
- 2 revolute joints in the shoulder to provide rotations about the anterior/posterior axis and the Mediolateral axis.
- 1 Prismatic joint in the arm to help in extending the arm length.
- 3 revolute joints in the hands to move the two fingers and thumb.
- 2 more revolute joint in the wrist to rotate the hand about an axis parallel to the arm and one to rotate the hand about an axis perpendicular to the axis of the arm.

For this project

- 2 revolute joints in the shoulder to provide rotations about the anterior/posterior axis and the Mediolateral axis.
- 1 Prismatic joint in the arm to help in extending the arm length.
- 2 Revolute joints in the wrist to rotate the hand(end effector) about an axis parallel to the arm and one to rotate the hand about an axis perpendicular to the axis of the arm.
- 1 Revolute joint in the finger to move the end effector.

All the modifications wer done using **Solidworks** and **Vrep**. You can see below in fig(2.1) the model that was developed and is used for this project:

The arms are modelled as two 6-DOF manipulators including a spherical wrist. These arms are mounted on a mobile base which is a differential drive system with 2 casters in the front.

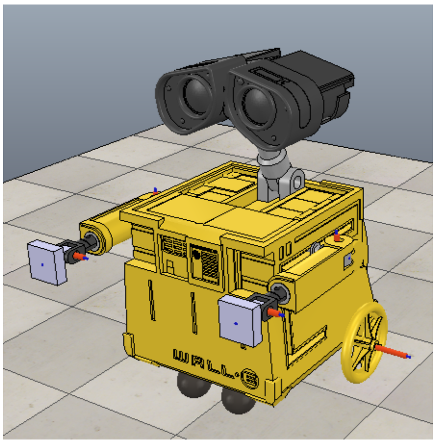

Figure 2.1: Solid model used for simulation and validation purposes

2.1 Link parameters and constraints

To improve replicability and make this robot as accurate as possible with the scale defined in the movie, the following link lengths(more appropriately distances between the joint frames) were chosen:

Note: Since the robot is fiction based, less accuracy can be expected with the dimensions for manufacturing purposes and link lengths may vary if the robot maybe used for real world application

Link Lengths:

1. Length of the Shoulder link: $l_1 = 0.0392$ m
2. Length of the Arm: $l_2 = 0.2000$ m
3. Length of the Elbow: $l_3 = 0.0792$ m
4. Length till the Wrist $l_4 = 0.1620$ m
5. Length to the 2nd Wrist Joint $l_5 = 0.0200$ m
6. Length to the End-effector Center $l_6 = 0.0688$ m

The Joint Constraints are selected based on the freedom of movement, taking into account the possible collisions of the robot with itself and the surroundings.

Joint Constraints:

1. The shoulder joint can move through $-40°$ to $180°$ even with full actuation at the prismatic joint and even to $-60°$ when there is no actuation at the prismatic joint.

2. The 2^{nd} Shoulder joint has the limits $10°$ to $-180°$.

3. The Prismatic Elbow joint has 0 m as the reference position and extends upto $0.2500m$.

4. The Wrist Joints 1 and 3, as mentioned in the scene hierarchy, have no constraint thus ranging from $-180°$ to $180°$.

5. The Wrist joint 2 has the motion range from $-150°$ to $+150°$

2.2 Assumptions and Simplifications

This section lists some of the major assumptions that have been taken into account for developing this model.

1. All bodies are rigid bodies and all joint are considered ideal.

2. To make the CAD model simple and to keep the simulation time less, the tracks are replaced with two actuated wheels and a caster wheel in the front.

3. All joints are considered ideal and thus, no friction or compliance is taken into account.

4. The casters are considered ideal, therefore, pose no constraint in the deriving the kinematics.

5. Friction is considered between the tires and ground such that the robot moves with a no slip condition. Also, friction between the hands and the objects is also taken into account.

6. Restrictions and constraints in the joints are explained in the Model Description section.

7. There are different types of objects that this robot can handle, but for the simplification of the simulation only a rectangular box is considered.

8. For the final task the obstacles are considered stationary, which might not be the case in a real scenario but the simulation of moving objects is out of the scope of the project.

9. The direction of the gravitational force is taken as the negative direction of the Z axis of the world coordinate frame.

10. The kinematic modelling of the system is done as a decoupled system, decoupling the manipulator kinematics annd the mobile base kinematics. It is assumed that the robot will never execute both motions simultaneously.

Chapter 3

Methodology

Here is a brief description of the plan of implementation. The project was divided into milestones, which are as follows:

Task 1 : Model the Forward and Inverse Kinematics in MATLAB.

Task 2 : Develop the mathematics for Dynamic Modelling of the robot.

Task 3 : Pick a rectangular box off the ground using both hands keeping the orientation of the object constant.

This is the most primitive task this type of robot will perform. For eg., picking up a supply item from its respective place so that it can be taken from this place to its required destination.

Task 4 : Place the object on top of another structure(for eg., a Table) and move back.

When this robot brings a supply box to a room it needs to keep the box in an accessible place and move back so that a nurse can use it while the robots waits for a following command.

Task 5 : Take an object from point A to point B where both the points lie on a straight line.

This is a simpler version of the task, which is the main design objective if the robot, that is to carry a box from the supply area to the hospital room.

Task 6 : Object Avoidance for carrying an object from point A to point B on a collision free path.

This will be the ultimate objective of this robot to carry the box of supplies to the required destination while avoiding obstacles in the hospital corridors for eg., chairs, humans standing in the way and any other similar objects.

Note: My goal for this project will be to complete till task 4. The tasks 5 and 6 will be completed based on the time left. Task 3 can be considered my fallback goal and task 4 as the goal for this project to be termed successful.

3.1 Robot Modelling

The Modelling of the robot starts with the Forward Kinematics of the Manipulator Arms and the projection of the COM of the base is considered as the origin of the base(w) frame. This modelling is then extended to find the Inverse Kinematics of the arms. The Kinematics of the mobile base comes next.

3.1.1 Forward Kinematics of the Manipulator Arms

The figure(3.13) below shows the D-H frame assignment with respect to the base(w) frame.

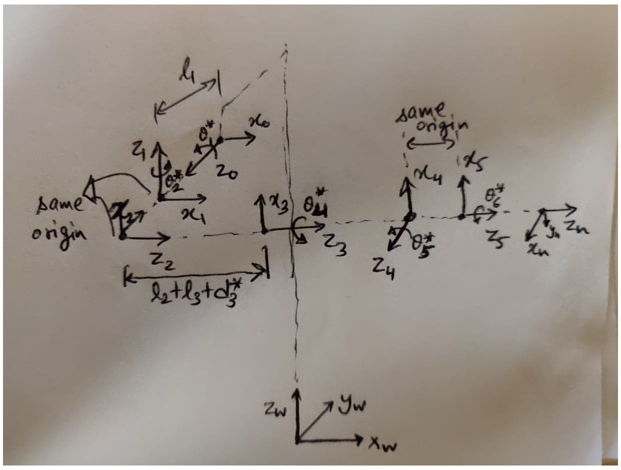

Figure 3.1: D-H frame assignment

A DH table based on this frame assignment was considered.

Note: The DH table was constructed with the shoulder joint as the base frame. Once the DH table is constructed the full transformation matrix is pre-multiplied with a transformation matrix that gives the end effector position and orientation w.r.t. the desired base frame

The DH parameters for robot's right arm are mentioned below:

T_i	θ_i	d_i	a_1	α_i
1.	θ_1^*	l_1	0	$-\pi/2$
2.	$\theta_2^* + \pi/2$	0	0	$\pi/2$
3.	$\pi/2$	$l_2 + l_3 + d_3^*$	0	0
4.	θ_4^*	l_4	0	$-\pi/2$
5.	θ_5^*	0	0	$\pi/2$
n.	$\theta_6^* + \pi/2$	$l_5 + l_6$	0	0

Using the DH table we can find the end effector position in shoulder coordinate frame using:

$$T_n^0 = T_1^0 * T_2^1 * T_3^2 * T_4^3 * T_5^4 * T_n^5 \tag{3.1}$$

where each individual transformation matrix is written as follows:

$$T_i^{i-1} = Rot(z_i) * Trans(z_i) * Trans(x_i) * Rot(x_i) \tag{3.2}$$

As a final step the T_n^0 is pre-multiplied with the matrix A_0 where,

$$A_0 = T_b^w * T_0^b \tag{3.3}$$

where,

$$T_b^w = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0.6000 \\ 0 & 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & cos(\pi/2) & -sin(\pi/2) & 0 \\ 0 & sin(\pi/2) & cos(\pi/2) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.4}$$

$$T_0^b = \begin{bmatrix} 1 & 0 & 0 & 0.0900 \\ 0 & 1 & 0 & -0.3508 \\ 0 & 0 & 1 & -0.0050 \\ 0 & 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & cos(\pi/2) & -sin(\pi/2) & 0 \\ 0 & sin(\pi/2) & cos(\pi/2) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.5}$$

and T_b^w takes the world frame(projection of the COM of the mobile base) to the center of the mobile base and T_0^b takes the center frame to the shoulder coordinate frame. Thus, the final transformation matrix for the end-effector position in the world coordinate frame can be obtained as:

$$T_n^w = T_b^w * T_0^b * T_1^0 * T_2^1 * T_3^2 * T_4^3 * T_5^4 * T_n^5 \tag{3.6}$$

Similarly, the DH table and the transformation matrices were developed for the left arm with slight changes:

T_i	θ_i	d_i	a_1	α_i
1.	θ_1^*	$-l_1$	0	$-\pi/2$
2.	$\theta_2^* + \pi/2$	0	0	$\pi/2$
3.	$\pi/2$	$l_2 + l_3 + d_3^*$	0	0
4.	θ_4^*	l_4	0	$-\pi/2$
5.	θ_5^*	0	0	$\pi/2$
n.	$\theta_6^* + \pi/2$	$l_5 + l_6$	0	0

$$T_b^w = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0.6000 \\ 0 & 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & cos(\pi/2) & -sin(\pi/2) & 0 \\ 0 & sin(\pi/2) & cos(\pi/2) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.7}$$

$$T_0^b = \begin{bmatrix} 1 & 0 & 0 & 0.0900 \\ 0 & 1 & 0 & 0.3508 \\ 0 & 0 & 1 & -0.0050 \\ 0 & 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & cos(\pi/2) & -sin(\pi/2) & 0 \\ 0 & sin(\pi/2) & cos(\pi/2) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.8}$$

3.1.2 Validation of Forward Kinematics

The validation of Forward Kinematics was done by finding out the end effector position for two different joint variable configurations, $q = [\theta_1, \theta_2, d_3, \theta_4, \theta_5, \theta_6]$

Case 1. The first test was done for the home configuration, $q = [0, 0, 0, 0, 0, 0]$. The result obtained by using the calculations based on the eqn(3.6) are shown in fig(3.2):

For the same joint angle configuration, q the results obtained from v-rep are shown in fig():

Twn_1 =

$$\begin{pmatrix} 0 & 0 & 1.0 & 0.62 \\ -6.123\ 10^{-17} & 1.0 & 0 & -0.39 \\ -1.0 & -6.123\ 10^{-17} & 0 & 0.595 \\ 0 & 0 & 0 & 1.0 \end{pmatrix}$$

Figure 3.2: Calculated Transformation matrix

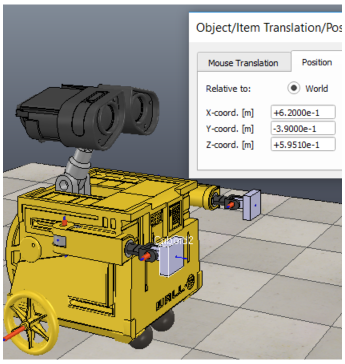

Figure 3.3: Obtained Position matrix case1

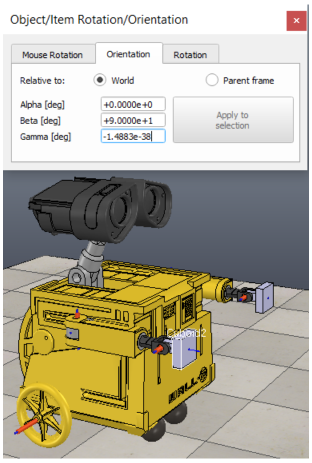

Figure 3.4: Obtained Orientation case1

The fig(3.2) represents the euler angles obtained using v-rep and we can compute the orientation matrix which comes out to be the same as the orientation part shown in fig(3.2).

Case 2. The second test was done for the configuration, $q = [\pi/3, 0, 0.1, 0, \pi/6, 0]$. The result obtained by using the calculations based on the eqn(3.6) are:

Twn_1 =

$$\begin{pmatrix} 1.0 & 0 & 0 & 0.3606 \\ 0 & 1.0 & 6.123\ 10^{-17} & -0.39 \\ 0 & -6.123\ 10^{-17} & 1.0 & 1.152 \\ 0 & 0 & 0 & 1.0 \end{pmatrix}$$

Figure 3.5: Calculated Transformation matrix case2

For the same joint angle configuration, q the results obtained from v-rep are:

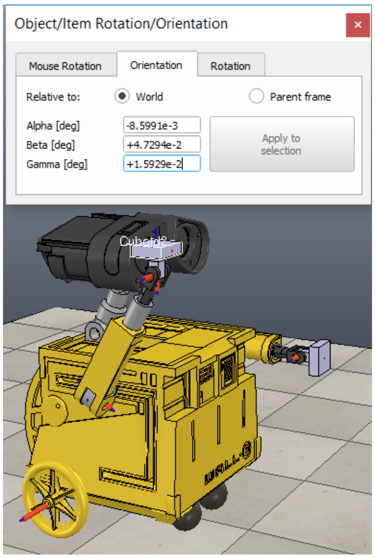

Figure 3.6: Obtained Orientation case2

As it can be seen from the above results, the end-efector position and orientation obtained through calculations and the simulated system are same atleast to the last 0.1 cm and 1°. The slight difference that is observed here is because of issues in inaccurate CAD modelling and the dynamics engine used by v-rep. Even then the results seem very accurate and it can be claimed that the calculation of the Forward Kinematics of the system is validated.

3.1.3 Inverse Kinematics of Manipulator Arms

Now, that we have a Forward Kinematic model of the system, we can develop an inverse kinematic model for the system. As mentioned in the previous chapter, the manipulator arms are 6 DOF where the last three joints constitute a spherical wrist. This type of system can be kinematically decoupled and the inverse position and the inverse orientation problem can be solved seperately. The following formulation has been done for only the right arm of the robot. The inverse kinematics for the left arm of the robot can be calculated similarly.

Inverse Position Problem

The presence of a spherical wrist means that the axes of the joints 3, 4 and 5 intersect at o_c(wrist center) and hence the frames 4 and 5 assigned by the DH convention will always be at the wrist center. The important point here is that the motion of the final 3 links about these axis will not change the position of o_c. Thus, the position of the wrist centeris thus a function of only the first three joint variables.

In case of WALLE, we can write that given any desired end-efector position o_d :

$$o_c = o_d - (l_5 + l_6)R_d \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \tag{3.9}$$

where R_d is the desired orientation and,

$$R_d = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \tag{3.10}$$

Thus,

$$\begin{bmatrix} x_c \\ y_c \\ z_c \end{bmatrix} = \begin{bmatrix} o_d^x - (l_5 + l_6)r_{13}^d \\ o_d^y - (l_5 + l_6)r_{23}^d \\ o_d^z - (l_5 + l_6)r_{33}^d \end{bmatrix} \tag{3.11}$$

Here, the end-effector position and orientation are considered in the world coordinate frame. Now, given the position of the origin of shoulder frame in world coordinates as (x_0, y_0, z_0), we can geometrically solve for the joint varibles θ_1^*, θ_2^* and d_3^* for a given $o_c = [x_c, y_c, z_c]^T$.

Connsider the top view of the system in fig(3.7). Using this projection we can define

$$L = \sqrt{(x_c - x_0)^2 + (y_c - y_0 + l_1)^2} \tag{3.12}$$

and thus using $\theta = atan2(X, Y)$ we can write,

$$\theta_2^* = atan2((x_c - x_0), (y_c - y_0 + l_1)) \tag{3.13}$$

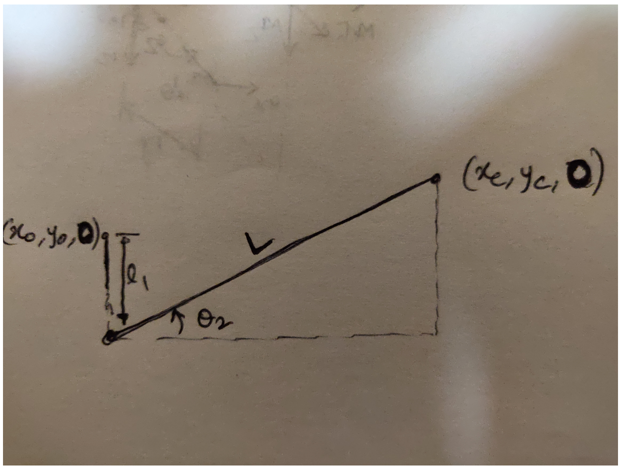

Figure 3.7: Top View

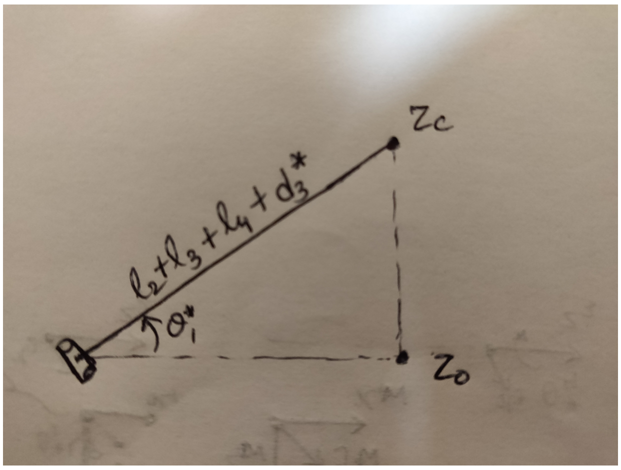

Figure 3.8: Side View

Now, considering the side view as shown in fig(3.8), we can write

$$d_3^* = \sqrt{(z_c - z_0)^2 + L^2} - l_2 - l_3 - l_4 \tag{3.14}$$
$$\theta_1^* = atan2(L, (z_c - z_0)) \tag{3.15}$$

Inverse Orientation Problem

As we have the joint values using which we can define the position of the wrist center, we can now find the remaining three joint variables for defining the end effector orientation. We know,

$$A_0 * R_3^0 * R_6^3 = R_d \tag{3.16}$$

Therefore, we can write,

$$R_6^3 = (A_0 * R_3^0)^T * R_d \tag{3.17}$$

We know that the R_6^3 can be written as the R_{ZYZ} Euler-Angle Transformation as follows:

$$R_6^3 = \begin{bmatrix} s\theta_4 s\theta_6 - c\theta_4 c\theta_5 c\theta_6, & c\theta_6 s\theta_4 + c\theta_4 c\theta_5 s\theta_6, & c\theta_4 s\theta_5 \\ -c\theta_4 s\theta_6 - s\theta_4 c\theta_5 c\theta_6, & -c\theta_6 c\theta_4 + s\theta_4 c\theta_5 s\theta_6, & s\theta_4 s\theta_5 \\ c\theta_6 s\theta_5, & -s\theta_5 s\theta_6, & c\theta_5 \end{bmatrix} = R_3^0 * R_d \tag{3.18}$$

Using this equation we can deduce that, from the last column and the last row of R_6^3,

$$c\theta_4 s\theta_5 = r_{33} c\theta_1 - r_{13} s\theta_1 \tag{3.19}$$
$$s\theta_4 s\theta_5 = -r_{23} c\theta_2 + r_{13} c\theta_1 s\theta_2 + r_{33} s\theta_1 s\theta_2 \tag{3.20}$$
$$c\theta_5 = r_{13} c\theta_1 c\theta_2 + r_{23} s\theta_2 + r_{33} s\theta_1 c\theta_2 \tag{3.21}$$
$$c\theta_6 s\theta_5 = r_{11} c\theta_1 c\theta_2 + r_{31} c\theta_2 s\theta_2 + r_{21} s\theta_2 \tag{3.22}$$
$$-s\theta_6 s\theta_5 = r_{12} c\theta_1 c\theta_2 + r_{32} c\theta_2 s\theta_1 + r_{22} s\theta_2 \tag{3.23}$$

Thus, assuming $\theta_5 \neq 0$, we have two solutin sets which can be written as:

Case 1. $\theta_5 > 0$:

$$\theta_5 = atan2(r_{13} c\theta_1 c\theta_2 + r_{23} s\theta_2 + r_{33} s\theta_1 c\theta_2, \sqrt{1 - (r_{13} c\theta_1 c\theta_2 + r_{23} s\theta_2 + r_{33} s\theta_1 c\theta_2)^2}) \tag{3.24}$$

and,

$$\theta_4 = atan2((r_{33} c\theta_1 - r_{13} s\theta_1), (-r_{23} c\theta_2 + r_{13} c\theta_1 s\theta_2 + r_{33} s\theta_1 s\theta_2)) \tag{3.25}$$

$$\theta_6 = atan2((r_{11} c\theta_1 c\theta_2 + r_{31} c\theta_2 s\theta_2 + r_{21} s\theta_2), - (r_{12} c\theta_1 c\theta_2 + r_{32} c\theta_2 s\theta_1 + r_{22} s\theta_2)) \tag{3.26}$$

Case 2. $\theta_5 < 0$:

$$\theta_5 = atan2(r_{13} c\theta_1 c\theta_2 + r_{23} s\theta_2 + r_{33} s\theta_1 c\theta_2, - \sqrt{1 - (r_{13} c\theta_1 c\theta_2 + r_{23} s\theta_2 + r_{33} s\theta_1 c\theta_2)^2}) \tag{3.27}$$

and,

$$\theta_4 = atan2(-(r_{33}c\theta_1 - r_{13}s\theta_1), -(-r_{23}c\theta_2 + r_{13}c\theta_1 s\theta_2 + r_{33}s\theta_1 s\theta_2)) \tag{3.28}$$

$$\theta_6 = atan2(-(r_{11}c\theta_1 c\theta_2 + r_{31}c\theta_2 s\theta_2 + r_{21}s\theta_2), \\ (r_{12}c\theta_1 c\theta_2 + r_{32}c\theta_2 s\theta_1 + r_{22}s\theta_2)) \tag{3.29}$$

Now, if $\theta_5 = 0$ we get a set of infinite solutions for θ_4 and θ_6 as:

$$\theta_4 + \theta_6 = atan2(-(r_{31}c\theta_1 - r_{11}s\theta_1), (r_{32}c\theta_1 - r_{12}s\theta_1)) \tag{3.30}$$

If $\theta_5 = 0$, for simulation purposes it is assumed that $\theta_4 = 0$ to get a unique value of θ_6.

3.1.4 Validation of Inverse Kinematics

The solution obtained using the inverse position kinematics is a unique solution and the two solutions for the cases of different signs of θ_2 were accounted for in the simulation. The validation of the inverse kinematics was also done by selecting three different end-effector positions and orientations. Then the above formulation is used to calculate the joint variables required by the robot to attain that pose. The required joint variables are calculated using MATLAB and and passed to the v-rep model. The pose of the robot is then compared to the required pose, which should be same.

Case 1. The first test was done for the home configuration, $q = [0, 0, 0, 0, 0, 0]$. The results obtained in the previous subsections are used to calculate the required joint angle configurations which come out to be as expected: Passing these joint variables to the robot in vrep, the results were obtained

Case 2. The second test was done for the required transformation, $q = [pi/3, 0, 0.1, 0, pi/6, 0]$. The results obtained in the previous subsections are used to calculate the required joint angle configurations which come out to be as: Passing these joint variables to the robot in vrep, the results were obtained

The two cases used to validate the inverse kinematics were taken from the forward kinematic validation to show that the results obtained using both the formulations are consistent. It can be observed from the two cases that the inverse kinematics formulation is efficient in providing a unique solution

Tik1 =

$$\begin{pmatrix} 0 & 0 & 1.0 & 0.62 \\ 1.225\ 10^{-16} & -1.0 & 0 & -0.39 \\ 1.0 & 1.225\ 10^{-16} & 0 & 0.595 \\ 0 & 0 & 0 & 1.0 \end{pmatrix}$$

q1 =

$$\begin{pmatrix} 0 \\ 0 \\ 2.776\ 10^{-17} \\ 0 \\ 0 \\ 3.142 \end{pmatrix}$$

Figure 3.9: Calculated Joint Variables - case1

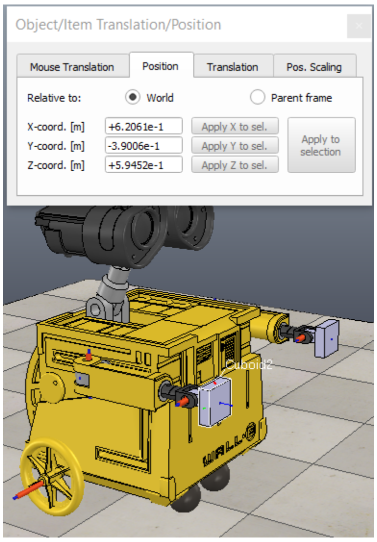

Figure 3.10: Obtained robot pose - case1

for a required end effector pose for all the test cases, taking into account the joint constraints as mentioned in the previous chapter. Thus, we can safely assume that the algorithm is correct and has been validated through the resuts shown above.

Tik1 =

$$\begin{pmatrix} 1.0 & -9.073\ 10^{-45} & 6.494\ 10^{-33} & 0.3606 \\ 9.072\ 10^{-45} & 1.0 & 6.123\ 10^{-17} & -0.39 \\ -6.494\ 10^{-33} & -6.123\ 10^{-17} & 1.0 & 1.152 \\ 0 & 0 & 0 & 1.0 \end{pmatrix}$$

q1 =

$$\begin{pmatrix} 1.047 \\ -1.21\ 10^{-12} \\ 0.1 \\ -2.096\ 10^{-12} \\ 0.5236 \\ 2.42\ 10^{-12} \end{pmatrix}$$

Figure 3.11: Calculated Joint Variables - case2

later be combined to compute the motion of the robot as a whole

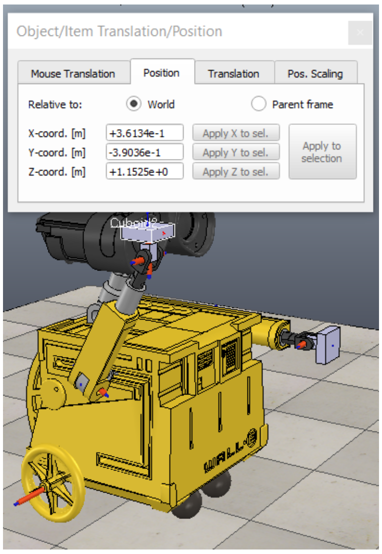

Figure 3.12: Obtained robot pose - case2

3.1.5 Kinematics of Mobile Base

Now, that the robot is able to achieve any end-effector pose that's within "reach", when the robot is stationary. In this subsection we model the Kinematics of the mobile base. This enables the robot to carry the manipulators to any desired point in the X-Y plane, making the possible workspace to be the entire X-Y plane in case there are no obstacles in the way. For this project we consider a differential drive based system with two fixed wheels(nonsteerable) and 2 casters in the front, which help in providing stability.

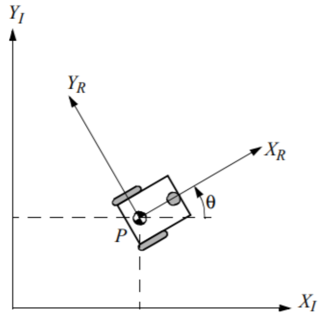

Figure 3.13: Global reference frame and robot local reference frame

As shown in the fig(3.13), the pose of the robot can be defined in the inertial frame as:

$$\xi_I = \begin{bmatrix} x \\ y \\ \theta \end{bmatrix} \tag{3.31}$$

As the robot's motion is governed by its components(wheels), it is necessary to map motion of the robot's local reference frame to the world frame. This mapping is obtained using the orthogonal rotation matrix for rotation about $z - axis$,

$$R(\theta) = \begin{bmatrix} cos(\theta) & sin(\theta) & 0 \\ -sin(\theta) & cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{3.32}$$

This equation is used to map the motion in local frame (X_R, Y_R) to the global reference frame (X_I, Y_I) by:

$$\dot{\xi}_I = R(\theta)\dot{\xi}_R \tag{3.33}$$

Forward Kinematics of the Mobile Base

In simple cases the eqn(3.33) is sufficient to provide a mapping between the two frames. Moreover, we can write the linear and angular velocity of the system as

$$v = \frac{r_1\dot{\phi}_r + r_2\dot{\phi}_l}{2} \tag{3.34}$$

$$\omega = \frac{r_1\dot{\phi}_r - r_2\dot{\phi}_l}{2d} \tag{3.35}$$

where $\dot{\phi}_l$, $\dot{\phi}_r$ are angular velocities of the left and right wheels respectively, r_1, r_2 is the radius of the corresponding wheel and $2d$ is the distance between the two wheels.

Kinematics Constraints in the system and Inverse Kinematics

To make the kinematic model accurate, the system is modelled with the constraints included in the kinematic equations. Our system constitutes of two fixed wheels and 2 casters. Since the casters are passicve and assumed to be ideal, they pose no kinematic constraint as they align themselves with the direcion of motion. Thus, only fixed wheels are considered for generating the constraint equations.

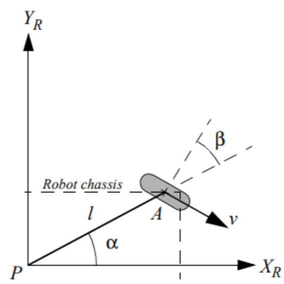

Figure 3.14: Fixed Wheel Kinematic Constraints

Figure(3.14) shows a fixed wheel in the robot's local frame. The constraint in the rolling direction can be writtern as:

$$\left[sin(\alpha + \beta), \quad cos(\alpha + \beta), \quad (-l)cos(\beta)\right] * R(\theta) * \dot{\xi}_I = r\dot{\phi} \tag{3.36}$$

where, α is the angular displacement of wheel center measured from the robot's X-axis and β is the angle between rotation axis and the line joining wheel center to the robot's origin and l is the distance of wheel center from the robot's origin.

For the two wheels in the system the above constraints can be combined in a matrix form and written as

$$\begin{bmatrix} sin(\alpha_1+\beta_1), & -cos(\alpha_1+\beta_1), & (-l)cos(\beta_1) \\ sin(\alpha_2+\beta_2), & -cos(\alpha_2+\beta_2), & (-l)cos(\beta_2) \end{bmatrix} * R(\theta) * \dot{\xi}_I = \begin{bmatrix} r\dot{\phi}_l \\ r\dot{\phi}_r \end{bmatrix} \tag{3.37}$$

where α and β for each wheel are:

$$\alpha_1 = 105.4612°; \quad \beta_1 = 15.4612° \tag{3.38}$$

$$\alpha_2 = -105.4612°; \quad \beta_2 = 164.5388° \tag{3.39}$$

and $l = 0.4876$ m.

Note: The constraint defined here is for pure rolling and is based on the fact that the robot follows a path with no slip in the lateral direction.

Using the equation(3.37) we have defined the relation between the desired velocity in the world frame and the angular velocity of each wheel. Hence, we can now define a trajectory and obtain the angular velocities required to follow it.

Validation of Mobile Base Kinematics

The code for the validation of the above equation is availabl in 'mobile.m' file shown by choosing two simple trajectories:

1. $(x(t), y(t)) = (0.1t, 0)$. The velocities required for following this trajectory are thus $\dot{x} = 0.1m/s, \dot{y} = 0$ and $\dot{\theta} = 0$. This system was simulated in MATLAB and the joint angular velocities for each wheel were calculated and then sent to v-rep. The bot thus follows the given path, as shown in the figure below. Here are different time snapshots of the simulation.

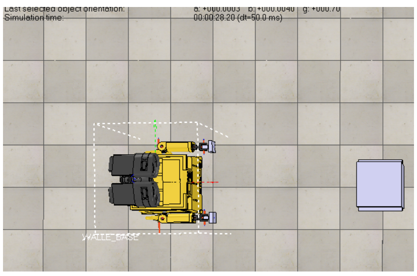

Figure 3.15: Trajectory followed by the robot at t1 = 28s

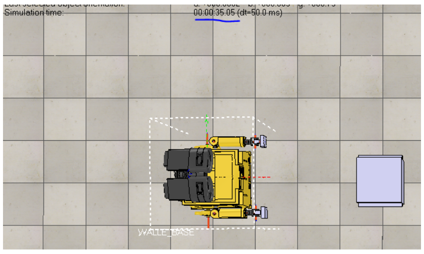

Figure 3.16: Trajectory followed by the robot at t2 = 35s

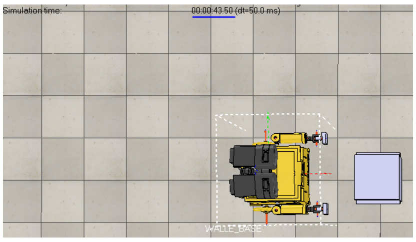

Figure 3.17: Trajectory followed by the robot at t3 = 43s

2. $(x(t), y(t)) = (0.1t, 0.1t)$ The velocities required for following this trajectory are thus $\dot{x} = 0.1m/s$,$\dot{y} = 0.1m/s$ and $\dot{\theta} = 0$. This system was simulated in MATLAB and the joint angular velocities for each wheel were calculated and then sent to v-rep. The bot thus follows the given path, as shown in the figure below.

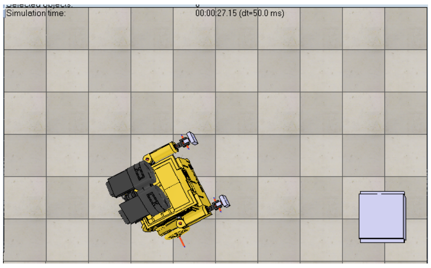

Figure 3.18: Trajectory followed by the robot at t1 = 27s

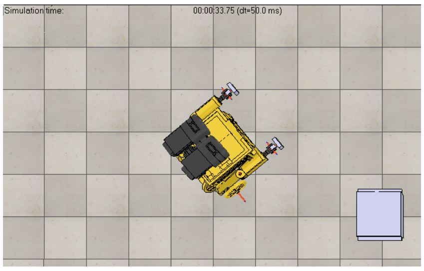

Figure 3.19: Trajectory followed by the robot at t2 = 33s

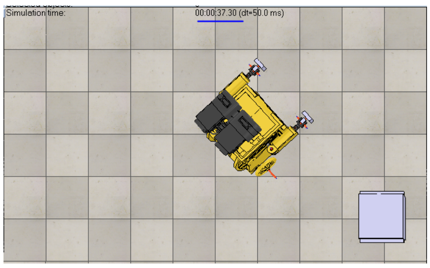

Figure 3.20: Trajectory followed by the robot at t3 = 37s

Chapter 4

Simulations and Results

Based on the equations developed in the previous chapter, the robot is now capable of completing multiple tasks. As per the planned milestones, the following have been completed:

Task 1 : Model the Forward and Inverse Kinematics in MATLAB.

Task 3 : Pick a rectangular box off the ground using both hands keeping the orientation of the object constant.

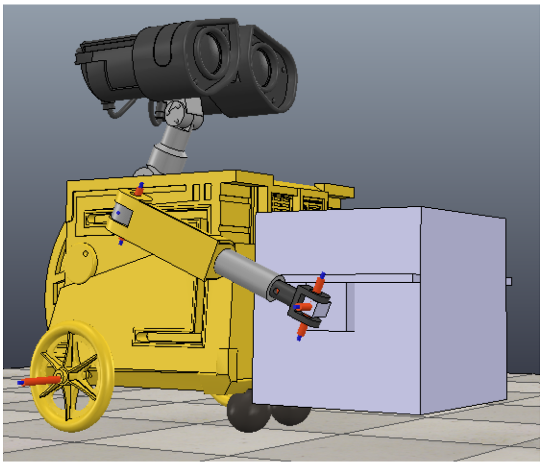

Figure 4.1: Walle picking up a box from the ground

Task 4 : Place the object on top of another structure(for eg., a Table) and move back.

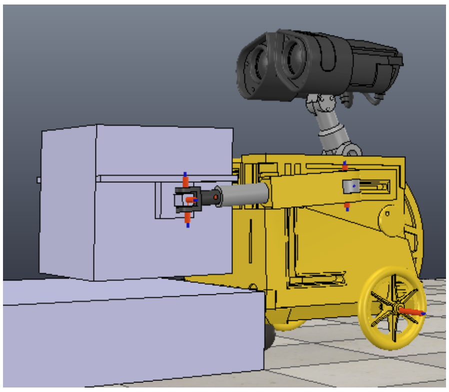

Figure 4.2: Walle leaving a box on the table

Task 5 : Take an object from point A to point B where both the points lie on a straight line.

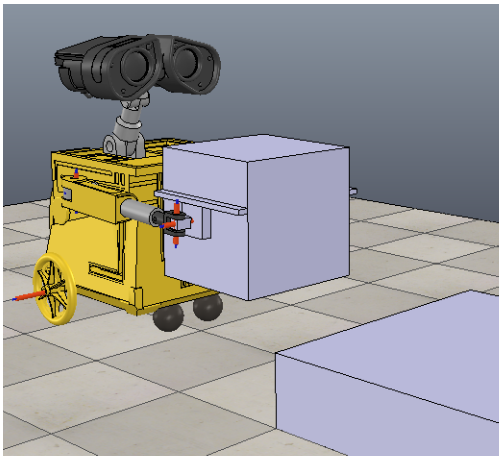

Figure 4.3: Walle moving from point A to point B with the box

Grasping a simple cuboid was tried first, but due to vibrations in the system and noise due to the v-rep simulation engine, the shape of the box was changed slightly to provide extra lifing capabilities.

The videos for the robot moving from one point to another and picking up the objects and placing them, is present in the 'modelling.zip' folder attached with this document. The codes used for the simulations and validations are available in the attached zip folder. To run the codes for simulations see the Appendix.

Chapter 5

Conclusion and Future Scope

The aim of this project was to model a mobile manipulator robot to complete certain tasks in a hospital environment. Thus, Kinematic Modelling of a decoupled system was carried out to simulate those tasks. The modelling was done by decoupling the system, with the assumption that both the systems will not be used simultaeously. This is a problem for the future scope. As I worked through this project, I have realized that even though the concept of mobile manipulators is quite old but still there still is scope for improvement.

Certain goals like grasping of a cuboid could not be achieved due to inaccurate CAD design and time constraints. Moreover, there is no reference of the robot model available online to verify masses and inertias and even dimensions of the robot's components, leading to arbitrary choices of torque values for simulation purposes. Such problems can be addressed as the future scope which will require work on accurate CAD design, dynamic modelling, stability and tip over analysis, etc.

In the end, I would like to thank Prof. Chad Kessens for allowing me to work on this project and also the TAs for supporting and helping me throughout the course.

Bibliography

[1] Disney,PIXAR *movie WALL-E(2008)*.
https://www.pixar.com/feature-films/walle/#walle-character-design

[2] Mobile Manipulator Robots.
https://en.wikipedia.org/wiki/Mobile_manipulator

[3] Nursing Times.
https://www.nursingtimes.net/nurses-waste-an-hour-a-shift-finding-equipment/1987381.article

[4] Robot Modeling and Control, by Spong, M.W. and Hutchinson, S. and Vidyasagar, M..
https://books.google.com/books?id=muCMAAAACAAJ

[5] E. Rohmer, S. P. N. Singh, M. Freese, "V-REP: a Versatile and Scalable Robot Simulation Framework," IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, 2013

Appendix

The folder modelling.zip contains two folders:

1. Videos
2. Code

The Videos folder contains simulation videos of 2 tasks which are mainly picking an object, travelling to another point and dropping the object. To run these simulations follow the steps written below:

1. Forward_ Kinematics.m
2. Inverse_ Kinematics.m
3. mobile.m
4. sim_test_ground.m
5. sim_test_table.m
6. Walle.ttt
7. walle_table.ttt

All the matlab files can be used to run both the '*.ttt' files in v-rep. The first three were used for validation of the Forward, Inverse and Mobile robot kinematics respectively. They are based on the mathematics developed in the Methodology chapter. The 'sim_test_ground.m' and 'sim_test_table.m' are the two main simultaion files. The former is the simulation of the robot going from a point to an object following the mobile kinematic constraints and once the object is within reach of the manipulator, the inverse kinematics kicks in to help the robot lift the object from the ground. This code will run with the Walle.ttt file.

The 'sim_test_table.m' is the simulation of the robot going to a raised platform on which an object is kept, picking the object and then moving back to the original place with the object held and then finally putting it on the ground. The codes are adequately commented and can be changed to make the robot do different tasks. The other files present in the folder are necessary for bridging the connection with V-REP. So, to ensure smooth running of files, make sure to keep all the files in the same folder.