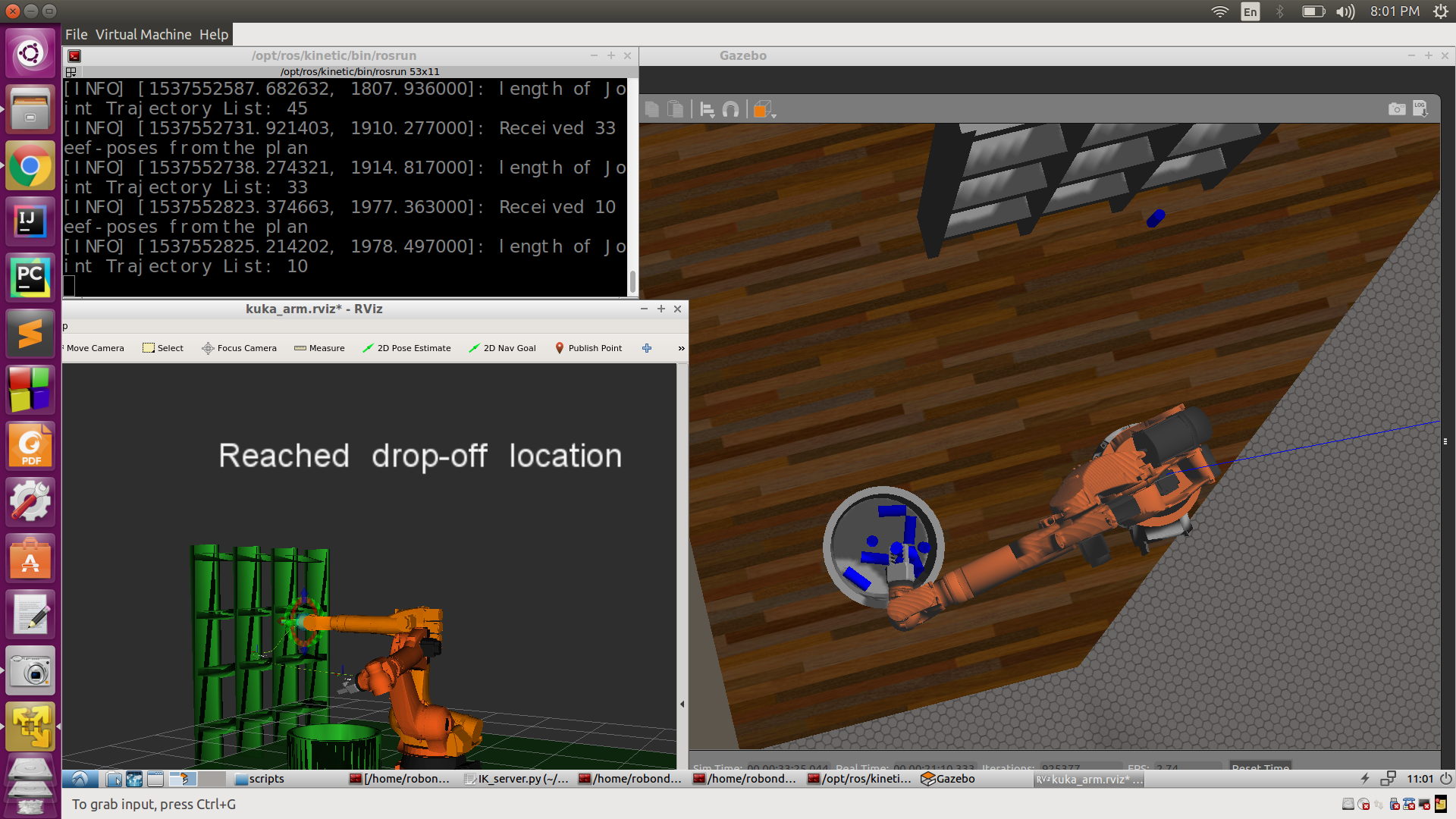
***Kuka KR210 Pick and Place Project***

This project was simulated by kuka KR210 which has 6 degree of freedom using gazebo, Rvis and ROS.

***Objective:***

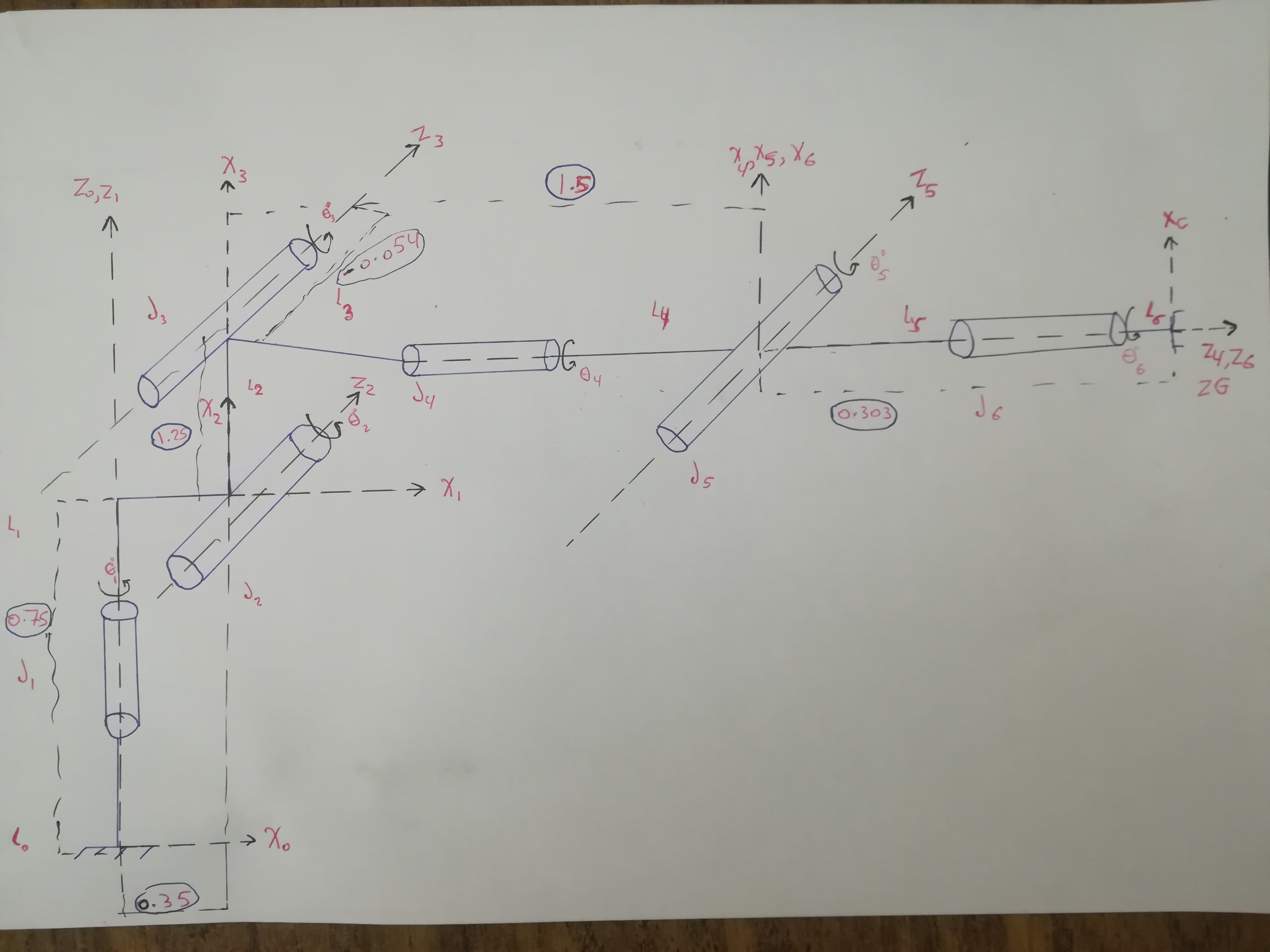
Is to detect a cylindrical object which spawn on one of the shelves randomly and grab it and throw it in the bin by calculating the path using kinematics and inverse kinematics techniques.

Successful pick and place!

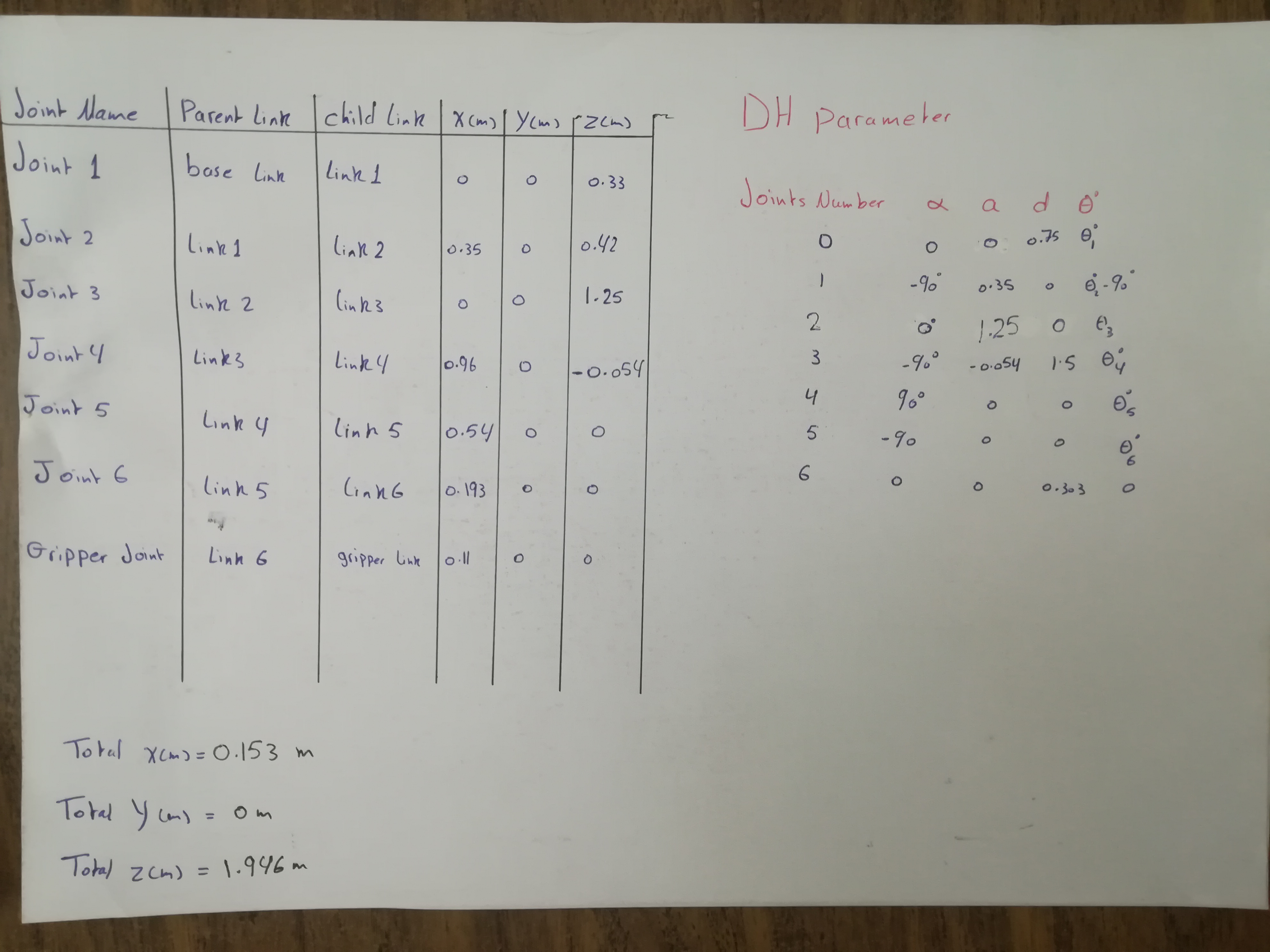


***Kinemtaics Analysis :***

First is to identify the XYZ axis of each joint.



Then drive the DH parameters and evaluate the constant values from the udrf file of the kuka arm



Build the transformation matrices and drive the transformation matrix from ground to end effector

*T0\_1 = TM\_Generator(alpha0, a0, d1, q1).subs(DH\_table)*

*T1\_2 = TM\_Generator(alpha1, a1, d2, q2).subs(DH\_table)*

*T2\_3 = TM\_Generator(alpha2, a2, d3, q3).subs(DH\_table)*

*T3\_4 = TM\_Generator(alpha3, a3, d4, q4).subs(DH\_table)*

*T4\_5 = TM\_Generator(alpha4, a4, d5, q5).subs(DH\_table)*

*T5\_6 = TM\_Generator(alpha5, a5, d6, q6).subs(DH\_table)*

*T6\_G = TM\_Generator(alpha6, a6, d7, q7).subs(DH\_table)*

*T0\_G = T0\_1 \* T1\_2 \* T2\_3 \* T3\_4 \* T4\_5 \* T5\_6 \* T6\_G*

Apply extrinsic rotation and gripper fixing orientation

*Rot\_Fixed = z\_rot.subs(y, radians(180)) \* y\_rot.subs(p,radians(-90))*

*ROT\_Error = z\_rot \* y\_rot \* x\_rot*

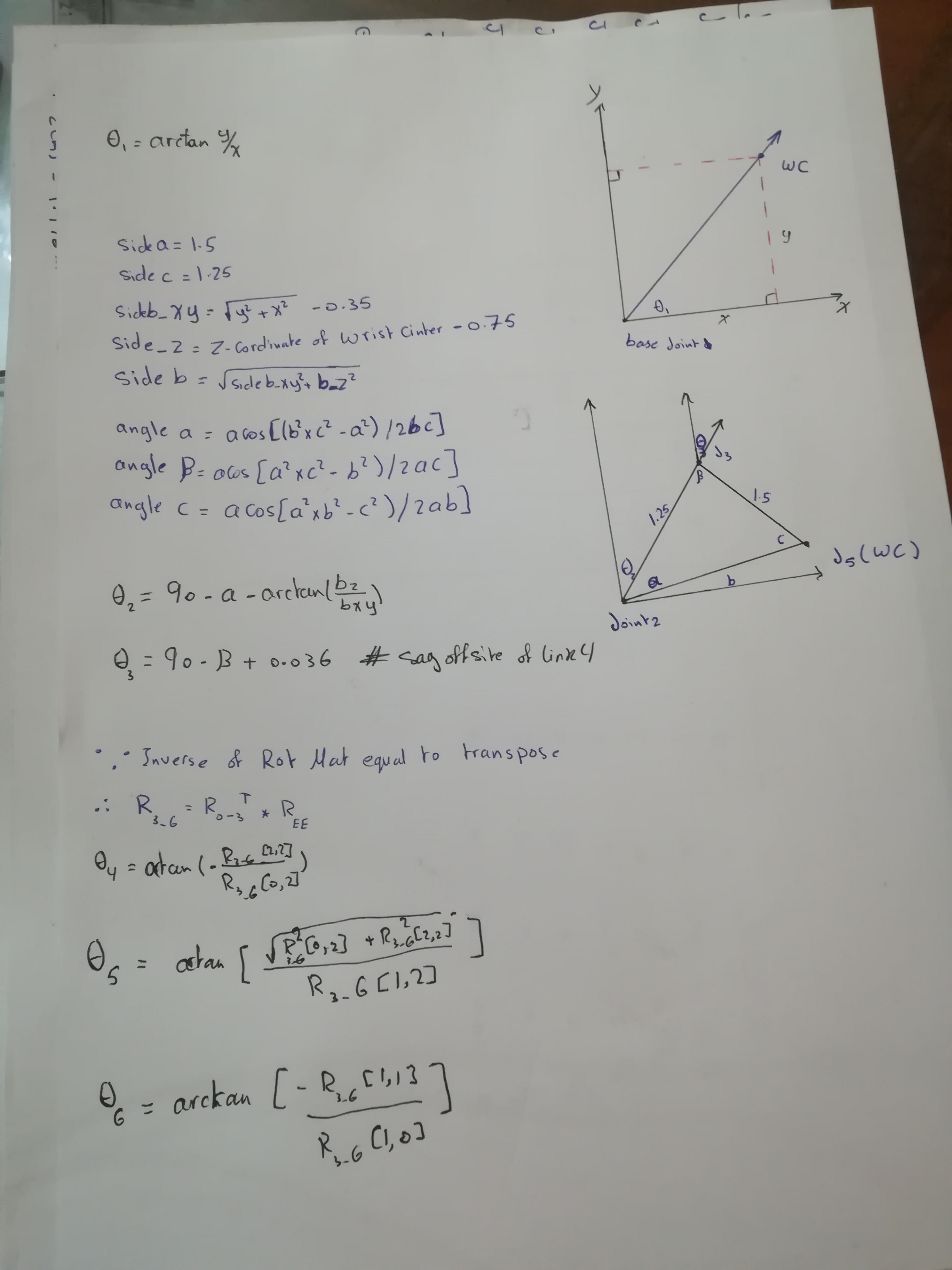
*ROT\_EE = ROT\_Error \* Rot\_Fixed*

Inverse kinematics Analysis :

First evaluate first three angels by identifing the 5th joint as the wrist center



Then evaluating the last three joint’s angles by using the individual DH transforms we can obtain the resultant transform and hence resultant rotation



***To setup the project***

Make sure you are using robo-nd VM or have Ubuntu+ROS installed locally.

### One time Gazebo setup step:

Check the version of gazebo installed on your system using a terminal:

$ gazebo --version

To run projects from this repository you need version 7.7.0+ If your gazebo version is not 7.7.0+, perform the update as follows:

$ sudo sh -c 'echo "deb http://packages.osrfoundation.org/gazebo/ubuntu-stable `lsb\_release -cs` main" > /etc/apt/sources.list.d/gazebo-stable.list'

$ wget http://packages.osrfoundation.org/gazebo.key -O - | sudo apt-key add -

$ sudo apt-get update

$ sudo apt-get install gazebo7

Once again check if the correct version was installed:

$ gazebo --version

### For the rest of this setup, catkin\_ws is the name of active ROS Workspace, if your workspace name is different, change the commands accordingly

If you do not have an active ROS workspace, you can create one by:

$ mkdir -p ~/catkin\_ws/src

$ cd ~/catkin\_ws/

$ catkin\_make

Now that you have a workspace, clone or download this repo into the **src** directory of your workspace:

$ cd ~/catkin\_ws/src

$ git clone https://github.com/udacity/RoboND-Kinematics-Project.git

Now from a terminal window:

$ cd ~/catkin\_ws

$ rosdep install --from-paths src --ignore-src --rosdistro=kinetic -y

$ cd ~/catkin\_ws/src/RoboND-Kinematics-Project/kuka\_arm/scripts

$ sudo chmod +x target\_spawn.py

$ sudo chmod +x IK\_server.py

$ sudo chmod +x safe\_spawner.sh

Build the project:

$ cd ~/catkin\_ws

$ catkin\_make

Add following to your .bashrc file

export GAZEBO\_MODEL\_PATH=~/catkin\_ws/src/RoboND-Kinematics-Project/kuka\_arm/models

source ~/catkin\_ws/devel/setup.bash

For demo mode make sure the **demo** flag is set to "true" in inverse\_kinematics.launch file under /RoboND-Kinematics-Project/kuka\_arm/launch

In addition, you can also control the spawn location of the target object in the shelf. To do this, modify the **spawn\_location**argument in target\_description.launch file under /RoboND-Kinematics-Project/kuka\_arm/launch. 0-9 are valid values for spawn\_location with 0 being random mode.

You can launch the project by

$ cd ~/catkin\_ws/src/RoboND-Kinematics-Project/kuka\_arm/scripts

$ ./safe\_spawner.sh

If you are running in demo mode, this is all you need. To run your own Inverse Kinematics code change the **demo** flag described above to "false" and run your code (once the project has successfully loaded) by:

$ cd ~/catkin\_ws/src/RoboND-Kinematics-Project/kuka\_arm/scripts

$ rosrun kuka\_arm IK\_server.py

Once Gazebo and rviz are up and running, make sure you see following in the gazebo world:

- Robot

- Shelf

- Blue cylindrical target in one of the shelves

- Dropbox right next to the robot

If any of these items are missing, report as an issue.

Once all these items are confirmed, open rviz window, hit Next button.

To view the complete demo keep hitting Next after previous action is completed successfully.

Since debugging is enabled, you should be able to see diagnostic output on various terminals that have popped up.

The demo ends when the robot arm reaches at the top of the drop location.

There is no loopback implemented yet, so you need to close all the terminal windows in order to restart.

In case the demo fails, close all three terminal windows and rerun the script.