Group Project Description

Control for Robotic, IIAV3120

Reactor-X 150 robotic arm manipulator

1. Introduction/Information

The group project for the course IIAV 3120 (Control for Robotics) is an obligatory task that all the students enrolled in this course should fulfill. It counts 40% of the course grading. This is a group project and each student should be a part of a group. The students have the responsibility to form the groups.

A technical report should be delivered by each group. The deadline for submitting the report and the date for the possible demonstration is given at the end of the course homepage (https://web01.usn.no/~roshans/cfr) (scroll to the bottom). Each group will receive a score (from 0 to 40) for the project work. Since there will be no presentation, the grade for the project work is solely based on the quality of the submitted report. So make sure that you submit a good report.

2. Process background

Robotic arm manipulators are widely used in industries for various applications. They are used in automotive, aerospace, electronic/electrical industries, shipping and trade etc. (just to name a few), for example, for performing repetitive tasks like those involved in an assembly line. USN has recently purchased several units of a ReactorX 150 robotic arm manipulators from Trossen Robotics. These robotic arms are planned to be used in teaching and research activities here at USN. The ReactorX 150 offers 5 degrees of freedom and a full 360 degree of rotation. At the heart of the ReactorX150 is the Robotis DYNAMIXEL X-Series smart servo motors and DYNAMIXEL U2D2 which enables easy access to Dynamixel software development kit. Figure 1 shows the Reactorx150 robotic arm manipulator.

Figure 1: ReactorX 150 robotic arm manipulator

3. Tasks

The tasks have been divided into three parts:

(i) Development of simulator with URDF

(ii) Testing inverse kinematic solution on DH model

(iii) Testing inverse kinematic solution on real ReactorX 150 robot arm

Note: This task 3(iii) is optional for students including the online students, but I highly recommend the campus and industry master students to fulfill this task.

3.1 Development of simulator with URDF

The simulator can be developed using your computers where MATLAB and SIMULINK have been installed. The following tasks should be completed:

(A) **Forward kinematics:**

(i) Process Description: Please describe the RX150 Reactor-X robotic arm manipulator. You have re-use the figures/pictures that is available in the lecture notes.

(ii) Download the URDF file of RX150 and the geometry files from here: https://web01.usn.no/~roshans/cfr/downloads/rx150-urdf-and-geometry-files.zip

(iii) Unzip the downloaded file from (ii) and use it to make a simulator for the virtual simulation of the RX150 robotic arm manipulator in Simulink/Simscape. Take note of these things when designing the virtual simulator:

a. In your simulator, you should be able to change the joint angles by making use of sliders or knobs
b. Manually change the knob or slider position to change the joint angles. See how the robot arm responds by looking at the 3D figure in Simscape explorer. Take some snapshots and put it in your report.

(B) **Inverse kinematics**

(i) Trajectory generation: Create a trajectory of your choice. However do not make your trajectory too simple (like just a straight line), but some level of complex trajectory generation is expected.

(ii) Trajectory control: Using Simulink and the inverse kinematic solver of Simulink, design a simulator such that your robot's end effector follows the trajectory. Take note of these things:

a. Make use of only the first 4 joints (starting from the base joint which gives 360 degree rotation up to the 4th joint), i.e. neglect the grippers and the 5th joint.
b. Both the robot arm and the trajectory should be visible at the same time in your plot.
c. Take some time-series snapshots of your plot (for example, 5 or 6 snapshots at different points of the trajectory that shows that your robot is indeed following the trajectory. You can show these snapshots as collections of subplots in a single figure).
d. Make a short video where you can capture the screen and show the movement of the robot arm along the trajectory. Upload the video to youtube (or similar platform) and put the link of the video in the report.

(iii) Store and save the solution of the inverse kinematic solvers as the time series data and also as arrays. You should store the values of the joint angles for each joint that was calculated by inverse kinematic solver. Remember that inverse kinematics means finding the joint angles if the end effector position is known.

3.2 Testing inverse kinematic solution of task 3.1 (B) on DH model

This task can also be performed using your computers where MATLAB and Simulink has been installed.

(i) Using pen and paper and following the Denavert-Hartenberg (DH) representation, make the kinematic model of the RX-150 robot arm. The dimensions (all in mm) of the robot arm is shown in Figure 2.
Your DH model should produce three algebraic equations for calculating the end-effector's x, y and z coordinate positions when the joint angles are known.

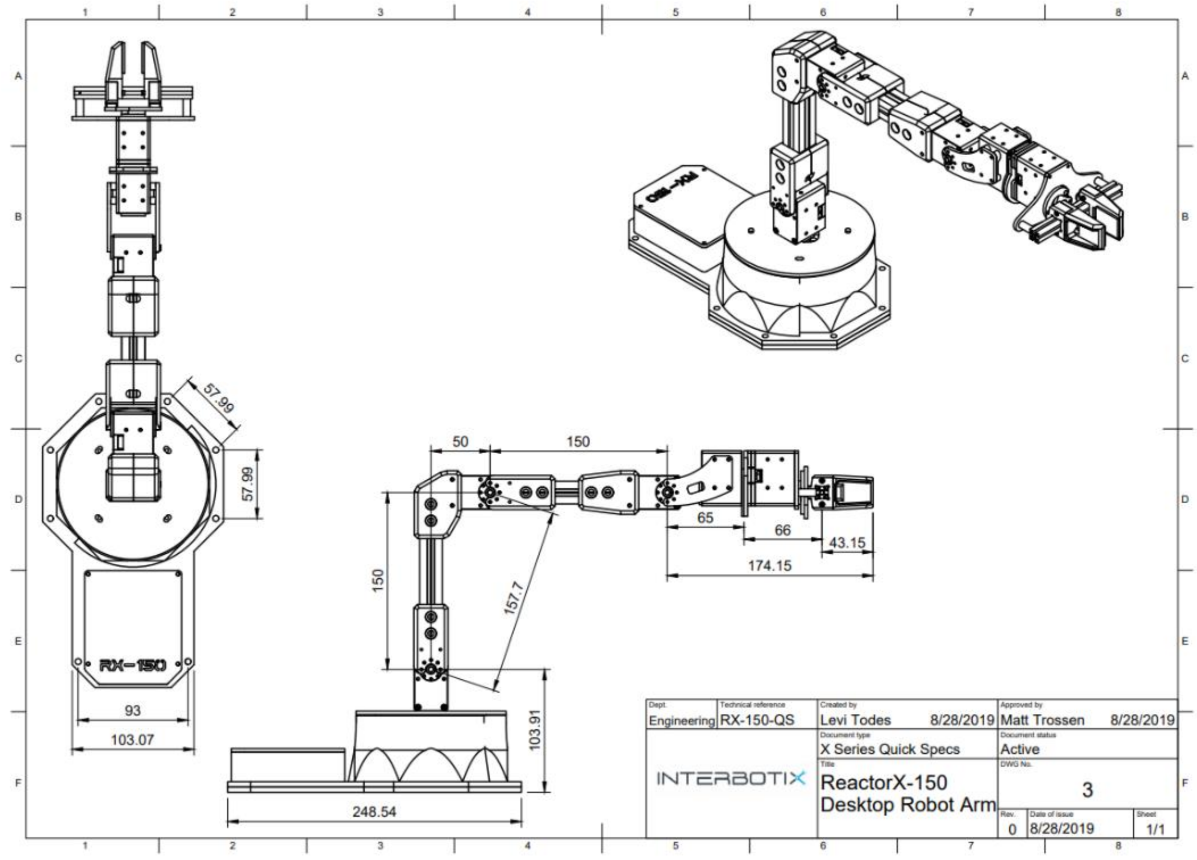

Figure 2: Dimensions of ReactorX 150 robot arm. All dimensions are in millimeter.

(ii) In MATLAB script, use your DH model equations from task 3.2 (i), and then calculate the end effector positon (x_e, y_e, z_e) for each of the stored values of the joint angles from task 3.1 (B) (iii). Make note of these points:

a. You can evaluate the DH model equations for each of the stored joint angles either using a 'for loop' or you can also do it vectorially. Choice is yours.
b. Store the calculated end-effector positions (x_e, y_e, z_e).
c. Plot the stored end-effector positions and the trajectory points (from task 3.1, B (i)) in the same 3d plot. Do they match? If they don't something is wrong with your DH model.

3.3 Testing inverse kinematic solution of task 3.1 (B) on real ReactorX 150 robot arm

As mentioned before, this task is optional for students including the online students. However, I strongly recommend the campus students and the industry master students to perform this task.

(i) Connect the real ReactorX 150 lab robot arm to your computer/Simulink. The class teacher will show you how to do this in the classroom.

(ii) Using one slider for each joint (the first 4 joints), test whether the real robot is responding properly or not.

(iii) Then you can replace the sliders for each joints with the values of the joint angles that you stored in task 3.1, B (iii). For this you will use the stored time series data. Make sure that you select the ‘stop time’ and ‘sampling time’ in Simulink solver configuration. And most importantly make sure that real time pacer is active.

(iv) Make a small video to show that the real robot is following the trajectory properly. Upload the video to youtube (or similar platform) and put a link to the video in your report.