ROBOTICS TOOLBOX

MAE 547

Final Project Report

Team

Shravan S Rai: 1217333728

Gaurav Sharma : 1217313305

Shredha K.P : 1218867507

Contributions –

Dh Parameters - Gaurav Sharma, Shredha K.P

Homogeneous Transformations- Shredha K.P

Robot Dynamics - Shravan S Rai

Robot Control - Shravan S Rai

GUI - Shredha K.P, Gaurav Sharma, Shravan S Rai

This is a GUI based Robotic Toolbox based on Peter Corke's Robotics Toolbox. Here users can define the robot using an easy to use GUI and generate its dynamics and control characteristics. To get started you need MATLAB 2019 with Robotics System Toolbox and Peter Corke's Robotic Toolbox version 9.10 installed. Here is a step by step guide to help users using this tool :

1. Main Window:

- GUI starts with the MainWindow App which presents the menu to the user.
- Menu provides three operations to the user : 1. Robot Description 2. Homogeneous Transform 3. Robot Dynamics 4.Robot Control
- Select Robot description to define robot parameters.

2. Robot Description:

- First Enter the number of links.
- Then select whether you know the DH parameters of the robot or not.
- If "YES", then enter the DH parameter for the link and select the type of joint : "Prismatic" or "Revolute".
- Update the DH parameters for the link by selecting the "Update the link" button.
- Repeat steps 3 and 4 for each link.
- If you don't know the DH parameters. You will need to enter 4 parameters, namely :
    - Distance along Z axis (d)
    - Distance along X axis (a)
    - Angle about X axis (alpha)
    - Angle about Z axis (theta)
- Repeat step 6 for each link and update the parameters for each link by selecting "Update the link" Button.

3. Homogeneous Transform:

a. First enter the number of rotations to be performed.

b. Then enter the rotation number.

c. Enter the axis and angle of rotation and press update.

d. Repeat step b-c for each rotation to be performed.

e. Enter the Vector A position

f. Press the Transform Vector button to obtain the transformation matrix and plot.

3. Robot Dynamics:

- Describe the robot as described above.
- Enter Link parameters such as centre of gravity(CG), moment of inertia(I), Jm and gravity for each link.
- Update the parameters for each link.
- After describing the robot properly, you can select whether you want to see dynamic equations or not by typing in “Y” or “N”; under the Robot definition tab.
- Users can also enter Torque (as a function of time), initial joint variables and joint rates as input in vector form. Ex: [t exp(t) sin(t)].
- Enter Run time describing the time duration you want to run the simulation for.
- Press “Run Simulation” to run the simulation for the dynamics.

4. Robot Control :

- The robot can be defined just like how it was defined in the Robot Dynamics section.
- Now enter the initial condition for joint variables and joint rates.
- Select the dimensions of operational space you want to work in, say 2D or 3D.
- Now enter the waypoints for the trajectory to be tracked by the robot.
- Select the type of motion control you would wish to perform: 1. PD Control with gravity compensation or 2. Inverse Dynamics Control.
- When you click on the button, it will redirect to simulink and the waveforms for desired position and actual position will be plotted on the scope.
- We are using the Robotics System Toolbox for the trajectory generation module to generate the trapezoidal velocity profile.

If the gui window for controls crashes,

1. Robot.m file contains pre defined 3 link planar arm, with all the compentents. Run Robot.m
2. For PD Control with gravity compensation, run the PD_Control_Gravity_Compensation.slx file. It will run with a defined Xd value. You can change the value by clicking on the box. It should be a 3x1 vector.
3. For Inverse Dynamics Control, run the inverse_dynamics_control.slx. It has a predefined trajectory using the trapezoidal velocity generation block. You can vary the waypoint values and see the manipulator converge to the given ponts.
4. The results for both 2 and 3 would open as soon as the simulation is started. The plot of actual position and the desired position can be seen through the scope output.

Difficulties faced:

1. The matlab app designer is not user friendly and requires a lot of knowledge about matlab data flow methods.
2. Peter Corke Library had few functions which behaved uncharacteristically as shown in Fig 1.
3. Peter Corke Serial link object does not permit plotting from the gui. So the robot is plotted separately from the gui.

References:

Robotics - Modelling, Planning and Control by Bruno Siciliano Lorenzo Sciavicco Luigi Villani Giuseppe Oriolo

Robotics, Vision and Control: Fundamental Algorithms in MATLAB by Peter Corke

Mathworks Forum.