Introduction to Robotics
Assignment
Due: May 2nd 2019

The assignment is to model and control an ABB IRB1600-8/1.45 robot. The kinematic data for the robot is available in the datasheet attached.

You will use the robotics toolbox for MATLAB for this assignment. Information on how to obtain and install the toolbox is provided below. An example file to demonstrate the use of the toolbox is also included. The final submission should include a report showing your work, as well as any supporting code.

The exercise contains the following parts:

(a) Set up DH-frames and obtain the DH parameters for the robot. Write a function to obtain the transformation matrix representing the end-effector frame.

(b) Model the robot in MATLAB using the Robotics toolbox. Verify the output of the function you wrote in (a) by comparing with the output of the function `fkine()` in the robotics toolbox.

(c) Given the transformation matrix representing the position and orientation of the end-effector, geometrically solve the inverse kinematics problem for the position. You will need to provide analytical expressions to compute the first three joint angles of the robot. How many solutions exist?

(d) Compare the results of (c) with the output of the `ikine()` function in the robotics toolbox. You will need to specify appropriate options to obtain multiple solutions.

(e) Write a function to generate a joint space trajectory to move the robot from a start position to to reach a desired goal position, subject to maximum joint acceleration and joint velocity constraints. The trajectory should pass through two specified goal positions. The robot must stop at the goal position.

(f) Design a independent joint PD controller to move the robot along the trajectory designed in (e).

(g) Using the robotics toolbox, implement the PD controller designed in step (f). You should provide a function that implements an inverse kinematics based controller to move the robot end-effector from a start position to a goal position through two waypoints. The waypoints as well as the start and goal positions should be provided using appropriate transformation matrices.

(h) Design and implement a PD trajectory controller using nonlinear decoupling to follow the same trajectory as in (g) above. Compare the performance to that of the independent joint controller.

The robotics toolbox can be obtained from http://petercorke.com/wordpress/toolboxes/robotics-toolbox.

Additional information:

- Inertia Matrices: Assume the robot to consist of hollow uniform rectangular beams made out of metal (steel, aluminum, iron etc.).
- Gear-ratios: Include gearboxes with following gear ratios [-100 100 100 -60 -60 40]:1 in your model.
- Motor inertia: Motor inertial can be assumed to be 50% of link inertia when transformed to the arm side, i.e, after the gear-box.
- Peak motor torques: Assume available motor torques to be [5 10 5 .6 .6 .5] Nm.