MECH 444/544
Introduction to Robotics

Instructor: Prof. Cagatay Basdogan

PROJECT 03

In this project, your goal is to develop a dynamical model of the PHANToM Premium 1.5 haptic robotic arm and design a control architecture in Simulink to move its end-effector point on a straight path while compensating for positional and velocity errors. The control architecture should utilize a PID controller such that the end-effector of the device will follow a trajectory in the form of a line segment, given arbitrary start and target points in 3D space.

1) Using the inertia and Coriolis matrices and gravity vector of the device (see "dynamical_properties.m" file), create a "user-defined" MATLAB function as a Simulink block which takes applied torque to each joint as input vector and returns the corresponding vector of joint angles. (Hint 1: You may prefer using several additional Simulink blocks, so that you will create a subsystem instead of a single function for the same purpose. Hint 2: You may also feed the vectors of joint angles and velocities back to your function by using a memory element).
2) Create a "user-defined" MATLAB function as a Simulink block to calculate forward kinematics which takes joint angles as input and returns Cartesian coordinates of the end-effector point as the output (use the transformation matrix given below).
3) Obtain the basic Jacobian matrix using the equations of the forward kinematics. Create a "user-defined" MATLAB function as a Simulink block to compute the basic Jacobian matrix for the given joint configuration.
4) Develop a trajectory planner in Matlab for the end-effector of the robot which generates a trajectory in the form of a line segment for the given arbitrary start and target points in 3D.
5) Develop a closed-loop control architecture for the device in Simulink, which utilizes a PID controller such that the end-effector of the device follows the pre-defined trajectory with minimum tracking error in the shortest time.
6) Use auto-tuning option of the PID block in Simulink to tune the controller gains (if necessary, play with "response time" and "transient behavior" options)
7) Show the animated model of the robotic arm while it follows the desired path (show the side and top views of the robot simultaneously as in Figure2, show the desired path in black color, show the incremental movements of the end-effector point in red color)

Project Details:

- Make sure that your controller does not apply more than 1 N of force to the end-effector at any time (use a force saturation block in Simulink).
- Prepare your Simulink models such that any 3D point (within the workspace of the device) can be specified as the start and target.
- In the simulations, the end-effector should move first 1) from current point to a given start point following a linear path, and then 2) from the start point to the target point following a linear path. The device should come to a full-stop at the end of the path.

At the end, please provide

- the plots representing the force outputs of the controller,
- the plots representing the desired and simulated trajectories (joint angles and end-effector position) as a function of time,
- the plots representing the tracking error as a function of time for the end-effector,
- the average tracking error, and the task completion time for the linear path.

Additional information regarding the device geometry and kinematics:

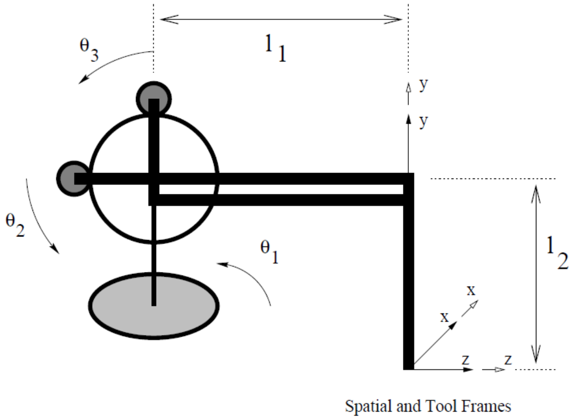

Figure 1: Zero configuration on the manipulator

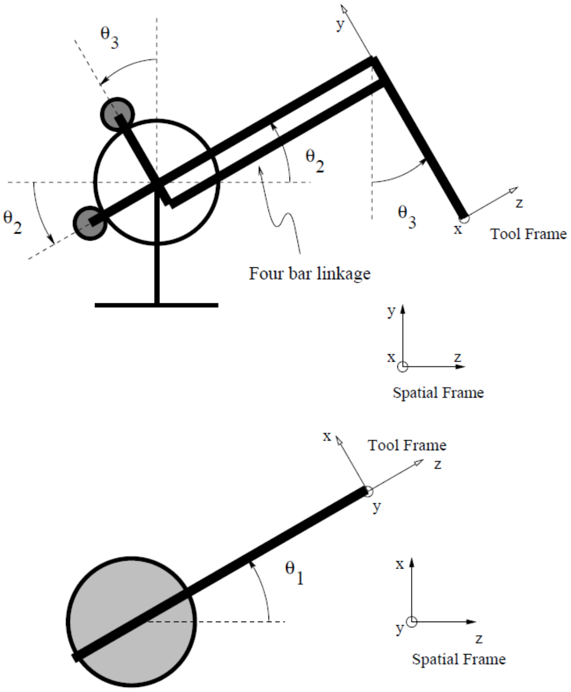

Figure 2: Side and top views

Transformation matrix of the end-effector with respect to origin:

$$
{}^{0}_{E}T=\begin{bmatrix} \cos(\theta_1) & -\sin(\theta_1)\sin(\theta_3) & \cos(\theta_3)\sin(\theta_1) & \sin(\theta_1)(l_1\cos(\theta_2)+l_2\sin(\theta_3)) \\ 0 & \cos(\theta_3) & \sin(\theta_3) & l_2-l_2\cos(\theta_3)+l_1\sin(\theta_2) \\ -\sin(\theta_1) & -\cos(\theta_1)\sin(\theta_3) & \cos(\theta_1)\cos(\theta_3) & -l_1+\cos(\theta_1)(l_1\cos(\theta_2)+l_2\sin(\theta_3)) \\ 0 & 0 & 0 & 1 \end{bmatrix}
$$