ECE4560 - Homework #12

Due: Nov. 27, 2017

Problem 1. (40 pts) Consider, as usual, the three-link rotational planar manipulator from the previous homeworks, c.f. Figure 1. We've worked out the forward and inverse kinematics for this little guy. Now, let's continue with the manipulator Jacobian(s).

(a) What is the body manipulator Jacobian as a function of $\alpha = (\alpha_1, \alpha_2, \alpha_3)^T$?

(b) What is the spatial manipulator Jacobian as a function of $\alpha = (\alpha_1, \alpha_2, \alpha_3)^T$?

(c) Let the link lengths be $l_1 = 1$, $l_2 = \frac{2}{3}$, and $l_3 = \frac{1}{2}$. If the joint angles are set to $\alpha = (\pi/3, -\pi/4, \pi/6)^T$ and the instantaneous joint velocities are $\dot{\alpha} = (-\pi/12, \pi/8, -\pi/12)^T$, what is the instantaneous body velocity of the end-effector?

(d) Using the setup from part (c), what is the instantaneous spatial velocity associated to the end-effector motion?

(e) Suppose that there was a tool in the end-effector and that the end-effector frame to tool frame transformation was $h = (1/4, 0, 0)$, as given in vector form. What is the effective instantaneous body velocity of the tool frame given the solution from part (c)?

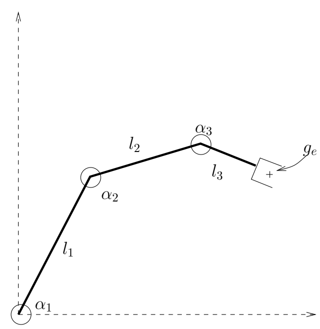

Figure 1: Planar 3R Manipulator.

Problem 2. (40 pts) Let's keep working with the three-link rotational planar manipulator, whose link lengths were specified to be $l_1 = 1$, $l_2 = \frac{1}{2}$, and $l_3 = \frac{1}{4}$. Rather than trajectories in the joint configuration space, this problem considers trajectory generation using the joint velocity space. Use the same setup from the previous problem as far as the initial conditions and the final desired end-effector configuration.

(a) Given the initial joint angle $\alpha(0)$ and the vector-form derived trajectory,

$$\left\{ \begin{array}{c} x_e^*(t) \\ y_e^*(t) \\ \theta_e^*(t) \end{array} \right\} = \left\{ \begin{array}{c} x_e^*(0) \\ y_e^*(0) \\ \theta_e^*(0) \end{array} \right\} + \frac{t}{T} \left(\left\{ \begin{array}{c} x_e^*(T) \\ y_e^*(T) \\ \theta_e^*(T) \end{array} \right\} - \left\{ \begin{array}{c} x_e^*(0) \\ y_e^*(0) \\ \theta_e^*(0) \end{array} \right\} \right),$$

compute the associated body velocity as a function of time, $\xi_e^b(t)$ then use resolved rate trajectory generation to obtain a trajectory in the joint space.

Plot the resulting joint angles and the end-effector configuration. as functions of time. How does it look compared to the splined version?

(b) Given the initial joint angle $\alpha(0)$ and the computed (constant) body twist ξ^b from the last homework, use resolved rate trajectory generation to flow along the twist ξ. (This twist was from taking the logarithm of the end-effector transformation from start to end configurations.)

Plot the resulting joint angles and the end-effector configuration. as functions of time. How does it look compared to the splined version?

Note: The following problem should be done by everyone. The link lengths to use should be: $l_0 = 4.35$, $l_1 = 4.725$, $l_2 = 4.725$, and $l_3 = 5.12$. Where l_0 is the vertical rise at the base, then the remaining are the distances to the next joint.

Problem 3. (35 pts) Let's continue with manipulator Jacobians, but this time consider the Lynxmotion manipulator. It is definitely recommended that one use a computational program to help out with the numerics. In general, past students using a mathematical program tend to have an easier job at solving these types of problems, especially those who write an appropriate function relating to the problem assignment as opposed to hardcoding a script tailored to the homework only.

For the Lynxmotion manipulator, with the straight up home position, answer the following:

(a) What is the body manipulator Jacobian in symbolic form?
When giving the solution in symbolic form without actually computing out everything, you will need to provide a definition for every variable left in symbolic form. The examples from the class notes should provide a good example of what I am looking for in symbolic form. In particular, I definitely want the $\mathfrak{se}(3)$ twist portions of $J_i^b(\alpha_i)$ terms to be defined numerically. The adjoints can be left symbolically.

(b) Given the joint configuration $\alpha = (-\pi/3, -\pi/6, \pi/4, -\pi/2, 0)^T$, and the instantaneous joint velocities $\dot{\alpha} = (\pi/10, -\pi/20, \pi/20, 0, -\pi/16)^T$, what is the instantaneous body velocity of the end-effector?
Here is where the symbolic form and the class definitions go together to aid the coding step. If done properly, you should be able to get the Matlab code to look quite similar to the actual mathematical write-up from the previous problem.

(c) What is the associated spatial velocity?

Problem 4: Manipulator. (30 pts) Continue with the Lynx6 manipulator adventures. In this case, work out the resolved rate for position adventure.

This lab problem will use the Lynx6 manipulator Jacobian (using your link lengths) to follow a trajectory. Basically, given $\alpha(t)$ and the body manipulator Jacobian, one can use resolved rate trajectory generation for follow the trajectory. The $\alpha(t)$ function will be the spline function from the last lab.

For this problem, attempt to generate a trajectory from the initial position $Pos(g_i) = d_i$ from the previous spline problem that moves with a constant body velocity ξ to the final location $Pos(g_f) = d_f$ in the same amount of time as given for the spline problem. Since the initial joint values α were known, the start condition is known. The trajectory should then flow according to the linear velocity v (in the world frame) for the allotted time. Of course, this velocity should be converted from the world frame to the body frame via the inverse transformation $g_e^{-1}(\alpha(t))$ as the manipulator moves. Turn in the time varying signal $\xi(t)$ that gets computed along the trajectory for the constant veloity path (in world frame) to follow.

Using the past resolved rate trajectory generation code for `piktul` as an example, add the necessary functions that will do so for the `lynx6`. Call the function `genTrajectoryPos` where it is assumed that some kind of $E(3)$ function of time and velocity function of time is given in a structure built with `traj.pvec`, `traj.pdot`, and `traj.tspan`. In reality, you will probably only need the velocity structure field, not the configuration structure field. The output of the system should be a set of joint angles versus time that will move the arm to follow the specified position trajectory. It can be in the output format of `ode45` and should be in a structure with the fields as per: `jtraj.alpha` and `jtraj.time`.

You will also need to write a function called `Jacobian` that computes the (linear and angular) body velocity of the end effector as a function of the α coordinates. For completeness, you should include all α coordinates in it (the last one will just have a zero column since it doesn't contribute to movement, it just opens/closes the gripper). For sure you will also reuse the `followJointTrajectory` function to make it happen. The script you wrote for the last homework can be recycled and modified to work for this problem. Note that since only the position should be followed, all that you will need is the top three rows of the Jacobian for the position part. Use the `pinv` Matlab function or the proper slash command (is it forward or backslash for overdetermined systems?).

Let's check to see if the manipulator is really following the trajectory. Plot the end-effector position as a function of time, as well as the straight line path. They should be close, if not equal. Plot both the position as a function of time, as well as the position as a parametric plot.

Tips: You can always use the `lynx6_display` function to test out the code. If you can get the resolved rate code working to generate the joint angles versus time, then using `lynx6_display` in a loop will display it as a movie in a Matlab figure. Remember to include the `drawnow` command at the end of the loop to force the figure to refresh.

Problem 4: Biped. (40 pts) Work out the next step in the biped project.

Problem 4: Turtlebot. (30 pts) Moving along in the Sensing Part 2 module, work out bullet 4. Demonstrate that you can follow the blob around the halls or room.

Problem 4: Custom Lab. (40 pts) If you are working on a group project, then your group contact will review the submissions associated to the prior tasks and provide a new set of tasks. You should work out whatever baby steps this group contact assigns the group as opposed to the lab problem.