Assignment 6:
Multi-Degree-of-Freedom Kinesthetic Devices

PDF file due on Canvas by 11:59 pm PDT on Thursday, May 21, 2020
Please write clearly or type your responses.
Submit some parts using the MATLAB Grader assignment on Canvas

Optional Reading

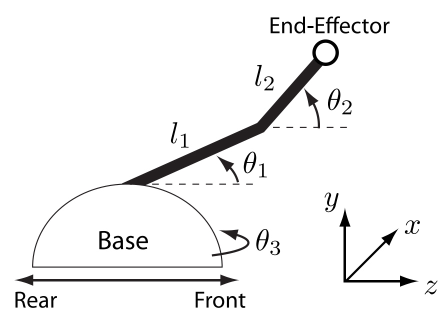
- G. Campion, Q. Wang, and V. Hayward. The Pantograph Mk-II: a haptic instrument. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 193-198, 2005. *This paper presents the design and kinematics of a 2-DoF kinesthetic haptic device in a pantograph configuration. Brandon has annotated the paper to correct a sign error in the published document.*

- D. Ruspini, K. Kolarov, and O. Khatib. Haptic Interaction in Virtual Environments. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 128-133, 1997. *This paper introduced the concept of the proxy and presents algorithms for rendering 3D virtual environments.*

These papers are on Canvas under Files > Papers.

1. Phantom Omni Kinematics

A 3-degree-of-freedom haptic interface, the Phantom Omni, is shown below. In this device, two dofs are created with a revolute-revolute serial arm, and the third dof simply rotates the serial arm. The diagram below (at right) shows a configuration of the device that is not achievable due to mechanical joint limits. However, this is a convenient choice of coordinates for computing the kinematics. The joint angles are absolute, i.e. measured with respect to ground. The end-effector is the point at the center of the wrist/gimbal where the stylus is attached to the main part of the device. Note that the wrist is not actuated, so we will not consider it nor the stylus in our analysis. The coordinate system shown is just to define the *x, y,* and *z* directions; the origin of the coordinate system actually placed at the center of rotation of link l_1., shown as a black circle in the left image below.



A. Develop a forward kinematic model of the Omni. In other words, compute the *x, y,* and *z* positions of the end effector as functions of θ_1, θ_2, and θ_3. Use the link lengths given in the figure above. The center of the world frame (0, 0, 0) is at the center of rotation of link l_1. Show your derivation of the kinematics in your submitted solution.

B. Determine the Jacobian of the Omni. Don't forget the chain rule when taking partial derivatives! Write out the matrix so that it can be used in an equation of the form $[\dot{x} \quad \dot{y} \quad \dot{z}]^T = J[\dot{q}_1 \quad \dot{q}_2 \quad \dot{q}_3]^T$. Show your derivation of the Jacobian in your submitted solution.

C. Enter the forward kinematics and Jacobian (Parts A and B) into the accompanying MATLAB Grader assignment, such that you can change the angles and observe the change in output. A reality check for seeing if the forward kinematics are being calculated correctly is to have your MATLAB script construct a simple stick drawing of the device.

Assuming the device currently has joint angles $\theta = [45 \quad -45 \quad 0]^T$ degrees, which is near the center of its workspace, use your forward kinematics and Jacobian implementations to:

i. Compute the resultant end effector position in meters.
ii. Compute the Jacobian for this specific configuration.
iii. Compute the vector of Cartesian endpoint velocities in meters per second given the joint velocities, $\dot{\theta} = [180 \quad 90 \quad 0]^T$ degrees per second.

Note that the numbers given above are in degrees, not radians. In addition to passing the pretests in MATLAB Grader, report your computed values for C.i. and C.iii in your written submission. Feel free to work in MATLAB and enter your solution code to pass the pretests in MATLAB Grader.

D. Suppose that you want the robot end-effector to push on the user with a Cartesian force vector of $f = [4 \quad 1 \quad 3]^T$ N. Your device is at the same position as stated above. What is the vector of joint torques needed to create this force at the end-effector? To complete this problem, use your computed Jacobian and perform the calculation to complete the pretest in MATLAB GRADER. In addition, include your computed values in your written submission.

E. Are there any singularities of this device? You don't have to calculate them from the Jacobian (unless you prefer to); you can just use your intuition about the geometry of the device. Explain in your written submission.

2. 3-DOF Rendering Examples

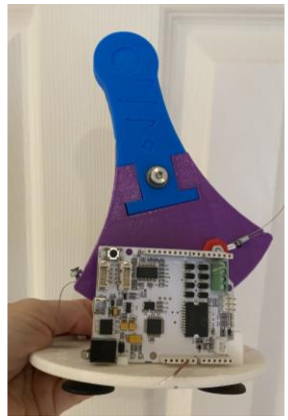
In this problem, you will examine haptic rendering algorithms for two simple virtual environments, which were discussed during the interactive session on Thursday, May 14. For each part, submit your response via Canvas in the same file as for Problem 1. You do not need to do anything in MATLAB for this question.

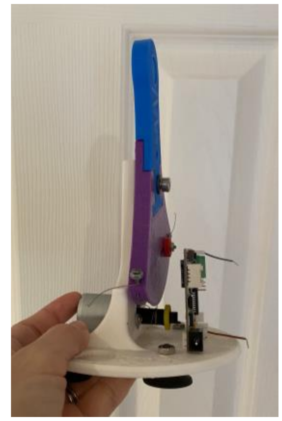
A. Write an algorithm (pseudocode is fine) for rendering being inside a virtual sphere. The center of the sphere is at (x_c, y_c, z_c). The radius of the sphere is R. The position of the user is (x, y, z).

B. Write an algorithm (pseudocode is fine), for rendering a box where the interior of the box is solid, i.e. the user's cursor is *outside* the box. The center of the box is at (x_b, y_b, z_b). The square box has side lengths, L. The position of the user is (x, y, z).

3. Hapkit Assembly

This portion of the assignment is due at the end of the day on Friday, May 22. As you will see below, you will submit your images to a google form, not Canvas. (There is nothing to submit on Canvas for this problem.)





Assemble your Hapkit following the interactive session instructions from May 19 and 21. Ensure that your Hapkit handle moves side to side throughout its workspace with relatively low friction. Take two pictures of your fully assembled Hapkit, one from the front and one from the side, similar to the two example images shown at right. Make your images PNG files, and name them LastnameFirstname-front.PNG and LastnameFirstname-side.PNG. Submit these via the following google form: https://bit.ly/2T2RSkG . You will need to be logged in to your Stanford email address to access the form. Once you have completed this step, you can move on to the testing and rendering activities in Assignment 7.