

Assignment 5: Kinesthetic Device Design and Control

PDF file due on Canvas by 11:59 pm PDT on Thursday, May 14, 2020

Please write clearly or type your responses.

Submit some parts using the MATLAB Grader assignment on Canvas.

In this assignment, you are going to develop a MATLAB simulation of a 1-DoF kinesthetic haptic system. This will be based on the model shown in Lecture 9 and that was discussed during the interactive session on May 5th. You will then run several simulations to investigate the dynamic properties of this system.

1. Simulation of a 1-DoF Kinesthetic Haptic System

- A. Access the MATLAB Grader Assignment and look at the starter code provided. There are function templates that you are expected to complete, with pretests available so you can check your implementation. To pass the pretests, your functions must produce the output specified. Unless explicitly stated, the inputs and outputs are scalar quantities. To run function pretests individually, you can comment out the untested function calls in the “Function Pretests” code block and their corresponding functions under the “Complete all the functions here” code block. The latter can be achieved by using `% {` and `% }` to execute block commenting.

Constants: The simulation requires the setting of some physical constants, some of which you will define in the main script and some that are to be defined inside the `simulate` function. For the values of the device and human parameters, use the following (which approximate the values for your Hapkit):

$$\begin{array}{lll}
 m = 0.03 \text{ kg} & k_{\text{human}} = 500 \text{ N/m} & k_{\text{wall}} = 500 \text{ N/m} \\
 b = 1 \text{ Ns/m} & b_{\text{human}} = 10 \text{ Ns/m} & x_{\text{wall}} = 0.025 \text{ m (+}x\text{ is deeper into wall)}
 \end{array}$$

You will also need to define the human input x_d as an oscillating desired position centered at -0.01 m with amplitude 0.04 m (that is, at the peak the user will be inside the wall) and a frequency of 0.4 Hz.

In addition, define the start and stop times (in seconds) for your dynamic simulation, as well as the time increment of the calculation. Time should start at $t_{\text{start}} = 0$ s and end at a time t_{end} that allows you to visualize several oscillations of the human input. The time increment, T , should be 0.1 ms.

State Tracking: Inside `simulate`, you will want to set up some empty vectors to track important system state variables, used later in the code for plotting. (The vectors’ length is determined by the number of data points you collect, i.e. the total time of the simulation divided by the time increment.) The variables you should record are x_d , v_d , x_{handle} , v_{handle} , a_{handle} , f_{felt} , and f_{actuator} .

Dynamic Simulation: Simulate the dynamics by stepping through time “manually” (i.e., in an explicit loop, which increments the time by T each time through the loop). Your simulation should include the following computations for each time step:

- Numerical integration of the handle variables. (`euler_integrate`)
- Compute the force applied by the virtual environment system at this time step. (`compute_interaction`)

- Compute the force between the human hand and the handle (the force in the spring and damper that connect the user's desired position to the mass of the handle). (`compute_human_force`)
- Compute the force perceived by the human. (`compute_perceived_force`)
- Compute the force due to the device's linear damping. (`compute_device_friction`)
- Compute the sum of all the forces on the handle. (`compute_resultant_force`)
- Compute the handle's acceleration for the next iteration. (`compute_acceleration`)

Plotting: Create a figure with 2x1 subplots showing the variables listed below. In both plots, include axis labels and legends. Adjust the time length of your simulation such that several oscillations of the user's input are clearly visible.

- Position signals of the user's desired position x_d and the handle position x_{handle} , as well as a dashed line representing the virtual wall location.
- The force applied by the device's actuator f_{actuator} and the force felt by the user f_{felt} .
- Comment on these plots. For instance, what causes differences between force applied by the device and felt by the user?

2. Performance of a 1-DoF Kinesthetic Haptic System

Run the simulation under different conditions in order to create plots of the following situations:

- The user is moving in free space. I.e., change the default conditions so that the user is never in contact with the virtual wall. What force does the user feel? At what points in time are the forces largest?
- The user is moving in constant contact with the virtual wall. I.e., change the default user input so that the user's position is always inside the virtual wall. How well does the handle position track the desired position? How does the magnitude of the wall force compare to those from device dynamics?
- The user is moving in and out of a soft virtual wall: make k_{wall} small so you see interesting behavior. How well does the handle position track the desired position? How does the magnitude of the wall force compare to those from device dynamics?
- The user is moving in and out of a very stiff virtual wall: make k_{wall} large enough so that you see some interesting behavior. How well does the handle position track the desired position? At what stiffness does the system show signs of instability? Using your result from 2C, what changes could stabilize it?

Attach all plots to your submission for the written responses. Submit the MATLAB portion via MATLAB Grader on Canvas.