

250a Haptics Lab

From CCRMA Wiki

The goal of this lab is to introduce three-dimensional active force feedback. Each haptic effect can easily be coupled to a sound because the feedback loop is controlled directly Pd. Different haptic effects suggest different sounds; different sounds suggest different effects. What are the most interesting mappings? Which are appropriate, controllable, even "expressive". Try each sound control with and without haptic feedback enabled. Are there some that simply cannot be played without haptics?



Contents

- 1 Part 0: Setup
- 2 Part 1: Test setup and run spring
- 3 Part 2: Wall
- 4 Part 3: Damping (Optional)
- 5 Part 4: Haptic Landscapes (Optional)
- 6 Part 5: Virtual Mass On Spring
- 7 Part 6: Virtual Bowed String
- 8 Part 7: Build Your Own

Part 0: Setup

- Nevermind about running `bash` and changing `.bashrc`. With this year's version of the driver, that is no longer necessary.
- Install the new and improved `.pdsettings` file on your CCRMA linux account according to the directions on this page (<http://ccrma.stanford.edu/courses/250a/lectures/pd/>) .

- You may wish to verify that no one has changed the wiring on your machine in the MAXLAB.
 - The Falcon should be connected over USB to the machine by way of a USB hub.
 - A -30V DC power supply should be plugged into the Falcon. The power supply has a yellow band on the end of the plug.
- If you need to make a 250a directory, type

```
~> mkdir ~/250a
```

- Download the patches Falcon-pd-linux.zip (<http://ccrma.stanford.edu/courses/250a/labs/Falcon-pd-linux.zip>)
- If firefox put them on your desktop, type

```
~> mv ~/Desktop/Falcon-pd-linux.zip ~/250a
```

- Unzip the files by typing

```
~> cd ~/250a unzip Falcon-pd-linux.zip
```

Part 1: Test setup and run spring

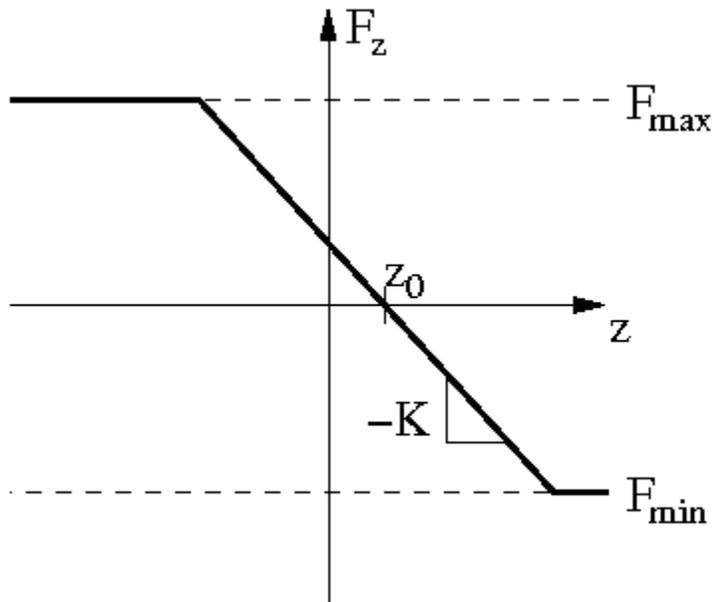
- Run 3D-spring.pd

```
~>cd ~/250a/Falcon-pd-linux/spring ~>pd 3D-spring.pd&
```

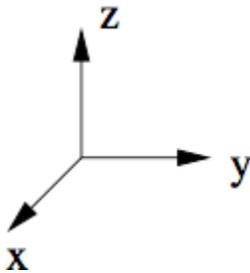
- Test the spring governed by $F_z = f_2(z_0 - f_1)$. F_z is the force exerted by the motors in the z-axis, f_1 is the position of the Falcon grip in the z-axis, f_2 is the spring stiffness, and z_0 is the rest position of the spring. In this example, we have $z_0 = 0$.

1. Check the messages in the Pd window to ensure that the Falcon has been initialized properly.
2. **Home the Falcon by pushing the Falcon grip all the way forward toward the NOVINT sign and then pulling it all the way back in a line extending out perpendicularly from the NOVINT sign.**
3. Hold down the center button of the grip to enable haptics.
4. Increase the stiffness of the spring from zero by dragging the number box.
5. Make sure that you understand how the "expr" object works.

In practice, the motors exert forces that are limited in magnitude, i.e. the Falcon limits the maximum force in both extremes, so the actual force as a function of position is slightly different from the idealized spring:



6. Try restricting the Falcon grip's motion to lie in a plane, a line, etc.
7. What is the rest position for the spring in the X direction?
8. Why is this rest position not equal to zero?



Part 2: Wall

A wall is just like a spring that engages right as the user starts pushing into the virtual wall. The spring's rest position is the position of the virtual wall.

1. Close the previous patch and load `1D-wall.pd`. (**Don't try to open two patches at the same time that include the Falcon object! If pd crashes, you can kill all instances of it by typing `"killall pd"` in the terminal.**)
2. Try changing the stiffness of the virtual wall.
3. How stiff is a real wall?
4. Notice that the wall is implemented using a subpatch instead of `expr`. Look inside the subpatch to see how it is implemented.
5. Rigid virtual objects can be represented using significantly stiff walls with complex geometries. This is how haptics can be used to help CAD designers study the shape of complex objects.

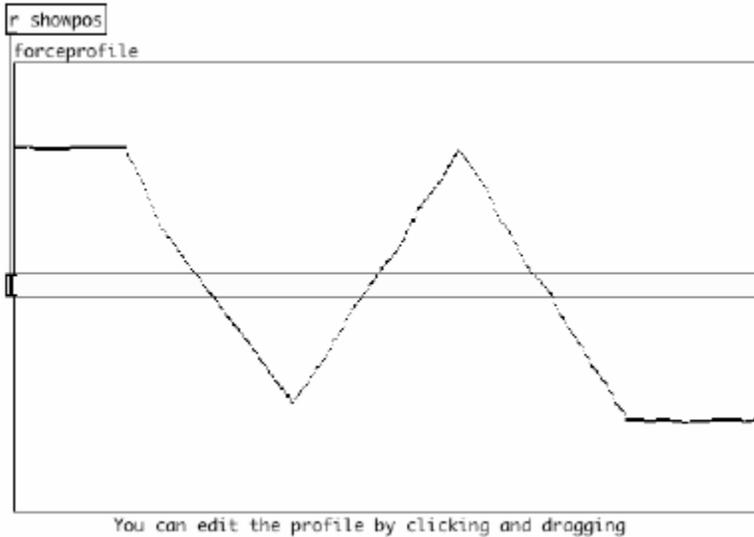
Part 3: Damping (Optional)

To implement damping, we need to control the Falcon according to $F = -Rv$, where v is the velocity of the Falcon grip, and R adjusts the amount of damping. In other words, when the falcon grip is moving in a particular direction, the motors exert a force to try to make the grip stop moving. Since the Falcon does not measure velocity, the Pd patch obtains an estimate of the velocity by differentiating the position measurement.

1. Close the current patch and open `3D-damper.pd`.
2. Adjust the amount of damping by increasing the slider. You should feel like you are moving your hand through a bowl of thin molasses or jelly.
3. Try enabling the "damping strobe" which turns the damping on and off over time.
4. Turn off the damping strobe, open the properties dialog for the number box feeding the "s frictcoef" object, and edit the lower limit so that it is -1.
5. **Now be very careful because the following step can cause the haptic device to move very quickly. Try to avoid banging the grips against the limits of the haptic workspace.** Adjust the number box so that it reads -1 so that the damping is negative. Now you will find that the Falcon grip moves about in a strange manner, which doesn't occur in nature. If you have the grip moving in any direction, then the motors push the Falcon grip in that same direction.

Part 4: Haptic Landscapes (Optional)

The patch restricts the motion of the Falcon grip to the Y-axis using springs in the X-axis and Z-axis. The force in the Y-axis is a function of the Y-position. The function is specified by a user-editable graphical array. At the center of the array, a horizontal slider (hslider) shows the Y-position of the Falcon. The example below specifies two equilibrium points, toward which the Falcon grip is pushed by the motors.



1. Close the current patch and open `force-profile.pd`.
2. On a separate sheet of paper, copy the graphical array and draw X's over the equilibrium points.
3. On the reverse of the separate sheet of paper, draw a force profile resulting in three equilibrium points, test it in Pd, and draw X's over the equilibrium points.
4. Experiment with some other force profiles.

Part 5: Virtual Mass On Spring

Now the Falcon grip is connected to a virtual mass by way of a spring.

1. Close the current patch and open `bounce-perc.pd`.
2. Before enabling haptics, manipulate the motion of the virtual mass and listen to the synthesized sound.
3. Now enable haptics by holding down the center button on the end of the Falcon grip.
4. Reduce the damping to zero. What is the difference between enabling haptics and not having it enabled?
5. Is it easier to control the motion of the virtual mass with haptics enabled?

Part 6: Virtual Bowed String

Actually you are enabled to bow a horizontal, virtual surface.

1. Close the current patch and open `bowed-string.pd`.
2. Try bowing the virtual surface.

3. Note that you can stop the "string" from vibrating if you hold the bow gently against the surface.
4. Is it easier to bow if it exerts a damping force on you? You can check by adjusting the BOW-STRING DAMPING slider.
5. Every time you press a key, the note played by the model changes.

Part 7: Build Your Own

Make your very own musical controller using the Falcon and one of the above patches as a starting point. Here are some ideas, but feel free to follow your own interests:

1. Try to impart kinetic energy from your body into a physical model of an instrument.
2. Make a record scratching interface allowing you to feel the record.
4. Can you make a sphere that exerts a spring force on the musician if the musician ever tries to press into it? What sort of music would you put to that?
5. Can you program the haptic interface so that it allows you to feel a texture?
6. Make the virtual mass in part 5 increase if the "number of objects" parameter is increased.

Make us proud. (And be ready show off your controller at the next lab session!)

Retrieved from "https://ccrma.stanford.edu/wiki/250a_Haptics_Lab"

- This page was last modified on 22 October 2009, at 00:25.