Overview:

Implemented a FSM whose state is controlled over vendor specific usb. Slightly modified vendor requests from hellousb project. Control torque (current), control location (position) for the different states. Uses simplified PID loop concept- only simplified I term. Sufficient given operational frequency vs. code speed. could help reduce problems from noisy current, but that should be fixed in hardware. USB allows you to set state and value. Keeps track of position with magnetoresistive sensor as implemented in last lab. Command line interface with user input debugging and help.

Code snippit: FSM switch statement (screen grab). See virtualenvironments.c for full code

Trace capture of bad waveform for current reading in a specific duty cycle/ frequency range.

Link to video of command line interface and operation.

Github link. Look specifically in hapticPaddle and MP2\_report. Main code is in virtualenvironments.c (for microcontroller side of things) and setEnvironment.py (for command line USB interface)

For graphs, have the microcontroller print over serial (at a much faster rate- say 100Hz? May need to improve band width/ increase baud rate.) have it print position (in degrees) as tracked by the microcontroller, force (current) . Plot position vs. time for the spring graph. Plot current vs. diff(position) for the damper. Plot current vs. position for the texture. Plot current vs. position for the wall?