*Notes taken posthumously (started taking current notes on 21/10/24)*

First step was to get IRCAM’s Rave object working in Pure Data (to potentially load directly onto the Bela). Unfortunately, the Pd implementation seems to be very poorly optimized, and trying to run just one model immediately maxes my laptop’s CPU to 100% and doesn’t run well. Had a more experienced stranger in the Pure Data discord try it himself, and he came to the same conclusion.

Successfully got the MaxMSP implementation of Rave running. Currently with two models simultaneously, at about 60-65% CPU load. It was at about 45-50% CPU load running only one, so theoretically each additional Rave model will add about 10-15% load, but need to test this further. Then investigated using RNBO~ to export the Max patch to C++ and load it onto the Bela that way.

Successfully exported a basic Max/RNBO patch to C++ and got it loaded and running on a Bela, but RNBO cannot make use of any Max externals (such as the Rave object) so in the end it is not a viable solution for my project.

Downgraded from Bela to Teensy for ease of development since I will no longer need Bela’s level of processing power on the microcontroller (all the signal processing/Rave implements will be handled on the laptop, and the posture reading/machine learning will be handled on the Teensy).

*21/10/24*

A person using a computer to connect wires to a circuit board

Description automatically generatedSuccessfully have a simple neural network reading two pots on a Teensy and mapping to the mix levels of two “postures”. It is trained to do all of either, or a 50/50 mix depending on the pot positions. Surprisingly accurate considering it is only using 2 inputs, although it is not very relevant to actual postures this way.

Next step trying to send some of this data to the Max patch, have found very little help online for how to do this so far, most forum posts seem to be assuming a lot of preexisting knowledge of how to do this.

A screenshot of a computer

Description automatically generatedEventually got it working thanks to an article on maker.pro. Every place I looked was using these like specific port numbers and other things when I actually only needed the letter, in this case “I”, associated with the teensy serial port.

Unable to send anything useful from Max back to the Teensy though (for example if I want to do the ML commands from the Max patch). It claims serial doesn’t understand strings as input, and sending the letters as bytes and converting in the Teensy code also doesn’t work.

*22/10/24*

A screenshot of a computer

Description automatically generatedCan now send serial messages from Max back to the Teensy. Can do all the training messages as well from within the Max patch. To get it connected right, for whatever reason you must first open and close the Arduino IDE serial monitor, then after closing it open the Max patch.

*An indeterminate number of days later, I remember to update this document....*

*27/11/2024*

I discovered that the reason the Max patch is so unbearably laggy is that it just doesn’t work well with Windows audio drivers. Used exact patch on a Macbook with flawless results. For presenting/demoing the project will likely need to borrow someone else’s computer. Mine is enough to develop it, barely.

Have the “suit” all set up. Sewed on sleeves for the sensors on the elbows and shoulders, put in the bend sensors, and sewed them into place. Connected them all to a breadboard for now for testing. Sensor values seem a bit inconsistent, but still need to figure out how to save the training to the SD card before I can judge on more in-depth testing.

A person using a wire to connect wires

Description automatically generatedA person sitting at a desk with a computer

Description automatically generated

Have some latent space controllers linked up to nn~ objects in the max patch. I'm thinking about using a Wii Nunchuck controller with a Grove Nunchuck sensor to control them. Gives easy access to a lot of parameters (buttons, joystick, motion controls).

A screenshot of a computer

Description automatically generatedA white remote control next to wires

Description automatically generated

*2/12/24*

The Wii Nunchuck requires/outputs a 5v signal, so it cannot be run directly into the Teensy. I ordered a 3.3v-5v level shifter to run it through, so progress on integrating that stops for today.

Suit actually feels very accurate, with only a couple seconds of training it VERY reliably mapped between the violin and sax positions, to the point that a blend will need additional training points. Still need to investigate including and try training a 3rd simultaneous instrument.

Figured out how to save the models onto the Teensy’s built in SD card, so can now save training sessions for later. Will be good for testing if one training set can work for multiple people and wearing sessions since the sensors will likely show slightly different data due to the relatively loose attachments to the shirt. Commands can be run from the max patch.

Sensors wiggle a bunch and sometimes even flip over, completely reversing their output values. Need a better way to hold them in place in the long term. The mapping worked well for 3 separate instruments though.

*4/12/24*

I am now successfully reading the nunchuck in the Teensy. The issue wasn’t actually the voltage, the Teensy’s 3.3v power seems to be enough to power it, the issue was that the Grove nunchuck board was plugged in upside down… Someone really needs to tell them to add some documentation for the products they make.

*8/12/24*

Set up a second neural network to run simultaneously on the Teensy. The first one will be for training instrument postures and will take the original four bend sensors (elbows and shoulders) as well as the gyroscopic X/Y data from the nunchucks. The second network will take the buttons and the joysticks of the nunchucks to train for altering the latent space parameters. Currently implemented and operational in the Arduino code and Max patch.

Will need many in-between training states for the latent space training, so that you can reliably bring the parameters back to 0.