

SIXT33N Project Report

Circuit

Final Design

This multistage project employs centroid clustering, PCA, and closed loop control to navigate a battery-powered vehicle with voice commands. The project has a few major components. The front-end-circuit (Fig. 1) is composed of a mic-board, an op-amp buffer, a DC offset input, and a variable gain amplifier. As usual, the buffer is included to prevent the loading of the mic board due to the DC offset and variable gain amplifier. Next, we include a DC offset in order to utilize the entire voltage range - the Launchpad does not support negative voltages, so we need to shift to the center of the voltage range. Finally, our OS2 pin is a level-shifter. Since we are introducing a DC offset with OS1, and our mic board has an amplifier, we need to introduce a level shift to ensure that the amplified output is still centered after the OS1 offset.

Gain and Frequency Response

Notice that both OS1 and OS2 (Figs. 2 & 3) are voltage dividers with buffer op amps to prevent the loading of the larger front-end circuit when they're connected. OS1 is a simple 1.65V DC input, whereas OS2 is tunable (notice the potentiometer). We also have a bandpass filter coming from the output of our mic board. We desired cutoff frequencies at approximately 250Hz and 2500Hz for our low-pass and high-pass circuits, respectively. Since we were limited to the resistors and capacitors available in the lab, these frequencies ended up being 320Hz and 3200Hz, respectively. Low-pass: [$R = 500\Omega$, $C = 0.1\mu F$], High-pass: [$R = 5k\Omega$, $C = 0.1\mu F$]. With the typical frequency response given by $f_C = (2\pi RC)^{-1}$, we found that these values are a good range for capturing the human voice.

PCA Classification

Commands

The commands we used for this project are as follows: **Drive Straight: "FORWARD"**, **Drive Straight Slow: "BREAK"**, **Turn Left: "WATERMELON"**, and **Turn Right: "ROMEO"**. Initially, we tried some command words with different pronunciations such as, "STRAIGHT," "SLOW," and "LEFT," but we realized that they were too similar after the principle component analysis since the clusters are very close to one another - making it difficult to distinguish commands while maintaining a low false-positive rate. After some thinking (and a hint from a lab assistant), we decided to move away from this pronunciation-differentiation model to a set of words with different hard consonants and different numbers of syllables. After looking at the projection of our training data onto our first two principle components, we were satisfied with the separation generated by this final set of words.

Processing

In order to make the PCA Classification more robust, we separate different commands by using words with different numbers of syllables. In this way, the classifier can classify different commands easier with less

principle components, which reduces the processing time during classification. This also decreases memory usage within SIXT33N, making the project more stable. We also decreased the threshold for the voice volume to 300 from the default value of 700 for the classifier to reduce the chance of missing any command. Although, we did have to increase this value from 250 as we found that very occasionally, other groups in the laboratory were able to trigger a command on our SIXT33N.

Controls

Open Loop Model

In an Open Loop Model, the output of the current state has no influence on the input of the next state. We assume the SIXT33N model is perfect without any model mismatch under the Open Loop Model. Without mismatch, δ , defined: $\delta[k] = d_l[k] - d_r[k]$ is going to approach a constant as time goes on, and the project would work perfectly if this is true. However, model mismatch nearly impossible to avoid, so we use a Closed Loop Model.

Closed Loop Model

Under the Closed Loop Model, we feed back what we got from the last step as a reference to the current state. Therefore, if there is any model mismatch, we can correct it by feeding in corrected data into the SIXT33N. With the manually correcting for each step, the SIXT33N is able to go straight after initial correction. See the block diagram of our closed-loop control scheme in (Fig 4). In this diagram, we have $d_L[k + 1] = d_L[k] + \theta_L u_L[k] - \beta_L$ (shown for left wheel, same form on right).

Turning: In order to allow for turning with reference tracking, we needed to allow one wheel to turn faster than the other, but with a specified difference in rate. With knowledge of the separation between the wheels, this lets us turn the car at a specific radius. We used the formula $\delta[k] = \frac{v^* l k}{r}$ where r is the turn radius, l is the wheel separation, and v^* is the desired velocity. In the final implementation, we had to modify $v^* \rightarrow \frac{T_e v^*}{T_d}$ to account for the period difference between the control and data collection loops. Nevertheless, this $\delta[k]$ term is incorporated into our closed loop model and allows us to turn in a specified direction depending on whether it is positive or negative.

Choosing Controller Values

The process of how we choose our k value was a bit long. First we tried to use the graph plotted by iPython to choose a k value for the project in order for it to drive straight without turning or oscillating, but this only gave us a specific range of k values for us to try them out during the next stage. After realizing that graphing is not enough for picking the k value, we started trying all the k values we got from the range narrowed by plotting the graph. If the car still turns, we increased the k value since the feedback is not enough for correcting the driving direction. If the car is was oscillating, we decreased the k value since the feedback corrected too much. After this iterative process, we picked our final k-value to be 0.4 for the car to drive straight.

General Comments

Gabriel Chen: I haven't had a physics lab for a while and this project really helped me beat the fear of doing physics. It was very frustrating at some point such as correcting the driving direction of SIXT33N, and we ended up taping the back wheel to make the closed-loop feedback to the project work fine. This is a really fun project and I really enjoy everything we encountered, either that was frustrating or straightforward.

Thomas McClave: After having taken Physics 111A last semester (introduction to circuit design), this project was a very interesting way to combine some simple algorithms (K-means, control, etc.) with analog circuitry. Although frustrating at times, this entire project experience has proven to be very rewarding once we got everything working. It's really cool to see everything working together and I feel like more difficult projects are certainly within reach!

Video

N/A - We performed a live project demo before a GSI.

Figures

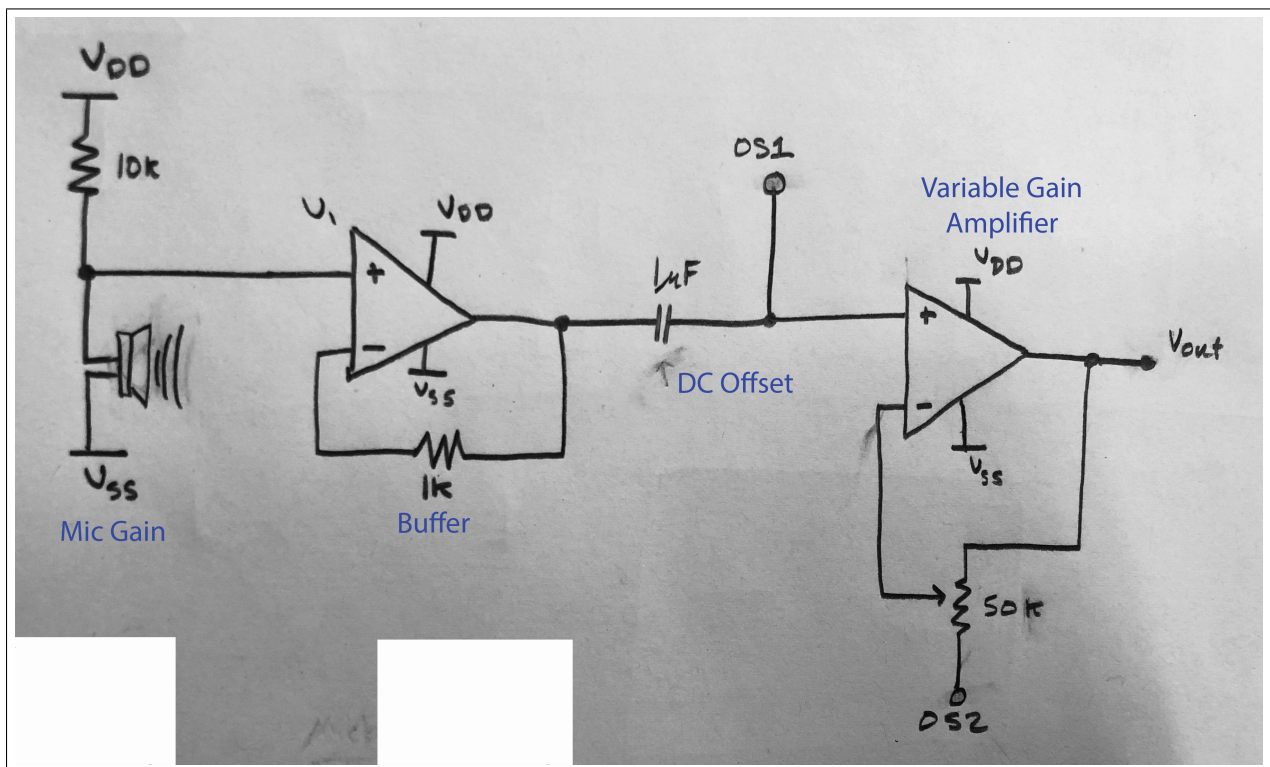


Figure 1: Full Circuit Diagram

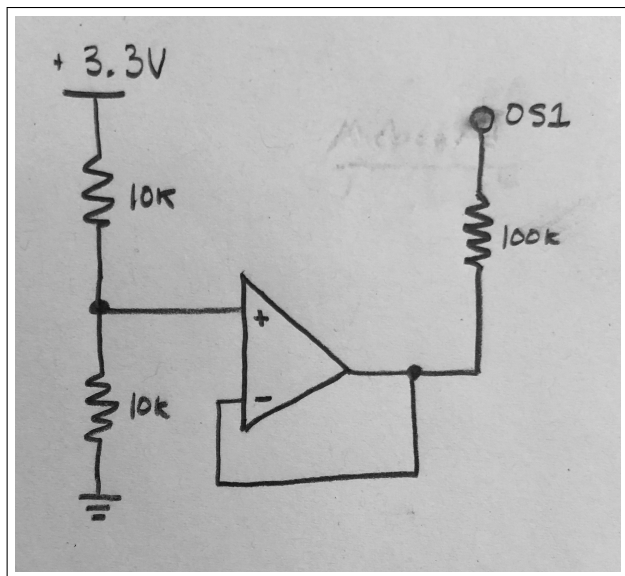


Figure 2: OS1 Circuit Diagram

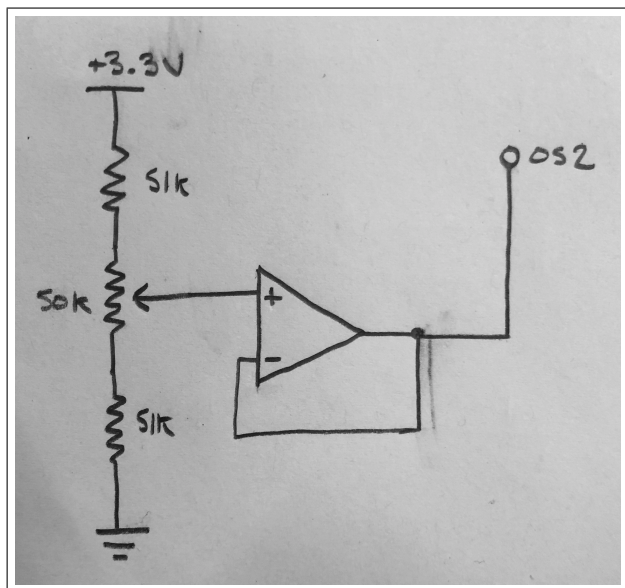


Figure 3: OS2 Circuit Diagram

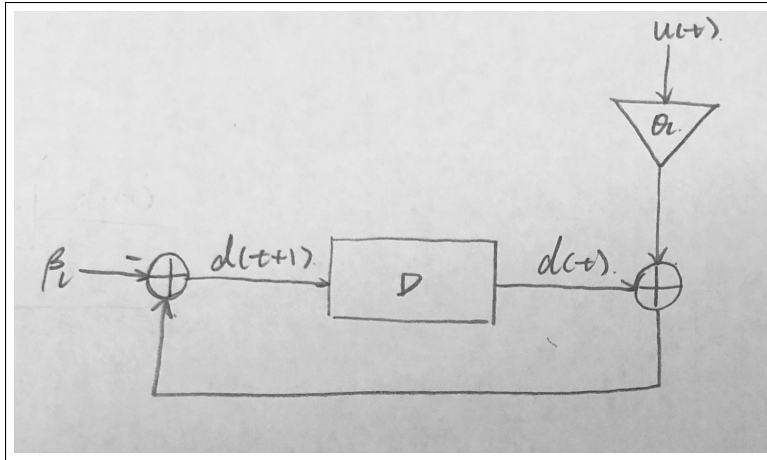


Figure 4: Closed Loop Control Scheme Block Diagram