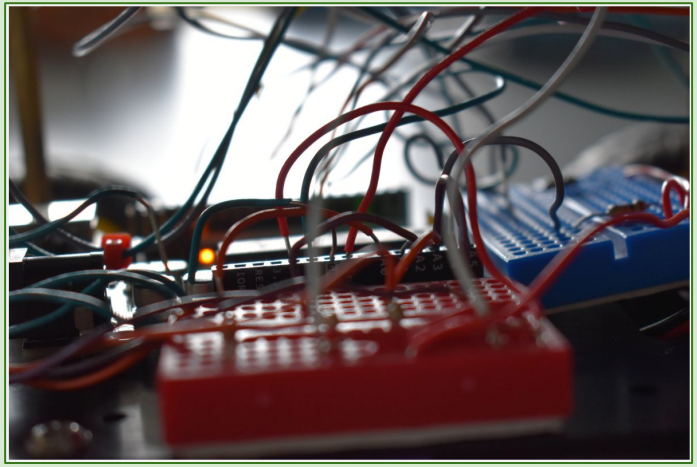
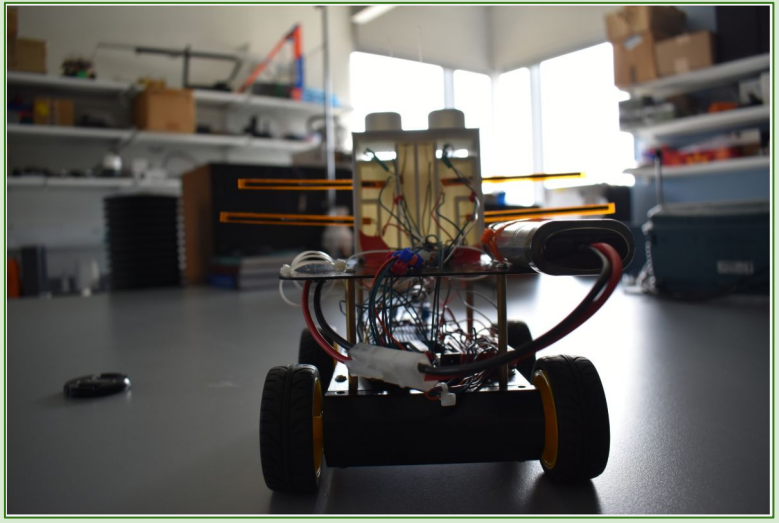
Box and Whisker Plot? No. More Like Cat and Flexor Bot!

By: Caitlyn Meaker and Forest Ray

Presentation Outline



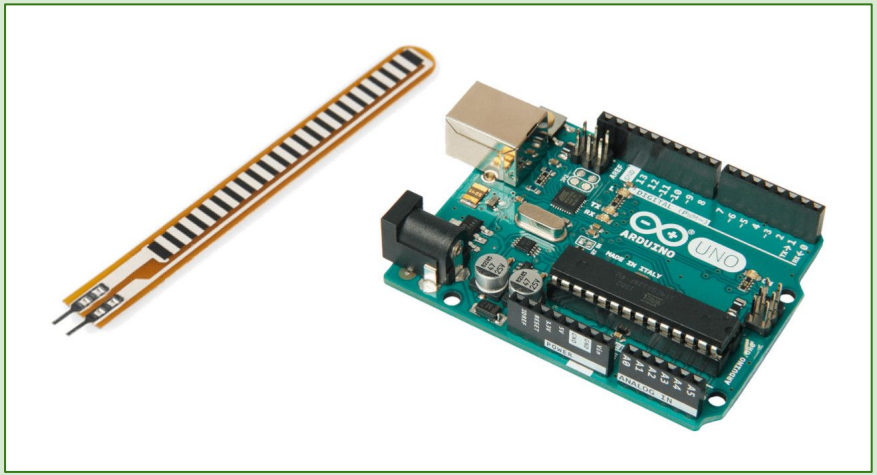
- Project Goal
- Background
- The flex sensors
  - Circuit Diagram on Robot
  - Sensor Code
- Code
  - Other Robot Functions
  - Data Collection
- The Set Up
  - Construction of the Robot and Tunnel
- Results
  - Video
  - Data
  - Did We Achieve Our Goal?
- Discussion
  - Limitations
  - Future Experiments
- Questions



Goal

- Utilizing the flex sensors, build a robot that is able to navigate tight spaces
  - Inspiration: Cat whiskers

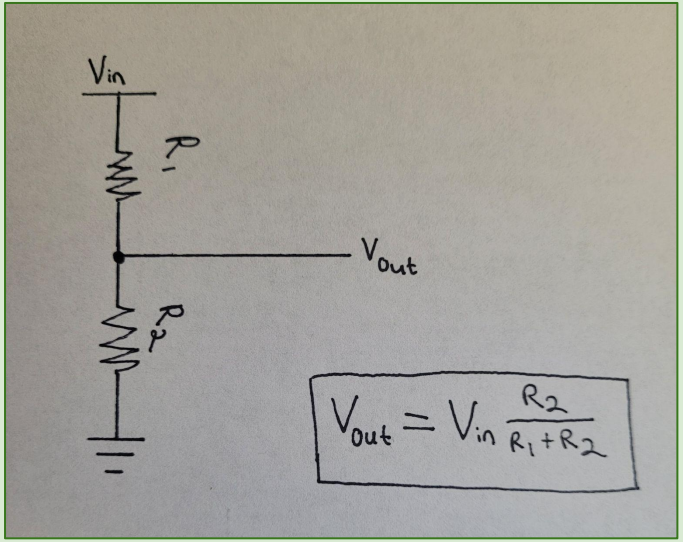
Cat whiskers. 16 Nov. 2023. https://www.nasc.cc/cat/a-cats-whiskers-how-do-they-work/. Accessed 19 Apr. 2025.



A Flex Sensor and Arduino Uno . https://peppe8o.com/wp-content/uploads/2023/04/flex-sensor-arduino-featured-image.jpg.

Background (What Does the Circuit Do?)

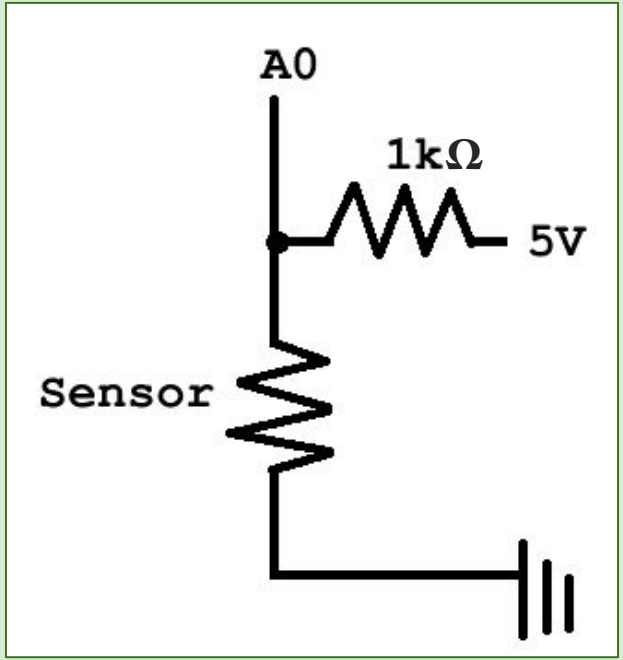
- Voltage divider
  - Read through Analog Pins
  - Two Resistors
    - One fixed and one variable
- Variable resistor = Flex sensor
  - Resistance changes when flexed
  - Ohm's Law: V=IR
- Monitor with Arduino
  - The change in voltage is measured by the analog pins
    - The Arduino monitors and responds

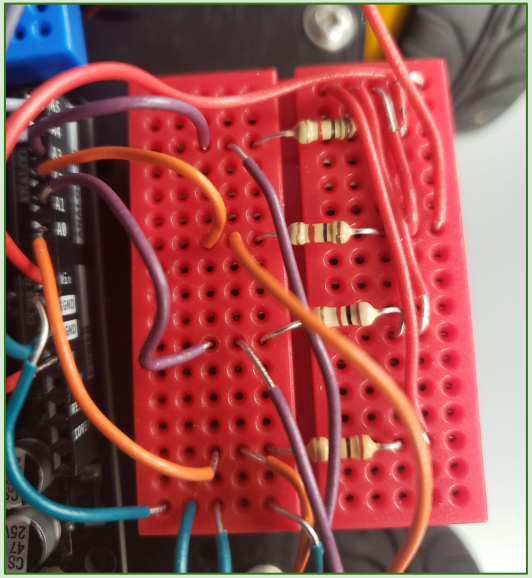


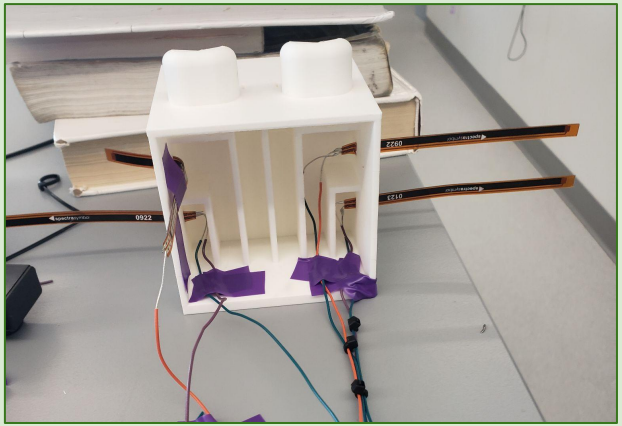
Voltage Divider Equation

Background (Continued)

- Our robot's voltage divider
  - Four flex sensors
    - The Variable
  - Four constant resistors







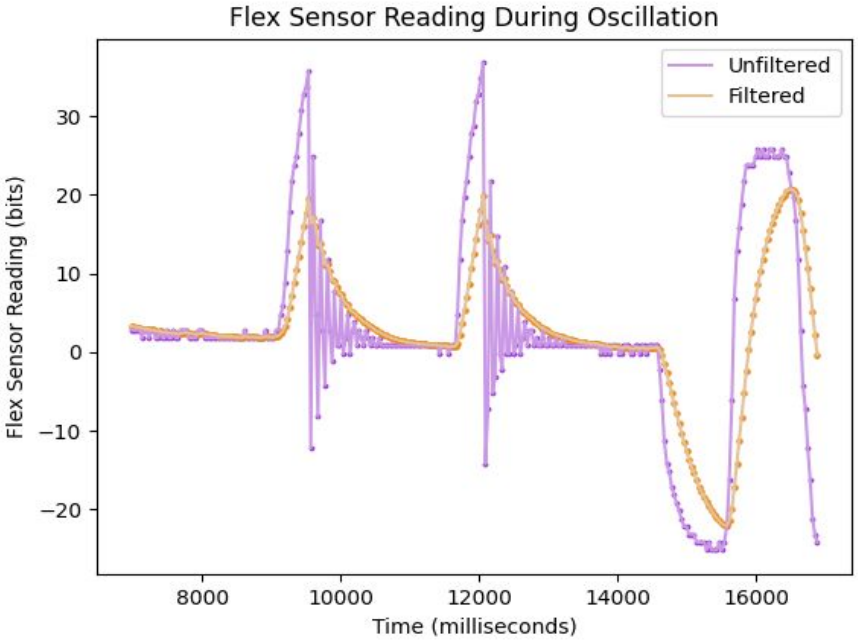
Cleaning Up the Sensor Data

Calibration

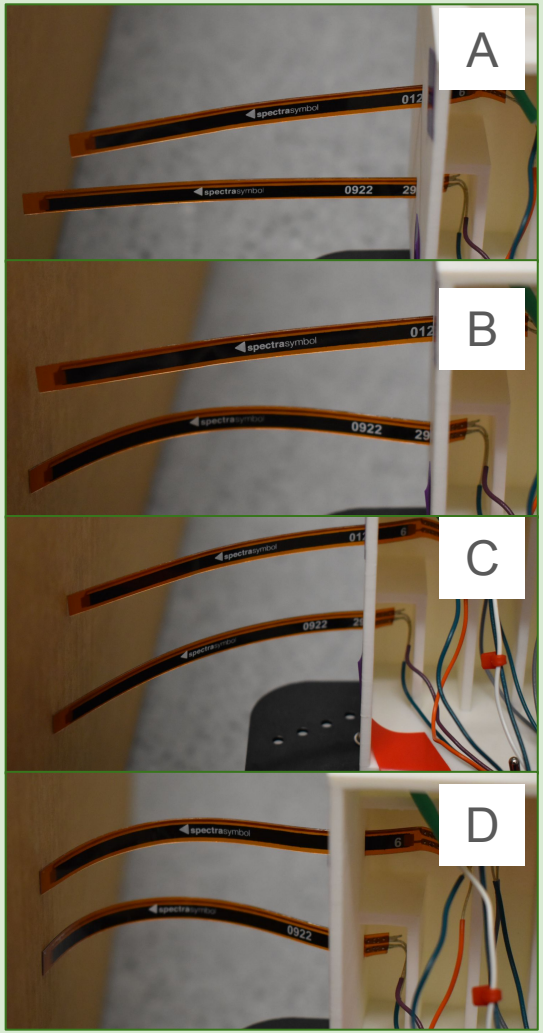
```
int number_of_readings = 10;
float sum_1 = 0; float sum_2 = 0;

for(int ii = 0; ii < number_of_readings; ii++){
  sum_1 = sum_1 + analogRead(FLEX_PIN_1);
  sum_2 = sum_2 + analogRead(FLEX_PIN_2);
}

baseline_1 = sum_1/float(number_of_readings);
baseline_2 = sum_2/float(number_of_readings);
```

1. Average 10 readings to get a baseline
2. Subtract baseline readings from subsequent readings

Filtering

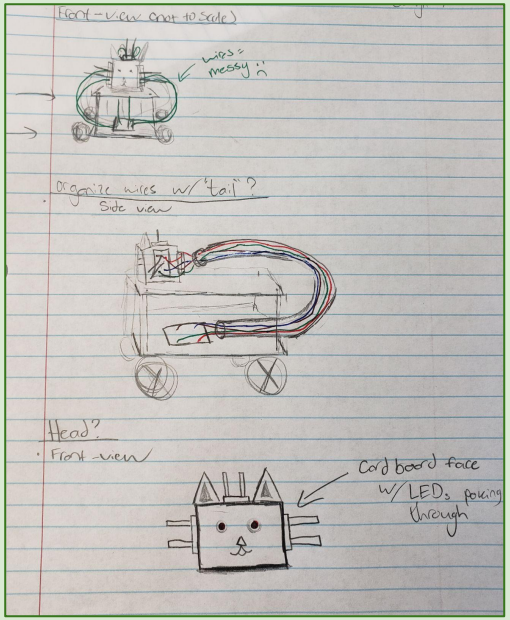


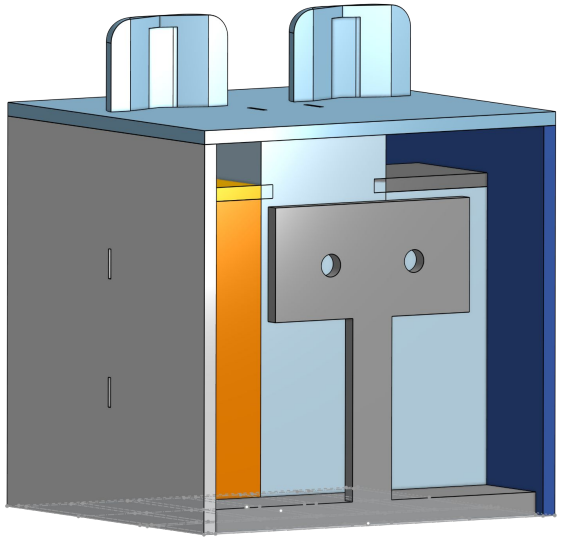

```
filtered_flex_1 = (0.90 * previous_flex_1) + (0.10 * flex_reading_1);
filtered_flex_2 = (0.90 * previous_flex_2) + (0.10 * flex_reading_2);
```

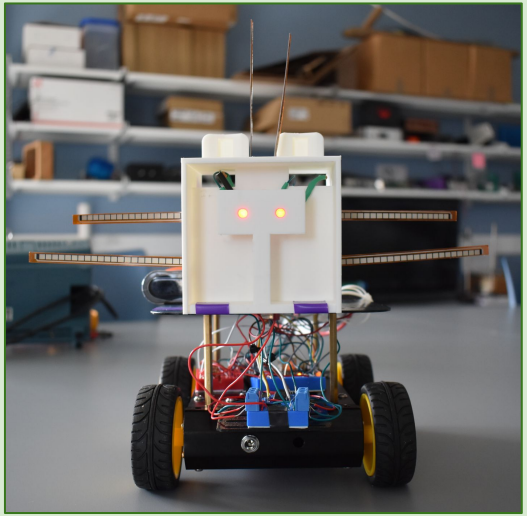


	A	B	C	D
A	State 1 (move forward)	State 2 (default adjust)	State 3 (weighted adjust)	State 4 (turn)
B	State 2 (default adjust)	State 2 (default adjust)	State 3 (weighted adjust)	State 4 (turn)
C	State 3 (weighted adjust)	State 3 (weighted adjust)	State 2 (default adjust)	State 5 (Stop)
D	State 4 (turn)	State 4 (turn)	State 5 (Stop)	State 5 (Stop)

Building a Cat Robot





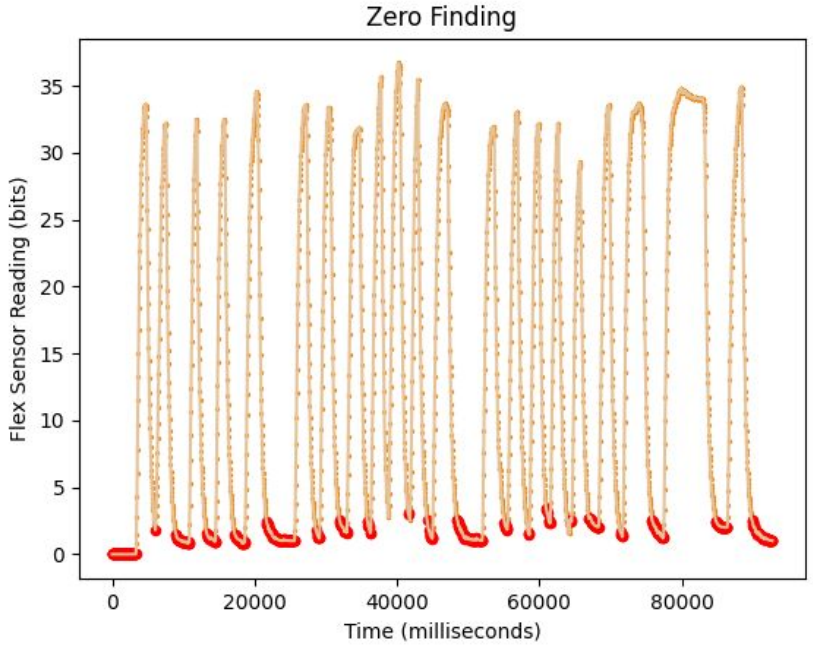


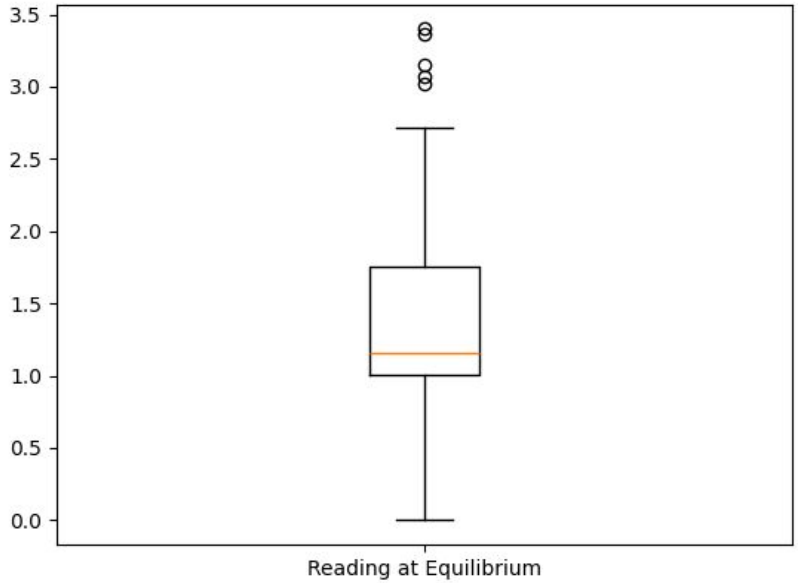
1. Planning → 2. 3D Modeling → 3. Assembly

Results

- Tunnel
  - Right and left turn → change direction
    - Can reverse
  - Narrows at end → stops
- Goal
  - Achieved

But.... Limitations





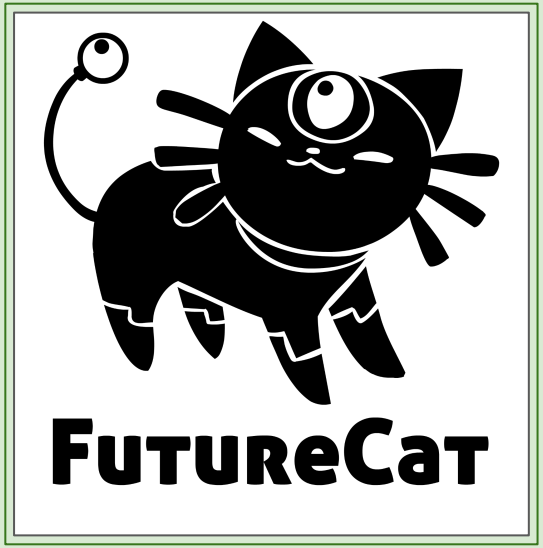
- Accumulating error
  - Sensors are calibrated at rest position
    - Once bent and released their position can change. However, this is not accounted for
    - Though small, overtime the error collects

But.... Limitations (Continued)

- Tunnel needed many adjustments
  - If a turn was too narrow and the Cat could not adjust, it would just stop even if the other set of sensors were not being touched
  - If an area was too wide the Cat would head straight for the next wall and crash
- Robot's speed
  - If it went too fast it would crash and/or try to climb the walls
- Sensors
  - Could get overwhelmed and need to be recalibrated

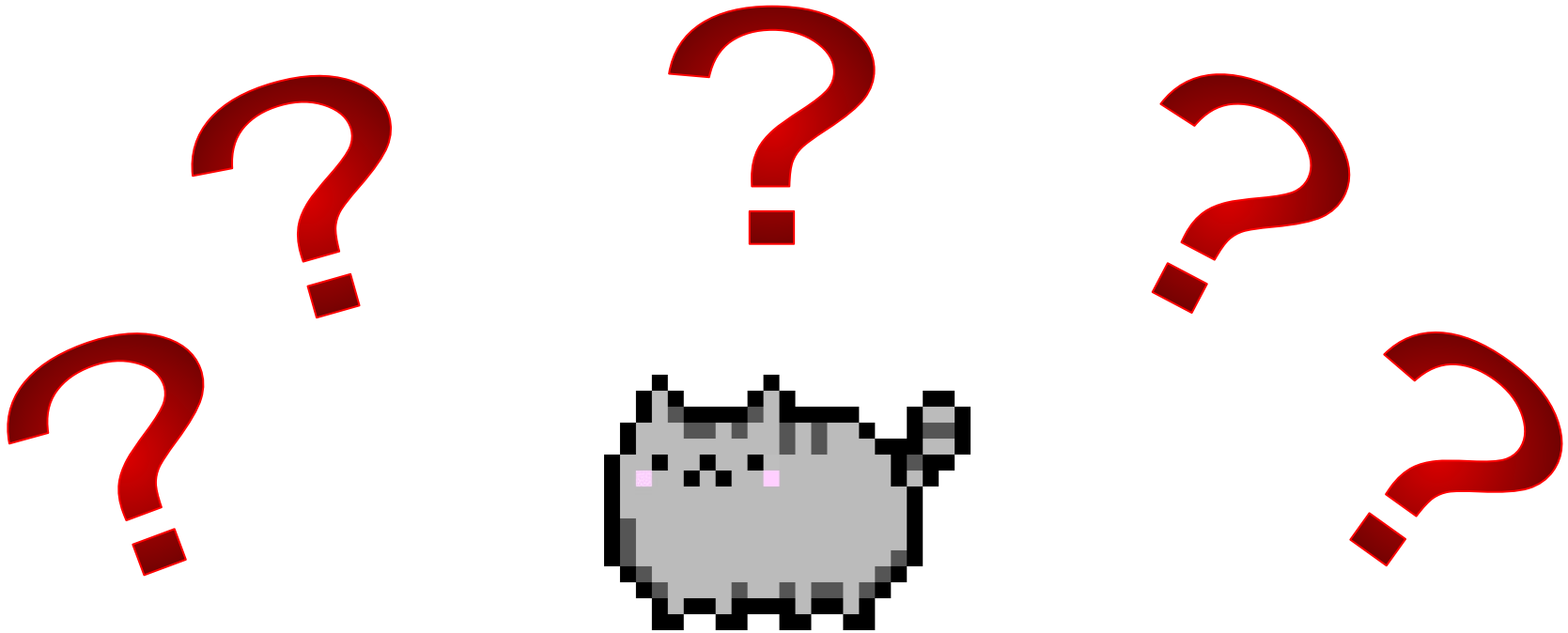
What Could be Done in Future Experiments

- More time and more/better equipment
  - Fine tune the code to allow for better adjustment of direction and velocity.
  - Determine more precise baselines
    - i.e be able to account for the change in position of the sensors
- Build a better/more sturdy tunnel
- Improved code
  - Improve range
    - Navigate more aggressive turns and other barriers



FutureCat. https://www.gematsu.com/companies/future-cat-games. Accessed 19 Apr. 2025.

Questions?




#GETSOME