Animatronic Fish Pillow: Enabling Expressive Interactions in a Touch Sensitive, Cable Actuated Robot

Team FlippityFlappity

Rundong Tian **Ryan Feng** **Eric O'Neill**

INTRODUCTION

The goal of this project is to create an interactive IKEA fish pillow that reacts to its environment by outfitting it with various sensors and actuators. Servos actuate a custom designed mechanical spine that will allow the fish to bend and complete different motions, such as 'flopping' and 'walking'. The fish will use tuned capacitive touch sensors and an accelerometer to receive data from the environment. NeoPixel LED strips are attached to the fish for additional interactivity.

SYSTEM ARCHITECTURE

Beam Bending Model

To understand how the fish will bend when actuated, we model the fish as a single beam. We define one "flap" as a bending of this beam in an arc with our preferred angle of curvature. Using a cable driven system, we capitalize on the changes in length induced by the compression and tension forces from bending in the beam. In a two cable scenario, one for the inner face and outer face of the beam, shortening the length of the inner cable and expanding the length of the outer cable induces the bending behavior we wish to achieve.

This contraction and expansion is achieved by the rotation of a servo and spool around which cables are wrapped. One of the main constraints on our mechanical design was the limited range of motion of servos and symmetric allowance for both contraction and expansion, restricting our maximum contractible cable to:$4R_{spool}$.

To allow a full range of motion, we used three cables positioned equidistantly around the spine as opposed to the two cable system considered previously. To achieve a certain position of the beam, the ratio of contraction and expansion of each cable is governed by the following set of

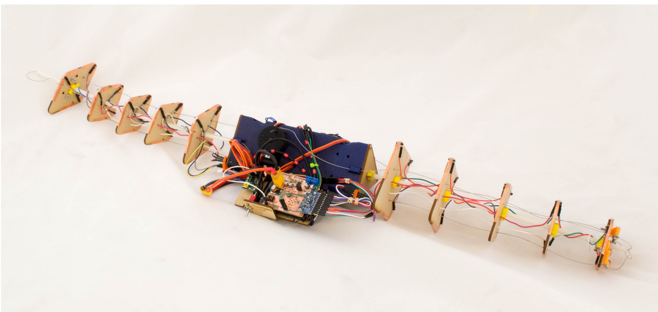

Figure 1. electromechanical backbone of our fish

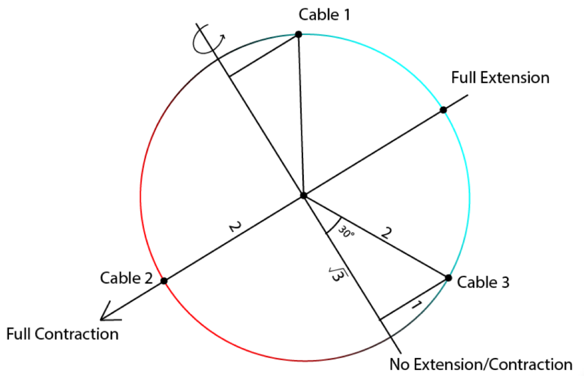

Figure 2. Example of contraction in Cable 2 direction and resulting ratio of extension of Cables 1 and 3 as a cross-section of the beam.

equations: Cable 1:cos(theta) , Cable 2: cos(theta - 2pi/3), Cable 3: cos(theta - 4pi/3). Theta is defined from Cable 1 and rotating counter-clockwise (Figure 1).

These considerations govern our mechanical design of the cable system, servo spool, and fish rib. Understanding the relationship of axial tensile and compression forces to bending also informs the way in which the servos must be commanded in software.

Mechanical System

Our mechanical system consists primarily of (1) a motor housing which holds three servos and the electronic system, (2) two flexible nylon rods extending out from the central motor housing to the head and tail of the fish, (3) ribs positioned at regular intervals along the nylon rods, and (4) a cable system connecting the servos to the nylon rods which bends the two rods symmetrically when the servos are actuated.

The physical characteristics of our system (size of the fish, maximum rotation of the servos), combined with the mechanics of the system we modeled guided the design of the wooden ribs, spool interfacing the cables with the servos, and choice of material for the central spine. In particular, the size of the spools were directly determined from our beam bending model to reach a target radius of curvature for the spine.

All components were fabricated using the resources at the CITRIS Invention Lab. Various subsystems of our mechanical design can be seen in figures 3 and 4.

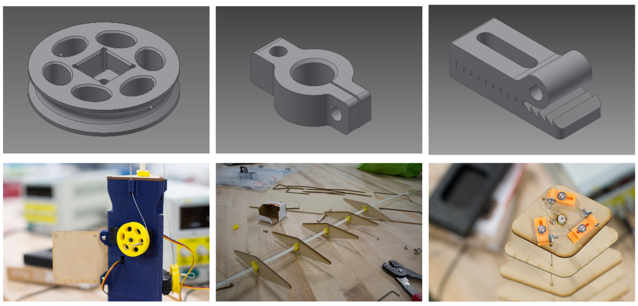

Figure 3. From left to right: spool for cable, clamp to connect the rib array to the nylon rod, adjustable cable clamps

Electrical System

A high level overview of our electrical system can be seen in figure 5. The Mbed FRDM-KL25Z microcontroller was used to control or monitor the following sensors and actuators: (1) the onboard accelerometer, (2) capacitive touch sensors, (3) three servos, and (4) two NeoPixel strips.

A Pololu Servo controller communicating through a serial interface was used to easily control the speed and acceleration of the servos, in addition to their position. Because servos are closed loop position control devices, actuating a servo at a desired speed would require constant resetting of the reference position, tying up the resources of the microcontroller.

Custom capacitive touch sensors were made by attaching copper tape to the edges of the wooden ribs. This allowed us to position the capacitive touch pads in strategic zones throughout the length of the fish (Figure 7). To detect the touches, we used the MPR121, an I2C capacitive touch controller.

The entire system was powered from a 7.4V LiPo battery, regulated to 5V with a high efficiency switching regulator for the servos and addressable LEDs.

The stall torque of each servo is 2.3A, and the max expected current draw from each pixel in the addressable

Figure 4. CCW from top left: laser cut ribs, buffer pieces to increase the distance between the spool and the first rib, one section of the servo module, assembled mechanism with one nylon rod

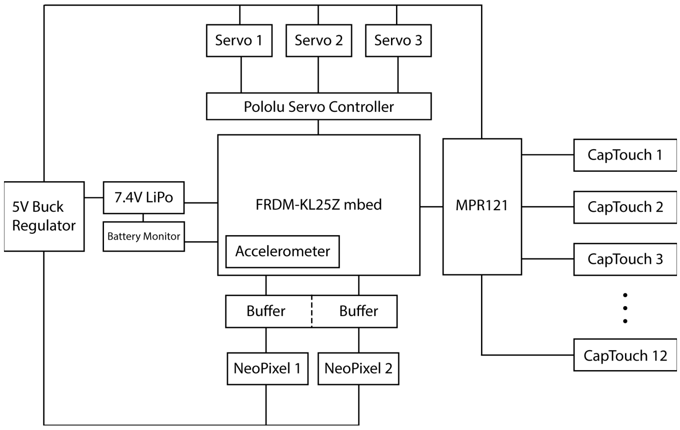

Figure 5. Electronic System Diagram

light strips at half brightness is 30mA. The worst case scenario current draw of our entire system is 11A.

A custom PCB (Figure 6) was designed with the high current requirements of our system in mind, using large planes to route power rather than traces.

INTERACTIVE BEHAVIOR

The objective of our behavioral design was to create various reactions to convey life and emotion. Thus, we constructed a hierarchical finite state machine of the fish's behavior beginning at the highest level with these emotions and ending at the lowest level with servo and NeoPixel actuations. An important consideration was simplicity at the lowest level FSMs, as the actuation of the spine was the most difficult task to design and debug, while statechart transitions, touch sensing, and NeoPixel color arrays were more reliable. Additionally, a concurrent state machine is implemented to intermittently monitor battery life as a safety precaution.

We began designing our FSM by exploring statements such as, "tickling a happy fish will make it laugh" or "a curious fish will move and look around on its own until you say hi or it gets fed up". This led to the creation of four emotion states in a high level FSM: CURIOUS, MAD, HAPPY, SAD (figure 8). Each of these states has its own FSM that represents the emotion, whose states are actions to perform. In the high level emotions FSMs, the output is the change of the emotion state, while in each action state, the output is

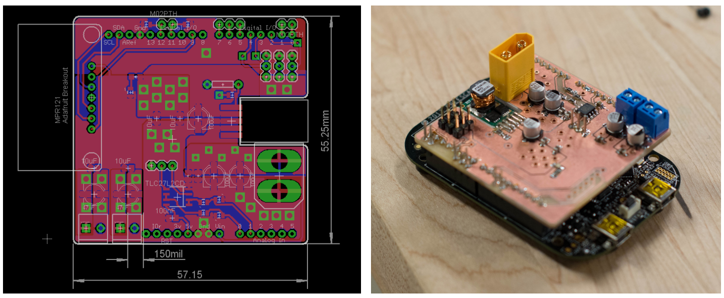

Figure 6. Custom PCB shield for the FRDM-KL25Z. In addition interfacing with our sensors and actuators, it includes a connection for the LiPo battery, op-amp circuit to monitor the battery voltage, and screw terminals for the power rails

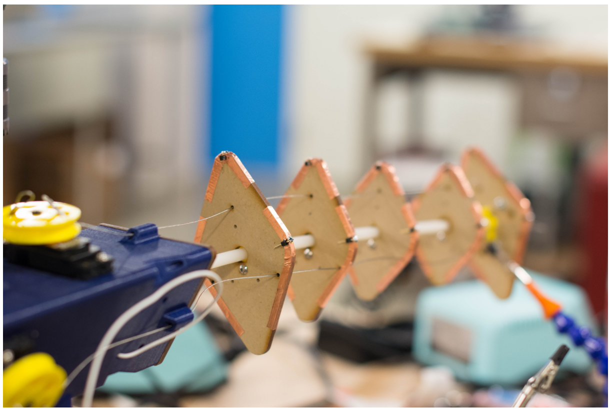

Figure 7. Capacitive touch pads are positioned along the length of the fish by attaching copper tape to the existing wooden ribs

actuation of the NeoPixel and servos. Servo action in these states can be consolidated into six actions states: REST, LOOK_LEFT, LOOK_RIGHT, FLOP, WALK, FLAP, where each action differs depending on the emotion. For example, a flap is continuous actuation up and down, where an angry flap is a tantrum, a happy flap is laughing, and a sad flap is the slow breathing of hibernation. The movements made in each emotion state are augmented through colors displayed through the addressable LED strip.

Finally, to simulate the fish's self-determination, an internal variable `trans` is generated on each iteration based on a pseudo-random number generator to choose whether to take a transition between action state. As an example, consider the FSM in the happy state in figure 10. A happy fish, if touched, will nondeterministically choose to laugh or chase the source of the touch. When it has not been touched, it will attempt to call attention to itself, then it will look around once more for attention, then become sad. While this could be easily implemented on a timer, a non-deterministic transition is more appropriate for a "live" being.

SOFTWARE IMPLEMENTATION

The structure of the program is to first initialize and calibrate all sensors and actuators, then enter a loop which polls all sensors for new data and passes sensor and timer data into the statechart. The statechart would first determine transitions based on the sensor data, and then command the actuators accordingly based on the current state. This structure enabled us to easily add, remove, or change states as we saw fit, which made it easy to modify the behavior of

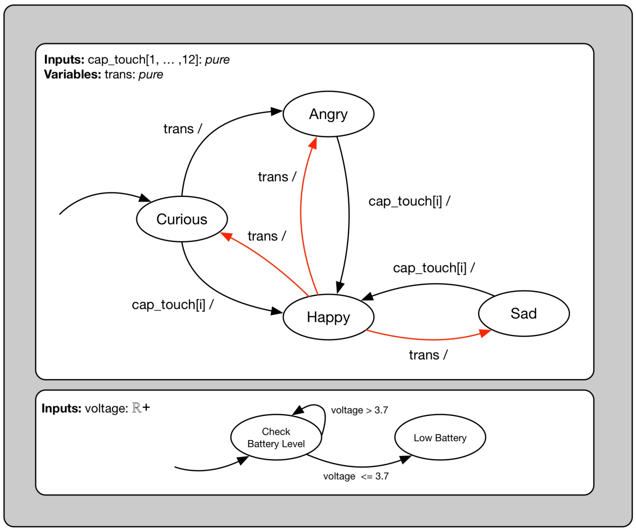

Figure 9. High level FSM containing all emotion states. A concurrent state machine monitors the LiPo battery voltage to prevent over-discharging

our fish. In addition to our main loop, our code contained two additional concurrent processes: monitoring the voltage level of the LiPo batteries intermittently using a timer interrupt, and reading the state of our capacitive touch array when an external hardware interrupt is triggered by the MPR121 chip.

We used the MPR121 library for the capacitive touch sensors, the Multi-WS2811 library for the NeoPixel strips, and the MMA8451Q library for the on board accelerometer.

DISCUSSION

Our system hardware is modular enough that new sensors and actuators could easily be added. One sensor that we

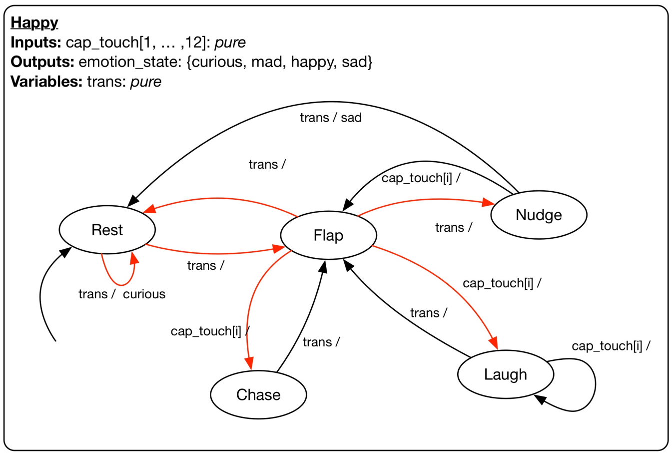

Figure 10. Low level FSM for the Happy emotion state

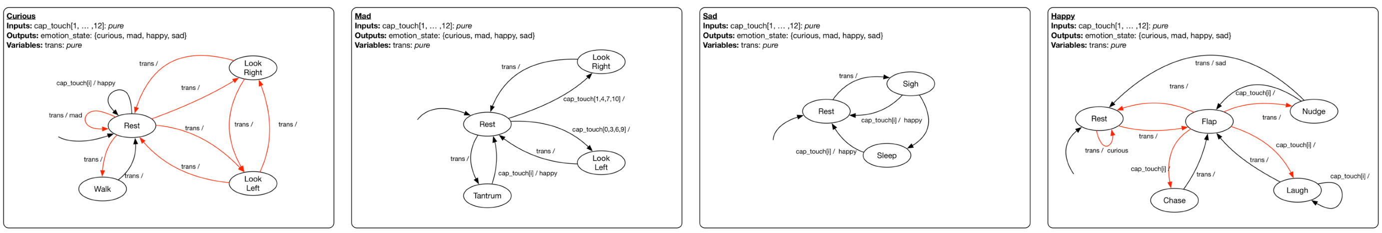

Figure 8. FSMs for each of our high level emotion states

thought would be entertaining would be a Hall effect sensor near the mouth of the fish that could sense a magnet attached to a fishing rod. This would allow for interesting interactions with the fish that would simulate fishing.

Another improvement to the system that we had in place would be a better "walking" motion. The fish's current behaviors mostly require it to remain in a static location, however if the fish could move from place to place it could convey a wider range of emotions and be more fun to play with. This could also work well with the fishing behavior described above, as the fish could simulate an attempted escape.

CONCLUSION

In this project, we designed, prototyped, and programmed an animatronic fish which responded to its environment. This allowed us to explore the interactions and behaviors that were possible with our array of sensors and actuators. In the end, the fish could display a series of emotions both through the color of light it emitted and through controlled movement from the servo motors. These emotions could be transitioned both non-deterministically and through user input from capacitive sensors and an accelerometer. Overall, this project served as an excellent learning tool which helped our group understand both the complexities of integrating multiple sensors and actuators into a physical system, and in modeling and implementing a series of motions that can be defined as behaviors.

ACKNOWLEDGEMENT

We would first like to thank our professors, Edward Lee and Alberto L. Sangiovanni-Vincentelli, and our GSI Ben Zhang. In addition, the digital fabrication resources of the CITRIS Invention Lab were invaluable for the implementation of our physical prototypes. Finally, the following code libraries and breakout boards were used in this project:

Pololu Micro Maestro Servo Driver: http://www.pololu.com/product/1350

Adafruit MPR121 capacitive touch breakout: http://www.adafruit.com/products/1982

MPR121 library: http://developer.mbed.org/users/4180_1/notebook/mpr121-i2c-capacitive-touch-sensor/

Multi_WS2811 library: http://developer.mbed.org/users/bikeNomad/code/Multi_WS2811_test/

MMA8451Q accelerometer library: http://developer.mbed.org/components/FRDM-KL25Z-on-board-Accelerometer/