Adrienne Chang

(626) 818 – 2900
adc005@uscd.edu

Professional Portfolio

Last updated: July 2018

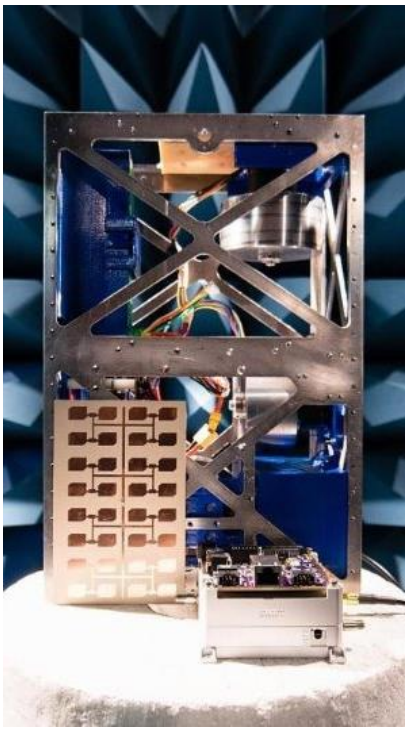
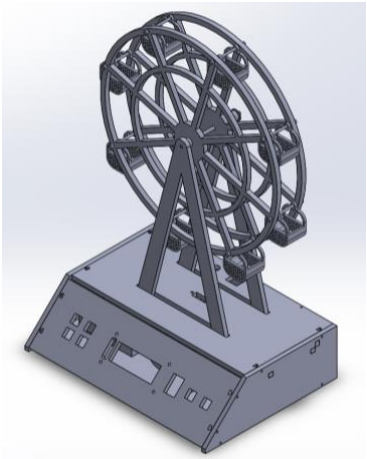


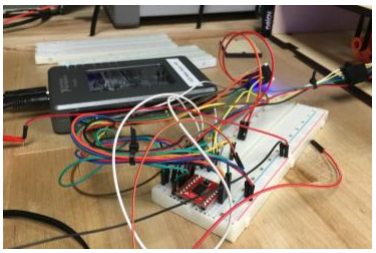
Table of Contents

LabVIEW Elevator System

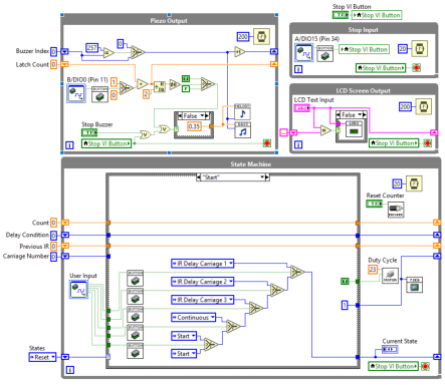


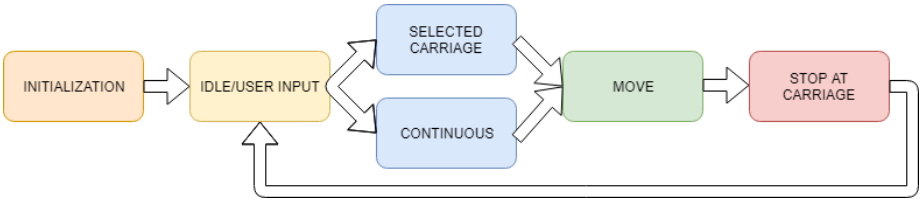
I created a functioning ferris wheel for a 10-week project course utilizing LabVIEW's state machine architecture and the myRIO Student Embedded Device.

To simulate an elevator system, each carriage would represent a stopping point similar to stopping at each floor of a building. The mechanical system was modelled in SolidWorks 2017 and the exported DXF files were used to laser cut the pieces out of ¼" plywood. As part of a rapid prototyping class, all the connections were made with temporary joints (butt joints and press fit) for easy troubleshooting.



The basic mechanical movement utilized a DC motor and a timing belt to turn the wheel. An IR sensor placed below the carriages monitors the passing carriages, providing feedback to the state machine.



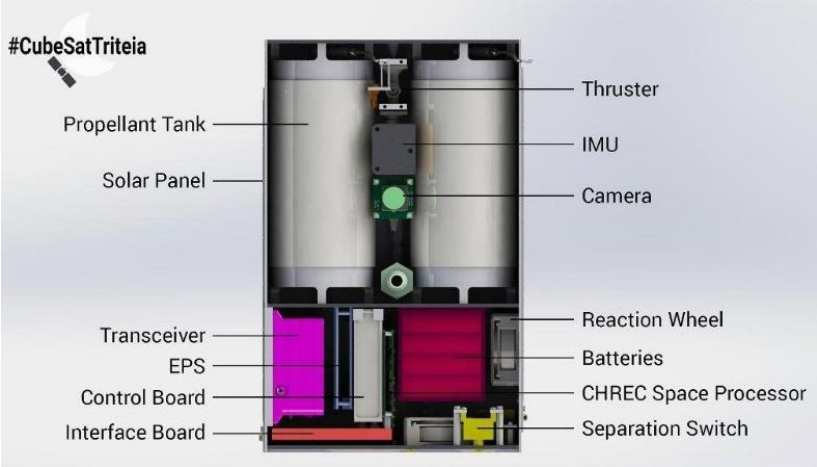


Based on the user input, the elevator will rotate the ferris wheel to bring the designated carriage to the bottom. A second option is to continuously run the motor until the user chooses to stop the ferris wheel.

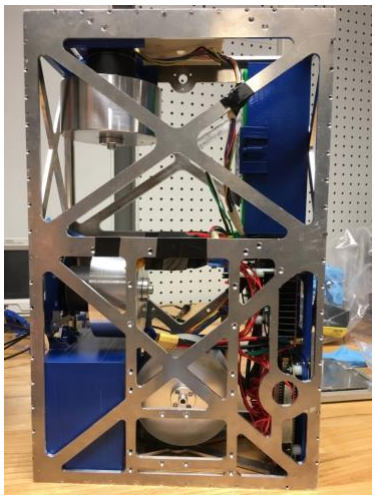
The code was structured and refined over a five-week period. SubVIs were first created, one for each module that was implemented (motor/encoder, IR sensor, buttons, LCD, 7 segment display) and implemented into the state machine to streamline the code. Parallel loops were utilized to allow separate modules to run smoothly. I further optimized the code by reducing the number of states from 9 to 5. The final product was selected as best design out of the entire class and presented to industry representatives.

TRITEIA 6U CubeSat

I worked on TRITEIA, a semi-autonomous chemically propelled 6U CubeSat which competed through the final round of the NASA's Cube Quest Challenge.

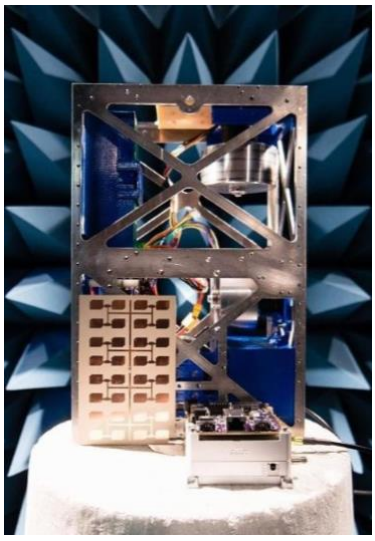


As the CAD lead of the structures team, I oversaw the 3D modeling of the chassis in SolidWorks 2017. The modeled chassis consisted of six aluminum sheets that were designed to fit within the constraints of the payload as well as house and mount all of the components.



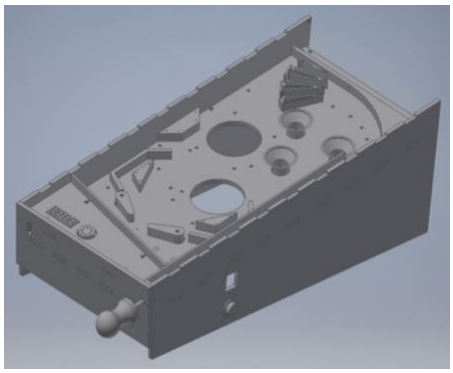
To see whether our design was feasible, the structures team created model parts for every component down to the screws we would need. The overall assembly was uploaded to GrabCAD Workbench to manage version control. From there, any changes made would be pushed to keep the most updated version accessible to all team members.

I constantly communicated with the other teams to stay updated on each subsystem's requirements. In order to understand how the components needed to be arranged within the chassis, I met with team leads of propulsion, avionics, communication and power to understand the constraints on their components and relayed respective changes to the structures team.



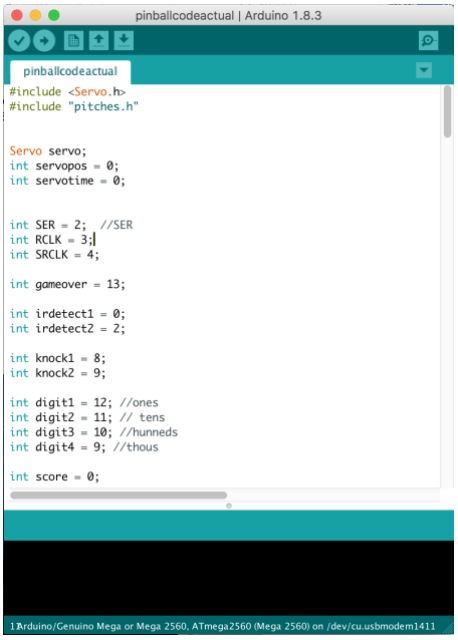
I assisted fabrication of the chassis out of six 7075 T6 aluminum sheets which were selected for its high strength and light weight. I helped program CNC machines to cut out the chassis and tapped the respective holes that would hold the chassis together. The final design and submission was chosen to be a backup SLS payload in last ground tournament of NASA's Cube Quest Challenge.

Pinball Design Project



I designed and built a functioning pinball machine over a 10-week project course utilizing an Arduino Mega and class provided supplies.

The first iteration was a prototype made out of cardboard. I built the playing field to ensure that the final design can fit within the restricted size and leave enough room for the planned modules.



I then 3D modeled the entire pinball assembly and its individual parts in Autodesk Inventor. The exported DXF files were used to laser cut the structure from acrylic and plywood while more complicated shapes like the bumper caps were created with Zortrax 3D printers. I assembled the base structure using M2 cap screws and bolts to allow the ability to disassemble and troubleshoot through the prototyping phase.

I built a MOSFET driver circuit to control a pull-type solenoid which provided the main mechanics of the flipper and tested it with a function generator with a variable duty cycle. The same basic circuit was used for the bumpers just with a different input. I built a test stand for the bumper to find which design would work best for it to register an input.

I prototyped each module individually by building the circuits for them first and then coded on Arduino to test their functionality. The modules that were incorporated into the final design were the speaker, 7 segment displays, flippers, DC motor, servo, IR sensor, piezo and bumpers.

The resulting final design allowed a user to play three rounds per game while scoring through the multiple sensors placed around the playfield and was showcased to our faculty and students at the end of the quarter.

A Comparison of Supervised Learning Algorithms in Python

Table 4. KNN of ADULT Dataset

Partition (Training/ Validation)	Average Train Accuracy	Average Validation Accuracy	Average Test Accuracy
20% / 80%	74.59%	73.20%	76.48%
50% / 50%	75.75%	74.53%	76.68%
80% / 20%	76.71%	75.54%	74.90%

Table 5. KNN of COV_TYPE Dataset

Partition (Training/ Validation)	Average Train Accuracy	Average Validation Accuracy	Average Test Accuracy
20% / 80%	76.67%	76.22%	89.50%
50% / 50%	76.14%	75.67%	89.67%
80% / 20%	76.45%	75.99%	89.33%

Table 6. KNN of LETTER Dataset

Partition (Training/ Validation)	Average Train Accuracy	Average Validation Accuracy	Average Test Accuracy
20% / 80%	66.51%	66.19%	95.06%
50% / 50%	67.15%	66.56%	95.49%
80% / 20%	68.37%	67.96%	94.99%

Table 12. Mean Test Accuracies

Method	Dataset	Average Testing Accuracy/Dataset	Average Testing Accuracy/Method
KNN	ADULT	76.02%	86.90%
	COV_TYPE	89.50%	
	LETTERS	95.18%	
SVM	ADULT	75.60%	79.57%
	COV_TYPE	70.13%	
	LETTERS	92.97%	
ANN	ADULT	78.20%	82.77%
	COV_TYPE	83.02%	
	LETTERS	87.09%	

Table 13. Test Accuracies Comparison

Method	Results	Literature Data (Caruana and Niculescu-Mizil)
KNN	86.90%	81.50%
SVM	79.57%	78.10%
ANN	82.77%	84.20%

For a final project, I emulated the results of three classifiers used in Caruana and Niculescu-Mizil's "An Empirical Comparison of Supervised Learning Algorithms." I trained and tested three datasets from UCI's repository using K-Nearest Neighbor (KNN), Support Vector Machine (SVM), and Artificial Neural Network (ANN).

Before analyzing the data, I converted the datasets to csv files for easier implementation and randomized the dataset. I then extracted the first 5000 instances for training and testing and separated the classification column. Cross-validation with five folds and grid search was applied to each of the algorithms before three different partitions of training/validation were performed.

I concluded generalizations from the results of the different test sets. For KNN, the best partition to use is 50% training and 50% validation. For SVM, more support vectors can be drawn to the decision boundary when the training set is larger thus increasing test accuracy. Since ANN is a neural network, it performs better as it's given more data to learn from.

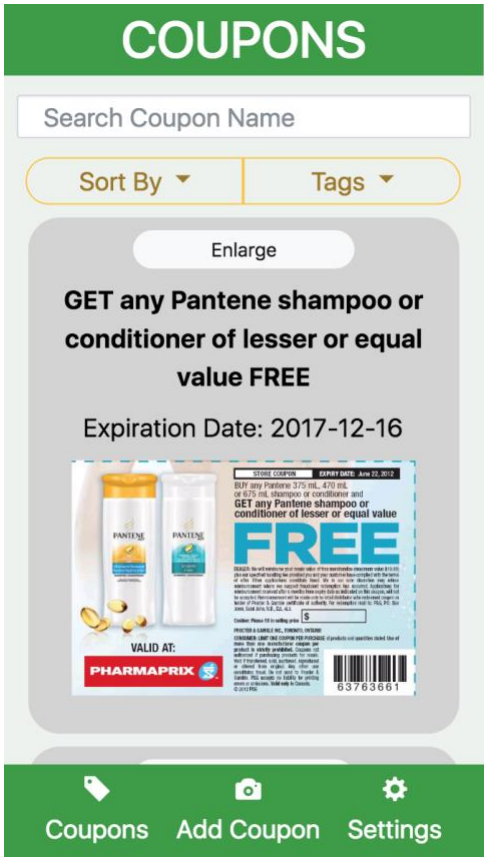
After comparing my results with Caruana and Niculescu-Mizil's results, the differences in the testing accuracies can be attributed to the varied implementation of parameters and the use of only 5000 instances of the entire datasets. Overall, the results obtained could be greatly improved by implementing the algorithms on the entire dataset with a larger number of metrics and partitions to really observe the pros and cons of the three algorithms for each dataset.

Coupon Saving Web Application

I researched and prototyped a web application using the needfinding process, heuristic evaluations, and A/B testing. The final design is a web application designed to help senior users collect and keep track of their coupons.

To start the needfinding process, I went to the local grocery store and observed seniors during their shopping process. The seniors were asked to think aloud and were questioned on some of the choices they made. Different app solutions were extracted from this collected data.

I created storyboards that would exemplify the use of the app solution and flushed out the screen designs. I created paper prototypes that would simulate the app design and proceeded to get feedback.



To assess the problems within the prototype, I performed heuristic evaluations on three users. The three biggest issues were word phrasings, user control and freedom, and error recognition. I created improved designs on wireframes so that images were clearer and more senior friendly.

I performed a second round of user testing on two users by giving them specific tasks and seeing how well they accomplished it with minimal help. After correcting unintuitive designs on the app, I used Google Analytics to perform A/B testing on 20 users. The results showed problems users would run into that inhibited the ease of using the app.

The final design was shown to the class and students could test the functionality of the web application.