

Shock Dyno Control GUI and MCU Interface

Milestones:

- M1: Demonstrate simple command and control of microcontroller through serial connection (ideally Teensy 3.2) from MATLAB. Hook up a sensor of some sort (LDS for example) and show that the MCU can send and receive data at a decent rate, also hook up a controlled device (e.g LEDs or an analog device) and show that they can be controlled
- M2: Create Basic GUI that works with the serial connection that shows a live plot of incoming sensor data, or at least writes all the data to some permanent storage. This should have some or all of the features listed below
- M3: Integrate digital pot code with MCU code to show that pot can be controlled from MATLAB. Prototype and test code robustness features (what about e-stop, calibration or sensor checks at startup, making the code work with windows and mac...you decide)
- M4: Refine GUI to be working 'final model'. Should include all or more of the features listed below (unless there are some hiccups and we change our mind)
- M5: Test MCU and MATLAB code with hardware prototypes of control system and conditioning system. Essentially, test the entire system before everything is finally assembled/taken off breadboards to verify functionality.
- M6: Run first shock dyno tests. Tweek and improve as necessary
- M6: Create documentation/instructions for repair/maintenance/how things work
 - Completion deadline: **October 22 2019**

Deliverables/Explanation:

- The goal of the GUI is to make the operation of the shock dyno simple, safe and interactive. The MATLAB should do front and back end where possible, meaning it must make the GUI work but also do some math/computation on the data as it arrives. Understanding this backend stuff means understanding what the purpose, limitations and mechanisms of the shock dyno are. The end goal is for it to produce relationships used to process data off the car, or to quantify changes resulting from tuning.
- You will need to collaborate to make sure that the commands from MATLAB can be used to control the other systems, and so that those systems can quickly feed data through the MCU and back to MATLAB. Collaboration at all stages will be key.
- The functions of the GUI will change with time, but the first demand is to make it run tests on the shocks we have in order to process the midnight mayhem and other data. So, the required features are:
 - Have a two modes/tabs. One to do simple spring constant tests where the motor is off and only FT, LDS data are being read in steps to determine a spring constant. The other is to control the motor for dynamic tests.
 - For spring constant mode: have a 'save to array button' that takes the current FT and LDS measurement (or a short time average) and writes it to an array.

- For dynamic mode: have a motor control input panel. Offer types of input like steady state actuation or a frequency sweep.
- Also for dynamic mode: chart the incoming FT and LDS data in a live plot that gives the user a sense of whether things are working properly.
- Also for dynamic mode: have robustness limits: an e-stop button, built in controls on max speed, max rate that speeds can build up, ways of determining and reporting what the current oscillation frequency is...be creative here, but more info to the user is better than less. This will be discussed in person nearer the time.
- We should strive for something truly foolproof that checks as much of the system as possible to make sure everything is working properly and is safe to go. This is a good space to innovate in.
- Must be well documented for future (uninformed) users (instruction set in particular)

Tips, details and pertinent information:

- MCU to be used: Either the Arduino Nano: <https://store.arduino.cc/usa/arduino-nano>
Or the Teensy 3.2: <https://www.pjrc.com/store/teensy32.html>
- Digital pot we have: <https://www.sparkfun.com/products/10613>
- DAQ GitHub: <https://github.com/GTOR-BajaSAE/DAQ> Check here for previously developed Teensy-MATLAB code, that will be the best place to start.
- GET THE GUI TOOLBOX FOR MATLAB
- Purchasing spreadsheet:
https://docs.google.com/spreadsheets/d/1ILN35p0WuqsaUOu9AnwBnsDROKAHuUL8cqE_y23wues/edit#gid=1937511476
 - Use line item number "21" and "DAQ shock dyno" for purpose. Ask Billy before adding anything to the list. Visit the invention studio/Hive to see if they have hardware before buying anything.