Formula Hybrid - High Level Design

Diego Contreras, Owen Degnim, Kyle Fulk, Zhuoran Han, Brendan Mahoney

Professor Mike Schafer

Senior Design - EE41430

Fall 2020

Table of Contents

1. Introduction

We will be working with the Notre Dame Formula SAE Hybrid Racing team to improve the electrical system in the hybrid vehicle. The car is a series hybrid powered by a capacitor bank in series with an internal combustion engine (ICE). A generator motor converts mechanical energy from the ICE to electrical energy. The capacitor bank serves as an energy buffer between the generator and the electric hub motors on the two front wheels. Each of these motors is controlled by a Kelly motor controller. This document outlines the overall design of the electrical system for the hybrid vehicle. All system components are selected under the criteria of the Formula Hybrid rules for the 2021 competition.

2. Problem Statement and Proposed Solution

2.1 System Status Interface

The current system status interface uses an LCD inside the car to update the driver on fuel level, ultracapacitor charge, vehicle speed, engine speed, and error messages. The system is also designed to have off-track transmission via an RF transmitter. The transmitter uses UART communication to transmit diagnostics of the vehicle to an off-track computer.

Currently the system is unable to transmit data to an off-track computer using UART communication in real time and there is no data logging feature. Also the team last year was unable to fix issues with the LCD display inside the car, such as brightness and color contrast. The off-track RF transmitter needs additional work to determine the baud rate of the transmission system. Signals are determined to be sent and received, but our group cannot decipher such signals without the correct baud rate, which was not documented by the previous year's group.

We plan to solve the transmission problem by fixing the communication protocol. We will attempt to find the baud rate of the UART communication. Alternatively, we can implement other protocols such as SPI and I2C. Last year's project used MATLAB to log data. We plan to improve the GUI and monitor the vehicle in real time, which may require us to use different software tools. For the LCD display, one planned improvement is the addition of a display for motor temperature and controller temperature. We also plan to fix the LCD brightness and color contrast in order to make the display more visible to the driver.

2.2 CAN Bus

The current CAN system is not able to distinguish the difference between the left and right hub controller messages. This is problematic for fine-tuning the torque-vectoring method that has been developed. The CAN contains error messages for the motors, so not being able to distinguish the right from the left delays the debugging process.

To fix the problem with the CAN drive, one of the possible solutions is to use the "Preferred CAN address." Using this method we would set the identifying bits and use the Saleae Logic Analyzer to identify the corresponding addresses. The other possible solution is to implement two separate CAN busses, one for each side. According to the documentation from the previous team, the PIC32MX795, and the motherboard that they were using would be able to accommodate an additional CAN bus system and thus facilitate the transmission of error messages from the individual motors.

2.3 Engine Feedback Loop

The current engine feedback loop consists of the ICE, a sensor to measure the motor velocity, a servo to adjust the ICE throttle, and a PID controller implemented in the motherboard. The purpose of the system is to hold the ICE at a set RPM to balance efficiency and output.

The RPM sensor uses an off-the-shelf IC chip along with a low pass RC filter to measure the frequency of the alternator output. The RPM sensor was not implemented last year due to the cancelation of classes in the spring. The biggest problem with the system is noise that occurs on the two signals, RPM sensor-to-motherboard and motherboard-to-servo. The system works when the motors are turned off, but the EMI generated by the motors and motor controllers cause erroneous RPM readings and random servo settings. The current algorithm is very simple in that it only holds the engine at a set RPM. This algorithm doesn't account for current accumulator charge values. It also must be disabled to run the car in idle mode.

Due to the cancellation of in person classes last semester, last year's team was unable to work on the engine feedback system and integrate the RPM sensor into the system. The primary goal for this year is to continue last year's goal of adding a noise resistant cable between the motherboard and servo motor. The signal noise problem can be solved by simply using shielded cables. To sense the engine RPM we will take advantage of the Kelly Motor Controller. Instead of using a separate circuit sensor like last year's team proposed, we will use the motor's hall sensors and the Kelly Motor Controller's speed measurement tools to read the RPM from the generator, which is the same as the engine RPM.

A secondary goal for this year is to implement software that would be able to change the RPM setpoint based on the voltage of the capacitors. Instead of only operating at maximum ICE efficiency, the new system will be able to adapt to the energy needs of the vehicle. If the capacitor voltages were measured to be below a threshold, the RPM setpoint would be increased. If they were measured above a threshold, the RPM setpoint would be decreased.

3. System Requirements

Most of the safety considerations are covered in the Formula Hybrid rules handbook. The final system should adhere to these electrical safety requirements as much as possible when handling the overall system. High voltage equipment will be physically inaccessible when charged, and previously implemented safety interlocks for powering down the system before maintenance will continue to be used. High voltage elements are isolated from the low voltage system through DC/DC converters.

3.1 System Status Interface

LCD Screen: The LCD screen is used to clearly communicate all errors and status/conditions of the vehicle. Live data such as vehicle speed and accumulator voltage is supplied through the LCD. The LCD is also designed to display transition states when flipping the switches on the vehicle, such that one can tell if the car is in neutral, waiting, charging, or other states.

RF transmitter: The RF transmitter uses UART in its communication system, so proper circuitry and overall set-up is required. The motherboard must be able to connect and send messages through UART to the RF transmitter. The transceiver must be able to achieve a minimum distance of 0.25 miles (0.4km).b

3.2 CAN Bus

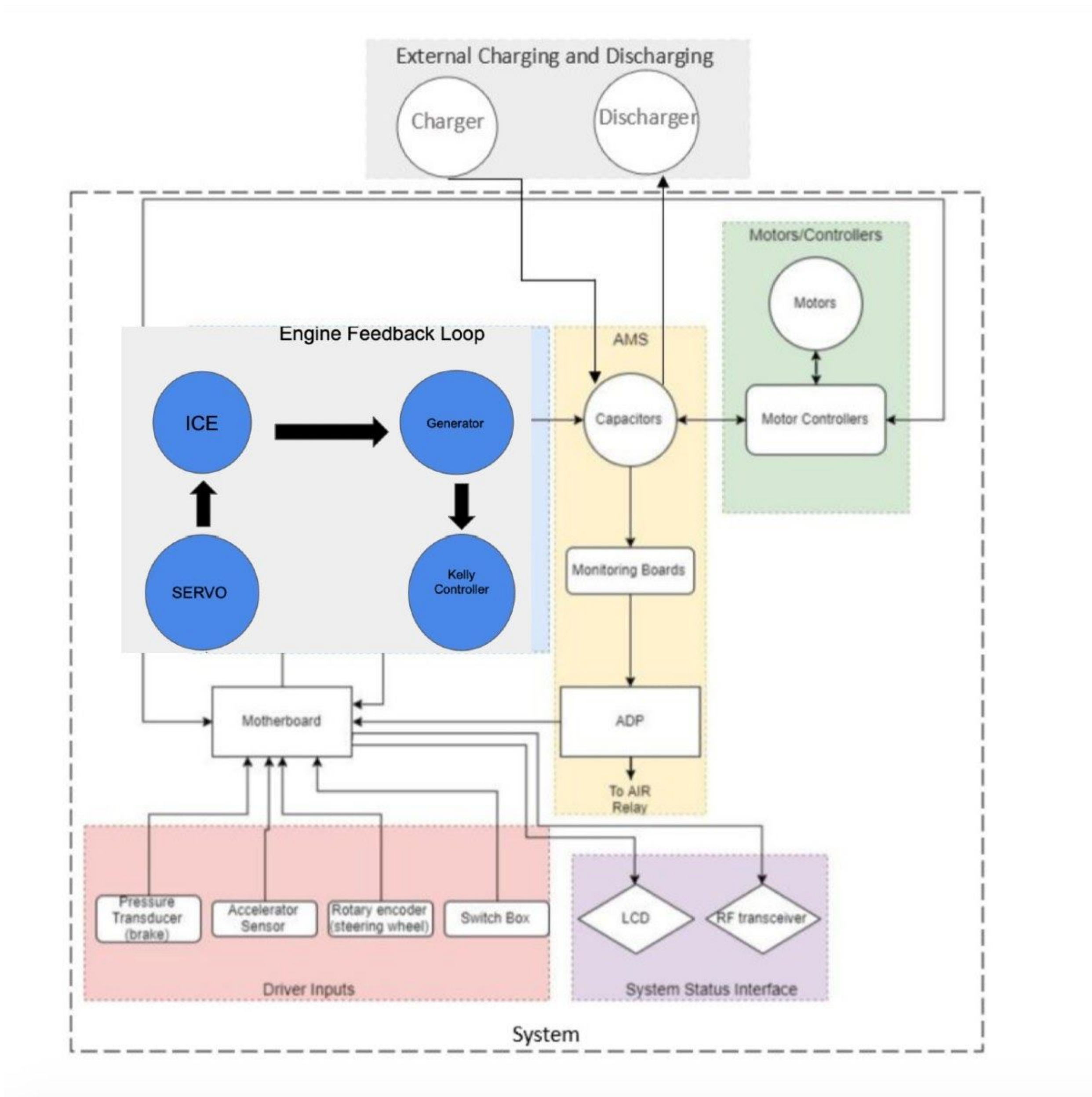
Our system must differentiate between the CAN messages from the left and right hub motors. The system must also be able to transmit the error messages from each individual motor to the driver and off track pit crew.

3.3 Engine Feedback Loop

The motor RPM will be stable through the expected load range. Large overshoot and stalling will not be acceptable. The engine feedback loop should be able to adjust the ICE throttle through a PID controller, so that the user can balance between efficiency and power by changing the RPM setpoint. The system must handle electrical noises when the vehicle is running.

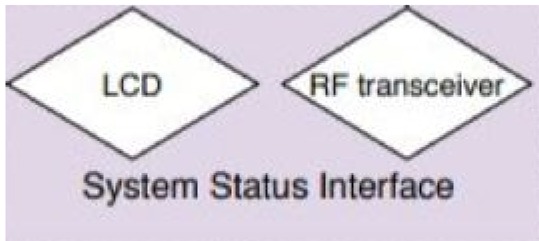
4. System Block Diagram

4.1 Overall System:

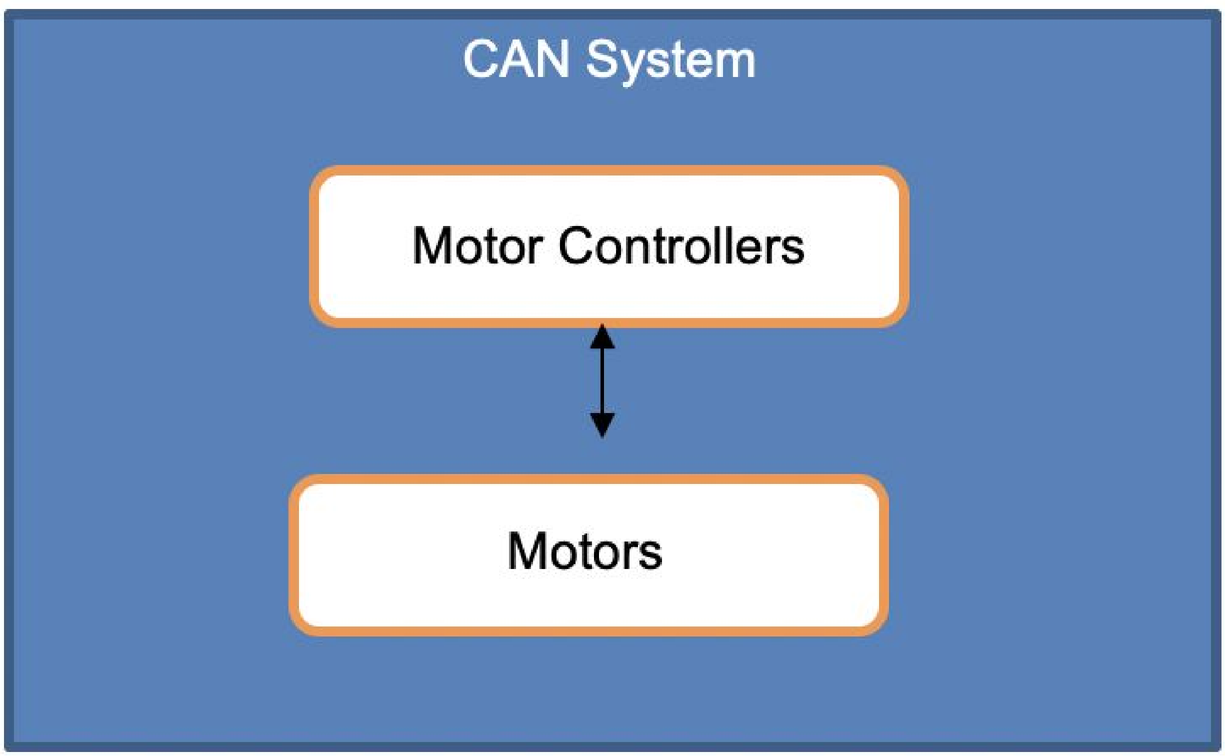


4.2 Subsystem and Interface Requirements:

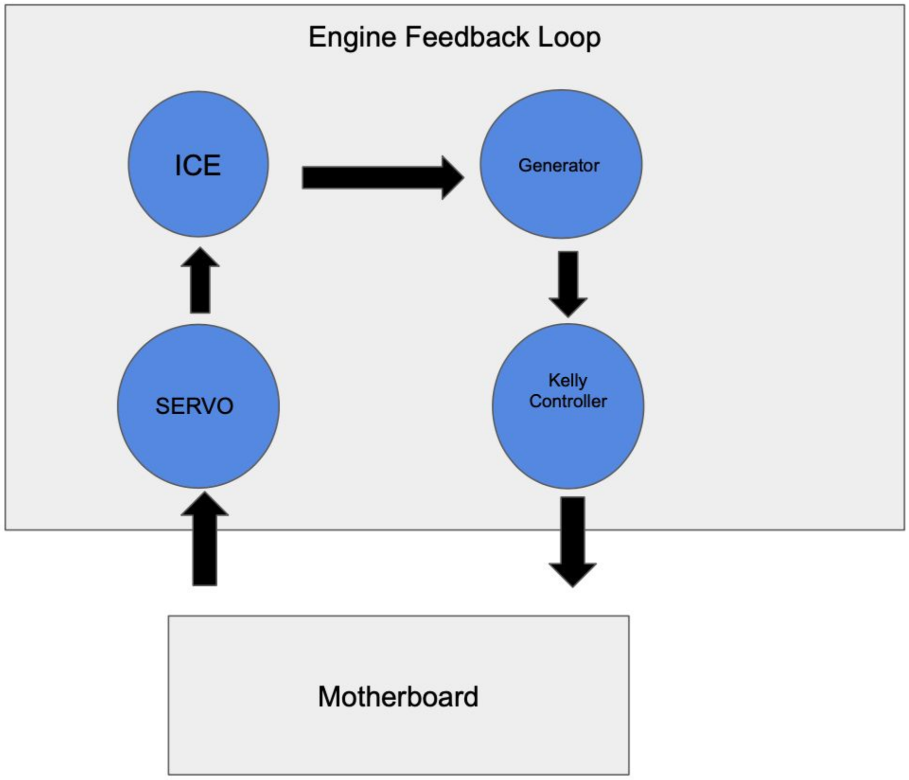
4.2.1 System Status Interface



4.2.2 CAN Drive



4.2.3 Engine Feedback Loop



4.3 Future Enhancement Requirements

There are a number of potential features we may add once the car's baseline functionality has been achieved. The first of which is an enhanced LCD display which would include additional information such as engine oil temperature, tire pressure, and a graphic for charging/ discharging status and rate. Additionally, while we hope to transmit data to the off-track team, one possible enhancement is allowing the off-track to write data to the car, essentially allowing them to take control of the engine operating parameters and charging/discharging characteristics.

5. High Level Design Decisions

5.1 System Status Interface

The communication between the vehicle and the off-track team is achieved through RF communications. The transmission protocol is achieved through either UART or SPI. The vehicle should have a LCD display to show information about the car. We decided to add more features to the display and fix the brightness issue. A 32-bit microcontroller will be sufficient for embedded systems on the vehicle. We also plan to use Matlab for the GUI that monitors real-time data transmitted from the vehicle.

5.2 CAN Bus Addressing

In order to ensure unique addressing for each of the Kelly Controllers, the continued use of the PIC32MX795 device is optimal. This microprocessor has 2 separate CAN bus inputs which would allow the implementation of 2 CAN buses on the system if the Kelly KLS Configuration Program method was determined to be faulty. An identical additional CAN transceiver would be necessary on the motherboard to provide this functionality. Since the Kelly controllers themselves are very costly, replacing the controllers solely because of this issue would not be feasible when workarounds are possible.

5.3 Engine Feedback Algorithm

More advanced engine control algorithms are needed to allow for operation in different racing environments. These will be implemented using the same PID controller that is currently on the system. However, information about the energy remaining in the buffer will be included before the control loop. This added information will allow the development of an algorithm that will prevent the capacitors from overcharging and prevent depletion.

6. Open Questions

The engine feedback loop is the biggest open question for our team. We plan to do the RPM speed sensor in a completely different way than previous years as indicated in our subsystem design document. Given the new approach we plan to take, previous teams have no testing experience or data to provide.

7. Major Component Costs

PCB Board(s) Fabrication: ~ $50

Quad/Dual Supercapacitor Auto Balancing (SAB™) MOSFET Array: (if not purchased last year)

- $3.47 each
- Qty: 15
- Total component cost: $52.05

Programming Cable USB to RS232 with SM-4A Adapter: $25

8. Conclusions

This high level design provides a background of the project from previous years and will serve as our upgrade and re-design game plan. We have begun reading through and understanding previous years documentation and we will be able to address problems within the system. The ultimate goal this year is to get a working Formula Hybrid Race Car, as the team is planning on competing in the spring.

9. References

Our Preliminary Design Proposal: (uploaded on Sakai)

SAE Guidelines:
https://formula-hybrid.org/wp-content/uploads/2020-Formula-Hybrid-Rules-Rev-1-1.pdf

Previous Team's High-Level Design:
http://seniordesign.ee.nd.edu/2020/Design%20Teams/formula/High%20Level%20Design.pdf

Team's Final Report:
http://seniordesign.ee.nd.edu/2020/Design%20Teams/formula/Final%20Report.pdf

Kelly Controller Datasheet:
https://kellycontroller.com/wp-content/uploads/kls-8080i-ips/KLS8080I-IPS-Opto-isolated-Sinusoidal-BLDC-V1.10.pdf

Kelly Broadcast CAN Protocol:
https://kellycontroller.com/wp-content/uploads/kls-8080i-ips/Sinusoidal-Wave-ControllerKLS-D-8080I-8080IPS-Broadcast-CAN-Protocol.pdf

PIC32MX795 Datasheet:
https://www.sparkfun.com/datasheets/Components/SMD/PIC32MX.pdf