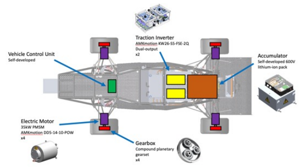
The University of Akron
College of Engineering and Polymer Science

ZIPS RACING – VEHICLE CONTROL UNIT

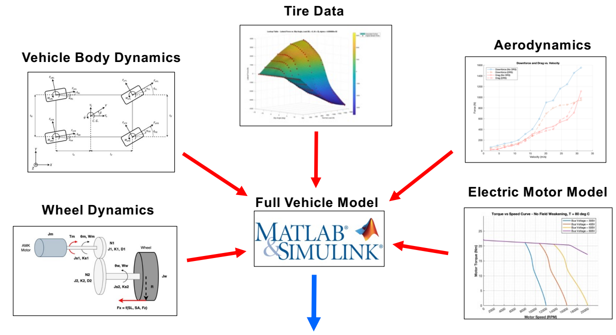
DT10 - Tetra Engdahl, Brian Glen, Ryan Stoller, John Wozencraft

Project Background

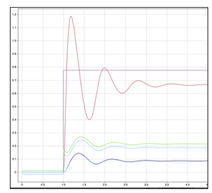
Zips Racing designs and builds an electric formula-style race car to compete against other universities in the Formula SAE challenge every year. This project's objective is to design a vehicle control unit to control a new four-wheel drive electric powertrain and improve vehicle performance by implementing torque-vectoring and regenerative braking algorithms, while ensuring safe and reliable operation of the vehicle.



Model Development



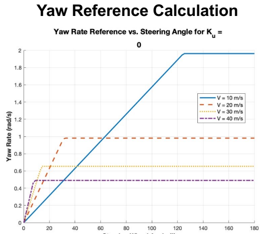
A non-linear vehicle model was made in MATLAB Simulink to analyze the transient cornering response of the vehicle, before it is built. The model is based on the double-track vehicle dynamics model



The model takes in (accelerator, brake, and steering inputs, allowing simulation of different driving maneuvers. Steady-state cornering, step-steer, and sine-steer maneuvers were prioritized.

Yaw Controller Design

Yaw Reference Calculation



Goal

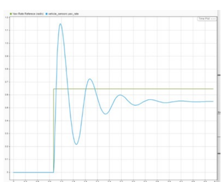
Control the yaw-rate to achieve a desired understeer behavior for the vehicle, by actuating the yaw moment. By increasing the steady-state yaw rate, the vehicle can achieve a higher cornering speed.

Method

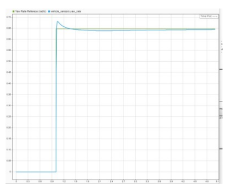
PID controller with gain scheduling. Non-linear model is tuned using a gradient-descent method by analyzing the step response at different operating points.

Target Response

10% overshoot, zero steady-state error, with the fastest rise time.

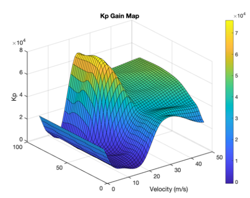


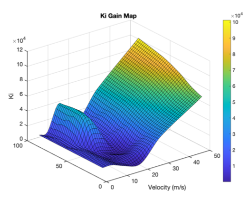
Natural Response Sample



Tuned Response Sample

Results:





Torque Optimization

Goal

Calculate all four motor torques in real-time to meet objectives.

$$\textit{minimize} \quad w_1\frac{f_1}{a_1} + w_2\frac{f_2}{a_2} + w_3\frac{f_3}{a_3}$$

$$\textit{subject to} \quad l_i \le A_i \le u_i$$

Method

Multi-objective constrained optimization was chosen. Lookup tables for vehicle data are linearized around the vehicle's operating point in real-time. An embedded quadratic programming solver, **CVXGEN**, is used to solve a matrix form of the problem to find the best solutions for each motor's torque command.

Decision Variables: $T = \begin{bmatrix} \tau_{FL} \\ \tau_{FR} \\ \tau_{RL} \\ \tau_{RR} \end{bmatrix}$

Objective functions

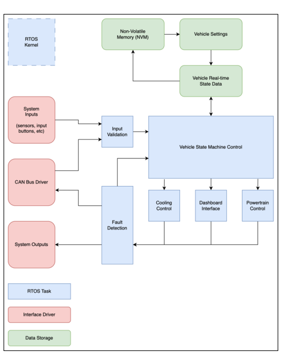
Functions of the motor torques

Achieve yaw controller output	$f_1 = (M_{z,tv} - M_z)^2$
Achieve driver longitudinal acceleration target	$f_2 = (a_{x,ref} - a_x)^2$
Minimize slip ratios to prevent wheel spin	$f_3 = SL_{FL}^2 + SL_{FR}^2 + SL_{RL}^2 + SL_{RR}^2$

Constraints

Individual Motor Torque Limit	$\tau_{min,i} \le \tau_i \le \tau_{max,i}$ for $i \in \{FL, FR, RL, RR\}$
Battery Power Limit	$P_{min} \le \frac{1}{\eta_{batt} + \eta_{inverter}} \sum_{j=0}^{4} \frac{\tau_j \omega_j}{\eta_j} \le P_{max}$
Total Torque Limit	$\tau_{min} \le \sum_{j=0}^{4} \tau_j \le \tau_{max}$
Slip Ratio Magnitude Limit	$SL_{min,i} \le SL_i \le SL_{max,i}$ for $i \in \{FL, FR, RL, RR\}$

Firmware Development



Real-time Operating System

The firmware uses freeRTOS to schedule multiple tasks, ensuring the requirements of the control system are met. Some major tasks are:

CAN Communication Task – Handles communication between auxiliary vehicle devices.

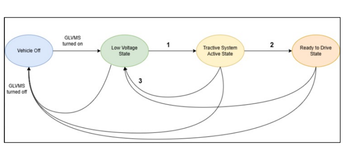
Fault Task – Checks for all vehicle faults and determines appropriate response.

Torque Control Task – Determines vehicle control mode and utilizes torque optimization algorithms to output individual motor torques.

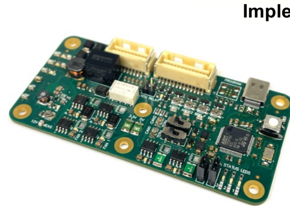
Vehicle Data Task – All tasks send data to this task, storing the current vehicle state. Tasks interpret this state to determine proper functionality.

Vehicle State Machine Control

The vehicle control unit runs through a central finite state machine (FSM) that determines the running mode of the vehicle. All RTOS tasks communicate with the FSM so that it can determine the state and allowable functionality of the vehicle.



Implementation

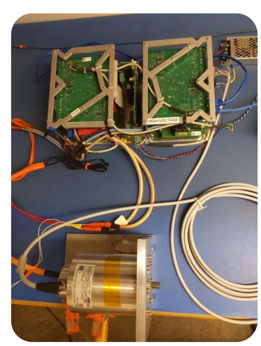


Vehicle Control Unit Hardware

- STM32F405 Microcontroller
- CAN bus for vehicle communication
- Analog / Digital sensor inputs
- Powertrain safety interlock control

Hardware-in-the-Loop Testing (HIL)

To validate the embedded implementation of the control system, a HIL test was setup. The Simulink simulation sends driver inputs and sensor data to the hardware via a python CAN interface. The device under test calculates the control system output using the simulation time and updates the vehicle model.



Powertrain Bench Testing

The VCU was bench tested with the high voltage powertrain to develop the firmware and validate safety fault logic in a controlled environment. Driver inputs and dash controls were also mocked up to validate proper functionality.

• 2024-2025 Senior Design Capstone Project • Dr. Farhina Haque, Faculty Advisor • Dr. Michael Lichter, Senior Design Coordinator • Department of Electrical and Computer Engineering • College of Engineering • University of Akron •