EECS 373 F19 Final Project Proposal

Justin Beemer, Michael Reber, Adithya Subbiah, Karthik Urs

UPDATED: 10/24

PROBLEM

Michigan Mars Rover (MRover for short) is a student-run robotics team housed in the Wilson Student Team Project Center. Every year, we build a new 50kg off-road robot to compete in the University Rover Challenge. The competition takes place in the southern Utah desert, where rovers complete tasks similar to those a real Mars Rover may have to on an astronaut-assist mission. The competition demands a variety of functions including off-road mobility and robotic manipulation, while enforcing an $18,000 spending limit for on-rover hardware. The field is also highly competitive and made up of teams from around the world.

In order to continue improving our competition performance, we are driven to create designs with increasing performance on a monetary and mass constraint. For that reason, we are attempting to change our traditionally permanent magnet brushed DC (BDC) actuator motors to permanent magnet brushless DC (BLDC) to take advantage of higher power to mass and power to dollar ratios. In order to enable this change and continue to support BDC motors, we are proposing a custom dual motor controller than can drive both BDC and BLDC motors on the same hardware.

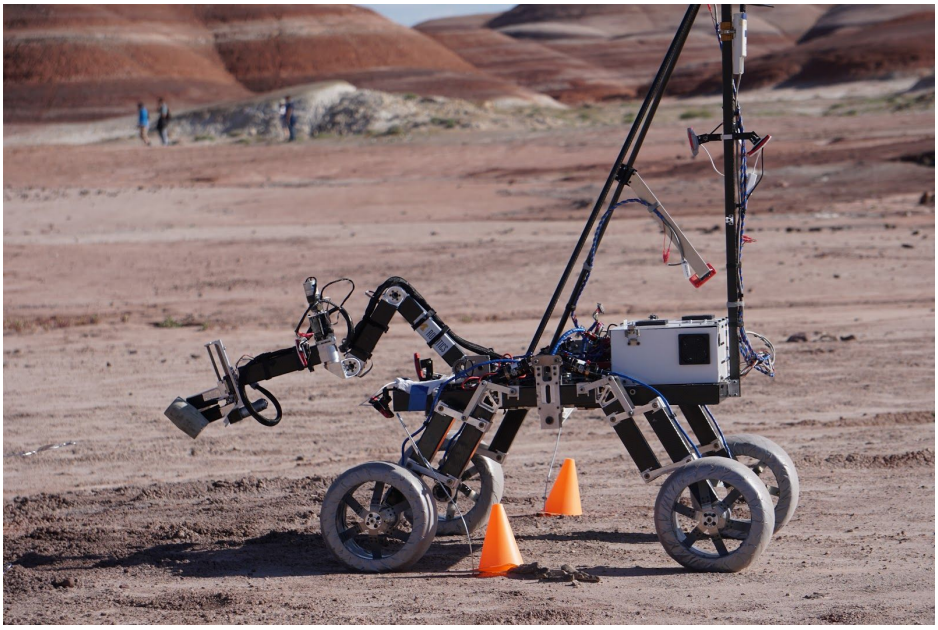

Figure. MRover's 2019 rover, "Bowser," at competition in Utah with the robotic arm equipped.

TECHNICAL BACKGROUND

Permanent magnet DC motors are an extremely common type of motor. Motion in these motors is caused by an electromagnetic force attempting to align two magnetic fields -- one created by the permanent magnet, and one created by electrical current. Here is some relevant background on these types of motors.

Commutation and Control

In order to keep the motor spinning, the two field vectors must be kept at an angular displacement relative to each other in a process called Commutation. In a BDC, commutation is handled by the commutator, which is a

piece of mechanical hardware that automatically connects the windings on the rotor to current sources in such a way that the rotor's field vector is behind that of the stator's permanent magnets. The speed of a BDC is simply determined by the voltage level driving the current through the windings.

In a BLDC, the commutation process must be executed by an electronic controller, as the permanent magnets are on the rotor and the windings are on the stator. Specifically, the windings on the stator must create a net field vector ("current vector") that stays in advance of the rotor's permanent magnet field ("shaft vector" or "rotor flux"). There are many ways of controlling the commutation process, but we will focus on two methods: Trapezoidal Control and Field-Oriented Control (FOC).

Trapezoidal Control

Trapezoidal control is a simple way of handling commutation for a BLDC. It relies on three sensors to inform the position of the rotor, and selectively turns on or off coils to create a field vector in advance of the shaft. It is easy to implement and computationally lightweight, and many motors come with the three sensors (Hall Effect sensors) pre-installed. However, the three sensors are digital and can only practically encode six states, meaning that the current vector must lie in one of six evenly spaced angles. The optimal displacement between the two field vectors is 90 degrees, but with trapezoidal control, the displacement will vary from 30 to 120 degrees over the course of one state. This lack of precision in commutation control results in torque ripple and a loss of performance compared to the ideal.

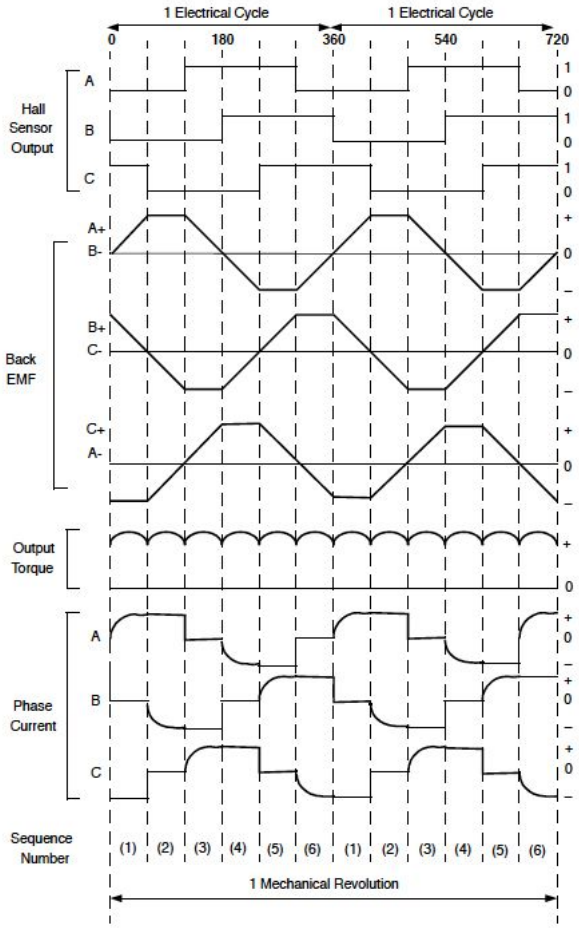

Figure. Commutation sequence and waveforms with Trapezoidal Control.
https://www.motioncontroltips.com/faq-trapezoidal-back-emf/

Field-Oriented Control

Field-oriented control utilizes better sensing and uses more computation in order to keep the field vector as close to 90 degrees ahead of the shaft vector as possible. Specifically, there are dedicated sensors to detect the shaft position and current flows through the windings with high resolution. Using those sensors and some math, precise knowledge of the displacement between the shaft vector (a.k.a. rotor flux) and current vector can be known and controlled to be 90 degrees (in order to optimize torque output). This method typically results in less torque ripple and higher power output.

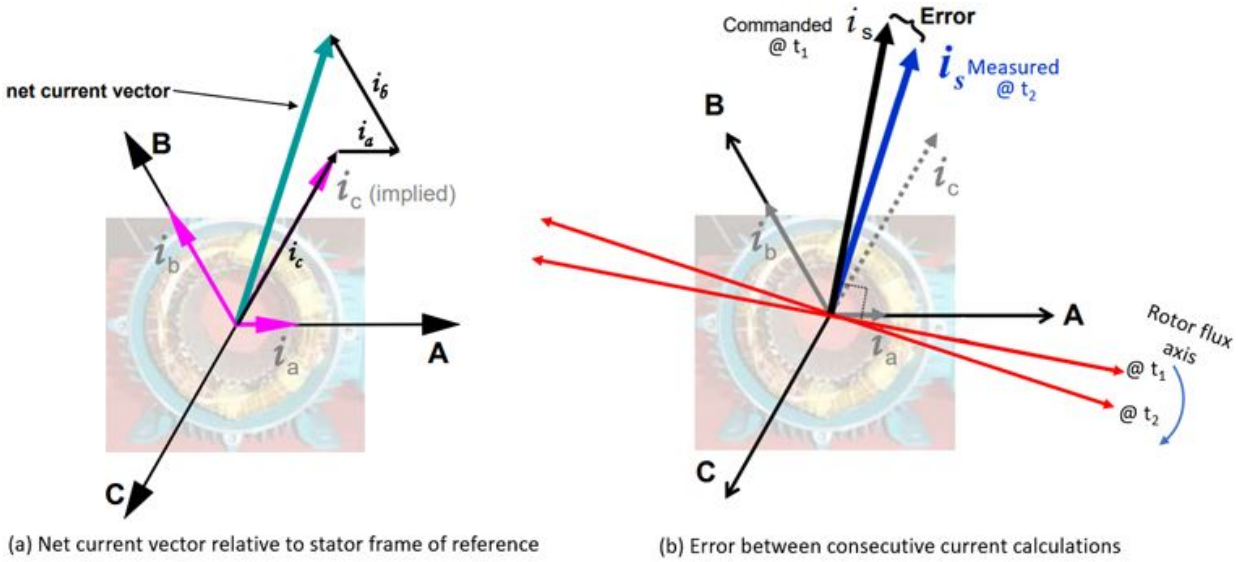

Figure. Key vectors associated with FOC.
https://www.electronicdesign.com/power/field-oriented-control-helps-create-tomorrow-s-cordless-power-tools

REQUIREMENTS

The motor controller must fulfill the following high-level requirements:

1. Electrical:
    a. The controller shall safely deliver up to 80A at 42V to a triplet-pole BLDC motor.
    b. The controller shall safely deliver up to 40A at 42V to a BDC motor.
    ~~c. The controller shall cause less than 3% voltage ripple on the DC bus.~~
    d. The controller shall be capable of interfacing with:
        i. 4x limit switches
        ii. 4x analog inputs
        iii. 2x digital quadrature encoder inputs
        iv. Digital command interface
            ~~1. Ethernet to onboard computer~~ I2C to central linux microcontroller
        v. Digital telemetry interface
            ~~1. Ethernet to onboard computer~~ I2C to central linux microcontroller
        vi. Gate Drivers via SPI
2. Physical:
    a. A set of 8+ controllers shall occupy less than 100 cm^3 per controller.

b. A set of 8+ controllers shall be less than 30g per controller.
c. The controller shall be capable of operating at full power for 30 seconds at 30 degC ambient temperature.

3. Controls:
   a. The controller shall be capable of driving a motor at an angular velocity to within +/- 1% command in typical rover operating conditions.
   b. The controller shall drive a motor with less than 5% torque ripple.
4. Budgetary:
   a. The total production cost shall be less than $60 US per controller, not including labor of students.

SCOPE AND SOLUTION

The solution to the posed problem is made of several main components, seen in the figure below and the subsections below:

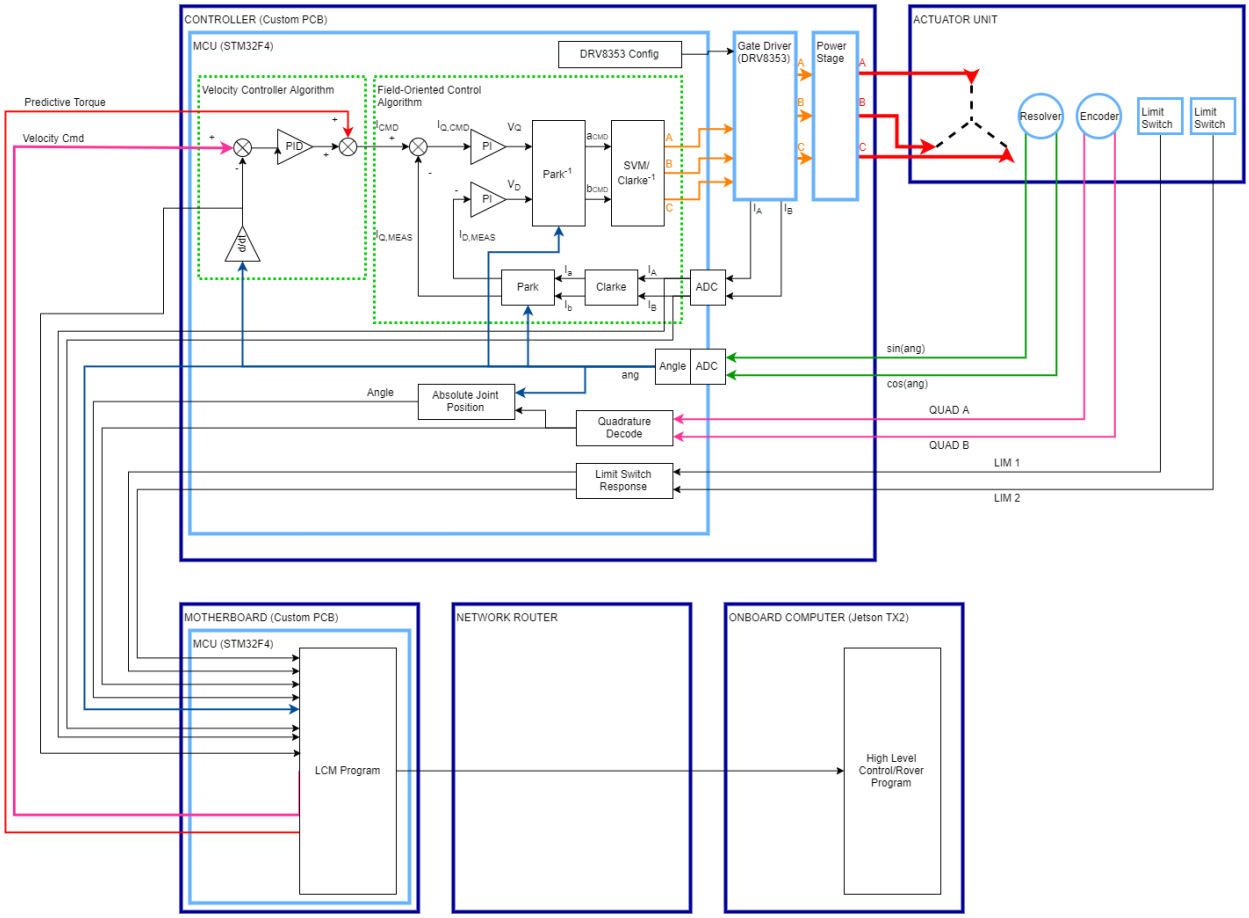

Figure. Technical block diagram of the proposed solution made up of the Actuator Unit, Controller, and Motherboard.

Actuator Unit

The Actuator Unit is responsible for creating generalized physical motion. It is made up of the following key components:

- Motor: BLDC or BDC
- Hall Effect Sensors, if pre-installed on BLDC motor
    - Used to detect rotational position of motor roto in order to achieve trapezoidal commutation control of the motor
- Resolver
    - Used to detect rotational position of motor rotor in order to achieve field-oriented control (FOC) of the motor
    - PN: MLX90380LGO-BAB-103-RE
- Encoder
    - Used to detect the rotation position of gearbox output shaft to enable position and velocity controls
- Limit Switches
    - Used to signal approaching mechanical self-collision

The components listed above are controlled from the Controller.

Controller

The Controller is responsible for meeting the bulk of the presented requirements. As a whole, it receives commands from the Motherboard; sends power to the Actuator Unit; reads data from the sensors located on the Actuator Unit; and sends telemetry back to the Motherboard. The Controller is made up of the following key components:

- Microcontroller:
    - Interprets commands from the Motherboard via I2C
        - Velocity Command
        - Predictive Torque feedforward control term
    - Runs a velocity regulation control algorithm
        - 10 kHz
        - Outputs a torque command
        - Uses velocity data from the Actuator Unit Resolver
        - Uses the Predictive Torque to enable faster torque response
    - Runs a motor control algorithm
        - In the case of BLDC:
            - ~~Run a Field-oriented control algorithm at 10kHz using the torque command from the velocity regulation algorithm~~
            - Passes a duty cycle to the Gate Driver for Trapezoidal control
            - OR handles commutation based off of Hall Effect sensors in software
            - OR runs an off-the-shelf field oriented control algorithm, adapted and configured for specific use
        - In the case of BDC: pass-through to Gate Driver
    - Reads data from the Actuator Unit
        - Reads Resolver signals through an ADC ~~for use in FOC and~~ as telemetry
        - Reads Encoder signals as telemetry
        - Reads Limit Switches to stop motion when necessary

    - Send telemetry back to the Motherboard at 1 kHz
    - PN: ~~STM32F469BIT6~~ STM32F303RET7
    - Will be using off-the-shelf nucleo development board
- Gate Driver
    - Receives commands from the Microcontroller
        - SPI for configuration and diagnostics
        - Trapezoidal commutation on DRV8353: (1x Mode)
            - Interprets PWM duty cycle from microcontroller to deter
            - Command proportional to motor current
            - Deadtime Implemented by DRV8353
        - Trapezoidal commutation on MCU: (6x Mode)
            - Hall effect commutation logic implemented on MCU
            - Gate Driver receives 6 PWM signals for the powerstage Fets
            - Deadtime implemented on MCU
        - Field Oriented Control: (x3 Mode)
            - Uses high resolution position sensing for vector current control
            - Sends 3 PWM signals to gate driver
            - DRV8353 handles deatime
    - Drives the gates of the MOSFETs in the Power Stage
    - Returns conditioned current-sense analog signal that represents the current running through a motor phase leg.
        - Read with ADC on microcontroller
    - PN: DRV8353RSRGZT
- Power Stage
    - Made up of 3x half-bridges using Silicon MOSFETs and additional gate protection circuitry
        - Deatime watchdog
        - Clamps
        - Desaturation
    - Driven by signals from the Gate Driver
    - Run at 36Vdc bus voltage
    - PN: CSD18563Q5A
        - Other PNs for protection circuitry TBD
    - Designing custom board with power stage

Motherboard

The Motherboard is responsible for communicating the existing Onboard Computer and distributing commands and reading telemetry from all the motor controllers on the rover. It's made up of one key component:

- Microcontroller
    - Runs Lightweight Communication and Marshalling (LCM) to communicate with the Rover Onboard Computer through Ethernet
    - Distributes commands from the Onboard Computer to all Controllers on the Rover.
    - ~~PN: STM32F469BIT6~~
    - BeagleBone Black

RELEVANCE TO EECS 373

In implementing the solution presented above, we will use almost all the material covered in EECS 373 in the following ways:

- GPIO:
    - Limit Switches
    - Quadrature Encoder Signals
- Interrupts:
    - Periodic control loops
        - Speed Controller
        - Torque Controllers
        - FSM/Fault Handling
        - Telemetry
    - Limit Switch triggers
- Timers:
    - Accurately timing a control loop
    - Quadrature mode for digital encoders
- Serial Communication:
    - Full-duplex I2C bus between motherboard (master) and controllers (slaves)
    - SPI from STM MCU to DRV8353R
- ADCs:
    - Resolver
    - Current sensing
- Embedded Systems Engineering:
    - Using commercially-available, industry standard components
    - Researching components and interpreting datasheets

DESIGN EXPOSITION DELIVERABLE

On December 12th, we will present the following at Design Expo:

- Motherboard
- 4x Controllers
- 4X Actuator Units
- Mechanical Structure

DE-SCOPE PLAN

In the event that executing the presented scope or meeting all the requirements of the project is too difficult, we have a plan to de-scope certain elements. Here is the order in which the de-scopes would occur is necessary:

1. Loosen power, torque ripple, or voltage ripple requirements
2. Program the Motherboard ethernet connectivity using mbed OS

3. Abandon FOC and only do trapezoidal BLDC control and BDC control; focus on custom ethernet connectivity on Motherboard

REFERENCES

ODrive Robotics
H-Bridge Secrets
DRV8353RS Three-Phase Smart Gate Driver Evaluation Module
Field Oriented Control of Permanent Magnet Motors