B.O.A.T

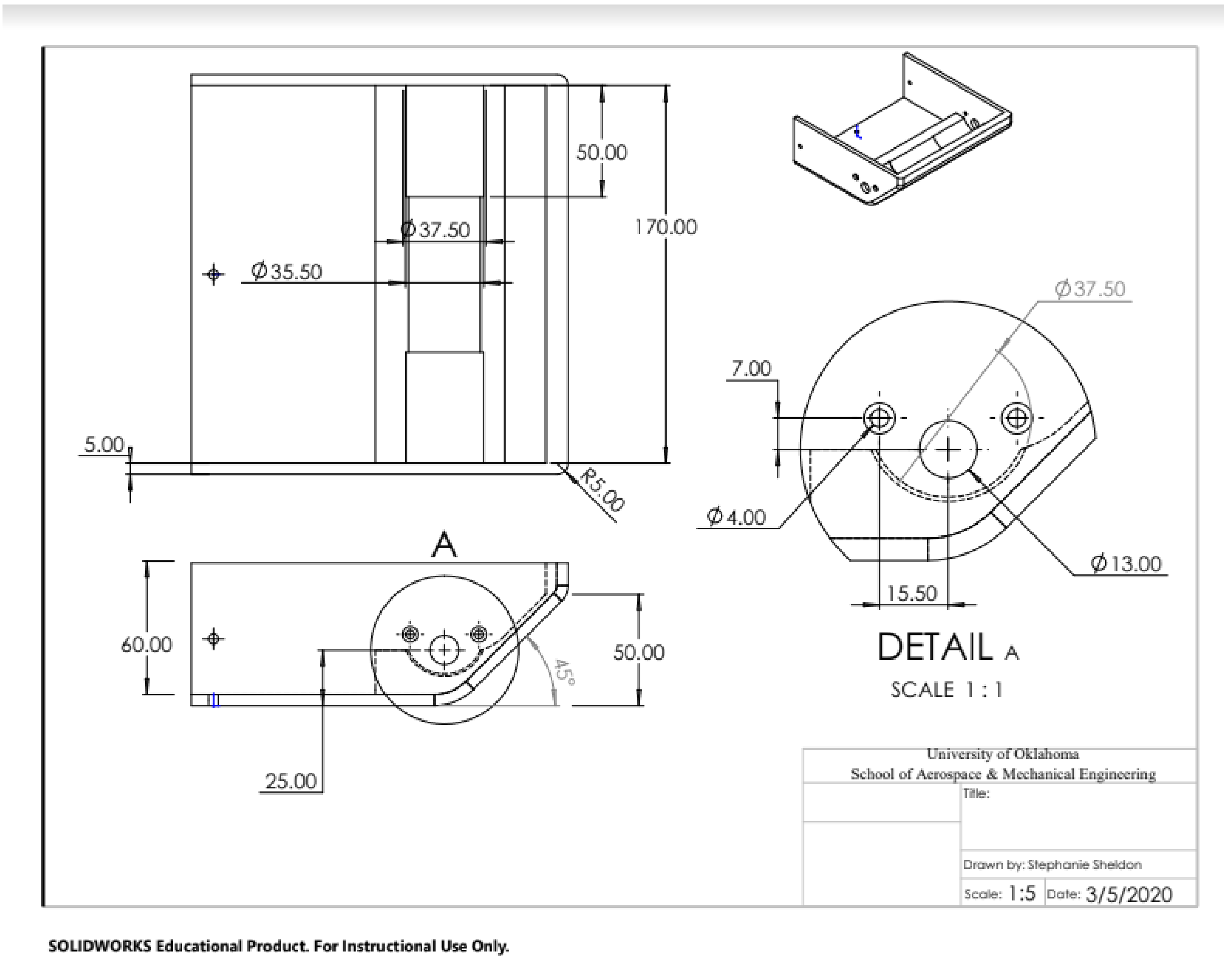
Mercury Challenge 2020 Design Report
Sooner Competitive Robotics
The University of Oklahoma

Team Captain: Sarah Brown
Vice Captain: Stephanie Sheldon
Electrical Lead: Jorge Exinia
Mechanical Lead: Stephanie Sheldon
Software Lead: Cody Black

Table of Contents

1. Mechanical Design

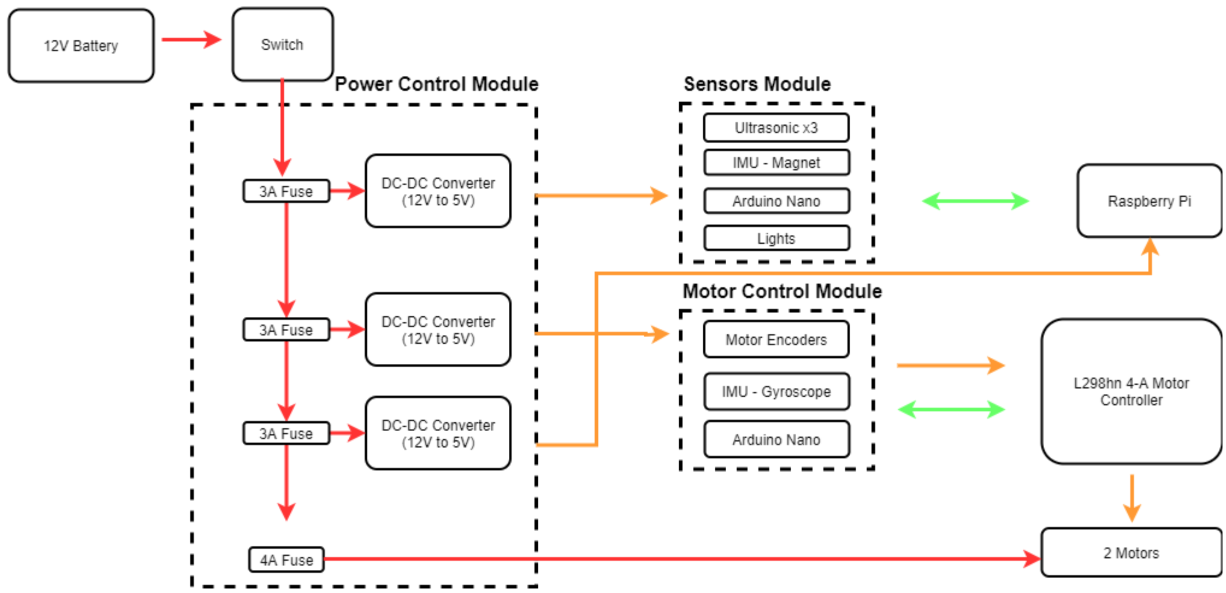
1.1 Mechanical Block Diagram



Mercury mechanical design originally focused on a tread based drive and a two layered robot for sufficient storage for electrical components. This design has hence shifted to focus on a chain drive approach featuring six wheels (three per side) interconnected with different chains. These chains are in turn connected to the two motors. This design allows for control of movement in addition to less stress placed on the motors due to the excessive traction the treads created.

2. Electrical Design

2.1 Electrical Block Diagram



2.2 Controller Systems

The B.O.A.T's control system relies on various sensors to maneuver the robot through the course. The first set of sensors are all connected to a single microprocessor that is mounted on to a printed circuit board. There are three ultrasonic sensors connected to this microprocessor that give data about B.O.A.T's location. Also on this PCB is an inertial measurement unit that we utilize for its magnatomenter. This magnetometer is used to sense the magnetic field of the block. The final thing on this arduino is a series of lights that we use for communication between parties.

The second microprocessor is used to control the motors and control an inertial measurement unit. This microprocessor controls a motor controller that is connected to motors and controls the movement. The motors are also equipped with encoders that feed their data into the microprocessor. The IMU connected to this microprocessor also feeds data about the movement of B.O.A.T. Both of the microprocessors are connected to a Raspberry Pi.

2.3 Power System

The entirety of B.O.A.T is powered on a 12 volt battery but the power is distributed evenly to the needed systems via 3 buck converters that step down the voltage to 5 volts. The 3 buck converters are mounted on a printed circuit board that have trace widths that manage the current draw of all the various sensors, motors, or microprocessors. The then stepped down voltage is connected to separate printed circuit boards used to connect and power the subsystems. The first subsystem powers a microcontroller and a 5 volt rail on the respective printed circuit board. The sensors on this subsystem are powered on of the 3.3 volt pin on the microcontroller. The second subsystem powers another microcontroller and another 5 volt rail. The motors controller and encoders are powered off of the 5 volt rail while other sensors in said system are powered off the microcontroller's 3.3 volt pins. The final subsystem is simply 5 volts to our raspberry pi. The motors in the are powered directly off of the 12 volts of the battery.

2.4 Safety System

For safety purposes, the electrical systems are all attached with fuses in appropriate areas. These fuses help prevent excessive current and voltage draws, preventing damage to any electrical components. Additionally, all wires are attached with secure connectors to help prevent any wires from fraying.

3. Software Design

3.1 Operator-B.O.A.T Communication System

The communication system between B.O.A.T and the Operator consists of a few subsystems. At the base of it all is the VPN subsystem that ensures the connection between B.O.A.T and the Operator is network independent. This subsystem consists of an OpenVPN server running on a cloud server. B.O.A.T will connect to this VPN on system boot and the Operator will manually connect to the server. By using this system, B.O.A.T and the Operator's interface will be able to communicate as if they were connected via a LAN connection.

The video feedback subsystem uses Motion, a FOSS video streaming program that runs on B.O.A.T's Raspberry Pi. The program runs a simple web server along with an RTP stream of B.O.A.T's camera. Using the VPN service allows the Operator to simply then view the stream by connecting to B.O.A.T's VPN address in their browser.

The control system uses Python's standard socket library to deliver control data from the Operator to the B.O.A.T over TCP. We choose TCP for reliability and to ensure that instantaneous events such as button presses do not get lost as they might over UDP.

The Operator's interface hosts the TCP server which B.O.A.T connects to. This was chosen so that when B.O.A.T loses power or internet connectivity it can simply reconnect to the server. When B.O.A.T receives a control packet from the Operator, it decodes the packet and sends the appropriate control instructions to the appropriate Arduino over USB. Conversely, B.O.A.T will send feedback to the Operator such as information regarding magnetometer readings. This is accomplished in the same way as the control system by having B.O.A.T send a TCP packet to the Operator's interface which decodes the packet and displays necessary information to the Operator.

3.2 Autonomous System

B.O.A.T's autonomous system is designed to be able to complete the known autonomous course. The system works by having a state machine that keeps track of the robot's current location within the course. The state machine is updated based on B.O.A.T's encoders which predict the robot's location. Three ultrasonic ranging sensors are then used to verify that we are in the correct location by matching where the wall's should be and where the sensors say B.O.A.T is. The state machine will instruct how B.O.A.T should move.

4. Bill of Materials

4.1 Parts list/Bill of materials

Raspberry Pi	x1
Arduino	x2
Ultrasonic Sensors	x3
IMU Chips	x2
Battery	x1
Webcam	x1
Motors	x2
Chassis	x1