Fort Collins Football Club
(FC^2) 2018 Team Description

Dr. Hamid Reza Chitsaz, Avpreet Singh Othee, Erik Holmgren, James Dix, Dan Luckner, Annemarie Kibbe, Ryan Jensen, Jiaxin (Jack) Zhao, Michael McMann, Cameron Ghia, Mark Stratford

Colorado State University, Colorado / USA

Abstract

Fort Collins Football Club is a newly formed team based out of Colorado State University, working to participate in the RoboCup small size league. Intrigued by the engineering challenges involved in designing both a physical robot and algorithms for the complex task of soccer, Fort Collins Football Club is determined to contribute to the of advanced decision making and control algorithms.

1 Intro

Fort Collins Football Club (FC2) is based out of Colorado State University in Fort Collins, Colorado, USA. Our team consists primarily of students at both the graduate and undergraduate level. Our robot design is based off of the work of other teams, but with our own modifications. We use a variety of manufacturing techniques, including machining and 3D printing, to produce the robots. We are writing our own firmware and agent software. Control software will utilize an application of Partially Observable Markov Decision Processes (POMDPs) to perform decision making processes and facilitate machine learning. Major challenges of FC2's effort include designing and building our robots without a completed design from a previous year, designing a control interface and pipeline around a server-client-robot architecture, and the application of POMDPs for complex decision making, including definitions of actions, states, and processing. Fort Collins Football Club is supervised by Dr. Hamidreza Chitsaz, who is an Assistant Professor in the Computer Science department. Even though the Fort Collins Football Clubs supervisor, Dr. Chitsaz, won the RoboCup world championship in the middle size league in 1999 and world third place in 2000 as part of SharifCE team, this is Fort Collins Football Clubs first entry into the RoboCup competition.

2 Electrical Design

The electrical design includes the on-board computation unit, motor controllers, power management circuit and solenoid actuation circuit. The next sections describe the implementation and function of each subsystem.

Initial design of the computing platform on the robots had a single board computer (Raspberry Pi), connected to a separate controller. The design was selected keeping in mind several factors. These included future upgrades, use of Wi-Fi as a wireless communications interface, high processing power of available on the raspberry pi. This year's design was redone from ground up. We were given some valuable suggestions in the peer review by other teams. One of the suggestion was to not use the Wi-Fi as the wireless communications interface. Wi-Fi has a high data throughput but is not robust, in the sense that connectivity is not guaranteed. In a crowded area with a lot of devices, there might be connectivity issues. This scenario is highly likely at the actual competition venue. During the development process for this year's design, it was noticed that the raspberry pi was redundant and the ARM microcontroller was sufficient for implementation of all the necessary control algorithms and other functions. The raspberry pi was also acting as a Wi-Fi receiver. As mentioned above, the Wi-Fi was dropped as the wireless interface, thus rendering the raspberry pi as a redundant add-on. In the newer latest design, the computation unit is an ST microelectronics ARM with floating point unit (FPU). A dedicated FPU has many advantages such as decreased computation time in performing floating point calculations. Microcontroller without a dedicated FPU performs the floating point calculations using ABI that might take up to hundreds of clock cycles. FPU allows implementation of digital signal processing algorithms such as digital filters, controllers, etc. We are using ARM-GCC toolchain [ref] to compile the machine code. First implementation of the controller firmware used "std-peripheral" library supplied by ST. The firmware was eventually moved to the Hardware Abstraction Layer (HAL) library. The migration was necessitated because of simpler interface, and compatibility with other libraries from various vendors. HAL is also CMSIS compatible.

Adhering to suggestions and improvements proposed by reviewers for the last year's paper, the Wi-Fi was dropped and the design was switched with Nordic Semiconductor's nRF24 based wireless module. It has a moderate data rate of 1Mbps. It has the addressing feature that allows user to set custom address to the transmitter and receiver. The transceiver implements a five byte addressing mode. This prevents crosstalk and any unwanted interference. The nRF has a low power consumption, thus increasing operating time of the robot when powered using a battery. nRF24 is interfaced using SPI. Robots have an onboard micro-controller. The nRF module on the robot, acting as a receiver for most time, is directly interfaced with the MCU's SPI peripheral. nRF24 communication is implemented on a modified nRF24 library ported for use on the STmicro MCU. The link for the original nRF24 library can be found in [link]. The nRF24 on the robot acts a transceiver transmitting telemetry and diagnostics data back to the server. The server side nRF24 module is interfaced using a USB to SPI device controller. The interface controller is FT4222, manufactured by FTDI. The nRF24 library is ported to be used with the USB-SPI device controller.

The team has been following a two step design methodology. The first step involves design and implementation of a set of six robots manufactured using rapid-prototyping technologies. In the second step, a different set of robots are designed. These robots go through extensive design iterations and testing. The goal is to use the use the robots designed in the first step for feasibility studies and rapid implementation and rigorous testing. The robots designed in step two are would be the ones used in the actual competition.

The first generation robots are put together using low cost components that are easy to procure and implement. These robots are driven using four permanent magnet direct current (PMDC) motors. These are low cost motors that can be easily controlled using low cost H-bridge motor drivers. The robots are driven using four motors driving four omni-directional wheels. The wheel arrangement is standard. The micro-controller is interfaced with the motor drivers and motor encoders using a custom developed printed circuit board. The robot chassis are manufactured using rapid prototyping technique. We used 3-D printers to fabricate the chassis. The second generation robots are designed and fabricated to be long lasting and highly robust. The robots are driven using brushless direct current (BLDC) motors. BLDC motors are synchronous motors that are driven by a three-phase bridge driver. BLDC motors have several features that make them superior to PMDC motors. Some of these features are, higher torque, higher efficiency etc. These advantages come at a cost of higher complexity in their implementation control. Figure below shows a prototype board for BLDC motor driver from STmicro, L6229PD. The team has procured the motors, and is in the process of designing circuit to house the motor drivers. The robots are controlled using a two layered control

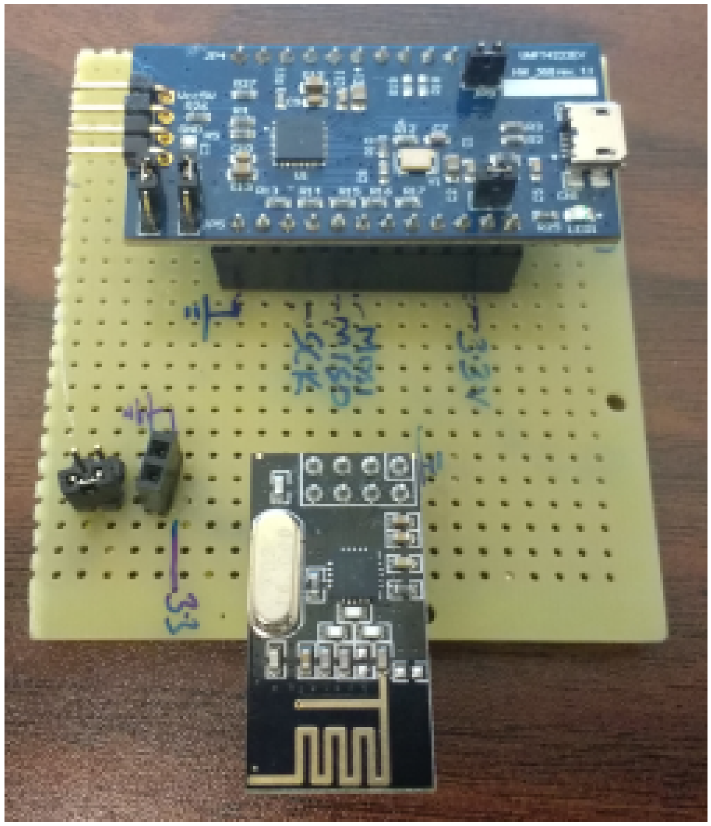

Figure 1: USB-SPI interface, connects the nRF module with server

strategy. A low level control implemented on the onboard MCU controls the motor speed. The motor speed is determined by the transformation from the robot velocity components to the individual motor velocities. The control algorithm senses the motor speed using encoders. The controller is implemented using a PID algorithm. The level two, or higher level control is associated with the overall robot control. That is achieved using the ssl-vision software. Robot path trajectories generated by the AI are converted into position data. This position data is then used to generated velocity profiles using interpolation. Team has been experimenting with several interpolation techniques.

The second generation robots implement solenoids and dribbler motor. It allows the robot to manipulate the ball. Solenoid mechanical design is explained in the following sections. The solenoid is driven using a high voltage capacitor. The solenoid design was streamlined using a simple calculator for computing different values for the solenoid [ref]. A designer must balance various aspects to achieve desired performance. The calculator allows designer to tune various features, thus, helping to streamline the design process. The design requires the solenoid to be operated at voltages higher than 200V. The solenoid is driven by a high voltage capacitor that acts as a power source. Capacitor can discharge a large amount of current is short time span. The robots are powered by a rechargeable lithium-ion battery. To charge the capacitor to a high voltage, a step-up converter is required. Figure below shows a step-up converter simulation implemented using MC34063 DC-DC converter. The converter is powered by the onboard battery. The step-up converter is setup as a non-isolated boost converter.

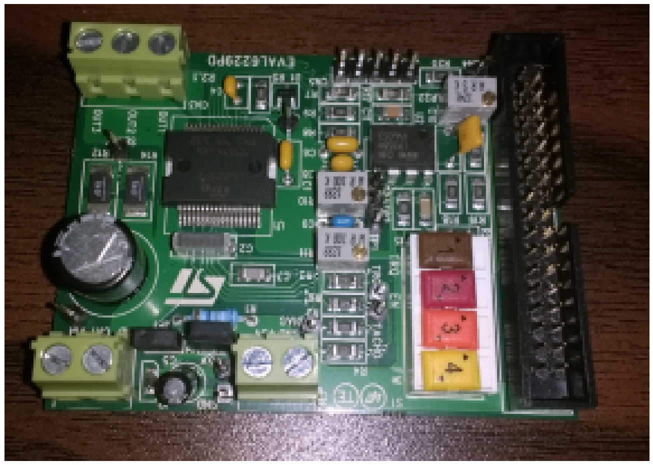

Figure 2: BLDC driver

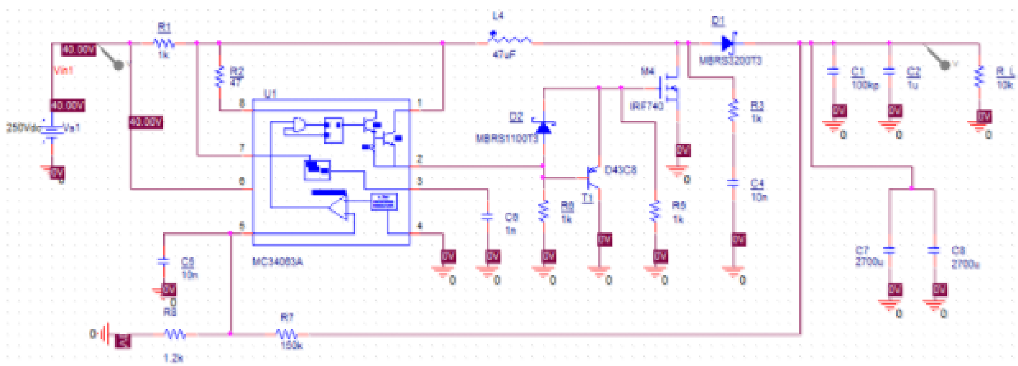

Figure 3: Orcad Capacitor Charging Simulation Schematic

3 Mechanical Design

This is the second time that we are submitting an application to qualify for the RoboCup competition. Compared to the first Fort Collins Football Club robot model, the second generation has multiple changes and re-designs. First, the material choices were reconsidered. The previous robot model utilizes 3D-printing tech, and the whole robot body is printed from ABS. However, this produces multiple issues such as the durability of the structure, the thickness of each components, and the shear stress that produced from the printed layers. In the second generation, the ABS main body was replaced by an aluminum alloy. Only motor mounts were printed from ABS. This re-design improves the durability and decreases the thickness. Also, manufactured aluminum alloy has better shear stress resistance than printed ABS. Second, the kicker-chipper system was redesigned. The previous model did not have an effective way to produce a good kicking and chipping force and the solenoid system that was used was not compact enough to fit into the robot body. In the new design, a dual round solenoid system was used to produce the desired kicking and chipping forces. This system utilizes an offset stacking position and the linkage system to provide a reliable, compact design. Third, the motor-wheel mounting system was improved. In the previous model, the motor the was used was the Pololu 3214, its power was not enough for our purpose. In the second generation, the Maxon 200142 replaced the Pololu 3214. This new motor has a disk-like shape and a similar diameter as the wheel. This constraint yielded a completely new mounting system, a L shape mounting bracket that connects both the wheel and motor securely and provides a compact design.

3.1 Dual Solenoid Kicker-Chipper System

The kicker-chipper system is comprised of two separate solenoids that each act independently of each other and have different functions. The upper solenoid pulls a chipper arm that connects to a plate that lofts the ball into an arch. A linkage system is used to transfer linear motion to rotational motion between the solenoid and the chipper arm. This system is designed to have minimal friction losses and points of failure. The lower solenoid pushes the ball forward in a horizontal motion. With the use of dual solenoids instead of system, risk of overheating is reduced, response of input signals is increased, and less modes of failure are introduced. With less sources of failure within the robot, the reliability and manufacturability are optimized. As seen in the figure below, the solenoid system houses the two solenoids but also has a section devoted to the dribbler system (area pictured below, dribbler motor not pictured). The purpose of the dribbler is to maintain control of the ball and is powered by a small DC motor connected to a gear system which is connected to an axle covered in a neoprene coating. The neoprene coating acts as a high friction surface to assist in maintaining constant contact with the ball and the gear system increases the torque and decreases the RPM of the dribbler axle to reach an optimized rotational velocity.

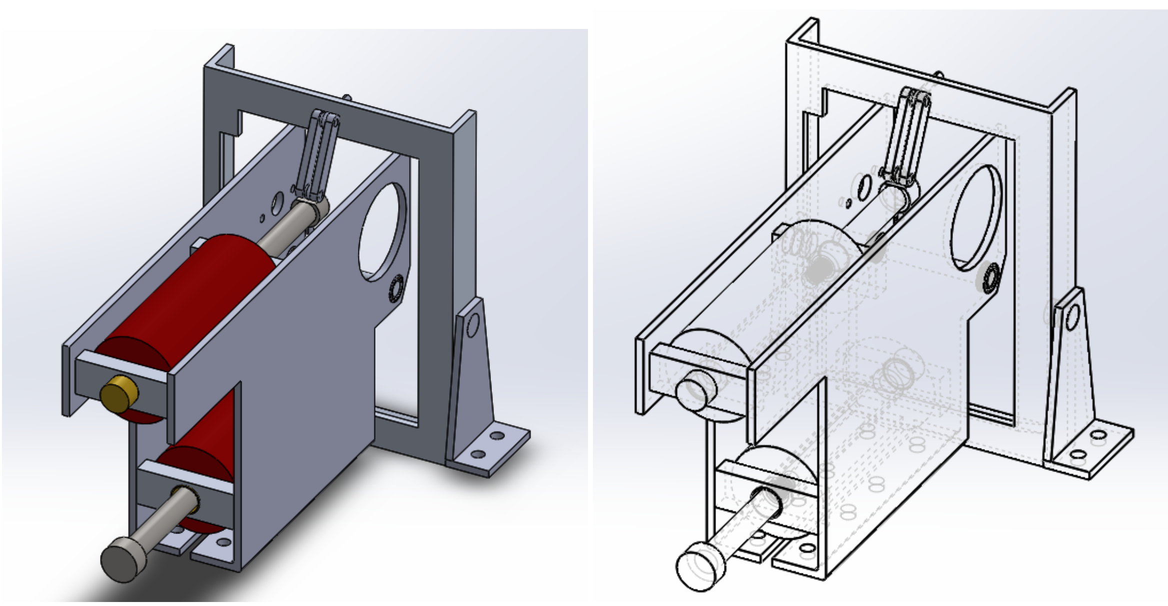

Figure 4: The dual solenoids kicker-chipper system

3.2 Wheel-Motor Mount System

The 3.64 gear ratio was selected because of the driving force the robot needs. For the desired fear ratio, a 20-80 gear teeth set was picked as our transmission system. This gear has similar diameter with the omni wheels and the Maxon motors that we are using. A new motor mount was needed to be designed to make clearance between the wheel, motor, and gear. For compact purpose, an L shaped motor mount was designed.

All components are supported by the fully 3D printed motor mount. The motor is secured to the top of the mount with three through holes with its axles protruding to the other side of the mount where the 22 tooth gear is attached. The 80 tooth gear is mated to the bottom of the 22 tooth gear. This results in a 3.64 gear ratio. The large gear is connected to the omni wheel with a four millimeter axle that travels through the bottom of the mount. An axle is supported by two bearings that lie just inside the face of the motor mount so that the large gear and wheel may sit flush.

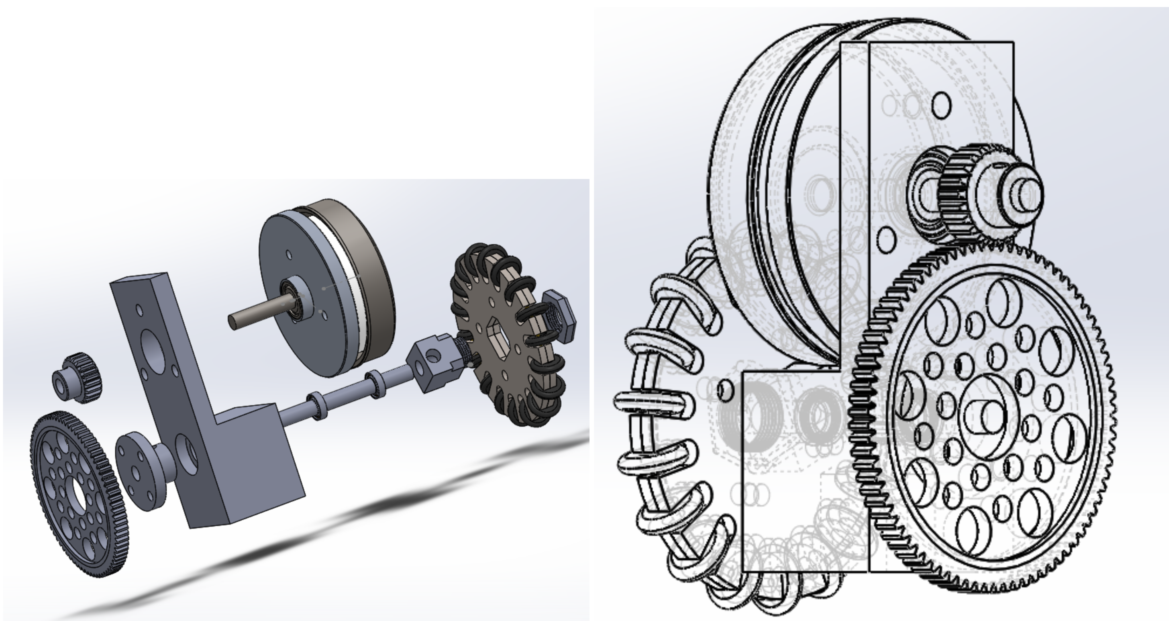

Figure 5: The wheel-motor mount system

4 Software Design

A previous attempt at developing software for small size league utilized partially observable Markov decision processes, or POMDPs. In essence, this approach involved constructing a graph structure of a finite number of field states, and linking nodes with a weight of how likely one state would transition to another given some event occurring on the field. This presented the problem of finding a compromise between describing states in enough detail to make accurate decisions, and having a reasonable number of states. As research continued, it became clear that POMDPs would not be a feasible solution without a substantial amount of preprocessing, possibly combined with some form of reinforcement learning. For this reason we have decided to approach the decision making process from a more orthodox standpoint this year, adapting pieces from other teams open source work to our needs. It is our hope that once we have a solid foundation, we can revisit the POMDP approach and integrate it into our solution.