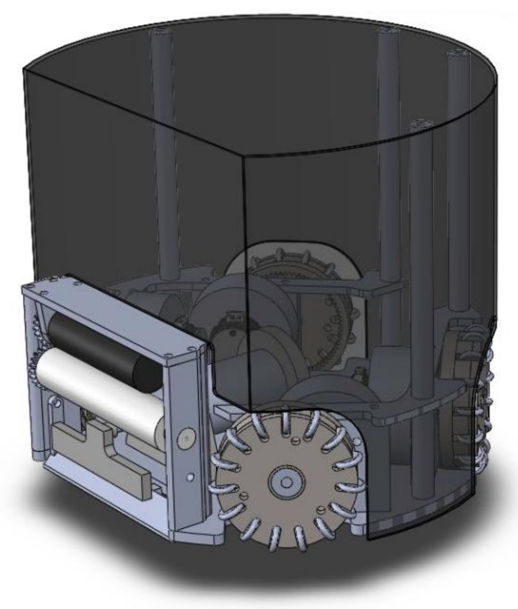
RoboTurk 2014 Team Description

Semih İşeri[1], Meriç Sarıışık[1], Kadir Çetinkaya[2], Rüştü Irklı[1],
JeanPierre Demir[1], Cem Recai Çırak[1]
[1]Department of Electrical and Electronics Engineering
[2]Department of Computer Engineering

Middle East Technical University (METU)
06531 Ankara, Turkey
semih.iseri@gmail.com



Abstract. *This paper briefly explains the current development status of the RoboTurk, RoboCup Small-Size League (SSL) robot team. RoboTurk SSL robot soccer system is designed under the RoboCup 2014 SSL rules to participiate in RoboCup competition being held in João Pessoa, Brasil. This year, we made radical changes in the 2011's design[1]. Actually, we have redesigned all the electronics hardware and the software. Most of the effort spent on embedded software and the motor driver boards on the robot side and the whole game planning software.*

1. Introduction

RoboTurk RoboCup SSL robot soccer team is a project that is studied by the members of IEEE METU Student Branch Robotics and Automation Society since 2008. However, being undergraduate students as well as dynamically changing society members slowed down the development process considerably. After a year of development gap, we gathered a new team and redesigned the most critical parts from scratch.

We can divide the current system in two parts; one part consisting of actual robots, namely *ssl robots*, and the central decision making part, namely *game planner*.

The major improvements we have done so far can be listed shortly as follows;

–Motor control board design; we replaced our earlier brushless motor controller board as it is highly prone to failure and hard to manufacture.

–Main controller board; we replaced our earlier unreliable master-slave configured multi PIC microcontroller based controller board with a single STM32 microcontroller based main board.

–Software design; we are currently developing a new probabilistic method to extract cost data for path planning and game planning.

In the following sections, we state the details about the current development stage of two main units, *ssl robots* and *game planner*.

2. SSL Robots

In RoboCup SSL, there are six robots per team in a normal game. Each SSL robot can be investigated by considering its mechanical hardware, electronics hardware, and on board software components.

2.1. Mechanical Hardware

The mechanical hardware consists of the chassis, motors, kicking and dribbler mechanisms and the omni-wheels.

Chassis is limited to the size of 18cm in diameter and 15cm in height. Our current design is the same design that we used in 2011 [1]. This base weighs 1419 grams excluding the circuit boards.

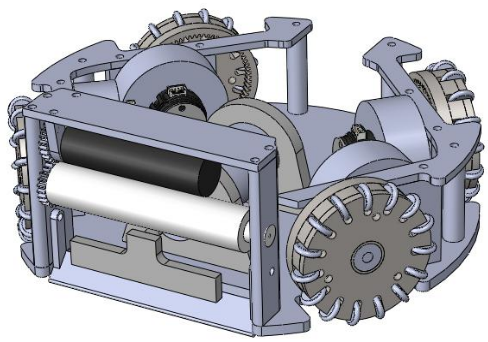


Fig. 1: The robot base

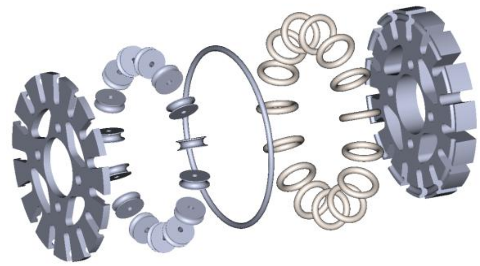


Fig. 2: Exploded view of an omniwheel.

Omni-wheels are the common choice for holonomic drive systems which enable the robot to move in all directions. This year, we are using the same wheels that we used in 2011 [1], which have 15 rollers with rubber gaskets to increase grip. As we are still using our 2011 chassis design, omni-wheels are placed symmetrically with 120 degrees apart in the front and 90 degrees apart at the back.

Motors; Each omni-wheel is driven by 30 Watt Maxon EC45 flat with extended shaft brushless DC (BLDC) motor with E4P encoders.

Kicker; We use a tubular push type solenoid to kick the ball. Currently, we do not have a chip kicker.

Dribbler is actuated by 40 Watt Maxon EC16 BLDC motor. The material on the dribbler cylinder is silicon rubber.

2.2. Electronics Hardware

Electronic hardware of each ssl robot consists of three main parts:

Main Controller Board is a basic board that houses the battery protection circuit and connectors for other peripherals, namely STM32F4DISCOVERY board, IMU and communication modules, BLDC controllers and kicker circuit. Such a board allows us to use off-the-shelf modules, thus reducing our development and debugging time.

For the on-the-robot computing purposes, we use STM32F4DISCOVERY development kit which houses a STM32F407VG microcontroller. This is a powerful microcontroller featuring a 32-bit core with maximum clock frequency of 196MHz, 1MB flash and 128KB RAM. Also, it has the peripherals such as several PWM outputs, hardware quadrature encoder readers and several communication ports to ease our development and increase the reliability.

STM32 based microcontroller is responsible for every action running within the robot. This microcontroller runs the following tasks:

–Reception of commands from the AI computer via wireless module

–Gathering encoder data and running PID over brushless motors by PWM

–Gathering all sensor information on robot

–Passing some of the information to other boards

Motor Controller Board contains L6235 as the main component, which is an all-in-one BLDC motor driver. It receives the sensor data of the BLDC motor and the inputs sent by STM32 microcontroller to its logic decoder to generate the driving signals for its internal MOSFETs. Each motor is driven by a single motor controller board and each motor controller board contains a single L6235 IC. Every module is supplied with one PWM, one direction and 14.8V unregulated LiPo power input (16.8V when full, 14V when empty).

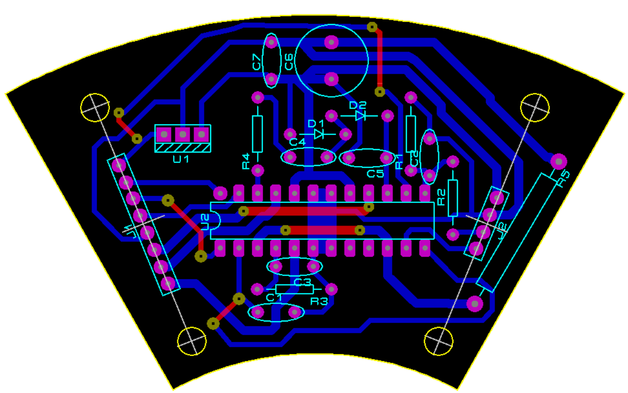


Fig. 3: PCB design of the motor controller.

There are 5 motor controller boards in each robots. Also, in order to reduce our development and debugging time, the motor controller boards are interchangeable, enabling all five motors to use the same motor controller board design.

Kicker Controller Board contains a DC-DC booster to charge a capacitor, and a capacitor discharging circuit to discharge the charges to the main kicking solenoid. There is a small PIC microcontroller in this circuit to generate square wave for booster toroid and stable the voltage at 200V. This circuit uses a 200V 2000uF capacitor to store the charges. This capacitor is discharged on a tubular push type solenoid S-20-100-H which has approximately 3500 turns @ 25 awg. A power transistor is used for switching the discharge.

2.3. On Board Software

On board microcontroller runs on each robot in order to make the robots behave according to the game plan provided by game planner. The basic tasks, such as PID and communication, are driven by interrupts to ensure flawless operation.

In a classical manner, the software gets the messages from the game planning computer such as:

–Translational Speed

–Translational Direction

–Rotational Speed

–Rotational Direction

–Dribbler Speed

–Kicker Speed

–Local State Request

Robot can respond to messages such as “Local State Request” with the information such as:

–Ball Posession

–Battery Level

–Temperature Information

We are working on a method to run the motion control in the on board microcontroller to ease up the work of the planning computer and increase the overall performance. In this method, there is a synchronized clock in every robot and the planning computer and the planning computer asks robots to “be there at time t” with parameters such as final position. For this kind of operation, robot relies on its IMU to approximate the current position while the planning computer supplies position fixes, to eliminate drifts, using global vision.

3. SSL Game Planner

Since we rely on a motion control software running on the on board controller, the game planner consists of two parts only, namely the path planner and strategy planner. Currently we are working on proofs for the concepts that are about to be introduced.

Path Planner is the part of the game planner that draws the best route with minimal cost. The cost is, unlike many of the planners, calculated in terms of momentum. This approach is intended to find the fastest path instead of shortest path. In order to calculate the fastest path, we incorporate an estimation based on the current position and velocity of rival team. We are currently working on this concept.

Game Planner will be based on path planner algorithm, therefore it will be designed after the fastest path concept is proved.

4. Conclusions and Future Work

In this document we have shown the current development stage of the RoboTurk SSL robot soccer team. We have emphasized the major changes in the electronic systems and software architecture. We are trying to avoid the failures that we had in 2011. For the purpose, new robot design will be more reliable and less work-intensive, while the same functionality is preserved. However, we are being redesigned the software architecture from the scratch.

With our young and motivated team, our aim is to make the first step into the competition by building a completely functional robotic soccer system.

References

1. Uyanık K., Yıldırım M., Çamdere S., Sarıışık M. and Olgunsoylu S. (2011). RoboTurk 2011 Team Description