RoboFEI 2017 Team Description Paper

Danilo Pucci, Caio B. José, André O. da Silva, Filippe O. A. Chaves, Guilherme P. de Oliveira, Vinícius M. Alves, Lucas Carlos dos Santos, Lucas A. da Silva, Abner R. Santos, Danilo R. Vassoler, Iago A. Martins, Victor D. de Almeida, Marcos A. P. Laureano, Reinaldo A. C. Bianchi, and Flavio Tonidandel

Robotics and Artificial Intelligence Laboratory
Centro Universitário da FEI, São Bernardo do Campo, Brazil
{flaviot, rbianchi}@fei.edu.br

Abstract. This paper presents the current state of the RoboFEI Small Size League team as it stands for RoboCup Small Size League competition 2017, in Nagoya, Japan, as well as works that are still under development. The paper contains descriptions of the mechanical, electrical and software modules, designed to enable the robots to achieve playing soccer capabilities in the dynamic environment of the Small Size League.

1 Introduction

For the RoboCup 2017, RoboFEI team intends to use the same main board that has been used over the last years and a modified secondary board which is responsible for charging the capacitors and activating the kick systems. The mechanical design is basically the same of previous years with minor modifications, however, new studies are being conducted to create a new kicking system that will be able to change the kicks direction. New features and updates are being implemented in the strategy software so that support for new programming techniques or algorithms is implemented more transparently.

2 Electronic Design

Since RoboCup 2014, ROBOFEI's team released its current electronic design as open source to the community. All the schematics, layouts and firmware are available online, under a Creative Commons license, in RoboFEI's open-source repository.

RoboFEI's eletronic consists of two boards. The main board is responsible for the embedded software for robot control and processes all information regarding the robot's movement [1]. The other board is responsible for charging the capacitors and activate the kicker system, converting electrical into mechanical energy.

This year, we are working on a new kick board for our robots. The main motivation for this is because our current circuit aren't working at the best efficiency;

the board became old, obsolete, often burns, causing excessive maintenance, and the capacitors take a long time to completely charge.

This current circuit is using a Boost converter topology based on MC34063, but the low efficiency and the long time spent to charge a capacitance of $5400\mu F$ up to $160V$ aren't getting the expected performance, so we will change the converter topology from Boost to Flyback.

To achieve this, we chose the LT3751, from Linear Technology, that is a DC-DC Flyback controller optimized for charging capacitors to a predetermined voltage [2]. It is easy to work, because there is a lot of external resistors, that we can change their values to determine the output parameters, frequency, the charging current, overvoltage and undervoltage lockout operation.

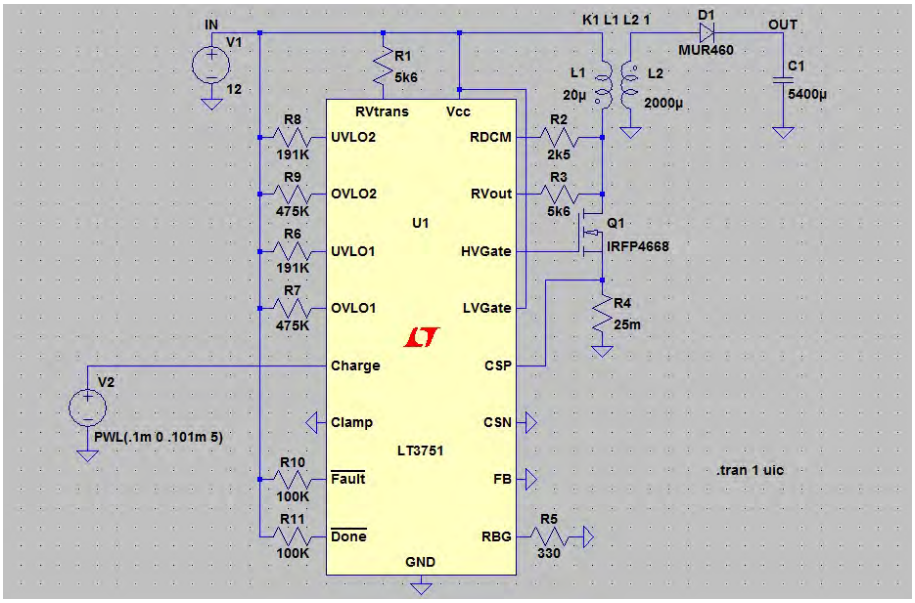


Fig. 1. Flyback converter circuit using LT3751

After some software simulations using LTspice IV, also from Linear Technology, we realized that the Flyback converter circuit, controlled by the LT3751 (see Figure 1), can charge the capacitance faster and provide more power efficiency than our current Boost converter circuit [3].

We are still simulating and doing tests with a prototype. The new board design will be working on a better efficiency and without an excessive maintenance. We will also change the DC-DC converter topology from Boost to Flyback, then the robots will be able to shoot more times in a smaller period of time.

3 Mechanical Design

In compliance with the SSL rules, the height of the robot is 148 mm, the maximum percentage of ball coverage is 15% and the maximum projection of the robot on the ground is 176 mm. The Robot weighs about 2.6kg and the general design can be seen on Fig 2.

Fig. 2. The RoboFEI robot.

The current robot uses a 6000 series aluminum alloy as main material, the factor hardness/weight has a good relation and less frequent part replacements are needed. Wheel axes and the small rollers of the omni-directional wheels are exposed to severe stress thus are made of stainless steel instead.

Some parts that were formerly made of nylon or composite materials are now being manufactured in ABS using a 3D printer. Even though the ABS polymer is structurally weaker, the low cost and ease at which it can be worked on were enough to justify the change since those parts have minimal mechanical stress.

3.1 Work in Progress

We are currently studying the viability of a new mechanism that could be able to vary the direction which the ball is launched. In order to do so, the mechanism will use two parallel solenoids, both connected to the kicking plate. Each solenoid will be triggered with a tiny difference in time, thus there will be a small displacement between each solenoid's plunger along their axis, which will slightly slope the kicking plate, as we can see in Fig 3.

This mechanism will allow our robots to correct the direction of a kick after receiving a pass, or even trick our opponents as to the direction our robot will actually shoot. The launching angle will be determined by the amount of time one solenoid will trigger before the other, which will be calculated based on the desired kicking direction.

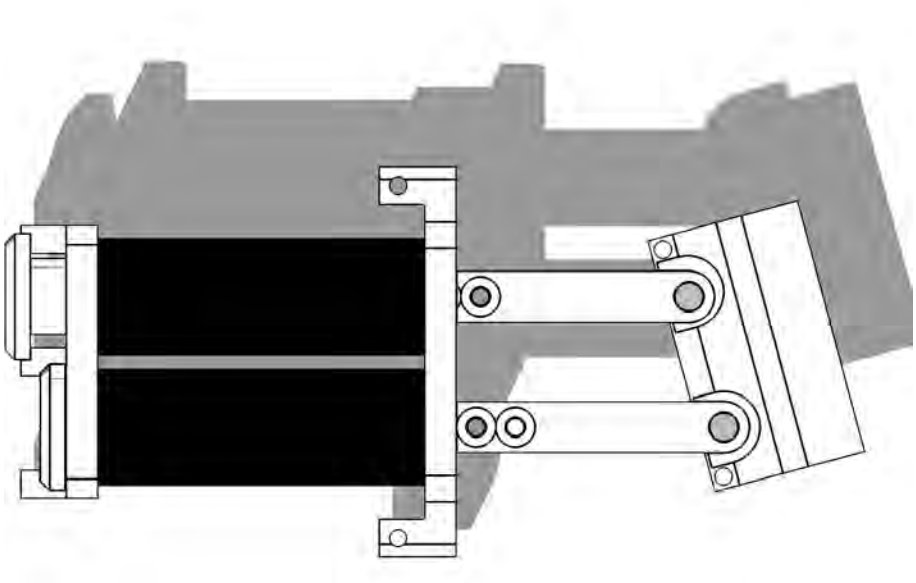
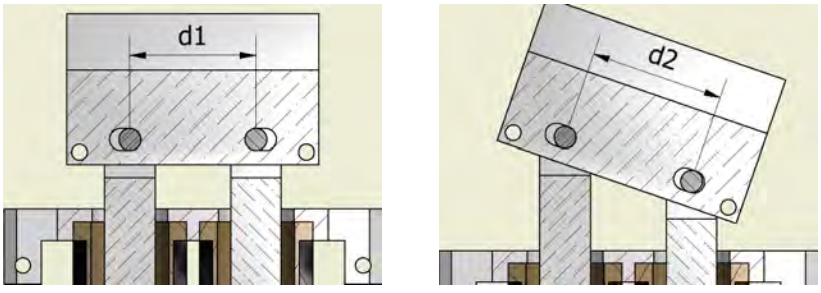


Fig. 3. Top view of the directed kick device scheme.

Our initial idea is to be able to work within a range of a 15 degrees variation to each side, however, the geometry may allow the mechanism to go up to approximately 20 degrees if other limitations permit. To allow the kicking plate's slant, the slots that link it to each plunger are elongated in such a way that minor transversal displacements may occur, as shown in Fig 4.

Since we are currently testing the viability of the mechanical model, we have yet to implement modifications to the electronic and software systems to support this new feature, however, we hope to have at least one robot with this new mechanism fully working by this year's RoboCup.



(a) Straight (b) Angled

Fig. 4. Joint detail.

4 Software Development

On the past few years our software development process undergone several modifications in order to control all the new implementations that were being made. Nowadays this process is well consolidated and functional, however (as usually happens in software projects of long durations), many programmers with different programming styles have contributed to our team's software. To minimize the impact of such variety of styles, some aspects of our development process are being verified and updated:

- Updating of several libraries used in the project - some third party's libraries' functionalities date back to five years ago and are no longer supported. To this end, the software is being revised to support more recent versions. Such change affects mainly the visualization and simulation software, since the libraries used were OpenCV 1.0 (now being updated to OpenCV 3.1) and PhysX 2.8 (updated to PhysX 9.14);
- Due to the libraries update, other modifications, related to using deprecated functions and incorrect types of data, are necessary;
- Good code building practices - there wasn't much worry regarding the best ways of software development, which is now being revised to implement modularization practices, object orientation, and others tactics;
- Together with such good practices, a new interface is under development. This interface will allow the software to adjust in real time a variety of skill and game strategies;
- Project adaptation to the newest C++ compiler, Visual C++ 2015 (we currently use Visual C++ 2012).

4.1 Data Recording Software

It is possible to acquire various information from game logs. We are currently working on analysing the efficiency and productivity of both teams in play. For such task, a new implementation is in test phase and it aims to present, in a consolidated way, some information regarding the match, such as: ball possession, goal kicks, successful passes, etc.

For this analysis to be possible, every situation that may occur during a SSL game are separated in two types:

- Events: Informations received directly from SSL-Referee. Based on the nature of the event the software merely records the information (like in a Stop Game or Time Out command) or starts an analysis to also check what was the cause of the event. For example, in the case of a goal, which robot scored it.
- Actions: Differently from events, there is no external sign that an action is occurring. To identify exactly when a action is happening the software is constantly analysing the match. We currently want to identify two possible types of actions: Kicks and Passes, which can occur while the game is in the following states: Normal Game, Indirect Kick, Direct Kick and Penalty.

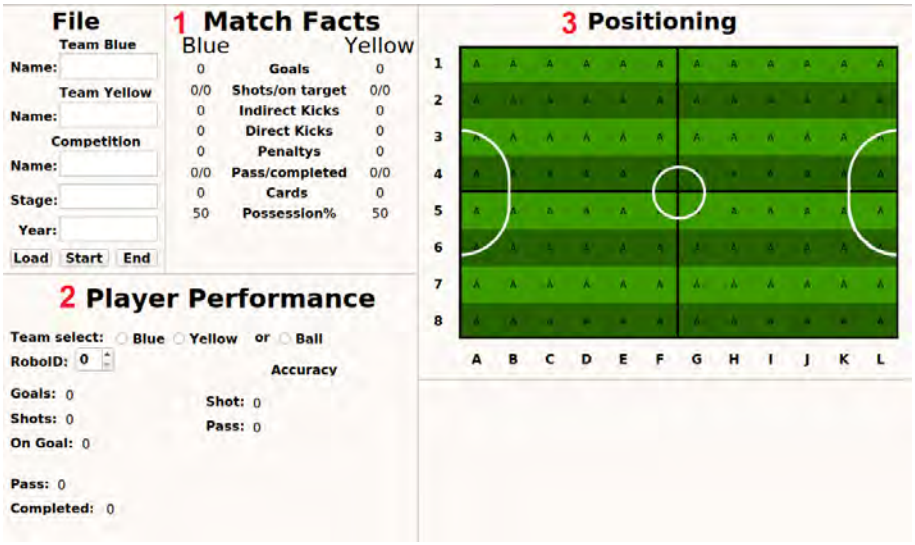


Fig. 5. Data Recording Interface.

Every information gathered is shown in a graphical interface, as we can see on Fig. 5, where both team's statistics are displayed in section 1 (Match Facts). Information regarding each robot individually is presented on section 2 of the interface (Player Performance) and on section 3 it will be shown the movement performed by the selected robot or even the ball over the course of the game.

With this tool, besides being able to gather statistic data from any match from which a .log file was generated, we also intend it to work in real time, communicating with the strategy system and allowing the team to adapt to different situations based on the feedbacks provided by the data recording software.

4.2 Work in Progress

The current software architecture does not allow "radical" changes to be implemented, such as new path-plain algorithms, learning-and-response algorithms, or even the possibility of including new plays during the championship. To solve this situation, a new architecture is being developed and we hope to use it in the next event.

As can be seen in Fig. 6, the new structure provides for some steps:

- Action: any basic action that the robot can perform, such as: kicking, passing, checking with the ball, etc;
- Move: a move is made up of one or more actions and some rules can be checked for that play, there may be preconditions, post conditions, conditions (during the play) or a more elaborate technique Decision, for example). Taking as an example the play "kick the ball towards the opponent's goal", a precondition is to ensure that the player is facing the opponent's goal;

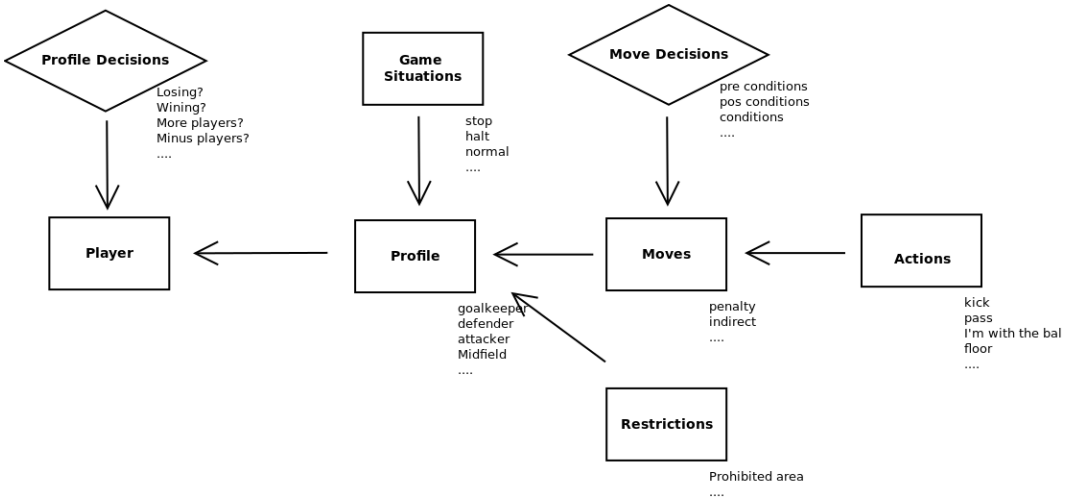


Fig. 6. Relationship between actions, plays and player profile.

- Profile: consists of several plays previously registered according to the role to be played by the robot (goalkeeper, attacker, etc). The behavior (match status) and possible restrictions (such as goalkeeper can not leave the defense area) will be applied to each profile;
- Player: is the profile assigned to the robot and is influenced by a set of game situations. For example, a robot marked as a defender can assume the role of attacker if the system realizes that it has more players in the field. Or an attacker take on the role of defender in the opposite situation.

The system used a database structure that will allow the inclusion of new profiles and plays, allowing the selection in real time according to the specificities of the game (no code recompilation required).

As can be seen in figure 7, the system will be divided into 2 parts, both of which will receive SSL-Vision and SSL-Referee information. The separation was planned so that it is possible to implement a simulator in a transparent way, ie put the system itself to play against you. For this, a simulator will be developed that will play the role of SSL-Vision and SSL-Referee simultaneously.

Finally, as can be seen in Fig. 7, the software will have 2 layers of control, in order to take advantage of the multi-thread capacity and parallel processing of the current processors:

- 1ªlayer or main control: this layer will be responsible for receiving the SSL-Vision and SSL-Referee information and will also make the initial settings of the game, as well as will allow modifications during the game (such as manually modifying a player's profile, for example).
- 2ªlayer or control per profile: responsible for coordinating the robots set for that profile. In this layer there will be the decisions and intelligence of the

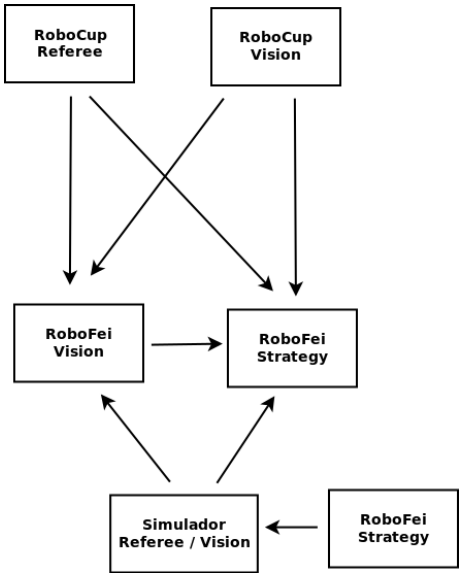


Fig. 7. Simulator architecture.

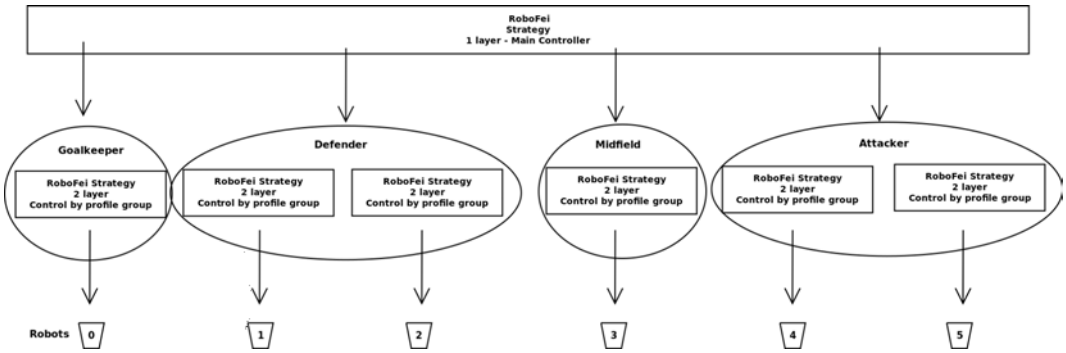


Fig. 8. System architecture.

robot. Currently as possible there are 6 robots in the field, you can have 6 different profiles. If a robot has its profile changed manually, the main control will notify profile control of the new situation.

We hope that this new software framework will allow researchers in the team to implement agile and independent new techniques and algorithms.

Acknowledgments

We would like to thank, in advance, the Small Size League Committee, for the consideration of our material. We would like also to immensely thank the staff of Centro Universitário da FEI, for all the help we always received.

References

1. Danilo Pucci, Fernando Rodrigues Jr., Caio Schunk, Caio Braga José, Victor Torres, Thiago Silva, Victor Amaral, André De Oliveira Da Silva, José Angelo Gurzoni Jr., Reinaldo A. C. Bianchi, and Flavio Tonidandel. Robofei 2014 team description paper. 2014.
2. Lt3751 datasheet, high voltage capacitor charger controller with regulation. Linear Technology Corporation. (www.linear.com), 2016.
3. Michael E. Islas. Improvement Techniques for high voltage capacitor charging methods. University of Central Florida, 2008.