# RoboFEI Small Size League 2010 Team Description Paper

José Angelo Gurzoni Jr., Eduardo G. Nascimento, Daniel Malheiro, Gabriel Francischini,
Milton Cortez, Bruno Tebet, Samuel Martinez, Rodolpho Felizatti,
Reinaldo A. C. Bianchi, Flavio Tonidandel
Robotics and Artificial Intelligence Laboratory
Centro Universitário da FEI
São Bernardo do Campo, SP, Brazil
{jgurzoni, flaviot}@fei.edu.br

***Abstract*— This paper presents an overview of the RoboFEI team state for the RoboCup Brazil Small Size League competition 2010. It 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 RoboCup Small Size League.**

## I. INRODUCTION

RoboFEI team has been competing in the RoboCup Small Size League since 2008, both in the RoboCup Brazil and RoboCup world championships. Based on the experience acquired during these participations, RoboFEI presents a new hardware for the 2010 contest, which brings significant advances both in its mechanical and electronic systems. This paper shows the description of the new hardware, featuring comparisons between the old and new robot versions, and also describes the software modules which compose the strategy system, including the state predictors and a dynamic role selection method, based on market economy.

## II. ELETRONIC DESIGN

### A. Main Board

The main board features a Xilinx Spartan 3 FPGA (X3CS400) in a PQ208 packaging, responsible for performing all the logic functions. It is used as CPU, through its Microblaze 7.1 IP core, as brushless motor controller, controls the radio, the kicker board, and interact with all the sensors. Integration of all these functions in the same IC has eliminated difficulties related to the communication and synchronization of different micro controllers, while at same time reducing considerably the number of components on the board. This is an advance in relation to the previous design, which had an ARM7 as main CPU and dedicated 8-bit micro controllers for each motor, allowing faster reading of the odometry sensors and simplified firmware programming. The Xilinx Spartan 3, with its IP core operating at 50 $MHz$, also provides faster computation, because of its hardware FPU (floating-point arithmetic unit). The embedded firmware is stored in a 256K words RAM memory connected to the FPGA, to ensure even complex firmwares can be load.

The radio used on the board is a TRW-24G transceiver (based on the Nordic nRF2401A IC) operating at 2.4 $GHz$, set at 250 $Kbps$ data rate. The board also has five brushless motor drivers designed with the IRF7389, an SMD IC featuring N-P complementary channel MOSFETs, an AD7918 Analog-to-Digital IC for motor current sensing, and JTAG interface for in-circuit programming and debugging.

The power to the main board and motors can be provided by either one or two 3-cell (11.1V), 2200 $mAh$, LiPo batteries.

### B. Kicker Board

The kicker board is responsible for controlling both the shooting and the chip kick devices. It has a boost circuit designed with the MC34063 IC, which uses a $100KHz$ PWM signal to charge a 3300 $\mu F$ capacitor up to 200V. This IC controls the whole circuit, sparing the main-board's CPU from the need to generate the PWM signal and monitor the capacitor's charge. In relation to the previous design, it represents an increasing of 80% in the capacitor's tension and a 5 times faster charging rate, due to the PWM's higher frequency.

The kicker board features an independent power supply, fed by a 2-cell (7.4V) LiPo battery with 1300 $mAh$ charge. The connections between the kicker and main boards are opto-coupled, to avoid spikes and eventual damage to sensitive electronic circuitry.

## III. 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 146 $mm$.

The mechanical design for 2010 presents a significant evolution, in relation to the previous robot. The key changes are the chip-kick implementation, which the previous model lacked, the improvement of the damper system for better ball reception and handling, and a more robust, yet practical, robot frame, designed considering assembly and disassembly simplicity.

On the previous robot, replacement of circuit boards and batteries was difficult, requiring a good number of screws to be removed. To solve this problem, sliding rail stands were added as support for the electronic boards. The batteries are also slided into their positions, within plastic supports easily accessible from the back of the robot. A general view of the robot can be seen in figure 1.

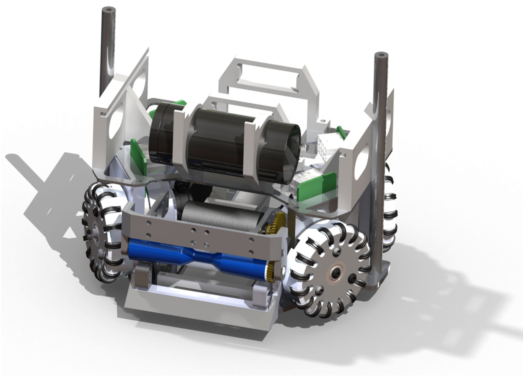


Fig. 1. Mechanical view of the robot

## A. Driving System

During RoboCup 2009, it became explicit that, to effectively play an SSL game, a team has to match the demanding parameters of speed and acceleration employed by the other competing teams. It is a must on a game based on quick reactions, and the previous robot, using Faulhaber 2232 DC motors, was not up to that task. Also, the new design was planned not only to match, but to exceed the current status of the competitors. To accomplish this goal, the motors selected are the Maxon EC-45 50W motor, a motor capable of outperforming the motor becoming the *de facto* standard of the League, the Maxon EC-45 30W. With 6700 RPM no load speed and 822 $mNm$ stall torque, the EC-45 50W allows the RoboFEI 2010 robot to use 3:1 reduction ratio and yet be capable of accelerating above 6 $m/s^2$.

The higher power the motor also allow the robot to be symmetric, with all four wheels disposed at 33° in relation to the longitudinal axis, without making it slower than the faster robots currently on the League.

However, there is a trade-off to be balanced. Adopting the 50W version of the EC45 motor, instead of the 30W, results in around 350 grams weight increase, mainly because the additional 20W power requires a larger battery. This trade-off is acceptable, though, as the robot gains more than 3 times the maximum stall torque and 1.5 times the maximum speed.

## B. Wheels

The new wheel design focused on solving the excessive vibrations the previous design had, caused both by the backlash between the small wheel and its mountings, by the small wheels disposition. To achieve this goal, some design changes were applied. The distance from the small wheel to the center of the main wheel was reduced and the small wheels and its axis are now machined as a single piece. This allowed a reduction on the amplitude of the vibration caused by the wheels on the robot, from 2.0 to 0.38 $mm$. With less vibration, the control of the robot becomes smoother and the stress on the mechanical and electronic parts is reduced.

Aside from these benefits, the wheel also became easier to mount and dismount. The rubber rings of the small wheels were also changed, from the O-rings, that used to loosen during the game, to H-rings, which thinner profile, less prune to loosening.

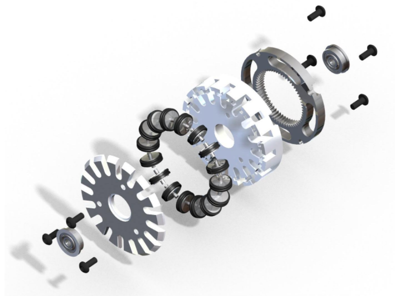


Fig. 2. Exploded view of the wheel

The new wheels can be seen in exploded view on figure 2. They have 58 $mm$ diameter, body made of aluminum and 16 small wheels made of stainless steel. They have two bearings, not one like the previous model, to better handle axial loads.

## C. Kick System

The Kick device is composed of a 30 $mm$ diameter cylindrical solenoid, built of a 14 $mm$ diameter SAE1020 steel core, where 4 AWG21 wires are coiled, in parallel. A major improvement on the new kick device is the increase in the distance traveled by the plunger, to 36 $mm$. The longer the distance, the more acceleration the plunger achieves, resulting in stronger kicking force.

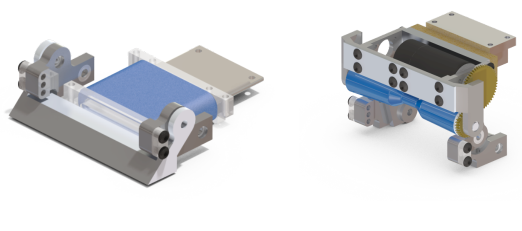
The chip-kick device consists of a rectangular solenoid mounted under the cylindrical solenoid of the kick device. Its core is made of nylon, to reduce weight, where a 3.75 $mm$ width steel plunger rests. When activated, the plunger pushes the aluminum part, launching the ball with a 45○ angle. It can be seen in figure 3(a).

## D. Roller and Damper Systems

On the 2010 robot, the Maxon EC22 20W motor replaces the Faulhaber 2224 as the motor on the roller device. Not only the EC22 is more powerful than the 2224, it achieves 5 times more no-load speed, 35500 RPM, allowing the roller device to have greater angular speed while maintaining the required torque.

For improved efficiency, the rolling rod itself, made of rubber, has its ball contact width increased to 82 $mm$, and its outer diameter dwindles 2 ○ toward the center of the rod. As the rod rolls, this dwindling causes the ball to move toward its center.

The damping system is mounted on the back of the roller device, as figure 3(b) shows. The damping is responsible for improving the ball control during pass receptions and helps the ball control when it is on the roller and the robot moves. The articulated parts of the device use bearings, to improve efficiency.



(a) (b)

Fig. 3. (a) Chip Kick assembly view (b) Damper system views

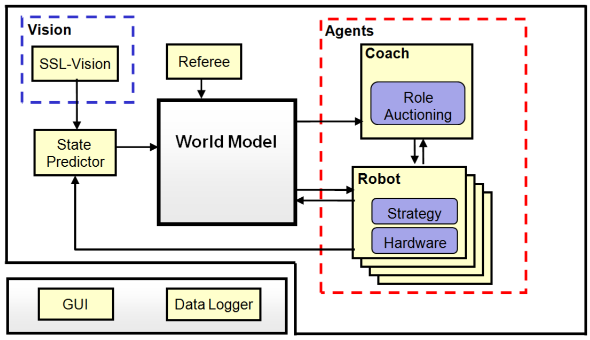



Fig. 4. Block Diagram of the software system

## IV. PATH PLANNING AND OBSTACLE AVOIDANCE

The path planning and obstacle avoidance algorithm employed is based on the Rapid-Exploring Random Tree (RRT) with KD-Tree data structures, proposed by [1], and on the ERRT algorithm developed by [2], complemented by an algorithm to include preferred path heuristics and set the angle of approach. The algorithm based on RRT was chosen because (i) its capacity to efficiently explore large state spaces using randomization, (ii) the probabilistic completeness offered, (iii) its lookahead feature and (iv) the easiness of the algorithm's extension, when new constraints or heuristics are deemed necessary.

This section focuses on describing this add-on algorithm, which is implemented on top of the ERRT base algorithm.

The add-on algorithm has the function to set the angle which the robot approaches the ending point, as commanded by the strategy layer, an item that many path planners do not treat. It is not desirable, for example, that a robot going to the ball on the defensive field accidentally hits the ball in the direction of its own goal, or yet, that an attacking robot arrives at the ball in a position in between the ball and the opponent's goal. To create a path that conforms to the angle of approach requirement, a circular virtual obstacle centered on the ending point is created, with a $10°$ width circle segment and vertex at the desired angle removed. This effectively forces the path planner to create a path the reaches the ending point passing through this $10°$ opening. The radius of this obstacle-like constraint is set to a value close to half the size of a robot.

## V. SOFTWARE SYSTEM

RoboFEI software system consists of world modeling, visualization and data logging blocks and logically independent agent modules. Figure 4 shows the architecture diagram.

### A. World Modeling via State Predictor

The world model is updated by the state predictor module. This module receives vision data from the SSL-Vision and motion command data from the agent modules, sent when they command the robots via radio, and performs state predictions, The prediction is to advance the positions sent by the SSL-Vision from their original capture time to the present and then forwarding one strategy cycle in the future, the so called latency of the strategy system. This latency is currently on the order of $80ms$.

The prediction algorithms used for ball, robots and adversaries are different. The ball prediction is made by an Extended Kalman filter (EKF) (see [3]), a well known method for position estimation.

The robot's prediction is performed similarly to [4], with multi-layer perceptron neural networks. These networks are trained off-line to learn the robot's motion model, receiving past frames and motion commands as input and a frame $n$ steps in the future as output. Once trained, the networks are used for on-line estimation of the robot's position and rotation.

As for opponent estimation, currently it is done with simple extrapolation of the last velocity data and Gaussian functions.

### B. Agent Modules

Each robot player is an independent module, executing its own instance of one or more strategy submodules and its hardware specific functions (such as motion control and sensing). The current implementation relies basically on a layered strategy architecture and a market based approach for dynamic allocation of functions, both described ahead on this section.

### C. Strategy module

Building multi-agent systems in a layered architecture with different levels of abstraction is a popular approach (see [5], [6] and [7]) well suited as foundation for machine learning algorithms, one of the research goals. For this reason, the strategy module architecture was divided in three abstraction layers.

The lowest layer has the so called *Primitives*. Primitives are actions that mostly involve directly activating or deactivating a hardware module such as to kick the ball with a given strength, activate the dribbling device, rotate or move to a position.[1]

[1] Actually, moving to a position is a special case of a primitive with underlying complex logic. It calls the path planning system to perform obstacle avoidance.

On top of the primitive layer, is the *Skills* layer. Skills are also short duration actions but involving use of one or more primitives and additional computation, such as speed estimation, forecasting of objects' positions and measurement of primitive tasks' completion. This layer has a small set of skill functions, yet that represent the basic skills required in a robot soccer game, like shooting the ball to the goal (aiming where to shoot), passing the ball to a teammate, dribbling, defending the goal line or tackling the ball (moving toward the ball and kicking it away).

One example of such skill is the Indirect Free Kick skill, which employs a multi-criteria weighted evaluation to determine the best for the robot to pass the ball to. Grids are constructed in different areas of the field, then the weighted multi-criteria evaluation function is employed to decide which of the grids contains the best candidate position. Once the area is chosen, the function recomputes using a finer grid, to determine the exact position. The objectives evaluated are the Euclidean distance of each position, in relation to both the robot and the ball, the width of the angle a robot in that position would have to to kick to the goal and the distance between the current positions of the teammates receiving the pass and the chosen positions in the grid.

The skills are employed by the *Roles* layer, which contains different roles, created using combinations of skills and the logic required to coordinate their execution. There are roles called fullback, defender, midfielder, striker, forward and attacker. No particular robot is tied to a given role (except the goalkeeper), and there is no limitation on how many instances of the same role can exist, what allows dynamic selection mechanisms to unrestrictedly create role combinations.

### D. Market-Based Dynamic Role Selection

Dynamic role selection is key to the strategy, as it allows the team to have, for example, three defenders when in a defensive situation and three attackers and a mid-fielder when attacking, as well as to adapt to different opponent behaviors.

On RoboFEI 2010, a market-based approach for role allocation is under experimentation. The algorithm is designed in a similar fashion to the Murdoch [8], with the tasks being the player roles. The available roles are auctioned by the coach and the players bid for them. These bids represent the utility, or fitness, of the player to perform the role, calculated by the player itself, and are key for the auctions to work properly, as improper values would mislead the selection.

The utility functions represent how well a player can perform that role, given the teammates, opponents and ball positions on present and few past frames. The functions consist of evaluation metrics for a particular role, producing a scalar as result of the weighted sum of each metric. The weights are empirically adjusted by the programmer.

An auction works as follows:

- The start of the auction is announced by the coach, along with the list of roles;
- The player's utility function calculates a scalar value for each of the roles being auctioned and submits back to the coach;
- The coach, using the Hungarian method [9], selects the best combination of winners, maximizing the scalars received;
- The coach announces the winner players, who start to perform the given roles.

## VI. Conclusions and Future Works

### A. Conclusions

The new mechanic and electronic designs significantly improve RoboFEI team's research and competition capabilities. Although a new revision of the designs is expected for 2011, the current robot is expected to be in service for the next coming years.

As for the software architecture presented, its modularity is an advance with allows faster and cleaner code development. the architecture also serves to achieve the multi-agent paradigm, where each robot can be programmed as a self-contained, independent agent, which is important for the future of the AI research within the laboratory.

### B. Future Works

Evolving the action and role selection algorithms is the focus for the near future. A promising approach, in which work has already begun, is based on the Pareto Optimality [10].

## VII. Acknowledgments

We would like to immensely thank the staff of Centro Universitário da FEI, for all the help we always received.

## References

[1] A. Atramentov and S. M. LaValle, "Efficient nearest neighbor searching for motion planning," in *IEEE International Conference on Robotics and Automation*, 2002, pp. 632–637.

[2] J. Bruce and M. Veloso, "Real-time randomized path planning for robot navigation," in *Proceedings of IROS-2002*, 2002.

[3] G. Welch and G. Bishop, "An introduction to the kalman filter," Department of Computer Science, University of North Carolina, Tech. Rep. TR 95-041, 2001. [Online]. Available: http://www.cs.unc.edu/ welch

[4] S. Behnke, A. Egorova, A. Gloye, R. Rojas, and M. Simon, "Predicting away robot control latency," in *Proceedings of 7th RoboCup International Symposium*. Springer, 2003.

[5] M. J. Mataric, "Learning in behavior-based multi-robot systems: Policies, models, and other agents," *Cognitive Systems Research*, pp. 81–93, April 2001.

[6] M. Bowling, B. Browning, and M. Veloso, "Plays as effective multiagent plans enabling opponent-adaptive play selection," in *Proceedings of International Conference on Automated Planning and Scheduling (ICAPS'04)*, 2004.

[7] B. Browning, J. Bruce, M. Bowling, and M. Veloso, "Stp: Skills, tactics and plays for multi-robot control in adversarial environments," in *IEEE Journal of Control and Systems Engineering*, vol. 219, 2005, pp. 33–52.

[8] B. Gerkey and M. Matarić, "Sold!: auction methods for multirobot coordination," *Robotics and Automation, IEEE Transactions on*, vol. 18, no. 5, pp. 758–768, 2002.

[9] H. W. Kuhn, "The hungarian method for the assignment problem," *Naval Research Logistics Quarterly*, vol. 2, no. 1, pp. 83–97, 1955.

[10] Y. Jin and B. Sendhoff, "Pareto-based multiobjective machine learning: An overview and case studies," *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, vol. 38, no. 3, pp. 397–415, 2008.