RoboIME: From the top of the world to LARC 2019

Carla S. Cosenza, Gabriel B. da Conceição, Gabriel F. Santos, Felipe W. V. da Silva, Antônio S. G. Pereira, Erick T. da Silva. Leonardo G. Gonçalves, Lucas B. Germano, Lucas G. Corrêa, Luciano de S. Barreira, Luis D. P. de Farias, João G. O. C. de Melo, Matheus Bozza, Matheus P. de Souza, Nicolas S. M. M. de Oliveira, Onias C. B. Silveira, Rebeca P. dos Reis, Yugo Nihari, Guilherme M. Gonçalves, Gustavo C. K. Couto, João M. de L. Filho, Vinicius de F. L. Moraes, Luis R. L. Rodrigues and Paulo F. F. Rosa

Instituto Militar de Engenharia, Rio de Janeiro, Brasil

rpaulo@ime.eb.br
http://roboime.com.br

***Abstract*— This paper describes the electronic, mechanical and software designs developed by the RoboIME team in order to join the LARC 2019. The overall concepts are in agreement with the rules of Small Size League 2019. This is the sixth time RoboIME participates in the LARC.**

I. Introduction

RoboIME is a Small-Size team from the Instituto Militar de Engenharia, IME, located in Rio de Janeiro, Brazil. This is the thirteenth time the team takes part in competitions, being the best results second place in the RoboCup 2018 and first place in the Latin American Robotics' Competition 2017.

All students that work in the SSL project are members of the Laboratory of Robotics and Computational Intelligence at IME. Team's previous works were used as reference[3] , as well as the help from former members of the team as consultants and tutors.

This article describes the team's general information and improvement in the most recent semester, since the previous TDPs for LARC 2018 have detailed explanations on the previous changes. This article is organized as follows: software in section II, embedded electronics in section III and mechanical design in section IV. Conclusions and future works are discussed in section V.

II. Software Project

This paper reports the main improvements and changes since the 2018 LARC project.

The main focus of the project was to rebuild all artificial intelligence with a hierarchical architecture, based on the STP(Skills, Tactics and Plays) structure, aligned with object-orientated programming. Hence, few changes in robot's behavior were implemented in order to obtain a functional and organized project.

A. New Software Architecture

It was necessary to transform and innovate the form of treating the intelligence programming at RoboIME, in these terms, there were great structural changes the code, since the code was reformulated for object-oriented programming. In this aspect, it was possible to modify the way of approaching the artificial intelligence, following the STP architecture.

The behavior of the robot is divided in a hierarchical form. Skills are the pillar of the architecture since they are the minimum robot actions. Skills are selected by a State Machine that gives the robots orders through a set of skills. This set originates a tactic. In this sense, a set of tactics controls the robot's behavior in each determined Play. Besides, the play is the architecture's component that dictates, through a set of tactics, the team's behaviour at any given moment, such as a pass state or a direct kick. The architecture was based on the following reference: [1]

B. Plays

Plays were divided in three different types in order to allow for each group of robots to execute its tasks in parallel processes.

All plays are descendant classes of an abstract class ”Play”.

The play's structure was based on reference [2] .

1) Offensive Plays: Responsible for controlling the non-defense robots. Some examples:

- PassState
- NormalGameWithEnemyBallPossession
- DirectKick

The Offensive Plays have priority when picking the robots. The remaining robots will execute Defensive Plays. It is utilized parameters passing by reference in order to allow them to function fully in parallel between Offensive and Defensive plays.

2) Defensive Plays: Responsible for controlling the defense robots. Some examples:

- NormalGameWithDuelist
- Barrier
- ManMark

Defensive robots perform tasks much less complex than offensive robots. Hence, in parallel plays, it is possible to achieve a better reaction time for the defending robots.

3) Keeper Plays: Goalie will execute just one play in which all its strategy is implemented in order to have a faster decision process.

C. Personalities

With the AI advancing, changes were made to the personalities (see Personalities in [3] in order to have a better understanding on how other personalities work).

- *Second Striker:* The robots can now assume this new temporary offensive personality, which appears in the passing state and direct or indirect kick for the ally team. Outside these situations, the personality ceases to exist and gives its place to another defender. The main objective of the personality is for the Attacker to have more than one possibility for passing and, therefore, making it less predictable. In passing situations, the second striker gets ready to receive a pass, assuming a slightly retreated position, also chosen by the Inverse Best-Y method, different from the one assumed by the striker. This personality can only be active when all the robots are in the field and can be deactivated manually through a button before the match or during the breaks.

D. Test Module

- *Pass Test:*
  In order to test the difference between the pass efficiency to striker and the second striker, the pass test module was created. Initially, the ball is placed in a random position, inside the field. One robot goals in this spot to pass the ball to another one, which stays in position to receive it and then kicks to the goal. Besides, everything is positioned fully automatic to focus on doing many tests without human interference.
  The quantity of random points is predetermined. After all of them have been generated they will be placed in a array. After each failure attempt, when the ball collides with other robot or missed the goal, the ball is positioned based on the array order, until it reaches the end.
  This test has a statistical module to count the number of successful kick attempts in order to verify the efficiency of the pass strategy executed by this two robots in certain personalities.

E. Penalty Kick

The penalty mode aims to trick the goalkeeper through the change of the target that the penalty taker is directed to. The strategy consists on finding the farthest goal point from the goalkeeper and then change, continuously, as the goalkeeper adjusts to the new target. When close to the ball, the target is locked and the kick occurs.

III. Electronics Project

During RoboCup 2018 it was possible to improve the debugging, enabling an easier recognition and substitution of the problem due to the modularization. During LARC 2018, the newer version of the main board completely replaced its older version, as well as the kicker board, maintaining the modularization and the debugs and repairs of the robot.

For this year's competition, the firmware has been the focus of work, enabling our robots to use more of the features that came from the new main board improvements. This board's positioning of components has also made possible to change the communication module model and even couple it to a small antenna without the need to make any changes in the firmware.

Also, some changes in the PID control algorithm are under test and a test firmware was developed in order to make estimations about the motor's model through serial commands on the USB port, which will also help with debugging.

A. Firmware

The firmware was improved from a bare metal code that ran with hardware interruptions to execute the communication and the wheel controls, to a real time operating system. FreeRTOS was chosen due to its ample support in online community and its robustness.

Previously, the firmware in the transmitter was the same as the one in the robots, being the connection to a serial port the trigger to call some functions instead of others. This was also changed with the addition of the FreeRTOS tasks, in order to make the code clearer and more readable.

With such architecture, new additions to the code are easier to make, not being necessary to know the whole flow of the code, just the particular module being modified.

The transmitter's firmware consists of two main tasks, responsible for reading the USB COM port and transmitting its commands through the NRF24l01P and the other responsible for reading and writing in the command line for debugging purposes.

Besides, the robot's firmware has a task for receiving the commands from the transmitter, another for controlling the four wheels' speeds, a third for activating the kicks and check for the ball sensor and a fourth for transmitting back data for the software.

Many major changes are being implemented this year in order to make the right use of the new components of the main board such as the Inertial Measurement Unit (IMU), the SD card, the motor current sensors and the ID display. With the FreeRTOS all those features become easier to use, but things like the IMU or the current sensors have to be taken very carefully in order to get the right results out of it.

While some of those implementations are still in study, the problems with the ID display were solved. In LARC 2018, every time the robots were turned off and on, they had their ID resettled to "0", which may cause many confusions during a match. This problem was solved by saving the robot's ID value inside the Flash memory of the microcontroller every time the ID button is pressed. By doing this, the ID value maintains saved even when the robot is not turned on.

B. Control

Maintaining the robot at the expected speed or position is very important so that the software is able to work properly. To try to keep the robots moving as expected it was used a PID control algorithm for each wheel independently, but using the same constants for all of them. Since each wheel have non-ideal movements, subjected to friction with the ground and its motor properties, this implementation causes some movement mistakes which, by now, are only corrected by the AI software. The goal is to detect and reduce those errors by changing the PID controller algorithm.

The desired speeds for each wheel are calculated from the normal, tangent and angular speed values, which are calculated by the AI software and sent to the robot through RF communication. These 3 velocities describe the actual movement of the robot and are converted to 4 wheel speeds by the following matrix:

$$V_{wheels} = DV_{1x3}$$

$$\begin{bmatrix} v_1 \\ v_2 \\ v_3 \\ v_4 \end{bmatrix} = D \begin{bmatrix} v_x \\ v_y \\ w.R \end{bmatrix}$$

where:

$$D = \begin{bmatrix} -sin\phi & cos\phi & 1 \\ -sin\phi & -cos\phi & 1 \\ sin\theta & -cos\theta & 1 \\ sin\theta & cos\theta & 1 \end{bmatrix}$$

In order to overcome this situation, the matrix D^+ was added in the feedback loop, which is the pseudoinverse from D, that associates the normal, tangent and angular speeds with the speeds of the wheel:

$$V_{1x3} = D^+ V_{wheels}$$

By doing this and changing the feedback system of the closed loop control, the errors for the PID algorithm are now calculated based on the normal, tangent and angular speeds. This approach allows the controller to detect the effects that each wheel causes on the robot's velocities and, since the normal, tangent and angular speeds are linearly independent, the control can act separately on each of them so that they are superposed in the end. The major changes in the feedback system of the controller are shown below:

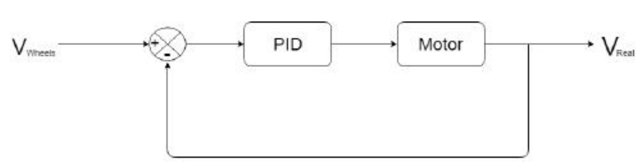

Fig. 1: Previous PID control

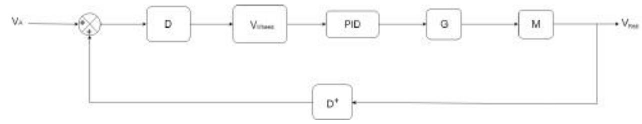

Fig. 2: Current PID control model

C. Board Designs

In LARC 2018 the main board was completely substituted by its newest version as well as the kicker module, with small modifications of architecture regarding the position of specific components.

The motor module remained unchanged since the last competition, proving itself to be very reliable, even on the newest version of the main board, with only few remaining problems during the games, either in the RoboCup or LARC.

1) Stamp module: This module is responsible for performing all the logical functions, serving as a brain for the electronic system. The module is a commercially available board - the STM32F4-Discovery; it is a development kit that aggregates an Arm Cortex M4 microcontroller with a series of peripherals like a debugger, a motion sensor, two push buttons and two USB plugs.

2) Main Board: The Main Board, figure 3, provides physical support to the other modules and the connection between them and the robot's actuators, sensors and battery. Most of the main board is composed of simple routes and planes making these connections. But it also implements some important circuits, such as the currents flow to each motor and the battery voltage.

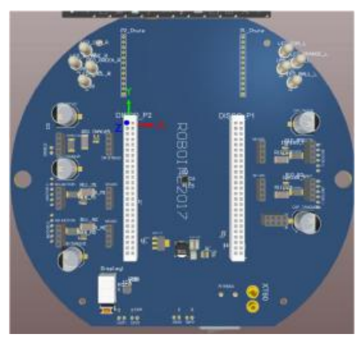

Fig. 3: Main board 3D top view

For 2018's RoboCup the new Main Boards were fabricated adding of more sensors and actuators, but they could not be used during the competition due to issues with the firmware adaptation for the new pinouts and functions of the board. For LARC 2018 those issues were solved and the old boards were completely substituted, even though some of the new features are still not in use.

Along with some component changes and the addition of more sensors and actuators, this board also had some changes in the components position, allowing the robot to carry a new communication module and couple it to an external antenna, in order to reduce the communication losses during the matches.

3) Kicker Module: In 2018 the Kicker Board was completely changed to the new version, which implements a flyback circuit topology, using the LT3750 IC, for charging our 2200μF electrolytic capacitor up to 180V in approximately 5 seconds. For discharging it, this board has a step-up converter circuit, operated by the MC34063 IC, that stores up to 15V inside a tantalum capacitor in order to achieve an appropriate voltage level for triggering the IGBT gate pin.

IV. Mechanical Project

The mechanical project still is developed using CAD (Computer Aided Design) and CAM (Computer Aided Manufacturing) softwares. Therefore, the team members are involved on the conception and on the machining itself. CNC milling, CNC lathe and 3D printing were used for the manufacturing of the new parts.

The participation in the LARC 2018 made it possible to notice many opportunities of improvements of the mechanical project, not only because of the real experience of the matches but also because of the exchange of information with other teams. Next, it will be described the improvements made since the competition at João Pessoa.

A. Omni Wheels

The robots omni wheels have been completely modified. The main objectives of this change were to smooth the robot's movements and use a commercial external gear for the transmission instead of a internal gear. The omni now features two layers of 10 small wheels each (see figure 4). The use of a gear with half the number of teeth and the reduction of the diameter allowed the robot to go faster. The lost of torque was compensated with the effort of making lighter parts.

Fig. 4: Omni Wheel's isometric view

Besides, the main bodies of the omni are now printed, in contrast with the preview version, which were machined in aluminum. That reduction of mechanical resistance was compensated by using a PETG filament reinforced with carbon fiber. The wheels are now much lighter and inexpensive.

Futhermore, the motor supports are not symmetrical to the omnis anymore. This change allowed the motors to be lower and consequently making the supports smaller and creating more room for the kick system (See figure 5)

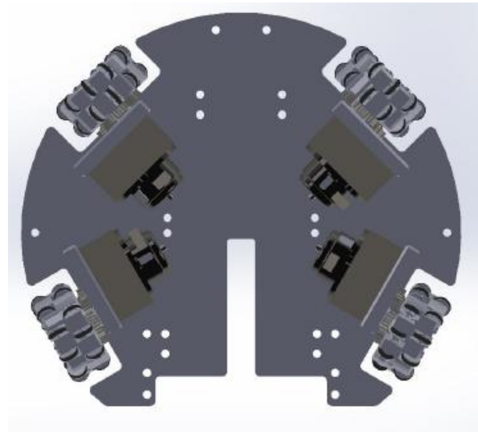

Fig. 5: Omni Wheel's top view

B. The Kick System:

The current kick system has some differences when compared with last version to LARC 2018.

1) Low Kick: The low kick was designed to pass over the high kick plate, in order to optimize the useful space inside the robot. This system consists of a cylindrical solenoid, a piston which has the same geometry, a spring on one of the edges of the piston and the low kick plate on the other edge (figure 6).

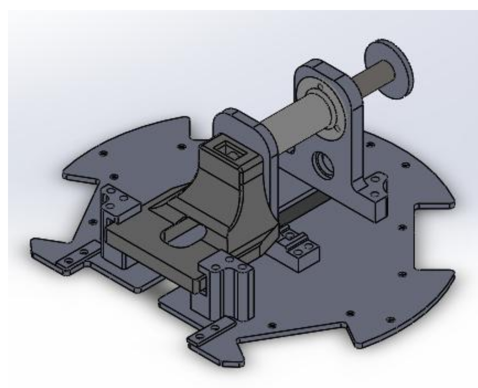

Fig. 6: Low Kick System

While a current is passing through the solenoid, the piston goes ahead and then the low kick plate hit the ball. This movement stretches a spring and when the current finishes, the spring pushes the piston back again.

2) Guide System to Low Kick: There are two guides, on the sides of the low kick plate (figure 6)that aims to help the kick plate to go ahead as straight as it can go and turn the kick more accurate.

3) High Kick: The high kick system has the same activation as the low kick system, but with some differences. The high kick was created to use the lowest part of the robot and its kick plate was designed to pass under the low kick's one (figure 7).

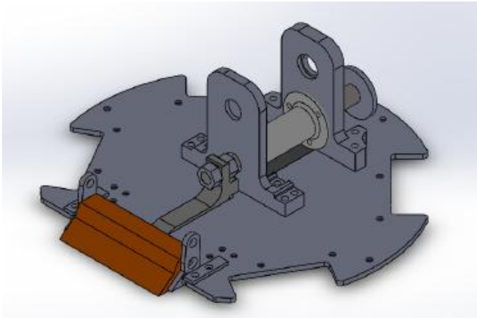

Fig. 7: High Kick System

The piston moves and hits a little plate, which has a ramp on its front. This ramp was designed to cause the ball to move upward.

The two systems has the same activation and piston geometry in order to let the analysis and the modeling simpler.

C. Battery Holder

The battery holder consists of a mechanism which can hold the battery in a specific place. Besides that, it needs to be easy to handle, that means it has to be easy to be placed and removed.

In the modeling, it was kept in mind the best geometry to permit accessibility and do not let the battery free to move when the robot moves around. This project has a cover to the battery and two guides, that will maintain the battery in a certain height and position (See figure 8).

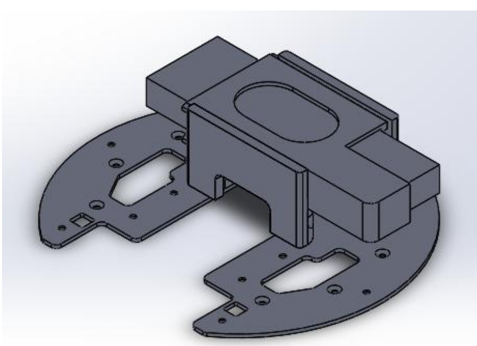

Fig. 8: Battery Holder System

The main idea to use this holder is to reduce the motion's interference when the robot moved and the battery was free.

D. The roller

The roller of the dribbler was made using a 3D printed mold and silicon with a catalyst. Many different molds were made in order to test which was the best format. The ideal roller would be able to guide the ball properly to its center. The current was designed having in mind that the helix pitch has to be smaller than the contact surface of the ball with the roller. In addition, there is also a space so that, in the middle of the roller, the ball is in equilibrium. Hence, the roller is expected to guide the ball satisfactorily towards the center. This will be confirmed through tests that will be done until LARC 2019.

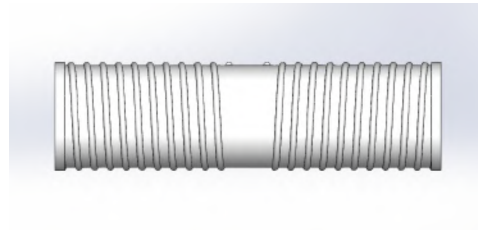

Fig. 9: The roller

V. Conclusions and Future Works

For this competition, RoboIME is aiming into continuing the progress established last year: experimenting a new approach to the software project, modularizing the electrical project and producing more reliable CADs and CAMs in the mechanical project.

A. Acknowledgement

This research was partially supported by the Army's Department of Science and Technology (DCT), Fundação Carlos Chagas Filho de Amparo à Pesquisa do Estado do Rio de Janeiro - FAPERJ(grant E-26/111.362/2012); Fundação Ricardo Franco (FRF), Fábrica de Material de Comunicação e Eletrônica (FMCE/IMBEL) and the companies Altium, SKA and National Instruments. The team also acknowledges the assistance of Clara Luz de S. Santos and Jorge L. de Castilho Jr from FMCE. Special thanks to all former members of RoboIME. Without their support, this team would not be here.

References

[1] M. Bowling M. Veloso B. Browning, J. Bruce. Stp: skills, tactics, and plays for multi-robot control in adversarial environments. pages 1–20. Available at http://papersdb.cs.ualberta.ca/~papersdb/uploaded_files/556/paper_p33.pdf.

[2] Florian Schwanz Gunther Berthold Malte Mauelshagen Tobias Kessler Christian Konig, Daniel Waigand. Tigers mannheim, atificial inteligence. pages 10–17. Available at https://tigers-mannheim.de/download/papers/2011-AI-Structure_Koenig.pdf.

[3] Carla S. Cosenza Gustavo C. K. Couto Luciano de S. Barreira Luis D. P. Farias Luis R. L. Rodrigues Jan L. L. Segre Matheus C. Castro Nicolas S. M. M. de Oliveira Onias C. B. Silveira Renan P. de Souza Victor Bramigk Yugo Nihari Paulo F. F. Rosa. Roboime: on the road to robocup 2019. pages 1–12. Available at https://github.com/roboime/roboime-tdp.