ULtron 2018 Team Description Paper

Simon Bouchard, Francis Lachapelle, Philippe Lebel, Benoit Verret

Université Laval, Québec, Canada,
contact@robocupulaval.com,
http://robocupulaval.com/

Abstract. This paper presents the work done since last year by the team ULtron in order to participate in the international competition in Montréal, Canada. Our new pathfinder algorithm - the *Path-Partitioner* - will be explained as well as our path reshaper which updates the path in real time according to dynamic and kinematics constraints. Next, the new software architecture, which decouple the artificial intelligence and the real world control of the robot. Finally, the new iteration of Delta robots, featuring many improvements, is presented.

Keywords: RoboCup, Small Size League, ULtron, SSL

1   Introduction

ULtron is a team of undergraduate and graduate students at Université Laval in Quebec city, Canada. The team is active since 2013. The team is used to present the project to the public in local events, schools, local fairs and exhibitions to promote robotic and artificial intelligence. This will be the 2nd participation of the ULtron team to an international competition. The team is trying its best to be part of the RoboCup Soccer Family!

2   Software Architecture

There are 3 main modules for controlling the team. The engine, the pathfinder and the AI module. The engine deals with the tracking and the controller of the robot. The AI module takes strategic decisions according to the game state. The pathfinder try to dynamically find a valid path to reach the position wanted by the AI module according to the robot's dynamic and constraints. All the code, which is public, can be found on Github at this URL: https://github.com/RobocupUlaval/StrategyIA.

2.1   Engine

The main task of the Engine is to track the entities on the field and control our robot's position on the field according to the AI decisions. The Engine is implemented in Python with multiprocessing and communicate with the other

Fig. 1. Part of the ULtron 2017 team

processes with message queues. It contains two algorithms: the Tracker, which tracks the entities on the field, and the Controller, which computes the robot commands in accordance with the desired position. The main loop can be summarized as:

1. Tracker module:
   - Get frame from the VisionReceiver's queue;
   - If a frame is available, update the corresponding Kalman filters assigned to the ball and each robot;
   - Predict the next state for each Kalman filter to update the state between the frame;
   - Remove the entities which are undetected for a certain amount of time (robot switch, new ball, etc.);
2. Send the tracked states of the entities to the AI module to update his game state;
3. Controller module:
   - Get message from the AI module to assign the robot target and state (kick, dribbler, speed, etc.);
   - Update the position of our robots in the controller;
   - Execute the PID controller assigned to each of our robot for each state (x, y, orientation);
4. Send the robot's command to the RobotCommandSender's queue and the Tracker for the Kalman filtering.

Note: The VisionReceiver listen to the SSLVision server and store the vision's protobuf packet in a queue. The RobotCommandSender is the interface to communicate with the robot. It sends the commands to a base station (STM32). Then, the base station sends the command to each robot with an nRF.

2.2 Artificial Intelligence

The main task of the Artificial Intelligence module is to make decisions according to the state of the game. It is an adaptation of the STP [1] method for multi-robots control compound with a «Strategy Selector» and a «Role Mapper». The Strategy Selector choose the best strategy according to the current game state. The roles needed for the chosen strategy are assigned by the Role Mapper to the robot. It tries to optimize the assignation according to the position of the robot and the ball on the field. The Goalkeeper role is always set according to the referee interface.

2.3 Deterministic Pathfinder Algorithm

The pathfinder algorithm used to compute the robot's path is based on a geometric approach. The pathfinder algorithm can be divided into two parts. One part simply tries to compute a path that avoids collisions while the other modifies the path to accommodate dynamic constraints and to allow fluid motion of the robots.

While the first part of the algorithm has been developed by ULtron's team, an algorithm that is very similar to ours has already been explained in a STOx's RoboCup 2014 paper. Since some teams approached us in the RoboCup 2017 to describe in depth the algorithm, we plan to present some slides about it at the RoboCup 2018 and publish a portable version of it when time allows.

The second part is the subject this TDP will be mostly about. The *path reshaper* aims at modifying the path to allow target speeds other than 0 at intermediate points of the path.

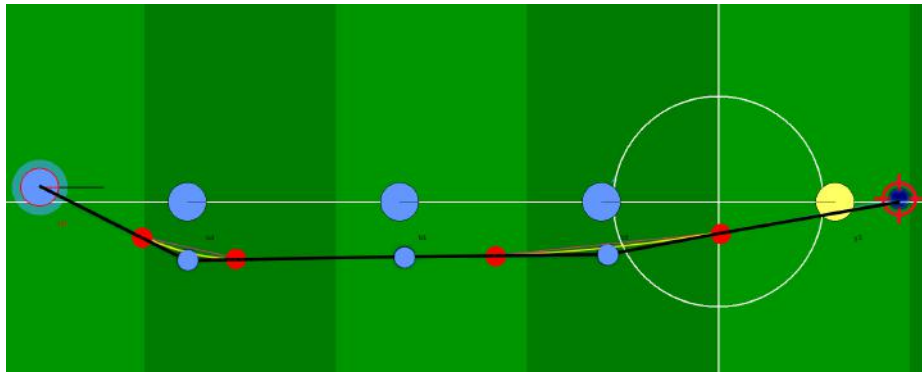The figure 2 shows a basic example of the routine.

Fig. 2. How the *path reshaper* modifies the raw path according to constraints. Blue dots are the original pathfinder points, red dots are the new points. The yellow line indicates the estimated robot's trajectory.

The theory of this *path reshaper* is based from basic kinematics physics. Knowing the maximum acceleration of a robot, we try to fit circular curve segments to the path as in figure 3 using:

$$2\theta = ||\arctan(P_3 - P_2) - \arctan(P_1 - P_2))|| \tag{1}$$

Then, we can incrementally increase the radius of the segment until the maximum deviation distance is reached:
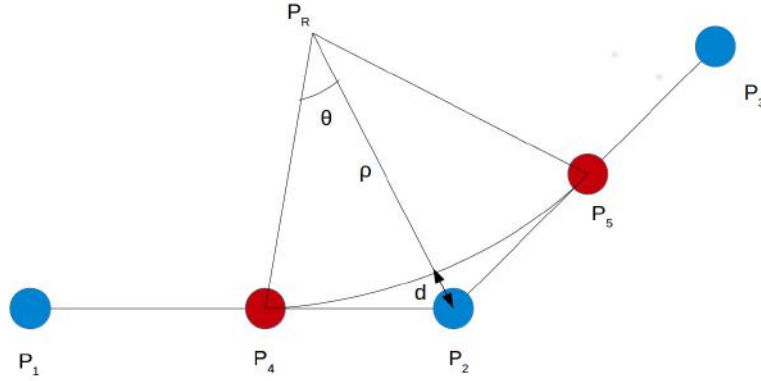
$$d = \frac{\rho}{\sin\theta} - \rho \tag{2}$$

Fig. 3. The curve is positioned according to a maximal deviation d and the points of the initial path P_1, P_2 and P_3.

Once the radius of the segment is computed, the speed at which the robot can enter the turn is computed with:

$$a = \frac{v^2}{\rho}, \tag{3}$$

where a is the robot's maximum acceleration, v is the maximum speed of the robot which takes the turn and ρ is the radius of the segment.

Once the radius of the segment is computed, its location is determined by geometric constraints. If we assume a maximal deviation d from the path, we can compute the location of the points P_4 and P_5 as demonstrated in figure 3.

The location of P_4 and P_5 is given by:

$$P_4 = P_2 + \sqrt[2]{(d + \rho) - \rho^2} \cdot \frac{(P_1 - P_2)}{||(P_1 - P_2)||} \tag{4}$$

$$P_5 = P_2 + \sqrt[2]{(d+\rho) - \rho^2} \cdot \frac{(P_3 - P_2)}{||(P_3 - P_2)||} \tag{5}$$

Once these points and the velocity are computed, the center of rotation must be found. Since the desired trajectory is a circular arc, the robot must accelerate toward the center of rotation to reproduce the wanted behavior. Figure 4 shows the required acceleration through the completion of the path.
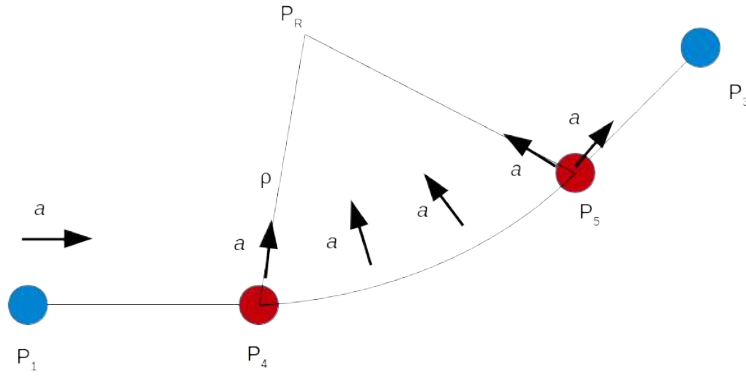

Fig. 4. The robot must accelerate in the direction of a during its displacement.

This center of rotation can be obtained from the equation:

$$P_R = \frac{P_4 + P_5 - 2 \cdot P_2}{2} \cdot \left(1 + \frac{\rho}{d}\right) \tag{6}$$

Since the velocities, positions and directions of the accelerations are known, a trapezoidal speed control (or a more smoothed polynomial) can be implemented to make the robot's motion more efficient.

3   Mechanical Design Upgrades

The team currently works on building a whole new iteration of robots. The Delta model will feature many improvements such as a damper system on the dribbler, a push-pull chip kick and bigger wheels for a smoother displacement. Figure 5 presents this new design.
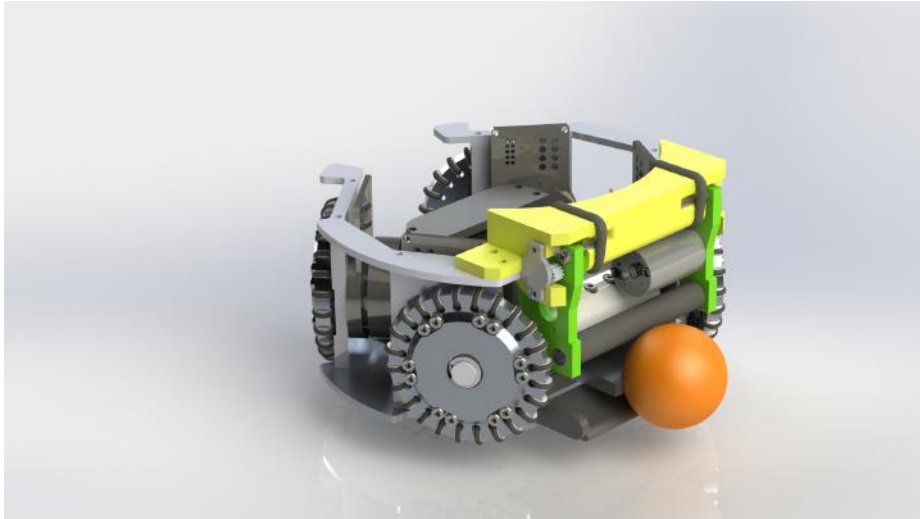

Fig. 5. The new delta robot

3.1   Dribbler System

The previous dribbler presented some problems such as being too narrow, too stiff and had an overall design hard to assemble. The narrowness of the dribbler required a precise positioning of the robots and rendered our kicks difficult to accomplish. Also, the stiffness of the assembly was causing the ball to jump on the dribbler roller. We troubleshot this issue by assembling the dribbler with o-rings, isolating the system from the main structure of the robot. This has the effect of limiting the intensity of the impulsions the dribbler could give back to the ball. To allow a more refined solution, the team had to redesign the dribbler. This new design can be seen in Figure 6.

The damping system consists of a rubber-coated roller mounted on a structure (in green) pivoting on the chassis (yellow). Two o-rings retain the pivoting assembly. The stiffness of these o-rings are chosen empirically according to the desired to damp. Laser sensors are placed on the chassis to read whether the structure pivots or not, allowing to sense if the ball is in contact with the dribbler.
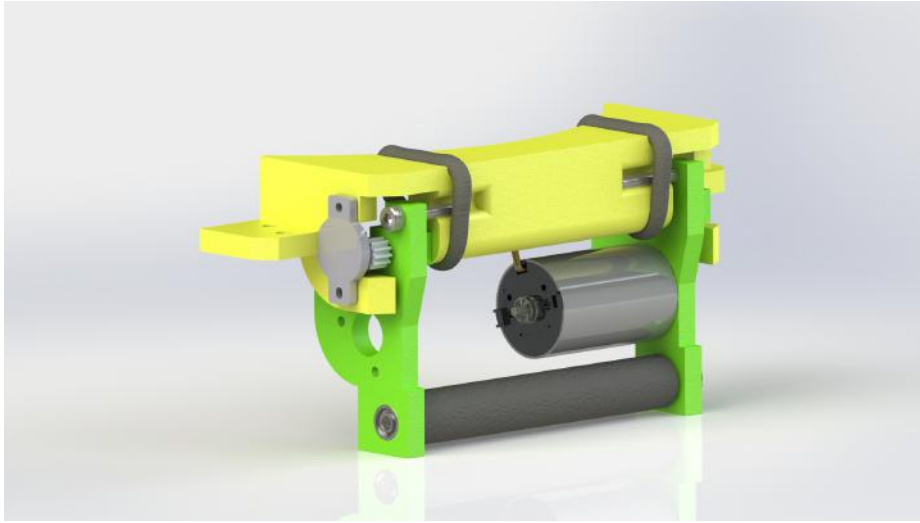
Fig. 6. The new dribbler system

3.2 Push-Pull Kicker

The design of this kicker assembly has been inspired by many current RoboCup teams designs. The chipper will be activated by a coil in a pull configuration. The focus of this concept is to allow both kicking system to utilize the same type of coil and the same dimensions of plungers to simplify the assembly of electrical components. This new kicker is shown in Figure 7.

3.3 Wheels

The Omni-wheels will be 65 mm in diameter. The team wanted to implement this diameter increase to smooth the reading on the encoders. Bigger wheels result in smaller gaps between the perpendicular wheels on the circumference. The team also wanted to make the whole assembly thinner than the previous version. This resulted in machining the internal gear directly in the inside face of the wheel. The concept is shown in depth in 8.

4 Conclusion

A description of the technology used to produce the ULtron robot team has been presented. The team's second competition robots are ready to be used to implement more complex strategies. Currently, the team is working on making
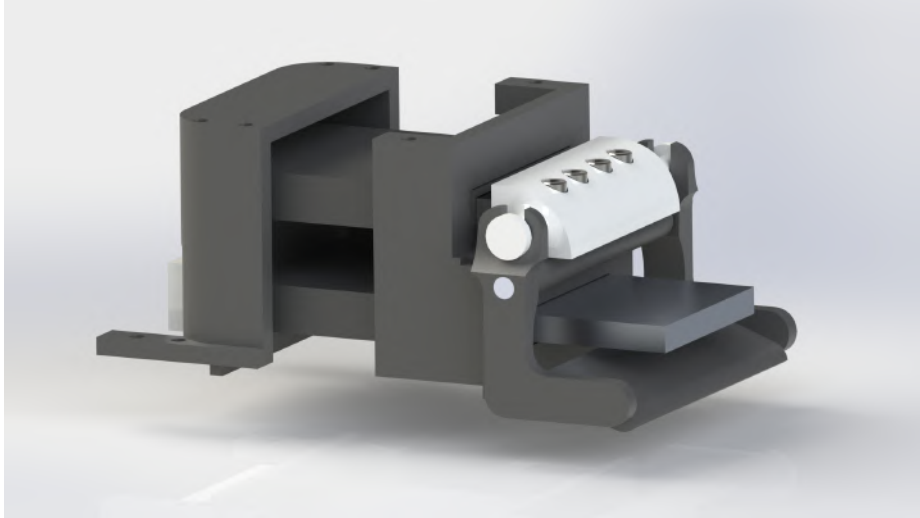
Fig. 7. The new kicker-chipper system

Fig. 8. The new omni-wheel system

the mechanical and electrical system more robust as well as improving the embedded system's control loop and developing an artificial intelligence that can play a full game on its own.

References

1. Browning B., Bruce J., Bowling M., & Veloso M. (2004), STP: Skills, tactics and plays for multi-robot control in adversarial environments. IEEE Journal of Control and Systems Engineering, 219, 33-52.
2. Kalman, R. E. (1960). A new approach to linear filtering and prediction problems. Journal of basic Engineering, 82(1), 35-45.
3. Rodríguez, S., Rojas, E., Pérez, K., López, J, Quintero, C. & Calderón J. (2014). Fast Path Planning Algorithm for the RoboCup Small Size League. Robocup 2014.