Robot Sports Team Description Paper

Ton Peijnenburg[1], Jurge van Eijck[1], et al.

[1] VDL Enabling Technologies Group (VDL ETG), De Schakel 22, 5651 GH Eindhoven,
The Netherlands
Ton.Peijnenburg@vdletg.com, Jurge.van.Eijck@vdletg.com

Abstract. Robot Sports is an open industrial team, meaning that its participants are all employed by or have retired from various high-tech companies in the Dutch Eindhoven region and students. The team will participate with the intent to develop additional skills that must be added to traditional engineering practices for high-end mechatronic equipment to develop autonomous robotic systems, or teams of autonomous robotic systems. Robot development practices from the domain of Artificial Intelligence in turn may be used to improve high-end equipment development effectiveness and efficiency. Most participants' companies currently work on some kind of robotic product..

Keywords: Robot Sports, RoboCup, MSL.

1   Introduction

This year the team will play with new robots, which is an evolution of the previous generation.

The previous generation robots of the Robot Sports Team were developed as a mix of the Philips robot design used in the MSL competition [1], design advancements developed by the Philips team after the last tournament participation and the Tech United TURTLE robot design from the year 2012 [2]. Due to wear and tear a new generation of robots was required.

Based on the multiple years of experience playing in the MSL we designed new robots that are more reliable, better maintainable, safer and cost efficient in transport:

- We changed from three omnidirectional wheels to four.

- We improved the ball handle mechanism and the kicker.

- We utilized the maximum width/height to guard against lob shots

- We made the camera case more sturdy and the mound more flexible for calibration.

- We made a modular design to reduce transport costs, the robots are designed to be transported via a suitcase in a partially disassembled state.

- - With the experiences of the previous generations we decide to focus the re-design of the electronics for the new platform on safety and reliability. This results in:
    - Modules based on their primary functionality
    - Wiring minimized and more reliable connectors resulting in
      - o IO modules connected via ethernet
      - o minimizing the use of USB connectors
    - Start of status monitoring of the various sub systems

For safe testing of the robots an additional module will be added to each robot. Together with a safety button at the side-line all the robots can be stopped immediately via an independent communication channel. However, during a match these additions are not available.

2 Player Hardware

To control the robot, we use a general-purpose Intel NUC with i5 processor and Ethernet. On the NUC we are running Ubuntu 16.04 64-bits. The motion control tasks need strict real time behavior; these will be handled by a separate industrial motion controller.
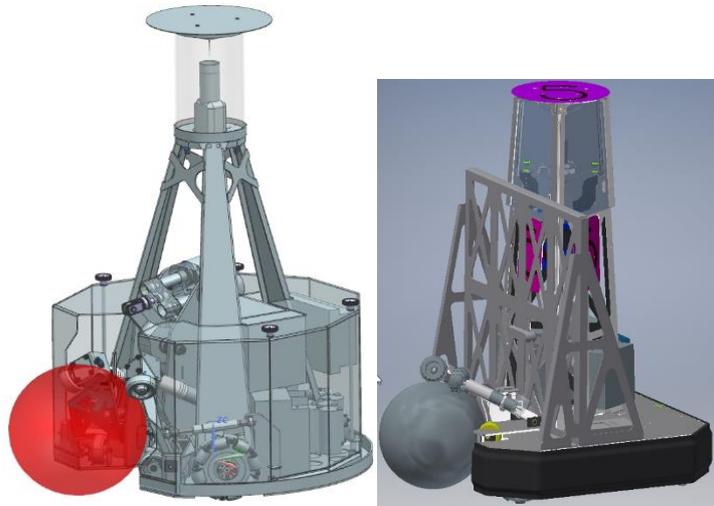
Fig. 1. Robot old generation left, new generation right.

The robot frame is designed entirely in sheet aluminum, which provides rigidity at the lowest possible weight, while keeping cost down.

The robots use a modular industrial motion control system (NYCe4000) that is used to control and power the robot's four drive motors.
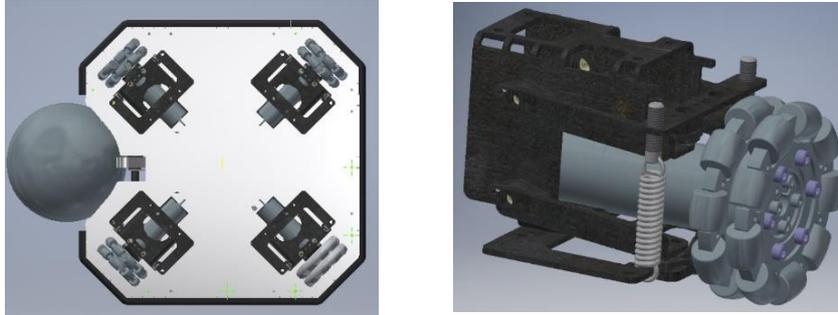
Fig. 2. New drive unit

The NYCe4000 system also controls the kicker and the ball handling actuators, both of which were pioneered by the Philips RoboCup Team [1].

We have an electromagnetic kicking mechanism. Automotive solenoids are used for actuation of a lever. One of two "feet" can be selected which will kick the ball. One foot kicks low over the floor, the other kicks a lob shot.

A capacitor stack is charged; discharge is done through a custom IGBT based switch that can be pulse modulated to control shooting power and -duration. The shoot control is implemented on a Arduino DUE microcontroller that interacts via LAN with the Intel NUC. The capacitor stack charger is a dedicated design focused on safe and fast full charge results. The charge process is measured and controlled by a Teensy 4.0.

3   Sensing

Our robots have a GigE camera from Point Grey with a 1280 x 1024 pixel image sensor. The camera + omni-mirror combination is designed with a compromise in resolution close by and far away. This compromise comes at the cost of some image distortion and a closed-down iris.

From the center of the field the robot camera is able to give an overview of 12 x 18m of the playing field. With the lines at the far corners visible at a distance of 10.8m.

For the self-localization samples from all visible field lines used. The line samples are translated from camera coordinates into robot coordinates and are then matched with hypothesis of field orientations at a resolution equal to the line width in a 2x 1D fashion. Knowledge about the long and short edges of the field is taken into account here. To resolve north south playing field ambiguity, an electronic compass unit is used.

With the camera ball sized object can be detected at up to a distance of 7 meters. Discrimination between a ball and environment is done based on color segmentation in the YUV domain. Color segmentation for field and ball colors is based on (semi) auto calibrated segmentation parameters.

Balls close by will get priority over ball detections far away. Obstacle detection and/or competitor detections are filtered based on shape descriptors like area and elongation. Obstacles inside the playing field and close by will get priority over other obstacle detections.

Clustering of features is achieved using proven algorithms from Tech United, both for ball tracking and player tracking. These algorithms have been re-tuned for our specific sensor characteristics.

In order to improve the environment awareness are using Intel RealSense D435 as replacement of the Microsoft Kinect recently. The Intel Real Sense is equipped with an RGB sensor capable of capturing images at 1920x1080 and a depth sensor capable of generating a 1280x720 depth image, both at 30 Hz.

The Real Sense depth image allows us to map a pair x,y_{img} to x,y,z_{world} coordinates very accurately, which speeds up and improves feature detection since we can filter out the field and the background. This also makes it possible to track aerial balls.

Our Intel RealSense software is able to detect and track multiple balls and robots. It can distinguish between opponents and teammates via searching for colored labels.

By combining the robot and ball detections the software can draw additional conclusions about robots, such as whether the ball handling system of a robot is holding the ball and if so, where it is aiming using basic math. This is for example very useful when defending the goal, because the keeper can calculate the trajectory of the ball. The Real Sense can be used for improving the pass precision because a player can position itself accurately independent of the self-localization accuracy.

4   Reasoning

We believe that the reasoning that is required for soccer should be reactive. Our behavior must react quickly, making a non-optimized but appropriate decision. This is a tradeoff between timing and quality.

The robot behavior is implemented as a set of executable skills. These skills have dedicated responsibilities and effectively run parallel to each other. A domain specific language called *Stactics*. The name *Stactics* comes from state and tactics. It allows to express robot behavior in a higher language.

Stactics consists of two types of skills: action and decision skills. At the highest level, a finite state machine (FSM) is present with fixed transitions between states. It controls the highest-level states of the robot. The *Stactics* FSM is quite similar to Maes' action selection dynamics [4,5]. The FSM decides when and which transition is made. When a transition is made the set of skills that are relevant for that state are made active.

Our robot planner is a variation of the visibility graph [6], which was used on the first general purpose mobile robot Shakey [7], fitted for the soccer domain. On the edges of the created graph (robot planner) by the visibility graph heuristic functions can be added. Via this mechanism opponents can be avoided, while keeping distance to the field boundaries. Restricting the edges to the target vertex and extra costs the approach ball can be influenced. Also, the robot's own velocity vector can be taken into account. Via constraint-based optimization the best path is determined.

The result of the robot planner is a list of x-y points. This describes a rough path. The rough path is used by a movement skill, which smoothens the path and takes velocity and acceleration constraints into account. The skill then sends velocity setpoints

to the motion system of the robot. Rotation skills can in the meantime perform orientation of the robot while driving.

We are using a heuristic based team planner, which uses the robot planner to calculate for every available player a path to an objective, until no players are available. The team planner combines dynamic role assignment and strategic positioning. The dynamic role assignment is made more robust by taking previous assignments into account and allow some hysteresis.

The Robot Sports Team uses RTDB [3] to exchange and synchronize data between team players, which results in a fast and accurate shared world model.

A major change in the team behavior is the change from zone defense to a man-to-man defense. For this feature it is required to select the most dangerous opponent to as player to be defended. The algorithm to determine this opponent, is inspired by the paper on Prioritized Role Assignment for Marking [8].

5   Education and Technology promotion

We have developed an educational environment in which scholars can familiarize themselves in programmatic interaction with a robot. Guided by instruction material, the scholar composes a set of movement instructions by which the robot should move through a given maze.

The scholar defines the movement instruction in the online visual drag-and-drop programming environment Snap! [9]. The programming language has been extended with a number of custom blocks which are able to interact with the control software on our robot. The student is free in choosing his approach: building a complete program and test it, or build it step-by-step. The direct feedback and the interaction with the real world provide a valuable and entertaining exercise.

Even though the maze is relatively simple, the combination of lateral and rotational movement blocks provides the scholar with many alternatives ways to come to a correct solution. Once the scholar grasps the basics, more complex exercises with the same maze and building blocks can be made [10].

Fig. 3. Student testing his solution for the workshop

6   Outlook

In season 2019-2020 we made a new generation of robots, based on the previous generation. That previous platform created a foundation by having all functionality on a workable level. We benchmarked our performance against European teams, in particular the ASML Falcons during monthly practice matches in our shared facility. This brought us to the level where we are able to play soccer. To close the gap to the MSL top teams, we need to make our foundation more robust and at the same time, more advanced. Making the hardware more robust and automating calibrations are key elements. The time from unboxing the team to ready to play must decrease significant without the need of having an expert available. This challenge is similar to the installation and calibration of high-tech production machines in production environments.

Our future generation of robots in our team will be a two-wheeled robot. This will be a cheap platform based on technology of a hoverboard or more particular an Oxboard [11]. The Hoverboard drive and sensing will be the base of the RobotSports 1500. Key features will be a much higher wheel base than the typical MSL robots and a significant cost-price reduction. Speed and outdoor capability has been demonstrated by the Mobile Virtual Players [12]. We are aiming at reducing the cost price to less than 1500 US dollar. In season 2020-2021 we will work on the design of this two-wheeled robot platform.

7   Support

The Robot Sports team is currently supported by the following companies: VDL Enabling Technologies Group and Maxon Motor Benelux. The team shares a dedicated location with the ASML Falcons team in the beautiful city of Veldhoven, near Eindhoven.

8   Conclusion

Although the RoboCup community is currently dominated by university teams, the Robot Sports team will bring its collective background of decades of engineering skills and will strive to find the best match between the findings of modern AI research and new development paradigms from multi-agent system development.

It is expected that industrial design engineers can contribute significantly to new generations of robot systems required for the RoboCup challenge, and secure future support for the challenge by enabling industrial applications.

References

1. A.T.A. Peijnenburg, T.P.H. Warmerdam et.al.: Philips CFT RoboCup Team Description. In: preliminary proceedings 2002 RoboCup conference, July 2002.

2. Turtle Robot description on robotic open platform, http://www.roboticopenplat-form.org/wiki/TURTLE, last accessed: 2020/01/30.

3. Santos, F. , Almeida, L., Pedreiras, P. ; Lopes, L.S.: A real-time distributed software infra-structure for cooperating mobile autonomous robots. In Proceedings of 14th IEEE Interna-tional Conference on Advanced Robotics, Munich, Germany (2009).

4. P. Maes: Artificial Life Meets Entertainment: Interacting with Lifelike Autonomous Agents. In: Comm. ACM, Vol. 38, No. 11, Nov. 1995, pp. 108-114.

5. P. Maes: A bottom-up mechanism for behavior selection in an artificial creature, 1991.

6. Lozano-Pérez, Tomás; Wesley, Michael A. (1979), "An algorithm for planning collision-free paths among polyhedral obstacles", Communications of the ACM 22 (10): 560–570, doi:10.1145/359156.359164.

7. http://www.ai.sri.com/shakey/.

8. MacAlpine P., Stone P. (2017) Prioritized Role Assignment for Marking. In: Behnke S., Sheh R., Sarıel S., Lee D. (eds) RoboCup 2016: Robot World Cup XX. RoboCup 2016. Lecture Notes in Computer Science, vol 9776. Springer, Cham.

9. http://snap.berkeley.edu/.

10. E. Snel, R. Burgers, J. van Eijck and T. Peijnenburg (2020) Interactive Workshops with MSL robots: submitted for RoboCup World Cup XXIV.

11. Oxboard homepage, http://www.oxboard.eu, last accessed: 2020/01/30.

12. Mobile Virtual Player homepage, http://www.mobilevirtualplayer.com, last accessed: 2020/01/30.