Robot Sports Team Description Paper

Ton Peijnenburg[1], Jürge van Eijck[1], Noah van der Meer[2]

[1] VDL ETG, De Schakel 22, 5651 GH Eindhoven, The Netherlands
[2] Eindhoven University of Technology, De Zaale, Eindhoven, The Netherlands
ton.peijnenburg@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, or are active students. The team participates intending 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. Technologies from the domain of Artificial Intelligence in turn may be used to improve high-end equipment and its development effectiveness and efficiency. Most of the participants currently work on robotic products and/or robotic technologies in their products. This year, the team will report on newly designed hardware for their soccer robots, on activities for promoting technology in higher education as well as their progress in applying new technologies in machine vision based on deep neural networks to improve object detection.

Keywords: robotics, machine vision, machine learning, artificial intelligence, motion control, RoboCup, MSL

1 Introduction

The Robot Sports team is an open industrial team supported by the companies VDL Enabling Technologies Group and Maxon Motor Benelux. The team shares a dedicated location with the ASML Falcons team in the city of Veldhoven, near Eindhoven. 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-out of mechanical parts, a revision was required.

2 Robot hardware

The revisions to our robots' hardware are aimed at making them faster, more reliable, easier to service, safer and more efficient to transport. Robots will have four omni-directional wheels for better stability and traction, an improved ball handler mechanism with better placement of passive and active wheels, and a new camera tower that can be separated for transport and provides more easy access for camera adjustments. In addition, the control electronics will be modified to include off-the-shelf motion controllers as well as custom designed, microcontroller based I/O, control and safety modules.

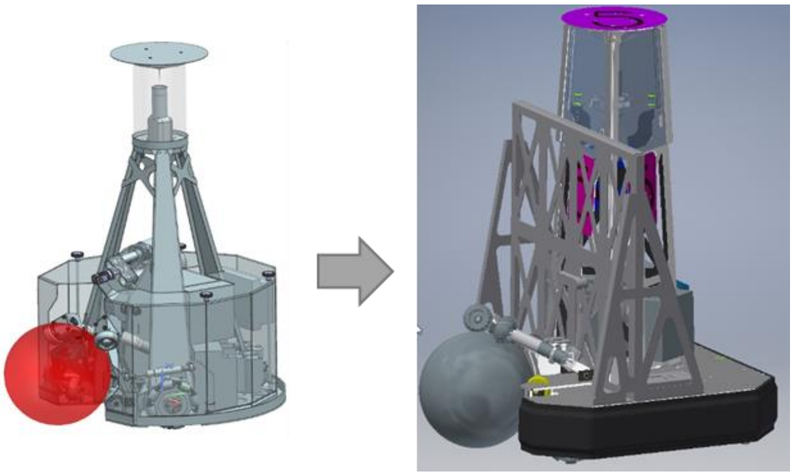

Fig. 1. Our old (left) and new (right) robot.

The robot frame is designed entirely in sheet aluminum, which keeps the weight down while still providing the required sturdiness and keeping cost down. Plate thickness has been increased to 6 mm for the main structural plates, and 3 mm for the other plates. A four-wheel configuration is chosen, combined with individual suspension for all wheels to avoid over-constrained design and secure proper traction. In the first-generation Philips robots, a deformable base plate provided this functionality, but now rigidity of the base plate was considered more important.

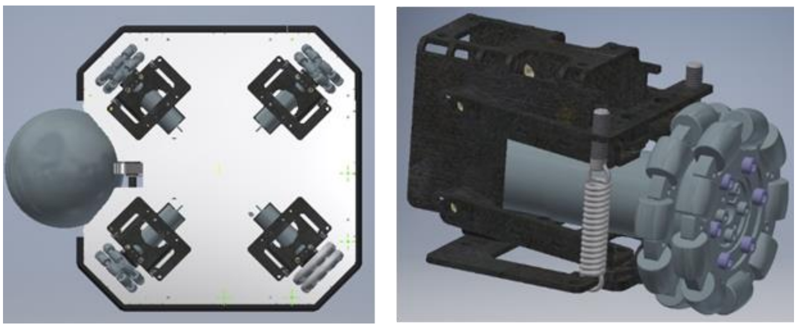

Fig. 2. New wheel configuration (left) and detail of wheel unit with suspension (right).

Robot control is hosted on a general-purpose Intel NUC DC53427 with i5-3427U processor. On the NUC we are running an Ubuntu 20.04 64-bits OS in combination with custom control software. Motion control tasks for the drive- and ball handler wheels are hosted on three dual axis Roboclaw motion controllers [3] which are coordinated by a Teensy 4.0 microcontroller [4]. Interfacing to the main PC is via the LAN Ethernet bus. General I/O control is centralized on a customer board based on a microcontroller which includes PLC functionality for (main) power control, and safety circuits.

An additional 900 MHz RF module can be added to each robot that interfaces with the safety controller to provide remote kill switch functionality which can be used during experimentation and demonstrations. During a match, this functionality is removed since it is not in line with current RoboCup MSL regulations.

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 new charging circuit has been developed to charge a capacitor stack. Discharge is done through a novel custom IGBT based switch that can be pulse modulated to control shooting power and -duration. Control is implemented on a microcontroller that interfaces via LAN Ethernet to the Intel NUC.

3 Visual sensing

Our robots have a GigE camera from Point Grey with a 1280 x 1024pixel 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.

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 considered here. To resolve north south playing field ambiguity, an electronic compass unit is used.

With the camera, a ball sized object can be detected up to 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.

We have recently worked on integrating an additional stereo vision system to supplement the omni-directional vision system installed on the robot. Specifically, we have worked with the ZED2 2K Stereo Depth sensor developed by Stereo-Labs [11]. The ZED2 contains two synchronized high-resolution RGB cameras which can deliver frames at framerates up to 100fps. Through specific calibration and triangulation, these two cameras can be used to estimate the depth of objects present in the image. The ZED2 device can be used to determine distances be-tween 0.5m and 20m with a very tolerable error [12]. The main advantages of using the ZED2 are that the estimation of positions of objects is typically more accurate than what can be achieved using the omni-directional vision system, and the fact that it allows for the detection of airborne objects such as balls passing through the air.

In the past, we have used similar devices such as the Microsoft Kinect to supplement the omni-directional camera with great success. In indoor environments with artificial lighting, these devices that essentially employ active IR projection and detection to estimate depth perform quite well. On the other hand, performance in outdoor conditions under direct illumination from the sun is typically severely limited.

The ZED2 does not suffer from this constraint due to the passive nature of the depth estimation. As the RoboCup community moves closer to its 2050 goal of challenging human opponents, this is highly relevant as it would allow for outdoor matches.

In order to detect objects such as the ball and other robots in the frames delivered by the ZED2 sensor up to high distances, we have worked with Deep Neural Network frameworks such as Tensorflow and PyTorch [13][14]. Recently we achieved very promising results with the YOLO neural network using the latter framework [15]. We are currently also investigating the use of the MobileNetV3 network developed by Google, which is supposed to be particularly suitable for resource-constrained systems [16].

4 Behavior and 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. A finite state machine (FSM) controls the highest-level states of the robot. 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 considered. 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 [5] to exchange and synchronize data be-tween 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 several 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].

We have experimented with this mase in school settings, as well as the more public setting of the Eindhoven Maker Faire. In both settings, students were able to engage well and, in the latter, even involve their parents in explaining them about Snap! Both elementary and secondary schools in The Netherlands use Scratch ([19], which uses similar concepts and is highly like Snap!) in their curricula as to support (basic) education in computer science and skills such as "computational thinking".

Fig. 3. Students testing their solution for the maze during the workshop.

6 Outlook

For a future generation of robots, we are considering two-wheeled robots. We aim for a cost-effective platform based on technology of a hoverboard, e.g., an Oxboard [11]. Key advantages include a much higher wheelbase than the typical MSL robots, creating compatibility with natural sports environments including artificial and natural turf, and the ability to create mixed settings with human players. Speed and outdoor capability have been demonstrated by the so-called Mobile Virtual Player, a remote-controlled platform used to augment professional sports training [12]. After finalizing our current platform revision, we plan to continue our work on the design of this two-wheeled robot platform.

On a shorter term, our team participates with ASML Falcons in a follow-up to the 2020 MSL workshop to define a mixed-team protocol and create a demonstrator for a mixed-team match using robot players from both Falcons and Robot Sports. We consider the mixed-team option as an important element for accelerating innovation by allowing more teams to participate in RoboCup MSL, even with less than five robots, and to make steps towards matches where humans can play with (or against) robots.

The team is currently investigating Behaviour Trees (BT) as replacement for the home brew FSM solution, aiming at more flexibility, faster decisions and improved debugging and replay capability. A BT defines a composition of a set of tasks and the switching between these tasks. It also allows complex tasks to be composed of simple tasks, which matches with the current setup of our software architecture. The implementation is based on the BehaviourTree.CPP library. Groot is used for display and replay.

7 Conclusion

In the previous season 2019-2020 we started with revision of our robots. We extended this into season 2020-2021 due to the COVID19 lockdowns and their impact on team work as well as the cancellation of all RoboCup events. With our previous hardware, we benchmarked our performance against European teams, and specifically the ASML Falcons during our monthly practice matches in our shared facility. This brought us to the level where we are now: we can play a basic level of robot soccer. In order to close the gap to the top teams, we need to make our robots more robust and at the same time, more advanced.

Making the hardware more robust prevents downtime during tournaments and automating calibrations reduce the time we need from unboxing our robots to be ready for a fist match. This challenge is not unlike installation and calibration of high-tech equipment in its production environment. More robust also includes more robust sensing with less dependency on changing environments.

More advanced in our case implies faster motion, better ball control and faster responses. Especially the latter is performance characteristic that has system-wide impact when improving. When improvements for these aspects have been made, more advanced robot and team behavior will become relevant.

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.roboticopenplatform.org/wiki/TURTLE, last accessed: 2020/01/30.
3. Basicmicro RoboClaw Dual 34VDC, https://www.basicmicro.com/motor-controller2, last accessed 2021/03/20.
4. PJRC Teensy 4.0 Development Board, https://www.pjrc.com/store/teensy40.html, last accessed 2021/03/20.
5. Santos, F. , Almeida, L., Pedreiras, P. ; Lopes, L.S.: A real-time distributed software infrastructure for cooperating mobile autonomous robots. In Proceedings of 14th IEEE International Conference on Advanced Robotics, Munich, Germany (2009).
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 and resubmitted for RoboCup World Cup XXV.
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.
13. Stereolabs, “StereoLabs Website,” 2021, accessed: 2021-03-18. [Online]. Available: https://www.stereolabs.com/.
14. Ortiz, L. E., Cabrera, E. V., & Gonçalves, L. M. (2018). Depth data error modeling of the ZED 3D vision sensor from stereolabs. *ELCVIA: electronic letters on computer vision and image analysis*, *17*(1), 0001-15.
15. Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., ... & Zheng, X. (2016). Tensorflow: A system for large-scale machine learning. In *12th {USENIX} symposium on operating systems design and implementation ({OSDI} 16)* (pp. 265-283).
16. Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., ... & Chintala, S. (2019). Pytorch: An imperative style, high-performance deep learning library. *arXiv preprint arXiv:1912.01703*.
17. Bochkovskiy, A., Wang, C. Y., & Liao, H. Y. M. (2020). Yolov4: Optimal speed and accuracy of object detection. *arXiv preprint arXiv:2004.10934*.
18. Howard, A., Sandler, M., Chu, G., Chen, L. C., Chen, B., Tan, M., ... & Adam, H. (2019). Searching for mobilenetv3. In *Proceedings of the IEEE/CVF International Conference on Computer Vision* (pp. 1314-1324).
19. Scratch, project website, https://scratch.mit.edu/, last accessed 2021/03/20.