Robot Sports Team Description Paper 2024

Ton Peijnenburg, Jürge van Eijck[0000−0002−2774−9009], Noah van der Meer[0000−0001−5729−2671], Charel van Hoof, and Rob Burgers

VDL ETG, De Schakel 22, 5651 GH Eindhoven, The Netherlands
ton.peijnenburg@vdletg.com
http://www.robotsports.nl

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. This year, the team upgraded their robots from 2023 to improve serviceability and robustness. Inside the robot, a low-cost motion controller is used. The 2 decades-old software stack has been refactored to remove technical debt and create a healthy foundation for further improvements. The self-localisation is improved to make it more robust and reliable.

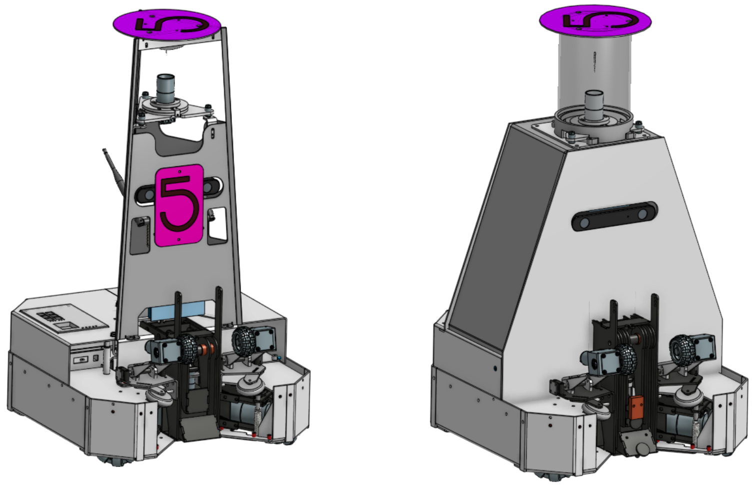
Keywords: robotics · machine vision · machine learning · artificial intelligence · motion control · RoboCup · MSL.

1 Introduction

The Robot Sports team is an open industrial team supported as main sponsor by VDL, an international industrial family business with 105 operating companies, headquartered in Eindhoven the Netherlands. The team shares a dedicated facility with the Falcons team in the city of Veldhoven, near Eindhoven. The team started the Philips RoboCup Team which started participating in 2002 and was renamed to VDL Robot Sports in 2012.

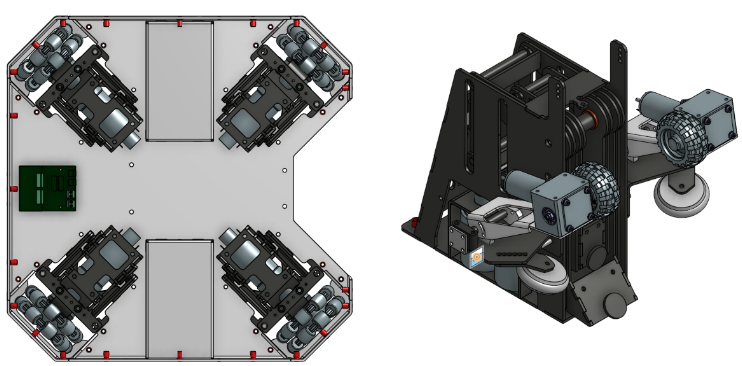
2 Robot design 2024

The design goals of our platform 2023 were to make the robot faster, more reliable, easier to service, safer, more efficient to transport and more cost-effective. We did not meet all requirements with this platform last year, mainly affecting reliability and serviceability, which is why some parts have been redesigned for the 2024 platform. The robots have four omni-directional wheels for better stability and traction, a new ball handler mechanism with passive and active wheels, and a new camera tower that can be separated for transport and provides more easy access for camera adjustments. The control electronics include off-the-shelf motion controllers as well some custom designed, micro-controller based I/O, control and safety modules. A stereo depth sensor camera (ZED2[3]) was added, in combination with CUDA based processing [4] for AI vision purposes.



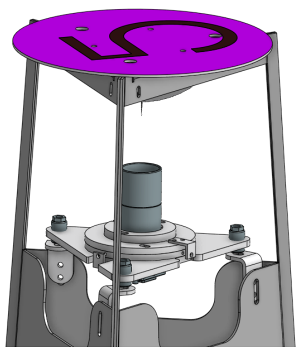
(a) Platform 2023 (b) Platform 2024

The robot frame is designed entirely in sheet aluminum, which keeps the weight down while still providing the required sturdiness and keeps cost down as well. The VDL online metal lasercutting service OrderOn was used to create the aluminium parts in correct dimensions directly from the CAD file. A four-wheel configuration is chosen for stability and increase of drive power, combined with individual suspension for the wheels to avoid over-constrained design and secure proper traction.

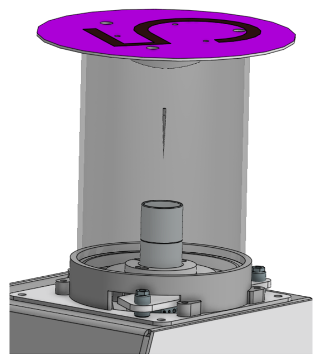


(a) Bottom unit (b) Kicker and ball handler

The platform 2023 was a big step forward compared to the previous generation. The 2023 Bordeaux tournament has been used to get feedback for the desired improvements, o.a. a more robust tower to withstand a direct hit by a ball, and improved access for serviceability, and we reverted to the original tubular camera protection to avoid reflections and avoid obstructing the camera view by bars (see picture below).



(a) Vision tower 2023



(b) Vision tower 2024

Robot control is hosted on an AAeon 8251AI system. The 8251AI brings high performance AI capabilities to the edge in an extremely compact form-factor. In addition, the unit has a small mass, possesses excellent IO facilities and has low power consumption (15W, 6-Core power mode). On the 8251AI we are running an Ubuntu 20.04 64-bits OS in combination with custom control software. The motion control tasks for the drive- and ball handler wheels are assigned to motor drives and a MCU, see section 3. Interfacing to the main computer 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. During demonstrations and experiments an 900 MHz RF module is connected with with the robot's safety controller to provide remote kill switch functionality. 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 AAeon 8251AI.

3 Low-cost motion control

An important objective of our team is to keep our robot cost low. That implies that all design- and redesign decisions have been taken with that objective in mind. Motor drives and motion control can make up a significant fraction of the cost of a robot. When updating our robot, we therefore decided to move from an integrated, industrial grade motion controller, which also offered limited access, to low-cost motion controllers that enable a more accessible design. We chose RoboClaw dual axis controllers from Basicmicro [1]: two boards of 2x30A to drive the four robot wheels, one board of 2x7A to drive the two ball handler wheels. The boards each provide velocity and position control for dual brush DC motors. Contrary to other low-cost solutions, the drives feature a fast quadrature encoder interface (up to 9.8 MHz). This allowed us to reuse our existing Maxon DC motors. Various interfaces are available and we have chosen to use the packet serial interface. A Teensy 4.0 MCU [2] was added for forward and inverse kinematics, as well as monitoring of the drives. Teensy offers a powerful NXP i.MX RT1060 processor with extensive I/O capabilities, packed on a very compact board.

A motion studio application is available to program the drives and tune the controllers for the specific DC motors used. Most settings can also be uploaded from the MCU during operation. Extensive measures have been taken to deal with issues in communication between the drives, MCU and the host. All messages are appended with CRC checks, and availability of the communication links is monitored. Drives are shut down if communication is lost for more than 200 ms, which equals about 8 updates of the robot's main control loop.

The cost of the setup is USD 350 for the motion control system and drives for all motors, excluding the drive motors themselves. Disadvantages of the setup chosen were the need for some custom glue electronics. In addition, there were some synchronization issues between the drive's firmware and the provided software library. These have been resolved quickly by Basicmicro, but did cost significant effort to troubleshoot. See table 1 for an overview of positive and negative aspects of our motion control system.

Table 1: Pros and cons of new motion control system

Positive aspects	Negative aspects
– Motion Studio (Windows only) – Stable and robust drive electronics – Open design, accessible to all developers to fine-tune motion performance – Low-cost	– Sync issues between library and firmware versions for RoboClaw drives – Integrated current measurement is poor for low current values – Need glue electronics and PCB – Development and debugging time

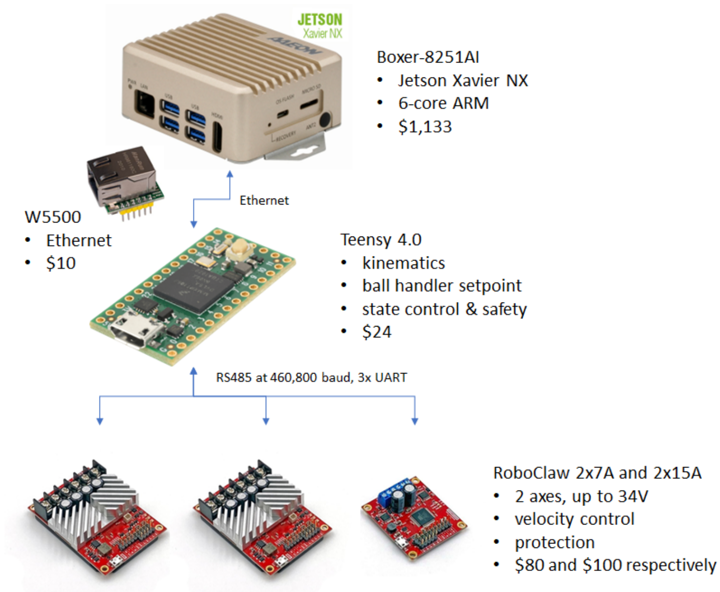


Fig. 1: low cost motion control

4 Technical debt and basic improvements

VDL Robot Sports started in 2000 and has been active in RoboCup since 2002. Over more than 23 years, the software has grown incrementally with contributions by many people. As a result, we have to work with a significant amount of technical debt at the moment, slowing development down significantly. Some symptoms are:

– **Code duplication** - similar calculations are done in multiple locations with subtle variations and outcomes, leading to individual tasks drawing different conclusions e.g. on ball possession status or the position of opponents.
– **Incorrect order of calculations** - aged data is being used while newer data is available.
– **Loss of architecture control** - structure is lost, data paths have become cluttered, older design choices made for different constraints (e.g. determinism) may not be relevant anymore.
– **Lack of code review and unit testing** - errors slip in easily and too few safeguards are in place to maintain consistency.

Even though it was tempting to start with a clean sheet, the decision was taken to clean up the existing code base. This decision was motivated by the familiarity with the code base and the short timeline to the next World Cup. In principle, our software architecture itself is still fit for purpose but needs to be cleaned up to behave in a consistent manner. The requirement of the short timeline was compounded by the available hours that our team could deliver. The steps we are taking are:

- **Re-affirm coding rules**, design patterns and architecture choices.
- **Clean up code** from the ground up, where "ground" is the hardware / software interface and the data flow in the robot control is used as a guide.
- **Improve consistency** in the data, improve assessability of the world model and use it consistently → along data flow.
- **Adding (unit) tests.**
- Step-by-step **offline replay functionality** to aid in trouble-shooting.
- **Adding dedicated test modes** for main robot capabilities to increase reliability when making incremental changes and improvements.

Step-by-step offline replay re-executes the software based on a recording of the input data acquired during a physical play. With the offline replay functionality, the software reproduces the identical internal states and outputs as it did during physical play. A step-by-step mode allows pausing of the software at any given moment e.g. to inspect and analyse the software state. A faster-than-real-time (FTR) mode allows the code to advance to the desired game situation as fast as possible. FTR mode has shown rates of more than 40 times faster than real-time. Real-time (RT) mode runs the software at the same pace as its original. Replay supports on-the-fly switching between step-by-step, FTR and RT mode. As data has been recorded, replay can be done as often as desired.

5 MSL reference architecture

The MSL reference software architecture (MRA) was initiated by Robot Sports. Together with the Falcons, we worked on the first setup of the MRA. During RoboCup 2023 in both teams demonstrated during the technical challenge to swap components between two teams.

The MSL reference architecture is a design based on the commonality of the software architectures of the MSL teams in 2022 [5]. The architecture defines the interfaces of standard components, which fits on the existing MSL software designs. By defining clean interface and responsibilities of the components, the MRA components are designed to be shared and re-used.

MRA components are middle-ware independent, this allows existing teams become a MRA user by replacing a selective part of the software by one or more MRA components. The teams can publish a part of their code as a MRA component, so that is available for other teams with a well defined interface, test-code and documentation. This year, Robot Sports published several MRA components [8]: the proof-is-alive action, local ball tracking and local obstacle tracking. It is currently working on publishing the robot strategy and role assigner functional components. These components are created as part of a refactoring process within the team, the published components are relative complex components. We plan to use MRA components on our robots at RoboCup 2025, for 2024 the team decided to focus on refactoring remove technical depth and basic improvements.

6 Some words on self-localization

Our existing self-localization algorithm heavily relies on correct compass input, causing issues at the World Cup in Bordeaux. Analysis confirmed that feature extraction from the omni-vision camera was good and robust. However this input did not result in a reliable self-localization.

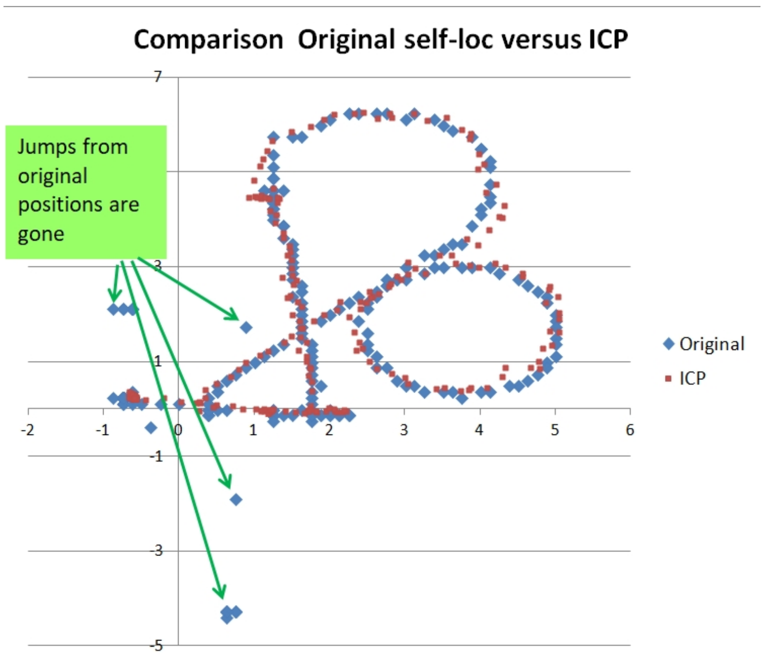


Fig. 2: Comparison between self-localization algorithms

Based on an evaluation of our self-localization approach, and comparison to other team's approaches, three options were identified. Comparison of the three options will give us the much needed improvement:

- Improving our existing algorithm.
- Implementation of the Point Cloud Library Iterative Closest Point algorithm (ICP). This algorithm was successfully implemented and evaluated for the LIDAR system we added to our goalie.
- Implementation of the (OpenCV) downhill solver. This solver is used by the Falcons MSL team. The solver is also published as a MSL Reference Architecture component.

After evaluating the options (including playing some testmatches), the best solution for us will be selected and be used during RoboCup2024.

7 AI object detection

This year, our AI based ball detection running on the image of the front camera will be enabled as default. It will have considerable more weight then the ball detected via the omni-vision camera. Last year our AI based ball detection have been running in the background, we see improvements compare to our original detection.

8 Conclusion

VDL Robotsports has renewed its platform. While we do this, we keep benchmark our performance against other teams. Our shared facility with the ASML Falcons allows us to play regular practice matches against other Dutch teams.

As one of the teams participating for a long time in RoboCup MSL, we have had our focus on keeping the cost of our robot low, by using off-the-shelf components for robot control and cost-effective construction solutions for our frame. During the tournaments, we can see what the trade-off with robot performance is relative to the other teams.

The new platform and refactored software will enable VDL Robotsports to make a step-wise change in soccer performance and close the gap with the top teams, for example with faster motion, better ball control and faster responses.

References

1. Basicmicro RoboClaw Dual 34VDC, https://www.basicmicro.com/motor-controller2, last accessed 2024/02/18.
2. PJRC Teensy 4.0 Development Board, https://www.pjrc.com/store/teensy40.html, last accessed 2024/02/18.
3. Stereolabs, “StereoLabs Website,” 2021, accessed: 2021-03-18. [Online]. Available: https://www.stereolabs.com/.
4. CUDA Zone, https://developer.nvidia.com/cuda-zone, last accessed 2024/02/18.
5. https://msl.robocup.org/history/2022-qualification-results
6. https://github.com/noahmr/zed-yolov5
7. https://github.com/noahmr/yolov5-tensorrt
8. https://github.com/RoboCup-MSL/MRA-components