University of Amsterdam

Dutch Nao Team

Technical Report

December 24, 2022

Students:
Lex Bolt

Derck Prinzhorn

Fyor Klein Gunnewiek

Hidde Lekanne gezegd Deprez

Lasse van Iterson

Supervisor:
Arnoud Visser

Abstract

In this Technical Report, the Dutch Nao Team lists its progress and activities in the past academic year with the previous report [3] as a starting point. Besides new developments this report also lists older developments when relevant.

This year, a new method of permanence was added to the framework by the means of object modelling. Additionally, the implementation of automatic camera exposure calibration increased the robot's ability to detect the ball and other points of interest on the field across various lighting conditions.

In July, the Dutch Nao Team participated in the RoboCup 2022 in Bangkok. After this event, the team decided to switch to a new framework. After considering multiple frameworks, the Dutch Nao Team chose to build further upon the Hulks 2021 code release, which is written in the Rust programming language.

Contents

1 Introduction

The Dutch Nao Team consists of students of different disciplines studying various degrees at the University of Amsterdam: six bachelor's students from Artificial Intelligence and Computer Science and two master's students from Artificial Intelligence. They are supported by a senior staff member, dr. Arnoud Visser. The team was founded in 2010 and competes in the RoboCup Standard Platform League (SPL), which is a robot football league in which all teams compete with identical robots to play football. The league was started to incentivise the development in robot science. Its current goal is to play against the human world champion of 2050, and win.

Since all teams participating in the Standard Platform League are obliged to use identical robots, the focus of the league is solely software oriented rather than hardware oriented. The robots need to be able to play autonomously. This includes finding the ball, locating itself on the field, and making decisions on how to play next, as well as communicating with teammates and being able to walk.

2 Hardware

The NAO robot is a programmable humanoid robot made by Softbank Robotics, formerly known as Aldebaran Robotics. Up until 2018, all versions 4.x and above of the NAO were equipped with the same computational hardware, only differing in their sensors or actuators. The release of the sixth version (V6) introduced a significant change in both hardware and proprietary software.

While the V6 hardware is significantly improved compared to previous versions, CPU resources are still scarce. This limits calculation-heavy tasks such as expensive pixel-wise image operations and deep neural network approaches. The NAO has two high-definition (HD) cameras and an inertial board that are both used for the competition. The HD cameras are located in the head of the NAO; one is aimed forward to look for far away objects and the other is aimed downwards for looking at objects close to the feet of the NAO. The inertial board is used for determining whether the robot has fallen, which happens regularly during matches. The NAO also possesses four sonars, an infrared emitter and receiver, pressure sensors and tactile sensors. Except for the pressure sensors, these sensors are used less frequently than the cameras and inertial board, as they are more prone to breaking down, resulting in faulty measurements. The pressure sensors however become one of the most important sensors, since the current walking engine relies heavily on the pressure information coming from the feet for stable walking.

The NAO robot has 25 degrees of freedom in its joints. The joints in the legs allow for stable bipedal movement and various kicking motions, the arms are generally used for helping the robot stand up again after falling and for stability while walking. It is also permitted for the goalie to hold the ball in its hands, but it is highly uncommon for teams to make use of this. Even though every robot is supposed to be the same, individual differences are noticeable when the robots is playing football. The movement of older robots is less stable and fluent, since the joints of these robots have been worn out. In order to ensure a robust walk for every robot, the joints for each individual robot need to be calibrated. Additionally, each robot's camera can shift inside its enclosure, resulting in a slight offset in the transformation with respect to the robot centre. To correct for this, the cameras can be calibrated.

The V6 is a 64-bit system and has an 1.91 GHz Intel Atom E3845 CPU which is a quad-core processor with one thread per core. The amount of RAM has increased to 4 GB DDR3 and the amount of storage has increased to 32 GB SSD over previous versions. Furthermore, the WiFi connection has improved over the older models and the fingertips are more resistant to impact. With the added computing power, new approaches to issues like localisation and ball detection will be possible. To make communication between the hardware and software possible, the LoLA (Low Level Access) has been used.[1]

[1] `http://doc.aldebaran.com/2-8/family/nao_technical/index_dev_naov6.html`

3 Camera calibration

In order to get the best performance out of our line and object recognition implementations, it is vital to have constant lighting conditions. However, this is not always possible to keep constant. Factors such as: time of day, different sizes of rooms and how close the robot is to a window have a significant impact to how much light reaches the robot's cameras. To help with these variations in lighting conditions the Dutch Nao Team implemented an automatic camera exposure module.

The robot's camera previously calculated the camera settings based on the entire image, which resulted in poor performance when one part of the image was over- or under exposed. Using the green field detection, ball detection and robot detection modules it is possible to weigh parts of the image differently. This gives the option to choose which parts of the image are more important to have proper lighting conditions. Instead of including areas outside of the field in the calculation, the new approach would exclude parts of the image that didn't have the field in them. It also placed a higher importance on parts of the field that had a ball or a robot on them.

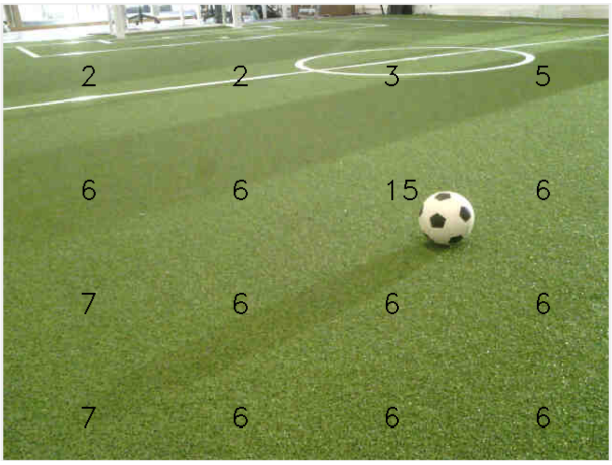

Figure 1: Important sections of the image have a higher weight.

Figure 1 gives an example as to how parts of the image have a difference in weight. The part of the image which contains the ball has the highest weight, whereas parts of the image that are further away from the robot have a lower weight.

In the end the automatic camera exposure didn't make the lighting changes fully automatic, because the system still relies on a goal brightness. This means that we would still have to manually correct a setting for different types of rooms, but slighter changes within rooms like distance to a window had a lot less impact.

4 Object modelling

Before this feature was added, our framework had object permanence only by remembering the last detection for a few seconds. However, with the combination of slow and fast object detection there was a need to be able to combine and denoise object detections. Specifically our YOLO object detector [3] works on cycle of 700 milliseconds and only on the top camera, while our Haar classifier works on a cycle of 17 milliseconds and in both cameras. Because of these differences the module has two main aims, firstly; give greater importance to the YOLO detection whenever it finishes a cycle while still using the Haar classifier for movement tracking, secondly; to account for the time difference between input received and detection verified for object location. Additionally, the topic of detected robot permanence also needed to be tackled.

The goal was achieved by the use of a Kalman filter [4]. The ball is assumed to be stationary, and observations after a certain amount of time are removed from the data. This makes sure the ball position can still change even though we do not do any velocity estimations. Additionally, the YOLO observations get a much higher importance. Since we found out that the YOLO model almost never produces a false positive, we decided to invalidate any Haar detections if they deviate too far from the YOLO detection. Hence the Haar classifier is used to update the ball position slightly. In the bottom camera there is no YOLO detection so no filtering is done for those observations.

The second goal was achieved by keeping location updates of the last second in memory. Then when the robot receives a YOLO detection, the location is recalculated 700 milliseconds into the past according to the odometry information. This recalculation is taking the current location and updates it inversely with the past odometry updates. Important to note is that this excludes any correction mechanisms such as re-localisation and the filtering of particles.

Finally, robot detections are assigned to the nearest modelled robot, or if there are none close by, a new robot is created. This solution only works for general robot locations as the margins for a 700 millisecond cycle are too wide for detecting robots that are right next to eachother. This requires future work.

In conclusion, an object modelling module is implemented which effectively combines object detections from two sources with different time delays and accuracies. This posed challenges but was solved with the use of a Kalman filter, custom detection filtering and time aware algorithms.

5 Gamecontroller

The gamecontroller [2] is the software that controls the state of the game while the robots are playing. It keeps the score, time and can send messages to the robots that contains information about the game. For example, if a robot makes a foul, the gamecontroller will tell that robot to stop moving and wait until it is allowed to return to the game.

For the 2022 Robocup in Bangkok, a new technical challenge was added that introduced 7v7 play. In order for this to work, the Dutch Nao Team and all other teams had to update their framework in order to support 7v7 matches. Furthermore, a new rule [1] was added that limits the total number of messages sent on the network to 1200 packets, and for every minute of irregular extra time, the limit is increased with another 60 packets per minute. In addition to the new rule, there were also some minor changes to the data that has to be sent to the game controller. The following new fields were added to the packets:

- uint8_t fallen - indicates if a robot has fallen
- float pose[3] - pose information containing x, y and theta
- float ballAge - indicates the time since the ball was last seen
- float ball[2] - contains the relative position of the ball to the robot

It is vital for the Dutch Nao Team to stay up to date with every new gamecontroller update. If the robots fail to connect to the game controller it is not possible to participate in the matches.

[2] `https://github.com/RoboCup-SPL/GameController/releases`

6 Results

6.1 Robocup Bangkok

The Dutch Nao Team qualified for the 2022 RoboCup in Bangkok with a video [3] and a paper [2]. This was the first in-person RoboCup since the 2019 RoboCup in Sydney. Therefore it was also the first event since 2019 where the Dutch Nao Team could test all the new modules against real robots from other teams. These new modules contain all aforementioned modules in this report as well as the modules in the 2021 and 2020 technical reports.

In the first round of the playoffs, the Dutch Nao Team beat Naova with a score of 1-0. In the second round, the Dutch Nao Team lost to B-Human with a score of 0-10. In the third round the Dutch Nao Team lost against SPQR with a score of 0-3. In the fourth round the Dutch Nao Team played a draw agains Nomadz, with a score of 0-0. In the fifth and final round the Dutch Nao Team also played a draw against UPennalizers with a score of 0-0. This resulted in the Dutch Nao Team not advancing to the knockout rounds, with a final position of 9th out of the 10 attending teams [4].

The RoboCup in Bangkok also hosted four technical challanges [5], but the Dutch Nao Team competed in none of them.

The members of the Dutch Nao Team who participated in the 2022 RoboCup, as seen from left to right on Figure 2, were:

- Lex Bolt
- Hidde Lekanne gezegd Deprez
- Wike Duivenvoorden
- Jakob Kaiser
- Lasse van Iterson

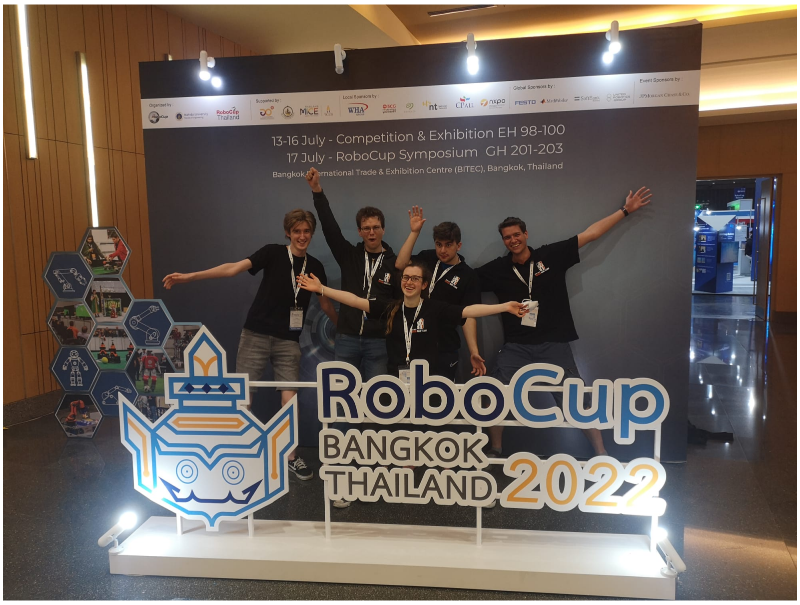

Figure 2: Participating members of the Dutch Nao Team at the 2022 RoboCup.

[3] https://www.youtube.com/watch?v=SsINcoBisJg
[4] https://spl.robocup.org/results-2022/
[5] https://spl.robocup.org/rc2022/

6.2 Foundation

In 2016, the Dutch Nao Team started a foundation in order to be able to receive money from companies in a transparent way. From 2021 untill September of 2022, the board of the foundation consisted of the following members:

- Chair: Wike Duivenvoorden
- Vice-chair: Hidde Lekanne gezegd Deprez
- Secretary: Thomas Wiggers
- Treasurer: Thomas Wiggers

From September 2022 onward, the board of the foundation consists of the following members:

- Chair: Lasse van Iterson
- Secretary: Fyor Klein Gunnewiek
- Treasurer: Lex Bolt

6.3 Public events

The past year, the team put a significant amount of effort into providing demonstrations, lectures and workshops. As the year before, all the funds raised by these events were used to fund the trip to the RoboCup in Bangkok.

In addition to the events scheduled by the team, the University of Amsterdam also invites us to their public events. In doing so, the team normally uses this opportunity to interact with young kids, upcoming students or alumni as a way of educating the public about robots and AI.

This year's events the Dutch Nao Team participated in were:

1. 20-4-2022: Elementary school Voorwegschool demo, where the Dutch Nao Team gave a demo with a robot to elementary school students.
2. 21-4-2022: Careerday, where the Dutch Nao Team had a stand at the careerday in Alkmaar.
3. 9-9-2022: Opening Lab 42, where the Dutch Nao Team showed their new lab to attendees of the Lab 42 opening.
4. 22-9-2022: Opening Lab 42 VIP, where the Dutch Nao Team showed their new lab to attendees of the Lab 42 VIP opening.
5. 27-10-2022: Elementary school Texel demo, where the Dutch Nao Team gave a demo with a robot to elementary school students.
6. 15-11-2022: Guest lecture Robotics, where members of the Dutch Nao Team gave a guest lecture about robotics.
7. 18-11-2022: Open Campus Day, where the Dutch Nao Team had a stand to show their work at the University of Amsterdam to potential new students.

7 Plans for 2023

7.1 Exploring a new framework

Starting from September, the Dutch Nao Team saw an influx of new members joining the team, which resulted in 23 active members as of December 2022. In September, the goal for the Dutch Nao Team was to explore a potential switch to a framework of another team and consider building on one of those rather than continuing with our DNT2017 [5] framework. The team was considering this change because the DNT2017 framework was getting old and cumbersome to work with and vital modules like the walking engine and localisation required major redesign. The team considered updating our own framework, but it also seemed like a valid option to explore frameworks of other teams and build further upon those. We explored Runswift[6], B-Human[7], Berlin United [8], Hulks C++ 2019[9] and Hulks Rust 2021[10]. The team ended up choosing to build further upon the Hulks 2021 code release, which is written in Rust.

[6] https://github.com/UNSWComputing/rUNSWift-2019-release
[7] https://github.com/bhuman/BHumanCodeRelease
[8] https://github.com/BerlinUnited/NaoTH
[9] https://github.com/HULKs/hulk/releases/tag/coderelease2019
[10] https://github.com/HULKs/hulk/releases/tag/coderelease2021

7.2 Rust

Since the Dutch Nao Team has become substantially larger in member count, the skills of the team grew aswell. Probably the most important example are the new software engineering skills in the team. We now have four experienced software engineers who all have a preference for programming in Rust instead of C++. This is mostly due to Rust being a very stable programming language, especially for a larger team, because writing and mostly maintaining the code is easier. In Rust, you have to write code explicitly rather than implicitly. In addition to this the project structure in Rust makes more sense and the package manager is also better. After a discussion, the team decided to go with the Hulks Rust framework. The team knew this was going to be a challenge, since the majority of the team didn't have experience with programming in Rust. Therefore the Dutch Nao Team created a roadmap for team members to self-study Rust.

7.3 Goals for 2023

With some old members leaving the team and many new members joining, the next challenge was to get the team started on working with the new framework. The Framework Team consisting of a team of software engineers focused on getting the Hulks Rust framework working on our robots. This took some time, but eventually they managed to build and deploy the new framework successfully. After this the rest of the team could get started on projects, for which the teams had to be created. We first set general milestones: changing a Hulks module, creating a new Hulks module. And later we made milestones a bit more specific. We had a brainstorm meeting about what we wanted to focus on and this resulted in the following topics:

- Ball detection
- Damage Prevention
- Strategy
- Localisation
- Teamplay (communication)

Afterwards, the team created project proposals for every topic, inventoried everyone's interests and created subteams on each project to get everyone started. Ball detection and damage prevention are expected to be implemented and finished on a short-term basis. Localisation and teamplay on the other hand will probably take more time. Strategy is rather a long-term goal for which the current projects mostly focus on preparing for real strategy implementations by getting simulations working.

7.4 New Framework from scratch

Since the Dutch Nao Team is larger in size and expects everyone to develop skills in AI and Rust, it is expected that the team is able to accept a challenge. While the rest of the team is gaining experience in creating features for the Hulks Rust framework, the Framework team is already busy building a new framework from scratch in Rust. When there is a basic but stable skeleton, the feature team will be ready to implement features in the new framework.

8 Contributions

The following list credits the people who worked on the additions mentioned in this report, in alphabetical order:

- **Derck Prinzhorn**, who wrote about the Dutch Nao Team's plans for 2023 and the switch to the new framework in Rust.
- **Fyor Klein Gunnewiek**, who worked on the automatic camera calibration.
- **Hidde Lekanne gezegd Deprez**, who worked on the ball object modelling module.
- **Lasse van Iterson**, who worked on updating the gamecontroller to the new standard.
- **Lex Bolt**, who organized and compiled the report.

9 Conclusion

This year, the Dutch Nao Team implemented two final modules for the old DNT2017 framework, namely the object modelling and automatic camera calibration module. The focus of the team will now lie on developing a new framework based upon the Hulks 2021 framework in Rust. The Dutch Nao Team is determined to learn Rust and implement new features that will improve our results at the following RoboCups.

References

[1] RoboCup Technical Committee. *RoboCup Standard Platform League (NAO) Rule Book.* 2022. URL: `https://spl.robocup.org/wp-content/uploads/SPL-Rules-2022.pdf` (visited on 12/19/2022).

[2] Wike Duivenvoorden et al. *Team Qualification Document for RoboCup 2022 Bangkok, Thailand.* 2022. URL: `https://www.dutchnaoteam.nl/wp-content/uploads/2022/02/DNTQualificationDocument2022.pdf` (visited on 12/19/2022).

[3] Jakob Kaiser et al. *Dutch Nao Team - Technical Report.* 2021. URL: `https://www.dutchnaoteam.nl/wp-content/uploads/2022/01/dnt_techreport_2021.pdf` (visited on 12/08/2022).

[4] Rudolph Emil Kalman. "A New Approach to Linear Filtering and Prediction Problems". In: *Transactions of the ASME–Journal of Basic Engineering* 82.Series D (1960), pp. 35–45.

[5] Douwe van der Wal, Pieter Kronemeijer, and Caitlin Lagrand. *Dutch Nao Team - Technical Report.* 2017. URL: `http://www.dutchnaoteam.nl/wp-content/uploads/2018/01/dnt_techreport_2017.pdf` (visited on 12/22/2022).