Team Qualification Document for RoboCup 2020, Bordeaux, France

Dutch Nao Team; Lekanne gezegd Deprez, H.; Kronemeijer, P.; Lagrand, Caitlin; Wiggers, T.; van der Wal, D.; Coltof, Q.; Zwerink, W.; Visser, A.

Publication date
2020

Document Version
Final published version

Citation for published version (APA):

General rights
It is not permitted to download or to forward/distribute the text or part of it without the consent of the author(s) and/or copyright holder(s), other than for strictly personal, individual use, unless the work is under an open content license (like Creative Commons).

Disclaimer/Complaints regulations
If you believe that digital publication of certain material infringes any of your rights or (privacy) interests, please let the Library know, stating your reasons. In case of a legitimate complaint, the Library will make the material inaccessible and/or remove it from the website. Please Ask the Library: https://uba.uva.nl/en/contact, or a letter to: Library of the University of Amsterdam, Secretariat, Singel 425, 1012 WP Amsterdam, The Netherlands. You will be contacted as soon as possible.
1 Team Information

This is the qualification document for the Dutch Nao Team with Hidde Lekanne gezegd Deprez as its team leader. The team consists of four master students, five bachelor students, one alumnus and one staff member from the University of Amsterdam, The Netherlands. In the last nine years the team has bought 23 NAO robots, although not all of them are operational anymore. The team currently has four NAO V6 robots, but has the intention to buy at least two more before the upcoming RoboCup. The qualification video is available on our YouTube channel\(^1\). A research report \([1]\) describing the technical details of the team’s work for RoboCup 2019, has previously been published on our website\(^2\).

2 Code Usage

In 2016, the team started implementing a custom framework based on ROS\(^3\) \([2]\). However, the team did not enter the competition with this framework since experiments showed that running a ROS core node and image publishing nodelet results in a frame rate of approximately 5 Hz, without any further processing. This was deemed too low for usage in RoboCup competition, so in the 2016

---

1 https://youtu.be/SKawcYDEhjo
3 http://www.ros.org
RoboCup competitions *B-Human’s* 2015 code release\(^4\) was used, extended with our own behaviour engine \([3]\) and ball detector \([4]\).

From April 2017 onward, the team has been using its own framework. The decision to start a new framework was made to provide the team with a codebase it fully understands and is documented in a way that is understandable for all members of the team, new and old. By creating its own framework, the team has gotten a better understanding of all components required to go from sensor values to high level actions. Ultimately, every team member should be able to largely understand its inner workings and feel comfortable with it. Our new framework is based on messages sent between modules, where each module represents one algorithm handling a task in the football playing robot. Each message shared between modules contains a representation. The system uses a message naming convention comparable to the ROS messaging system, which makes it easy to use for developers that have some experience with ROS.

So far, the team has noticed that despite the obvious drawbacks of having to recreate basic functionality, the educational value of our new framework has increased the motivation of (newer) team members and has had a positive impact on the overall productivity. A code release is planned when all basic functionality has been implemented and tested thoroughly.

A stripped version of the walking engine of BHuman\(^5\) based on rUNSWift’s walking engine\(^6\) has been integrated into the framework because of its proven stability and the large impact of walking on overall performance.

BHuman’s CABSL \([5]\) is planned to be used in our framework as behaviour engine because of its simplicity and clear design paradigm. This will replace the previous implementation of a behaviour engine \([3]\), which has become increasingly complex to use as the number of modules and behaviours in the system increases. The current implementation does not have states, but assigns a score to every behaviour based on the situation the robot is in. The weighting between decisions is hard to tune properly, and this is mitigated by using if-else statements, effectively turning it into a finite state automaton (FSA). CABSL also works like a FSA, and therefore seems better suited as behaviour engine.

### 3 Past History

The predecessor of the Dutch Nao Team was the Dutch Aibo Team \([6]\). The Dutch Nao Team debuted in the Standard Platform League (SPL) competition at the German Open 2010 \([7]\). Since their founding, the Dutch Nao Team has been qualified for the world cup competitions in Istanbul \([8]\), Mexico City \([9]\), Eindhoven \([10]\), João Pessoa \([11]\), Leipzig \([2]\), Nagoya \([12]\), Montreal \([13]\), and Sydney \([14]\).

Besides the major RoboCup events, we have attended multiple GermanOpens, IranOpens, the Humanoid Soccer School 2013, the Mediterranean Open 2011, the Colombia Robotics week, TechFest 2015\(^7\), the European Open 2016, Rodeo 2019 and every Robotic Hamburg Open Workshop between 2016 and 2019. At the Benelux Conference on Artificial Intelligence 2016 the team received the award for best demonstration \([15]\), at the Iran Open 2017 the team received the Award in the Open Challenge with a presentation on our behaviour engine.

\(^1\) [B-Human Code Release](https://github.com/bhuman/BHumanCodeRelease/tree/1fd87519e2bbb3ccbb5f28880438b692629f7c1)


\(^3\) [rUNSWift 2016 release](https://github.com/UNSWComputing/rUNSWift-2016-release)

\(^4\) [TechFest](http://techfest.org)
The results from 2018 onward in major RoboCup competitions are presented in Table 1a. In Montreal, we ended second in our first round robin pool and fourth in our second round robin pool, and in Sydney we were able to score twice in-game and promoted to the champions cup second round robin by beating Camellia Dragons in a penalty shootout. Table 1b shows the scores for the open competitions.

<table>
<thead>
<tr>
<th>Year</th>
<th>Round</th>
<th>Opponent</th>
<th>Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>2018</td>
<td>Round Robin</td>
<td>Aztlan</td>
<td>0:0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Naova</td>
<td>0:0</td>
</tr>
<tr>
<td></td>
<td>Second round</td>
<td>NTU RoboPAL</td>
<td>0:2</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Naova</td>
<td>0:2</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Unbeatable</td>
<td>0:0</td>
</tr>
<tr>
<td>2019</td>
<td>Round Robin</td>
<td>Starkit</td>
<td>2:0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>RoboEireann</td>
<td>0:0</td>
</tr>
<tr>
<td></td>
<td>Champions cup</td>
<td>Camellia Dragons</td>
<td>0:0[1:0]</td>
</tr>
<tr>
<td></td>
<td>Champions</td>
<td>Camellia Dragons</td>
<td>0:0[1:0]</td>
</tr>
<tr>
<td></td>
<td>play-in round</td>
<td>TJArk</td>
<td>0:6</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Nao Devils</td>
<td>0:9</td>
</tr>
<tr>
<td></td>
<td>Champions</td>
<td>UT Austin Villa</td>
<td>0:7</td>
</tr>
</tbody>
</table>

(a) Game scores for RoboCup 2018 and 2019.

Table 1: Game scores for the Dutch Nao Team in different competitions.

Although not visible in the scores, the field play has improved a lot, resulting in games with a lot of ball possession. Yet, without localisation scoring is difficult. The Dutch Nao Team will come well prepared to the competition in Bordeaux: in December 2019 the Dutch Nao Team attended the RoHOW\(^8\), and we are planning to participate in GermanOpen 2020\(^9\).

4 Impact

During the participation in the RoboCup, the Dutch Nao Team has provided its support or resources in a number of bachelor & master theses \([16,17,18,19]\) and projects that lead to publications on a large variety of topics \([20,21]\). At the Maastricht University, a PhD thesis is finished \([22]\) based on e.g. a paper on learning a more stable gait \([23]\), compared to the energy efficient gait from earlier work \([24]\). In an honours project a comparison was made on ball detection with classical image processing versus modern deep learning techniques \([25]\). The Dutch Nao Team extended the application of the Nao robot to the @Home league of the RoboCup: the Nao robot was used to help in a kitchen environment by finding a tomato and grabbing it from a table \([26,20]\). Finally, the Dutch Nao Team has made the penalty shootout situation into a standalone demonstration \([15]\).

\(^8\) See https://www.rohow.de/2019/en/teams.html

\(^9\) https://robocupgermanopen.de/en/major/teaminfos
which it premiered at the Benelux Conference on Artificial Intelligence 2016\(^\text{10}\) and won the first prize for best demonstration.

Earlier the Dutch Nao Team has published papers in the International Conference on Advanced Robotics [27], the Performance Metrics for Intelligent Systems Workshop [28], the RoboCup IranOpen Symposium [29], the RoboCup Symposium [30] and the international conferences as International Conference on Autonomous Robot Systems and Competitions [26]. The Dutch Nao Team also proposed and supervised RoboCup related projects as part of a compulsory course in the Artificial Intelligence bachelor at the University of Amsterdam.

Over the last years, our framework has grown quickly and has gotten to a point where most basic modules to play football properly are working. However, due to the necessity of those basic modules, some parts of the framework that could use improvement did not yet receive the attention they deserve. This year we worked on refining behaviours, improving load balancing between threads and supporting different operating systems to allow an easier introduction into the team. More importantly, we worked on compiling in 64 bits and integrated camera settings in the framework, allowing us to do camera calibration. The lack of camera calibration was a significant issue when playing with a low light level, as seen during the RoboCup in Sydney. Furthermore, for the coming RoboCup the team plans to develop the following techniques:

### 4.1 Scanlines

Last year the Dutch Nao Team implemented a line detector using the green chromaticity channel to filter out the field. While this line detector performed very well on nearby field lines, it often failed to detect field lines at the top of the image. This happened due to naive downsampling, which skipped every fourth row and column.

To improve localisation, the far away field lines needed to be detected as well. For this reason, the team implemented scanlines as an alternative to the naive downsampling. Our scanlines module determines the number of columns and rows that can be skipped dynamically within the image. This results in a high resolution at the top of the image and a low resolution at the bottom of the image, due to the respective sizes of field lines in the image. Additionally the resolution changes with the angle of the camera, to keep the amount of pixels used for line detection as low as possible while still viewing enough pixels to detect every line.

### 4.2 Perception

**Ball Detector** With the transition from the light grey V5 robots to the dark grey V6 robots, our ball detector started to detect more false positives in other robots. To improve our current method, a cascade classifier, we plan to compile a Deep Neural network library, such as TensorFlow\(^\text{31}\), for the NAO V6. Such a library enables us to easily develop, train and evaluate a model. The current region of interest selection procedure has a high recall and is fast, but the classifier is not accurate. By using TensorFlow, we can leverage optimisations we previously did not have access to and create a classifier that is both accurate and fast enough to be used on the robot.

**Ball Model** Besides improving the ball detector, we have started to work on a ball model. This model should interpret the detections from the ball detector, and judge whether they are likely to be the actual ball or not.

\(^{10}\) http://bnaic2016.cs.vu.nl
To implement this, we considered two solutions. The first is the simplest approach, using a simple threshold. If the number of ball detection’s in the same region is not larger than the threshold, the position of the ball model is not updated to these new detection’s. This attempts to filter out erratic false positives, as the actual ball is often detected multiple times. The second approach is to use Kalman filtering [32]. This is a much more sophisticated approach which is very effective in reducing the noise using multiple measurements.

However, when testing these techniques, we encountered difficulties with validating the implementation and tuning the parameters. Because we have no ground truth data available, we cannot properly evaluate the effectiveness of these solutions. Therefore we have not yet been able to successfully use these techniques.

**Middle Circle Detection** To improve localisation we have created a middle circle detector, this gives us very few options on the position of the robot and this can be done with almost no false positives.

This is done by transforming all detected lines to a top down view of the field. Then the middle circle is detected by trying to create a circle between all combinations of the begin and end points of all the lines. A lot of work is skipped this way because most possible circles are filtered immediately, this is so efficient that an approach such as RANSAC[33] is not required. The best found circle is then determined by checking which has the most points closest to the theoretical circle.

### 4.3 Localisation

Last year we developed localisation which enabled us to walk to the ready position and score from there. However, we frequently lost our location due to falling or robots obstructing the view. Therefore, to make the module more robust, we plan to extend the localisation module by adding re-localisation. We are working on integrating newly detected field features, such as the middle circle and corners, into our localisation module. This would make it possible to achieve better re-localisation and more precise estimation of the robots position on the field.

### 4.4 Interface

To improve the teams ability to evaluate the ball detector, we added a neural net to the interface which calculates ball detections locally on a laptop taking the camera-feed from a NAO as input. By running this neural net locally, it can be far larger allowing for more general and accurate predictions. This enables the team to see the difference between a ball-detection form the robot and a more accura detection from the local computer.

Additionally the ability to replay previously recorded sensor data has been added last year. Together with the previously mentioned module, this greatly enhances the teams ability to evaluate new ball detection modules.

### 4.5 NAO Version 6

The University of Amsterdam has given the Dutch Nao Team access to four NAO V6 robots, with more to come. This robot has multiple changes compared to the older versions, such as different hardware and changes to the NAOqi framework. Our framework has since been updated to be fully
compatible with these changes, deprecating V5 systems in the process. Before the RoboCup, we hope to exploit the faster hardware of the NAO V6 robot even more, by integrating deep learning in perception modules.

5 Other

For the broader community, the Dutch Nao Team continues to provide many lectures about robotics and AI, and demonstrations of autonomous football at companies and schools throughout the year. This spreads knowledge about robotics and AI, and is a way for the Dutch Nao Team to fund the trip to the RoboCup. After RoboCup 2016 a foundation was started to allow for transparent financial communication, solely for the benefit of AI and robotics research.

References


25. Lagrand, C., van der Wal, D., Kronemeijer, P.: Detecting a checkered black and white football. honour’s project report, Universiteit van Amsterdam (February 2017)


