NimbRo AdultSize Team Description 2018

Hafez Farazi, Philipp Allgeuer, Grzegorz Ficht, Dmytro Pavlichenko, Diego Rodriguez, and Sven Behnke

Rheinische Friedrich-Wilhelms-Universität Bonn
Computer Science Institute VI, Autonomous Intelligent Systems
Friedrich-Ebert-Allee 144, 53113 Bonn, Germany

{ farazi | pallgeuer | ficht | behnke } @ ais.uni-bonn.de
http://www.nimbro.net

Abstract. This paper describes the RoboCup Humanoid League team NimbRo AdultSize of Rheinische Friedrich-Wilhelms-Universität Bonn, Germany, as required by the RoboCup qualification procedure for the competition held from June 16–21, 2018 in Montreal, Canada. Our team uses self-constructed robots for playing soccer. This paper describes the software and hardware aspects of the robots, and highlights some of our scientific achievements, including in particular in areas of perception, motion, and localization without a compass on a soccer field of symmetric appearance.

1 Introduction

With the ongoing evolution of its rules, The RoboCup Humanoid League competition each year is becoming more realistic and closer to playing games against human soccer players. In 2017, for the first time, AdultSize class robots were

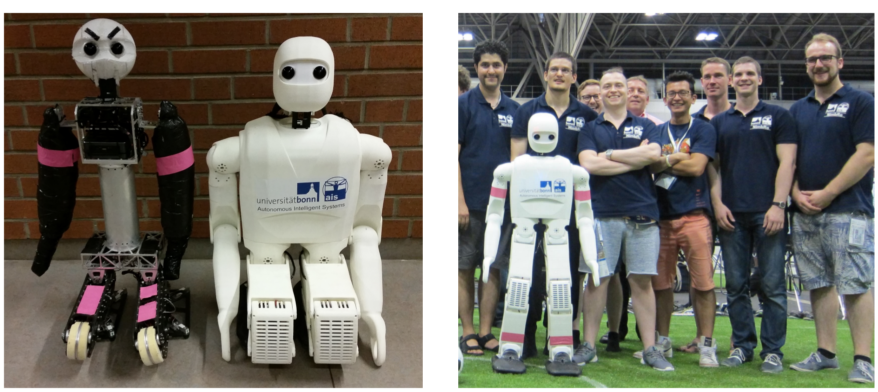

Fig. 1. Left: Our AdultSize Robots Copedo and the NimbRo-OP2. Right: The NimbRo team at RoboCup 2017 in Nagoya, Japan.

playing one vs. one matches, which meant bringing the soccer game to a larger scale, along with all of its inherent difficulties.

Over the past years team NimbRo has had a large number of successes in the RoboCup competitions. Most recently, in 2017, team NimbRo AdultSize and NimbRo TeenSize won the regular league competitions, as well as the technical challenges in both leagues [10] [6]. Furthermore, in 2017 we won the TeenSize Drop-In Challenge, and our AdultSize NimbRo-OP2 platform (see Fig. 1) won the RoboCup Design Award. In AdultSize, the combined score result was 46:1 over the five games that were played. Further back, between 2009 and 2013, team NimbRo also won the competitive Humanoid TeenSize soccer tournament five times in a row, also winning the technical challenges in the years 2012 and 2014. In 2015, the igus® Humanoid Open Platform, which was developed in cooperation with igus GmbH, had its RoboCup debut, where it won the RoboCup Design Award. The following year, the igus® Humanoid Open Platform project, which is open-source in terms of both hardware and software, was awarded the first International HARTING Open Source Prize [1] and won TeenSize league [5].

In the 2018 competition, we want to display the advancements made in our open-source ROS framework, e.g. in the areas of perception, localization, walking, and soccer behaviors. At this point, our software includes no software modules from other teams. We also shared all of our recent software and hardware designs with the RoboCup community.

2 Mechanical and Electrical Design

2.1 NimbRo-OP2

The NimbRo-OP2, as seen in Fig. 2 was developed in preparation for RoboCup 2017. The robot is the effect of our experience gathered through previous work done on similar, albeit smaller platforms i.e. the igus® Humanoid Open Platform [2]. The mechanical design focuses highly on simplicity and cost reduction, without compromising on features and robustness. The robot is 134.5 cm tall and weighs 18.0 kg. Powered by a 4-cell LiPo battery and actuated by Robotis Dynamixel MX-106 actuators for all joints, the robot has enough torque and power to continuously perform various dynamic motions and stable walking for over 15 minutes. To provide the necessary torque for this, we have decided on a parallel kinematics leg structure in combination with external gearing.

Fig. 2. NimbRo-OP2 robot.

In total thirteen MX-106R actuators are used in each leg (8 in the knee, 2 in the ankle and 3 in the hip), three in each arm (2 in the shoulder and

1 in the elbow) and two in the neck to control the pan and tilt of the head. All actuators communicate through a Robotis CM740 board, which is running our custom firmware. Electrically, the actuators are connected on a RS485 Dynamixel bus. The parallel kinematics and externally geared roll joints utilise a master-slave connection with 1 master and 3 slaves in the knee, and 1 master with 1 slave in the hip and ankle. The inertial state of the robot is measured through a 6-axis IMU present on the CM740. To perceive the environment, the robot utilises a Logitech C905 USB camera fitted with a wide-angle lens. The mechanical parts of the robot were 3D printed out of PA-12 Nylon using an industrial-grade SLS 3D printer, excluding the external gearing which was CNC-milled out of brass. All of the electronics and sensors are housed inside the torso, apart from the camera which is located in the head.

Information processing on the robot is done with an Intel NUC computer with dual-core Intel Core i7-7567U processor operating at a possible maximum frequency of 4.0GHz. The onboard PC is not limited to the currently used PC, but in the design we have taken into account future possible upgrades by incorporating VESA75 and VESA100 standard mounts. There is also enough space inside the robot torso to fit a Mini-ITX board with a GPU. More detailed information on the robot's design can be found in [7].

2.2 Copedo

Copedo has been a core player of our TeenSize team at RoboCup. For RoboCup 2017 in Nagoya, he has been extended to be 131 cm tall and 10.1 kg of weight to fit into the AdultSize category. The update was based on another one of our robots—Dynaped, as it was upgraded in 2016. Copedo with his upgrade has received a set of new electronics and a new head, which made him compliant with our ROS-based open-source software. Copedo is constructed from milled carbon fiber parts that are assembled to rectangular shaped legs and soft foam-filled arms. The torso is constructed entirely from aluminum and consists of a cylindrical tube and a rectangular cage that holds the computer, control circuits and battery.

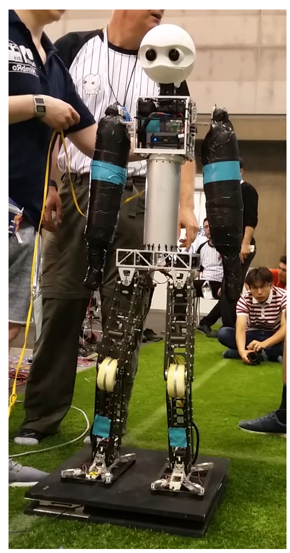

Fig. 3. Copedo robot.

In total, Copedo has 14 actuated DoF. The hip roll, thigh and shank pitch are driven by master-slave pairs of Dynamixel EX-106+ actuators. All other DoFs are driven by single actuators including EX-106+ actuators for ankle roll, EX-106+ actuators for hip yaw and RX-64 actuators for shoulder pitch, as well as the neck yaw and pitch. The robot has been fitted with cleats in the corners of its feet, to assist walking in artificial grass. The cleats have been mounted on beams with strain gauges to enable force sensing which is helpful for dynamic walking.

As for processing power - Copedo is equipped with a Intel NUC computer with dual-core Intel Core i7-7567U processor which has four logical cores and a base frequency of 3.5 GHz with Turbo Boost up to 4.0 GHz. The PC is fitted with 4 GB of RAM and a 128 GB solid state disk. Available communication interfaces include USB, HDMI, Mini DisplayPort, Gigabit Ethernet, IEEE 802.11b/g/n Wi-Fi, and Bluetooth 4.0. The PC is connected to the Robotis board, which communicates with all actuators on a RS485 star topology bus. The CM740 incorporates a 3-axis accelerometer and gyroscope, a 3-axis magnetometer can be added for a total of 9 axes of inertial sensory data.

3 Perception

3.1 Computer Vision

Our humanoid robots perceive the environment through digital Logitech C905 camera. Each camera is equipped with a wide-angle lens and an infrared cut-off filter. In this configuration, its field-of-view is nearly 150°. Our vision system, amongst many other features, can reliably perceive game-relevant objects using shape, texture, and colour information. After detection, we project each object into egocentric world coordinates by using the intrinsic and extrinsic camera parameters. Variations in hardware result in projection errors, which produce errors in distance measurements. We calibrate the projection matrix with the Nelder-Mead [9] method. More details on our vision system can be found in [4].

Landmark Detection: A number of localisation relevant landmarks can be distinguished on the field. These include lines, goal posts, penalty marks and the center circle. For localisation, field lines are the most useful source of information. We detect them by using a Canny edge detector on the V channel in the HSV colour space. We then apply a probabilistic Hough line detector to extract line segments of minimum size. We discard other white but non-line objects. These line segments are then connected to produce less, but longer lines, finally achieving a set of lines that correspond to field lines and the center circle. An example output of our vision system detecting selected objects can be seen in Fig. 4.

Localisation and Breaking the Symmetry: To solve the global localisation problem, our method relies on having a source of global yaw rotation of the robot. We used to utilise the compass sensor in previous years, but due to the removal of magnetometer, introduced in the rules of the Humanoid League in 2017, we now use the integrated gyro measurements as the main source for yaw orientation. This integrated gyro measurement is a reliable source of orientation tracking, but it needs a global reference. In order to set the initial heading, we could either use manual initialisation or automatic initial orientation estimation. Although manual gyro initialisation can be done once before the start of each match, it can fail during the game. Note that, sometimes restarting the operating system of

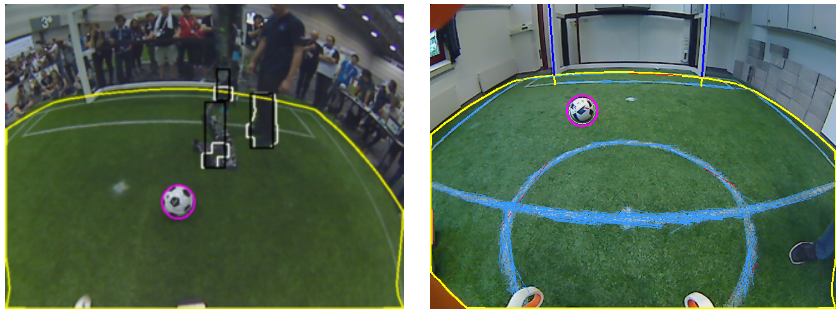

Fig. 4. Left: Visualisation of our obstacle detection. Right: A captured image with detected ball (pink circle), field lines (light blue), field boundary (yellow lines), and goal post (dark blue lines).

a robot is unavoidable, which will force a reinitialisation of the heading. Hence, we reformulated the heading initialisation problem as a classification task.

There are a few predefined positions and orientations, specified in the rules, that the robot can start in or enter the game from. As shown in Fig. 5, the robot can start in four different positions. In two of the spots, the robot should face the opponent goal area—near the center circle and goal area. The other two sets of locations are at the touchline in the robot's own half—facing the field. We employ a multi-hypothesis version of our previous year localisation module, which is initialised with four instances of initial locations. During a brief period before the robot enter the field, it tries to find the most probable hypothesis among all running instances. This procedure stops when either the process times out or the robot finds the best hypothesis. Finally, the robot keeps the best instance and discards the rest. To make sure that the decision is correct, we also double check the result by checking perceived landmarks like goalposts and the center circle.

To add another source of information for supporting robot localisation especially for when the robot is kidnapped, we investigated different approaches to visual odometry. By utilizing the input of our camera system, we tried to assess the 6-DOF motion of the camera. So far, we experimented with SVO 2.0 [8] and DSO [3]. You can see the output of DSO on soccer field in Fig. 5. Both systems offer built-in support for integration into ROS and allow real-time processing of image data. While DSO uses a direct approach based on pixel densities, SVO 2.0 additionally considers features for analyzing visual input. When testing these approaches with smooth camera movements over short periods of time, both calculated stable and accurate motion traces. However, longer periods of time or abrupt and faster movements often lead to wrong results or complete failure of the systems. We will further investigate how to achieve more stable results by integrating IMU information into the visual odometry system.

Obstacle Detection: Obstacle detection/avoidance is a crucial ability in the game, especially when the robot is handling the ball near the opponent. In our

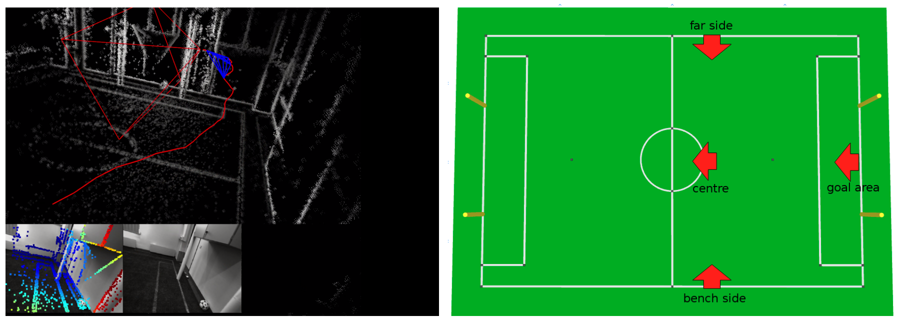

Fig. 5. Left: Output sample of DSO on soccer field. Right: Set of predefined positions the robot can start in.

software, obstacle detection is done mainly based on colour distribution model. By having the minimum and maximum height of the robot in AdultSize class, we roughly know what size to expect in each distance from the observer robot. We search for the respected box size in each distance level and discard obstacle candidates that are not in the expected size range. After detecting each obstacle, we compare the colour histogram of each of the bounding boxes to the expected model of the obstacle, which are then labeled as either rival robot, teammate, or the referee. We then filter and cluster the detection history in egocentric world coordinates. Finally, we predict the expected movement of the obstacle in accordance to the estimated changes in the robot's location. Note that, each of the clusters has a certainty level, which is increased when it is detected, and decreased otherwise. The effect of this method can be seen in Fig. 4.

4 Motion Generation

Motions are generated in the ROS software framework using a multitude of so-called motion module plugins. A main robot control node that is responsible for all of the motion generation and sensor and actuator communications, loads these plugins from a launch-time list, and executes them in every cycle independently of one another. Each motion module can specify based on the robot state and external inputs whether it requests to be active, and shares a common interface for the output joint commands that it wishes to send to the robot. Motion modules get information about the state of the robot from a *RobotModel* class object, that contains all sensor measurements, joint states, previous joint commands, state estimation results, and so on. The execution of the activated motion modules occurs in the order that they are listed in the launch file. Motion modules that execute later can override and thereby block or contribute other parts to the joint commands returned by earlier motion modules.

Typically, the robots are launched with the following motion modules in a game situation:

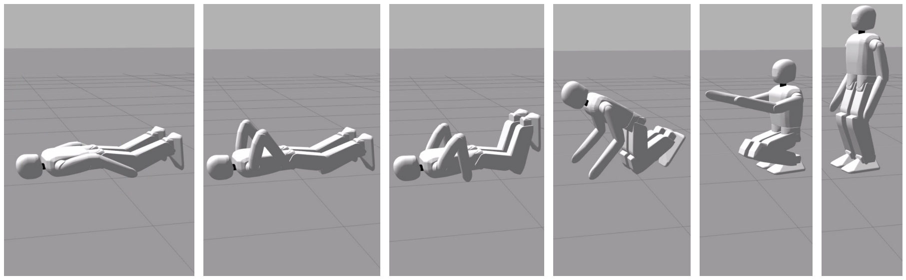

Fig. 6. Get-up motion from a prone position in simulation.

- **Motion player:** The motion player module implements a nonlinear joint space keyframe player, that allows predefined motions to be loaded from file during initialisation, and executed during runtime. Kicking and getting up are examples of motions that are implemented using the motion player.
- **Gait:** The gait motion module is responsible for the walking behaviour of the robot. It can take input from the gait command ROS topic, or a joystick, if one is connected and activated. Seeing as much software infrastructure is required to write a gait motion module that can interoperate with the robot control node, to avoid as much code duplication as possible we use a plugin scheme that outsources the actual implementation of the gait to a further 'gait engine' class. This allows for the implementation and switching between many gaits at launch time.
- **Fall protection:** If the robot detects that it has unbalanced and reached an irrecoverable orientation, the fall protection motion module forces the robot to relax its joints to protect it from the impending fall. After waiting a period of time for the fall to ensue, the estimated trunk orientation of the robot is used to trigger an appropriate get-up motion in the motion player module. An example get-up routine in simulation is shown in Fig. 6. For RoboCup 2018 we will attempt to demonstrate a get-up motion on the real NimbRo-OP2 robot.
- **Head control:** It is a common requirement for the behaviors to need to be able to set the gaze of the robot in a smooth way. The head control motion module implements the required setpoint tracking and filtering to ensure that the head motions are always safe, and velocity and acceleration limited. The head control motion module is given a lower priority than the motion player module however, to ensure that while for example kick motions are being played, the robot does not attempt to keep following head control targets.

5 Conclusions

We are well prepared for RoboCup 2018, and hope that we can demonstrate our latest developments in Montreal. In particular, want to demonstrate team play in 2 vs. 2 games. In addition to software improvements, we are working on upgrading the NimbRo-OP2 platform.

Team Members

Team NimbRo commits to participating in RoboCup 2018 in Montreal, and to provide a referee knowledgeable of the rules of the Humanoid League. Currently, the NimbRo soccer team consists of the following members:
Team leader: Sven Behnke
Team members: Hafez Farazi, Philipp Allgeuer, Grzegorz Ficht, Dmytro Pavlichenko, Diego Rodriguez, André Brandenburger, and Johannes Kürsch

Acknowledgements This research has been supported by the Deutsche Forschungsgemeinschaft (German Research Foundation, DFG) under grant BE 2556/13.

References

1. Philipp Allgeuer, Hafez Farazi, Grzegorz Ficht, Michael Schreiber, and Sven Behnke. The igus Humanoid Open Platform: A child-sized 3D printed open-source robot for research. In *Robot Soccer World Cup XX*. Springer, 2016. LNCS 9776.
2. Philipp Allgeuer, Hafez Farazi, Michael Schreiber, and Sven Behnke. Child-sized 3D printed igus humanoid open platform. In *Proceedings of 15th IEEE-RAS Int. Conference on Humanoid Robots (Humanoids)*, Seoul, Korea, 2015.
3. Jakob Engel, Vladlen Koltun, and Daniel Cremers. Direct sparse odometry. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 2017.
4. Hafez Farazi, Philipp Allgeuer, and Sven Behnke. A monocular vision system for playing soccer in low color information environments. In *10th Workshop on Humanoid Soccer Robots, Int. Conf. on Humanoid Robots*, Seoul, Korea, 2015.
5. Hafez Farazi, Philipp Allgeuer, Grzegorz Ficht, André Brandenburger, Dmytro Pavlichenko, Michael Schreiber, and Sven Behnke. RoboCup 2016 Humanoid TeenSize winner NimbRo: Robust visual perception and soccer behaviors. In *RoboCup 2016: Robot Soccer World Cup XX*. Springer, 2017.
6. G. Ficht, D. Pavlichenko, P. Allgeuer, H. Farazi, D. Rodriguez, A. Brandenburger, J. Kürsch, and S. Behnke. Grown-up NimbRo robots winning RoboCup 2017 Humanoid AdultSize soccer competitions. In *RoboCup 2017: Robot Soccer World Cup XXI*. Springer, 2018.
7. Grzegorz Ficht, Philipp Allgeuer, Hafez Farazi, and Sven Behnke. NimbRo-OP2: Grown-up 3D printed open humanoid platform for research. In *Proc. of 17th IEEE-RAS Int. Conference on Humanoid Robots (Humanoids)*, Birmingham, UK, 2017.
8. Christian Forster, Zichao Zhang, Michael Gassner, Manuel Werlberger, and Davide Scaramuzza. SVO: Semidirect visual odometry for monocular and multicamera systems. *IEEE Transactions on Robotics*, 33(2):249–265, 2017.
9. John A Nelder and Roger Mead. A simplex method for function minimization. *The computer journal*, 7(4):308–313, 1965.
10. D. Rodriguez, H. Farazi, P. Allgeuer, D. Pavlichenko, G. Ficht, A. Brandenburger, J. Kürsch, and S. Behnke. Advanced soccer skills and team play of RoboCup 2017 TeenSize winner NimbRo. In *RoboCup 2017: Robot Soccer World Cup XXI*. Springer, 2018.