Hamburg Bit-Bots and WF Wolves Team Description for RoboCup 2019 – Humanoid TeenSize –

Marc Bestmann[1], Hendrik Brandt[1], Timon Engelke[1], Niklas Fiedler[1], Alexander Gabel[2], Jasper Güldenstein[1], Jonas Hagge[1], Judith Hartfill[1], Tom Lorenz[2], Tanja Heuer[2], Martin Poppinga[1], Ivan David Riaño Salamanca[2], and Daniel Speck[1]

[1] Department of Informatics, Universität Hamburg,
Vogt-Kölln-Straße 30, 22527 Hamburg, Germany
`info@bit-bots.de`
`www.bit-bots.de`
[2] Ostfalia – University of Applied Sciences
On Exer 2d, 38302 Wolfenbüttel, Germany
`robo-wm@ostfalia.de`
`https://www.wf-wolves.de`

Abstract. This team description paper presents the developments made by the joint team Hamburg Bit-Bots & WF-Wolves. We present new software approaches we programmed and evaluated, like the *Dynamic Stack Decider* (DSD), our advances in image processing and our improvements to the walking engine. Additionally the newly developed foot pressure sensors as well as our progress towards a reliable and fast servo control are introduced. We, the joint team of Hamburg Bit-Bots and WF Wolves, hereby apply for participation in the RoboCup 2019 in Sydney, Australia in the **Humanoid TeenSize Team Competition**. The content of this paper is the same as the application for the KidSize [5].

Keywords: RoboCup · Humanoid · Soccer · World Model

1 Introduction

The Hamburg Bit-Bots and WF Wolves are applying to participate as a joint team in RoboCup 2019. In last year's competition we benefited greatly from our cooperation. We were able to exchange robots freely between leagues which allowed us to have more robots in play at multiple games. We have expanded our partnership further in terms of hard- and software. The Hamburg Bit-Bots have moved on from the Minibot platform used in previous years to focus more on improving the shared robot platform. The Hamburg Bit-Bots are a group of Bachelor, Master and Ph.D. students, that are supported by the Department of Informatics at the University of Hamburg. The WF Wolves are supported by the University of Applied Sciences Ostfalia.

1.1 Previous Achievements

The Hamburg Bit-Bots are part of a marketing campaign celebrating the 100th anniversary of the Universität Hamburg. Through the campaign, RoboCup itself is heavily advertised in Hamburg, Germany.

The Hamburg Bit-Bots developed the ImageTagger [11]. Currently multiple teams (see [9][10]) use this tool. Our publicly available instance holds more than 280,000 public images from the SPL and Humanoid League. 270,000 Labels have been manually created for these images.

In recent years, the members of the Team published a number of papers [7], [11], [12], [20], [21].

2 Research

2.1 Vision with Neural Networks

We developed and evaluated two different approaches to vision with neural networks. Both approaches are described respectively in [21] and [12]. Our efforts spent for data collection and image labelling (described in [11]) helped us to further improve both methods.

2.2 World Model

Currently we are working on separately filtering sensor data in a local and a global layer.

The filtering process in the global layer happens in the relative space of each robot. A particle filter is applied onto detected obstacles and players. To increase the amount of information fed into the filter when filtering ball measurements, the whole heat map generated by the FCNN is transformed into relative space and used as measurement in the filter.

In the global filtering layer, the information generated by the whole team is collected via the team communication module and filtered in a separate particle filter. Thus, the self localization and detections of one robot can be corrected by the others.

2.3 Behavior

We were using our own framework, the *stack machine*, for defining the robot behavior since 2014. This year, we rewrote large parts of the stack machine and renamed it to *Dynamic Stack Decider*. It consists of modules defining actions or decisions which are written in Python. Each decision module has a finite set of return values. Based on these, another decision or an action is pushed on the stack as defined in a simple Domain Specific Language (DSL). This approach allows fast changes in the game logic, without the need of touching the codebase. Modules on the stack define the active modules and the current path in the directed acyclic graph (DAG) defined in the DSL. The different modules have access to a shared blackboard to communicate with each other.

2.4 Motion

For the stabilization of our full body motions we are currently researching in two directions. Firstly, we are trying to compute, based on sensory input from foot pressure and IMU, secondary balance goals for inverse kinematics (IK) [19]. It returns the necessary joint values to reach the requested cartesian positions while staying stable.

Secondly, we are trying to transfer the promising results in recent deep reinforcement learning [17] onto our robot. Learned motion may have a higher performance as well as requiring less tuning.

2.5 Walking

Last year, we managed to achieve a good walking on the artificial grass (see section 4.6). Still, it needs to be improved for a higher speed and stability. Since the performance of the current walk engine depends highly on the chosen parameters, we evaluated different parameters in simulation. This allowed us not only to find the best combination automatically, rather than trying out per hand, it also gave us insight about the dependencies between different parameters, thus allowing us to fine tune them better. Similar to the motions (see section 2.4) we are also trying to apply deep reinforcement learning to this problem.

2.6 Hardware

In respect to research on our hardware, we are currently mostly working on increasing the frequency of the control cycle to allow faster reactions to disturbances. While our current servo controller board (see section 3.1) works fine, the cycle rate is limited due to the low performance of the micro controller and the limited baudrate. We are currently replacing this with a never version using an FTDI chip to increase the baudrate and split up communication into three separate communication busses, one for each leg and one for the arms. Our test results and calculations suggest that an update rate of 1 kHz can be achieved. We will release this project as open-source hardware. Furthermore, the new version will include two IMUs to improve the filtering for the estimated orientation of the robot.

Team Rhoban has developed a low cost pressure sensor to measure the ground reaction forces [14]. We improved on their open source design in terms of speed and resolution, see Section 3.2.

3 Hardware

Both teams (Hamburg Bit-Bots and WF Wolves) are using the same platform, called Wolfgang (see 1b), which is an advancement of the platform used last year. The main changes are in the electronics and sensors. The platform has the typical 20 DOF structure using Dynamixel MX-64 and MX-106. Three computers are installed on it: an Intel NUC, an Nvidia Jetson TX2 and an Odroid XU-4. They provide us with enough processing power to run FCNNs for the vision and an evolutionary IK to create stable motions. The computers are connected via GigaBit ethernet and the communication is handled via ROS message passing, allowing a complete abstraction from the distributed system when programming software.

3.1 Servo Control

In the last year we switched to the second version of the Dynamixel protocol. This allows a more stable and faster control of the servos. Furthermore, it is now possible to get a good estimate of the applied torque as well as using torque based control. We are currently using the DXL Board from Rhoban [3] with an upgraded firmware written by us, which supports the new protocol. This allows for an almost error free and faster control of the servos than the CM730 we and many other teams used in the past. We are currently reaching more than 200 Hz when reading and writing all servos. The software for controlling the servos is based on an improved version of the DynamixelSDK and follows the ros_control [8] standards. This enables us to switch between position, velocity and force control. It also provides the same interface as our simulation environment in Gazebo [15], simplifying the transfer from simulation to reality.

3.2 Pressure Sensors

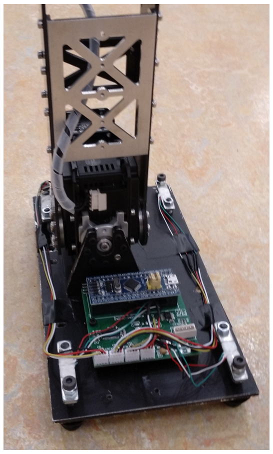

Fig. 1a: The hardware implementation of the ForceFoot. In each of the four corners of the foot, a load cell sensor is mounted. The sensor data is collected and fed into the Dynamxel bus system by an STM32F103 with an external analog to digital converter.

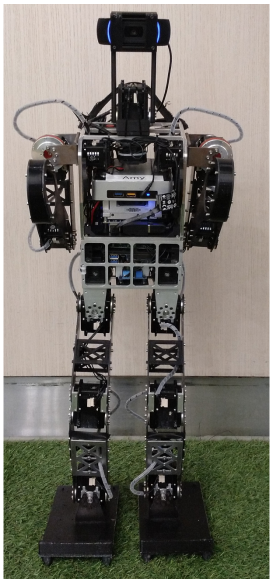

Fig. 1b: The Wolfgang robot platform used by both teams.

Inspired by team Rhoban's ForceFoot [4] we have improved on their design in a number of ways (see 1a). We have increased the possible update rate of the load cells from 80 Hz to a theoretical 9.5 kHz per cell. Since the data transfer rate through the bus is limited we use filtering on the load cells output to achieve a more stable measurement at a speed of 1 kHz. In the next months we will scientifically compare our measurement of the center of pressure to an industrial 6 axis force torque sensor and publish our results. Our improved version of the ForceFoot will be released as open-source hardware in the near future.

4 Software

The WF Wolves started to use ROS [18] in 2016 and the Hamburg Bit-Bots in 2017. We created a common set of messages for the RoboCup Soccer domain [6]. This allows us to easily share and exchange parts of our software. Our code is open and available online[3].

4.1 Vision

Our vision is currently able to reliably detect and identify multiple object types while running at about 10 frames per second on an Nvidia Jetson TX2. These include the moving ball, the field with its markings and goalposts as well as other robots (including team affiliation).

The observation of the environment and accurate detection of various objects is made possible by multiple specialized modules that form our vision. Each module is responsible for identifying or classifying a specific aspect of the picture for further feature extraction, creating a structure that allows our modules to function mostly independently and only use other modules end results.

This property is especially prominent for the obstacle detection module which not only localizes obstructions but also provides potential candidates of other robots and goalposts for the candidate module or goalpost module respectively. The obstacle detection module itself also utilizes other modules' results in order to perform its task, like the horizon detection which is used to model the edges of the field. The resulting horizon line is not completely straight and contains multiple dents which is used for the obstacle detection that outputs a series of candidate. These are used by the candidate module to assign team affiliation of the observed robots by identifying the color of its team markers.

For ball detection we have two alternative approaches. One approach is the FCNN proposed in [21]. The other approach we are evaluating has been proposed in [12]. The performance of this second approach was further improved by replacing AlexNet with MobileNetV2 which is optimized for mobile platforms, while the training process is already sped-up by using a finetuning approach instead of full training. This can be used to train the network on competition in a very short amount of time (in the order of five minutes).

---

[3] `https://github.com/Bit-Bots`, `https://humanoid.wf-wolves.de/explore/`

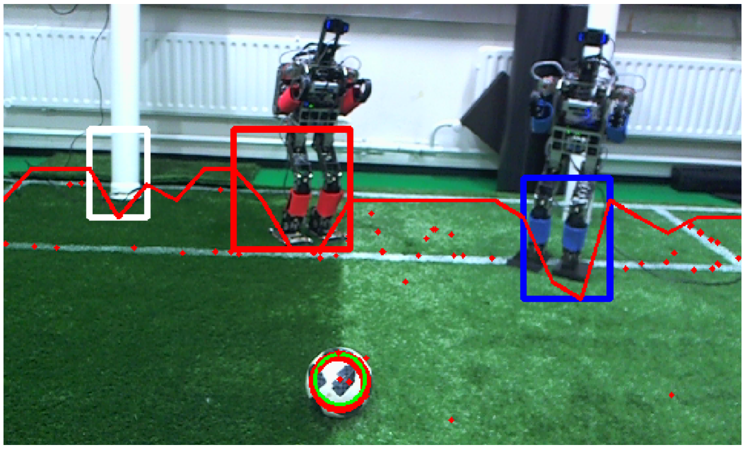

Fig. 2: Debug output of the vision system. The detected horizon is marked by a **red line**, field markings are represented as a set of **red dots**. Around detected obstacles, boxes are drawn (**black**: unknown obstacle, **red**: red robot, **blue**: blue robot, **white**: white obstacles e. g. goalposts). The best rated ball candidate and discarded candidates are indicated by a **green circle** or **red circles** respectively.

4.2 Team Communication

As in previous years, we are using the ”mixed team communication protocol” (mitecom) [1]. This enables us to have robots from our teams playing together as well as playing in a drop-in games. We already proved this ability in the last years and hope that more teams will be using the standard in this years drop-in challenge.

4.3 Localization

The main source of information for self-localizing on the field are the field lines. The vision provides points on the image which belong to lines. Those get transformed to positions relative to the robot and are put into the format of a laser scan to be handled as if they were walls. This allows us to use the standard particle filter based localization node provided by ROS (AMCL [13]). It enables the robot to do global localization for example after pickup, when no initial position is available (kidnapped robot problem). Additionally, we are planning to integrate this pose estimation with the additional information from the walking odometry and information from the IMU. This can be done with the robot_localization package [16] also provided by ROS. This seems to be a very promising approach in terms of accuracy, as the quality of the walking odometry is very high due to the new walking engine.

4.4 Behavior

Our behavior is implemented using the DSD framework (see section 2.3). A robot can play as goalie, offense or defense player. Based on that general role, the following behavior differs. While a goalie keeps standing in the goal when the ball is far away, an offense player walks towards the ball and kicks it in the direction of the opponents goal. The information on which the behavior is based is received from the vision and team communication via ROS messages. The navigation is done with move_base [2] which is widely used in mobile robotics.

4.5 Motions

For the full body motion, e.g. kicking and standing up, we are using teach-in spline animations. While this is similar to the previous years, we improved on it in two parts. Firstly, we replaced our old cubic spline implementation with a quintic one. This improves the smoothness and thereby the stability. Secondly, we improved our teach-in interface and added the possibility to make joints torque less. Currently we are working on closing the loop on this animations, see section 2.4.

4.6 Walking

Last year, we participated with an improved version of Rhobans open-loop QuinticWalk engine [14]. With this we were able to walk across the field even with minor disturbances. While this was already good enough to participate, we are currently working on improving this, see section 2.5.

5 Conclusion

Since we solved the fundamental problems of the artificial grass and the FIFA ball last year, we are now focused on improving on stability and performance. Hamburg Bit-Bots & WF Wolves are looking forward to participating in the RoboCup 2019 for the **Team Competition** in Sydney, Australia.

References

1. mitecom on github, `https://github.com/fumanoids/mitecom`, accessed: 02.12.2018
2. move_base - ROS wiki, `http://wiki.ros.org/move_base`, accessed: 02.12.2018
3. Rhoban communication board, featuring 9dof IMU and 3 dynamixel buses: Rhoban/DXLBoard, `https://github.com/Rhoban/DXLBoard`, accessed: 02.12.2018
4. Robotic foot design involving force sensors, `https://github.com/Rhoban/ForceFoot`, accessed: 02.12.2018

5. Bestmann, M., Brandt, H., Engelke, T., Fiedler, N., Gabel, A., Güldenstein, J., Hagge, J., Hartfill, J., Lorenz, T., Heuer, T., Poppinga, M., Salamanca, I.D.R., Speck, D.: Hamburg bit-bots and wf wolves team description for robocup 2018 humanoid kidsize (2019)
6. Bestmann, M., Hendrich, N., Wasserfall, F.: Ros for humanoid soccer robots. The 12th Workshop on Humanoid Soccer Robots at 17th IEEE-RAS International Conference on Humanoid Robots 2017
7. Bestmann, M., Reichardt, B., Wasserfall, F.: Hambot: an open source robot for robocup soccer. In: Robot Soccer World Cup. pp. 339–346. Springer (2015)
8. Chitta, S., Marder-Eppstein, E., Meeussen, W., Pradeep, V., Rodríguez Tsouroukdissian, A., Bohren, J., Coleman, D., Magyar, B., Raiola, G., Lüdtke, M., Fernández Perdomo, E.: ros_control: A generic and simple control framework for ros. The Journal of Open Source Software (2017). https://doi.org/10.21105/joss.00456, `http://www.theoj.org/joss-papers/joss.00456/10.21105.joss.00456.pdf`
9. van Dijk, S.G., Scheunemann, M.M.: Deep learning for semantic segmentation on minimal hardware. In: Robot World Cup XXII. Springer (2018)
10. Ficht, G., Farazi, H., Brandenburger, A., Rodriguez, D., Pavlichenko, D., Allgeuer, P., Hosseini, M., Behnke, S.: Nimbro-op2x: Adult-sized open-source 3d printed humanoid robot. In: Robot World Cup XXII. Springer (2018)
11. Fiedler, N., Bestmann, M., Hendrich, N.: Imagetagger: An open source online platform for collaborative image labeling. In: Robot World Cup XXII. Springer (2018)
12. Gabel, A., Heuer, T., Schiering, I., Gerndt, R.: Jetson, where is the ball? - using neural networks for ball detection at robocup 2017. In: Robot World Cup XXII. Springer (2018)
13. Gerkey, B.P.: amcl, accessed: 02.12.2018
14. J. Allali, R. Fabre, H.G.L.G.L.H.O.L.S.N.G.P.A.P.Q.R.: Rhoban football club – team description paper humanoid kid-size league, robocup 2018 montreal. In: Robot World Cup XXII. Springer (2018)
15. Koenig, N.P., Howard, A.: Design and use paradigms for gazebo, an open-source multi-robot simulator. In: IROS. vol. 4, pp. 2149–2154. Citeseer (2004)
16. Moore, T.: robot_localization. `http://docs.ros.org/melodic/api/robot_localization/html/index.html`, accessed: 02.12.2018
17. Peng, X.B., Abbeel, P., Levine, S., van de Panne, M.: Deepmimic: Example-guided deep reinforcement learning of physics-based character skills. arXiv preprint arXiv:1804.02717 (2018)
18. Quigley, M., Conley, K., Gerkey, B.P., Faust, J., Foote, T., Leibs, J., Wheeler, R., Ng, A.Y.: Ros: an open-source robot operating system. ICRA Workshop on Open Source Software 2009 (2009)
19. Ruppel, P., Hendrich, N., Starke, S., Zhang, J.: Cost functions to specify full-body motion and multi-goal manipulation tasks. In: 2018 IEEE International Conference on Robotics and Automation (ICRA). pp. 3152–3159. IEEE (2018)
20. Speck, D., Barros, P., Weber, C., Wermter, S.: Ball localization for robocup soccer using convolutional neural networks. In: Robot World Cup XX. Springer (2016)
21. Speck, D., Bestmann, M., Barros, P.: Towards real-time ball localization using cnns. In: Robot World Cup XXII. Springer (2018)