Berlin United - FUmanoids
Team Description Paper 2013

Daniel Seifert, Lutz Freitag, Julius Auer, Jan Draegert, Heiko Donat, Simon Gene Gottlieb, Robert Kriehs, Michael Schmidt, Steffen Puhlmann, Oscar S. Morillo Victoria, Simon Hohberg, Nicolai Steinke, Hamid Moballegh, and Raul Rojas

Institut für Informatik, AG Intelligente Systeme und Robotik,
Freie Universität Berlin, Arnimallee 7, 14195 Berlin, Germany
https://www.fumanoids.de

Abstract. This Team Description Paper describes the humanoid robot team *Berlin United - FUmanoids* and presents the updated generation of robots for participation in RoboCup 2013. A general overview of the team and its history will be given as well as insight to research interests and particular areas of the robots' software and hardware.

1 Introduction

Berlin United - FUmanoids is a humanoid robot team participating in the Humanoid KidSize League at RoboCup. The team was founded in 2006 as the successor of the Mid- and SmallSize team *FU-Fighters* and consists on average of two research staff members and about a dozen bachelor and master students.

During the time of participation at RoboCup, the team had significant successes in competitions, achieving 2nd place in 2009 and 2010, 3rd place in 2007 (its debut competition), 4th place in 2011 and reaching the quarter finals in 2008 and 2012. The team also scored 1st or 2nd place at Iran Open competitions (2008 to 2011), and 2nd (2008, 2011, 2012) and 3rd (2009) places in RoboCup German Open.

Since 2010 the team has cooperated with the Standard Platform and 3D Simulation League team *Nao Team Humboldt* at neighboring Humbold-Universität zu Berlin, resulting in the multi-league joint-research group *Berlin United* in 2011.

This paper presents the team's research interests, contributions to the RoboCup community as well as the hardware and software of the FUmanoid robot model that will be used in RoboCup 2013.

Notice of commitment: The team commits to participate in RoboCup 2013 and to provide a referee knowledgable of the rules of the Humanoid League.

2 Research and Contribution

The main *research interests* are:

- fast and stable *walking*;
- dezentralized *control architecture* for humanoid robots;
- *vision* algorithms not depending on the colored RoboCup environment but handling more natural environments and wider ranges of lighting conditions by using shapes, forms and other heuristics; and
- *communication* within single and mixed robot teams.

The team is committed to further the Humanoid League and the research exchange between teams. For this reason the team is one of few Humanoid League teams that releases their source code ([1], [2]). The release consists of *FUmanoid*, the main program running on the robot, *FUremote*, the control and debugging program based on Eclipse/RCP, as well as several helper scripts. An accompanying team report is currently being worked on. Theses and papers on the robots are available on our website[1].

Based on regular meetings and matches against *NaoTH* in 2010, we cofounded the local *RoBOW* [2] workshops for RoboCup teams (Humanoid KidSize and Standard Platform league) in 2011 and are actively engaged in their organisation. The workshops feature test matches, talks based on current research as well as discussion rounds. Due to the informal setup and without competition matches, information exchange between the teams is significantly higher than during regular RoboCup competitions. In 2012 three such workshops took place in Berlin, the last one in December with about 50 participants from nine German RoboCup teams (3 from the Humanoid League and 6 from the SPL). Additional workshops are planned for 2013[3].

Due to our partnership with Humboldt-Universität zu Berlin we are also interested in mixed teams. To that end, we are working on the design and implementation of a general communication protocol that will allow robots from different research groups to form a cooperating team across platform boundaries. Together with *NaoTH* we were awarded a RoboCup project grant in 2012 to work on an implementation, which we hope to present in the upcoming competitions.

3 Hardware

3.1 Mechanical Structure

For RoboCup 2012 a new robot model was designed and constructed with special attention being paid to simplicity as well as human-like proportions and capabilities. For RoboCup 2013, the basic structure remains the same. However, some updates were implemented in order to reflect the experiences made in RoboCup 2012 (see figure 1).

---

[1] `http://www.fumanoids.de/publications`

[2] `http://www.robow.de`

[3] The next one will take place on February 22-24 at TU Dortmund, organized by team *Nao Devils*.

One such change is a new foot design as the previously used load cell sensors[11] have been abandoned. The hip width has been increased to support better leg movement below the robot's center of gravity. For increased balance (see section 4.2), a further servo motor is being added to allow the torso to move laterally, imitating the human spine movement that keeps the torso upright. Additionally, the torso's depth is increased in order to move the battery inside the torso and into the center of the sagittal plane.

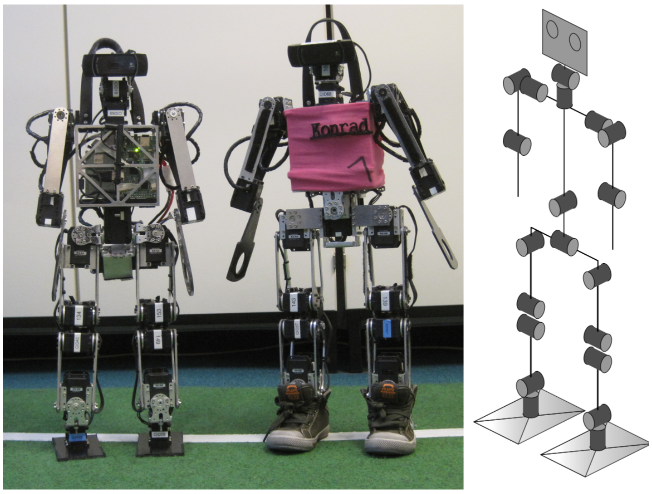

Fig. 1. 2012 model (left) with 2013 (middle) and its kinematics (right)

For actuation, Dynamixel servo motors from Robotis Inc. are used, namely RX-28 and RX-64 servos. They provide 20 degrees of freedom - 5 per leg, 2 for upper-body movement, 3 per arm and 2 in the head (figure 1). The legs feature a parallel kinematic which limits the feet to be always parallel, resulting in the swing leg to be parallel to the ground on impact and thus improving stabilization.

3.2 Sensors

The robot is equipped with the following sensors:

Actuators: The feedback of the actuators includes the current joint angle, the current motor speed, and the load. Because all of these values are derived from the feedback sensor of the actuators[4], the latter two values are less reliable. Joint position measurement is very helpful in stable gait generation for the robots. Other measured values which can be accessed through

---

[4] The position potentiometers found in the stock RX-28 and RX-64 servos have been replaced by hall sensors to prevent jitter caused by worn-out potentiometers.

the Dynamixel serial interface is for example the supplied voltage and the temperature of the servos, which is used for safety purposes.

IMU: The updated sensor board features an integrated 9 axis IMU, featuring gyros, accelerometers and magnetometers. A Kalman filter provides filtered output that is used by the robot for stabilization as well as calculation of the camera perspective in order to obtain localization data.

Camera: The robot is equipped with a commercially available webcam (Logitech HD Pro Webcam C910). Using a resolution of 640x480 (VGA) it delivers up to 30 frames per second.

3.3 Main Computing Unit

In order to satisfy increased performance requirements, last year's main computing unit is being replaced by an ODROID-X2 board. This board features an Exynos4412 Quad-core ARM Cortex-A9 CPU clocked at 1.7 GHz. It provides all necessary extension interfaces, like multiple USB ports (for the camera and the WiFi module), ethernet as well as an UART connection.

The operating system is Linux, based on a custom-compiled kernel and the Linaro distribution.

3.4 Sensorboard

In order to improve communication with actuators and sensors, a new sensorboard was developed. It supports connecting the actuators of each leg and the upper body separately in order to set and get servo positions in parallel. Two ARM Cortex M4 processors clocked at 168 MHz each are handling the Kalman filter for the IMU (see above), as well as servo communication. Data can be requested from the main unit and actions, e.g. movements of the robot, triggered via a dedicated serial connection.

4 Software

4.1 Architecture

In order to streamline the development of new functionality and to make it easier for new project members to start their work, a modular architecture based on the German Team's code and reimplemented and refined by our joint team is used [6]. Figure 2 shows the block diagram of the software which runs on the robot. On the lowest level the hardware interface provides access to the various parts of the robot's hardware. Above that is a set of classes that provide additional services, e.g. configuration management, communication control and handling of debug output. This also includes higher-level abstractions of the robot's hardware.

At the top are two module blocks consisting of various modules that are executed in a pre-defined chained order. The cognition module block is triggered by a new image (i.e. up to 30 times a second), whereas the motion modules are triggered at a 10ms interval (i.e. at 100 Hz).

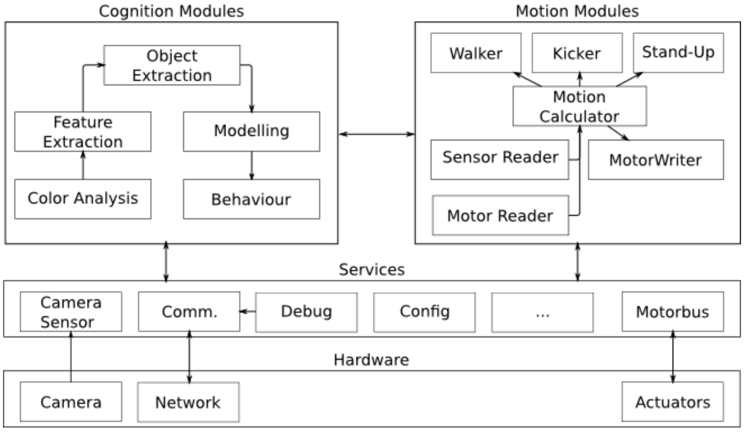

Fig. 2. Structure of the control software

Most of the lower layers are generic and feature interchangeable hardware support. Together with the module framework, these code parts make up the BerlinUnited Core Framework[5] which is also used in other robotics projects at our institute.

Due to the hardware interface encapsulating the actual implementations, the code supports easy adjustments to new hardware. At the moment two simulators (4.5) are supported, as well as the robot model of 2012 and the updated robot model of 2013. Additionally support of the DARwIn OP robot and the Nao robot are work in progress.

4.2 Locomotion and Stability Control

The walking system of the FUmanoids team is based on two different traditional walking paradigms, namely Passive Dynamic Walking and Zero Moment Point optimization. It is capable of dynamically walking in every direction with speeds up to 25 cm/s.

The basic forward walking trajectory generation is done by analysis of Passive Dynamic Walkers, as McGeer describes them [5]. Their mechanics are designed to efficiently use gravity for powering the gait. Since our robots are not designed in that way, we utilize the trajectories found on Passive Dynamic Walkers in our pre-computed trajectories. So our forward trajectories are approximations of measured PDW trajectories (see fig. 3).

PDWs are not capable of walking adjacent or rotating, but in a soccer game these movements are crucial to arrive at every point of the pitch.

Tackling this problem, our approach is highly inspired by the biomechanics and natural movement of the human gait. It is based on the Zero Moment Point

---

[5] This separation was added after RoboCup 2012 and will be reflected in the 2013 code release

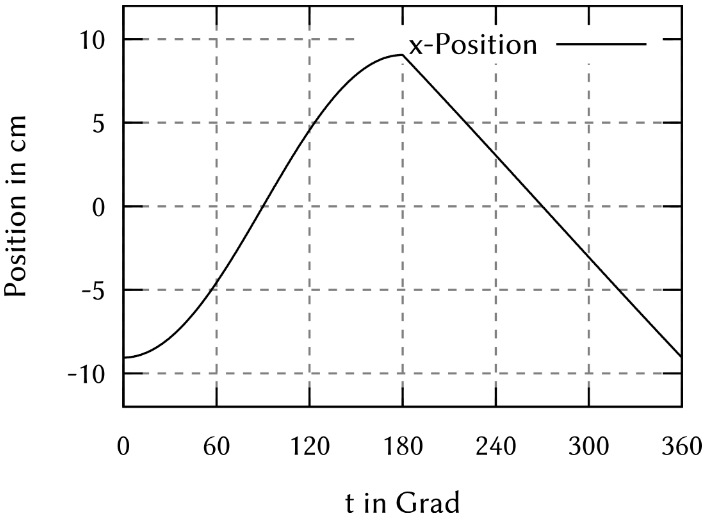

Fig. 3. The trajectory of the leg is based on a pendulum-like model: the passive dynamic walker. In the swingphase it acts like a pendulum (sinusoide). In the stance phase it moves linearly.

optimization and involves an inverted pendulum based model of the robot's upper body dynamics. By shifting the torso laterally, the COM always stays above the current supporting foot. Meanwhile the other foot can perform a step, since it carries no load.

This inverted pendulum model causes a lot of movement of the upper body, leading to an unstable walk. To handle this issue we also use a technique that can be found in natural biomechanics. Instead of shifting the whole upper body, the robot instead just moves its hip, while keeping the torso as stationary as possible. This still allows the robot to shift the COM, but ensures the stability needed for successful walking. Necessary changes in the robot's design have been shown in 3.1.

The resulting trajectories for the feet are then replayed in a cyclic way. However, they do not suffice to generate a stable walk, which is also capable of handling external forces, e.g. pushes. Therefore stabilization is needed. The most important method for stabilizing the gait is the step synchronization based on energy analysis of the walker [8]. Data from the IMU is used to invoke counteractions against destabilizing forces.

4.3 Vision

The vision module consists of three independent layers:

Color Analysis For the RoboCup 2011 the use of a manually generated color lookup-table was minimized. In the first step the colors of a set of sample pixels are analyzed to determine the range of the field color (green). The ball and goal colors (orange, yellow, blue) are initially determined through static thresholds which are dynamically improved when instances of the ball and goals have been seen.

Feature Extraction The features of the image are extracted in this layer. There are currently two different features defined, *field contour* and *edge traces*.

The field contour divides the image in two parts where the part above the field contour contains only information about objects outside the soccer pitch and can therefore be discarded. The other part contains the visible parts of the pitch and is used for further image processing steps. The calculation of the field contour is based on vertical scan lines of green classified pixels.

An edge trace is an ordered list of edge pixels which correspond to the same object boundary and have similar orientation, position and color. The *Gradient Vector Griding* algorithm [9] is used to calculate the edge traces inside the soccer pitch.

Object Extraction In the object extraction step all the relevant objects on the pitch are constructed out of the features. The currently provided objects are *ball*, *goals*, *side poles*, *field lines*, *field line features* and *obstacles*. The ball, field lines, field line features and obstacles are constructed using the edge traces, the positions of goals and side poles in the image are calculated by using the field contour. Currently, we test hue based self-calibrating vision modules for *field*, *ball* and *goal* detection. These vision modules will be able to handle changing *ball*, *goal* and *field* colors and lighting conditions automatically.

4.4 Modelling

The previously used self-localisation algorithm was based on a Monte Carlo particle filter. This has been replaced by a multi-hypotheses Kalman filter ([4], [10]) which provides similar good and robusts results with significantly lower computational overhead[10].

For obstacle and ball tracking, Kalman filters are employed. We also added local models of the goal and certain field features as fallback or for certain situations (e.g. positioning for kickoff).

4.5 Simulation

*Sim**[3][7], a simulator developed by the team, allows to simulate and test the FUmanoid software or teams of agents on a local computer. The FUmanoid program connects to the simulator through unix or network sockets and registers the requested sensors. Sensors can be real sensors like a camera or an IMU, but also artificial sensors like the ball percept or the goal model. By being able to inject such higher level information and bypassing certain low-level modules of the system, it is easy to debug the system and test modules in separation.

For 2013, support for SimSpark was added to the FUmanoid program, allowing to make use of the different features offered by the two simulators. To that purpose, we are working on automated nightly test games in the simulators, which should provide additional information on the impact of code changes to the quality of the game.

5 Conclusion

With the outlined improvements to the hardware and particularly the software of the robots we are looking forward to participate in the RoboCup 2013 competitions.

References

1. FUmanoids. FUmanoids Code Release 2011, 2011. Available online at http://www.fumanoids.de/publications/coderelease.
2. FUmanoids. FUmanoids Code Release 2012, 2012. Available online at http://www.fumanoids.de/publications/coderelease.
3. S. Heinrich. Development of a Multi-Level Sensor Emulator for Humanoid Robots. Diploma thesis, Freie Universität Berlin, Institut für Mathematik und Informatik, Deutschland, January 2012.
4. G. Jochmann, S. Kerner, S. Tasse, and O. Urbann. Efficient Multi-Hypotheses Unscented Kalman Filtering for Robust Localization. In T. Röfer, N. M. Mayer, J. Savage, and U. Saranli, editors, *RoboCup 2011: Robot Soccer World Cup XV*, volume 7416 of *Lecture Notes in Computer Science*, pages 222–233. Springer Berlin / Heidelberg, 2012.
5. T. McGeer. Passive Walking with Knees. In *Proceedings of IEEE Robotics and Automation Conference*, pages 1640–1645, 1990.
6. H. Mellmann, Y. Xu, T. Krause, and F. Holzhauer. NaoTH Software Architecture for an Autonomous Agent. In *Proceedings of the International Workshop on Standards and Common Platforms for Robotics (SCPR 2010)*, Darmstadt, November 2010.
7. S. Mielke. Kamerabildrekonstruktion der Simulationsumgebung für humanoide Fußballroboter. Diploma thesis, Freie Universität Berlin, Institut für Mathematik und Informatik, Deutschland, 2012.
8. H. Moballegh, M. Mohajer, and R. Rojas. Increasing Foot Clearance in Biped Walking: Independence of Body Vibration Amplitude from Foot Clearance. In Luca Iocchi, Hitoshi Matsubara, Alfredo Weitzenfeld, and Changjiu Zhou, editors, *RoboCup 2008: Robot Soccer World Cup XII*, volume 5399 of *Lecture Notes in Computer Science*, pages 157–165. Springer Berlin / Heidelberg, 2009.
9. H. Moballegh, N. von Schmude, and R. Rojas. Gradient Vector Griding: An Approach to Shape-Based Object Detection in RoboCup Scenarios. *RoboCup 2011: Robot Soccer World Cup XV*, pages 162–173, 2012.
10. S. Otte. Where am I? What's going on? – World Modelling using Multi-Hypothesis Kalman Filters for Humanoid Soccer Robots. Master thesis, Freie Universität Berlin, Institut für Mathematik und Informatik, Deutschland, 2012.
11. D. Seifert, S. Otte, J. Kulick, N. Schmude, L. Dohrmann, S. Heinrich, H. Moballegh, S. Mielke, L. Freitag, S. Hohberg, et al. Berlin United-FUmanoids Team Description Paper 2012. 2012.