Berlin United - FUmanoids
Team Description Paper 2012

Daniel Seifert, Stefan Otte, Johannes Kulick, Naja v. Schmude, Lisa Dohrmann, Steffen Heinrich, Lisa Dohrmann, Hamid Moballegh, Sebastian Mielke, Lutz Freitag, Simon Hohberg, Julius Auer, Max Losch, 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 new generation of robots for participation in RoboCup 2012. 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

This paper presents the team *Berlin United - FUmanoids*, a humanoid robot team participating in the Humanoid KidSize League at RoboCup. Consisting of students and research staff of Freie Universität Berlin, the team had significant successes in previous RoboCup competitions, achieving 3rd place at its debut in RoboCup 2007 and 2nd place in 2009 and 2010.

The team was founded in 2006 as the successor of the Mid- and SmallSize team *FU-Fighters*. In the last few years, contact and cooperation with the SPL-RoboCup team *NaoTH* at neighboring Humbold-Universität zu Berlin steadily increased and resulted in the joint research group *Berlin United*.

In the following sections we will describe the hardware and software of the robot model that we plan to use in RoboCup 2012. This year's robot model is the result of continued development of our models from previous years with significant improvements in both hardware and software.

Notice of commitment: The team commits to participate in RoboCup 2012 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 *locomotion*;
- 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
– versatile simulation, not only for humanoid robots in the RoboCup environment but also for different scenarios like autonomous cars. A versatile plug-in mechanism allows to simulate every sensor of the agent, which itself is defined by a *robot description file*.

The team is committed to further the humanoid league and the research exchange between teams. For this reason the source code used in RoboCup 2011 was published in December 2011. 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. It is available for download at https://www.fumanoids.de/publications/coderelease.

We are also actively engaged in the organisation of a local RoboCup workshop (*"RoBOW"*) as part of our joint research group *Berlin United* with the team from Humboldt-Universität zu Berlin. In 2011 two such workshops took place in Berlin with six teams participating (two from the KidSize league and four from the SPL). More workshops are planned for 2012.

Due to our unique partnership with Humboldt-Universität zu Berlin we are also interested in mixed teams. To that end, we are planning to design and develop a general communication protocol that will allow robots from different research groups to form a cooperating team even across platform boundaries.

3 Hardware

3.1 Mechanical Structure

For 2012 a new robot model was designed and constructed (figure 1) with special attention being paid to simplicity as well as human-like proportions and capabilities. To facilitate exchange of the players, all robots are mechanically identical.

For actuation, Dynamixel servo motors from Robotis Inc. are used, namely RX-28 and RX-64 servos. They provide 19 degrees of freedom - 5 per leg, 1 for upper-body movement, 3 per arm and 2 in the head (figure 1).

Continuing last year's switch to a parallel mechanism, the legs now fully utilize such a mechanism which guarantees that the foot will always be parallel to the ground. This improves the robot's stabilization during walking significantly.

3.2 Sensors

The Dynamixel servos are connected by a single bus structure. Based on this bus, additional sensors can easily be integrated with the hardware. The robot is equipped with the following sensors:

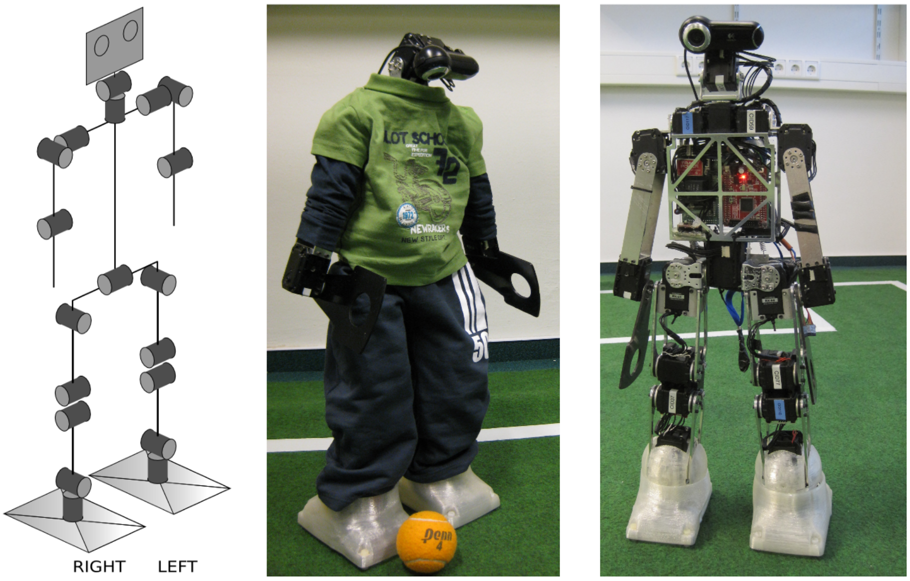

Fig. 1. Prototype of *FUmanoids 2012* model (middle/right) and its kinematics (left)

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 only feedback sensor of the actuators (the position potentiometer), the latter two values are less reliable. There are also other measured values which can be accessed through the Dynamixel serial interface, such as supplied voltage and temperature, which can be used for safety purposes. Joint position measurement is very helpful in stable gait generation for the robots.

Load sensors: Both feet consists of four load cells each and a custom-developed board that measures both ground contact of the foot as well as the load currently applied to it. This information is used to synchronize the walking with the mechanical properties of the robot.

IMU: A commercially available IMU (UM6) is used within the robot to support robot stabilization as well as calculation of the camera perspective in order to obtain localization data. The IMU unit features rate gyros, accelerometers and magnetic sensors.

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.

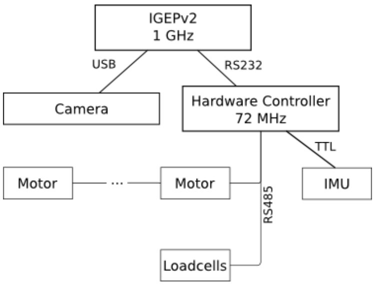

Fig. 2. Structure of sensors and computation units

3.3 Main Computing Unit

The main computational unit is an IGEPv2[1] board featuring a DM3730 processor running at 1 GHz. This CPU is from the ARM Cortex A8 family and includes a DSP that could be used for additional computing. The IGEPv2 motherboard provides all necessary extension interfaces required for a humanoid robot and features low weight and power consumption, making it ideal for our robots. The board also supports Wireless LAN which is used for team communication.

3.4 Hardware Communication Unit

For communication with the hardware units, namely motors and sensors, a microcontroller is used that serves as both a preprocessing and stand-alone motor control unit. This board features a 72 MHz ARM Cortex M3 processor. Data can be requested from the main unit and actions, e.g. movements of the robot, triggered via a dedicated serial connection.

4 Software Design

4.1 Architecture

For 2012 the FUmanoid software architecture was significantly improved. 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 refined by our joint team is used [5]. Figure 3 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 there are two module blocks consisting of various modules that are executed in a pre-defined chained order. The cognition module block is triggered

[1] http://www.igep.es

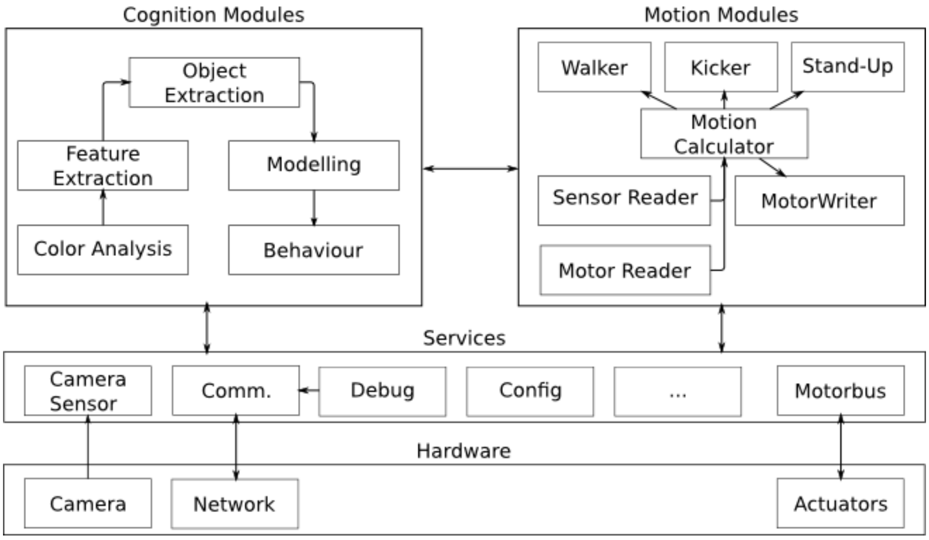

Fig. 3. Structure of the control software

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).

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 [4]. These machines are capable of walking down a slope without external power source. 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. 4).

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. For that reason we performed a Zero Moment Point (ZMP) optimization. This method is based on the dynamically extended center of mass control [8]. We computed the ZMP with the simplification of the 3D-Linear Pendulum method and the inverse ZMP with a preview controller, both described in [3]. For controlling the center of mass, the output of the preview controller, we used inverse kinetics [2].

The resulting trajectories are then replayed in a cyclic way, but do not suffice to generate a stable walk, which is also capable of handling external forces, e.g.

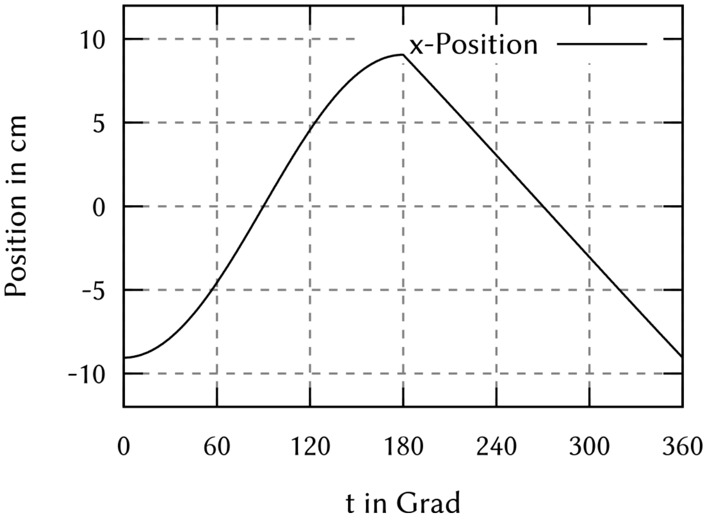

Fig. 4. The trajectory of the leg is based on a pendulum-like model: the passive dynamic walker. In the swingphase it acts like a pendulum (sinusoide). In the stance phase it moves linearly.

pushes. The most important method for stabilizing the gait is the step synchronization based on energy analysis of the walker [6]. With the pressure sensors in the feet of the robots, we are able to sense ground contact and respond to impacts accordingly. This feedback loop can stabilize the robot in most situations, where no external forces, like pushes, are applied.

To get higher stability in such cases an ankle-hip-strategy mixture is applied. When the external forces, determined by pressure distribution in the feet, are under a given threshold a PD-controller uses the ankle joint to keep the center of mass over the support polygon. If the forces exceed such a limit the hip joint is also used to keep upright. Such a behavior can be seen on humans as well [1].

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 [7] 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.

4.4 Modelling

In 2011 a Monte Carlo particle filter was used for the selflocalization of the robot. For this year we are using a Multi Hypothesis Unscented Kalman Filter which operates in a higher dimensional state than the particle filter. It tracks the robots' position and speed.

The module "worldmodel" as of 2011 was replaced by several small modules which model certain aspects. We apply a variety of filters (e.g. Kalman, particle and binary filters) to track teammates, opponents and the ball. Furthermore we provide local models of the ball, goal and the center circle as fallback or for certain situations (e.g. positioning for kickoff).

4.5 Simulation

*Sim**, the simulator of the FUmanoids, 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.

Sim* can not only handle the FUmanoid software but any kind of agent. Therefore it can be used for different scenarios (so called *SimCases*) than RoboCup. An agent can be defined by a *robot description file* which contains physical properties (proportion of links and actuators), and potential sensors of the agent (camera, IMU, range sensors, or artificial sensors like the ball percept).

5 Conclusion

With the outlined improvements to the hardware and particular software of the robots we are looking forward to participate in the RoboCup 2012 competitions.

References

1. Christopher G. Atkeson and Benjamin Stephens. Multiple balance strategies from one optimization criterion. In *Proceedings of the International Conference on Humanoid Robots*, pages 57–64, 2007.
2. Ronan Boulic, Ramon Mas, and Daniel Thalmann. A Robust Approach for the Center of Mass Position Control with Inverse Kinetics. *Journal of Computers and Graphics*, 20(5):443–452, 1996.
3. Shuuji Kajita, Fumio Kanehiro, Kenji Kaneko, Kiyoshi Fujiwara, and Kensuke Harada Kazuhito Yokoi. Biped walking pattern generation by using preview control of zero-moment point. In *Proceedings of the International Conference on Robotics and Automation*, pages 1620–1626, 2003.
4. Tad McGeer. Passive Walking with Knees. In *Proceedings of IEEE Robotics and Automation Conference*, pages 1640–1645, 1990.
5. Heinrich Mellmann, Yuan Xu, Thomas Krause, and Florian 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.
6. Hamid Moballegh, Mojgan Mohajer, and Raul 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.
7. Hamid Moballegh, Naja von Schmude, and Raúl Rojas. Gradient vector griding: An approach to shape-based object detection in robocup scenarios. In Thomas Röfer, Norbert Michael Mayer, Jesus Savage, and Uluç Saranli, editors, *RoboCup 2011: Robot Soccer World Cup XV*, Lecture Notes in Computer Science. Springer Berlin / Heidelberg, will be published in 2012.
8. Miomir Vukobratovic and Branislav Borovac. Zero-Moment Point - Thirty Five Years of its Life. *International Journal of Humanoid Robotics*, pages 157–173, 2004.