

Bold Hearts Team Description

RoboCup 2017 Kid Size

Sander G. van Dijk, Alessandra Rossi, Daniel Barry, Daniel Polani, Marcus Scheunemann, and Masood Ahmed

Adaptive Systems Research Group
School of Computer Science
University of Hertfordshire, UK

Abstract. In this paper we describe the structure and the software of the RoboCup Humanoid Kid Size division team *Bold Hearts*. We discuss the construction and the aspects of our software architecture, enlightening our solutions to important problems such as the action selection, vision, stability, and sensorimotor modeling. Finally, we provide an overview of the relevant experience and achievements of our team.

1 Bold Hearts

The team Bold Hearts has been founded as part of the Adaptive Systems Research Group at the University of Hertfordshire, Hatfield (UK). The team has participated in RoboCup since 2003 in the simulation leagues. Since the 2013 RoboCup competition, the team moved to the Kidsize humanoid hardware league.

2 Robotic Hardware and Design

We have used the standard DARwIn-OP robotic platform to make up our robotic team since the 2013 RoboCup World Championship in Eindhoven, the Netherlands. In the years after the transition, we adapted the DARwIn-OP framework and developed a full framework, including debugging tools, web-based inspection tools and other necessary modules to enable the robots to address the changing and increasingly challenging environments and rules of the game. While significant achievements can already be reached using the traditional DARwIn-OP hardware by judicious development of software controls (demonstrated for instance by our team winning the 2016 Iran Open), the need to adapt the hardware itself has become more and more prominent. The Bold Hearts have therefore moved towards developing custom robot hardware, utilizing 3D printing facilities.

A particular advantage of 3D printing technology is that it will allow us to develop modified parts for the robot, and equip them step-by-step with newer and more capable electronics and motors. Observing the developments of other teams in the league, we conclude that a significant risk can be involved with creating a new platform from scratch, and that it can take a long time to get such

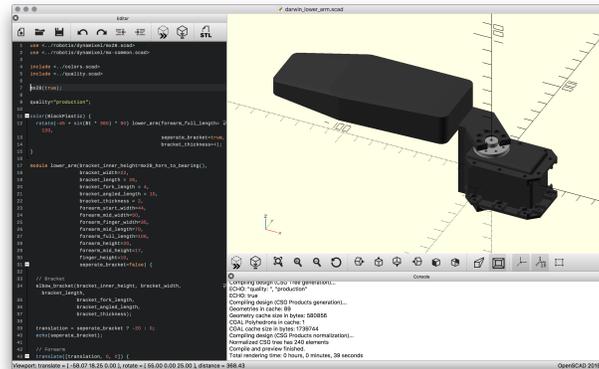


Fig. 1. Screenshot of 3D arm design, including implemented Dynamixel MX28 model.

a platform to perform at a competitive level. Our team stems from Computer Science and has a strong software design and development foundation, which was a major contribution to our past successes while still gaining experience working with hardware. We believe that modern well established practices from software development can be applied to robot design as well, most notably agile processes including rapid prototyping and rapid iterations. In practice, this means that it is a part of our team’s research to develop systematic techniques to link evolutionary changes of various robotic parts to respond to requirements in other modifications in robotic design, and also to support this by suitable incrementally adaptive learning algorithms. For example, rather than determining the full body layout up front, one can iteratively update parts based on requirements and added value in response to previous changes, such as lengthened arms to help standing up following changes to feet and/or legs to be better able to walk on high grass.

We have chosen to design our robot parts using OpenSCAD, instead of using a proprietary 3D CAD system. One of the main reasons is that the design is captured as plain text, in a programmatic and parameterised format. This allows the development of the designs to be tracked in version control by Git, created in any editor of choice, and be easily shared and reused. It makes it much easier to publish designs in a usable way, compared to a vendor-specific binary format, or post-processed 3D formats that remove ability to adapt a design easily such as STL. With this in mind, we intend to share our designs in the future, including models we have created of popular components, such as a range of Dynamixel motors and control boards. Figure 1 shows the OpenSCAD design process in action.

2.1 Long Term Experimentation

3D printing technology allows us to exploit different materials and structures to study different hardware and different robot morphologies and designs. We are equipped to include the study of non-stiff, elastic body parts with customized morphology in this endeavour, exploring the potential of ideas for “intelligent material design” stemming from ideas from morphological computation.

Our interests also lie in experimentation with alternative sensors, for instance by using cameras based on the principle of event-based imaging and to study the viability of their motion-detection capabilities for the specific tasks of the RoboCup scenario.

3 Vision

As described in our team’s description paper of last year, the vision system of our robots is constructed around a highly optimised run-length encoding based connected component detection system, that is able to extract a range of component features in a single pass over the image, such as size, shape, orientation and colour/shade distribution, to be used in classification of the detected objects. Although this system was still able to perform competitively in the previous season, a more elaborate method for object detection is required in the ambiguous and less structured environment of the current RoboCup Humanoid League scenario.

For this, our team is developing Convolutional Neural Networks (CNN) as a stage in the object detection pipeline. Such methods are very popular, however often also very computation intensive and used in the context of very large large datasets. Nevertheless, we start seeing the use of CNNs in the much more computationally restricted domain of RoboCup, for example in the best-paper winning work of team Hamburg Bit-Bots[7]. An efficient solution is however hard to find, the aforementioned authors for instance report that running a trained network takes approximately 0.3 seconds per image on their robots, which is not enough to keep track of a dynamic football game. We are researching two directions to overcome this problem.

Firstly, we move away from ‘traditional’ deep networks that consist of convolutional ‘feature detection’ layers, followed by one or more fully connected layers to arrive at an output over the whole input image. It is often these fully connected layers that are responsible for the bulk of the network’s weights, and as such for much of the training and processing time. More recently, so called *Fully Convolutional Networks (FCNs)* have been developed[4], specifically for *semantic segmentation*, which is the problem of assigning semantic labels to each image pixel. This fits exactly with the ubiquitous colour based labelling used by many RoboCup teams, so the first benefit of these methods is that their output can be directly integrated into existing vision systems like ours. The second crucial benefit is that the number of weights in the network is much smaller, because the heavy fully connected output layers are basically replaced by much

lighter (de)convolutional layers. Finally, a FCN can in theory be applied to image patches of any size, so can be run on smaller regions of interest after a low level preprocessing step, to save further computation time.

Besides this conceptionally different approach, we also work on more technical optimisation: the recent explosion of deep learning applications is partly fuelled by the power of GPU computation. It is not feasible to integrate the latest generation of energy hungry desktop GPUs into football robots yet, but more and more smaller footprint and embedded solutions start to become available. For instance, the Odroid XU4 platform our team is experimenting with contains a Mali-T628 MP6 ARM GPU capable of running OpenCL computations. We are working to develop specialised CNN implementations for this hardware to help obtain the required frame rate.

4 Stability

Maintaining balance and stability is another problem that has become much more prominent these years, due to the use of artificial turf as playing surface. The robots are no longer able to rely on a relatively hard and stable flat ground.

Firstly, the robots must be able to judge the current orientation. For this, we previously implemented a method to fuse measurements of the tri-axes gyroscope and the accelerometer to keep track of the orientation of the robot, using a gradient descent-based algorithm [5]. This algorithm operates by integrating changes in orientation as measured by the robot’s gyroscope, corrected by evidence of the orientation measured by the accelerometer, all represented by quaternions:

$$\mathbf{q}_{est} = \gamma \mathbf{q}_\omega + (1 - \gamma) \mathbf{q}_\nabla, \quad (1)$$

where \mathbf{q}_ω is the orientation derived from the gyroscope, \mathbf{q}_∇ is the orientation found by performing a gradient descent step, and $\gamma \in [0, 1]$ is a trade-off parameter. It turns out that a single gradient descent step at each cycle, rather than performing steps until convergence which would be too expensive, is sufficient to minimise the error over time [5].

Next, a measure of stability is defined. We have continued our work done last year on a method based on the rate of change of angular momentum[1]. The essence of this idea is that for full body stability, if the full moment resulting from all forces applying onto the robot sums up to zero at the Centre Of Mass (COM) of the robot; otherwise there is a change in angular momentum rate, meaning an increase of full body rotation. As shown in Fig. 2, there is a point (A) where a ground reaction force with a given direction should apply in order for it to pass through the COM, which is required to arrive at zero momentum.

Unfortunately, it is difficult to measure the centre of pressure (P) and the direction of the ground reaction force (R) directly; only one of our robots is equipped with foot sensors, and those only provide an estimate of (P), not of (R). However, if one realises that the acceleration of the COM is proportional to the sum of gravity and the ground reaction force, the direction of R can be estimated by the difference between measurement acceleration and gravity. The

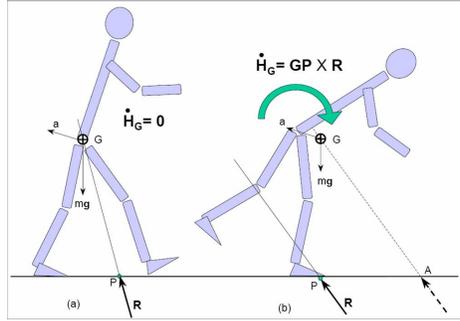


Fig. 2. Depiction of dynamics of humanoid robot and their effect on its stability. Fig. (a) shows the ground reaction force (R) passing through the centre of mass (G), resulting in conservation of full body rotation. Conversely, Fig. (b) shows a scenario where R does not extend through G, causing change in rate of momentum. The point A shows where the ground reaction force *should apply*, and the distance between A and P is a measure of instability. Adapted from Goswami and Vinutha[1].

direction of gravity can be deduced from the orientation filter described above, and acceleration of the COM can be estimated from the robot’s accelerometer in the robot’s torso. Projecting along the direction deduced in this way, again using the filtered orientation, one arrives at an estimate of A. It is known that to be stable, this point should at least fall within the robot’s support polygon, so the distance of this point from the border, or centre, of this polygon can be used as a measurement of stability.

Initial results have shown that tracking and controlling this reference point enables the robots to remain stable to a level similar to when using the COP directly, which is shown in action in Fig. 3. This year we have been working on integrating this measure in more aspects of our motion modules, as well as using reinforcement learning methods (within the Options based framework described below) to remain stability over extended times based on the stability reference point.

5 Behaviour and Action Selection

Our behavioural framework is based on the so-called *Options* model[8]. An *option* is a higher level abstraction of an action, specifically forming a temporal abstraction: an option is not necessarily instantaneous such as low level control actions, it can be active over a length of time.

Formally, an option is described by three components: an *initiation set* \mathcal{I} that determines in what states the option can be selected; a *policy* π that describes the action selection when this option is selected; and a *termination condition* β that (stochastically) determines when the option ends. The case where the policy directly selects primitive actions amounts to a single level of abstraction. However, one can also allow the selection of other *sub-options*, incurring a recursion



Fig. 3. Robots actively maintaining stability. Robot 2 (right) is equipped with FSR foot sensors and controls measured COP to be in the centre of support polygon. Robot 5 (left) has no foot sensors but derives stability reference point from IMU and orientation and controls this reference point to be within support polygon

that forms an unlimited hierarchy. Finally, primitive actions can be modelled as options that always terminated after one step, ensuring we can use a single structure throughout the whole hierarchy.

The Option model was first formulated in the context of Reinforcement Learning, and has been used and further developed in that area of research extensively for the last 15 years. However, without applying RL the framework is abstract enough to allow for a multitude of implementations of the options, like finite state machines, artificial neural nets or decision trees.

6 Sensorimotor Modelling

One of the challenges with iterative body model design is the time cost of remodeling the robotic dynamics for every newly generated model. As part of the team effort, we therefore undertake the development of novel techniques for sensorimotor modelling with the mid-term goal to achieve a more generic and lightweight intelligent adaptation to novel robotic hardware. The challenge of recalibrating in short cycles requires a rethinking of robot control development: it needs to take care of quick and cheap adaptation to new hardware and environments. For this, traditional control methods need to be complemented by more generic approaches to robot control.

As one of the research agendas of the team, we investigate a number of concepts which have been developed specifically for the purpose of generic behaviour-generation, especially *empowerment* [3]; empowerment was originally developed as a “taskless utility”, first demonstrated in a highly abstracted RoboCup-inspired scenario [2]. Empowerment is an information-theoretic nonlinear generalisation of the control-theoretic concept of the combined *controllability-observability* matrix and has been successfully used for a variety of generic control scenarios, both discrete and continuous [6]. It can be interpreted as the “degree of freedom” locally available to an agent, or, alternatively, as the efficacy of the external part

of the agent’s sensorimotor loop, and can be used in lieu of a reward function to drive the agent’s behaviour by moving towards regions of higher empowerment. Notably, it does not need an explicit goal or fully mapped utility landscape to be computed, but still acts as a “natural” value function in many of the studied scenarios.

Current work has established a link between traditional control methods and empowerment (to be published). This will accelerate adoption of the methodology and incorporation into engineering, and our RoboCup team will offer a main opportunity to do so. In particular, we are interested in expanding the use of empowerment to deal in a generic fashion with the challenges posed by varying and ill-defined robotic dynamics — being “ill-defined” is, in a way, an unavoidable feature of robots designed by rapid prototyping and 3D-printing, since both tolerance and specifications are likely to vary much more strongly from any pre-planned setup. Our goal is to investigate the empowerment measure to generically produce a number of behaviours, most prominently stabilisation, but possibly also other desired behavioural patterns with the purpose of avoiding to have to hand-design them at each hardware development iteration.

One challenge that needs to be overcome for the purpose of using empowerment is the forward modelling. Empowerment, different from reinforcement learning, does *not* need a reward function, and neither does it require a global model of its operation space, but it *does require* a local forward model. Such a model could be externally provided, but it would be attractive to make use, in agreement with the spirit of the sensorimotor interpretation of empowerment, of purely robot-*internal* information to construct it.

7 Experience and Achievements

Our team has been active in RoboCup since 2002, and hopes to participate in the Humanoid league in 2017 for the fifth year in a row. Besides the world championship, we always aim to participate at as many open competitions and related workshops and meet-ups.

The following are the detailed achievements and contributions of team Bold Hearts in the Humanoid League over the last few years.

- 2nd round Robocup World Championship 2016 (1st in group)
- 1st Iran Open 2016
- 2nd round Robocup World Championship 2015 (1st in group)
- 3rd German Open 2015
- 2nd Robocup World Championship 2014

8 Acknowledgements

Team Bold Hearts would like to acknowledge the crucial open source projects used to develop our team: OpenSCAD, libwebsockets, GitLab, RapidJSON, and Google-Test¹.

We would also like to extend our gratitude to team NUbots, for donating spiked feet pads created by them to help provide our robots with enhanced walking and stability, and inspiration for further development.

References

1. Ambarish Goswami and Vinutha Kallem. Rate of change of angular momentum and balance maintenance of biped robots. In *Proceedings of the 2004 IEEE International Conference on Robotics & Automation*, volume 4, pages 3785–3790, 2004.
2. Alexander S. Klyubin, Daniel Polani, and Chrystopher L. Nehaniv. All else being equal be empowered. In *Advances in Artificial Life, European Conference on Artificial Life (ECAL 2005)*, volume 3630 of *LNAI*, pages 744–753. Springer, 2005.
3. Alexander S. Klyubin, Daniel Polani, and Chrystopher L. Nehaniv. Keep your options open: An information-based driving principle for sensorimotor systems. *PLoS ONE*, 3(12):e4018, Dec 2008.
4. Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3431–3440, 2015.
5. S.O. Madgwick, A.J. Harrison, and A. Vaidyanathan. Estimation of IMU and MARG orientation using a gradient descent algorithm. In *IEEE International Conference on Rehabilitation Robotics*, page 2011:5975346, 2011.
6. Christoph Salge, Cornelius Glackin, and Daniel Polani. Empowerment — an introduction. In M. Prokopenko, editor, *Guided Self-Organization: Inception*, volume 9 of *Emergence, Complexity and Computation*, pages 67–111. Springer, 2014. Preprint available as <http://arxiv.org/abs/1310.1863>.
7. Daniel Speck, Pablo Barros, Cornelius Weber, and Stefan Wermter. Ball localization for robocup soccer using convolutional neural networks. In *Springer Lecture Notes on Artificial Intelligence*, volume 9776, 2016.
8. Richard S Sutton, Doina Precup, and Satinder Singh. Between mdps and semi-mdps: a framework for temporal abstraction in reinforcement learning. *Artificial Intelligence*, 112(1-2):181–211, 1999.

¹ <http://www.openscad.org>, <https://libwebsockets.org/>, <https://gitlab.com/>, <http://rapidjson.org/>, <https://github.com/google/googletest>