

Bold Hearts Team Description

RoboCup 2016 Kid Size

Sander G. van Dijk, Daniel Polani, Marcus Scheunemann, and Daniel Barry

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

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

1 Team and Robots

Team Bold Hearts, from the Adaptive Systems Research Group at the University of Hertfordshire, UK, has participated in RoboCup since 2002. Initially solely in the simulation leagues, we spread out to the kid-size humanoid hardware league in 2013. Up to the World Championship of 2015 in Hefei, China, we have used the standard DARwIn-OP robotic platform to make up our robotic team. To keep up with the changing and more challenging environment that the team has to play in these years, we started work to firstly adapt the DARwIn-OP framework, and secondly to extend our team with whole new robots.

The main facilitator in these efforts is the use of 3D printing technology. This allows us to rapidly prototype alterations to a robot's body parts and develop new platforms. In this light we have opted to base the extension of our team on the *Hambot* platform[1]. With the possibility to print all major body parts, and to reuse a large amount of DARwIn-OP parts, we felt that this platform will allow us to grow in the league fastest.

Together with the original developers, the Hamburg Bit-Bots, we have submitted a proposal for RoboCup Federation Support, to help further developing this open platform, and equip it with a complete open software and behaviour framework in order to lower the threshold for other teams to use and benefit from this platform even further. We are very happy that our proposal has been accepted by the RCF Evaluation Committee and we look very much forward to be able to continue this and present the results at the World Championship in Leipzig, 2016.

Algorithm 1 EXECUTEOPTION

Require: $\langle \mathcal{I}, \pi, \beta \rangle$

- 1: **repeat**
 - 2: Observe current state s
 - 3: Select sub-option $o \in \mathcal{I}$ according to probability distribution $\pi(s, \cdot)$
 - 4: EXECUTEOPTION(o)
 - 5: **until** $\beta(s) \geq \mathcal{U}(0, 1)$
-

2 Behaviour and Action Selection

In this section we will discuss our behavioural framework, which has grown out of the experience obtained in the simulation league where the focus lies heavily on this aspect of the RoboCup challenge. From this experience we have extracted some properties that a useful behaviour framework should have: it should be highly modular so separate parts can be developed in parallel, different pieces of behaviour can easily be reused, and behavioural solutions with different implementations can be interchanged; it must be as nonrestrictive as possible, as not to hit future limitations; hierarchy is important, to be able to build up complexity over time; and finally it must directly enable the use of well developed machine learning techniques.

Our behavioural framework is based on the so-called *Options* model[12]. The final outcome of a behavioural framework is the selection of primitive actions: actions that are instantaneously executed, usually once per some defined time step, or think cycle. In our scenario these actions consist of direct motor control, or acts of communication. More complex behaviour however consists of tightly connected series of such basic actions, and to make such behaviour feasible it must be described in the form of higher levels of abstraction. An *option* is such a higher level abstraction of an action, specifically forming an abstraction over time: an option is not necessarily instantaneous, 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 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. Algorithm 1 lists the full process of executing an option.

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 [9, 6, 7, 11], which will allow us to apply and further develop this work in the context of RoboCup. However, the framework is abstract enough to allow for a multitude of implementations of the options. For

instance, the ubiquitous finite state machine is, in our framework, implemented as an option where the policy is determined by the state of the machine, and the termination condition is triggered by the state machine transiting into some final state. Other action selection mechanisms such as artificial neural nets or decision trees fit into this framework just as easily.

3 Vision

The vision system provides some of the most critical data for building the agent's belief about the state of its environment. One problem is to be able to process large amounts of vision data in real time on limited computational platforms. To overcome this, in previous years we developed a sampling technique that applies a uniform spacing across the ground plane relative to the agent. Consequently the greatest pixel density is maintained at great distances, while sparse sampling occurs in the foreground where objects are largest on the camera plane. Due to this, and further optimisation of our vision stack, our robots are able to fully process 640x480 pixels within 4ms.

The second problem is that of correctly recognising the environment from this vision data. This problem has become harder in the Humanoid league due to a move away from colour coded environments to more realistic football scenarios. This means we have to develop a new process that is more based on shape and appearance.

The first stage of this process is still similar to the popular traditional methods based on colour segmentation: a number of colour classes are defined and trained, which are used to classify pixels that are sampled. Pixels in the same colour class are then segmented into connected components using a fast run-length encoding method.

In contrast to earlier years however there is no 1-to-1 mapping from colour class to object class: the ball, goal posts and lines are all white. Under certain lighting conditions and with specific materials used at a venue it may be possible to separate some of these classes using the Round Table training tool described above, but most likely at least some will overlap. In this case the connected components are tested and finally classified using shape and position based predicates such as roundness, expected size given robot posture, vertical orientation, and whether or not there are intersections with the visible field edge and/or horizon. The run-length encoding method allows most of these properties to be determined in a single scan of the image, as well as to indicate discontinuities where a connected component should be split up into multiple objects (e.g. where long runs from horizontal field lines connect to vertical goal posts).

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.



Fig. 1. 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

Firstly, the robots must be able to judge the current stability. 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 [8]. 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 $\mathbf{q}_{\omega}$ is the orientation derived from the gyroscope, $\mathbf{q}_{\nabla}$ 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 [8].

Another indicator of stability can be the centre of pressure (COP) as measured by sensors in the feet: the further away from the centre of the support polygon, the less stable the robot is. There are FSR based foot sensors available for the DARwIn-OP platform, however it is expensive to equip the full team with these (we currently have 1 robot using such sensors), and additionally the degree of instability can no longer be judged when the COP is at the edge of the support polygon. Therefor we work on developing a method based on the rate of change of angular momentum[2] to reduce full body rotation, which results in the derivation of a stability reference point similar to the COP (or Zero Moment Point) solely from the smoothing orientation tracking algorithm and the high frequency dynamics measured by the IMU.

Initial results show that tracking and controlling this reference point enables the robots to remain stable to a level similar to when using the COP directly. Current work is ongoing to further develop this method and research it’s properties, as well as using reinforcement learning methods (within the Options based

framework described above) to remain stability over extended times based on the stability reference point.

5 Sensorimotor Modelling

The development of novel hardware, especially with comparatively short turnaround times as is expected by the use of 3D printing, creates new challenges to robotic control. One of them is the time cost of re-modeling the robotic dynamics for every newly generated model. While recalibration is also necessary in repeated use of the same hardware, or in different instances of “cloned” hardware of the same make, it is significantly exacerbated when the design of robots is fluid.

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 movement and behaviour for robots which are printed and configured 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* [5]; empowerment was originally developed as a “taskless utility”, first demonstrated in a highly abstracted RoboCup-inspired scenario [4]. Empowerment is an information-theoretic nonlinear generalization 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 [10]. 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.

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 in this case. Our goal is to investigate the empowerment measure to generically produce a number of behaviours, most prominently stabilization, 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 modeling. Empowerment, different from reinforcement learning, does *not* need a reward function, and neither does it require a global model



Fig. 2. Screenshot of vision debugging. Rectangles: detected objects; small filled squares: object ground positions; orange line: horizon; green line: field boundary; blue lines: detected lines.

of its operation space, but it requires 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. For this purpose, we will investigate recent methods from approximative algebra, a recent branch of applied symbolic algebra [3], for their use to reconstruct the robot’s movement space from measured data. These models will then feed into the determination of empowerment as a local behaviour-generating function.

6 Round Table

For debugging we built an HTML5 application called *Round Table* which runs in modern browsers and is served directly from the robot via HTTP. The core state of the agent is modelled as a set of immutable state objects, and each of these objects is exposed from the agent via WebSockets. Our debugger subscribes for updates to whatever state objects are required for the particular task at hand.

Further, most configuration settings are able to be modified using this browser-based tool. Examples include watching the live video feed, viewing real-time charts of IMU data, 2D and 3D models of the agent’s local and world frames, behavioural reasoning, game state and detailed per-thread timing breakdowns just to name a few.

The website is served directly from the robot. This approach makes it possible to debug using any device with a recent web browser, with the debug interface always up to date with the software running on the robot.

This year we have further enhanced this tool to be able to for instance directly train parts of the vision system, develop and test motion scripts, and visualise more of the agent’s internal state. Furthermore, we have developed a UDP tunnel called *Draw Bridge* that allows one to use the Round Table interface to analyse a robot’s behaviour better during matches.

This also ties in to our efforts to decouple the tool further from specifics of the DARwIn-OP platform, so it could be integrated more easily in the open modular framework being developed for the new platform, and/or released as a separate open source tool that could be freely used elsewhere.

7 Experience and Achievements

Our team has been active in RoboCup since 2002, and hopes to participate in the Humanoid league in 2016 for the fourth 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 2015 (1st in group)
- 2nd Robocup World Championship 2014
- 3rd German Open 2015
- 3rd Iran Open 2014
- 2nd German Open 2014

References

1. Marc "Bestmann, Bente Reichardt, and Florens" Wasserfall. "hambot: An open source robot for robocup soccer". In Luis "Almeida, Jianmin Ji, Gerald Steinbauer, and Sean" Luke, editors, *"RoboCup 2015: Robot World Cup XIX"*, pages 339–346, Cham, 2015. Springer International Publishing.
2. 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.
3. Daniel Heldt, Martin Kreuzer, Sebastian Pokutta, and Hennie Poulisse. Daniel heldt,1, martin kreuzerb,2, sebastian pokuttac,3, hennie poulisse. *Approximate computation of zero-dimensional polynomial ideals*, 2009.
4. 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.
5. 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.
6. George Konidaris and A.G. Barto. Building portable options: Skill transfer in reinforcement learning. In *Proceedings of the 20th International Joint Conference on Artificial Intelligence*, pages 895–900, 2007.

7. O Kozlova, O Sigaud, and C Meyer. Automated discovery of options in factored reinforcement learning. In *Proceedings of the ICML/UAI/COLT Workshop on Abstraction in Reinforcement Learning*, pages 24–29, Montreal, Canada, 2009.
8. 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.
9. A. McGovern and A.G. Barto. Automatic discovery of subgoals in reinforcement learning using diverse density. In *ICML 01: Proceedings of the Eighteenth International Conference on Machine Learning*, pages 361–368, San Francisco, CA, USA, 2001. Morgan Kaufmann Publishers Inc.
10. 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>.
11. Özgür Şimşek and Andrew G. Barto. Skill characterization based on betweenness. In D. Koller, D. Schuurmans, Y. Bengio, and L. Bottou, editors, *Advances in Neural Information Processing Systems 21*, pages 1497–1504, 2009.
12. 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.