Humanoid Robots and Cognitive Systems Research: An Epistemological Case Study Based on the iCub

Bidan Huang,
Artificial Models of Natural Intelligence
University of Bath, BA2 7AY, UK
Email: b.huang@bath.ac.uk

Jason L. Leake
Bath Institute of Medical Engineering
Bath BA1 3NG, United Kingdom
Email: jason@leake.me.uk

Joanna J. Bryson
Artificial Models of Natural Intelligence
University of Bath, BA2 7AY, UK
Email: j.j.bryson@bath.ac.uk

I. Introduction: Background and Approach

We present a case study concerning our experience of constructing a basic manipulation behaviour on one of the currently-dominant cognitive robotics research platforms. This work has led us to also examine the question—Are the goals of cognitive systems best addressed through the use of humanoid robotics? *Cognition* is defined as the ability to perceive, learn and reason, and is expressed through action. A *cognitive system* achieves this through a variety of cooperating subsystems. *Embodied cognition* is an approach holding that intelligence likely to be useful to humanity must share our perspective and problems [1]. Its wide-spread acceptance motivates effort in a difficult and expensive but compelling discipline: the building of human-like robots. The *de facto* standard for embodied cognition research in Europe is the iCub, with around 20 robots delivered. It is a high-end humanoid with the proportions of a 3.5 year old child [2].

Another area of consensus is the importance of modular decomposition in making a system both tractable to human design and maintenance, but also more probable and amenable for natural mechanisms such as evolution and individual learning[3]. An important goal of artificial cognitive systems is to build a robot that is as easy to direct on a new task as a human. Modularity provides a useful framework for achieving this goal. Under this framework, cognitive components—sensing and memory—are built into action thereby generating a robust and scalable intelligence.

Behaviour Oriented Design (BOD) is a methodology for constructing modular cognitive systems [4]. The BOD architecture consists of: *a)* A library of behaviours, implemented as objects, and *b)* A reactive plan, which in BOD is a POSH (Parallel-rooted, Ordered, Slip-stack, Hierarchical) plan. Each behaviour object represents a perception/action/learning module and implements the actions needed to carry out its tasks. Primitives called from POSH are methods on OOD objects coding behaviour modules. Memory and learning are implemented in these module, to support action and perception.

Our original goal was to examine BOD's utility for humanoid robotics. We present work conducted by two individuals each with good degrees and years of professional programming experience, but whose previous robotics experience was limited to intensive week-long robotics summer schools.

II. Phase I: Simulation

One reason the iCub is presented as an entry-way to cognitive systems is its open-source, community-supported simulator. In theory one can build prototype iCub code, then transfer this to one of the 'public' iCubs such as are available at the annual summer schools. In practice our experience supports the more common roboticist wisdom that starting with a simulator can lead to ill-grounded expectations.

We chose to use the Python version of the POSH engine [5]. The iCub middleware package YARP [6] had an existing Python interface but it only provides access to low level functionality which is normally accessed by higher level iCub-specific code. We therefore produced a higher-level Python interface using C++ to communicate with this iCub software.This new interface allowed the Python POSH engine to control the robot and permitted the BOD behaviours to be implemented quickly.

We hoped to test BOD in the context of a non-trivial task. Ultimately the task became locating and picking up blue and yellow cubes with the right hand. Blue cubes would be dropped onto the floor to the right of the robot and yellow cubes passed to the left hand and dropped to the left of the robot. We defined the simulated robot's working environment and task-approach using BOD's iterative development. In lieu of 3-D vision, the table height was found by physical exploration—iCub's hand was placed above the table and lowered until the touch sensors fired. The cube was found by colour vision. Having located the cube the hand was moved towards it using visual feedback. This requires handling error conditions properly, for example if the hand moves out of view. With such a task, one of the greatest benefits of using an iCub is its comprehensive set of libraries. For the simulator work the iCub's iKin forward and inverse kinematics library was used, which contain iCub-specific solvers.

Although development time varies greatly with individual and experience, it is still useful to communicate some level of expectation of how long a project might take. The major activities of this phase can be summarised as follows (in weeks): Setting up environment etc. 1 wk; Writing and refactoring the POSH/iCub interface 2 wks; Inverse kinematics 1.5 wks; Grasping 2wks; Vision processing; 1wk; POSH plan and Python behaviour modules 1wk; iterative testing activities such as simulator modifications 3wks; **total** 3 months.

We found inverse kinematics slowed development, and vision processing was significantly simplified by the simulator. We worried about the extent the vision could be re-used for the robot, but OpenCV proved to be intuitive and flexible. Interfacing the Python POSH engine to the iCub was far more straightforward than anticipated. Grasping was the major problem, because the simulator lacked several features necessary to do it. The absence of force control meant that although the robot's fingers closed around the object it was necessary to use a magnet feature in the simulator to 'glue' the cube to the hand to lift it reliably.

III. Phase 2: Robots

The objective of this phase was to port the code running in the simulator to the real robot. There are significant differences between the iCub simulator and the real robot. The simulated visual scene is poor in lighting and surface effects. The simulation does not include positional errors caused by mechanical imperfections such as whiplash, so we neglected to use sufficient proprioception to track the real location of the robot. In general compliance in the robot and objects in its world is not easy to represent. In particular, there is no provision for measuring forces.

What surprised us more was that the first two physical iCub robots we experimented with lacked both force *and* touch sensing. This eliminates one of the most basic aspects of cognition—feedback on the results of your actions—cannot be achieved except possibly by over-reliance on vision. Natural cognition relies on sensor fusion and feedback, even though it often uses open-loop control to at least initiate a gesture.

Our attempts to implement our task on a real robot occurred at Imperial University and Plymouth University. Varied estimates from experienced iCub engineers suggested transferring the simulation would take a few days or a few weeks. The tasks and time (in days) required for this were: Installing (and learning) Linux, installing iCub software and getting the simulator running, 11 d; Studying the simulator version of the code, 3 d; Learning YARP and using YARP to communicate with the iCub 5 d; Learning basic image processing and OpenCV 5 d; Changing the vision to work with the real robot 4 d; Replacing the touch sensor strategy in the original program with visual feedback 6 d; Reading data from iCub and running the code on the iCub for testing 2 d; Moving the iCub's arm to the desired position 2 d; **total** approximately 5 weeks, and the task was not fully achieved.

The lack of noise in vision and actuation in the simulator lead to later problems. We were surprised that the iCub which has a large software community provided no general libraries for robust object tracking. The lack of touch and force sensing was problematic. During experiments a motor failed when moving a joint to its maximum angle—this should be avoided, and emphasised that the sophistication of the robot is bought at a price of fragility. Similarly, iKin plans its path without taking the iCub's body into account, meaning it might plan a path through the body which would seriously damage the robot. Although the simulator should be used to test code before use on the robot, this is by no means sufficient to guarantee safe operation, and any motion of the real robot must be supervised.

IV. Discussion and Conclusions

Overall, the iCub proved to be a good platform for our case study. It has a large software library, a functional simulator and there are many users able to offer advice. Still, the time taken to develop a such a simple task was surprising for what is one of the most advanced cognitive systems platforms. The iCub has demonstrations developed by many institutions at considerable research expense and provided open-source. Whilst admittedly a great deal of the time expended was due to our inexperience with robotics, we spent considerable time talking to experts and we were familiar with the literature on the "right" approach to robotics from the beginning.

Our experience leads us to question whether bespoke research robots is the only or even best way forward for cognitive systems research. We next plan to explore the usability of industrial robots for cognitive systems. There is also a question as to what is more human-like—a robot with two arms and a head, or one with sufficient motor control, eye-hand coordination and sensing to reliably grasp a pin? A nearly-blind, one-armed robot may be more capable of a typically animal-like cognitive grasping strategy than even the most advanced purpose-built cognitive systems platform. Contemporary industrial robots can be taught basic gestures quickly, are rugged but capable of precise action, and force control is an off-the-shelf component.

In conclusion, while many people consider that grasping is already "solved"—that is, papers and videos exist which demonstrate it. But until we can build such 'simple' operations as easily, reliably and robustly into a new task as a one-year-old infant can, a cognitive component is no sense solved.

Acknowledgement

Our thanks to Tony Belpaeme, Tony Morse, Yiannis Demiris, Yan Wu, Kyuhwa Lee, Paul Fitzpatrick and Giorgio Metta for their time, iCubs and laboratory space.

References

[1] R. A. Brooks and L. A. Stein, "Building brains for bodies," *Autonomous Robots*, vol. 1, no. 1, pp. 7–25, 1994.

[2] N. Tsakarakis, G. Metta, G. Sandini, D. Vernon, R. Beira, F. Becchi, L. Righetti, J. Santos-Victor, A. Ijspeert, M. Carrozza, and D. Caldwell, "iCub – the design and realization of an open humanoid platform for cognitive and neuroscience research," *Journal of Advanced Robotics*, vol. 21, no. 10, pp. 1151–1175, 2007.

[3] J. J. Bryson, "Cross-paradigm analysis of autonomous agent architecture," *Journal of Experimental and Theoretical Artificial Intelligence*, vol. 12, no. 2, pp. 165–190, 2000.

[4] J. J. Bryson and L. A. Stein, "Modularity and design in reactive intelligence," in *Proc* 17^{th} *International Joint Conference on Artificial Intelligence*. Seattle: Morgan Kaufmann, August 2001, pp. 1115–1120.

[5] J. J. Bryson, T. J. Caulfield, and J. Drugowitsch, "Integrating life-like action selection into cycle-based agent simulation environments," in *Agent 2005: Generative Social Processes, Models, and Mechanisms*, M. North, D. L. Sallach, and C. Macal, Eds. Chicago: Argonne Nat. Lab., October 2005, pp. 67–81.

[6] G. Metta, P. Fitzpatrick, and L. Natale, "YARP: Yet another robot platform," *International Journal on Advanced Robotics Systems*, vol. 3, no. 1, pp. 43–48, 2006.