Abstract. RoboCup inspires and motivates our research interests in cognitive robotics and machine learning, especially vision, state-estimation, locomotion, layered hybrid architectures, and high-level programming languages. The 2014 rUNSWift team comprises final year undergraduate honours students, Master and PhD students, past RoboCup students and supervisors who have been involved in RoboCup for over a decade. New developments in 2014 include machine learning robot recognition, improved cascading and foveation in vision to see far-away features, revised locomotion and kicking, shared Kalman Filter state-estimation for localisation, a coach robot, rearchitected behaviour and skills, and challenge specific innovations.

1 The Team

The RoboCup Standard Platform League (SPL) is an excellent training ground. Participant team members need to collaborate on the development of a highly complex software system, deliver it on time, and within budget. This year the UNSW SPL team comprises a mix of undergraduate, masters and PhD students, both past and present. Several of our team members have full-time jobs in industry and have come back to participate in the 2014 competition by devoting their own time and assisting a new crop of students. By participating in RoboCup the team is engaged in a unique educational experience and makes a significant contribution towards research, as is evident from the list of references at the end of this article.

The 2014 rUNSWift team members are Luke Tsekouras, Jaiden Ashmore, Zijie Mei (Jacky), Belinda Teh, Oleg Sushkov, Ritwik Roy, Roger Liu, Sean
Harris, Jayen Ashar, and faculty members Brad Hall, Bernhard Hengst, Maurice Pagnucco, and Claude Sammut.

The team has the financial support of the School of Computer Science and Engineering at the University of New South Wales. The School provides considerable organisational support for travel. The team benefits from a wealth of experience and the legacy code from past past rUNSWift teams in the standard platform league (including the previous Sony four-legged league), simulation, and rescue competitions.

A UNSW team has taken part in every RoboCup competition since 1999. In the following sections we describe our broader research interests and list our contributions over the years. Team reports, code and videos are available at internet address: http://www.cse.unsw.edu.au/about-us/help-resources/for-students/student-projects/robocup/.

2 Research Interests

The vision of many robotics researchers is to have machines operate in unstructured, real-world domains. Our long-term aim is to develop general-purpose intelligent systems that can learn and be taught to perform many different tasks autonomously by interacting with their environment. As an approach to this problem, we are interested in how machines can compute abstracted representa-
tions of their environment through direct interaction, with and without human assistance, in order to achieve some objective. These future intelligent systems will be goal directed and adaptive, able to program themselves automatically by sensing and acting, and accumulating knowledge over their lifetime.

The School of Computer Science and Engineering at the University of New South Wales is arguably the premiere Australian computing school. Autonomous Systems is a priority research area for UNSW. Our general research focus, of which the RoboCup SPL is a part, is to:

- further develop reasoning methods that incorporate uncertainty and real-time constraints and that integrate with the statistical methods used in SLAM and perception
- develop methods for using estimates of uncertainty to guide future decision making so as to reduce the uncertainty
- extend these methods for multi-robot cooperation
- use symbolic representations as the basis for human-robot interaction
- develop learning algorithms for hybrid systems, such as using knowledge of logical constraints to restrict the search of a trial-and-error learner and learning the constraints
- develop high level symbolic robotic languages that provide abstractions for a large range of deliberation, planning and learning techniques so as to simplify robot programming

3 rUNSWIFT 2014 Robotic Architecture

The rUNSWIFT robotic architecture shown in Figure ?? is a task-hierarchy for a multi-agent team of five Naos. We use a fault-tolerant network-centric architecture. This means that each robot may have a slightly different view of the world and therefore of its role on the team. The approach has the advantage of providing some redundancy in case individual robots are disqualified or stop working.

Starting at the root-level the game-controller invokes the high-level states for playing soccer. At lower levels, the vision system makes sense of the kaleidoscope of pixel values at 30 frames per second, and the walk generators execute temporally extended walk actions that invoke primitive state transitions as the robot transitions between poses 100 times each second.

This year for the first time the rules include a coach robot that can oversee the game. The coach sits on the game-controller table, and is only allowed to provide higher-level strategic support to the team from this vantage point.

3.1 Vision

Our vision system evolved significantly over the last fifteen years. From the beginning, in 1999, we used a simple learning system to train the colour recognition system. In 2001, we used a standard machine learning program, C4.5, to build
a decision tree recogniser. Also in 2000, our vision system became good enough to use robot recognition to avoid team mates (Sammut & Hengst, 2003).

In recent years, we have updated the vision system to recognise the field-boundary, field-markings and to rely less on colour by using edge-features. We have introduced a foveated vision system and virtual saccades to maximise scarce computational resources. To disambiguate the symmetric field we introduced a natural landmark recognition algorithm that is able to localise the robot in real-time. This tailored SLAM approach, based on one-dimensional local image features was presented at last year’s RoboCup Symposium (Anderson, et al, 2012). A spin-off development – a visual compass to help odometric corrections for adjusting for slippage and collisions, was presented at the RoboCup Symposium last year (Anderson and Hengst 2013).

This year we developed a new goal detection module and utilised high resolution foveas to track far-away vision landmarks. We upgraded the camera interface to allow access to the full resolution image in the top camera. We aim to be able to detect goal posts from anywhere on the field and be able to track field-line features over 5m away.

**Localisation** The 2000 competition saw the initial use of a Kalman filter-based localisation method that continued to evolve in subsequent years (Pham et al, 2002). The localisation system evolved to include a multi-modal filter and distributed data fusion across the networked robots. In 2006, we went from treating the robots as individuals sharing information, to treating them as one team with a single calculation spread over multiple robots. This allowed us to
handle multiple hypotheses. It also allowed us to use the ball for localisation information.

A significant innovation in 2012 was the use of a variation of the Iterative Closest Point (ICP) algorithm extending previous work on field-line matching (Sheh & Hengst, 2004; Ratter, 2011) to other visual features and objects. This novel algorithm was presented at ICRA (Peter Anderson, Youssef Hunter, Bernhard Hengst, 2013). A summary of innovations for 2012 was presented at ACRA in 2012 (Harris, et al).

This year we are porting the 2006 distributed multi-modal Kalman filter localisation innovations developed for the AIBOs to the Naos. We track multiple hypothesis modes for the pose of the robots and ball on the field. Each hypothesis mode consists of the robot pose, ball position and velocity, and the poses of the teammate robots. The robots exchange observations of field lines, goal posts, and the ball with each other over wireless. In this way a robot can incorporate teammate observations into its own filter and improve its localisation accuracy and consistency. This also helps to disambiguate the two symmetric sides of the field.

**Locomotion** In 2000, we introduced the UNSW walk, which became the standard across the four-legged league (Hengst et al, 2002). Almost all the other teams in the 4-legged league at the time adopted a similar style of locomotion, some starting from our code. The flexibility of this representation led to another major innovation in 2003. We were the first team to use Machine Learning to tune the robot’s gait, resulting in a much faster walk (Kim & Uther, 2003). In succeeding years, several teams developed their own ML approaches to tuning the walk.

Bipedal locomotion research in our group includes the application of Machine Learning to gaits. PhD student Tak Fai Yik (a member of the champion 2001 four-legged team) collaborated with Gordon Wyeth at the University of Queensland to evolve a walk for the GuRoo robot (Wyeth, et al, 2003), which was entered in the humanoid robot league. We have continued to research bipedal locomotion methodologies and learning strategies (Hengst, et al, Humanoids 2011; Hengst, CLAWAR 2013).

This year machine learning using a Nao physics simulation has inspired the development of a new walk and balancing mechanism for use in competition. In addition, our Open Challenge submission is the development of a bipedal heel-to-toe walk using the Nao. Our aim is to start the development of a more natural walk that is more efficient than typical robotic walks. A characteristic is the complete extension of the support leg during all phases of the walk. It is expected that stability will be improved by making use of a lengthy double support phase.
Software Engineering and Architecture Throughout the software development of the AIBO and then Nao code, we have adopted a modular, layered architecture. The lowest layers consist of the basic operations of vision, localisation and locomotion. The behaviours of the robots are also layered, with skills such as ball tracking, go to a location, get behind ball, etc, being at the lowest level of the behaviour hierarchy, with increasingly complex behaviours composed of lower-level skills. Originally, all the behaviours were coded in C/C++ but in 2005 and 2006, as from 2010 onwards the upper layers were replaced by Python code.

One of the key reasons behind the UNSW team’s success has been its approach to software engineering. It has always been: keep it simple, make the system work as a whole and refine only what evidence from game play tells us needs work. This practical approach has had a strong effect on our research because it has informed us about which problems are really worth pursuing and which ones are only imagined as being important.

4 Participation and Performance

A UNSW team has taken part in every RoboCup competition since 1999. Details of awards are as follows:


- 3rd place: 2005, 2012
- 4th place: 2013
- Challenges: 2nd in 2003, 2012
- Challenges: 3rd in 2011

Simulation soccer: 2001 - 2003

- 7th place: 2002


- 3rd overall: 2005
- Semi-finalists and 2nd in autonomous robot challenge: 2006
- 2nd in Mobility: 2009
- Award for innovative user interfaces: 2009
5 Acknowledgements

The 2014 rUNSWIfit team wishes to acknowledge the financial and organisational support from the School of Computer Science and Engineering, University of New South Wales, and the contribution to the code base from past rUNSWIfit teams that has been integrated into the latest source code.

References

RoboCup SPL related publications:

7. Robot Localisation Using Natural Landmarks, Peter Anderson, Yongki Yusmaithia, Bernhard Hengst, Arcot Sowmya, RoboCup Symposium, 2012
9. Learning Ankle-Tilt and Foot-Placement Control for Flat-footed Bipedal Balancing and Walking, Bernhard Hengst, Manuel Lange, Brock White, Humanoids, Bled, Solovinia, 2011
12. 1999–2009 team reports as well as selected code are available at URL: http://www.cse.unsw.edu.au/about-us/help-resources/for-students/student-projects/robocup/