

# NUIM Team Description for 2010 RoboCup Standard Platform League - Nao RoboEireann

Prof Rick Middleton  
The Hamilton Institute  
The National University of Ireland Maynooth  
Co Kildare, Ireland  
<http://www.nuim.ie/robocup>  
email: Richard.Middleton@nuim.ie  
February 2010

No Institute Given

## 1 Team Leader and Team

### 1.1 Team Leader

Rick Middleton <http://www.hamilton.ie/rick/>

### 1.2 Academic Team Members

The team combines researchers from Computer Science, Electronic Engineering and the Hamilton Institute (an applied mathematics research institute, <http://www.hamilton.ie>). Academic team members and research interests are described below.

Name	Department	Research Interests
Dr Robert Lawlor	EE	Signal Processing
Mr John McDonald	CS	Vision & Image processing
Dr Sean Mcloone	EE	Intelligent Systems & Control
Dr Seamus Mcloone	EE	Intelligent Systems, Distributed Interactive Media
Mr Fiachra Matthews	Hamilton	BCI, Imaging, Signal Processing
Prof Rick Middleton	Hamilton	Control, Signal Processing, Robotics
Dr. Diarmuid O'Donoghue	CS	Analogical Algorithms, Genetic Optimisation
Prof Barak Pearlmutter	CS/Hamilton	The Brain & Computation, Machine Learning
Prof John Ringwood	EE	Mathematical modelling, Feedback Control Systems
Dr Selim Solmaz	Hamilton	Feedback Control, Dynamics, Intelligent Systems
Mr Rudi Villing	EE	Perceptual Signal Processing, SW Engineering
Dr Adam Winstanley	CS	Intelligent Systems & Signal Processing

### 1.3 Student Team Members

Name	Degree	Project
Alex Buckley	M.Sc. (Hamilton)	Kicks/Locomotion
Ciaran Burke	B. Eng. (EE)	Challenges
Aodhan Coffey	B. Eng. (EE)	Vision
Paul Murray	B. Eng. (EE)	Behaviours
Sean Redmond	B. Eng. (EE)	Locomotion
Tom Whelan	B. Sc. (CS)	Vision

## 2 Recent Team Progress and Interests

### 2.1 Background

In 2008, we were part of the joint Newcastle/Maynooth team, *NUManoids*, who won the world championship. In 2009, we decided to field an individual team to compete and to rewrite the code for our new team. Unfortunately, with limited personnel in 2009 (one (new to RoboCup) person full time from Feb 2009, others only available after May), we were not able to successfully integrate code to form a complete working system. In 2010, we competed in the German Open and came 6th out of 12 teams.

### 2.2 Robot Vision

Some of the key challenges of vision in this environment are the real time, limited CPU, rapidly moving camera (active perception) nature of the image data available. Our vision system in 2009 was based on openCV, which allows many advanced computer vision algorithms to be implemented. However, our experience to date is that it has been difficult to obtain the speed needed for high levels of performance in the SPL. It has proven useful for generating colour segmentation algorithms. We have algorithms based on optimisation working in the HSI space for captured images. The system seeks the optimal HSI space bounds for each colour to minimise a compromise of false alarms and misses in colour segmentation. This is then used to generate a look up table in the robot colour space for colour segmentation.

We also have preliminary experience with image correlation based algorithms for line/goal detection. Image correlations have the advantage of being highly robust to noise and occlusions in classifying fixed field objects. Initial algorithms have shown promising performance, though with significant demands on memory, and high CPU demands. We are examining working with sparse computations of image correlations to improve performance.

Related past publications from the team include [1], [2], [3].

### 2.3 Locomotion

It was clear from last years RoboCup that the walk was one area that could be dramatically improved on. The main problems we identified from a locomotion point of view from last year were the robots were taking too long to get to the ball, they would often have to stop and change direction, and when they got to the ball they were taking too long to position themselves for a kick. Our aim was to create a stable walk for the Nao to allow it to dynamically change direction without having to stop. That is, if during a walk, vision noticed that the robot was off course, we could adjust without having to stop. Another goal we set ourselves was to add in some dynamic balancing to the walk cycle. If the robot was about to fall over in the middle of a walk, we wanted it to be able to correct itself.

**The Walk Engine** Given the time constraints, constructing our own walk from scratch would have been ambitious. So we decided to use the Aldebaran walk as a template to build from. The first step was to use the Aldebaran walk engine to create a stable walk for the Nao on our carpet, second to recreate that walk without using any Aldebaran functions, and finally to improve on, and add as much functionality to that walk as possible.

We had some difficulty adjusting the walk to the carpet in our robotics lab. The heavy friction and inconsistent pile introduced oscillations into the walk that the robot couldn't recover from. We found however that by adjusting the hip height and the joint stiffness values especially, we could almost eradicate these oscillations from the walk. Some final adjustment to the torso orientation (so the robot would lean into the walk) led to quite a stable open loop walk for our carpet. As mentioned, the hip height was a huge factor in creating a stable walk, we found that on thicker carpets, the hip height needed to be lowered to keep stable. The joint stiffness values for each joint were kept only just above the minimum values needed for the Nao to support it's own weight. For example, the major supporting joints such as the knee and hip pitch were kept at approximately 60% of the maximum, whereas the ankle roll joint was kept as low as 25% to allow the foot to roll on impact, keeping the whole foot on the floor.

We sampled the body angles of the robot and found, excepting the start and end phases, that during the walk the joint angles were periodic and, for the most part, resembled sinusoids. We found that we could recreate these signals quite accurately using five terms of a Fourier series for each joint. There were some main advantages of recreating the signal rather than just sampling it and having the Nao move through the sampled angles, namely, the amount of data needed was a lot less, with 5 parameters per joint and  $n$  joints, we only needed  $5n$  data points to make Nao walk, compared to the thousands of sampled angles that would be needed otherwise. As well as that, the recreated signal was a lot smoother than the sampled signal, being a function of time and not just sampled data. By updating the commanded joint angles for the robot every 20ms, Nao went through the periodic part of the walking movement smoothly and accurately. While this technique seemed the way forward for what we were

trying to achieve, a different method was needed to make the robot move through the transient parts of the signal at the start and finish of the walk.

Cubic spline interpolation, with 30 splines making up the signal, gave a very good estimation of the transient signals at the start and end of the walk. Using the combination of cubic splines and Fourier series, we had recreated the Aldebaran walk quite well. Some tweaking was needed for use on different robots but overall, this method gave us more control over what the robot was doing during the walk.

As mentioned previously, the aim was to have Nao able to change direction dynamically. We sampled the body joints transcribed by the robot during the Aldebaran function walkArc, in which the robot walks in an arc with a given radius. Looking at graphs of the body angles during this function and a graph of the walk straight body angles, we noticed that they were quite similar and once or twice in each period there was a short time where the body angles of the two were equal (or almost equal), with equal derivatives.

In this short time there was the possibility to switch from walking straight to walking in an arc without introducing any kind of jerk or disturbance. We took two arc walks of different radii (20cm, 50cm, each modelled by a fourier series), in each direction and had the robot switch, at the point where the angles were equal, between the different walks. Using this method the robot was able to change direction without stopping.

The carpet on the pitches for the RoboCup were very different to our own, with a lot less friction and far less depth. This meant that the walk we had configured for our robotics lab didn't work at the venue. This meant finding an Aldebaran walk that was stable on the pitches we would be playing on and going through the above process again. The main problem we faced was getting this newly sampled walk to work on each of the robots individually, we had achieved this in our lab but were unable to manage it during the RoboCup.

For example, the robot we first tested and sampled on worked well, but the next robot didn't lift it's feet high enough and would trip over the ground. To allow for this we had to increase the ankle roll amplitude to induce more of a side to side rocking motion, although this seemed a simple problem to fix, it introduced discontinuities between the change over from the start of the walk to the periodic part. This in turn introduced a twist at the start of the walk, the timing would be slightly off and on the first step the foot hit the ground, causing it to turn away from the ball, or whatever target it was aiming for. For these reasons, we used the Aldebaran walk in all three of our matches.

## 2.4 Kicks

Our kick engine was designed around carefully designed poses for backswing, mid strike, foot lift and recovery. In all cases, key aspects of torso, arm and leg movements were considered in the design. The basic design consisted of a small number of main kick designs: (i) Ball in front of right foot; kick straight ahead with right foot; (ii) Ball in front of left foot, kick with right foot, to the left. (iii) Ball to the right of the right foot, kick straight ahead.

In addition, by interpolating between different kicks, we could, within a range of placement of the ball, independently specify direction of the kick. Tests, in the laboratory, at RoboCup in Graz (2009), and also at the German open, showed that we could kick more than the length of the field.

## 2.5 Localisation and World Modelling

Our localization was based in 2009 on a single model unscented Kalman filter. This filter shows great promise of improved performance, particularly during initialisation (that is after initial start up, or after a robot is penalised), compared to the extended Kalman filter, with similar, though slightly higher, CPU requirements. In the future we will include multiple model versions of this filter.

Related past publications from the team include [4], [5].

## 2.6 Behaviour and Team Play

Behaviour and Team play were not the focus of our 2009 effort. We had done tests on some potential fields based strategies, together with simple logic based rules and state machines for lower level behaviours.

## 3 Recent Team Publications

### References

1. S. Nicklin, R. Fisher, and R. Middleton, "Rolling shutter image compensation," in *Proc. RoboCup Symposium*, June 2006. Bremen.
2. N. Henderson, R. Middleton, and R. King, "An application of gaussian mixtures: Colour segmenting for the four legged league using hsi colour space," in *Proc. RoboCup Symposium*, July 2007. Atlanta.
3. P. Corcoran, A. Winstanley, and R. Middleton, "Lidar object segmentation using background-subtraction for vehicle collision avoidance," in *Proc. China Ireland Int. Conf. on Information and Communications Technologies*, August 2009. Maynooth.
4. R. Middleton, M. Freeston, and L. McNeill, "An application of the extended kalman filter to robot soccer localisation and world modelling," in *Proc. IFAC Symposium on Mechatronic Systems*, September 2004. Sydney.
5. M. Quinlan and R. Middleton, "Multiple model kalman filters: A localization technique for robocup soccer," in *Proc. RoboCup Symposium*, July 2009. Graz.