Stable encoding of robot paths using normalised radial basis networks: application to an autonomous wheelchair

Guido Bugmann*, Paul Robinson and Kheng L. Koay

School of Computing,
Communications and Electronics,
University of Plymouth,
Plymouth PL4 8AA, UK
*Corresponding author
E-mail: gbugmann@plymouth.ac.uk
E-mail: paul.robinson@plymouth.ac.uk
E-mail: K.L.Koay@herts.ac.uk

Abstract: A Neural Network (NN) using Normalised Radial Basis Functions (NRBF) is used for encoding the sequence of positions forming the path of an autonomous wheelchair. The network operates by continuously producing the next position for the wheelchair. As the path passes several times over the same point, additional phase information is added to the position information. This avoids the aliasing problem. The use of normalised RBFs creates an attraction field over the whole space and enables the wheelchair to recover from perturbations, e.g. due to the avoidance of people.

Keywords: path following; path representation; RBF neural network; robust control; sequence learning.

Reference to this paper should be made as follows: Bugmann, G., Robinson, P. and Koay, K.L. (2006) 'Stable encoding of robot paths using normalised radial basis networks: application to an autonomous wheelchair', *Int. J. Vehicle Autonomous Systems*, Vol. 4, Nos. 2-4, pp.239–249.

Biographical notes: Guido Bugmann is a Reader in the University of Plymouth's School of Computing, Communications and Electronics, where he develops human–robot dialogue systems, vision-based navigation systems for robots and investigates computational properties of biological vision and planning. He previously worked at the Swiss Federal Institute of Technology in Lausanne, NEC's Fundamental Research Laboratories in Japan and King's College, London. He has three patents and more than 100 publications. He studied physics at the University of Geneva and received his PhD in Physics at the Swiss Federal Institute of Technology in Lausanne. He is a Member of the Swiss Physical Society, the Neuroscience Society, the British Machine Vision Association, AISB and the Board of EURON.

Paul Robinson has researched intelligent autonomous systems for over 10 years. This work has mainly been applied to Federation of International Robot-soccer Associations (FIRA) robot football systems. Advances in relevant technologies from microrobotics to machine vision and AI has led from the three-a-side teams of the mid-1990s to the present 11-a-side international teams. He is a leading UK proponent of robot football as a platform for

developing and demonstrating intelligent autonomous systems in a well-defined environment. He and his team have designed several generations of MiroSot robots, the most recent of which were recently on display at the 2006 FIRA UK Robot Football Championships held at the Headquarters of the Institution of Mechanical Engineers in London. His stated aim is for the UK to become a major international player in the world of autonomous robot football.

Kheng Lee Koay is a Postdoctoral Researcher at the Adaptive Systems Research Group at the University of Hertfordshire, UK. He received his BSc degree in Robotics and Automated Systems and PhD degree in Intelligent Vision-based Navigation System from the University of Plymouth, UK in 1997 and 2003, respectively. His research interests include mobile robotics and human–robot interaction, especially in the aspect of human-centred socially acceptable interaction and evaluation methodology.

1 Introduction

This paper describes a part of the control system of an autonomous wheelchair that was exhibited in the South London Gallery for a month in 1997. For 7 hours per day, the wheelchair had to perform a repeated sequence of circles, spirals and figures of eight in an unmarked 7m × 7m area. The public was allowed to enter the area and the wheelchair used sonar for obstacle detection. An obstacle caused the wheelchair to stop. If the 'obstacle' did not move after a few seconds, the wheelchair initiated an avoidance manoeuvre that caused it to leave the desired path. Our problem was to design a control system that encodes the complex path and enables the wheelchair to recover from path disturbances, e.g. due to obstacles. This latter stability property would also enable the system to be insensitive to variations of the starting point, when restarted in the morning.

In Section 2 the use of a control approach based on a map in Cartesian coordinates rather than Perception-to-Action (PtA) principles is justified. In Section 3, the basics of the Normalised Radial Basis Function (NRBF) network are given. This is sufficiently different from the standard RBF network to warrant a short description. In Section 4, the encoding of the path is described in details, including the method for encoding phase information. In Section 5 general properties of the system are discussed, such as the creation of an attraction field and learning capabilities. These properties are demonstrated through experiments described in Section 6. The conclusion follows in Section 7.

2 Control philosophy

The control system was designed as a three-stage process. In the first stage, the position of the wheelchair within the gallery is determined. This was to be done by using a combination of sensors: sonar, vision, shaft encoders and gyroscope. In the second stage, a Neural Network (NN) uses the current position information to determine the next way-point along the path to be followed. In the third stage, a standard controller determines the speed of each wheel to move the wheelchair towards that way-point. Only the second stage is described in detail in this paper.

On the implementation side, a 133 MHz laptop PC was mounted on the back of the wheelchair (Figure 1) and communicated serially with a Handy-Board controller board (Martin, 2001) whose low-power H-Bridge outputs were used to generate potentials mimicking the operation of the control joystick of the wheelchair. This reduced the cost of the control system by making use of the existing high-power motor drive components of the wheelchair. The price to pay, however, was a more-complex control algorithm, as the wheelchair controller was low-pass filtered for safety reasons. The handy board performed self-localisation using direct shaft-encoder and gyroscope input and indirect sonar and vision information received from the laptop. The laptop managed sonar sensing and vision and ran the NN simulation environment, CORTEX-PRO, to generate new way-points. For practical reasons[1], the number of sensors was reduced in the final implementation to sonar and shaft-encoders.

Figure 1 Wheelchair controlled by a laptop Pentium PC 133 MHz running the neural network simulation software, CORTEX-PRO. Sonar sensors in small white boxes are used to avoid collisions and for self-localisation

The NN developed for this application continuously gives a new target position before the old one is reached and 'pulls' the wheelchair along the path. Another approach, based on encoding PtA sequences was considered but not deemed suitable for this problem. In the PtA approach, visual images from the environment or given sets of sensor readings are associated with given actions, e.g. 'when this pattern is seen from this angle, turn left'. This could not be used for the following reasons: first, with people moving around, the gallery could not provide a reproducible sensory signature of a position. We thought of using a camera directed towards the ceiling but the ceiling did not have sufficiently distinctive patterns. Second, PtA sequences are not stable against the deviations of the path. If the wheelchair finds itself in an untrained position off the path, no adequate control action is produced. Third, there was a potential aliasing problem that the memoryless PtA systems cannot handle. In particular, the path requested by the artist, Donald Rodney, repeatedly passes over the same point, with the wheelchair facing in the same direction each time (Figure 3). Therefore, it would have been impossible to define a sensory signature that distinguishes each pass.

With the system proposed in this paper, only the desired path needs to be encoded; however, adequate control actions are produced over the whole space due to the attraction field generated by the NN, as described in Section 5.

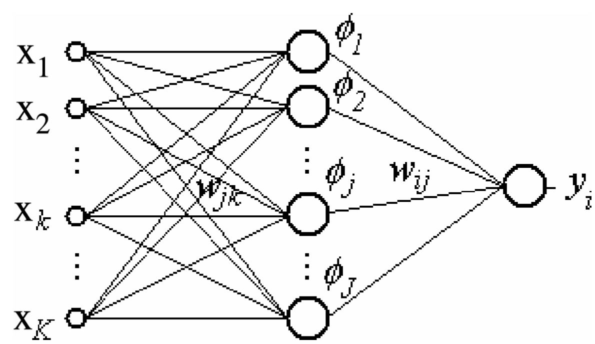
3 Normalised radial basis functions nets

NRBF nets are functionally very different from RBF nets. A NRBF net (Figure 2) comprises a hidden layer of RBF nodes and an output layer with normalising linear nodes [standard RBF nets do not have the normalising denominator in Equation (1)] (Broomhead and Lowe, 1988; Brown and Harris, 1994). The function of NRBF nets is given by:

$$y_i(\mathbf{x}) = \frac{\sum_j W_{ij}\phi(\mathbf{x}-\mathbf{x}_j)}{\sum_j \phi(\mathbf{x}-\mathbf{x}_j)} \tag{1}$$

where y_i is the activity of the output node i, $\phi(\mathbf{x}-\mathbf{x}_j)$is the activity of the hidden node j, with a RBF function centred on the vector $\mathbf{x}_j$, $\mathbf{x}$ is the actual input and w_{ij} are the weights from the RBF nodes in the hidden layer to the linear output node. Such a net is a universal function approximator in the same way as the standard RBF net (Powell, 1987), providing RBF nodes have small enough receptive fields [defined by σ in Equation (2)].

Figure 2 Radial basis function neural network architecture



The function $\phi(\mathbf{x}-\mathbf{x}_j)$ of a hidden node j is usually the Gaussian RBF:

$$\phi(\mathbf{x}-\mathbf{x}_j) = \exp\left(-\frac{\sqrt{\sum_{k=1}^{K}(x_k - w_{jk})^2}}{2\sigma^2}\right) \tag{2}$$

where σ is the width of the Gaussian and K is the dimension of the input space. The 'weights' w_{jk} between node k in the input layer and node j in the hidden layer do not act multiplicatively as in other neuron models, but define the input vector $\mathbf{x}_j = (w_{j1}, \ldots, w_{jK})$ eliciting the maximum response of node j ($\mathbf{x}_j$ *is* the 'centre of the receptive field').

As a result of the normalisation of the outputs by the total activity in the hidden layer, the output activity becomes an activity-weighted average of the input weights. The weights from the most active hidden-layer nodes contribute most to the value of the output activity. For instance, in the extreme case where only one of the hidden-layer nodes is active, then the output of the net becomes equal to the *weight* corresponding to that hidden node, whatever its actual activity. Thus RBF nodes in the hidden layer are used here as case indicators rather than as basis functions proper.

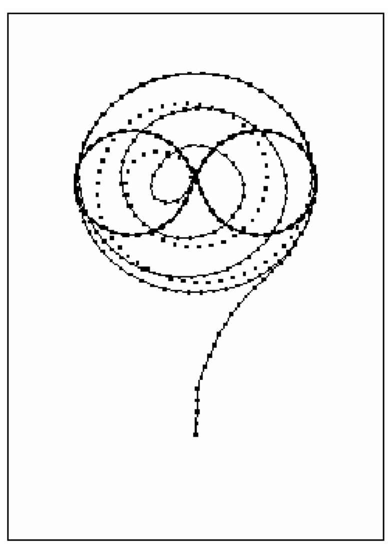
A similar normalisation principle is used in the 'centre of gravity defuzzification method (Brown and Harris, 1994). Our approach is a special case of the approach proposed by Shao, Kee and Jones (1993) for selecting linear functions $L_{ij}(\mathbf{x})$ (instead of the constant weights W_{ij} used here). In (Rao and Fuentes (1996), Equation (1) was used to compute the normalised motor output vectors in robots. NRBF nets are also useful in pattern classification applications (Bugmann, 1998).

A net with the Equation (1) was originally proposed for sequence encoding in the case of robot arm paths (Althoefer and Bugmann, 1995). This architecture is extended in the next section with a phase-encoding feature that enables encoding of the complex path of the wheelchair, which passes repeatedly in the same point in space at different phases of the sequence.

4 Path encoding

The demanded path for the wheelchair comprises two large circles along the periphery of a 7m × 7m, followed by an inward spiral. Once in the centre of the square, three successive figures of eight are performed, followed by an outwards spiral. After that the sequence restarts with two circles (Figure 3).

Figure 3 Path encoded by the neural network. The rectangle indicates the walls of the gallery. The figure is produced by simulating the motion of a vehicle starting in the lower half of the image. The outward spiral is indicated by dots only. Examples of real paths are shown in Figure 7



4.1 A sequence of half-circles

The demanded path was divided into 25 half-circles (four for the large circles, five for the inward spiral, 3×4 for the figures of eight and four for the outward spiral). Each half-circle was represented by 4 to 12 RBF nodes, depending on the diameter. The receptive field centres of the nodes were equidistantly distributed along the half-circle. Their three output weights represented the position (*x, y*) and orientation φ of the wheelchair at the next position, i.e. the next node, in the half-circle. These values are given as inputs to a standard control system, which issues motor commands. Figure 4. shows an example of four half-circles characterising one figure of eight. Figure 5 shows the components of the NRBF NN encoding the figure.

Figure 4 Definition of a figure of eight by four half-circles

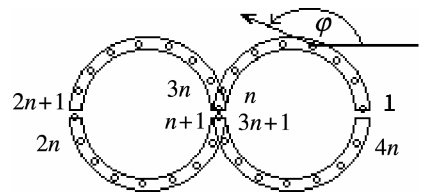
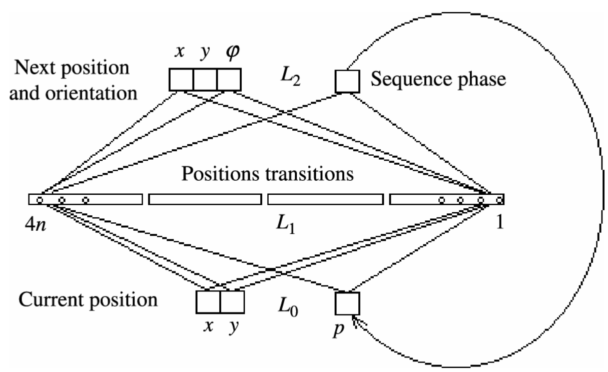


Figure 5 Neural network encoding the demanded path. The width of the receptive fields for the positions was set to one fifth of the radius of the half-circle. For the phases, the width receptive field was set to 1. L_0: input layer, L_1: hidden layer, L_2: output layer.



4.2 Off-line learning

Learning the desired path is done through a one-pass learning procedure, by setting the input weights of each hidden node to the position (*x*, *y*) of one way-point [Equation (4)] and its output weights to the position of the next way-point in the path [Equation (5)].

$$w_{j,1} = x_{1,n}\,;\ w_{j,2} = x_{2,n}\,;\ w_{j,3} = x_{3,n} \tag{4}$$

$$w_{1,j} = x_{1,n+1}\,;\ w_{2,j} = x_{2,n+1}\,;\ w_{3,j} = x_{3,n+1}\,;\ w_{4,j} = x_{4,n+1} \tag{5}$$

where $x_{1,n}$ and $x_{2,n}$ are the x and y Cartesian coordinates of way-point n, respectively; $x_{3,n}$ is the phase input and $x_{1,n+1}$ and $x_{2,n+2}$ are the x and y Cartesian coordinates of the next way-point ($n + 1$), respectively; $x_{3,n+1}$ is the phase output of the next phase and $x_{4,n+1}$ is the expected orientation of the wheelchair at way-point $n + 1$. This is a very fast training procedure. The number of hidden nodes recruited in the network equals the number of way-points along the path. An additional output node is used to encode the orientation of the wheelchair at the next way-point. This parameter is used in the low-level control algorithm. The use of the phase node is explained in Section 4.3.

This network operates by ‘pulling’ the vehicle towards the next way-point as soon as it enters the area under the control of the hidden node tuned to the nearest way-point. The control areas of hidden nodes correspond to a Voronoi tessellation of the space (Bugmann, 1998).

4.3 Avoiding aliasing by phase encoding

It can be seen in Figure 4 that several half-circles have nodes centred on the same position. To make sure that only one of them becomes active at a time, a ‘sequence phase’ node was added to the network used in (Althoefer and Bugmann, 1995; Figure 5). The weights from each of the nodes in layer L_1 to the phase node are equal to their position in the sequence (or ‘phase’). For instance, if the first node in the sequence is active, the sequence phase node will have an output 1. If the tenth is active, the output will be 10, etc. The output of that node is used as input by the ‘position transition’ nodes in layer L_1. Their input weights for the phase are set to their phase – 0.5. e.g. the tenth node has a receptive field (for phases) centred on 9.5. In that way, nodes start to become activated when the system is in the phase prior to their own (or in their own) and when the wheelchair is in the position defined by the two weights from the ‘current position’ in layer L_0. Therefore, when a position corresponds to many nodes, only the one receptive to the current phase becomes activated and can indicate the next position in the path. A special routine was written to reset the phase at the end of the sequence, to enable a repeat of the path during the time when the wheelchair was exhibited.

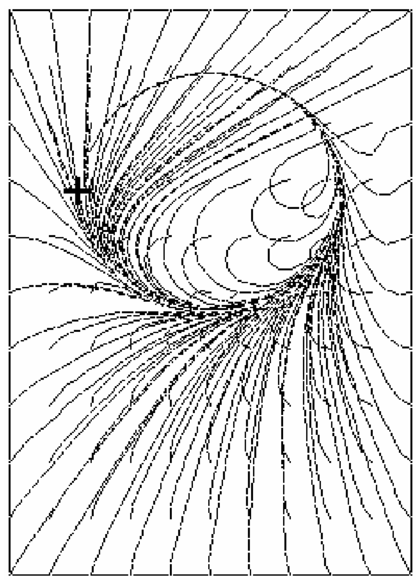
5 Properties

5.1 Attraction field

RBF nodes with a Gaussian function produce a response over the whole input space (x, y, p). This response is very weak for most combinations of position (x, y) and phase p. For instance, when the wheelchair is far from the path, only a very weak response is elicited in any of the nodes in layer L_1 of the net. However, the weakness of the responses is compensated for by the normalisation in Equation (1), enabling the network to output a value for the next position, as encoded in the weights to the layer ‘next position’. Thus normalisation results in an attraction field that leads the wheelchair towards the demanded path from whatever starting point (Figure 6).

The smooth approach-curves in Figure 6 are due to the internal dynamics of the network. Let us assume a starting point as in Figure 3. Initially, the phase p is set to 0.5, so that mainly the first node is activated. This node is centred on the position indicated by a cross in Figure 6 and is part of a descending half-circle. Thus the first waypoint indicated by the network is one node ahead of the first node. However, being active, the first node causes the phase to become $p = 1$. This in turn enables the second node to become active, which generates a way-point one node ahead of the second node. Thus the wheelchair is given a changing goal as it approaches the path. Interestingly, this movement of the way-point also occurs when the wheelchair is on the path and it needs to be controlled to avoid way-points too far ahead of the actual position. This control involves either a lower frequency of updating of the sequence phase node, or a balancing of the role of position and phase in the activation of nodes in layer L_1, as explained below.

Figure 6 Illustration of the attraction field generated by the neural network. The figure shows simulated paths with various starting positions. The initial phase is set to 0.5, so that only the first node can initially be active and thus determine the target of the motion. Its activity, however, sets the phase to 1, which enables the succeeding node to become active and so on. This causes a progressive curvature of the simulated path towards nearer nodes on the demanded path. Only the initial steps of the path are shown. The cross in the upper-left quadrant marks the position of the receptive field of the first node



It can be seen from Equation (2) that the activity of any node in layer $L1$ of the net is the product of three one-dimensional Gaussian functions centred on their preferred x, y and p, respectively. Let us assume that these Gaussians have different widths. If the width for p is large (low selectivity), the winning (most active) node is determined by the position of the wheelchair. However, if the width for p is small (high selectivity), the value of p becomes most important in determining the activity of the node. In this case, the net can run through the sequence irrespective of the position of the wheelchair. We have found that a good balance between the role of position and phase is obtained when the width for the phase is 1. In practice, selecting a different width σ for each input requires multi-variate RBF nodes, but that poses no special problems.

5.2 Learning

In this application, the path was defined in advance and the weights of the network were set accordingly. However, the path can also be learnt on the spot. In this case, a user pushes the wheelchair through the desired path, while the NN progressively recruits new nodes in layer L_1. Due to the attraction field, only the desired path needs to be learnt. The wheelchair can then enter into the path from any starting point and recover successfully from deviations. This feature is also very useful when a robot has to follow a continuously re-updated sequence of way-points generated by a receding horizon control strategy (Koay, 2003).

5.3 Aliasing

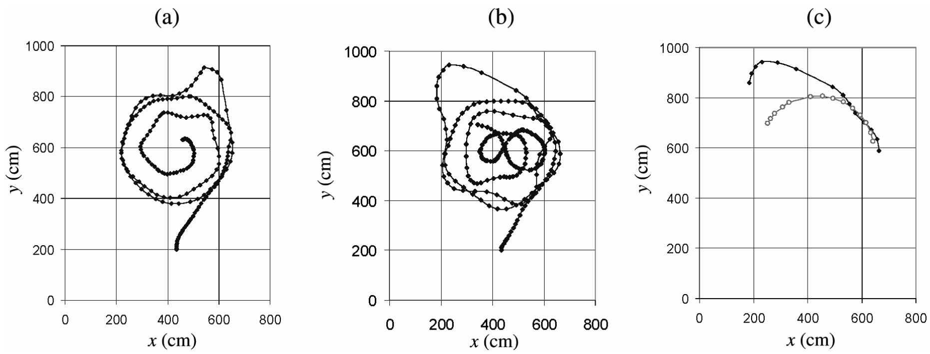
The (x, y) position of the vehicle was used as input to the sequence encoding network. In this case aliasing occurs when the same, or similar, position reoccurs at different times in the path. By adding a phase node we have avoided the situation where the vehicle jumps from one phase of the path to another, hence solving the aliasing problem.

In PtA systems, aliasing is a known problem (Rao and Fuentes, 1996). In these systems, some position-specific complex sensory picture is used instead of the (x, y) position. It should also be possible to avoid aliasing in PtA systems by adding phase information to the sensory picture.

6 Experiments

The aim of the following experiments was to generate data showing that the proposed encoding method was stable. A number of real world path-following tasks, based upon the patterns of Figure 3, were initiated from the same starting point in a 8 × 12 m room. In the case of the second circle of the complex path in Figure 3, the wheelchair was manually driven about 3 m off course but successfully recovered from this large perturbation. The initial parts of two of these paths are shown in Figure 7a and b. Figure 7c shows details of the way-points generated by the NN. It can be seen that the waypoints are placed along the desired path, bringing the wheelchair back on track. The control of the physical motion of the wheelchair is complicated by low-pass filters built in by the manufacturer for safety and comfort. Thus perfect path-following is not achieved here. However, the principle of stable path encoding is clearly demonstrated. In addition, Figure 7a and 7b show the wheelchair passing repeatedly over the same points in different phases of the path. This illustrates the effectiveness of the proposed phase encoding scheme.

Figure 7 Graphs a and b show paths followed by the wheelchair under the control of the guidance of the artificial neural network described in previous sections. During the second circle, the wheelchair was moved off course through manual override. Data show the wheelchair returning on the desired path. Graph c shows a detail of graph b (filled diamonds), in addition to the corresponding way-points generated by the neural network (open circles). The wheelchair moved with a speed of upto 0.5 m s^{-1}. It moved slower in tight curves. Odometric position data were recorded approximately every 2 s by the on-board computer



7 Conclusion

An NN architecture has been described that encodes paths in a stable way. The use of NRBF nets allows for the generation of on-path way-points for any position of the vehicle. This enables recovery from disturbances as well as initial entry into the path. The NN architecture also implements a new phase encoding principle that solves the aliasing problem, thereby making complex path following possible. The combination of position-based and phase-based way-point encoding schemes requires an appropriate balancing of the weight given to each cue. Experiments have demonstrated the successful implementation of the proposed method in a 2D case. Its application to higher-dimension problems such as non-holonomic vehicles or an industrial manipulator would not entail significantly higher computational costs.

Acknowledgements

The authors are especially grateful to Donald Rodney who commissioned this work. They gratefully acknowledge support in many forms by Mike Phillips, Nigel Barlow, British Aerospace Systems and Equipment (Plymouth), Mike Denham, Peter Frere (Lucas Advanced Engineering Centre, Birmingham), Steve Hill, the Henry Moore Foundation, David Keating, Peter Nurse, Penny and Giles Drives Technology Ltd, Plymouth Disability Equipment Centre, Alan Simpson, Dave Stevens, the South London Gallery and others mentioned in the project's web page linked from: http://www.plymouth.ac.uk/staff/gbugmann

References

Althoefer, K. and Bugmann, G. (1995) 'Planning and learning goal-directed sequences of robot-arm movements', In F. Fogelman-Soulié and P. Gallinari (Eds), Paper presented at the *International Conference on Artificial Neural Networks (ICANN'95),* Paris, Vol. 1, pp.449–454. In proceedings.

Broomhead, D.S. and Lowe, D. (1988) 'Multivariable functional interpolation and adaptive networks', *Complex Systems*, Vol. 2, pp.321–355.

Brown, M. and Harris, C. (1994) *Neurofuzzy Adaptive Modelling and Control*. Hemel Hempstead, UK: Prentice-Hall.

Bugmann, G. (1998) 'Normalized radial basis function networks', *Neurocomputing (Special Issue on Radial Basis Function Networks)*, Vol. 20, pp.97–110.

Koay, K.L. (2003) 'Intelligent vision-based navigation system', Thesis, University of Plymouth, UK.

Martin, F.G. (2001) *Robotic Explorations: A Hands-on Introduction to Engineering*. Englewood Cliffs, NJ: Prentice-Hall.

Powell, M.J.D. (1987) 'Radial basis functions for multivariate interpolation: a review', In J.C. Mason and M.G. Cox (Eds), *Algorithms for Approximation* (pp.143–167). Oxford: Clarendon Press.

Rao, R.P.N. and Fuentes, O. (1996) 'Learning navigational behaviour using a predictive sparse distributed memory', Paper presented at the *From Animals to Animats: The Fourth International Conference on Simulation of Adaptive Behaviour*, MIT Press, pp.382–390. In proceedings.

Shao, J., Kee, Y.V. and Jones, R. (1993) 'Orthogonal projection method for fast on-line learning algorithm of radial basis function neural networks', Paper presented at the *INNS World Congress on Neural Networks,* Portland, OR, USA, Vol. 3, pp.520–535. In proceedings.

Note

[1]The illumination scheme in the act gallery blinded the camera and the differential gyroscope required too complex integration methods to prevent excessive drift.