Information System for Autonomous Mobile Robot Interaction
Position Paper

Nadina Battagliotti, Daniela López De Luise and Jin Sung Park
AIGrouo, Palermo University, Mario Bravo 1050, Buenos Aires, Argentine

Keywords: Mobile Robots, Distributed Information Systems, Distributed Self-organizing Maps, Computational Intelligence.

Abstract: This paper presents what the authors consider to be a new distributed memory architecture for self-organizing distributed systems. The work in progress presented here focuses on the representation of a distributed memory for multi-robot systems. It states the main characteristics of a shared environment and provides suitable interfaces to allow autonomous mobile robots to query, publish and exchange relevant information without requiring a central data storage. The memory is based on a combination of fuzzy systems, distributed Self-Organizing Maps (SOM), a new specific design information handshake between robots and a new model for approach for world-map global administration.

1 INTRODUCTION

This paper describes a proposal to solve the problem of allocation of different types of services to a set of heterogeneuos autonomous nobile robots. Multi-robot solutions have been gaining popularity with the development of cooperative strategies and enhanced communication technologies but also due to the new advances towards the internet of the future. Multi-robot systems offer various advantages against single-robot solutions because of the distributed character that allows for parallel processing and robustness thanks to the redundancy. Lot of work has been done in cooperative robot systems, like in localization, formation control, cooperative transport and sensing, etc. (Boella G., 2003) (Findler, N.V., 1995) (Findler, N.V., 2005) (Jennings, N., 1995)(Maza, I., 2005)(Hachour, O., 2008)(Dillmann, R., 1995)

One important topic when dealing with multi-robot systems is the knowledge representation, which can be global (collective awareness of a whole robot team) or local (awareness of each individual member). In centralized systems, the implementation of a global memory can require lots of resources. Besides, centralized systems are prone to total failure if the central processing unit fails. Distributed systems on the other hand are more robust due to the redundant character of those architectures allowing an individual unit to get the role of another unit that has broken down. Nevertheless, a fully distributed system presents a high complexity and is more difficult to manage than a centralized one. This brings along the need for an advanced memory representation that combines the advantages of both approaches distributed and centralized. The implementation of such memory requires not only the definition of software architecture but also the specification of hardware requirements regarding communication between the individual entities in the whole system. The systems knowledge should also be shared with human users that can interact with the system via spontaneous connection using a smart phone. For scalability, an information exchange between systems should also be made possible. The field of Computational Intelligence offers various concepts and tools to develop a suitable approach to the presented situation. Following is an overview of the current Soft Computing alternatives in the field.

Neural Networks are used for a variety of applications from speech processing, vision, optimization, communications, classification activities, control systems, associative memories, etc (Simencio, E., 1992). In all the cases the network is made of a set of processing units (neurons) connected to others via links. The links have different strengths according to the knowledge acquired by the neurons. Every neuron has a vector of links usually named as weight vector. Most of Neural Networks require a training set to internalize the patterns of the problem. During the learning process, the desired outputs are compared to the cur-

Battagliotti N., López De Luise D. and Park J..
Information System for Autonomous Mobile Robot Interaction - Position Paper.
DOI: 10.5220/0004222403760382
In *Proceedings of the 5th International Conference on Agents and Artificial Intelligence* (ICAART-2013), pages 376-382
ISBN: 978-989-8565-38-9

rent outputs. If there is any difference between them, the internal weights are iteratively changed according to certain learning algorithms. The process stops when a convergence criterion is met.

The number of neurons, its disposition into layers, the way they are interconnected, the approach used to update weights and the mechanism by which a neuron forwards its knowledge to another neurons conform the architecture of the network. The architecture determines its velocity and precision (Wilson, W., 1993).

But in certain architectures it is possible to let the network learn in a competitive fashion. Competitive learning is a rule based on the idea that only one neuron from a given iteration in a given layer will fire at a time. These types of networks are known as self-organizing networks. In this case the current input is compared to the weight vectors of the neurons and the closest vectors determine the winning neuron (Kohonen, T., 1991).

The Kohonen's Self-Organizing Maps (SOM) is probably the most relevant competitive network architecture. It has been used successfully in many problems such as automatic speech recognition (Behme, H., 1993), clinical voice analysis (Godino-Llorente, J., 2000) (Hiltunen, T., 1993), etc. Specifically in the robotic field it is intensively used in many problems like robot navigation (Hu, H., 2000), cloud classification from satellite images (Kilpatrick, D., 1995), kinematic of a robot arm (Kieffer, S., 1991), adaptive controller for autonomous mobile robots (Kim, Y., 1992), robot motion planning in dynamic environments (Knobbe, A., 1995), spatial understanding and temporal correlation for mobile robot (Krishna, K., 2000), etc. This project uses an extension of the SOM named Distributed Self-Organizing Map (DSOM) where NN is trained using decentralized learning. There are many DSOM implementation approaches(Pascual-Marqui, A., 2001) (Lobo, V., 1998) and applications in this field (**?**). In (Lobo, V., 1998) it is applied with the sonar noise to detect the shape of a navy. In the context of this project, it is also used for obstacle recognition. But the information comes from different types of sensors that feeds asynchronously even for the same object. This requires a redesign of parts of the original DSOM model to let it aggregate patterns from partial, noisy and incomplete data of the obstacle.

Regarding the world map, the area is segmented into portions. Boundaries between those segments are not sharp but can be fuzzy. Fuzzy logic (FL) has been successfully used to solve many complex problems. It is inherently robust, can be modified and tweaked easily to improve the system performance and can control nonlinear systems (Cox, E., 1992) (Lee, C., 1990). In general, FL is used in many applications where noise, error and missing data is typical, mainly in control theory and artificial intelligence. Among other applications are wireless communications (Erman, M., 2009), velocity induction for a motor (Kumar, V., 2005), operational meteorology (Murtha, J., 1995), data mining and Information Retrieval(Meunier, B., 2007), etc. This project proposes a new FL usage in the world map segment administration. Although it has been used in philosophy (Kosko, B., 1993) and psychology (Didelon, C., 1991) for evaluating the individual usage and characteristics of its cognitive maps (or interpretative maps), the approach to manage segments of a robots world-map is a contribution in the field. In the context of this work, FL is important to guarantee the proper control of the robot during the migration from one segment to another in the map. If the services provided by the robot or its abilities are critical, the fuzzy boundaries allow the system to take a period of time to adapt the current activity softly.

This article is organized as follows: Section 2 describes the proposed architecture for a distributed memory. Later in 3, the proposed information approach is presented. Section 4 describes the software that is used in this work. Conclusions and future work are presented in 5.

2 PROPOSED ARCHITECTURE

2.1 Global Description

Fig. 1 shows the proposed global system architecture. The physical workspace covered by the system is logically split into segments (S_i). Every partition has a **L**ogical **C**entral **P**oint (LCP) that regulates the activity of the agents within S. It can also handle a local world map and define its boundaries. A team of autonomous robots is constantly providing services within each segment. Any robot is able to perform a set of different services. To do so, it can download from the LCP the service that it should provide and the corresponding hardware configuration required to perform these services. It can also upload its sensory data to its corresponding LCP.

2.2 Logical Central Point

A Logical Central Point (LCP) is a logical role that behaves as the manager of a segment S. A LCP can be any data processing unit, for example a desktop computer of a processor mounted on a robot. Custom

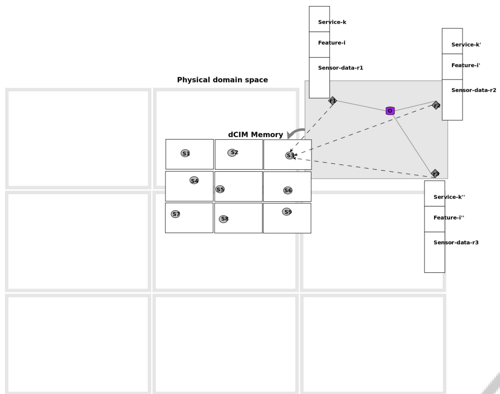


Figure 1: Representation of the proposed global architecture.

services are physically located and associated to one or more LCP. The activity of any LCP can be:

- Manage shared information with other LCPs using a specific LCP-LCP handshake.
- Manage information of the robots in its segment S using a specific LCP-robot handshake.
- Work as a petition center.
- Work as a passive receiver of service requests from another LCP.
- Manage local storage block to be able to make inferences and command fast actions to the robots in S.
- Allocate robots to service petitions.
- Request robots to other LCPs.
- Send robots to other segments upon a request from other LCP.
- Delegate its role to other unit.
- Make a robot to change its service.
- Make a robot to change its hardware configuration (sending it to a specific LCP whose segment is advocated to this tasks).
- Manage a distributed world-map updated with the information received from the robots in S.

To perform its activity, LCP has a local storage with information like:

- ID and service of each robot within its segment.
- Number of robots in its corresponding segment S_i.
- Local world map defined as a set of cells, each representing a physical area.
- The boundaries of S_i.

In this approach, every world map remains private for its corresponding LCP. Each segments boundary is defined by:

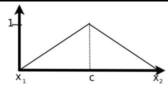
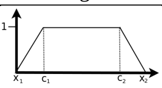
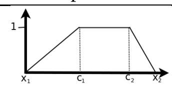
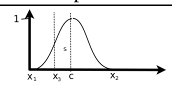
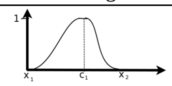
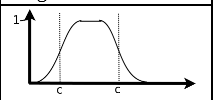
- Border section
- ID of the neighbor that shares every border section
- Border strength

The border strength is defined in terms of a fuzziness function. For example, in Fig. 1, the neighbors of S1 are:

- S2: with border S1-2 and fuzziness = 0 (i.e. no fuzziness)
- S4: with border S1-4 and fuzziness = 5 (fuzziness with member function 5)
- S5: with border S1-5 and fuzziness = 3 (fuzziness with member function 3)

In this context, the boundary is one or more borders. And a border is a set of cells of the local world map. In all cases, fuzziness refers to a number of a predefined member function. Table 1 is an example of functions that are classical in the field.

Table 1: Classical membership functions.

ID	Membership formula	Shape
0	None	None
1	$\mu(x, x_2, c) = \begin{cases} c^{-1}x & 0 \le x \le c \\ (x_2 - c)^{-1}x + 2 & c \le x \le x_2 \end{cases}$	 triangular
2	$\mu(x, x_1, x_2, c_1, c_2) = \begin{cases} \frac{x - x_1}{c_1 - x_1} & x_1 \le x \le c_1 \\ 1 & c_1 \le x \le c_2 \\ \frac{x_2 - x}{x_2 - c_2} & c_2 \le x \le x_2 \end{cases}$	 trapezium
3	$\mu(x, x_1, x_2, c_1, c_2) = \begin{cases} \frac{x - x_1}{c_1 - x_1} & x_1 \le x \le c_1 \\ 1 & c_1 \le x \le c_2 \\ \frac{x_2 - x}{x_2 - c_2} & c_2 \le x \le x_2 \end{cases}$	 trapezoid
4	$\mu(x, c, s) = \frac{e^{-(x-c)^2}}{2s^2}$	 normalized gaussian
5	$\mu(x, \alpha, \beta, c) = \left[1 + \left\lvert \frac{x - c}{\alpha} \right\rvert^{2\beta}\right]^{-1}$	 generalized bell
6	$\mu(x, \alpha_1, c_1, \alpha_2, c_2,) = \left[1 + e^{-\alpha_1 (x - c_1)^2}\right]^{-1} - \left[1 + e^{-\alpha_2 (x - c_2)^2}\right]^{-1}$	 2 sigmoidals
7	...	...

The membership function shape is very important to define the transition behavior when a robot trespass the boundaries of the current segment. Among others, it defines:

- *if the Membership Function is Symmetric*: the LCP consumes the same period of time to get ready to delegate the robot and to release the robot to the target LCP (see Fig. 2).

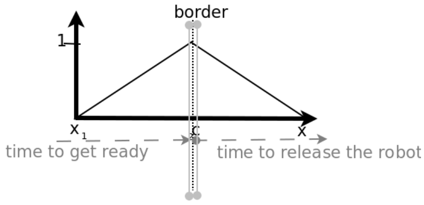


Figure 2: Fuzzy border trespassing.

- *if the Membership Function is Asymmetric*: the LCP requires different amount of time to prepare the robot delegation and to release the robot.
- *if there is no Membership Function*: the original LCP delegates the navigation and control of the robot instantly to the target LCP.

The strength of the border can also be understood not only as the ability but also as the disposition of the current LCP to allow a robot to pass-through that segment of the boundary.

2.3 World Map

A world map is a representation of the internal information of a segment. As the LCP receives information via handshaking with the robots within its assigned segment, the current information about obstacles and robots positions in the world map gets constantly updated.

By using the concept of Distributed Self Organized Maps (DSOM) the LCP is able to recognize known obstacles from asynchronous and partial information gathered from the robots in the current segment. Although by definition, the SOM is intended to classify data into known categories (e.g. types of objects), it was also demonstrated (Lobo, V., 1998) that DSOMs are able to recognize new objects.

A world map is a small part of the global workspace, because it is intended to model just the physical space covered by a given segment S_i (and managed by the corresponding LCP).

2.4 Robot Distribution

Robots are assigned groups to segments handled by LCPs. Every robot can only belong to a segment. When a LCP detects an increase in the workload (e.g. because the robots have many service requirements, or because the LCP itself has received many queries from a terminal operator), it can negotiate with the nearest LCPs to get additional robots.

Conversely, a LCP can receive from its neighbors a request for borrowing one or more robots from its segment. Then it will decide whether to accept or not those requests according to its current own workload.

We assume that each robot is able to navigate autonomously within its segment with a predefined collision-avoidance function and that a robot will never cross the boundaries defined in the LCP world-map. The LCP will monitor the states of its robots to improve their collision-avoidance algorithms and so prevent deadlocks and critical situations.

While executing a service, a robot simultaneously keeps sampling its environment so its LCP can update its world map. It also sends a fragment of this information to the closest robots in the same area.

If there is a dynamic obstacle within a segment, then one or more robots will detect its presence and forward it to its LCP and closest robots. All this information is used by the LCP to recognize the type of obstacle (by using the DSOM). The LCP can also inform to another neighbor LCP that the obstacle is about to enter to its area and provide obstacle data (i.e. velocity, shape, type and last known position).

Upon request, the LCP can reassign a robot a new service. If the robot hardware features necessary to provide the new service require human intervention (for instance to exchange a tool, remove or add a wheel, add a new hardware module, etc.), the LCP can command the robot to trespass the boundary of the current segment and pass to a special maintenance segment. After the robot has its physical configuration changed, it updates its internal status, returns to its former segment and follows the LCP indication to perform the new service. This can also imply the robot being transferred to a new segment.

3 INFORMATION APPROACH

Each robot uses a negotiation with its neighbors. During that negotiation, it shares a reduced amount of information (only with its neighbors). The approach is similar to that used in the computer networks to solve the problem of routing information from a source to target. But in this proposal the data being routed consists of a bunch of information required to acknowledge the features of the surroundings.

Below there is a comparison between the typical activity of a router and the Distributed robot Handshaking Approach (DHA) [1]. Table 2 shows the features taken from the router approach that we adapted for robot-to-robot handshaking. While within a router, data packets are exchanged, in the DHA, this occurs with sampling data from the robots sensors. Each robot in the DHA is analog to a node in a routing

[1]It is mentioned in section it is mentioned in section Future Work as number 15

scheme. The difference is that in the router scheme, the destination node can be each node in the network, while in the DHA, the destination node can only be a neighbor node. Furthermore, contrary to the router approach, the DHA considers every LCP storing update information about the environment; within the router approach, a table of routings is stored.

Table 2: Features extracted from router approach for robot-to-robot handshaking.

Feature	Router	DHA
Content	Data packets	Current sampling changes
Source	A node calling another	The robot that is sampling
Target	The destination candidates (can be more than one)	The robots that are closest
Feed-back	Navigation status of each candidate	Sampling changes of each closest robots (when keep being closer)
Activity	Select best destination using the feed-back from every candidate	Select best destination using the feed-back from every candidate
Storage	Table of routings	Current cells and surroundings cells[2]
Updating process	Upon every network configuration change[3]	Upon every physical change in the land that is being detected

Something similar is implemented between LCP and robots. The information exchanged is reduced to allow the LCP to manage a number of robots properly, as in Table 3. In this case, in the DHA, the source is given by the LCP itself. Whereas every robot inside the LCPs segment can communicate with the LCP, in the handshake, only the robots that are capable to perform the task in question are considered targets. In this handshaking, the information stored is given by the current set of tickets (queries) and services being performed by the robot.

The handshake between two LCPs is defined as in Table 4.

This handshaking is analog to that for robot-to-robot communication. In this case, only neighbor LCPs can be set as targets. The information to be exchanged here is represented by the global changes in the environment affecting more than one LCP. Additionally, information about tickets under negotiation and about robot sharing is also exchanged.

Table 3: Features extracted from router approach for robot-to-robot handshaking.

Feature	Router	DHA
Content	Data packets	Current world-map changes of interest for the robot
Source	A node calling another	The LCP that has to command a robot
Target	The destination candidates (can be more than one)	The robots that are able to perform the task
Feed-back	Navigation status of each candidate	Current status and availability
Activity	Select best destination using the feed-back from every candidate	Select the best robot to accomplish the task and afterwards monitor it
Storage	Table of routings	Current tickets and services being performed by each robot
Updating process	Upon every network configuration change	Upon every new ticket or service completion

4 SYSTEM REQUIREMENTS

4.1 Software

The software tools are mostly open, except for the simulators required to test the functionality:

- Eclipse framework
- Java 2
- svn, subversion system
- Mantis, bug tracker
- LMS simulator

5 CONCLUSIONS AND FUTURE WORK

There are many pending items for the project. Among others the following items are to be refined yet:

1. The internal language (numeric or textual) to represent service features
2. The internal language (numeric or textual) to represent hardware features
3. The internal language (numeric or textual) to represent sampling information
4. Parameters to define the segments:

Table 4: Features extracted from a router approach for handshaking between LCPs.

Feature	Router	DHA
Content	Data packets	Current global changes that affects more than one LCP. Inter LCP ticket negotiation to provide the services. Robots sharing (between two LCPs)t
Source	A node calling another	The LCP that has to fulfill a request
Target	The destination candidates (can be more than one)	The neighbor LCPs
Feed-back	Navigation status of each candidate	Current status and availability of the target LCPs
Activity	Select best destination using the feed-back from every candidate	Select the best LCP to share the task and assign it
Storage	Table of routings	Current status and services
Updating process	Upon every network configuration change	Upon events

- if the boundaries are static or may change its size and shape
- if the boundaries are fuzzy o sharp
- if the LCP defines its boundaries automatically or it needs external information to know its boundaries

5. Parameters for robot-central point handshake:
   - LCP to robot: current service, hardware configuration
   - Robot to LCP: sampling data
6. Parameters for the handshake between central points (LCP).
7. Evaluate whether to implement traditional Kohonen's SOM or Fuzzy c-Means (less computational complexity).
8. The implementation of the logical central points in one or more physical processor, and its real localization in the real world.
9. Define the approach to balance the resources distribution (resources are parts of robots, robots, knowledge, parts of its world-maps, etc.)
10. Define the concept of closest robots: whether they are all the robots within certain distance, or the directly linked conceptually. It must be defined a new metric to assess that.
11. Define if a LCP can be used as a petition centers or it just receives petitions from a central point.
12. Define if traditional SOM or fuzzy c-means has to be used.
13. Define the special segments. The hardware configuration can be changed manually or using robotics.
14. Define if there is just one special segment or many. In the later case, the workload administration approach.
15. Define the contention algorithms between robots to reduce their information storage and amount of data to be interchanged.
16. Define the contention algorithms between a robot and its LCP, to reduce their information storage and amount of data to be interchanged.
17. Define the local robot memory updating approach.
18. Define the number of cells in the world-map that corresponds to the surroundings term.
19. Define a hierarchy
20. Define roles & inter-role communication
21. Create a wiki to share and publish information about the project, put the publications, etc
22. Evaluate the set of member functions according to the service and hardware configurations.
23. Evaluate the shape and size of segments.
24. Evaluate the dynamic of changes in the boundaries of the segments.
25. Evaluate the proper set of fuzzy operators to work with the fuzzy boundaries.

REFERENCES

Behme, H. (1993). Speech processing by hierarchical segment classification. In *International Joint Conference on Neural Networks*. IEEE Press.

Boella, G.(2003). Norm governed multiagent systems: The delegation of control to autonomous agents. In *Proc. of IEEE/WIC IAT*. IEEE Press.

Cox, E. (1992). Fuzzy fundamentals. In *IEEE Spectrum*. IEEE Press.

Didelon, C., de Ruffray, S. (1991). A world of interstices: Fuzzy logic approach in the analysis of interpretative maps. In *The Cartographic Journal*. British Cartographic Society.

Dillmann, R., Kauser, M., Wallner, F., Weckesser, P. (1995). Priamos: An advanced mobile system for service, inspection, and surveillance tasks. In *Modelling and planning for sensor based intellinget robot systems*. H. Bunke, T. Kanade, H. Noltemeirer Eds.

Erman, M., Mohammed, A. (2009). Fuzzy logic applications in wireless communications. In *IFSA-EUSFLAT*. IFSA-EUSFLAT.

Findler, N.V. (1995). Multi-agent coordination and cooperation in a distributed dynamic environment with limited resources. In *Artificial Intelligence in Engineering*. Elsevier.

Findler, N.V. (2005). A robust business resource management framework based on a peer-to-peer infrastructure. In *7th IEEE Int. Conf. on E-Commerce*. Kluwer Academic.

Godino-Llorente, J. (2000). Non supervised neural net applied to the detection of voice impairment. In *IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE Press.

Hachour, O. (2008). Path planning of Autonomous Mobile robot. In *Int. Journal of Systems Appl., Eng. & Develop.*. University Press. Issue 4(2).

Hiltunen, T., Leinonen, L. (1993). Visualization and classification of voice quality with the self-organizing map. In *International Conference on Artificial Neural Networks*. S. Gielen and B. Kappen.

Hu, H. (2000). *Landmark-based navigation of industrial mobile robots*. University of Portsmouth, Portsmouth, 1st edition.

Jennings, N. (1995). Controlling cooperative problem solving in industrial multi-agent systems using joint intentions. In *Artificial Intelligence*. Elsevier.

Kieffer, S., Morellas, S. (1991). Neural network learning of the inverse kinematic relationships for a robot arm. In *International Conference on Robotics and Automation*. IEEE Computer Society Press.

Kilpatrick, D. (1995). Unsupervised classification of antarctic satellite imagery using kohonens self-organizing feature map. In *IEEE International Conference on Neural Networks*. IEEE Press.

Kim, Y. (1992). Robust adaptive control of an autonomous mobile robot. In *Second International Conference on Automation, Robotics and Computer Vision*. Nanyang Technol. Univ.

Knobbe, A., Kok, J. (1995). Robot motion planning in unknown environments using neural networks. In *International Conference on Artificial Neural Networks*. NParis : EC2 & Cie.

Kohonen, T. (1991). *The self Organizing Map*. IEEE Press, New York, 1st edition.

Kosko, B. (1993). *Fuzzy Thinking: The New Science of Fuzzy Logic*. Hyperion, New York, 1st edition.

Krishna, K. (2000). Spatial understanding and temporal correlation for a mobile robot. In *Spatial-Cognition-and-Computation*. Lawrence Erlbaum Associates Inc.

Kumar, V., Joshi, E.R. (2005). Hybrid controller based intelligent speed control of induction motor. In *Journal of Theoretical and Applied Information Technology*. Little Lion Scientific.

Lee, C. (1990). Fuzzy logic in control systems. In *IEEE Trans. on Systems, Man, and Cybernetics*. IEEE Press.

Lobo, V., Bandeira, N. (1998). Ship recognition using distributed self organizing maps. In *Neural Networks Proceedings of the International Conference EANN 98*. Systems Engineering Association.

Maza, I., Viguria, A., Ollero, A. (2005). Cooperacin helicptero autnomo-robot mvil terrestre para aplicaciones de bsqueda y localizacin. In *Proc. of I Congreso Espaol de Informtica*. pp 109-116, Thomson, Granada, Spain, 2005.

Meunier, B. (2007). Real-world fuzzy logic applications in data mining and information retrievaly. In *A Spectrum of Theoretical and Practical Issues*. Springer.

Murtha, J. (1995). Applications of fuzzy logic in operational meteorology. In *Scientific Services and Professional Development Newsletter*. Canadian Forces Weather Service.

Pascual-Marqui, A. , Pascual-Montano, K. (2001). Smoothly distributed fuzzy c-means: a new self-organing map. In *Pattern Recognition*. IEEE Press.

Simencio, E. (1992). *Artificial Neural Networks*. IEEE Press, New York, 1st edition.

Wilson, W. (1993). A comparison of architectural alternatives for recurrent networks. In *Fourth Australian Conference on Neural Networks*. Sydney University Electrical Engineering.