High Speed Neural Control for Robot Navigation

Sílvia S.C. Botelho, Eduardo do Valle Simões, Luís Felipe Uebel & Dante Barone
Instituto de Informática - Universidade Federal do Rio Grande do Sul
Caixa Postal 15064 - Porto Alegre, RS 91501-970, Brazil
E-mail: SILVIACB, EDSIM, UEBEL, BARONE@inf.ufrgs.br

ABSTRACT

This work addresses the real time control of the Khepera mobile robot [1] navigation in a maze with reflector walls. Boolean Neural Networks such as RAM [2] and GSN [3] models are applied to drive the vehicle, following a light source, while avoiding obstacles. Both neural networks are implemented with simple logic and arithmetic functions (*NOT*, *AND*, *OR*, *Addition*, and *Comparison*), aiming to improve system speed. The results obtained are compared with two other control strategies: Multi-layer Perceptron (MLP) [4] and Fuzzy Logic [5].

1. INTRODUCTION

The robot navigation into a maze [6] problem has been chosen because of the high potential for application of the developed techniques to the industrial environment, where great flexibility and dynamic configuration are necessary nowadays [7]. The solution of this problem with traditional methods leads to many difficulties like: ***i*)** real time processing; ***ii*)** sensor noise toleration; ***iii*)** environment change endurance.

The solution presented herein employs neural networks due to their strong noise toleration and great generalisation capacity [8]. A common problem of neural networks in this case can be low speed processing. Therefore, Boolean neural networks were applied to reduce execution time [9], since they use simple arithmetic and logic functions, instead of complex floating point operations. RAM based neural networks [2] are made of memory mapped neurons, grouped in distinct classes. The input pattern addresses memory positions, where network knowledge is placed. Network output comes from calculating the sum of the activated neurons in each class. GSN (*Goal Seeking Neuron*) based networks have their neurons grouped in pyramids, and differ from RAM since it allows the propagation of an undefined state ***U***.

2. THE KHEPERA'S WORKSPACE

The Khepera mobile robot [1] has an internal MC 68331-16MHz microcontroller, 256 Kbytes of RAM, an internal battery, and 8 peripheral infra-red sensors. These sensors inform the distance from obstacles and measure environment light.

The host computer can send commands through an RS232 serial line to activate two DC motors, responsible for driving its wheels. The robot can turn to any direction by applying different speeds to each wheel. The Khepera robot and its workspace are shown in Figure 1.

3. COMPORTAMENTAL RULE-BASED CONTROL

The implemented control systems are all based on a set of 14 rules, shown in Table 1, that

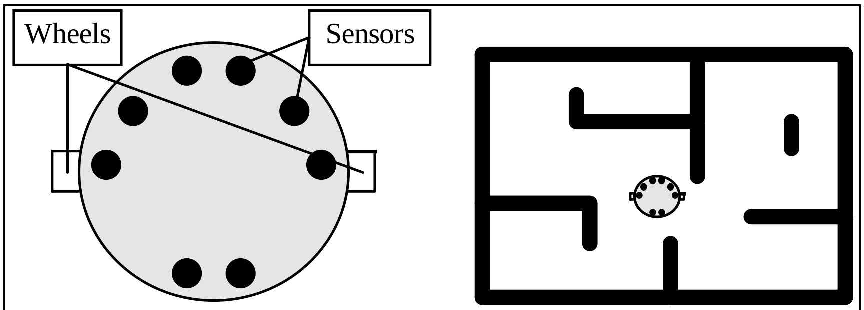

Fig. 1: Khepera Robot and Its Workspace.

Tab. 1: Set of 14 Rules used to teach all control systems (**FD** Front Distance; **LD** Left Distance; **RD** Right Distance; **LL** Left Light; **RL** Right Light; **W1** Wheel 1; **W2** Wheel 2; **BF** Back Fast; **BL** Back Low; **FF** Front Fast; **FL** Front Low).

	FD	**LD**	**RD**	**LL**	**RL**	**W1**	**W2**
1	0	0	0	1	1	BF	BF
2	1	1	1	1	1	FL	FL
3	0	0	1	1	1	BF	BL
4	0	1	0	1	1	BL	BF
5	1	0	0	1	1	FL	FL
6	1	1	1	0	1	FL	FF
7	1	1	1	1	0	FF	FL
8	1	0	1	1	0	FF	FL
9	1	1	0	0	1	FL	FF
10	0	1	0	0	1	BL	BF
11	0	0	1	1	0	BF	BL
12	0	1	1	1	0	BF	BL
13	0	1	1	0	1	BL	BF
14	0	1	1	1	1	FL	FL

describes the necessary behaviour to robot navigation. All employed systems need training patterns, provided by the specified rules, because they use a supervised learning algorithm.

Boolean Neural Network Controllers

The GSN model implementation makes use of two pyramids to control each wheel. All pyramid neurons are trained with the rules from Table 1. The robot sensor values are directly provided to the pyramid inputs, and the network outputs are applied to the wheel motors. After the training phase, each neuron is manipulated as a "*black box*". All possible combination of neuron input is analysed to convert all network into a combinational circuit. Therefore, the circuit logic minimisation is processed according to [10]. Then, a description of the minimised neuron network is generated into a *C language* code. Consequently, the network circuit can be directly executed into the microprocessor ALU (*Arithmetic Logic Unit*). This fact decreases the derives from the use of execution time of the algorithm.

The advantage of the implemented technique is the employment of the microprocessor ALU to execute the necessary logic operation set of the GSN algorithm. The simple logic functions (*NOT*, *AND*, and *OR*) are faster to execute than complex floating point operations used by a greater number of robot control models. Table 2 presents the output function of one of the robot wheels.

The RAM neural network has 4 behaviour classes with 5 neurons to control each wheel. The RAM model was implemented according to the same approach used for the GSN, with the inclusion of simple additions and comparisons, both necessary to the RAM model output evaluation.

Multi-layer Perceptron and Fuzzy Controllers

A Multi-layer Perceptron model and a Fuzzy System were built with the same described rules, aiming to evaluate the performance of

Tab. 2: GSN implementation of the control of a robot wheel.

Wheel 1	**Minimised Logic Function**
Bit0 = 0: Front **Bit0** = 1: Back	**Bit0** = RL FD + RD LL !LD FD + !RD LL LD FD
Bit1 = 1: Keep Current State	**Bit1** = LL !RL + !LD !RD LL !RL + LD RD LL !RL
Bit0 = 0: Fast **Bit0** = 1: Low	**Bit0** = !FD !LD RD LL + !FD LD !RD LL + FD !LD RD !LL RL + FD LD !LL RL + FD LD RD RL
Bit1 = 1: Keep Current State	**Bit1** = LL !RL + !LD !RD LL !RL + LD RD LL !RL

Tab. 3: Comparison Between
Four Control Algorithms.

Control	Collisions per Minute	Algorithm Delay (μs)
Fuzzy	7	1040
MLP	10	850
RAM	11.5	30
GSN	6	26.2

the two implementations. The Multi-layer Perceptron neural network has three layers with 19 neurons to control each wheel. The Fuzzy System contains 5 input variables, each one presenting 2 membership functions, and an output variable with 4 membership functions.

4. RESULTS

A group of tests was performed according to the processing time, number of collisions, and required memory. Table 3 presents the delay of processing a cycle of each control algorithm, using a 486 DX2-66 MHz computer as a host. The Boolean neural networks great simplicity and their implementation as elementary logic functions are responsible for their greater performance (about 40 times faster).

Table 3 also shows the average collision number per minute, measured during the robot navigation through different maze configurations. The RAM model has presented the largest collision number, due to the fact that it is possible to occur a draw among two or more behaviour classes, presenting the same sum. It can lead to an incorrect response, depending on the situation. The Multi-layer Perceptron model differs from RAM, because it works with analog output values to control the wheel. It reduces the deviation time and can introduce more collisions. On the other side, the GSN network has presented the best results due to its great manoeuvre speed, since it does not have the same functional problems as the RAM model. The Fuzzy System has also shown good results in terms of collision rate.

The Boolean neural networks, mainly the RAM model, tend to occupy a great portion of memory. Therefore, memory allocation can be a significant problem in some Khepera applications, limiting the control system choice. The Boolean neural network mapping into logic functions and their direct execution in the microprocessor ALU have also allowed a reduction of the memory required by the control algorithm.

5. CONCLUSIONS

The Boolean models, mainly the GSN, have shown the best performance compared to the other two. The great speed provided allows a fast control that can improve the decision rate in complex sensorial systems, like artificial vision. The Boolean neural network memory requirement was reduced by the use of simple ALU logic functions. It allows the application of Boolean neural controllers into low cost microprocessors, like the Khepera one.

The application of the presented technique is not restricted to mobile robot applications. It can be applied in mechanical arms, industrial process control, and complex pattern recognition, that demands a high processing capacity.

This work will continue with the development of an embedded real time learning system, implemented according to the same techniques, that will make use of the Boolean neural network one-shot learning to quickly map the environment where the robot is moving.

6. REFERENCES

[1] K-Team, "Khepera User Manual Version 3", Laboratoire de Microinformatique, Swiss Federal Institute of Technology, Lausanne, Switzerland, 1994, 23 p.

[2] C. E. Myers, "Delay Learning in Artificial Neural Networks, Chapman & Hall", 2-6 Boudary Row, London, 1992, 157 p.

[3] E. C. D. B. Filho, D. L. Bisset and M. C. Fairhurst, "A Goal Seeking Neural for Boolean Neural Networks", Proc. International Neural Network Conference, Paris, France, Vol. 2, July 1990, pp. 894-897.

[4] J. Freeman, "Neural Networks: Algorithms, Application and Programming Techniques", Computation and Neural System Series, USA, July 1992, 400 p.

[5] L. Zadhe, “Common-sense Knowledge Representation Based on Fuzzy Logic”, Computer, October 1983, pp. 61-65.
[6] J, Latombe, “Robot Motion Planning, Kluwer Academic Publishers”, USA, 1991, 651 p.
[7] J. A. Tanchoco, “Material Flow Systems in Manufacturing”, Chapman & Hall , West Lafayette, USA, 1994, 398 p.
[8] I. Harvey and D. Clhiff, “Issues in Evolutionary robotics”, In J. Meyer, H. L. Roitblat, and S.W. Wilson, editors, From Animals to Animals II: Proceedings of the Second International Conference on Simulation of Adaptive Behaviour, Cambridge, MA, 1992.
[9] L. F. Uebel and D. A. Barone, “DIANNE: A GSN Neural Network VLSI Implementation”, 5th International Symposium on IC Technology, Systems & Applications, Singapore, September 1993, p. 678-682.
[10] E. V. Simões, L. F. Uebel and D. A. Barone, “Fast Prototyping of Artificial Neural Network: GSN Digital Implementation”, Fourth International Conference on Microelectronics for Neural Networks and Fuzzy Systems, Turin, Italy, September 1994, pp. 192-201.
[11] E. V. Simões, L. F. Uebel and D. A. Barone, Hardware Implementation of RAM Neural Networks, Pattern Recognition Letters, April 1996.
[12] L. F. Uebel, E. V. Simões and D. A. C. Barone, “A Comparision between Three Different GSN Model Hardware Implementations with the Appliance of an ANN Fast prototyping System”, World Congress on Neural Networks, Washington, USA, July 1995, CDROM Proceedings.