

# **FIXME - the name of your project**

Final Report for CSM6960 Major Project

*Author:* Stefan Klaus (stk4@aber.ac.uk)

*Supervisor:* Dr. Myra Wilson (mxw@aber.ac.uk)

18th July 2015

Version: 0.1 (Draft)

This report was submitted as partial fulfilment of a MSc degree in  
Intelligent Systems (G496)

Department of Computer Science  
Aberystwyth University  
Aberystwyth  
Ceredigion  
SY23 3DB  
Wales, UK

## **Declaration of originality**

In signing below, I confirm that:

- This submission is my own work, except where clearly indicated.
- I understand that there are severe penalties for plagiarism and other unfair practice, which can lead to loss of marks or even the withholding of a degree.
- I have read the sections on unfair practice in the Students' Examinations Handbook and the relevant sections of the current Student Handbook of the Department of Computer Science.
- I understand and agree to abide by the University's regulations governing these issues.

Signature .....

Date .....

## **Consent to share this work**

In signing below, I hereby agree to this dissertation being made available to other students and academic staff of the Aberystwyth Computer Science Department.

Signature .....

Date .....

## **Acknowledgements**

I am grateful to...

I'd like to thank...

## **Abstract**

Include an abstract for your project. This should be no more than 300 words.

# CONTENTS

<b>1</b>	<b>Background &amp; Objectives</b>	<b>1</b>
1.1	Aim of this project . . . . .	1
1.2	Information about the project . . . . .	1
1.2.1	Choice of development environment and programming language . . . . .	1
1.2.2	Choice of robot . . . . .	2
1.2.3	Choice of environment design . . . . .	2
1.3	Analysis . . . . .	2
1.3.1	Communications . . . . .	2
1.3.2	Control Algorithm . . . . .	2
1.4	Process . . . . .	4
<b>2</b>	<b>Design</b>	<b>5</b>
2.1	Overall Architecture . . . . .	5
2.1.1	Control Algorithm . . . . .	5
2.1.2	Artificial Neural Network . . . . .	5
2.2	Some detailed design . . . . .	6
2.2.1	Even more detail . . . . .	6
2.3	User Interface . . . . .	6
2.4	Other relevant sections . . . . .	6
<b>3</b>	<b>Implementation</b>	<b>7</b>
<b>4</b>	<b>Testing</b>	<b>8</b>
4.1	Overall Approach to Testing . . . . .	8
4.2	Automated Testing . . . . .	8
4.2.1	Unit Tests . . . . .	8
4.2.2	User Interface Testing . . . . .	8
4.2.3	Stress Testing . . . . .	8
4.2.4	Other types of testing . . . . .	8
4.3	Integration Testing . . . . .	8
4.4	User Testing . . . . .	8
<b>5</b>	<b>Evaluation</b>	<b>9</b>
	<b>Appendices</b>	<b>10</b>
<b>A</b>	<b>Third-Party Code and Libraries</b>	<b>11</b>
<b>B</b>	<b>Code samples</b>	<b>12</b>
	<b>Annotated Bibliography</b>	<b>13</b>

## LIST OF FIGURES

1.1	Roulette Wheel Selection . . . . .	4
2.1	IR sensor response against distance . . . . .	6

## **LIST OF TABLES**

# Chapter 1

## Background & Objectives

### 1.1 Aim of this project

The goal of this project is to create a program that controls a swarm of E-Puck robots in order to map an environment.

The E-Pucks always need to stay in communication, as they share a global map.

The control algorithm is an Artificial Neural Network, a evolutionary algorithm.

The neural network is trained so that it will learn to develop a solution on its own. The benefit of neural networks are that they are extremely versatile. A trained neural network will perform very good even in an different environment. Neural networks also allow for quick reactions of robots with very little processing time once they are fully evolved.

### 1.2 Information about the project

#### 1.2.1 Choice of development environment and programming language

Since the control algorithm for the robots is an artificial neural network its needs to be trained before it can be tested. Therefore a simulator is needed. The simulator chosen for this project has been created by Elio Tuci(elt7@aber.ac.uk) and Muhanad Hayder Mohammed(mhm4@aber.ac.uk) at Aberystwyth University.

The other candidate was Cyberbotics Webots<sup>1</sup>.

The simulator by Elio Tuci was chosen since I worked with it throughout the second semester of this master course and therefore know it well. Other reasons include that simulator is build for the creation of evolutionary algorithms and already posses an implemented genetic algorithm.

At the beginning of the project it was deemed ambitious to create a working genetic algorithm as well as artificial neural network and train it to perform the tasks explained in this chapter.

The programming language chosen for this project is C++ as the simulator is written in it.

---

<sup>1</sup><http://goo.gl/BrPK98>



### 1.2.2 Choice of robot

### 1.2.3 Choice of environment design

## 1.3 Analysis

### 1.3.1 Communications

The Communication capabilities of the E-Puck where analysed. The Standard E-Puck comes with bluetooth communication and posses now WiFi capabilities.

Bluetooth communication for this project has been deemed infeasible as Bluetooth communications can take somewhere around  $19.5 \pm 4$  seconds. A multi robot exploration and mapping project such as this requires almost constant communication, in which case bluetooth connection times of  $\sim 19$  seconds are too long.

There are a few proposed ways to implement WiFi communications on the e-puck robot. One of the methods was proposed by Christopher M. Cianci *et al.* [2] is the creation and implementation of a WiFi extension board for the e-puck, enabling communication between ZigBee and other IEEE 802.15.4 compliant transceivers.

There designed communication board is based on the MSP430 Microcontroller<sup>2</sup> and the Chipcon CC2420<sup>3</sup> radio.

Allowing the e-puck a communication range between 15cm and 5 meters.

However such an board is not commercially available and would need to be custom designed and build, which is outside the spectrum of this project.

For the purposes of this project the use of the official e-puck range and bearing board has been deemed appropriate, as it is would be commercially available.

The range and bearing board is an extension board for the E-Puck which allows for localisation and local communication between E-Pucks using infra-red transmission. The board is powered by its own processor and consists of 12 sets of IR emission/reception modules. The board was first designed and build by Guílrrrez *et al.* [4].

### 1.3.2 Control Algorithm

The control algorithm for this project is an artificial neural network assisted by an genetic algorithm.

The background and use of those will be explain in further detail in this section.

#### 1.3.2.1 Artificial Neural Network

The control algorithm used for this project is an artificial neural network(ANN). Artificial neural networks are inspired by the brain.

A biological brain functions by passing electrical signals through nodes, so called neurons. Neurons of the brain can be compared to simple Input/Output connectors which transmit pulse coded

---

<sup>2</sup><http://www.ti.com/product/msp430f169>

<sup>3</sup><http://www.ti.com/product/cc2420>

analogue information. The relation between the inputs and outputs can be displayed a simple sigmoid function [6].

To a neuron not all inputs are the same however, different inputs(i.e. outputs from other neurons) have stronger influence on it than others, in neuroscience this is defined as synaptic weights.

This influence can be trained, in the course of a life a neuron *learns* to trust some inputs more than others, the same training is done in an artificial neural network. The synaptic weights are represented by the *weight* value each link between 2 nodes holds. This weight value is calculated and evolved using a genetic algorithm.

### 1.3.2.2 Genetic Algorithm

Genetic Algorithms(GA) are a evolutionary algorithm inspired by the genetics of living organism. An GA works by having a population of genes, which make out an chromosome. The genes in regard to this GA represent the weights used in the neural network.

A chromosome is constructed as followed, the first 1 gene holds a number representing the genotype length, in other words the number of genes in this chromosome and thereby the number of weights in the neural network. This number is calculated when the neural network is created at the start of the program and passed along to the GA.

The following genes represent each a weight for a specific link between 2 nodes in the neural network. The value is a number between 0 and 1.

The last gene in the chromosome holds the fitness assigned to this chromosome, which is calculated using the fitness function.

An genetic algorithms modifies the genes using operators which mimic their biological counterpart.

The genetic modifier described as *crossover* switches genes in chromosome to vary the genetic pool between generations.

This crossover can be done with either single genes, or groups of genes depending on the implementation, and of course any restrictions based on the nature of the data.

The other operator is mutation, which is the possibility that a random gene switches its value to another random value.

Which genes are chosen for crossover is based on the selection method implemented in the algorithm.

The third major part of an GA is the selection method, the method that is used to choose which chromosomes of the gene pool should carried over into the next generation.

The selection method implemented in this simulator is *Roulette Wheel Selection*. This selection method is part of the *Proportionate Reproduction* scheme this reproduction scheme chooses individuals to be carried over into the next generation based on their objective fitness function [3].

The basic part of the selection process is that the fittest individuals have the highest change to be carried over. This replicated nature in a way that a fitter individual tends to have a higher change of survival and will go forward to the mating pool of the next generation.

However weaker individuals(chromosomes) are not without a probabilistic change to get selected. It is called roulette wheel selection because its graphical representation is similar to a roulette

wheel, as figure 1.1 shows [11].

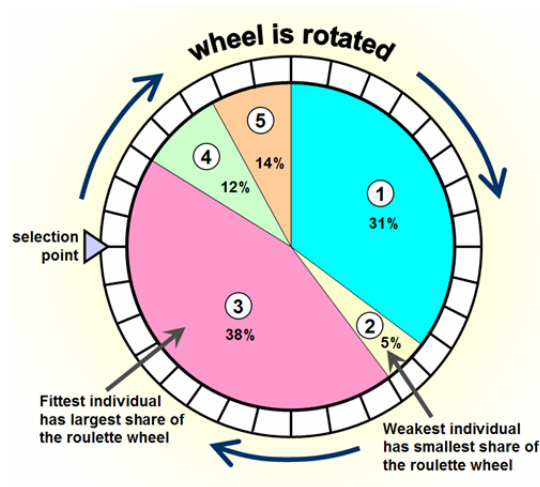


Figure 1.1: Roulette Wheel Selection<sup>4</sup>

As can be seen in figure 1.1 individuals with a higher fitness occupy a large of the overall available area.

The number of chromosomes used in this work is 100, of which an elite of 8 is chosen to be taken over into the next generation. As for the genetic operators, crossover has a probability of 0.3 and mutation a probability of 0.05 to occur.

## 1.4 Process

<sup>4</sup>Image credit: Newcastle University Engineering and Design Center <http://goo.gl/uwMSVB>

## Chapter 2

# Design

### 2.1 Overall Architecture

#### 2.1.1 Control Algorithm

In this section the design of the control algorithm will be explained.

#### 2.1.2 Artificial Neural Network

The robot swarm is controlled by an artificial neural network.

The neural network is consistent of 8 inputs, 3 hidden nodes, and 4 outputs.

The inputs are taken from the E-Pucks 8 IR proximity sensors. The hidden layer consists of 3 hidden nodes, which give more computational depth to the network.

From the 4 outputs of the neural the speed of the 2 stepper motors of the e-puck are calculated. This is done by calculating the difference between them.

The weights of the neural network are generated by a Genetic algorithm, which is part of the used simulator. There are bias nodes connected to the hidden and output layer, both of which are always set to 1.

The inputs to the ANN are returned from the E-Pucks 8 IR sensors and are a value between 0 and 4096.

As can be in the in figure 2.1 the returned IR sensors value rises drastically after the robot comes closer to an obstacle than 3 centimetres.

---

<sup>1</sup>Image credit: Webots User Guide <http://goo.gl/kyCINM>

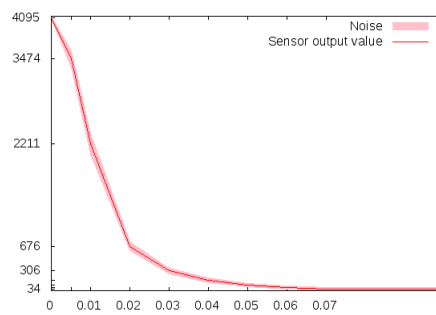


Figure 2.1: IR sensor response against distance<sup>1</sup>

## 2.2 Some detailed design

### 2.2.1 Even more detail

## 2.3 User Interface

## 2.4 Other relevant sections

## **Chapter 3**

# **Implementation**

## **Chapter 4**

# **Testing**

### **4.1 Overall Approach to Testing**

### **4.2 Automated Testing**

#### **4.2.1 Unit Tests**

#### **4.2.2 User Interface Testing**

#### **4.2.3 Stress Testing**

#### **4.2.4 Other types of testing**

### **4.3 Integration Testing**

### **4.4 User Testing**

## **Chapter 5**

# **Evaluation**



# Appendices

## **Appendix A**

# **Third-Party Code and Libraries**

## **Appendix B**

### **Code samples**

# Annotated Bibliography

- [1] A. Birk and S. Carpin, “Merging Occupancy Grid Maps From Multiple Robots,” *Proceedings of the IEEE*, vol. 94, no. 7, pp. 1384–1397, July 2006. [Online]. Available: <http://dx.doi.org/10.1109/jproc.2006.876965>
- [2] C. Cianci, X. Raemy, J. Pugh, and A. Martinoli, “Communication in a Swarm of Miniature Robots: The e-Puck as an Educational Tool for Swarm Robotics,” in *Swarm Robotics*, ser. Lecture Notes in Computer Science, E. Şahin, W. Spears, and A. Winfield, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2007, vol. 4433, ch. 7, pp. 103–115. [Online]. Available: [http://dx.doi.org/10.1007/978-3-540-71541-2\\_7](http://dx.doi.org/10.1007/978-3-540-71541-2_7)
- [3] D. E. Goldberg and K. Deb, “A comparative analysis of selection schemes used in genetic algorithms,” *Foundations of genetic algorithms*, vol. 1, pp. 69–93, 1991.
- [4] A. Gutiérrez, A. Campo, M. Dorigo, D. Amor, L. Magdalena, and F. Monasterio-Huelin, “An Open Localization and Local Communication Embodied Sensor,” *Sensors*, vol. 8, no. 11, pp. 7545–7563, Nov. 2008. [Online]. Available: <http://dx.doi.org/10.3390/s8117545>
- [5] A. Gutiérrez, A. Campo, M. Dorigo, D. Amor, L. Magdalena, and F. Monasterio-Huelin, “An open localization and local communication embodied sensor,” *Sensors*, vol. 8, no. 11, pp. 7545–7563, 2008.
- [6] J. Hopfield, “Artificial neural networks,” *Circuits and Devices Magazine, IEEE*, vol. 4, no. 5, pp. 3–10, Sept 1988.
- [7] R. Kala, A. Shukla, and R. Tiwari, “Robotic path planning using evolutionary momentum-based exploration,” *Journal of Experimental & Theoretical Artificial Intelligence*, vol. 23, no. 4, pp. 469–495, July 2011. [Online]. Available: <http://dx.doi.org/10.1080/0952813x.2010.490963>
- [8] Q. Meng and M. H. Lee, “Active Exploration in Building Hierarchical Neural Networks for Robotics,” in *Instrumentation and Measurement Technology Conference, 2006. IMTC 2006. Proceedings of the IEEE*. IEEE, Apr. 2006, pp. 2095–2100. [Online]. Available: <http://dx.doi.org/10.1109/imtc.2006.328464>
- [9] S. Thrun, “Learning Occupancy Grid Maps with Forward Sensor Models,” *Autonomous Robots*, vol. 15, no. 2, pp. 111–127, Sept. 2003. [Online]. Available: <http://dx.doi.org/10.1023/a:1025584807625>
- [10] K. M. Wurm, A. Hornung, M. Bennewitz, C. Stachniss, and W. Burgard, “OctoMap: A probabilistic, flexible, and compact 3D map representation for robotic systems,” in *Proc.*

*of the ICRA 2010 workshop on best practice in 3D perception and modeling for mobile manipulation*, vol. 2, 2010.

- [11] J. Zhong, X. Hu, M. Gu, and J. Zhang, “Comparison of performance between different selection strategies on simple genetic algorithms,” in *Computational Intelligence for Modelling, Control and Automation, 2005 and International Conference on Intelligent Agents, Web Technologies and Internet Commerce, International Conference on*, vol. 2, Nov 2005, pp. 1115–1121.