

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

1	Background & Objectives	1
1.1	Background	1
1.2	Analysis	1
1.2.1	Communications	2
1.2.2	Control Algorithm	2
1.3	Process	3
2	Design	4
2.1	Overall Architecture	4
2.1.1	Control Algorithm	4
2.2	Some detailed design	5
2.2.1	Even more detail	5
2.3	User Interface	5
2.4	Other relevant sections	5
3	Implementation	6
4	Testing	7
4.1	Overall Approach to Testing	7
4.2	Automated Testing	7
4.2.1	Unit Tests	7
4.2.2	User Interface Testing	7
4.2.3	Stress Testing	7
4.2.4	Other types of testing	7
4.3	Integration Testing	7
4.4	User Testing	7
5	Evaluation	8
	Appendices	9
A	Third-Party Code and Libraries	10
B	Code samples	11
2.1	Random Number Generator	11
	Annotated Bibliography	14

LIST OF FIGURES

LIST OF TABLES

Chapter 1

Background & Objectives

This section should discuss your preparation for the project, including background reading, your analysis of the problem and the process or method you have followed to help structure your work. It is likely that you will reuse part of your outline project specification, but at this point in the project you should have more to talk about.

Note:

- All of the sections and text in this example are for illustration purposes. The main Chapters are a good starting point, but the content and actual sections that you include are likely to be different.
- Look at the document on the Structure of the Final Report for additional guidance.

1.1 Background

What was your background preparation for the project? What similar systems did you assess? What was your motivation and interest in this project?

1.2 Analysis

Taking into account the problem and what you learned from the background work, what was your analysis of the problem? How did your analysis help to decompose the problem into the main tasks that you would undertake? Were there alternative approaches? Why did you choose one approach compared to the alternatives?

There should be a clear statement of the objectives of the work, which you will evaluate at the end of the work.

In most cases, the agreed objectives or requirements will be the result of a compromise between what would ideally have been produced and what was felt to be possible in the time available. A discussion of the process of arriving at the final list is usually appropriate.

1.2.1 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 ± 4 seconds. A multi robot exploration and mapping project such as this requires almost constant communication, in which case bluetooth connection times of ~ 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 Microcontroller¹ and the Chipcon CC2420² 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.* [3].

1.2.2 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.

1.2.2.1 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 analogue information. The relation between a the inputs and outputs can be displayed a simple sigmoid function [5].

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

¹<http://www.ti.com/product/msp430f169>

²<http://www.ti.com/product/cc2420>

represented by the *weight* value each link between 2 nodes holds. This weight value is calculated and evolved using a genetic algorithm.

1.2.2.2 Genetic Algorithm

1.3 Process

You need to describe briefly the life cycle model or research method that you used. You do not need to write about all of the different process models that you are aware of. Focus on the process model that you have used. It is possible that you needed to adapt an existing process model to suit your project; clearly identify what you used and how you adapted it for your needs.

Chapter 2

Design

You should concentrate on the more important aspects of the design. It is essential that an overview is presented before going into detail. As well as describing the design adopted it must also explain what other designs were considered and why they were rejected.

The design should describe what you expected to do, and might also explain areas that you had to revise after some investigation.

Typically, for an object-oriented design, the discussion will focus on the choice of objects and classes and the allocation of methods to classes. The use made of reusable components should be described and their source referenced. Particularly important decisions concerning data structures usually affect the architecture of a system and so should be described here.

How much material you include on detailed design and implementation will depend very much on the nature of the project. It should not be padded out. Think about the significant aspects of your system. For example, describe the design of the user interface if it is a critical aspect of your system, or provide detail about methods and data structures that are not trivial. Do not spend time on long lists of trivial items and repetitive descriptions. If in doubt about what is appropriate, speak to your supervisor.

You should also identify any support tools that you used. You should discuss your choice of implementation tools - programming language, compilers, database management system, program development environment, etc.

Some example sub-sections may be as follows, but the specific sections are for you to define.

2.1 Overall Architecture

2.1.1 Control Algorithm

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.

2.2 Some detailed design

2.2.1 Even more detail

2.3 User Interface

2.4 Other relevant sections

Chapter 3

Implementation

The implementation should look at any issues you encountered as you tried to implement your design. During the work, you might have found that elements of your design were unnecessary or overly complex; perhaps third party libraries were available that simplified some of the functions that you intended to implement. If things were easier in some areas, then how did you adapt your project to take account of your findings?

It is more likely that things were more complex than you first thought. In particular, were there any problems or difficulties that you found during implementation that you had to address? Did such problems simply delay you or were they more significant?

You can conclude this section by reviewing the end of the implementation stage against the planned requirements.

Chapter 4

Testing

Detailed descriptions of every test case are definitely not what is required here. What is important is to show that you adopted a sensible strategy that was, in principle, capable of testing the system adequately even if you did not have the time to test the system fully.

Have you tested your system on real users? For example, if your system is supposed to solve a problem for a business, then it would be appropriate to present your approach to involve the users in the testing process and to record the results that you obtained. Depending on the level of detail, it is likely that you would put any detailed results in an appendix.

The following sections indicate some areas you might include. Other sections may be more appropriate to your project.

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

Examiners expect to find in your dissertation a section addressing such questions as:

- Were the requirements correctly identified?
- Were the design decisions correct?
- Could a more suitable set of tools have been chosen?
- How well did the software meet the needs of those who were expecting to use it?
- How well were any other project aims achieved?
- If you were starting again, what would you do differently?

Such material is regarded as an important part of the dissertation; it should demonstrate that you are capable not only of carrying out a piece of work but also of thinking critically about how you did it and how you might have done it better. This is seen as an important part of an honours degree.

There will be good things and room for improvement with any project. As you write this section, identify and discuss the parts of the work that went well and also consider ways in which the work could be improved.

Review the discussion on the Evaluation section from the lectures. A recording is available on Blackboard.

Appendices

Appendix A

Third-Party Code and Libraries

If you have made use of any third party code or software libraries, i.e. any code that you have not designed and written yourself, then you must include this appendix.

As has been said in lectures, it is acceptable and likely that you will make use of third-party code and software libraries. The key requirement is that we understand what is your original work and what work is based on that of other people.

Therefore, you need to clearly state what you have used and where the original material can be found. Also, if you have made any changes to the original versions, you must explain what you have changed.

As an example, you might include a definition such as:

Apache POI library The project has been used to read and write Microsoft Excel files (XLS) as part of the interaction with the clients existing system for processing data. Version 3.10-FINAL was used. The library is open source and it is available from the Apache Software Foundation [?]. The library is released using the Apache License [?]. This library was used without modification.

Appendix B

Code samples

2.1 Random Number Generator

The Bayes Durham Shuffle ensures that the psuedo random numbers used in the simulation are further shuffled, ensuring minimal correlation between subsequent random outputs [?].

```
#define IM1 2147483563
#define IM2 2147483399
#define AM (1.0/IM1)
#define IMM1 (IM1-1)
#define IA1 40014
#define IA2 40692
#define IQ1 53668
#define IQ2 52774
#define IR1 12211
#define IR2 3791
#define NTAB 32
#define NDIV (1+IMM1/NTAB)
#define EPS 1.2e-7
#define RNMX (1.0 - EPS)

double ran2(long *idum)
{
    /*-----*/
    /* Minimum Standard Random Number Generator */
    /* Taken from Numerical recipies in C */
    /* Based on Park and Miller with Bays Durham Shuffle */
    /* Coupled Schrage methods for extra periodicity */
    /* Always call with negative number to initialise */
    /*-----*/

    int j;
    long k;
    static long idum2=123456789;
```

```
static long iy=0;
static long iv[NTAB];
double temp;

if (*idum <=0)
{
    if (-(*idum) < 1)
    {
        *idum = 1;
    }else
    {
        *idum = -(*idum);
    }
    idum2=(*idum);
    for (j=NTAB+7; j>=0; j--)
    {
        k = (*idum)/IQ1;
        *idum = IA1 *(*idum-k*IQ1) - IR1*k;
        if (*idum < 0)
        {
            *idum += IM1;
        }
        if (j < NTAB)
        {
            iv[j] = *idum;
        }
    }
    iy = iv[0];
}
k = (*idum)/IQ1;
*idum = IA1*(*idum-k*IQ1) - IR1*k;
if (*idum < 0)
{
    *idum += IM1;
}
k = (idum2)/IQ2;
idum2 = IA2*(idum2-k*IQ2) - IR2*k;
if (idum2 < 0)
{
    idum2 += IM2;
}
j = iy/NDIV;
iy=iv[j] - idum2;
iv[j] = *idum;
if (iy < 1)
{
    iy += IMM1;
}
```

```
if ((temp=AM*iy) > RNMx)
{
    return RNMx;
}else
{
    return temp;
}
}
```

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
- [3] 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>
- [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, 2008.
- [5] J. Hopfield, “Artificial neural networks,” *Circuits and Devices Magazine, IEEE*, vol. 4, no. 5, pp. 3–10, Sept 1988.
- [6] 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>
- [7] 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>
- [8] 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>
- [9] 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.