**Design of a virtual sensor using machine learning imputation techniques in a wireless sensor network.**

Final Report

**M. Matusowsky**

11093804

Submitted as partial fulfilment of the requirements of Project EPR402

in the Department of Electrical, Electronic and Computer Engineering

University of Pretoria

November 2018

Study leader: Mr D. Ramotsoela

Copy 1 of 4

|  |
| --- |
| **Part 1. Preamble** |

This report describes the work that I did in designing a virtual sensor for a wireless sensor network using a machine learning technique.

*Project proposal and technical documentation*

This main report contains a copy of the approved Project Proposal (Part 3 of the report) and technical documentation (Part 5 of the report). The latter provides details of the schematics, stripboard layout, and software code. I have included this appendix on a CD or DVD that accompanies this report.

*Project history*

This project does not build on any projects that were completed in previous years. stripboard design software as well as the circuit error checking software was provided by Fritzing.org, an opensource initiative. The stripboard sensor circuit was designed and soldered by myself. One of the equations I used was adapted from Steinhart [1]. The training algorithms were based on the work published by Montana [2]. I used the Tkinter python library to implement the graphical user interface. All other work in this report was my own and completed from first principles.

*Language editing*

This document has been language edited by a knowledgeable person. By submitting this document in its present form, I declare that this is the written material that I wish to be examined on.

My language editor was Mr Joseph Henry Mervitz

*\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*

*Language editor signature Date*

*Declaration*

I, Michael Matusowsky understand what plagiarism is and have carefully studied the plagiarism policy of the University. I hereby declare that all the work described in this report is my own, except where explicitly indicated otherwise. Although I may have discussed the design and investigation with my study leader, fellow students or consulted various books, articles or the internet, the design/investigative work is my own. I have mastered the design and I have made all the required calculations in my lab book (and/or they are reflected in this report) to authenticate this. I am not presenting a complete solution of someone else.

Wherever I have used information from other sources, I have given credit by proper and complete referencing of the source material so that it can be clearly discerned what is my own work and what was quoted from other sources. I acknowledge that failure to comply with the instructions regarding referencing will regarded as plagiarism. If there is any doubt about the authenticity of my work, I am willing to attend an oral ancillary examination/evaluation about the work.

I certify that the Project Proposal appearing as the Introduction section of the report is a verbatim copy of the approved Project Proposal.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

M. Matusowsky Date

**TABLE OF CONTENTS**

**Part 2. Summary** vi

What has been done vi

What has been achieved vi

Findings vi

Contribution vi

**Part 3. Project identification: approved Project Proposal** viii

1. Problem statement viii
2. User requirements ix
3. Functional analysis xi
4. Specifications xiii
5. Deliverables xvii
6. References xiv

**Part 4. Main report**

1. Literature study 1

1.1 Background and context of the problem 1

1.1.1 Digital modulation techniques 1

2. Approach 4

2.1 Design alternatives 4

2.2 Preferred solution 6

3. Design and implementation

3.1 Theoretical analysis and modelling 8

3.2 Simulation study and optimization 12

3.3 Hardware design 16

3.3.1 Design alternatives 16

3.3.1 Design procedure followed 17

3.3.2 Design equations and design calculations 17

3.3.3 Final design schematics 19

3.4 Hardware implementation

3.4.1 PC board layout and manufacturing process 22

3.4.2 Hardware integration and testing 25

3.5 Software design 27

3.5.1 Design alternatives 27

3.5.2 Design procedure followed 27

3.5.3 Design calculations 29

3.6 Software implementation 30

3.7 Final system integration and testing 32

3.8 Design summary

4. Results 34

4.1 Summary of results achieved 34

4.2 Qualification tests 34

4.2.1 Qualification test 1: testing of communication distance 34

4.2.1.a Qualification test

*Objectives of test/experiment* 34

*Equipment used* 34

*Experimental parameters and setup* 35

*Experimental protocol (experimental steps)* 36 4.2.1b Results and observations

objective description

*Measurements*  37

*Description of results* 38

*Statistical analysis* 39

4.2.2 Qualification test 2: measurement of bit error rate 40

4.2.2.a Qualification test

*Objectives of test/experiment* 40

*Equipment used* 40

*Experimental parameters and setup* 40

**Very important section - read notes in study guide**

*Experimental protocol (experimental steps)* 42 4.2.2b Results and observations

*Measurements*  43

*Description of results* 45

*Statistical analysis* 46

5. Discussion (subjective interpretation of the implications of the results)

5.1 Interpretation of results 50

5.2 Aspects to be improved in current design 51

5.3 Strong points of current design 51

5.4 Under which circumstances will the current system fail? 52

…

6. Conclusion 55

6.1 Summary of the work 56

6.2 Summary of the results and conclusions 58

6.3 Suggestions for future work 59

7. References 60

**Part 5. Technical documentation** 61

**LIST OF ABBREVIATIONS**

**TDD**  Test-driven development

**PIC32** Peripheral interface controller 32-bit

**MCU** Microcontroller Unit

**TCP/IP** Transmission control protocol/Internet protocol

**PC** Personal computer

|  |
| --- |
| **Part 2. Summary** |

This report describes work carried out on the design of a virtual sensor using machine learning in a wireless sensor network with the objective of using the virtual sensor to impute sensor values of a single sensor node, if a sensor node in the network fails, using the other sensor nodes that are still actively transmitting.

**What has been done**

A literature survey was completed on modern wireless communication systems as well as different machine learning implementations. The hardware and software for the sensor nodes and server was then designed and implemented from first principles. communication system was then designed from first principles. At the core of the hardware system is three PIC32 (Peripheral interface controller 32-bit) microcontroller units (MCU) with an ESP8266 WiFi module that includes a transmission control protocol/internet protocol (TCP/IP) framework, and all additional hardware was designed and implemented. A Python program was developed to simulate the system, as well as C code for the PIC32 processor. The system was implemented and tested throughout the development phase using a test-driven development (TDD) process. Four WiFi modules were used in a star network topology whereby one WiFi module acted as the TCP/IP server on the desktop personal computer (PC) and the other three modules acted as TCP/IP clients on the sensor nodes. The neural network for each sensor node was implemented on the server as well as on the corresponding sensor node’s MCU. The main results from one of the test cases is shown in the figures below.

(Add a figure or figures here – this will probably be a figure that also appears in section 4 of the report).

**What has been achieved**

Successful imputation using machine learning was achieved in a wireless sensor network that was deployed in multiple locations. The accuracy of the imputation depended highly on the amount of training data available where it was discovered that a minimum of 5 days of data was required before a sensor node would be adequately trained. Figure X shows a virtual sensor trained with 2 days of data while figure Y shows a virtual sensor trained with 5 days of data.

**Findings**

It was found, while analysing the weights of the neural networks, that the time input in the neural network was much more important than the sensor inputs with regards to imputing temperature data. Another important discovery was that due to glitches within the hardware, the sensed value by the thermistor would sometimes return a ground value and thus a filtering algorithm had to be implemented to be implemented to account for the rare glitches. Such an occurrence is shown in figure X below.

**Contribution**

New software that had to be mastered to complete this project was Fritzing. Fritzing is an opensource PCB/stripboard/circuit designing software that undergraduate students would not usually be aware of and has not been covered in any undergraduate module. The Fritzing software was used specifically for the stripboard module to plan, test and implement the circuitry on a stripboard. Further software that needed to be learned was the MPLAB Harmony Framework for PIC32 MCU’s which was extremely useful in configuring the right clock speed as well as the Universal Synchronous/Asynchronous Receiver/Transmitter (USART) and analog-to-digital converter (ADC) modules.

Due to the limited amount of program memory in the PIC32 and the requirement for string handling, some functions usually provided by the standard library for C had to be re-written from first principles due to the overhead in program memory introduced when using functions such as sprint() which would allow data to be converted from integers and floats to strings and then written to the USART buffer.

A server was developed using Python on a PC connected by USB with a WiFi module. The server would act as the communication center to periodically request and transmit readings from the sensor nodes as well as determine if a sensor node is offline. The code for the serial communication was implemented by the student with the use of the pyserial library.

Code for the neural network, including the training algorithms, was developed by the student without reliance on any existing libraries in python. A framework for training neural networks with a minimum of 3 inputs, theoretically unlimited hidden nodes and layers, and a maximum of 1 output was developed from first principles. A friend in a postgraduate course helped provide some clarity on the chosen training method that is used as it was complex and completely new to the student as no undergraduate module covers evolutionary models for training in neural networks. The study leader provided no assistance with regards to the training but did provide some advice regarding the output layer of the neural network as the student originally wanted to have the output be a binarized output layer with multiple output nodes. The study leader advised to use a single output node in the output layer.

There was a strong reliance on an existing python library called csv to deal with reading and writing from the local database.

|  |
| --- |
| **Part 3. Project identification: approved Project Proposal** |

This section contains the problem identification in the form of the complete approved Project Proposal, unchanged from the final approved version.

Paste the final, approved version of your Project Proposal into the final report at this point, including the references as they appear in the Project Proposal. I.e., the statement above will be followed by the first heading of the Project Proposal, “1.1 Problem statement”.

This should not be the graded copy (i.e. with pen markings), but a new printout of the final approved version, with renumbered headings (see description in study guide).

Be sure to read the warning in Appendix 4 regarding changes to the approved Project Proposal.

|  |
| --- |
| 1. **Problem statement** |

This is the first heading of the Project Proposal. The rest of the text of the Project Proposal then follows here, simply copied and pasted from the approved version (but with formatting corrected – see study guide)

|  |
| --- |
| **Part 4. Main report** |

**1. Literature study**

Your literature study, described in the study guide, goes here.

**2. Approach**

Commence this section on a new page

Your approach study, described in the study guide, goes here.

**2.1 Design alternatives**

**2.2 Preferred solution**

**3. Design and implementation**

Commence this section on a new page

**3.1 Theoretical analysis and modelling**

…

Very important table in this section.

The required content is described in the study guide. What appears here, is an example.

**3.7 Design summary**

This section summarises the project tasks and how they were implemented (see table 1).

|  |  |  |
| --- | --- | --- |
| **Task** | **Implementation** | **Task completion** |
| Design of a PCB for the main electronics | The PCB design was completed, using the PCBCAD package. This was done from first principles. | completed |
| Development of optimization routine | Optimization was completed in Matlab, but the Optimization Toolbox was used, and while some code was developed from first principles, numerical methods for optimization were taken off the shelf. | incomplete |

**Table 1**

**Design summary.**

Ensure that your table lists all of your tasks (technical design tasks, not including things like “write report”, or “bind report”). These would include student tasks in sections 2.3 and 2.5 of the Project Proposal.

**4. Results**

Commence this section on a new page

**Super important! The compulsory table mentioned in Appendix 4 of the study guide should appear here.**

**The following is just an example.**

**4.1 Summary of results achieved**

|  |  |  |
| --- | --- | --- |
| **Description of requirement or specification**  **(intended outcome)** | **Actual outcome** | **Location in report** |
| **Mission requirements of the product** | | |
| The system should provide continuous AC power to a household | The system could provide continuous power during daytime hours when grid power failed. | Section 4.2.6 |
| Motor speed should be controlled. | Stable feedback control of the motor speed was achieved. | Section 4.2.2 |
| Fuel consumption must be kept to a minimum. | Fuel consumption was slightly higher than expected. | Section 4.2.7 |
| Bit error rate (BER) should be low. | The measured BER was high. | Section4.2.1 |
| Delivered power should be adequate for the load. | The system could not deliver the required power into the load. | Section 4.2.3 |
| **Field conditions** | | |
| The system should supply power and actual environmental conditions (sunshine or rain; day or night) | The system was never tested under rainy conditions. The system could not supply power under any conditions other than bright sunlight. | Section 4.2.6 |
| The system should use actual real-time data, corrupted by noise, arriving over a noisy wireless link. | The system could work error-free for at least one hour under these actual field conditions. | Section 4.2.1 |
| **Specifications** | | |
| BER should be below 1E-6. | The BER was measured as 10 bit errors in 1000 bits. | Section 4.2.1 |
| The motor should reach 50 rpm. | 46 rpm was measured. | Section 4.2.2 |
| 2 kW should be delivered to the load. | The system could deliver 800 Watts into the load before overheating. | Section 4.2.3 |
| **Deliverables** | | |
| DC-DC convertors had to be designed and implemented by the student. | The student completed the design and implementation. | Section 4.2.8 |
| The inverter had to be designed and implemented by the student. | The student did not complete the inverter. The design was completed and simulated, but the implementation in hardware did not work correctly. | Section 4.2.8 |

**TABLE 2**

**Summary of results achieved.**

**4.2 Qualification tests**

**4.2.1 Qualification test 1: testing of communication distance**

4.2.1.a Qualification test

*Objectives of test/experiment*

*Equipment used*

*Experimental parameters and setup*

*Experimental protocol (experimental steps)*

4.2.1b Results and observations

*Measurements*

*Description of results*

*Statistical analysis*

**4.2.2 Qualification test 2: measurement of bit error rate**

4.2.2.a Qualification test

*Objectives of test/experiment*

*Equipment used*

*Experimental parameters and setup*

*Experimental protocol (experimental steps)*

4.2.2b Results and observations

*Measurements*

*Description of results*

*Statistical analysis*

**5. Discussion**

**See the study guide – this section is very important. You need to show that you can stand back and be critical of your own work. The worst possible thing that you can write here is “everything works perfectly”. There is no perfect design, and you as (aspiring) engineer should be able to point out the shortcomings of your design and/or results.**

**Tip: Use the headings in the study guide. These are *not* compulsory, but will help you to organize your thoughts, and the headings actually tell you some of the things that you are required to comment on.**

Commence this section on a new page

**6. Conclusion**

Commence this section on a new page

**6.1 Summary of the work**

**…**

**7. References**

**Here you need to be extremely honest about what you achieved, and did not achieve.**

**Conclusions need to be technical and MAY NOT relate to your personal experience (e.g. “I learnt a lot” would be a good example of what NOT to write).**

|  |
| --- |
| **Part 5. Technical documentation** |

This main report is supplemented with technical documentation. This provides more detail on the software that was used in the experiments, including program listings, a user guide and circuit designs. This section appears on the electronic medium that accompanies this report.

The CD (or DVD, or flash disk) is organized into the directories listed below.

*Main report*

**Use this text**

*Part 5: Technical documentation*

*Software*

*References*

*Datasheets*

*Author*

**indicate the actual directories on your CD here. The minimum required directories are listed in Appendix 4 to the study guide.**

*Datasets/Raw*

*Datasets/Final*