COMP4952: Thesis B

A Computer Model of Electrocardiogram Signals

By Andrew Au (z5020593)

Supervisor: Socrates Dokos

Assessor: Bruno Gaeta

Project Description

Electrocardiogram signals **(ECGs)** are signals generated by electric activity in the heart. These signals are captured by ECG machines.

ECGs are used to diagnose medical conditions in patients such as arrhythmia.

A mathematical model of an ECG describes the motion of an ECG.

A computer model of an ECG uses this mathematical model to generate ECGs.

Project Description

A computer model is beneficial as it can generate different signals based on regular or irregular heartbeats (heart arrhythmia).

Computer models are useful for research purposes since they can be used to generate different ECGs. Generated ECGs can be tested against different algorithms without having to record actual ECGs from a patient.

For instance, arrhythmia detection algorithms are integrated into ECG machines for patient monitoring. Computer models can be used to facilitate the process of refining of these algorithms by generating different ECGs for the given condition.

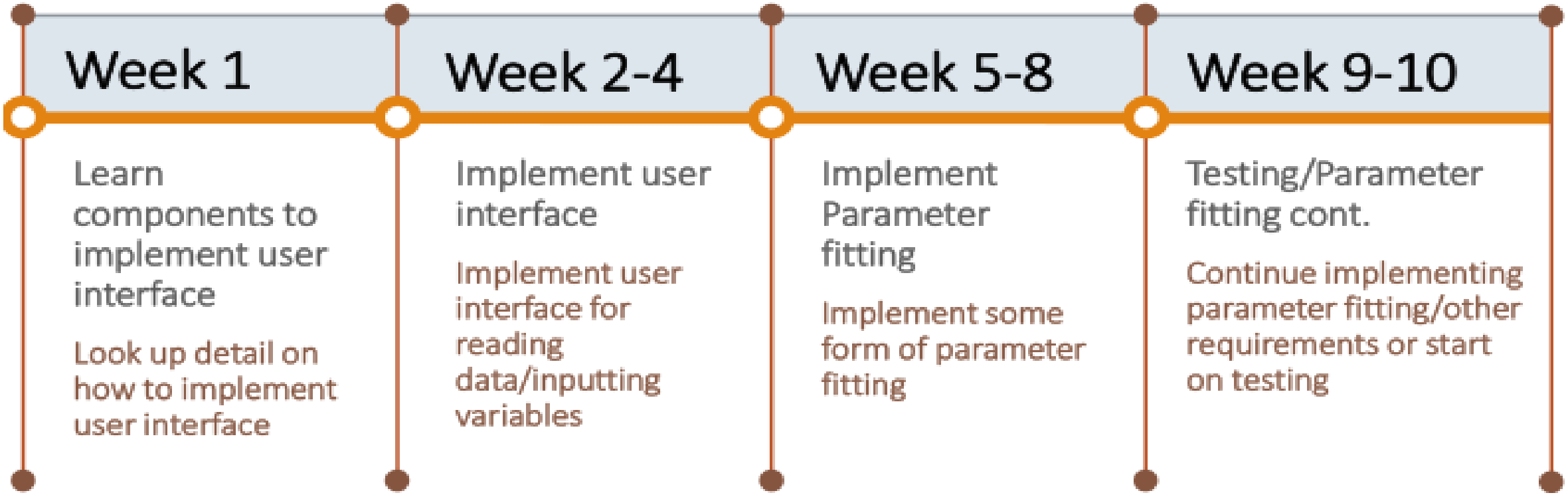
Project Plan

In CSE Thesis A (COMP4951) a plan was drawn up to develop an application that would use a computer model of ECGs that is both open source and that could generate normal and irregular heartbeats.

It was then proposed that the implementation would be able to perform parameter fitting to another ECG.

Project Plan: Thesis A (COMP4951)

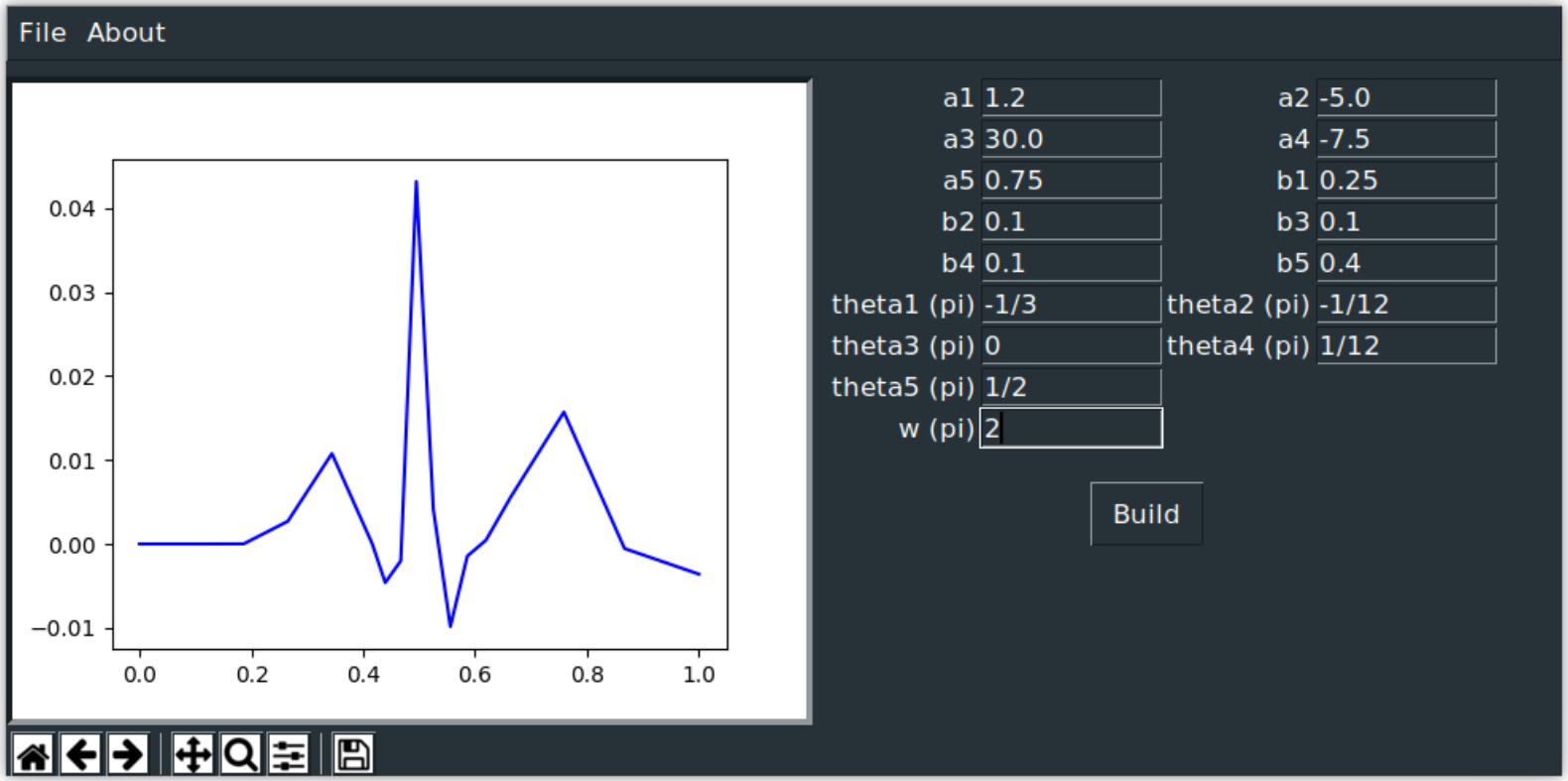
Project Plan for Thesis B (COMP4952) from Thesis A (COMP4951)



Project Plan: Thesis A (COMP4951)

Initial results:

- Python3
- Tkinter3
- SciPy
  + NumPy
  + Matplotlib



Progress Summary for COMP4952

Component	Status
Initial results (generate ECG on a plot)	Complete
Tkinter import/export parameters for ECG	Complete
Transition from Tkinter to Qt (PyQt)	Complete
Validate parameter values (number input and pi)	Complete
PyQt dialogs (parse warning, import/export parameters dialog)	Complete
PyQt import sample ECG data (using csv from Physionet)	Complete
Matplotlib use two axes to store two graphs (store both generated and sample)	Complete
PyQt Widgets (Slider to select timeframe of data	Complete
PyQt Graph/Matplotlib features (title, legends, labels)	Complete
Kalman Filtering	Pending

Implementation

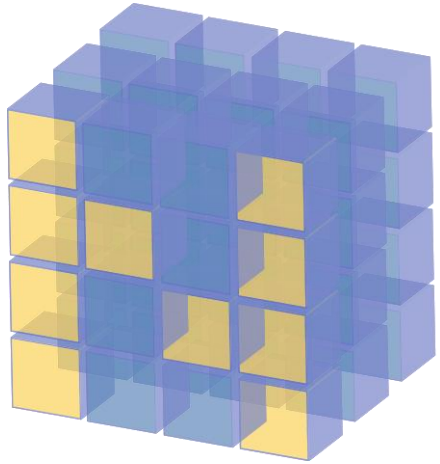
Python:

- Good set of libraries supporting scientific and numerical computing and Graphical User Interfaces (GUI)

Implementation

NumPy

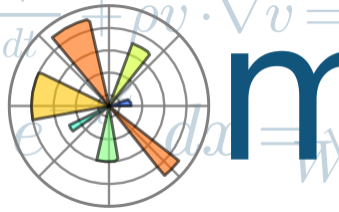
NumPy to implement and solve ODE outlined in [Dynamic Model for Generating ECGs].



NumPy

Matplotlib

Used to plot graphs. Has bindings for both tkinter and PyQt.



matplotlib

Graphical User Interface (GUI)

Tkinter3

De-facto standard GUI for Python. Object oriented layer on top of Tcl/Tk.

Qt5 (PyQt5)

Open-source widget toolkit for creating graphical user interfaces.

PyQt5 is a comprehensive set of Python bindings for Qt v5

Tkinter vs PyQt

Tkinter

Pros:

- Simple, suitable for small applications
- Usually comes with Python

Cons:

- Looks old, needs effort to make it look good
- Needs more lines for layouts, frames, etc

PyQt (based on Qt)

Pros:

- Adapts to operating system
- Modern, looks good
- Widgets are easy to use
- Less lines of code for strong features

Cons:

- Licensing requirements (for commercial use)
- Needs extra installation

Kalman Filtering

Method to perform parameter fitting against imported ECG sample data.

Kalman Filtering is a state based estimation method that can be used to filter noise.

In order to use Kalman Filtering for parameter fitting of ECGs, an *Extended Kalman Filter* needs to be implemented.

Details of *Extended Kalman Filter (EKF)* implementation:

Filtering ECGs Using Extended Kalman Filter

Project Plan: Thesis C

Week 1-3

Extended Kalman Filtering

Complete implementation of Extended Kalman Filtering

Week 4-6

Testing, Integration over Operating Systems

Test filtering over ECG samples obtained from Physionet.

Implement further changes to GUI for different operating systems.

Week 7

Packaging/ distribution

Prepare everything for open source distribution

Week 8-10

Final Report

Write final report

End of Presentation