Department of Electrical, Computer, and Software Engineering

Part IV Research Project

Compendium Report

Project Number: 34

Pacemakers for Gastrointestinal Diseases

Yogesh Dangwal

Glacer Barnett

Dr Avinash Malik and Dr Partha Roop

[15/10/2023]

Declaration of Originality

This report is my own unaided work and was not copied from nor written in collaboration with any other person.

[Yogesh Dangwal]

Name: Yogesh Dangwal

Contents

1. Introduction

This document delves into the short literature review, and project's progression, detailed in section 3, followed by a compilation of files presented in section 4. Our project's ambition was to tread uncharted waters in this domain—evaluating the potential of infusing real-time responsiveness into these medical devices. Our primary challenge was to transition the foundational research, which was predominantly in MATLAB, to C code to make it compatible with the embedded De1-SoC processor. Alongside this, we explored Machine Learning (ML) techniques, aiming to discern if we could derive binary predictions to gauge the precision of a confusion matrix.Readers are recommended to refer to the comprehensive report for an in-depth analysis of the project, covering topics such as the literature review, implementation processes, methodologies, and outcomes.

2. Literature Review

2.1.Gastrointestinal (GI) Motility, Gastric Electrical Stimulation (GES) and challenges

The crucial role of Interstitial Cells of Cajal (ICC) in generating bio-electrical slow waves, pivotal for smooth muscle contractions, and, thus, GI motility, is acknowledged.

Various GI motility disorders, such as gastroparesis, exhibit debilitating symptoms due to alterations in ICC-mediated slow-wave patterns.

Significant advancements in GES have highlighted its potential in modulating GI slow waves, and improving symptoms among gastroparesis patients.

Limitations, including non-optimal open-loop systems and the non-existence of FDA-approved closed-loop gut pacemakers, indicate the necessity for further exploration in this realm. While attempts (such as by Wang et al.) have been made to formulate closed-loop GES systems to better regulate gastric slow waves, a conspicuous gap persists, notably in effectively operating amidst noise during real-world applications that have yet no embedded implementation to it.

2.2. Signal Detection and Embedded Systems:

- The intrinsic dynamics of gastric slow waves and detection methodologies, despite their advancements, continue to wrestle with the conundrum of noise and real-world variability in signal detection.
- The application of algorithms and detection techniques in real-time and in online capabilities necessitates a seamless blend of accuracy, low computational and power demand, hinting towards the adoption and modification of methods like the Falling Edge Variable Threshold (FEVT) algorithm.
- Embedded systems in medical devices, such as pacemakers, have revolutionised treatment modalities, but the application within the GI domain demands exploration into low-power, efficient, and robust systems that cater to the specificity of GI

electrophysiology.

- Striking a balance between technological efficiency and biological compatibility within the constraints of an implantable device becomes imperative to forge ahead in the development of closed-loop GES systems, hosted on an embedded platform.

2.3.Development of Embedded Devices for Gastrointestinal Systems:

Developing embedded devices for gastrointestinal systems necessitates a meticulous blend of biomedical and technological expertise, addressing power management, biocompatibility, signal processing, wireless communication, and algorithm implementation. Researchers are focused on optimizing energy consumption through low-power microcontrollers and energy harvesting mechanisms, ensuring device safety with biocompatible materials and structural stability, and maintaining precise signal processing amidst the biological noise inherent to the gastrointestinal environment. Additionally, embedding secure wireless communication modules and implementing efficient algorithms capable of real-time decision-making and therapeutic interventions within hardware constraints are pivotal in enhancing patient management and improving quality of life amidst the complexities of gastrointestinal motility disorders. This compact convergence of multifaceted technological and biomedical insights propels the domain towards innovative, efficacious embedded medical solutions.

3. Project Progression

3.1.* Phase 1: Project Kick-off

Our journey began with foundational meetings to gauge the extent of research done at the University of Auckland in this domain. Luman Wang's endeavours revolved around a closed-loop gastric pacemaker needing an enhanced detection algorithm. The primary challenge was migrating from MATLAB to an embedded system. Considering the challenges faced by conventional detection methods, Machine Learning (ML) was proposed as a potential alternative.

3.2.Phase 2: Preliminary Studies

Our initial deliverable was the research literature. We delved deep into both ML techniques and orthodox methodologies. Key challenges in creating an ML algorithm emerged:

- The complexity and size of some ML models make them unsuitable for deployment on embedded systems, which might require model simplification, thereby potentially sacrificing accuracy or performance.
- Certain ML models were deemed unfit due to their "black-box" nature in a safety-critical application.
- ML's computational demands raised concerns about real-time feasibility.
- Early tests with rudimentary ML models for binary classification indicated data inconsistency.

On the Other hand, the orthodox methodology suggested by Marr and Vanpraseuth model seems to be very promising and much easier to migrate to embedded systems as compared to ML.

3.3. Phase 3: Development

Initial data sets (pig37 and pig41) showcased evident activations, rendering Luman Wang's methodology adequate. After having looked at other implementations, we then turned our focus on implementing the algorithm from Bull et al.'s paper, which, though comprehensive, lacked concrete coding instructions, necessitating our input

3.4.Phase 4: Evaluation and Refinements

Upon crafting our algorithm's initial version, rigorous testing ensued. Challenges like manual parameter setting for each channel and filtering issues emerged. Challenges in the refinement of the filter algorithm and dynamic allocation of arrays are solvable but require more time.

3.5. Phase 5: Showcasing Results

Post data analysis, we developed an interactive application for our demonstration, visualizing algorithmic processes. Tools included Python scripts for animation generation.

4. Documentation

4.1.- **Report Framework**

- final_report_ydan124_p4p (1).pdf

4.2.- **Detailed Compendium Report**

- Compendium.pdf

4.3.- **Exposition Day Showcase**

- main.py: Animation creation via Python

- *.mat: MATLAB data sets visualizing algorithmic stages

- README.txt: Script operation guide

4.4.- **Research Data (From Auckland Bioengineering Institute)**

- *.mat: Data sets for algorithm evaluation

4.5.- **Detailed Analysis**

- *.Algorith results.xlsx contain observed results

-*.Result condition.docx contain condition for result

4.6.- **Embedded Implementation **

- main.c functions for algorithm testing

- *.m and *.mat: MATLAB scripts and datasets for data extraction

- README.txt: Guide to run it

4.7.- **Mid-Term Presentation**

- Mid Year Presentation (1).pptx and midyearprogress.mp4: Slides and video presentation

4.8.- **Display Materials**

-ydan124_poster.pdf: Informational posters

5. Challenges and Reflection

Throughout our journey on this project, several challenges materialized, testing our team's tenacity and adaptability.

1. **Technological Challenges**: The process of translating MATLAB code to C, suitable for the De1-SoC processor, was no small feat. While MATLAB offers a plethora of built-in functions and high-level abstractions, working with C demanded a more intricate approach, with attention to memory management, data types, and algorithmic efficiency.

2. **Algorithmic Challenges**: We encountered several challenges in crafting our initial version of the algorithm. Some included setting parameters manually for each channel and dealing with filtering issues. It became apparent that while we had a strong theoretical basis, the practical implementation posed unforeseen complexities.

3. **Machine Learning Challenges**: While ML presented itself as a promising alternative, it introduced its set of challenges. Some models' "black-box" nature made them unsuitable for safety-critical applications like pacemakers. Additionally, the real-time feasibility of these models was another concern, given the computational demands.

4. **Data Consistency**: Our initial tests with rudimentary ML models for binary classification presented inconsistencies in the data. This compelled us to re-evaluate our methodologies and data sources.

Reflecting on these challenges, our team underwent a profound learning curve. We had to continuously adapt and iterate our methodologies, delve deeper into literature, and collaboratively solve the problems that arose. The interdisciplinary nature of the project meant that we often found ourselves at the intersection of electrical engineering, software development, and biomedical science, each bringing its unique set of challenges.

6. Conclusion

The 'Pacemakers for Gastrointestinal Diseases' project was a journey filled with complexities, learning experiences, and growth opportunities. Through our concerted efforts, we were able to delve into unexplored territories, amalgamating knowledge from different domains, and driving forth innovative solutions for gastric pacemakers. We are optimistic about the potential advancements this research can lead to, especially in the realm of real-time responsiveness in medical devices. It has been a rewarding endeavor, and our team is grateful for the guidance, resources, and insights provided by Dr. Avinash Malik.

Contribution

Code:-

- Filter (individually)
- Artifact Remove (individually)
- Artifact Detect (individually)
- Neo Filter (individually)
- Moving Average (individually)
- Edge detection (individually)
- Artifact Detect (individually)
- main.c (individually)
- Python wrapper (individually)

- GNU plot in c (individually)
- Poster(Collaboration with Glacer)
- Mid Seminar Presentation Slides (Collaboration with Glacer)
- Results (Collabaration with Glacer)