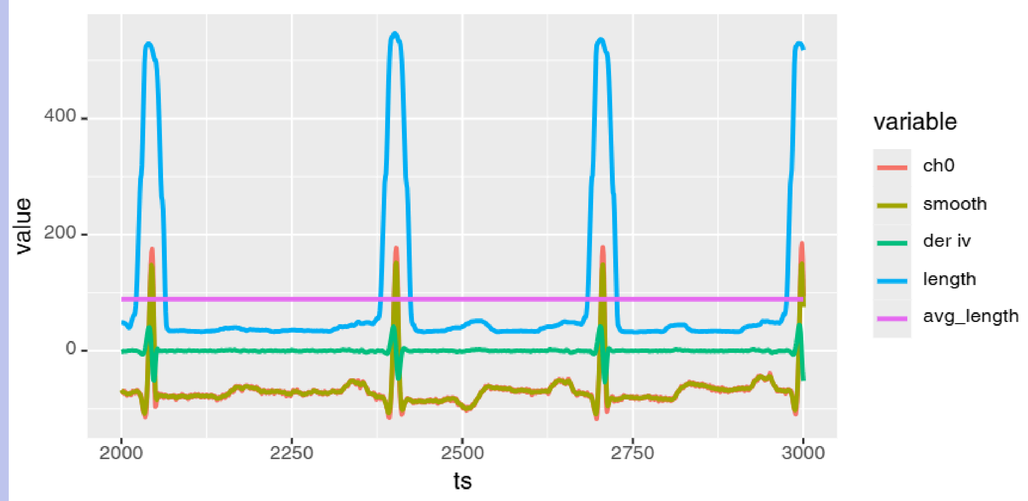
Streamlining HRV Input Using a Domain Specific Language

Laiba Effendi, [1,3] Konstantinos Mamouras [2]
[1] Kempner High School, Sugar Land Tx
[2] Rice University, Houston Tx
[3] Gifted and Talented Mentorship Program, Fort Bend ISD

INTRODUCTION:

Heart rate variability (HRV) is a key metric for understanding autonomic nervous system health. However, real-time analysis of HRV on wearable devices is challenging due to signal noise, motion artifacts, and limited processing power. Traditional filters like wavelets or Kalman filtering are powerful but too computationally heavy for low-power IoT devices. This project investigates whether a lightweight domain-specific language (DSL), built in Rust, can improve the speed and accuracy of HRV analysis by using adaptive signal processing techniques tailored to constrained environments.

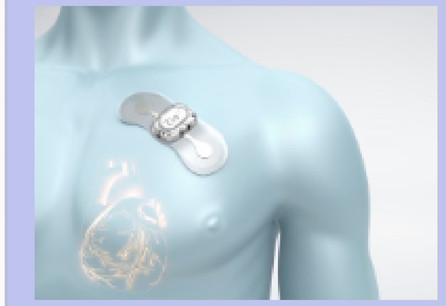


RESULTS:

To analyze heart rate variability (HRV), we identified R-peaks in the ECG signal and calculated the time intervals between them using a sampling rate of 360 Hz. The mean R-R interval was 0.811 seconds, corresponding to an average heart rate of approximately 74 beats per minute. The standard deviation of R-R intervals—used as a basic measure of HRV—was 0.048 seconds, indicating a moderate level of beat-to-beat variability. By applying the derivative to the signal, we enhanced the distinction between peaks and surrounding noise, allowing for more precise detection of R-wave peaks. This approach enabled us to reliably extract intervals from a large dataset of over 1,500 peak points, which was essential for accurate HRV assessment.

METHODOLOGY:

To develop a domain-specific language (DSL) for real-time heart rate variability (HRV) monitoring on IoT devices, I followed a structured process of designing, implementing, and evaluating the system. I began by defining core features like adaptive filtering, sliding window analysis, and HRV metric extraction, ensuring the DSL could process continuous physiological data with minimal delay. I used Rust to build it due to its low memory overhead and efficiency, which are ideal for low-power wearables. After designing the language, I implemented a lightweight interpreter that uses recursive least squares (RLS) filtering to adapt to real-time signal noise. To evaluate performance, I tested the DSL on physiological datasets and wearable sensor data, measuring speed, accuracy, and resource use. After confirming strong results, I deployed it on IoT-based monitoring systems to assess real-world performance and made refinements for better reliability and efficiency.



The goal of this research was to develop a system that enables efficient, low-latency filtering and feature extraction while remaining small and fast enough to operate on low-power IoT devices like the wearable monitors above.

CONCLUSION:

A domain-specific approach can improve the accuracy and efficiency of real-time HRV monitoring on wearable devices. By combining adaptive filtering, sliding window analysis, and Rust's performance, the DSL offers a reliable and scalable tool for health tech. In the future, this DSL could be expanded to include additional biosignals or be deployed on various IoT platforms to help make continuous health monitoring more accessible and responsive.