JY61 IMU SENSOR EXTERNAL VALIDITY: A FRAMEWORK FOR ADVANCED PEDOMETER ALGORITHM PERSONALISATION

Boris Bačić[1,2], Chengwei Feng[1] and Weihua Li[1]

Engineering, Computer and Mathematical Sciences, Auckland University of Technology (AUT), New Zealand[1]
Sports Performance Research Institute New Zealand (SPRINZ), Auckland University of Technology (AUT), New Zealand[2]

This case study aims to compare a low-cost inertial sensor prototype (JY61 IMU + ESP8266 MCU designed for real-time non-proprietary data streaming over Wi-Fi) with a high-end mobile (iPhone 13 Pro Max) using Matlab pedometer algorithm. Preliminary experiments used data collected at 100 Hz including 20, 50, and 100 steps on a partially carpeted and partially hard floor surface with the test subject wearing high heels and tennis shoes. The error comparison between the sensor (0–1%) and high-end mobile (0–2%) suggests that the sensor can be used as a privacy-preserving pedometer which is smaller, lighter and a low-cost alternative to mobile sensors. The experimental framework developed for advanced pedometer personalisation is applicable in education, gate pattern analysis and other sensor advancements for sport equipment and wearable technology applications.

KEYWORDS: wearables, sensors, Inertial Measurement Unit (IMU), Internet of Things (IoT), Human Activity Recognition (HAR), Human Motion Modelling and Analysis (HMMA).

INTRODUCTION: Advancements in 5G data communication progressing towards 6G technology and scalable big-data analytical healthcare platforms are aimed towards the next generation of IoT technologies integrating real-time data streaming from biomedical devices, mobiles, wearables, and inertial sensors (Štufi et al, 2020, Feng et al, 2023). Although wearable technology for sport performance, health, and activity monitoring is also improving and becoming increasingly accessible and convenient, there are still limitations and challenges associated with proprietary technologies including mobile sensors and applications. For the biomechanics community and sports practitioners, some limitations on generating new knowledge from raw data can be associated with the need for privacy preservation, and non-proprietary technology integration control with access to raw data and data streaming. As part of incremental research and development of the non-proprietary, low-cost (< $25), inertial sensor prototype (JY61 IMU with ESP8266 MCU), this case study aims to: (1) answer "can the sensor prototype be used as a pedometer?"; (2) produce an experimental framework to advance a decade old Matlab (R2014a – R2023b) open-source pedometer algorithm with additional functions and parameters that can be personalised; and (3) provide recommendations to aid the research and development of the latest generation of inertial sensors capable of high-rate data streaming tested on 100 Hz (10 ms sampling intervals).

METHODS: As part of cyclic and incremental design prototyping, the evaluation of the inertial sensor here is in the context of pedometer use for activity monitoring. The methodology involves boundary testing to: (1) compare errors between the sensor prototype and a high-end mobile device; (2) identify potential issues associated with the need for pedometer technology personalisation; and (3) provide recommendations from the experimental work conducted. The Matlab pedometer algorithm (MathWorks, n.d.) was chosen as it was designed to work on general population with low computational cost and on a range of older mobile sensors. For identified boundary test cases, algorithm improvements criteria included the same experimental settings for both devices, including the sampling rate of 100 Hz.
For the experimental data collection protocol, the authors used a surface area with carpet and hard floor, where the participant (co-author, 158 cm, 42 Kg), walked 20, 50, and 100 steps wearing high heels and then tennis shoes. The sensor prototype was attached to the mobile

using two-sided adhesive tape and held in the pocket (Figure 1). Throughout the data collection process, the sensor prototype and a mobile (iPhone 13 Pro Max) were used simultaneously to record acceleration data in the XYZ axes. The acceleration data from both devices were imported into Matlab for further data analysis.

Mobile: iPhone 13 Pro Max
Size: 78.1 mm x 160.8 mm x 7.65 mm
Weight: 240 grams

Added weight to the mobile (sensor + battery weight): ~ **29 grams**
Total weight (mobile + sensor + battery + adhesive): ~ **280 grams**

Inertial sensor prototype (version 2):
JY61 IMU module with ESP8266 Microcontroller
JY61 (accelerometer range: ±16 g, angular velocity: ±2000°/s)
Size: 29.3 mm x 34.6 mm x 15 mm
Weight: 6.7 grams
Battery (operation autonomy min. 10 hours): 3.7 V, 1200 mAh
Size: 54 mm x 34 mm x 5.7 mm
Weight: 22 grams
Sensor + Battery size: 54 mm x 34 mm x 22 mm

Figure 1: Experimental settings and equipment.

The produced framework (Figure 2) shows a general approach to a single iteration of source code analysis with visualisation of intermediate results in support of data-driven optimisation and evaluation of advanced features. Each iteration involved verification, performance logging, and analysis of enabled/disabled changes in parameter settings. Hence, the process involved multiple programme iterations, recording parameters and functions to be optimised for: (1) maximising algorithm performance (i.e. pedometer accuracy, while minimising errors); and (2) real-time accelerometer data stream processing (e.g. by prioritising low-computational costs). Note that the framework is also designed to be applicable to follow-up research contexts as well as with minimal changes, to related contexts involving high-frame rates signal processing desired for sport performance, rehabilitation and equipment research and development.

RESULTS: Comparing the original vs. advanced algorithms' performance on Table 1, there are noticeable similarities between the sensor (IMU) and mobile (Mob.) data sources.

Table 1: Pedometer algorithm comparisons on mobile and sensor acceleration data (100 Hz): Original (O) vs Advanced (A) algorithm settings using conceptual framework (Figure 2).

Original or **A**dv*). Settings	Shoes: **T**ennis or **H**igh heels	Test Steps	Steps Detected via IMU	Steps Diffe-rence	IMU Error	Steps Detected via Mob.	Steps Diffe-rence	Mob. Error
O	T	20	43	23	115%	53	33	165%
O	H	20	44	24	120%	45	25	125%
O	T	50	87	37	74%	85	35	70%
O	H	50	131	81	162%	128	78	156%
O	T	100	153	53	53%	180	80	80%
O	H	100	256	156	156%	280	180	180%
A	T, H	20, 50	20, 50	0	0%	20, 50	0	0%
A	T, H	20, 50	20, 50	0	0%	20, 50	0	0%
A	T, H	20, 50	20, 50	0	0%	20, 50	0	0%
A	T	100	100	0	0%	98	-2	-2%**)
A	H	100	101	1	1%	100	0	0%

*) **A**dvanced algorithm settings: 0.9 Standard Deviations (STD); Max. 4 steps/second.
**) If applied additionally, 5-point 'moving mean' data smoothing the error was -1%.

Figure 2(a) depicts the processing data flow diagram, combining the original algorithm (MathWorks, n.d.) highlighted in orange. The hexagonal shapes indicate advanced aspects with individual empirical evaluations as optional processing steps. Figure 2(b) shows a visual comparison of intermediate processing steps for the sensor prototype and mobile data with 'Step Threshold' as adaptive decision boundary supporting step count modelling and analysis. Note that the open-source MatWorks algorithm with its code is also accessible via the Matlab command: `openExample('matlabmobile/CountingStepsByCapturingAccelerationDataExample')`.
Applying the same common settings optimised for all experiments and real-time computing demands had a similar effect on both data sources. Figure 2(b) shows that the sensor and mobile XYZ orientations were not aligned, but the magnitude and signal patterns were similar, and both showed a recurring gait pattern revealing a personal movement 'signature'. As expected, for the original pedometer algorithm, wearing high heels showed the highest acceleration peaks (including potential outliers) and error rates compared to soft, rubberised tennis shoes (Table 1). Hence, the high-end iPhone and sensor prototype can both be used as a 100 Hz pedometer, relying on the advanced variation of Matlab algorithm with personalised settings.

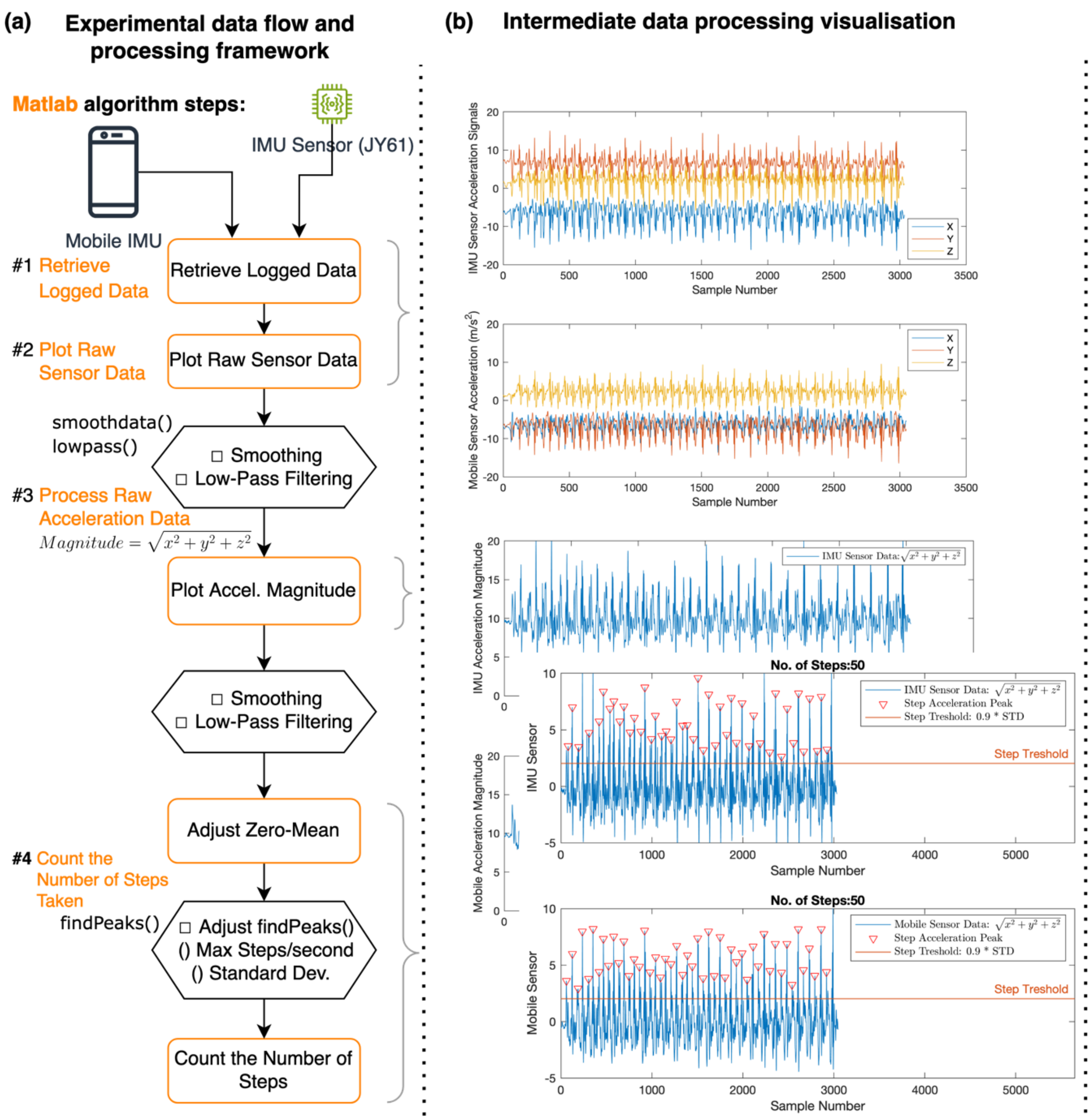

Figure 2: Conceptual framework for pedometer algorithm improvements and visual modelling.

DISCUSSION: Regarding technology integration aspects, the acceleration data from the Matlab mobile app (in MAT format) had to be recorded in MathWorks' cloud, so the use of offshore third parties/proprietary cloud data collection and sharing of personal information was unavoidable. The raw motion data detected from IMU, however, were transferred in real time (and in human-readable CSV text format) to the laptop via Wi-Fi range using a multi-platform application developed in Java, intended to be improved and shared as open-source.
The IMU sensor and software design for real-time data streaming over Wi-Fi is inspired by the Arduino open-source initiative (https://www.arduino.cc), which involves the sharing of pre-made electronic devices and wiring diagrams with source code demos that are conjointly intended for hobbyists, IoT enthusiasts and industry professionals seeking to accelerate the pace of science, research, and technological advancements (Cameron, 2020). After evaluating the sensor prototype and its software, the next two sensor versions were made-to-order for production on printed circuit board (PCB).
While both mobile and IMU sensor achieved similar error rates (sensor: 0-1% and mobile: 0-2%), the 'fit-for-purpose' Arduino design philosophy as open-source, low-cost, 'good-enough', rapid prototyping are favoured for privacy preservation and avoiding dependence on proprietary data transfer via third party cloud services. The experimental results indicate that the major algorithm improvement was not in data smoothing and filtering, but in adding the parameter of maximum number of steps per second to Matlab `findpeaks()` function and in optimising the standard deviation threshold (for personalisation and fine tuning the ratio of false positives and false negatives), which are also computationally less demanding for the intended applications of the sensor design for real-time data streaming. Hence, although data filtering and smoothing were beneficial during the parameter optimisation, it was not needed for the final algorithm solution. With a view to advancing the original Matlab pedometer algorithm for step counting that is inclusive of the past decade's mobile technology (based on 15–30 Hz sapling), higher sampling rates are expected to create new research opportunities for quantifying movements and pattern discovery, including for locomotor activities e.g. pathomechanics and gait parameters estimation, real-time human motion modelling and analysis (HMMA), and anomaly detection associated with rehabilitation.

CONCLUSION: The low-cost inertial sensor prototype (JY61 IMU with ESP8266 MCU, version 2), enabling real-time data streaming (100 samples/second, in human-readable CSV format) to a personal computer over Wi-Fi, can be used as a substitute for a high-end mobile phone (iPhone 13 Pro Max) for activity monitoring and advanced personalised pedometer use.
The produced advanced pedometer algorithm with its framework (Figure 2) for visual modelling and data analysis can aid parameter optimisation to accelerate scientific progress, facilitate learning, and, with minor modification, be transferrable to sport sensors, rehabilitation, and wearable equipment design contexts. For biomechanists and healthcare professionals, transparent design approaches empower end-users to keep control over their data, preserving privacy and security, by using non-proprietary and open-source technology integration.

REFERENCES

Cameron, N. (2020). *Electronics projects with the ESP8266 and ESP32: Building web pages, applications, and WiFi enabled devices*. Apress. ISBN-10:1484263359.
Feng, C., Bačić, B., & Li, W. (2023, 20 Nov.). Transforming healthcare with IoT streaming: Towards open-source IMU sensors for activity monitoring. *The Digital Pulse: Transforming Healthcare - NZ IoT Alliance*. (2023, November 20). New Zealand IoT Alliance. https://iotalliance.org.nz/event/the-digital-pulse-transforming-healthcare/
MathWorks. (n.d.). *Counting steps by capturing acceleration data from your mobile device*. https://au.mathworks.com/help/matlabmobile/ug/counting-steps-by-capturing-acceleration-data.html
Štufi, M., Bačić, B., & Stoimenov, L. (2020). Big data analytics and processing platform in Czech Republic healthcare. *Applied Sciences,* 10 (5), 1705. https://doi.org/10.3390/app10051705.