CHALMERS

Signal Processing Techniques for Step Counting, Activity Classification, and Distance Measurement Using a Single IMU

Bachelor's thesis in Electrical Engineering

GUSTAV CARLSSON, LUKAS SILLEN,
PHILIP STRANDLYCKE, DAVID TÖRNKVIST,
ALEXANDER WIKLUND, VENDELA WÄNESKOG

Department of Electrical Engineering
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2025
www.chalmers.se

Bachelor's thesis 2025

Signal Processing Techniques for Step Counting, Activity Classification, and Distance Measurement Using a Single IMU

Gustav Carlsson, Lukas Sillen, Philip Strandlycke, David Törnkvist, Alexander Wiklund, Vendela Wäneskog

Department of Electrical Engineering
E2
EENX16-VT25-38
Chalmers University of Technology
Gothenburg, Sweden 2025

Signal Processing Techniques for Step Counting, Activity Classification, and Distance Measurement Using a Single IMU
Gustav Carlsson, Lukas Sillen, Philip Strandlycke, David Törnkvist, Alexander Wiklund, Vendela Wäneskog

Supervisor: Elina Amani, Department of Electrical Engineering
Examiner: Fredrik Brännström, Department of Electrical Engineering

Bachelor's Thesis 2025
Department of Electrical Engineering
E2
EENX16-VT25-38
Chalmers University of Technology
SE-412 96 Gothenburg
Telephone +46 31 772 1000

Cover: An image visualizing movement measurement using a foot placed sensor. (Created via Chat GPT with the prompt *"Make a front page image for a bachelor's thesis about making a algorithm with signal processing with data from an IMU that calculates steps taken, activity level, and distance traveled. It should be a more technical design, should not include text or title, not any preferred color scheme"* and edited by Vendela Wäneskog, 2025)

Typeset in LaTeX
Gothenburg, Sweden 2025

Signal Processing Techniques for Step Counting, Activity Classification, and Distance Measurement Using a Single IMU
Gustav Carlsson, Lukas Sillen, Philip Strandlycke, David Törnkvist, Alexander Wiklund, Vendela Wäneskog
Department of Electrical Engineering
Chalmers University of Technology

Abstract

The aim of this thesis was to develop and validate an offline method for monitoring human walking and running using a single, low-cost Inertial Measurement Unit (IMU). We designed signal-processing algorithms to count steps, estimate distance and speed, and classify activity level. All from the 3-axis accelerometer and gyroscope data. Raw signals were filtered with low-pass Finite Impulse Response (FIR) filter, zero-velocity updates were applied at each detected gait event and dominant stride frequencies were extracted via Fast Fourier Transform (FFT) over sliding windows. In test with five subject on a 75 m straight path, step-count accuracy averaged 98% for walking and 94% for running. The distance estimates reached 96% accuracy in walking and 91% in running. Activity classification achieved 100% accuracy in controlled trials and 89% in mixed scenarios.

Sammandrag

Målet med projektet var att utveckla och evaluera en offline metod för att mäta gång och löpning hos en människa, endast med användningen av en låg-kostnads Inertial Measurment Unit (IMU). Vi designade signal processerings algoritmer som stödjer att räkna steg, klassificera aktivitets nivå, uppskatta distans och hastighet. Detta gjordes endast från accelerometer- och gyroskop data. Obearbetade signaler filtrerades med ett lågpass Finite Impulse Response (FIR) filter, noll-hastighets uppdateringar applicerades vid varje rörelse moment och den dominanta gång-frekvenserna var extraherade via en Fast Fourier Transform (FFT) över en rörande-fönster algoritm. I ett test med 5 deltagare, där varje deltagare gick och sprang på en 75m rak väg, mätte stegräkningen en noggrannhet på 98% för gång och 94% för språng. Distans estimeringen nådde 96% noggrannhet för gång och 91% för sprint. Klassificeringen av aktiviteter uppnådde 100% noggrannhet i kontrollerade försök samt 89% i blandade försök.

Keywords: Inertial Measurement Unit (IMU), signal processing, gait analysis, distance estimation, activity classification, step counting, Zero-Velocity Update (ZUPT).

Nomenclature

Below is the nomenclature of indices, sets, parameters, and variables that have been used throughout this thesis.

Indices

k	Index for frequency bins in the FFT
m	Index for detected movement segments
n	Index for signal samples within a time window
t	Index for timestamps in the signal

Sets

M	Set of movement segments (periods classified as active movement)
N	Set of number of samples

Parameters

$f_{min,walk}$	Minimum stride frequency threshold for walking (0.65 Hz)
f_s	Sampling frequency (Hz), fixed at 174 Hz
g	Acceleration due to gravity (9.82 m/s^2)

Variables

A_a	Activity classification accuracy
A_d	Distance estimation accuracy
A_s	Step counting accuracy
D_a	Actual distance traveled

D_m	Measured distance from algorithm
S_a	Actual step count (manually recorded)
S_m	Measured step count (algorithm output)
$a_{forward}$	Total acceleration forward in global frame
$a_{x,forward}$	Acceleration from $a_{x,local}$ that is in X-axis direction in the body-fixed coordinate system
$a_{x,local}$	Raw acceleration in X-axis direction in local IMU frame
$a_{y,forward}$	Acceleration from $a_{y,local}$ that is in Y-axis direction in the body-fixed coordinate system
$a_{y,local}$	Raw acceleration in Y-axis direction in local IMU frame
$a_{z,forward}$	Acceleration from $a_{z,local}$ that is in Z-axis direction in the body-fixed coordinate system
$a_{z,local}$	Raw acceleration in Z-axis direction in local IMU frame
d_{total}	Total estimated distance traveled
$f_{dominant}$	Dominant frequency of motion in a window or segment
pitch	Angular displacements used for orientation and gravity compensation on the y-axis
roll	Angular displacements used for orientation and gravity compensation on the x-axis
yaw	Angular displacements used for orientation and gravity compensation on the z-axis
s	Slope of linear velocity drift correction within a step
t_{end}	End time of a detected step
t_{start}	Start time of a detected step
$v_{forward}$	Estimated velocity forward in the body-fixed coordinate system
$x[n]$	Time-domain signal in a window
$x_{zp}[n]$	Zero-padded signal used in FFT
$\|X[k]\|$	Frequency-domain magnitude of signal via FFT

Contents

List of Figures

List of Tables

1

Introduction

Movement tracking is a widely utilized tool in health monitoring due to its broad range of applications. One of the earliest implementations of movement tracking was step counting, which was achieved using a pedometer. Early pedometers operated through mechanical components that recorded steps by detecting the rhythmic nature of walking. With technological advancements, these mechanical parts have been largely replaced by electronic sensors. This transition has enabled more accurate measurements and a broader spectrum of activity classification. For instance, modern movement trackers can measure parameters such as velocity, workout intensity, and elevation.

An additional consequence of the shift from mechanical to electronic systems is the significant reduction in sensor size. This miniaturization has facilitated the integration of movement sensors into everyday devices such as smartphones and smartwatches. The incorporation of such sensors into personal devices has given rise to the trend of "Wearable Technology," which continuously monitors activity levels throughout the day [1]. The ongoing popularity of this trend in the fitness industry reflects a sustained global interest in personal movement tracking solutions.

The worldwide adoption of movement tracking is driven by its relevance across multiple disciplines. A primary example is its use in medical applications. One study analyzed the gait of Parkinsons disease patients to predict the onset of episodes [2]. Another significant application lies in the field of athletics, where movement trackingand specifically gait analysisis employed to assist athletes in recovering from injuries. Both medical and athletic domains often utilize custom-built movement sensors tailored to their specific requirements. These sensors, known as Inertial Measurement Units (IMUs), comprise multiple sensor types. A key advantage of IMUs is their capability for indoor use, as they do not rely on external connectivity.

A challenge encountered when acquiring a movement tracker is the associated cost. Typically, consumers purchase specialized smartwatches or fitness watches equipped with movement tracking capabilities. While these devices are generally affordable for the average consumer, specialized movement tracking equipment tends to be significantly more expensive and less accessible. This project aimed to investigate the extent of functionality that could be achieved using a single IMU in conjunction with signal processing techniques.

1.1 Purpose

This thesis will design, develop, and validate an offline signal-processing algorithm for a single, foot-mounted, low-cost IMU to estimate step count, distance, and speed during walking and running. It will implement a step-detection method based on accelerometer and gyroscope signals, then compare its counts to manual tallies. From those detected gait events it will derive travelled distance and instantaneous speed without relying on magnetometers, barometers, or GPS.

To ensure reliable long-term monitoring, the work will introduce noise-reduction filters and drift-compensation schemes, quantify overall accuracy under varied conditions, identify key limitations, and suggest targeted improvements.

1.2 Limitations

This study has several limitations that define the scope of the project and its expected accuracy. One key limitation is to keep the device affordable, the system uses a single low-cost IMU sensor. This limits available measurements to those provided by a 3-axis accelerometer and 3-axis gyroscope, and omits a magnetometer entirely. Although multiple IMUs could improve accuracy, particularly in analyzing step symmetry and gait patterns, they would increase the system's complexity and cost. By limiting the setup to one sensor, the project remains within its intended affordability scope while still enabling meaningful activity recognition.

Another limitation is that the system relies on offline data processing rather than real-time monitoring. Although real-time tracking could provide immediate feedback, offline analysis allows for a more detailed examination and refinement of activity classification algorithms without requiring continuous data transmission or high computational power on a wearable device.

The system also does not measure elevation differences, such as stair climbing or hill gradients, due to the inherent challenges of IMU-based altitude estimation. Since IMUs are based on accelerometers, they are prone to drift and noise over time, making precise vertical displacement tracking difficult. Alternative solutions, such as barometers or GPS, would be needed for reliable elevation tracking, but these fall outside the project's scope.

Lastly, the system estimates distance and pace based on step count and an assumed or user-defined step length. Since an IMU does not provide direct spatial positioning like GPS, variations in stride due to terrain may affect the accuracy of distance and pace calculations. More sophisticated machine learning models could improve these estimations, but to maintain affordability and computational efficiency, this study focuses on a simpler approach.

By acknowledging these limitations, the study provides a realistic framework for evaluating the effectiveness of low-cost IMU-based activity monitoring while highlighting areas for potential future improvements.

2

Background

The background section provides the foundational knowledge necessary to understand the methodology and interpret the results, including the principles of IMU sensors, gait dynamics, and signal processing techniques.

2.1 The IMU sensor

An Inertial Measurement Unit (IMU) is a device that measures motion and orientation using a combination of sensors. Most IMUs contain at least two types of sensors, an accelerometer and a gyroscope. Some also include a magnetometer, providing additional orientation information. IMUs are typically categorized by the number of measurement axes they provide. A standard 6-axis IMU includes a three-axis accelerometer and a three-axis is referred to as a 6-degrees of freedom (DOF) IMU. A 9-axis IMU adds a three-axis magnetometer for enhanced orientation tracking.

Due to their small size, low cost, and versatility, IMUs play a key role in developing reliable and cost-effective monitoring systems, especially in fields like healthcare and motion tracking [3]. By combining data from these sensors, it is possible to estimate traveled distance, count steps, and perform activity classification. Unlike GPS-based systems, IMUs can operate indoors without relying on satellite signals, making them particularly useful in constrained or indoor environments.

2.1.1 Accelerometer

The accelerometer measures the linear acceleration acting on the device along three axes X, Y, and Z. By integrating this data over time, it is possible to estimate the device's velocity and approximate the distance traveled.

2.1.2 Gyroscope

The gyroscope measures the angular velocity of the device, how it is rotating, around three axes X, Y, and Z. By integrating this angular velocity, the change in orientation can be estimated, allowing the IMU to track the device's rotational movement.

2.1.3 Magnetometer

The magnetometer detects the magnetic fields surrounding the IMU, typically measuring the Earth's magnetic field unless influenced by nearby magnetic sources. This sensor helps improve orientation estimation by correcting for gyroscopic drift and stabilizing heading measurements.

2.1.4 Price classes of IMUs

IMUs vary significantly in price, ranging from around 100 SEK to nearly 1,000,000 SEK [4]. To help categorize this wide range of performance and cost, manufacturers often classify IMUs into four main performance grades: consumer-grade, industrial-grade, tactical-grade, and navigational-grade [5]. Consumer-grade IMUs are the least expensive and typically use Micro Electro Mechanical Systems (MEMS) technology [4]. These units generally range from approximately 100 SEK to just under 10,000 SEK, with more advanced MEMS IMUs are priced up to around 100,000 SEK. At the high end, navigational-grade IMUs are the most expensive that use Fiber Optic Gyroscope (FOG) IMUs priced between 100,000 SEK and 1,000,000 SEK. The most expensive IMUs are primarily found in high-precision applications such as aerospace systems (e.g., satellites and space shuttles) and military equipment.

2.1.5 Market standards

A common way to measure parameters such as counting steps and travelled distance is to use a wrist-wearable device. There are numerous devices on todays market with a range of functions and prices, utilizing different methods and hardware to collect the desired data. A common hardware component are accelerometers which can be found in IMUs and GPS. An example of an available device using accelerometers is the wrist wearable Garmin Vivofit [6], priced at about 700 kr in Sweden [7]. Another example is the Fitbit Zip (available on the secondary market), which is attached to your clothes, preferably at the waist [8].

A study showed that the mean error for the number of steps was for the Fitbit was 0.1% (with a standard deviation of +-0.6) and 3.9% for the Garmin (with a standard deviation of +-7.7) [9]. The latter study also showed that the distance mean error for the Fitbit was 4.1% (with a standard deviation of +-8.1) and 5.1% (with a standard deviation of +-11.4) for the Garmin. The study was done by letting participants walk on flat ground for 400 meters.

2.2 Frequency of human gait

Human gait is quasi periodic, showing slight variations in stride timing, stride length, posture and more whilst remaining a strong rhythmic structure [10]. Despite these natural fluctuations it exhibits a dominant frequency that remains fairly consistent [11]. Whilst walking the general person walks with a velocity of 1.27 - 1.35 m/s [12] and with a stride frequency between 0.9 - 1 Hz. Higher velocity results in higher stride frequency and the transition window of when people changes from walking to running happens between 1.13 - 1.23 Hz [13]. After the transition a normal running stride frequency is between 1.25 and 1.6 Hz [14] [15].

2.3 Sensor Fusion

Sensor fusion combines data from multiple sensors to produce more accurate and reliable estimates than using any single sensor alone [16]. In a 6-axis IMU, this typically involves integrating measurements from the accelerometer and gyroscope. By fusing their outputs, it is possible to estimate orientation and motion more accurately. Sensor fusion also helps mitigate individual sensor biases, reducing cumulative errors such as drift over time.

2.3.1 Bias and drift

IMU sensors often exhibit bias, which refers to a constant offset in their output. This means that a gyroscope may indicate angular velocity even when the device is stationary, and an accelerometer may detect acceleration without any actual movement [17]. Such biases can arise from sensor imperfections, temperature fluctuations, manufacturing inconsistencies, or mechanical stress on the MEMS sensor, particularly when mounted onto a circuit board or subjected to prolonged physical strain [18] . Over time, these small offsets can accumulate through integration, leading to significant errors known as drift. Because position and orientation estimates rely on the integration of acceleration and angular velocity, even small biases can cause significant inaccuracies. To mitigate the effects of bias, several techniques can be employed, including calibration, high-pass filtering, zero-velocity updates, and Kalman filter-based sensor fusion.

2.3.2 Global and local coordinates

When collecting data, the IMU measures it in a local coordinate system that is fixed relative to the device's own orientation. This is often referred to as a body-fixed coordinate system, typically centered at the devices center of mass [19]. The global coordinate system, by contrast, represents a fixed reference frame in the external environment, independent of the devices motion or orientation. In certain applications, the body to which the IMU is attached, such as a limb or shoe, can serve as a practical reference frame. To relate data between a global and local system, a transformation matrix can be used to convert coordinates from local to global frames, and vice versa.

2.4 Filters and signal processing methods

Raw sensor data from accelerometers and gyroscopes often contains noise arising from high-frequency electronic interference, mechanical vibrations, and other disturbances. To extract meaningful motion information, it is essential to apply filtering techniques that suppress this noise.

2.4.1 Finite Impulse Response filter

A finite impulse response (FIR) filter computes its output as a weighted sum of a finite number of input samples, with the weights defined by the filters coefficients. FIR filters do not utilize internal feedback, meaning that their output depends solely on present and past input samples, not on previous outputs. This structure imparts several desirable properties to the filter. Since the output is derived from a fixed number of input samples using finite-valued coefficients, the filter is inherently stable. Furthermore, it exhibits a fixed response duration, settling to zero precisely after N+1 samples, where N denotes the filter order.

A low-pass FIR filter can be designed using the least squares method, which determines the filter coefficients by minimizing the total squared error between the ideal and actual frequency responses across all frequencies. This approach effectively attenuates high-frequency noise while preserving the low-frequency components characteristic of human motion [20].

2.4.2 The ZUPT method

The ZUPT, zero-velocity update, is a method used to eliminate the drift of the accelerometer to ensure that errors produced from one step is not added to the next [21]. This is made possible by utilizing the fact that every time the foot hits the ground its velocity will at some point be zero and the errors can be removed.

3

Methods

The method consists of primarily signal processing methods that have been applied on raw sensor data. The signal processing methods are then collected in a file that we use to plot the data in the Graphical User Interface (GUI). To begin the data-collection setup is described.

3.1 Hardware

The IMU utilized in this project is the Seeed Studio XIAO nRF52840 Sense, which provides six degrees of freedom. This includes a 3-axis accelerometer with a sensitivity of up to ±16g and a 3-axis gyroscope configured with a sensitivity of ±2000 degrees per second (dps). The device was connected to a computer via a USB-C cable, which served both as a power source and a data transmission medium. The cost of the IMU was 179.76 SEK at the time of purchase, dated 14 February 2025.

Hardware communication was established using the Arduino IDE. Code was written in Arduino C and uploaded to the IMU. Once the device was programmed, Matlab was able to connect to the designated port and retrieve sample data.

3.2 Software

The visualization software was developed using Python, while data analysis was conducted in Matlab. The data analysis primarily involved signal processing and data formatting. The formatting process aimed to generate a '.mat' file compatible with the Python application. This '.mat' file could subsequently be visualized through a Python script.

The visualization script was built using a graphical library called PythonQt [22], resulting in an application widget. A key design focus for the widget was to make movement data accessible and comprehensible to the general public. An example of the graphical user interface (GUI) can be seen in Figure 3.1 and Figure 3.2.

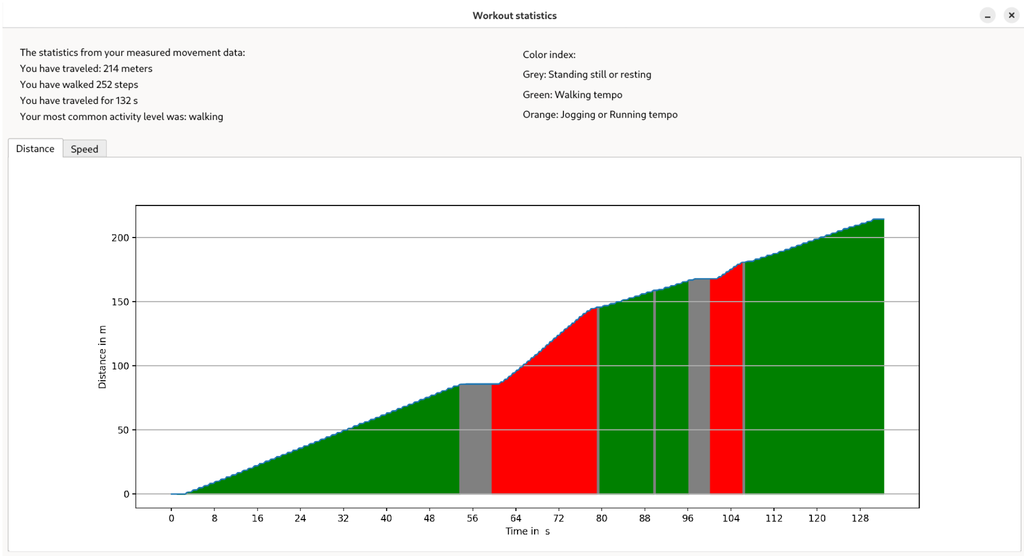

Figure 3.1: Image of the visualization software, in this instance it displays traveled distance over time. The color segments correspond to the measured activity levels. Additionally, movement statistics can be read from the upper right corner.

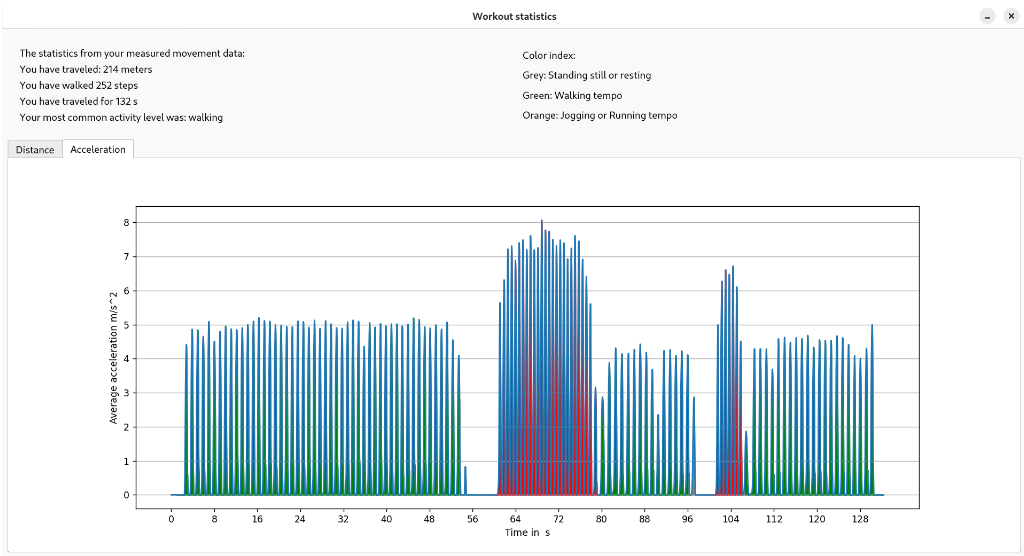

Figure 3.2: Image of the visualization software, in this instance it displays change in acceleration over time. The height and frequency of the graph indicate the intensity of the acceleration changes. The color segments correspond to the measured activity levels. Additionally, movement statistics can be read from the upper right corner.

3.3 Signal processing methods

The signal processing algorithms were developed in MATLAB, utilizing the "Sensor Fusion and Tracking Toolbox" and the "Signal Processing Toolbox" add-ons. These tools enabled the calculation of steps taken, distance traveled, and velocity, as well as the determination of activity level.

3.3.1 Step Counting

A first prototype of a step-counting algorithm was based on identifying individual steps using the ZUPT method. The developed step detection approach was founded on the analysis of three distinct motion signals: acceleration along the x- and z-axes, and pitch angular velocity. The method involved identifying peaks within each signal.

To account for alternating movement patterns, the z-axis acceleration and pitch angular velocity signals were inverted prior to peak detection. This inversion ensured the presence of peaks, rather than valleys, during the stance phase when the foot was in contact with the ground. The algorithm subsequently detected peaks near zero, as illustrated in Figure 3.3(a).

Step validation was performed through temporal alignment: peaks from all three signals were required to occur within a narrow time window to be recognized as a completed step. To prevent overcounting, a minimum time interval was enforced between consecutive steps.

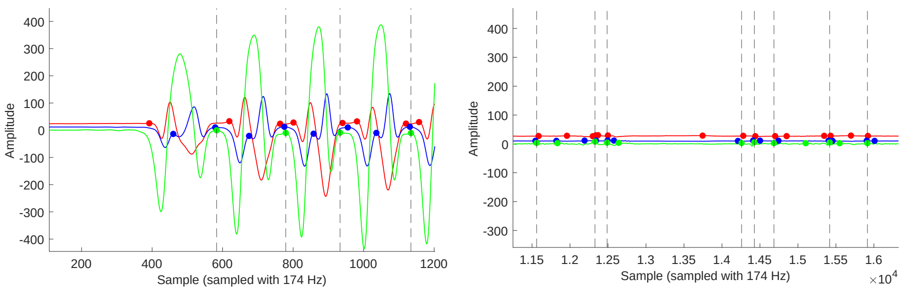

Figure 3.3: Two segments from one data set which shows how the first step counting algorithm counts steps. The left, (a), is when walking, the right, (b), is when standing still. The vertical dashed line is where the algorithm has found that a step is completed. The green line is the inverted pitch angular velocity. The red line is the inverted z-axis acceleration multiplied with 100. The blue line is the x-axis acceleration multiplied with 100. The dots represents where a peak near zero.

We observed the occurrence of falsely counted steps while the user was standing still, as illustrated in Figure 3.3(b). This issue arose due to minor fluctuations in the sensor data, caused by slight movements of the IMU or sensor noise, which produced small peaks in the signals. To mitigate false step detection during periods of inactivity, a second model was developed from a frequency-domain perspective. This approach was motivated by the understanding that human gait exhibits rhythmic characteristics, a pattern that is also evident in Figure 3.4, where each input signal demonstrates periodic motion.

By transforming these data signals into the frequency domain using a Fast Fourier

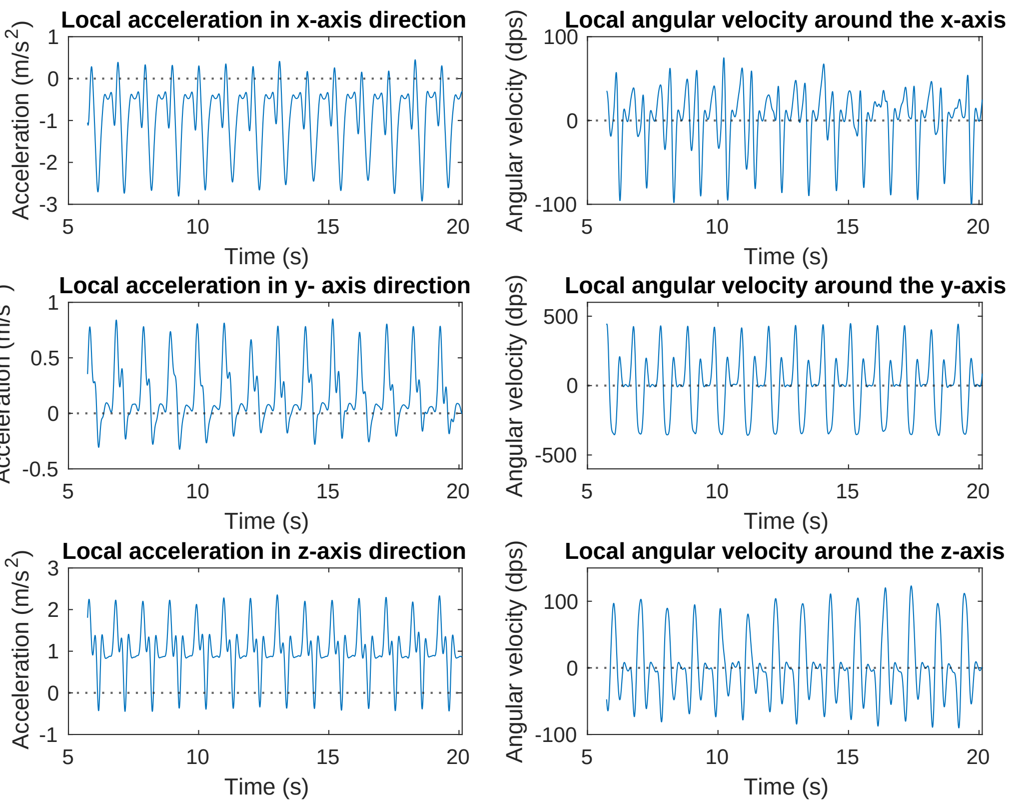

Figure 3.4: Raw (lowpass filtered) data from both the accelerometer and the gyroscope, during 14 steps. IMU placed on top of the foot.

Transform (FFT) and plotting the results, it becomes evident in Figure 3.5 that distinct peaks appear at specific frequencies across all signals. Consequently, it is unnecessary to detect and count individual steps based on signal peaks, which are susceptible to noise, bias, and false detections. Instead, the dominant frequency can be multiplied by the duration of activity to obtain a relatively accurate estimate of the step count.

The step detection algorithm initially focuses on determining whether the individual is actively mowing or stationary. To achieve this, the algorithm applies a sliding time window to the local acceleration data along the x-axis. As previously mentioned, the typical walking stride frequency ranges between 0.9 and 1 Hz, as evidenced by the leftmost peak in the acceleration data shown in Figure 3.5. To accommodate slower-than-average walking frequencies, the window size must be sufficient to capture frequencies as low as 0.65 Hz, which corresponds to a minimum window duration of approximately 1.53 seconds. Therefore, the time window is conservatively set to 2 seconds.

Ideally, the sliding distance should match the duration of a single step. However, since step durations vary across individuals and activities, it cannot be fixed precisely. Instead, the sliding distance is set to be shorter than the average running step duration, which is approximately 0.7 seconds [14, 15], in order to closely approxi-

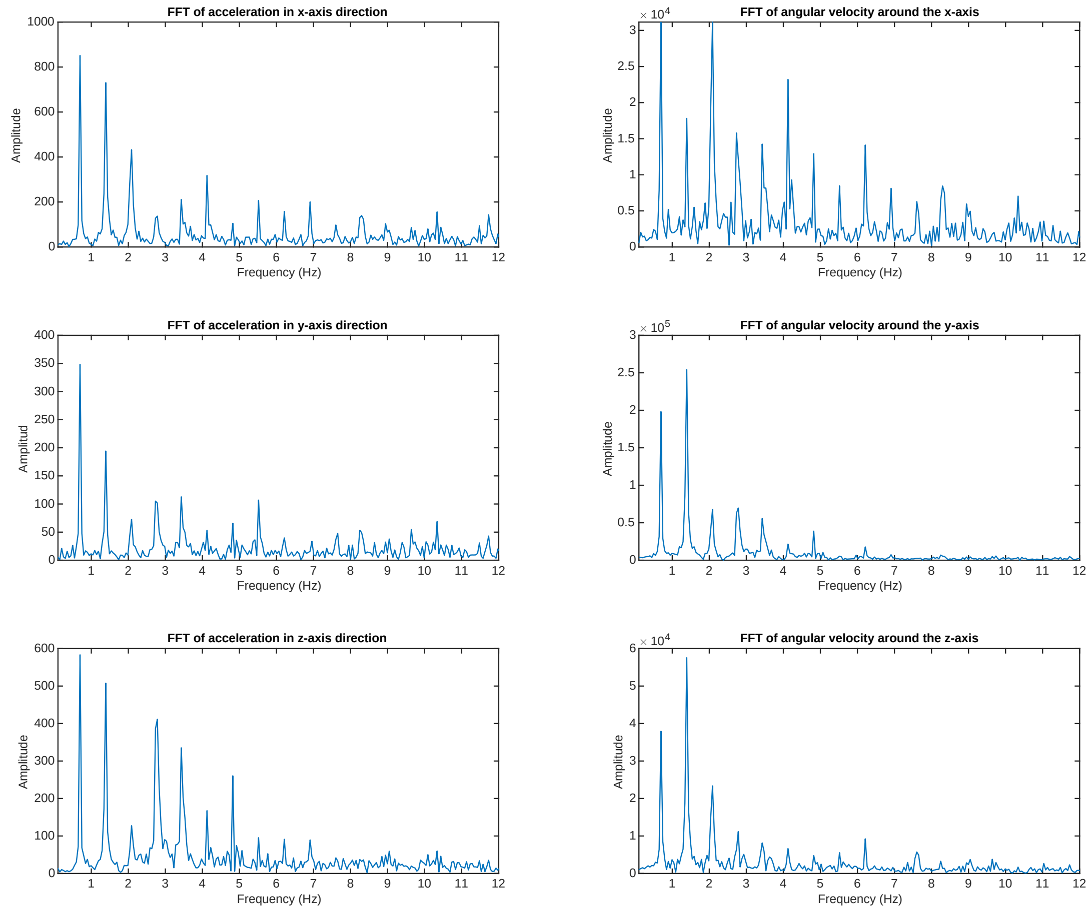

Figure 3.5: Fourier domain of data during 14 steps during walking, from both the accelerometer and the gyroscope. IMU placed on top of the foot.

mate the actual stop time. As a result, the sliding distance is set to 0.5 seconds, corresponding to 25% of the total window length.

An FFT is then applied to each time window to transform the time-domain acceleration signal into its frequency-domain magnitude representation, as described by the following equation

$$|X[k]| = \left| \sum_{n=0}^{N-1} x_{zp}[n] \cdot e^{-j\frac{2\pi kn}{N_{fft}}} \right|, \tag{3.1}$$

where $k = 0, 1, \ldots, N_{fft} - 1$, $N_{fft} = 500 \cdot N$, N is the number of samples in the time window, and $x_{zp}[n]$ is the zero-padded time-domain acceleration signal, defined as:

$$x_{zp}[n] = \begin{cases} x[n], & 0 \leq n < N \\ 0, & N \leq n < N_{fft} \end{cases}, \tag{3.2}$$

and used to enhance the frequency resolution of the FFT. The frequency corresponding to the k-th bin, denoted as $f[k]$, is computed as:

$$f[k] = \frac{k \cdot f_s}{N_{fft}}, \tag{3.3}$$

where $k = 0, 1, \ldots, N_{fft} - 1$ is the FFT bin index, and f_s represents the sampling frequency.

To determine whether the user is standing still or taking steps within each time window, the most dominant frequency, $f_{dominant}$, that exceeds $f_{min,walk} = 0.65$ Hz is identified. This leads to the following condition for detecting movement:

$$f_{dominant} > f_{min,walk}. \tag{3.4}$$

As illustrated in Figure 3.6(a), the most dominant frequency greater than $f_{min,walk}$ is 0.916 Hz, representing the stepping frequency of one foot. In contrast, Figure 3.6(b) shows $f_{dominant} = 0.753$, which still satisfies condition 3.4 despite the user being stationary.

To address such false positives, an additional condition is introduced by setting a minimum threshold for the magnitude of the dominant frequency:

$$|f_{dominant}| > 80, \tag{3.5}$$

where the threshold was determined experimentally to accommodate varying individuals and activity levels.

Therefore, for a time window to be classified as "moving", both conditions 3.4 and 3.5 must be satisfied.

To calculate the number of steps taken, as previously proposed, the dominant frequency is multiplied by the duration of movement. To facilitate this calculation, the algorithm segments the entire dataset into intervals classified as either "moving" or "not moving". If a time window is classified as moving while the preceding time window is not, the algorithm records the start time of that window as the beginning of a movement period. These points are indicated by dotted vertical lines in Figure 3.7. Conversely, if a time window is classified as not moving while the preceding window

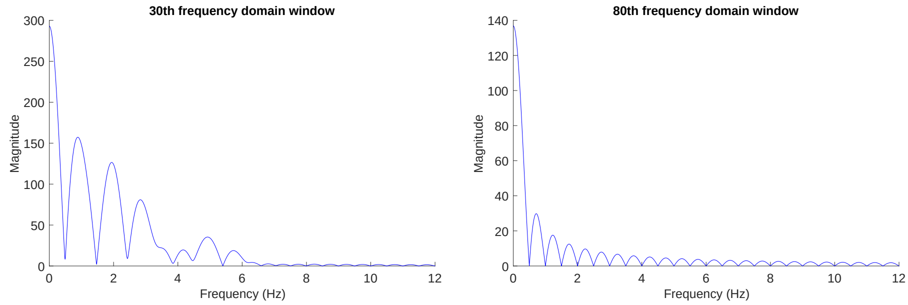

Figure 3.6: Frequency domain plots for a single time window. The left (a) shows the case where the individual is walking, while the right (b) represents the individual standing still.

was classified as moving, the algorithm records the start time of that window as the end of a movement period, represented by dashed vertical lines in Figure 3.7. Each interval between a start time and a corresponding end time is stored by the algorithm as a distinct moving segment.

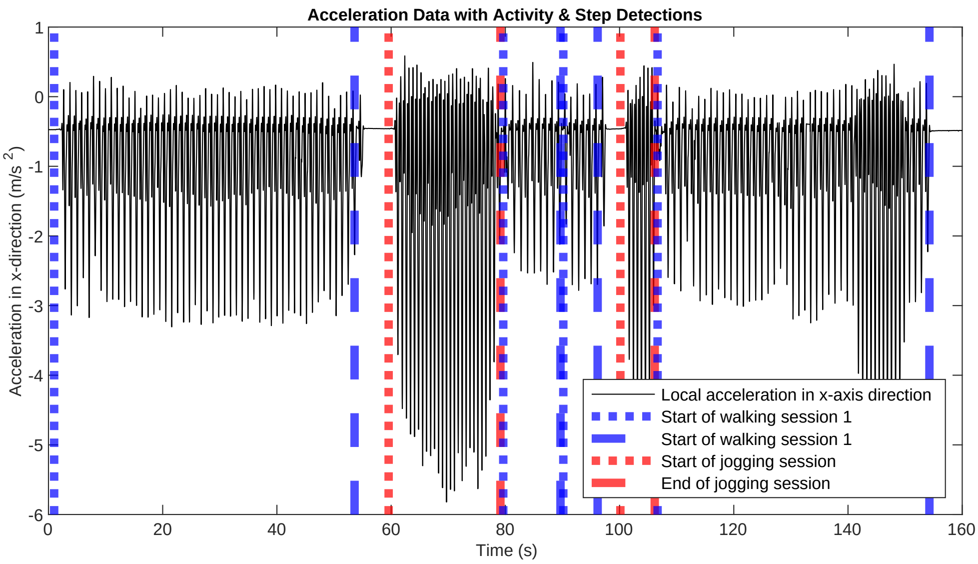

Figure 3.7: Plot of X-axis acceleration over time, with sections classified as walking, running, or standing still.

For each moving section, the algorithm calculates the most dominant frequency, $f_{dominant}[m]$, using the same method described in Equations 3.1, 3.2, and 3.3, where N in this case represents the number of samples within the entire section, rather than within a single time window. The value of $f_{dominant}[m]$ is then multiplied by the total duration of that section, $t_{tot}[m]$, as defined by the following equation

$$\text{Steps taken} = \text{round}\left(\sum_{m=1}^{M} f_{dominant}[m] \cdot t_{tot}[m]\right), \tag{3.6}$$

where M denotes the total number of detected moving sections.

Since this calculation accounts only for steps taken with one foot, the result must be multiplied by two to estimate the total number of steps taken with both feet, under the assumption that the user maintains a symmetric gait.

3.3.2 Activity level

To determine the user's activity level at each point in time, the algorithm utilizes the same moving sections and corresponding $f_{dominant}[m]$ values described previously in the step-counting algorithm. A threshold is then applied to distinguish between walking and jogging/running. If $f_{dominant}[m]$ falls below the threshold, the user is classified as walking during that section; if it exceeds the threshold, the user is classified as jogging or running.

The threshold is set to 1.2 Hz, based on findings in [13], which indicate that this frequency approximately corresponds to the transition point between walking and running. The classification can be formally expressed by the following function

$$\text{Activity level}[m] = \begin{cases} \text{walking}, & \text{if } f_{dominant}[m] < 1.2 \text{ Hz} \\ \text{running}, & \text{if } f_{dominant}[m] \geq 1.2 \text{ Hz} \end{cases} \tag{3.7}$$

Any time intervals that are not classified as either walking or running are labeled as "no-movement".

3.3.3 Distance and Velocity Measurement using sensor fusion

The distance and velocity algorithm was based on calculating the foots true forward acceleration using sensor fusion. This acceleration was integrated to obtain the velocity, which in turn was integrated to determine the traveled distance. The sensor fusion utilized IMU data across all six degrees of freedom, whose directions are illustrated in Figure 3.8, with the x-axis corresponding to the forward direction. Instead of directly using angular velocity in the calculations, the angular displacement of the IMU was computed by integrating the angular velocity data. Prior to any calculations, the raw data was filtered using a low-pass FIR filter.

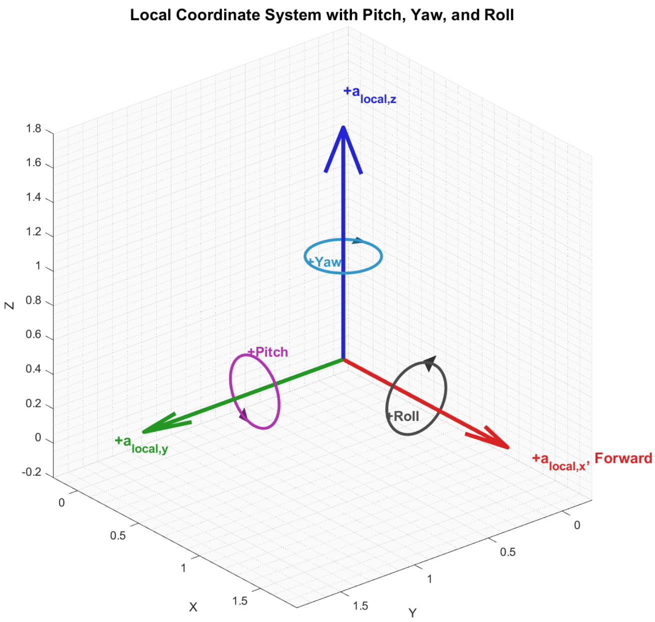

Figure 3.8: Visualization of the directions of positive x-, y- and z-axis accelerations, with the angular displacement, pitch, roll, and yaw marked with an arrow in the positive direction.

The signals exhibited drift that could not be easily modeled or filtered out, as there was no consistent mathematical trend, such as linear or polynomial, across the entire dataset. In addition to this drift, any inherent bias in the sensor data required a method to ensure accurate computation of velocity and distance. To address these issues, the first stage of the algorithm involved identifying the start and end points of each step. This step segmentation was essential for later applying the ZUPT method to both the velocity and angular displacement data. Step detection was performed using repetitive features in the data. These features were extracted by analyzing the acceleration in the x-axis direction, $a_{x,\mathrm{local}}$, as well as the derivatives of acceleration in the x- and z-axis directions, in combination with the derivative of the angular velocity around the y-axis.

Using these features, multiple potential start and end points for each step were identified. To ensure that only one start and one end were selected per step, additional conditions were implemented. Specifically, only one start or end point was allowed within 0.3 seconds of the previous point of the same type, with the first detected point being retained. To improve robustness and filter out falsely identified points, two additional conditions were applied. A time condition required that the duration of a step fall between 0.3 and 3 seconds. An order condition specified that a step begin must precede a step end, and that each step end must follow a valid step begin before a new step begin could be identified. A comparison between all detected points and the final start and end point for each step is shown on $a_{x,\mathrm{local}}$ in Figure 3.9.

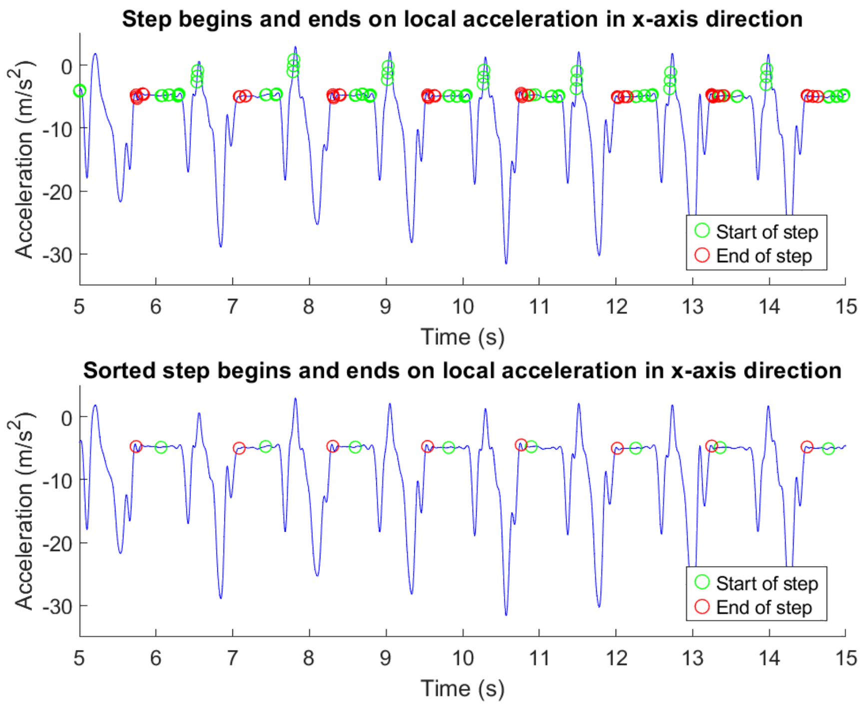

Figure 3.9: Comparison of step detection plotted on the local acceleration in the x-axis direction, (a) before sorting, upper plot, (b) after sorting, lower plot.

To calculate the foots true forward acceleration, $a_{forward}$, the acceleration data $a_{x,\text{local}}$, $a_{y,\text{local}}$, and $a_{z,\text{local}}$ needed to be compensated for the effect of gravity. This resulted in the adjusted accelerations, $a_{x,\text{real}}$, $a_{y,\text{real}}$, and $a_{z,\text{real}}$

$$
\begin{aligned}
a_{x,\text{ real}} &= a_{x,\text{ local}} - 9.82 \cdot \sin(\text{pitch}) \cdot \cos(\text{yaw}) \\
a_{y,\text{ real}} &= a_{y,\text{ local}} - 9.82 \cdot \sin(\text{roll}) \cdot \cos(\text{yaw}) \\
a_{z,\text{ real}} &= a_{z,\text{ local}} - 9.82 \cdot \cos(\text{pitch}) \cdot \cos(\text{roll}),
\end{aligned} \tag{3.8}
$$

where the gravitational constant 9.82 m/s^2 and the angular displacement of the IMU in terms of pitch, yaw, and roll were used.

Using the angular displacement, the contribution of the acceleration in each direction to the forward acceleration, $a_{x,\text{forward}}$, $a_{y,\text{forward}}$, and $a_{z,\text{forward}}$, was calculated by the following equations

$$
\begin{aligned}
a_{x,\text{forward}} &= a_{x,\text{ real}} \cdot \cos(\text{pitch}) \cdot \cos(\text{yaw}) \\
a_{y,\text{forward}} &= -a_{y,\text{ real}} \cdot \sin(\text{yaw}) \cdot \cos(\text{roll}) \\
a_{z,\text{forward}} &= -a_{z,\text{ real}} \cdot \sin(\text{pitch}) \cdot \cos(\text{roll}).
\end{aligned} \tag{3.9}
$$

The true forward acceleration, $a_{forward}$, was then calculated by summing the contribution of each acceleration

$$
a_{forward} = a_{x,\text{forward}} + a_{y,\text{forward}} + a_{z,\text{forward}}. \tag{3.10}
$$

Once $a_{forward}$ was calculated, a cumulative numerical integration using the trapezoidal rule was performed to obtain the velocity for each data point. The pitch, yaw, and roll were estimated using the same integration technique.

Due to drift and bias in the accelerometer and gyroscope, the velocity required drift correction. This was accomplished using the ZUPT method, assuming that the drift was linear over each step. The linear drift assumption was based on observing the velocity trend over a few steps, which exhibited a linear decrease from start to end, as illustrated in Figure 3.10 (a). By applying the ZUPT method, and assuming that the velocity, $v_{forward}$, is zero at both the beginning and end of each step, and knowing the start, t_{start}, and end, t_{end}, times for each step from the step detection, the drift was compensated. Initially, the velocity at the start of the step was set to zero, and the subsequent data points, up to the steps end, were adjusted based on the velocity at t_{start}, as in the following equation

$$
\mathrm{v}_{forward}(t) = \mathrm{v}_{forward}(t) - \mathrm{v}_{forward}(t_{start}) \quad \text{for} \quad t \in [t_{start}, t_{end}]. \tag{3.11}
$$

The linear drift for the velocity between t_{start} and t_{end} was then calculated separately for each step, resulting in the slope value s

$$
s = \frac{\mathrm{v}_{forward}(t_{end}) - \mathrm{v}_{forward}(t_{start})}{t_{end} - t_{start}}, \tag{3.12}
$$

where s represents the drift during that specific step.

The velocity at each data point between $\mathrm{v}_{\text{forward}}(t_{\text{start}})$ and $\mathrm{v}_{\text{forward}}(t_{\text{end}})$ was then adjusted for drift, and the corrected velocity, $\mathrm{v}_{\text{forward}}(t)$, was obtained by

$$
\mathrm{v}_{forward}(t) = \mathrm{v}_{forward}(t) - s \cdot (t - t_{start}) \quad \text{for} \quad t \in [t_{start}, t_{end}]. \tag{3.13}
$$

The velocity before and after the drift correction, along with the application of the

ZUPT method, can be seen in Figure 3.10. All values between steps, as well as any negative velocity estimates, were assumed to be zero.

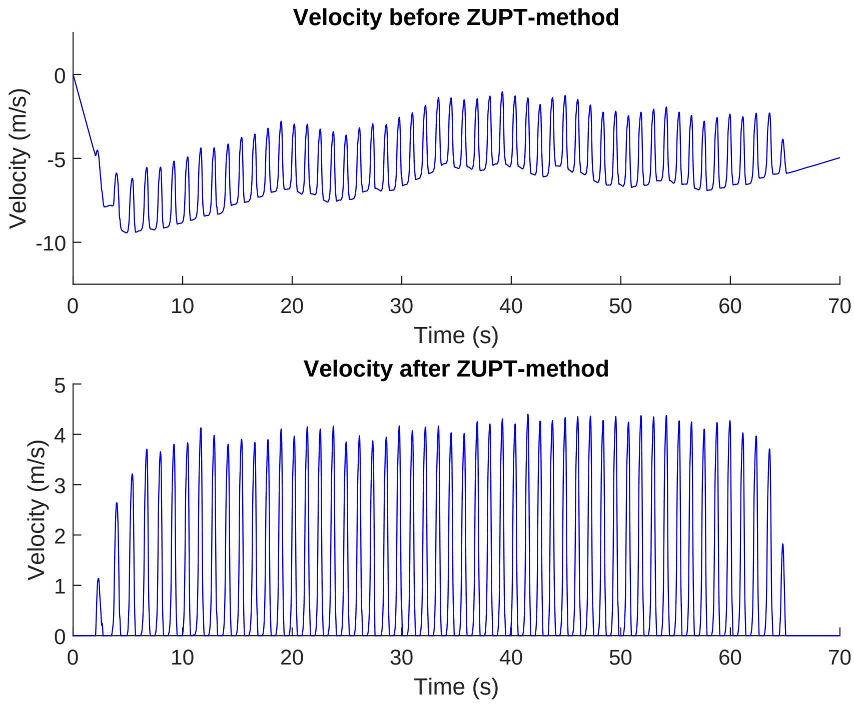

Figure 3.10: Velocity estimation in the $a_{forward}$ direction: (a) before applying the Zero-Velocity Update (ZUPT) method (upper plot), and (b) after applying the ZUPT method (lower plot).

The pitch, roll, and yaw also had to be drift-corrected, which was done using the same method as for the velocity, the ZUPT method, assuming linear decay for each step. The difference was that pitch and roll did not need to be zero at t_{start} and t_{end}. Instead, an initial offset, pitchstart(t_{start}) and rollstart(t_{start}), was considered for each step

$$\begin{aligned} \text{pitch}_{\text{start}}(t_{\text{start}}) &= \arctan\left(\frac{a_{x,local}(t_{\text{start}})}{a_{z,local}(t_{\text{start}})}\right) \\ \text{roll}_{\text{start}}(t_{\text{start}}) &= -\arctan\left(\frac{a_{y,local}(t_{\text{start}})}{a_{z,local}(t_{\text{start}})}\right). \end{aligned} \tag{3.14}$$

The initial values were subtracted from all values within each step to correct the pitch and roll. This adjustment was necessary because the IMU could have different angular displacements depending on its mounting. In theory, this also compensated for steps taken on inclined surfaces. The yaw was assumed to be zero at both t_{start} and t_{end}, just as with the velocity.

The distance was then obtained by integrating the velocity, using the same integration method as previously.

3.4 Tests

Multiple tests were performed, each targeting one specific feature of the signal processing.

To evaluate the accuracy of the algorithm, controlled walking and jogging tests were conducted under real-life conditions. Five healthy test subjects, four men and one woman, participated in the testing. Each subject completed the walking and jogging phase twice. The walking speed ranged between $1.23 - 1.83$ m/s and the jogging speed between $2.49 - 3.4$ m/s.

A 75-meter straight path was measured using a 5-meter tape measure. Marks were placed every 5 meters and re-measured from each point. First, the walking phase was tested. While one subject walked, another counted steps using a manual clicker, and a third recorded the time using the clock application on their mobile phone. The same procedure was repeated for the jogging phase. This produced a total of 20 tests, in which each subject walked or jogged 75 meters without turning or stopping.

For the step-counting and activity-level algorithms, additional tests were conducted to evaluate the system's performance under realistic conditions. These tests included scenarios such as turning, changes in speed, and standing still. As in the previously described tests, the step count was manually recorded using a clicker. It is important to note that these tests did not involve measurements of distance or velocity, as the conditions did not permit reliable manual measurement.

To calculate how accurately the algorithm estimated the number of steps taken, the following equation was used

$$A_s = 1 - \frac{|S_a - S_m|}{S_a}, \tag{3.15}$$

where A_s represents the step counting accuracy, S_a is the actual number of steps, and S_m is the number of steps measured by the algorithm.

The precision of the activity level identification was calculated using the following equations

$$A_a = \frac{N_c}{N_{\text{tests}}}, \tag{3.16}$$

for tests only involving one section of either walking or running and,

$$A_a = \frac{N_c}{N_{\text{sections}}}, \tag{3.17}$$

for tests involving multiple segments that included walking, running, or both,

where A_a is the accuracy of correct identification, N_{tests} is the number of tests performed, N_{sections} is the number of sections, and N_c is the number of correctly classified tests or sections.

To calculate how accurately the algorithm estimates the distance traveled, the following equation was used

$$A_D = 1 - \frac{|D_a - D_m|}{D_a}, \tag{3.18}$$

where A_D is the accuracy of the estimated distance, D_a is the actual distance traveled, and D_m is the distance calculated by the algorithm.

4

Results

The results of the tests described in the method section are presented below and are organized according to the step counting, activity level, and distance algorithms.

4.1 Step Counting

During tests in which the test subject alternated between walking and standing still, the algorithm estimated the number of steps with an accuracy between 96% and 100%, resulting in an average accuracy of 98%, as shown in Table 4.1. At most, it overestimated by one step or underestimated by two, with a mean error of zero steps, as shown in Figure 4.1.

In tests involving running and/or standing still, the algorithm achieved a step count accuracy between 88% and 100%, with an average accuracy of 94%, as shown in Table 4.1. The maximum overestimation was four steps, with a mean error of two steps too many, as shown in Figure 4.1.

For the test that included walking, running, and standing still, the algorithm counted 147 steps, which was one more than the actual number taken by the subject, as shown in Table 4.3.

Across all tests, the algorithm had a mean error of one extra step, as illustrated in Figure 4.1, while overall remaining close to the identity line, as seen in Figure 4.2.

2.5cm

1

Table 4.1: The table shows the actual steps taken, S_a, and the measured number off steps, S_m, during walking and/or standing still. The accuracy, A_s, is calculated using Equation 3.15.

S_a	S_m	A_s
50	48	0.96
50	49	0.98
54	52	0.96
51	51	1
42	43	0.98
41	42	0.98
52	53	0.98
54	54	1
50	51	0.98
53	52	0.98
52	52	1
47	46	0.98
25	26	0.96
Average A_s		**0.98**

Table 4.2: The table shows the actual steps taken, S_a, and the measured number off steps, S_m, during running and/or standing still. The accuracy, A_s, is calculated using Equation 3.15.

S_a	S_m	A_s
38	40	0.95
37	38	0.97
38	38	1
36	36	1
33	35	0.94
31	33	0.94
36	39	0.92
34	38	0.88
36	38	0.94
34	38	0.88
Average A_s		**0.94**

Table 4.3: The table shows the actual steps taken, S_a, and the measured number of steps, S_m, when the test person was switching between walking, running and standing still. The accuracy, A_s, is calculated using Equation 3.15.

S_a	S_m	A_s
146	147	0.99
Average A_s		**0.99**

Figure 4.1: Boxplot illustrating the distribution of the difference between the calculated number of steps, S_m, and the actual number of steps, S_a. The dashed line at 0 represents the true step count. Each boxplot displays the median, the 25th and 75th percentiles, as well as the upper and lower adjacent values.

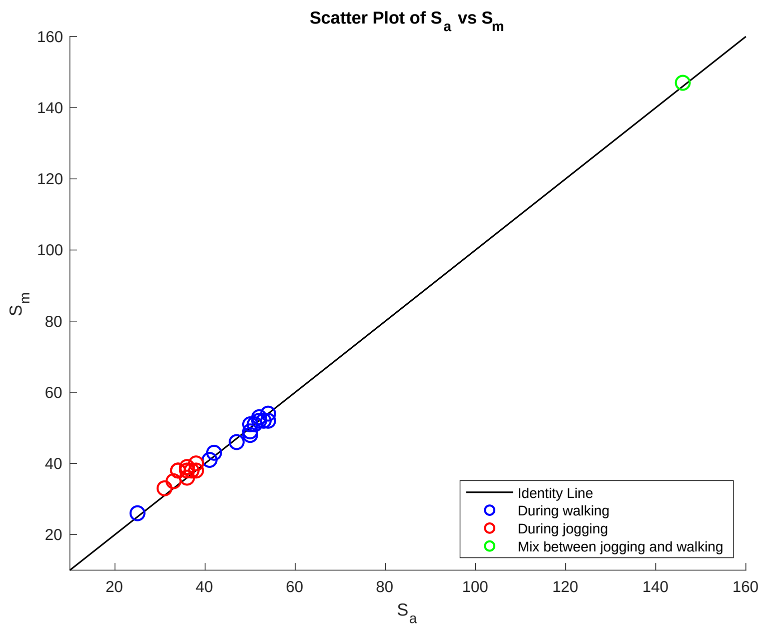

Figure 4.2: Scatter plot with identity line, illustrating the relationship between the actual number of steps taken during the tests, S_a, and the number of steps measured by the algorithm, S_m.

4.2 Activity Level

In the 20 tests involving either walking or running over a 75-meter straight path, the algorithm correctly identified the activity in all cases, resulting in an accuracy of 100%, as shown in Table 4.4.

In the additional tests where the subject switched between walking, running, and standing still, and included turns, the algorithm correctly identified all 9 walking sections and all 16 standing still sections, as shown in Table 4.5. However, it misclassified one running section, resulting in an accuracy of 80% for those segments, as shown in Table 4.5.

Table 4.4: The table presents the number of tests performed, N_{tests}, each involving a single section of either walking or running, along with the number of correctly classified sections, N_c. The accuracy, A_a, is computed using Equation 3.16.

Activity	N_{tests}	N_c	A_a
Walking	10	10	1
Running	10	10	1

Table 4.5: Test results for evaluations involving multiple sections consisting of walking, running, and/or standing still. N_s denotes the total number of sections, while N_c represents the number of correctly classified sections. The accuracy, A_a, was calculated using Equation 3.17.

Activity	N_s	N_c	A_a
Walking	9	9	1
Running	5	4	0.8
Standing still	16	16	1

4.3 Distance

The accuracy of the distance traveled during walking ranged between 93% and 98% across the ten trials of 75 meters, as shown in Table 4.6. The maximum calculated distance was 73.7 meters, the minimum was 70 meters, and the median was 72.3 meters, corresponding to a 96% accuracy, as illustrated in Figure 4.3.

For the jogging trials, the accuracy for the distance traveled ranged from 84% to 97% over the ten 75-meter tests, as shown in Table 4.7. The maximum calculated distance was 72.4 meters, the minimum was 63.3 meters, and the median was 68.5 meters, yielding a 91% accuracy, as illustrated in Figure 4.3.

In both the walking and jogging trials, the algorithm consistently underestimated the distance traveled, with the estimates being shorter than the actual 75-meter distance, as shown in Tables 4.6 and 4.7.

Table 4.6: The table shows the actual distance, D_a, and the estimated distance, D_m, during walking. The accuracy, A_d, is calculated using Equation 3.18.

D_a	D_m	A_d
75	72.513	0.97
75	71.247	0.95
75	73.434	0.98
75	73.667	0.98
75	73.436	0.98
75	73.588	0.98
75	71.584	0.95
75	70.000	0.93
75	72.012	0.96
75	71.537	0.95
Average A		**0.96**

Table 4.7: The table shows the actual distance, D_a, and the estimated distance, D_m, during jogging. The accuracy, A_d, is calculated using Equation 3.18.

D_a	D_m	A_d
75	68.312	0.91
75	67.172	0.90
75	72.402	0.97
75	68.591	0.91
75	70.988	0.95
75	68.994	0.92
75	68.391	0.91
75	70.123	0.93
75	63.257	0.84
75	66.229	0.88
Average A		**0.91**

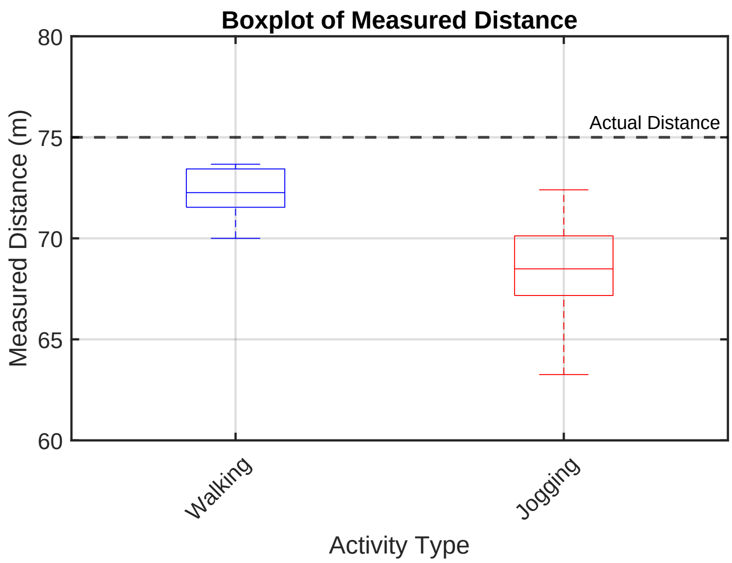

Figure 4.3: Boxplot showing the distribution of calculated walking and jogging distances. The red dashed line at 75 meters represents the actual distance walked and jogged during the activity. The boxplots illustrate the variability and central tendency of the estimated distances for each movement type.

5 Discussion

The discussion evaluates the system in relation to comparable products, assesses the validity of the conducted tests, explores algorithmic limitations, considers ethical implications, and outlines potential directions for future improvements.

5.1 Comparison to other products

The step counter demonstrates strong performance during walking, achieving an accuracy between 96% and 100%, with an average of 98%. However, accuracy decreases slightly during running, ranging from 88% to 100%, with an average of 94%. One test that combined both walking and running showed high reliability, reaching an accuracy of 99%, though its significance is limited due to being a single data point. Compared to the Fitbit and Garmin watches discussed earlier, the Fitbit outperformed the others with a mean error of only 0.1%, while the Garmins performance was more comparable to our system, showing a mean error of 3.9%.

The system also performs well in classifying activity levels during steady walking, running, and standing still. However, its accuracy diminishes in more dynamic activitiessuch as football or other sports that involve frequent, sudden changes in speed and direction. This limitation likely arises from the systems windowed analysis approach, which updates every 0.5 seconds and may not adequately capture rapid transitions in movement patterns.

The weakest aspect of system performance was distance estimation, where cumulative drift had the most pronounced impact. Nonetheless, accuracy remained relatively high, with results ranging from 93% to 98% for walking, and 84% to 96% for running. The average accuracy for distance estimation was 96% for walking and 91% for running. For comparison, the Fitbit exhibited a mean distance error of 4.1%, while the Garmin showed a mean error of 5.1%.

Across all three performance metricsstep counting, activity classification, and distance estimationthe system consistently performed better during walking than running. This discrepancy may be attributed to increased sensor bias at higher accelerations and angular velocities encountered during running. Compared to commercially available devices, the system delivers superior step detection performance and follows a similar trend of performing better during walking than running, as also observed with the Garmin and Fitbit. However, unlike these devices, our system is mounted on the foot. The Garmin is worn on the wrist, and the Fitbit is typically clipped to clothing, preferably at the waist. These differing placements result in distinct accelerometer signal patterns, as the motion of the hands, waist, and feet varies significantly, thereby requiring different signal processing strategies.

5.2 The validity of our tests

Bias and the resulting drift are issues that typically accumulate over time. However, since our tests were conducted over relatively short distances (75 meters), these effects are not fully captured or evaluated under ideal conditions. Although the ZUPT (Zero Velocity Update) method was used to reset the velocity at each step, thereby mitigating drift, it is uncertain how well the approach performs over longer compared to shorter distances.

Step counting was performed manually by an observer walking or running behind the test subject, depending on the trial. This method introduces a risk of human error, potentially leading to inaccurate step counts, either overcounting or undercounting, which could distort the perceived accuracy of the step-counting algorithm. Timing was also recorded manually, which may have introduced slight inaccuracies. However, given that the movement speeds during the tests were relatively low, any timing discrepancies are unlikely to have significantly affected the results.

Additionally, during the trials, the IMU was connected to a computer via a cable that the test subject had to carry. This setup may have influenced the natural walking or running motion, possibly altering stride patterns and restricting movement. As a result, the collected data may not fully represent unconstrained or real-world usage scenarios.

5.3 Limitations with algorithm

The algorithm for identifying activity levels performs well when the user maintains a consistent stride frequency during walking or running, accurately classifying the users activity level in almost all cases, as shown in Tables 4.4 and 4.5. However, when the user rapidly changes stride frequency without a pause in movement, the algorithm struggles to detect this transition. This is because the algorithm treats the entire sequence as a single activity segment, due to the absence of a non-movement window required to distinguish between two distinct phases.

Rapid changes in stride frequency also negatively impact the step-counting component of the algorithm, as it relies on the same segmented approach used for activity classification. When the algorithm fails to separate different activity phases and instead treats them as one continuous segment, it applies the dominant frequency over the entire duration. This leads to either an overestimationif the higher stride frequency dominatesor an underestimationif the lower frequency dominates.

As illustrated in Figure 4.1, the algorithm tends to overestimate the number of steps more frequently during running compared to walking. This is primarily due to the fixed two-second time window, which can result in the algorithm setting the start time up to two seconds early. When this early start time is paired with a higher stride frequency, the compounded error becomes more significant. One potential solution is to reduce the window length, which would improve the algorithm's temporal precision. However, shortening the window would also raise the minimum detectable stride frequency, limiting its ability to accurately register slower walking paces. A trade-off is therefore necessary, accepting some degree of overestimation in step counts in order to maintain sensitivity to slower gait patterns.

The algorithm for estimating velocity and distance also presents limitations, particularly in determining the start and end points of each step and in the assumptions used for distance calculation. Features for detecting step events were selected based on analysis of both derived (e.g., differentiated) and filtered raw data (e.g., acceleration), guided by visual inspection of graphs and signal values. During this process, it became evident that certain features are person-specific, varying according to individual gait patterns and whether the subject was walking or jogging. The final features were selected to be as general as possible to accommodate both activities and multiple users. Nonetheless, this generalization introduced occasional misclassifications and multiple detected starts or ends for single steps, which the algorithm subsequently filtered. This behavior is depicted in Figure 3.9. Consequently, while step start and end points were generally correctly identified, their exact temporal placement within the step cycle was sometimes inaccurate. This can be observed in the second graph of Figure 3.9, where the green-marked step beginnings are correctly placed within the general region of the steps, but vary in their exact location. This affects calculations involving the ZUPT method for velocity and angular displacement. As such, the algorithms distance accuracy may decline for individuals not included in the test set or when applied to other activities, such as runningdepending on whether the selected features are overfitted to specific test subjects and activities or sufficiently general.

In the velocity estimation component, the algorithm assumes a stepwise linear decay model for drift. This assumption was based on inspection of velocity graphs, as shown in the upper plot of Figure 3.10. If the true nature of drift differs from this linear model, the corrections applied in Equations 3.123.13 may introduce errors, leading to reduced accuracy in distance estimation.

5.4 The ethics of our device

Our low-cost IMU device poses minimal physical risk. To ensure user safety, the lithium ion cell must be enclosed in a shock absorbing and impact resistant casing. This design mitigates the risk of damage from foot strikes and other impacts, thereby reducing the likelihood of puncture or short circuiting. Additionally, the device must incorporate thermal protection measures. Even under worst-case operating conditions, such as continuous sampling and microSD card writing, the device temperature must remain sufficiently low to prevent harm.

Human gait is a biometric trait, meaning that the motion data collected by our IMU can potentially be used to identify individuals. As such, this data qualifies as "personal data" under the EU General Data Protection Regulation (GDPR), Regulation 2016/679, and must therefore be handled with appropriate safeguards.

In the fitness tracker market, applications often collect even more sensitive health metrics, including heart rate and sleep patterns, which are typically stored on centralized servers. Users frequently relinquish ownership of their data, and companies may share it through user-configurable settings. This practice introduces two significant risks: data breaches and unauthorized third-party access.

Our approach differs substantially from this model. All IMU data is stored offline, residing solely in the devices local memory. No wireless synchronization or cloud upload is implemented, thereby eliminating the risk of data interception or unauthorized distribution.

5.5 Moving forward

During the project, the primary focus was on signal processing. Had more time been available, additional effort would have been dedicated to developing a functional product prototype. Within the scope of signal processing, the following potential improvements could be made.

The primary area for improvement in the step counting and activity level algorithm lies in how the algorithm segments the data into periods of standing still, walking, or running. Enhancing the accuracy of this segmentation would improve both the step count and the identification of activity levels. One potential approach involves collecting additional data to evaluate whether parameters such as time window size, sliding window overlap, and threshold values can be optimized. As previously described, a smaller time window could more precisely identify the onset of each segment, though it would limit the detectable frequency range. A smaller sliding windowresulting in increased overlap between windows, could enhance the precision of segment termination detection. However, this would demand greater computational resources, potentially degrading performance and increasing latency for the end user. Adjusting the threshold value, $|f_{dominant}|$, would impact which time windows are classified as periods of movement. A higher threshold would result in fewer windows being labeled as moving, while a lower threshold would label more. Determining a more effective threshold requires a larger dataset to validate accurate classification across various users and activities.

When considering the distance estimation component of the algorithm, the key area for refinement is the identification of step initiation and termination. This is critical for subsequent velocity estimation via the ZUPT method and for computing accurate angles when estimating $a_{forward}$. Additional data could be analyzed to minimize subject-specific or activity-specific bias in the algorithms feature selection, thereby enhancing its robustness and generalizability. Step detection could also benefit from a more sophisticated method for selecting closely spaced step start and end candidates. Rather than simply selecting the first detected point within a 0.3-second window, which may not be optimal, selection could be linked to a calculated feature. For example, the chosen point could be where the ratio of x-axis acceleration within a small window before and after the candidate is maximized, indicating a more pronounced change in motion.

An effective strategy for improving step onset and offset detection may involve integrating the activity classification segment of the algorithm. By identifying the type of activity within each segment, activity-specific features could be used to tailor the step detection process. This would allow the algorithm to be optimized for specific activities, reducing activity-related bias while maintaining generalizability across users. As the accuracy of step detection improves, the overall precision of distance estimation is also expected to increase.

The current implementation of ZUPT and step counting relies on manually derived thresholds and conditions, which were developed through analysis of data collected during testing. Consequently, the algorithm may encounter limitations when applied to more diverse scenarios beyond the original development environment.

Another promising avenue for improving robustness and generalizability is the incorporation of machine learning techniques. Rather than depending on static thresholds derived from a limited and homogeneous dataset, machine learning models could be trained to recognize step patterns and zero-velocity phases directly from raw sensor data. By leveraging diverse training datasets, including varying terrains, gait abnormalities and more. The system could learn to detect subtle and generalizable features, ultimately yielding a more reliable and adaptive solution.

6

Conclusion

By evaluating a single, low-cost, six-axis IMU under varied gait conditions, this work demonstrates that accurate offline monitoring of walking and running can be achieved without GPS, barometer, or magnetometer data. In controlled walking trails, step counting reached 98 % accuracy (96-100 %) and distance estimation 96 % (93-98 %), while during running step counting achieved 94 % accuracy (88-100 %) and distance estimation 91 % (84-97 %). Activity classification was perfect in structured tests and 89% accurate in mixed scenarios.

Despite these strengths, performance degrades when gait speed shifts abruptly, as our fixed sliding-window analysis fails to capture rapid transitions. Over extended distances, drift accumulates and zero-velocity updates only partially compensate. The current segmentation approach relying on linear drift assumptions and static thresholds may also lack generality across different users and terrains.

Future work should implement adaptive windowing that adjusts length and overlap in real time and fuse classification with step detection to refine phase segmentation. Leveraging machine-learning models trained on diverse datasets to improve robustness across speeds, surfaces and users. Integrate these algorithms into a standalone wearable with onboard storage would maintain affordability and safeguard privacy.

The ability of low-cost IMUs to accurately track gait measures offline is validated by this signal-processing approach. With algorithmic refinements and extensive validation, it offers a viable foundation for an accessible wearable solution in everyday health and performance monitoring.

7

Use of AI Statement

AI has primarily been used to streamline the workflow of the project. The usage primarily focused on taking repetitive and manual tasks and automating them using AI. This allowed the group to focus on the development of our signal processing methods. Some examples of AI usage are improving text quality, summarizing documentation and making information gathering more efficient.

Firstly, improving text quality was primary done with Chat GPT [23]. It was used for spell checking, correcting tense and stylizing the text to an academic standard. Chat GPT was never used to generate original ideas and text. Additionally, Chat GPT was also given documentation from the IMU manufacturer and Matlab. Then it was either asked to summarize the document or asked relevant questions about the document. It enabled more efficient search through the documentation and less time was spent looking for the information. Lastly, generative AI was used to create specialized AND/OR searches to find academic papers which was done through Scopus AI [24]. The AI summaries of Scopus AI was never used as a source, it was only a method to efficiently access papers on the topic.

Bibliography

[1] A'Naja M. Newsome et al. "2025 ACSM Worldwide Fitness Trends: Future Directions of the Health and Fitness Industry". In: *ACSM'S Health & Fitness Journal* 28.6 (Nov. 2024), pp. 11–25. ISSN: 1536-593X. DOI: 10.1249/FIT.0000000000001017. URL: https://journals.lww.com/acsm-healthfitness/fulltext/2024/11000/2025_acsm_worldwide_fitness_trends__future.6.aspx.

[2] Nader Naghavi, Aaron Miller, and Eric Wade. "Towards Real-Time Prediction of Freezing of Gait in Patients With Parkinsons Disease: Addressing the Class Imbalance Problem". In: *Sensors 2019, Vol. 19, Page 3898* 19.18 (Sept. 2019), p. 3898. ISSN: 1424-8220. DOI: 10.3390/S19183898. URL: https://www.mdpi.com/1424-8220/19/18/3898/htm%20https://www.mdpi.com/1424-8220/19/18/3898.

[3] Lin Zhou et al. "How We Found Our IMU: Guidelines to IMU Selection and a Comparison of Seven IMUs for Pervasive Healthcare Applications". In: *Sensors (Basel, Switzerland)* 20.15 (Aug. 2020), p. 4090. ISSN: 14248220. DOI: 10.3390/S20154090. URL: https://pmc.ncbi.nlm.nih.gov/articles/PMC7435687/.

[4] Guidenav. *What is the price of Inertial Measurement Unit (IMU)*. URL: https://guidenav.com/what-is-the-price-of-inertial-measurement-unit-imu/.

[5] Inertialsense. *The 4 grades of IMUs (Inertial Measurement Units)*. URL: https://inertialsense.com/4-grades-of-imus/.

[6] *vívofit*. URL: https://www.garmin.com/sv-SE/p/143405#specs.

[7] *Garmin Vivofit 4 aktivitetsarmband L (svart) - Elgiganten - Elgiganten*. URL: https://www.elgiganten.se/product/sport-fritid/sport-fitness/aktivitetsarmband/garmin-vivofit-4-aktivitetsarmband-l-svart/12491?utm_source=google&utm_medium=cpc&utm_campaign=SE%20-%20Generic%20-%20Shopping%20-%20AO%20-%20SDA%20-%20Sports%20%26%20Hobby%20-%20Sports%20%26%20Fitness%20-%20Activity%20tracker&utm_id=18519176334&gad_source=1.

[8] Fitbit. *Wireless Activity Tracker*. Tech. rep. URL: https://staticcs.fitbit.com/content/assets/help/manuals/manual_zip_en_US.pdf.

[9] Yangjian Huang et al. "Validity of FitBit, Jawbone UP, Nike+ and other wearable devices for level and stair walking". In: *Gait & Posture* 48 (July 2016), pp. 36–41. ISSN: 0966-6362. DOI: 10.1016/J.GAITPOST.2016.04.025.

[10] Pu Duan et al. "Bio-Inspired Real-Time Prediction of Human Locomotion for Exoskeletal Robot Control". In: *Applied Sciences 2017, Vol. 7, Page 1130* 7.11 (Nov. 2017), p. 1130. ISSN: 2076-3417. DOI: 10.3390/APP7111130. URL: https://www.mdpi.com/2076-3417/7/11/1130/htm%20https://www.mdpi.com/2076-3417/7/11/1130.

[11] Mihaela I. Chidean et al. "Full Band Spectra Analysis of Gait Acceleration Signals for Peripheral Arterial Disease Patients". In: *Frontiers in Physiology* 9.AUG (Aug. 2018), p. 1061. ISSN: 1664042X. DOI: 10.3389/FPHYS.2018.01061. URL: https://pmc.ncbi.nlm.nih.gov/articles/PMC6090042/.

[12] Elaine M. Murtagh et al. "Outdoor Walking Speeds of Apparently Healthy Adults: A Systematic Review and Meta-analysis". In: *Sports medicine (Auckland, N.Z.)* 51.1 (Jan. 2021), pp. 125–141. ISSN: 1179-2035. DOI: 10.1007/S40279-020-01351-3. URL: https://pubmed.ncbi.nlm.nih.gov/33030707/.

[13] Ernst Albin Hansen et al. "The role of stride frequency for walk-to-run transition in humans". In: *Scientific Reports 2017 7:1* 7.1 (May 2017), pp. 1–8. ISSN: 2045-2322. DOI: 10.1038/s41598-017-01972-1. URL: https://www.nature.com/articles/s41598-017-01972-1.

[14] B. T. Van Oeveren et al. "Inter-individual differences in stride frequencies during running obtained from wearable data". In: *Journal of sports sciences* 37.17 (Sept. 2019), pp. 1996–2006. ISSN: 1466-447X. DOI: 10.1080/02640414.2019.1614137. URL: https://pubmed.ncbi.nlm.nih.gov/31079578/.

[15] Daniel E. Lieberman et al. "Effects of stride frequency and foot position at landing on braking force, hip torque, impact peak force and the metabolic cost of running in humans". In: *Journal of Experimental Biology* 218.21 (Nov. 2015), pp. 3406–3414. ISSN: 00220949. DOI: 10.1242/JEB.125500.

[16] Christian Lundquist. *Sensor Fusion for Automotive Applications*. 2011. ISBN: 9789173930239. URL: www.control.isy.liu.se.

[17] VectorNav. *IMU Specifications Explained*.

[18] STMicroelectrics. *This is information on a product in full production. LSM6DS3TR-C iNEMO inertial module: always-on 3D accelerometer and 3D gyroscope Datasheet-production data Features*. Tech. rep. 2017. URL: www.st.com.

[19] Paulo Flores. *SPRINGER BRIEFS IN APPLIED SCIENCES AND TECHNOLOGY Concepts and Formulations for Spatial Multibody Dynamics*. Tech. rep. 2015, pp. 11–13. URL: http://www.springer.com/series/8884.

[20] *Finite Impulse Response - an overview | ScienceDirect Topics*. URL: https://www.sciencedirect.com/topics/engineering/finite-impulse-response.

[21] Michael V. Potter et al. "Effect of IMU Design on IMU-Derived Stride Metrics for Running". In: *Sensors 2019, Vol. 19, Page 2601* 19.11 (June 2019), p. 2601. ISSN: 1424-8220. DOI: 10.3390/S19112601. URL: https://www.mdpi.com/1424-8220/19/11/2601/htm%20https://www.mdpi.com/1424-8220/19/11/2601.

[22] The Qt Company Ltd. *Qt for Python*. 2025. URL: https://doc.qt.io/qtforpython-6/.

[23] OpenAI. *Chat GPT*. URL: https://openai.com/chatgpt/overview/.

[24] *Scopus AI: Trusted content. Powered by responsible AI.* URL: https://www.elsevier.com/products/scopus/scopus-ai.

DEPARTMENT OF ELECTRICAL ENGINEERING
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden
www.chalmers.se

AVANCEZ
1829
CHALMERS