The University of Melbourne
Department of Mechanical Engineering

MCEN90032 Sensor Systems - Workshop 1

Step Counting with MATLAB

Tuan Khoi Nguyen (1025294)

Introduction

This report will demonstrate the process of counting steps using MATLAB Mobile Phone Sensors, which involves signal processing and filtering, as well as introducing further adaptive approaches that can covers a larger variety of step frequencies and speed changes.

Key goals

The report will include the following listed:

- Setting up and define initial measurement parameters.
- Calibration of the sensor to detect bias.
- Frequency analysis of the signal through Fourier Transform.
- Adaptive approach of setting up filter parameters for detected frequencies, using a Normal Distribution-based method.
- Adaptive approach of setting signal peak requirements to detect steps.
- Result analysis across different movement types: Strength and weaknesses.
- Method appendix: containing all the theories that aids in creating this report.

Stage 1

Setting of Main Experiment

This stage sets up the methods and parameters for the data collection process, to ensure that the data retrieved can provide as much information for the analysis as possible.

1.1 Assumptions

- Gravity at experiment location is exactly $9.8m^2/s$ [1].
- Human step frequency ranging mostly from 2-4Hz [2].

1.2 Orientation

For measurement simplicity, the main experiment will hold the phone face up, with the phone's top part heading towards user's left direction. That way, including the gravity, the expected true values of the acceleration in (x, y, z) direction will respectively be (0, 0, 9.8) m^2/s.

1.3 Sampling Rate

The average moving pace of a person is assumed to range from 2 to 4Hz. Therefore, a safe upper bound for the step frequency can be set to 5Hz. According to the Shannon-Nyquist theorem, sampling rate would need to be at least twice the signal frequency to capture its details. Therefore, the sampling rate for this task is set to 10Hz, exactly twice the upper bound of the movement frequency so that all signal features are fully observed, while limiting the amount of noise getting in just right.

1.4 Recording Processes

The process of recording will be done as follows:

- The process will begin with at least 3 seconds of standing still, as the first 3 seconds will be used for sensor calibration.
- An arbitrary, but tracked number of walking steps will be done, then the walker will stop for a few seconds.
- The step above will be repeated one more time.

With the steps above, the coordinates (a_x, a_y, a_z) of the acceleration is recorded. To capture all patterns, the norm $|a| = \sqrt{a_x^2 + a_y^2 + a_z^2}$ is calculated and used for the analysis.

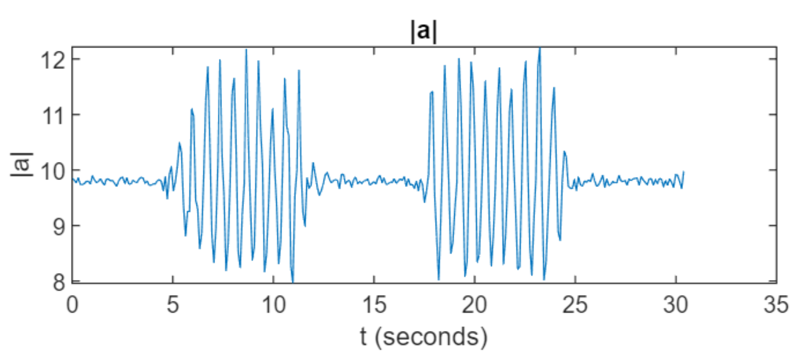


Figure 1.1: The retrieved acceleration norm of the walking signal

Stage 2

Data processing

2.1 Calibration

The first 3 seconds of recording is the calibration part of the signal. Observation shows that bias appeared given the known truth values, and there are slight fluctuations in the signal as well. Therefore, the process has 2 goals, the first one is being able to measure the bias value, given that we have known the true value on each axis, and to capture the noise magnitude and frequency in order to determine requirements for any peak jump to be considered a step.

```
Estimated bias:

  X0 = -0.0283
  Y0 = 0.0299
  Z0 = -0.0049
```

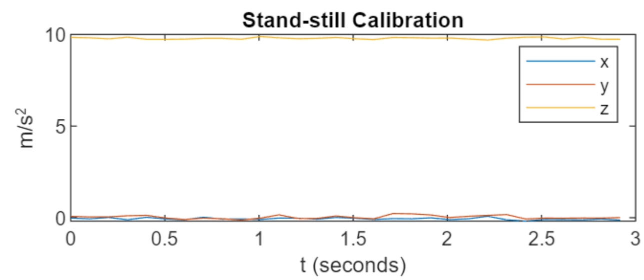


Figure 2.1: Calibration result for the first 3 seconds of signal

Bias value retrieval With the calibration data not being fully constant, it is suitable for the calibration signal to be fitted with a straight line in the form of $h(t) = C_0$ where C_0 is an arbitrary constant. Therefore, linear regression (details in Appendix A) is used to estimate C_0, which will be set as the bias value.

2.2 Frequency Analysis

To see the frequencies power for the signal, Fourier transform is used. Gravity and bias are removed from the signal before transformation, as they are equivalent to a very high-powered signal at 0Hz, which may make observations harder to carry out.

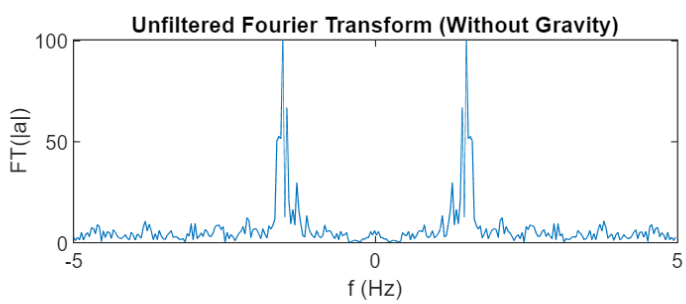


Figure 2.2: Fourier Transform of the signal

Looking at the signal in the frequency domain, we can now see where the walking frequency ranged from the peaks. Surrounding the step frequency, noises are also observed on higher frequencies, and as the phone orientation may not be fully stationary while moving, lower frequencies also appeared in the plot as a result of slight fluctuations.

While estimating an exact value can help to define single step frequency, it may not work efficiently on cases where cadence - multiple step frequencies happens. Therefore, a suitable method is needed to ensure that all movement frequencies are covered.

2.3 Adaptive parameter algorithm for different frequencies

2.3.1 Observation

It is noticeable that the power in the frequency domain acts like a density function, where it denotes how often the signal can be seen. Therefore, using Central Limit Theorem (Appendix A), we can approximate a normal distribution for the frequencies in the plot. This distribution will give us a confident estimate on the range of step frequency, and from there, make a more dynamic model that captures arbitrary speeds.

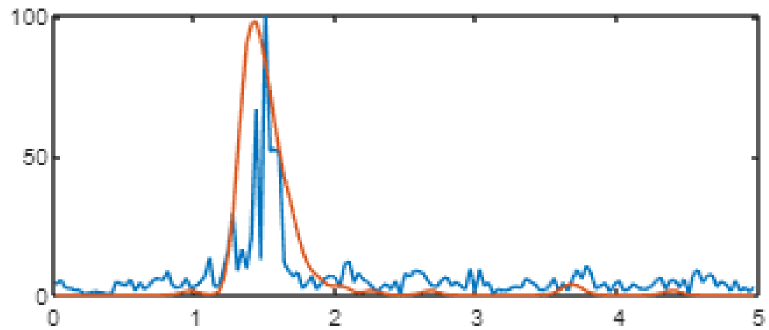


Figure 2.3: An approximation of the signal's frequency distribution in frequency domain

2.3.2 Implementation

First, the power values are normalized to have a sum of 1, so that it can imitate the properties of a cumulative distribution function, where the power is the probability, and the frequency is the value. Then, using the normal distribution calculation method (Appendix A), the mean and the standard deviation of the 'distribution' is retrieved.

With the mean μ and standard deviation σ, looking up on the z-score table will tell what range can cover a specified portion of the data. Trial and error from the current signal shows that low-frequency noise takes up 0.13% of data, equivalent to a lower bound of $\mu - 3\sigma$, and the high-frequency noise takes up 0.03% of data, equivalent to an upper bound of $\mu + 3.5\sigma$. Therefore, these bounds will be used as the main value for the filtering parameter in this task.

2.4 Filter set

With bounding on both sides, bandpass filter is a suitable choice to cover them at once. Using the adaptive method mentioned in the previous section, both bounds can be calculated from the distribution, which has the range of $(\mu - 3\sigma, \mu + 3.5\sigma)$. After filtering, the resulting signal is observed as follows:

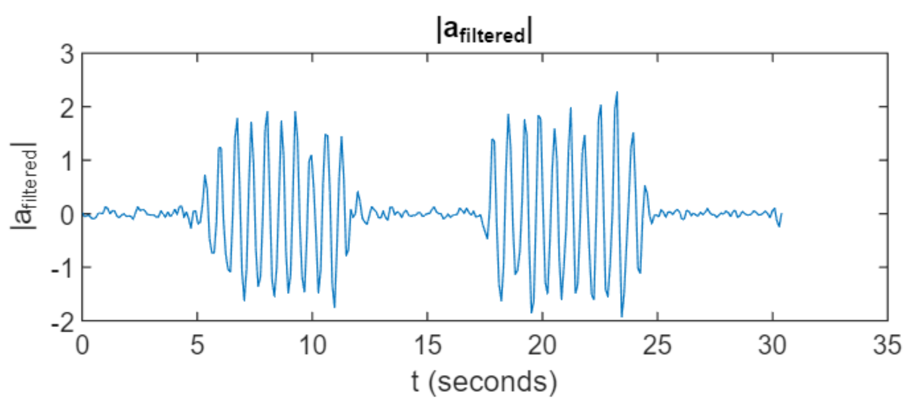


Figure 2.4: Acceleration norm, after applying bandpass filter

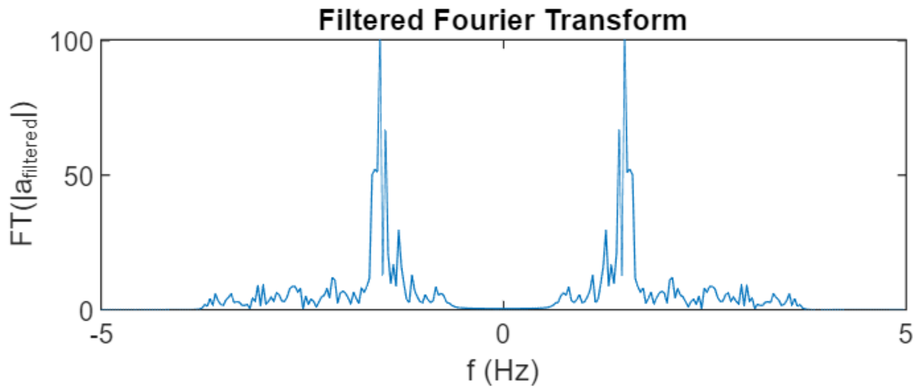


Figure 2.5: Bandpass filtered Fourier Transform of the signal

The result shows that this filter approach worked well for the sample signal: the waves are smooth across the steps on Figure 2.4, making the steps being more visible, and the outer noise are diminished in Figure 2.5. The process will now move on to step detection, based on the wave peaks of the signal.

2.5 Steps detection using signal peaks

From observation, each peak jump in the plot represents a step walked. Therefore, the built-in MATLAB function `findpeaks()` is used to find peaks that satisfy specific conditions. The conditions are set as follows:

Distance between peaks The calibration showed that drifting noise appeared at a low frequency, and potential noise may still occur at higher frequency due to filter not fully filtering them out. Therefore, the frequency bound is set again as follows for this function:

- Maximum frequency: The step frequency's upper bound retrieved from the adaptive method $(\mu + 3.5\sigma)$.
- Minimum frequency: The maximum frequency occurred during calibration process.

Peak prominence Peak prominence indicates how large the jump is. For this task, given that calibration is done, the lower bound for the peak prominence is the max prominence during the calibration process, which is an estimation of how large the noise can be.

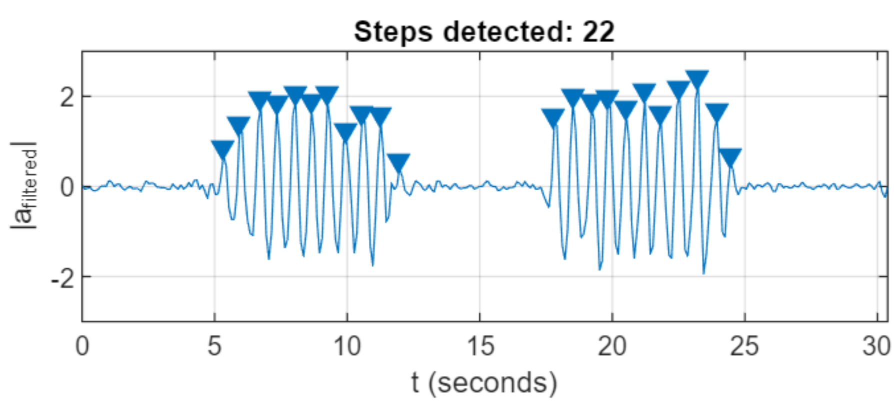


Figure 2.6: Detected steps from sample signal using the methods mentioned in this report

The final result, after applying the peak-finding function, has captured all the steps in the sample data without error, having an accuracy of 100%. This proves that for this type of signal, the methodologies mentioned in this report worked extremely well. The next stage will attempt to test these methods with further scenarios, to evaluate the robustness of the method though strength and weakness identification.

Stage 3

Results & Analysis

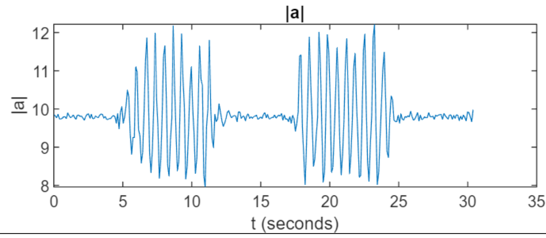
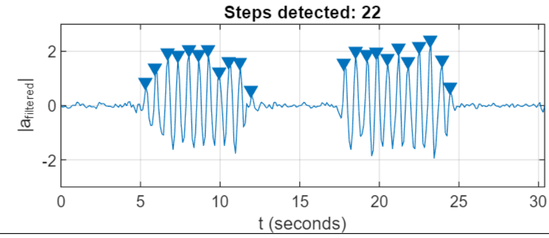
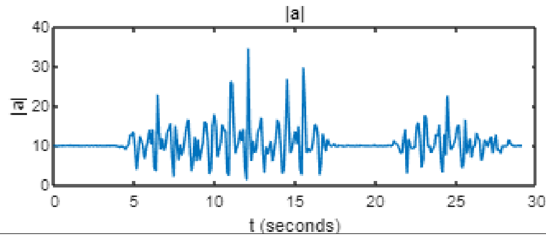
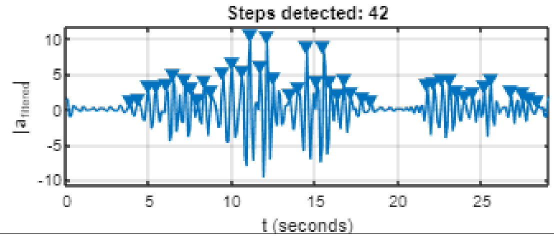
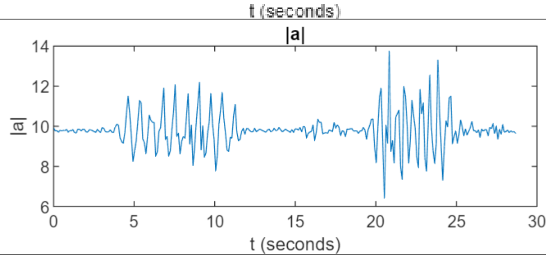
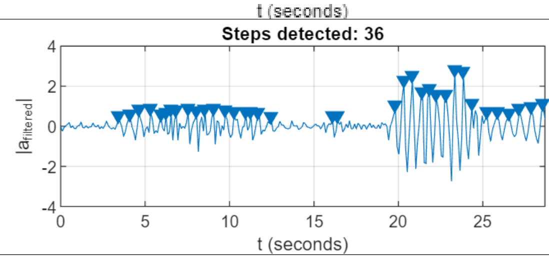
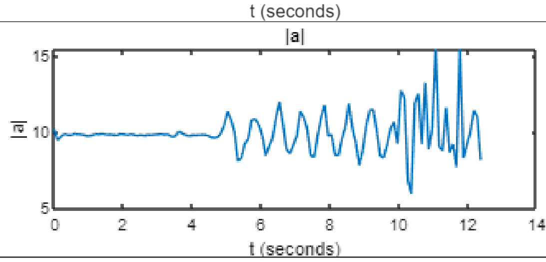
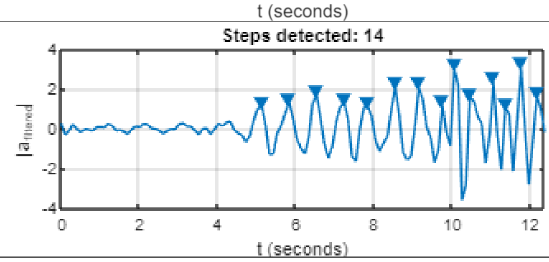
Details	Original signal	Filtered signal & Step count
Walking: 22 steps		
Phone in pocket [1]: 42 steps		
Walking then running: 22 steps		
Gradually increasing speed: 15 steps		

Table 3.1: Step detection result on different types of movement patterns, using step range of $(\mu - 3\sigma, \mu + 3.5\sigma)$

3.1 Evaluation

3.1.1 Compatibility strength

High accuracy results on Table 3.1 proves that the processing method is versatile in many different further cases of movement signals, which includes holding phone in pocket (skipping bias finding step), magnitude change and frequency change. It can be concluded that this approach is robust and covers most of the common movement signals well.

3.1.2 Weakness

The approach showed strength in approximation, where it was able to capture different ranges in the data, and could cover many cases of movement. However, it is noticeable that when no movement is made, waves still appeared in post-filtered signal, which may end up larger than the actual movement waves and caused measurement error, as seen in the 'Walking then running' case in Table 3.1. This proves that the method is still not fully accurate.

[1] Bias finding step is skipped in this case, as real value of the acceleration changes frequently by time.

This happened because the percentage of noise appearing was based on one signal only. In reality, the noise ratio might be different, hence defining an exact occurrence rate of noise might not be sufficient enough. Slightly modifying the step lower bound from $\mu - 3\sigma$ to $\mu - 3.5\sigma$ results in a more accurate result as seen in Figure 3.1. Therefore, for each signal, hyperparameter tuning is needed, or if automation is wanted, further research will need to be carried out to determine the relationships that leads to the right ratio.

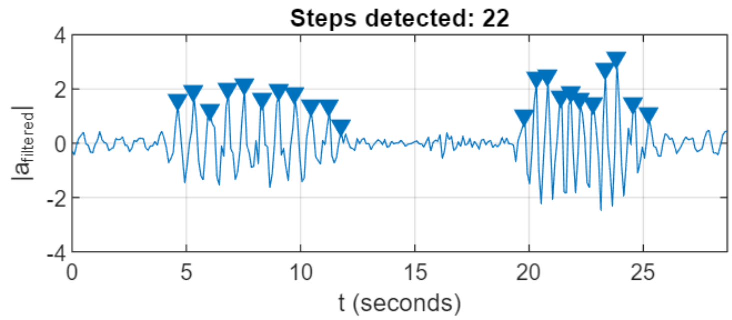


Figure 3.1: Detected steps from 'Walking then running' signal, using step range of $(\mu - 3.5\sigma, \mu + 3.5\sigma)$

3.2 Potential improvements & ideas

While the methodologies performed well, there are still room for improvements. However, due to the time constraints, these extensive ideas are held back, and will be considered later in the future.

Further calibration To have a more accurate estimation on the noise information, more calibration time is needed, and the process should also be done on different stages of the movement rather than at the start only, such as right after stopping.

Automated frequency range tuning The current process was based on a set of parameters that was estimated from a single signal through trial and error. To estimate a more precise range for each different signal automatically, further information is needed, such as ratio between stopping time and moving time, or rate of large magnitude noise, etc., and then the process will need to work out the relationship between them and the range.

Stopping-Moving Split A faster way of approaching with less tuning is to find a way to detect when the user is stopping and when the user is moving. This way, the methodologies would only need to consider the moving parts of the signal and do not have to worry about noise appearing at the stopping part.

3.3 Conclusion

Overall, the approach worked well and maintained its robustness on a large variety of movement sequences, despite using a global hard-defined set of parameters. However, to be even more precise, more evaluation will need to be done in order to capture the most accurate number of steps. These step detection results can later on be extended to many applications, where they will become meaningful data in health tracking, or for terrain identification on hiking tracks for example.

Bibliography

[1] Hone I. G., Richardson L. M., and Wynne P. Geophysical datasets over continental australia 1986-2002. *Record 100*, 2002.

[2] G. A. Cavagna, M. Mantovani, P. A. Willems, and G. Musch. The resonant step frequency in human running. *Pflugers Archiv European Journal of Physiology*, 434(6):678–684, September 1997.

Appendix A

Additional Resources

A.1 Linear Regression

Linear regression tries to fit a straight line in the from of $h(t) = \beta_0 + \beta_1 t$ through the given set of data points (x_i, y_i). Let:

$$B = \begin{bmatrix} \beta_0 \\ \beta_1 \end{bmatrix} \qquad X = \begin{bmatrix} 1 & x_0 \\ 1 & x_1 \\ \vdots & \\ 1 & x_n \end{bmatrix} \qquad Y = \begin{bmatrix} y_0 \\ y_1 \\ \vdots \\ v_n \end{bmatrix}$$

To find β_0, β_1, solve for $Y = XB$.

A.2 Central Limit Theorem

The larger the sample size, distribution of the sample will more approximate the normal distribution. Therefore, it is safe to assume that the powers in the frequency domain approximates a normal distribution.

A.3 Calculate Normal Distribution

To calculate the normal distribution with mean μ and standard deviation σ for a sample x with probability $P(x)$, we retrieve them as follows:

$$\mu = \frac{\sum_i P(x_i)x_i}{\sum_i P(x_i)} \qquad \sigma = \sqrt{\frac{\sum_i (x_i - \mu)^2}{\sum_i P(x_i)}}$$

A.4 z-Score Table

Given a constant c and distribution $N(\mu, \sigma)$, the z-table will show the value for $P(x \leq \mu - c\sigma) = P(x \geq \mu + c\sigma)$. This can help estimating the range to cover a specified portion of data. To see further instructions and the table itself, `http://www.z-table.com/` is a suitable website.