Measuring air time for runners in real time

Master's thesis in Embedded Electronic System Design

ELSA ANDERSSON

KRISTOFFER LINDH

Department of Computer Science and Engineering
CHALMERS UNIVERSITY OF TECHNOLOGY
UNIVERSITY OF GOTHENBURG
Gothenburg, Sweden 2020

Master's thesis 2020

Measuring air time for runners in real time

ELSA ANDERSSON
KRISTOFFER LINDH

UNIVERSITY OF
GOTHENBURG

CHALMERS
UNIVERSITY OF TECHNOLOGY

Department of Computer Science and Engineering
Chalmers University of Technology
University of Gothenburg
Gothenburg, Sweden 2020

Measuring air time for runners in real time

ELSA ANDERSSON
KRISTOFFER LINDH

© ELSA ANDERSSON, 2020.
© KRISTOFFER LINDH, 2020.

Supervisor: Lena Peterson, Department of Computer Science and Engineering
Advisor: Jonas Williamsson, CONSAT Engineering AB
Examiner: Per Larsson-Edefors, Department of Computer Science and Engineering

Master's Thesis 2020
Department of Computer Science and Engineering
Chalmers University of Technology and University of Gothenburg
SE-412 96 Gothenburg
Telephone +46 31 772 1000

Typeset in LaTeX
Gothenburg, Sweden 2020

Measuring air time for runners in real time

Elsa Andersson
Kristoffer Lindh
Department of Computer Science and Engineering
Chalmers University of Technology and University of Gothenburg

Abstract

Running has always been a popular way to exercise and one way to improve the running technique and prevent injury is to adjust the gait. In order to improve the gait the runner must gain knowledge about his/her own gait. In this master's thesis, a system has been developed that measure the time when a runner is in the air while running. The data about air time should be available to the runner with such a short delay that the runner could immediately adjust the gait. The system should also be light weight and small enough to be carried around whilst having a low power consumption in order to minimize battery size.

The system has been developed using one microcontroller and two inertial measurement units. A functional algorithm that can determine gait events in order to measure air time with an accelerometer has been developed. The precision of the system was low and needs to be adjusted in order to be useful to a runner. The small size of the system makes it wearable and due to the low power consumption, the system could be powered for eight hours with a small button cell battery.

Keywords: Gait event, Inertial measurement unit, Microcontroller

Acknowledgements

We would like to thank Consat Engineering AB and our supervisor there, Jonas Williamsson for giving us the opportunity to get to know the company and writing our masters thesis there.
We also want to thank David Kvist and Christoffer Mathiesen for taking the time to peer-review and proof read our report when we needed help.
We would also like to thank our supervisor at Chalmers, Lena Peterson for helping us complete this project through a pandemic. Finally, we would like to thank our examiner Per Larsson Edefors for his invaluable, and often very fast, feedback.

Elsa Andersson, Kristoffer Lindh, Gothenburg, October 2020

Contents

List of Figures

List of Tables

1

Introduction

Many people like to run, some compete and some just enjoy the exercise. All runners have a different gait that can be changed or adjusted with training. There are several reasons for gait training as paying attention to gait when running can prevent injuries or pain. Both achilles tendinopathy, a condition that causes pain in the achilles tendon, and patellofemoral pain syndrome, also known as *runner's knee*, can be reduced with gait training [1], [2]. It can also be used in rehabilitation of patients that have reduced walking ability due to stroke [3].

Today, gait analysis can be done in several ways, with image processing, sensors worn on the body or sensors placed on the floor being the most common methods [4]. Most gait analyses are used for clinical studies or rehabilitation in a controlled environment. Gait analysis performed with image processing or sensors on the floor demands special equipment and is often immobile. Gait analysis that can be used by recreational runners is usually done at places that sell running shoes by videotaping the runner [5] or in labs where the pressure under the feet can be measured [6]. These gait analyses are often one time examinations and cannot be used by runners every day to continuously improve their running technique.

1.1 Goal

The goal of this project was to start creating a system that could be used to improve gait for recreational or competitive runners and could be available for use anywhere and by anyone. This was done using *inertial measurement units*, IMUs, which was placed in a runner's shoes. The IMUs collected data about movements which was then processed to determine the air time of the runner wearing the shoes. The processing of the data was done in a processor close to the IMUs, so the raw data was not sent to an external unit. The system was designed to be small, light weight and, portable.

In order to be easy to use, the system was supposed to be able to operate for a fairly long time without having to be recharged and be possible to use anywhere. Finally, it should of course be able to measure air time fast enough so that the runner continuously gets updated about his/her gait.

One of the easiest ways to determine the air time for a runner in a lab would be to have the runner run on a floor with sensors, such as a force plate or a pressure mat and simply measure the time between the points where the runner makes contact with the force plate [4]. In this project, however, the goal was to build a system that could later be extended as well as be used anywhere, so an IMU was to be used even though it might not seem like the obvious choice for this purpose.

1.2 Delimitations

In gait analysis, many different kinds of data are needed and combined which will not be done in this project. This project will be a start that will use sensors to get air time of a runner. In the future, other data could possibly also be extracted from the raw data from sensors and more features could be included.

The focus of this project will be on the hardware and processor. The main idea is to determine, in real time, when a runner is in the air and not focus as much on the exactness of this determination. The algorithm and hardware decisions will both be taken with consideration of making the system small, ligh tweight and, with low power demand.

To make the system easy to use, some kind of feedback needs to be communicated to the user in an intuitive way, however this feature will not be considered during this project.

1.3 Ethical Considerations

In this project, there are a few ethical aspects that need to be taken into consideration. The future goal is that data about air time could be combined with other data into a gait analysis. This analysis would have the user as the main target audience but could possibly be shared with other people to compare different gaits and running styles. The data that is collected could possibly be sensitive information as it might be used to track people by recognizing the way they walk on video surveillance [7]. To be able to use the information in a suitable way, some additional information might be added by the user, such as age, length or weight and then more information could be extracted, for example, fitness level or what the terrain looks like. All information, added by the user, or extracted from the application could be considered sensitive information [8]. In this report and its corresponding project, these ethical considerations are irrelevant as it will not be used as a finished product but if that was the case, not only must the ethical aspect of privacy be taken into consideration but also exterior elements, such as the Swedish law [9].

2

Theory

In this chapter, the theory behind gait analysis will be presented as well as different algorithms and hardware that will be used to find gait events.

2.1 Gait

The way that a person walks can be described with the gait cycle. The movements a body performs between the point where one foot makes initial contact with the ground and the next time the same foot makes initial contact is called a gait cycle [10]. One half gait cycle, i.e. the movement between initial contact of one foot and initial contact of the other foot is called a stride. The gait cycle looks different depending on whether the person is running or walking [11].

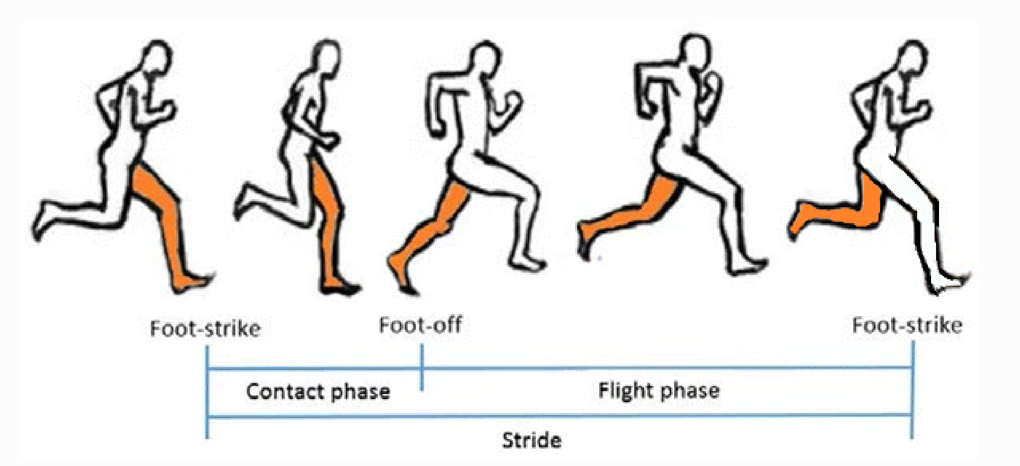


Figure 2.1: Picture of one stride (a half gait cycle) for a running person, with the left leg painted orange for clarification. Adapted from [12] with permission

As shown in Figure 2.1 the gait cycle and the stride can be described with different phases. The runner depicted to the left makes initial contact with the ground, this is called the *foot strike*, FS. The runner depicted in the middle pushes of the ground with the toes, the *foot off*, FO for the other foot. The movements until now is called ground contact phase. When the toes leave the ground, the runner is no longer touching the ground and is now in the flight phase until the heel(or some other part) of the runners other foot touches ground and the stride is repeated with the other foot. When both feet have had the ground contact phase with a corresponding flight phase, one gait cycle has passed [12].

2.1.1 Air time

Air time is defined as the time between FO for one foot and FS for the other foot, also known as the flight phase. The easiest way to determine the air time for one flight phase without having a pressure mat would be to detect the FO and the FS and then calculate the time between the two events.

2.2 Algorithms for gait event detection using wearable IMUs

There are several ways for gait event detection using IMUs. Some events can be found using only the accelerometer or the gyroscope or a combination of both [13]. Sensors can be placed anywhere on the body with placements on feet, lower legs, knees or hips being the most common. Often, some kind of threshold value is used to determine that an event has occurred, but it could also be that a maximum or minimum at the sensor output value is found [14]. Other events can then be detected using the events that have already occurred and the known facts about the gait cycle. When running, for example, there is always a mid swing between the FO and the FS.

Algorithms that use some kind of threshold value to detect an event often need the threshold value to be tested multiple times. Different runners have different gait cycles and the thresholds can vary a lot depending not only on the runner, but also outer aspects such as terrain or daily shape or fatigue. One way to compensate for these fluctuations is to use a machine learning algorithm that can find a relation between different values corresponding to events even when they vary [14].

FS is one of the easiest events to detect as it often corresponds to some sort of major peak in acceleration or angular velocity. The FO can often be found using a threshold or looking for another peak, sometimes in a time window related to the FS [15]. The fact that both FO and FS are much simpler to find than a more subtle event, such as the point where the whole foot is in contact with the ground, or different phases where the foot is in the air, makes the flight phase and thus the air time together with ground time one of the simpler gait attributes to compute.

For some events, different data need to be combined in order to determine when the event occurred. Different data that can be combined can come from multiple different sensors placed on different locations on the body. The data can also be saved data from earlier movements or knowledge about the other foot's placement in the gait cycle. The process of acquiring accurate information from a gait cycle can be memory and compute power intensive since a significant amount of information needs to be processed and stored.

In most studies performed on gait event detection, most or all data from the sensors has been stored and then analyzed after the data acquisition. The stored data can be processed for a long time using MATLAB or some other program and then outputted. To enable this, the sensors might need to be connected to a computer which provides power and data storage. This is often the case in studies as the runner often must run on a treadmill and be filmed or on floor sensors in order to get reference values to the tests.

Algorithms intended for mobile and low-power devices will be restricted in computing performance and storage space available limiting the information that can be processed without an external memory. If the algorithm must also be fast and output results continuously, there is no need to save as much data as the output should come almost immediately and then when the output is calculated, the old data is no longer interesting. Algorithms used outside labs are often predefined and have more general threshold values and methods than algorithms that are developed to be used in a laboratory controlled environment with predefined boundaries.

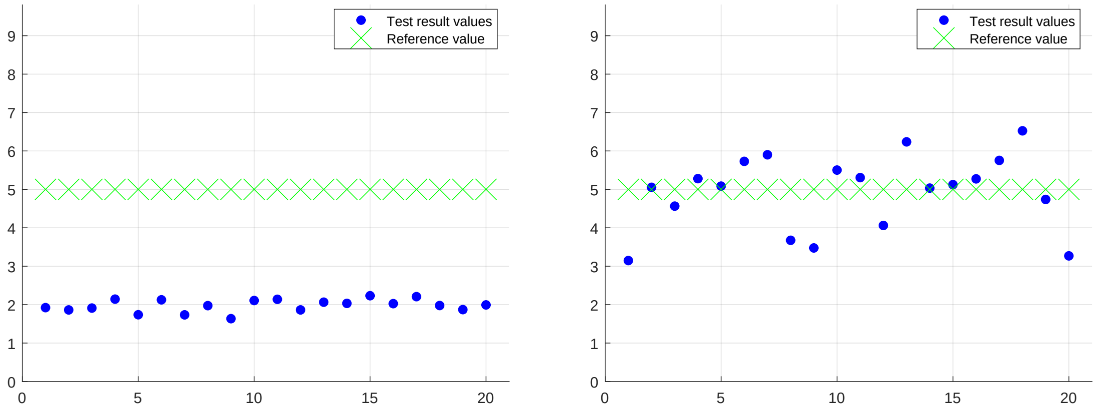
2.3 Precision VS accuracy

One way to compare results is to differentiate between precision and accuracy. By differentiating the two concepts, the results can be discussed in a more nuanced way.

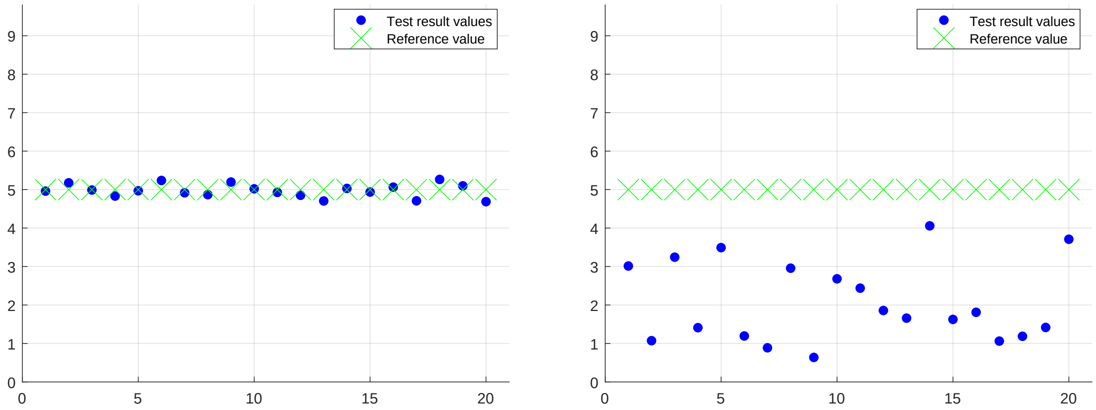
Precision can be described as the tightness of the results which result have higher precision if they are not scattered, but rather consistent even though they might not be close to the correct value [16]. Plots with high precision can be seen in Figure 2.2a and Figure 2.2c. A test with a high precision results could indicate high reliability of the results as it shows consistency. It could also indicate a systematic error in the validation process.

High accuracy is achieved when the result is close to the correct value and does not take the spread of the values into consideration. High accuracy could indicate that the correct values are being detected but that there are some random variances in the points being detected. This can be seen in Figure 2.2b.

The best case scenario can be seen in Figure 2.2c where both accuracy and precision are high. It is easy to see if accuracy and precision are high or low in these examples where you compare the results with a known correct reference value that does not vary over time. In reality, the determination of high and low accuracy and precision is not as easy to determine.



(a) High precision but low accuracy. **(b)** High accuracy but low precision.



(c) High accuracy and high precision. **(d)** Low accuracy and low precision.

Figure 2.2: Plots of the calculated and perceived ground time from the tests using one accelerometer

3

Development and evaluation

In this chapter, the methods for developing and implementing the algorithm will be explained as well as the used hardware. The test process will also be presented here.

3.1 Hardware

In this section, the hardware used in the project and the decisions behind implementation are presented.

3.1.1 IMU

Many accelerometers or gyroscopes sold today are already mounted on a inertial measure unit and, therefore, sold together. In the initial phase of the project, an IMU from Sparkfun was used [17] as it is a fairly well known IMU with good documentation and examples. For the purpose of the prototype developed in this particular project, a gyroscope or an accelerometer might have been enough, but with the hope of other features being developed in the future, an IMU was chosen with that in mind as it would not constrain data collection too much. The IMU is shown in Figure 3.1.

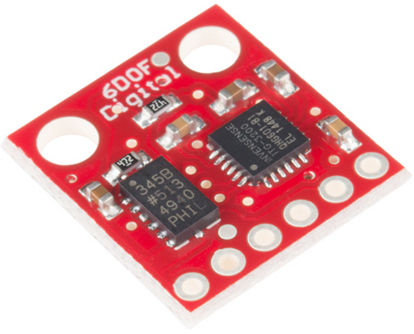


Figure 3.1: Picture of the used IMU with the accelerometer mounted on the left side of the PCB and the gyroscope the right (17x16mm LxW). From [17]. CC BY.

3.1.2 Accelerometer

The accelerometer on the IMU is an ADXL345 which has three axes and according to the datasheet low power and high resolution [18]. Typical current at a high operating speed is around 145 μA and all current consumption data provided by the supplier is shown in Table 3.1.

Table 3.1: ADXL345 current consumption's dependency on data rate. (T = 25 °C, $V_s = 2.5$ V, $V_{DD\ I/O} = 1.8$ V) From [18].

Output Data Rate (Hz)	Bandwidth (Hz)	Rate Code	I$_{DD}$(µA)
3200	1600	1111	145
1600	800	1110	100
800	400	1101	145
400	200	1100	145
200	100	1011	145
100	50	1010	145
50	25	1001	100
25	12.5	1000	65
12.5	6.25	0111	55
6.25	3.125	0110	40

The accelerometer also has a low-power mode with lower currents as shown in Table 3.2. If low-power mode is selected the output will have a somewhat higher noise.

Table 3.2: ADXL345 current consumption's dependency on data rate in low power mode. (T = 25 °C, $V_s = 2.5$ V, $V_{DD\ I/O} = 1.8$ V) From [19].

Output Data Rate (Hz)	Bandwidth (Hz)	Rate Code	I$_{DD}$(µA)
400	200	1100	100
200	100	1011	65
100	50	1010	55
50	25	1001	50
25	12.5	1000	40
12.5	6.25	0111	40

The accelerometer could be calibrated by using the fact that there should be 1 g pulling on the sensor at the Earth's surface [20]. A calibration that would provide eventual offsets for the sensor was considered but discarded as knowing the exact number of g-forces acting in a certain direction was not critical for this implementation. The acceleration was to be compared with earlier readings only and the exactness of the actual values was not of interest.

3.1.3 Gyroscope

The gyroscope used is an ITG3200 with three axes and low power for long battery life [21]. The gyroscope is significantly more power consuming than the accelerometer with a normal operating current of 6.5 mA and sleep mode current of 5 µA, the voltage supply is 2.1 – 3.6 V.

Just like the accelerometer, the gyroscope can be calibrated. The gyroscope has a temperature-dependent drift where a potential calibration should be performed at the same temperature as it will be used in. Once again, the exactness of the outputted data is not really important as when the maxima and minima occur are interesting and not the values. Also the algorithms utilizes the acceleration that is not sensitive to the drift instead of the angle output that depends largely on the drift. By not storing any offset data and by abstaining from calculating the exact values we could save some processor power and memory space.

3.1.4 Microcontroller

There are many microcontrollers to choose from on the market, with different areas of use. When designing a proof of concept or test project, development boards are usually used. The development boards consist of a microcontroller with connections to different pins so that wires can be connected and disconnected without soldering. Most development boards contain other parts as well, such as a few LED lights that can be used to signal that the microcontroller works as intended or some other jumpers that are used when configuring the board.

The development boards offer different ranges of flash memory size, power dissipation, performance and size along with other features.

A final product would not be assembled using a development board but rather a microcontroller that could be mounted directly on a board without any supporting components, and the wires would be soldered together.

3.1.5 Development board

The development board used in the initial phase was an STMicroelectronics NUCLEO-F401RE, shown in Figure 3.2. Just like the IMU, there were already simulink models that used this board together with the IMU provided. STM has a big board family with many boards with different specialties that can use similar configurations. The fact that it would be easy to change board made this family a good choice as different boards could be used throughout the project. Consat has previously worked with these boards and STM provides a program, STM32CubeMX, that can be used to configure pins as well as examples to get you started so help and guidance would be available when choosing this board.

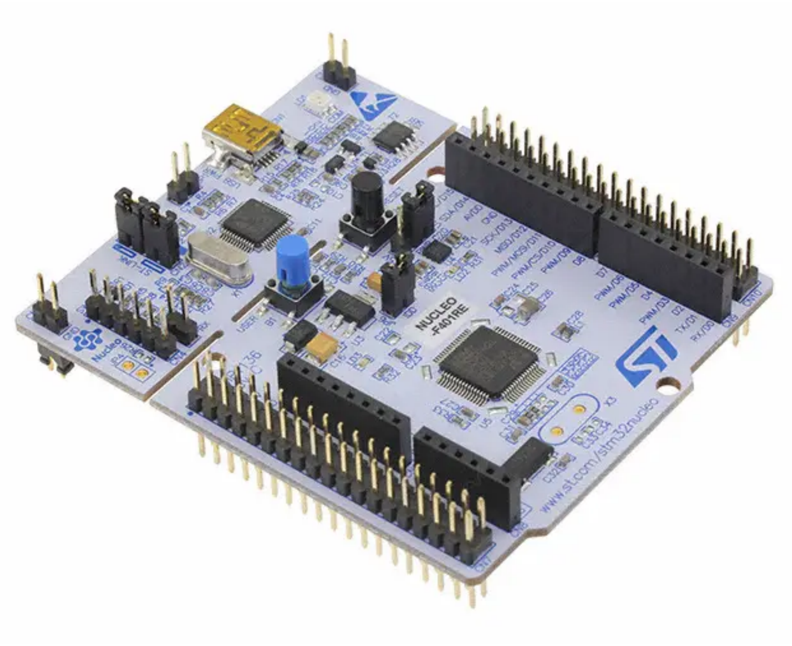


Figure 3.2: The NUCLEO-F401RE from STMicroelectronics. (82.5x70mm LxW)

The board might be switched for a smaller board with less flash memory size and, according to STMicroelectronics, "ultra low power".

3.1.6 Expansion board for power consumption measurement

An expansion board from STMicroelectronics was used to measure power consumption. The expansion board was a STM Power shield, X-NUCLEO-LPM01A.

3.2 Communication to hardware

Several communication protocols have been used in the project. There has been a need for communication with both the development board, with the processor as well as with the sensors, placed on another board.

3.2.1 Computer to processor

In order to communicate with the board from STMicroelectronics, universal asynchronous receiver/transmitter, UART, has been used. In the development stage, the initial code for communicating with the sensors, the algorithm and later adjustments have been sent using UART and a universal serial bus, USB, cable from the computer. The connection has then been used to observe data and calculate results from the board. The reason that UART has been used is that it was easy to implement using the provided program from STMicroelectronics, STM32CubeMX. In a final product, the board would not have to be connected through a cable to the computer rather a wireless connection would be used instead to output data so the choice of communication protocol would not impact the design of the final product.

3.2.2 Processor to sensors

For the communication between the processor and the sensors, inter integrated circuit, I^2C, was used. The I^2C protocol only requires two wires for the signals, serial data, SDA and serial clock, SCL [22]. The communication is implemented using *fast mode* with a data rate of 400 kbit/s.

The reason for using I^2C was mainly because the sensors are connected and mounted on a breakout board that only uses I^2C and has SDA and SCL as output pins. STM32CubeMX could be used to easily configure corresponding SDA and SCL pins for the processor.

However, I^2C is both slower and more power consuming than serial peripheral interface, SPI, which is also a communication protocol supported by the two sensors [22]. For the intended purpose of this project, SPI might seem more suitable but if the communication protocol would be changed to SPI, more wires would be needed and it would be more complicated to read and send data from the two sensors simultaneously.

3.2.3 Bluetooth connection

To make testing of the project easier, a Bluetooth module was used to send values to a computer. The Bluetooth module enabled testing of the system where a runner did not have to be physically connected to a computer, which made the system much more mobile. The Bluetooth module used was a HC-05 [23]. Calculated data about ground time or air time could be sent to a computer or mobile phone that was connected to the module.

3.3 Algorithm for event detection

Two or three different algorithms were initially developed for extracting air time. It is possible to calculate air time using only an accelerometer or a gyroscope, so for simplicity, algorithms were used that need data from one sensor only.

3.3.1 Algorithm using gyroscope

There are methods to detect gait events using only the gyroscope [24]. The method implemented is semi-real time as it needs to save data from part of a gait cycle before being able to detect certain events. The delay however, will be short enough so it would not be noticeable to a runner using the product.

The algorithm finds FO and FS by locating local minima of angular velocity and one local maximum as can be seen in Figure 3.3. The maximum represents the mid swing and the minima can then be identified as foot-off or foot-strike depending on whether they occur before or after the mid swing, hence the data about possible foot-off events needs to be stored until the mid swing is found and the FO is confirmed [24].

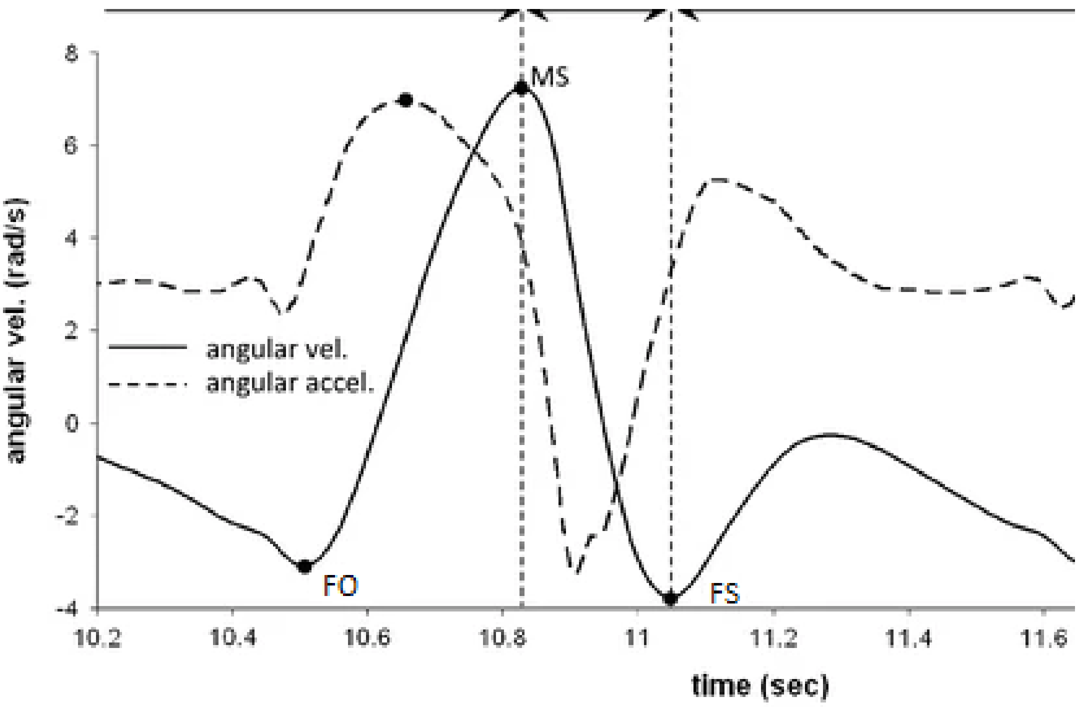


Figure 3.3: Picture of angular velocities from the gyroscope used in the algorithm. Reprinted by permission from Springer Nature Customer Service Centre GmbH: Springer Nature Medical & Biological Engineering & Computing [24] (Quasi real-time gait event detection using shank-attached gyroscopes, Jung Keun Lee et al), [COPYRIGHT] (2011)

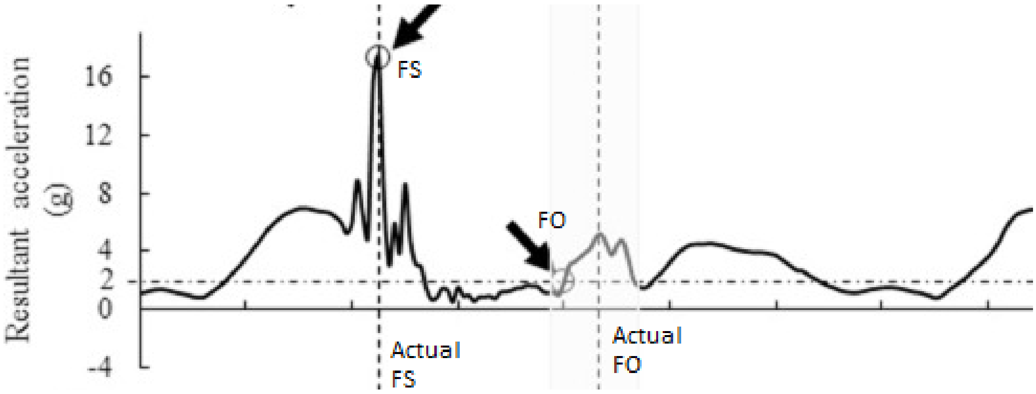
3.3.2 Algorithm using accelerometer

The algorithm used with the accelerometer was a combination of two different, previously tested algorithms as they seemed to be good at detecting different events [25]. The first method was used to determine FS. The FS finding method uses data from the total acceleration and determines the resultant magnitude from the different axis using equation 3.1.

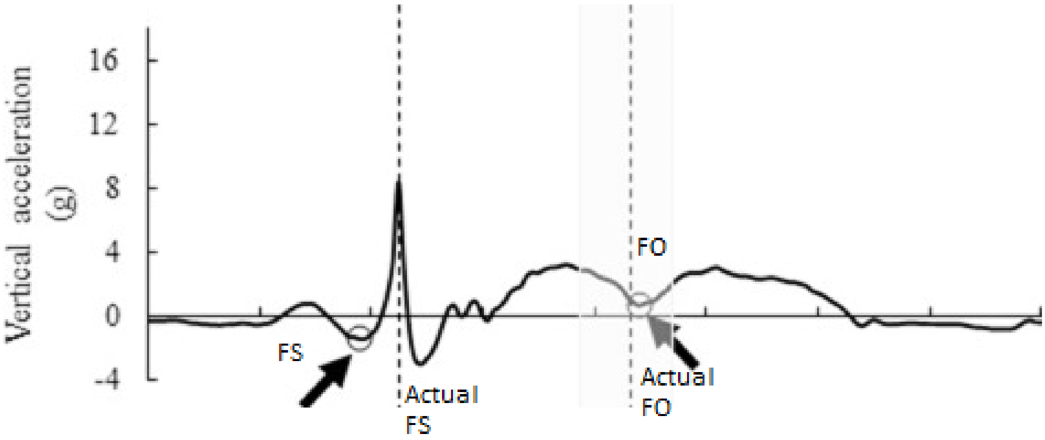
$$|a_{res}| = \sqrt{a_x^2 + a_y^2 + a_z^2} \tag{3.1}$$

The FS is then determined by finding the peak for the resultant [26] as seen in Figure 3.4a.

The second algorithm, which was used to determine FO, uses data from the vertical acceleration only and defined the toe off event as the minimum acceleration in the region of interest. This region is found at the end of the gait cycle. There are two distinct peaks of minimum acceleration, one immediately after FS and one later in the cycle, the later one is the peak of interest. [27] and can be seen in Figure 3.4b. In order to find the right peak, a time region was defined after finding FS that could be used as a guide to where to look for the FO.



(a) Picture of resultant acceleration.



(b) Picture of vertical acceleration.

Figure 3.4: Pictures of the extreme points that the acceleration based algorithm is detecting. From [25].

3.4 System setup

The system consists of a processor mounted on a development board and two IMUs as described in Figure 3.5. Two white cables were used for communication between the microcontroller and the sensors. Output was sent to the display from the microcontroller using Bluetooth.

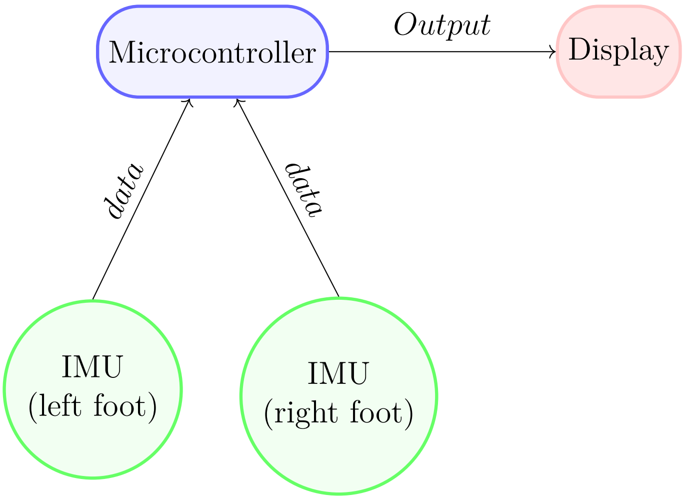


Figure 3.5: Setup of the system, including two IMUs, one microcontroller and some sort of display

3.5 Test process

The original idea was to test the system in a professional setup with help from a company that works with gait analysis. That way, a reference air time could be determined and then compared to the air time outputted from the system. Unfortunately, due to unpredictable external factors, these tests could not be carried out.

Some initial tests were done earlier in the project to assure that the system could detect some peaks in acceleration and match it to some gait event. The resolution and correctness however, was not tested at that stage and in order to verify the final implementation, additional tests needed to be conducted.

3.5.1 Measuring ground time for one foot using one sensor

At first a test to confirm the ability to detect ground time for one foot was performed. The theory was that if ground time for one foot could be detected correctly, by detecting the foot off and foot strike events, the same events could be detected for the other foot using the same algorithm and then the air time could be derived.

A runner, wearing the sensor on one foot was filmed while running on a treadmill. The sensor was placed in the shoe with the x-axes parallel with the foot movements. A computer, displaying the calculated ground time from the sensor was also visible in the filmed material. The authors watched the filmed material in slow motion and tried to decide at which times the different events occurred. The ground time was then calculated from the time difference of the perceived events. The reason for comparing the ground time and not the event times directly was because it was not possible to get a time stamp from the microcontroller for the events that would mach the video. Figure 3.6 shows a screenshot from one of the videos where the tests were conducted.

Figure 3.6: Screen shot from video used to test the algorithms

The frame rate for the video was 240 frames per second which would give a reasonable high resolution to the perceived ground time. The perceived event detection is however based on the authors judgement of the video material which could be very subjective.

This test was conducted with different update frequencies for the sensor as the update frequency has previously been stated as a parameter that affects the power consumption. It was also likely that the update frequency could affect the correctness of the system.

3.5.2 Measuring air time with both sensors implemented

With both sensors were implemented the air time could be calculated. The test to validate the air time calculations was similar to the tests used to test ground time. The calculated air time was sent to the terminal and then compared to the perceived air time from the video.

3.5.3 Update frequency vs power consumption

An expansion board was used to measure the power consumption of the system. During these test, both the data frequency from the sensors and the clock frequency in the microcontroller were changed. The sensor was once again placed in the shoe and the output was observed on the expansion board while the runner was running wearing the sensor.

The Results from all these test will be shown and compared in chapter 4.

4

Results

This project resulted in a system developed to measure air time for a runner. Even though the system was not designed to be used in the field yet, it was fairly comfortable to run wearing the system. The IMUs were placed in the shoes without too much consideration about placement even though the runner tried to align them in a somewhat correct way. The IMUs were connected to the microcontroller through cables which did affect the running experience negatively. The output of the system was shown on the terminal of an external device, a computer in this case, which was easy to see when running on a treadmill. The output was sent over a wireless connection which did not disturb the running experience. If the system was to be used anywhere else than on a treadmill, a more mobile device would be preferable, such as a mobile phone or a watch.

The system was fairly comfortable to wear even though some adjustments can be made to make it better.

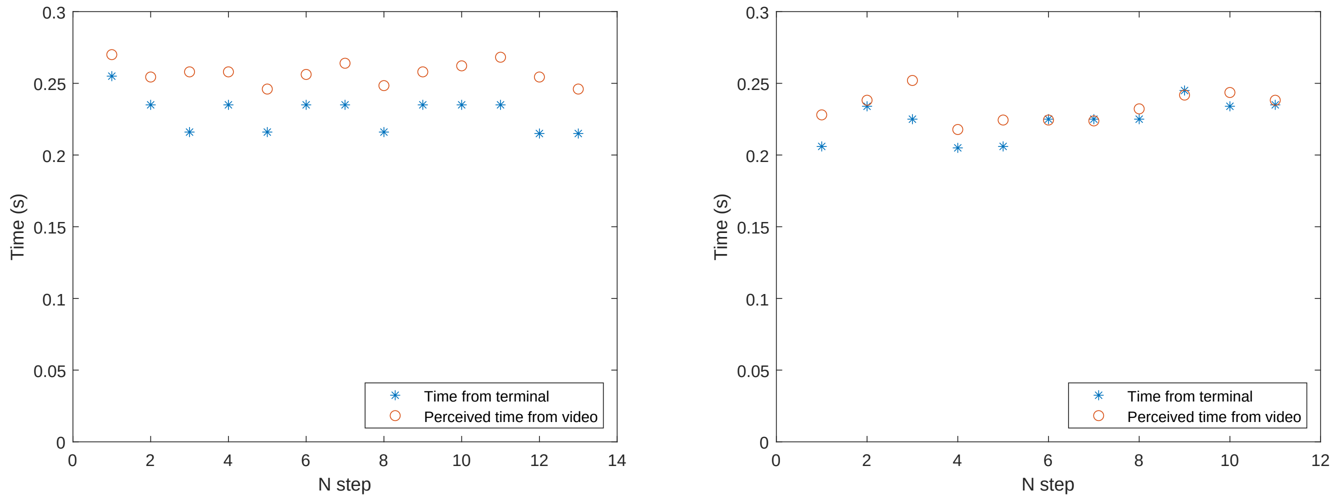
4.1 Verifying the function of the system

The function of the system was tested in the way presented in section 3.5. The test with two gyroscopes implemented was not conducted as the results from the first test were bad enough for the implementation to be considered a failure.

4.1.1 Measuring ground time for one foot using one accelerometer

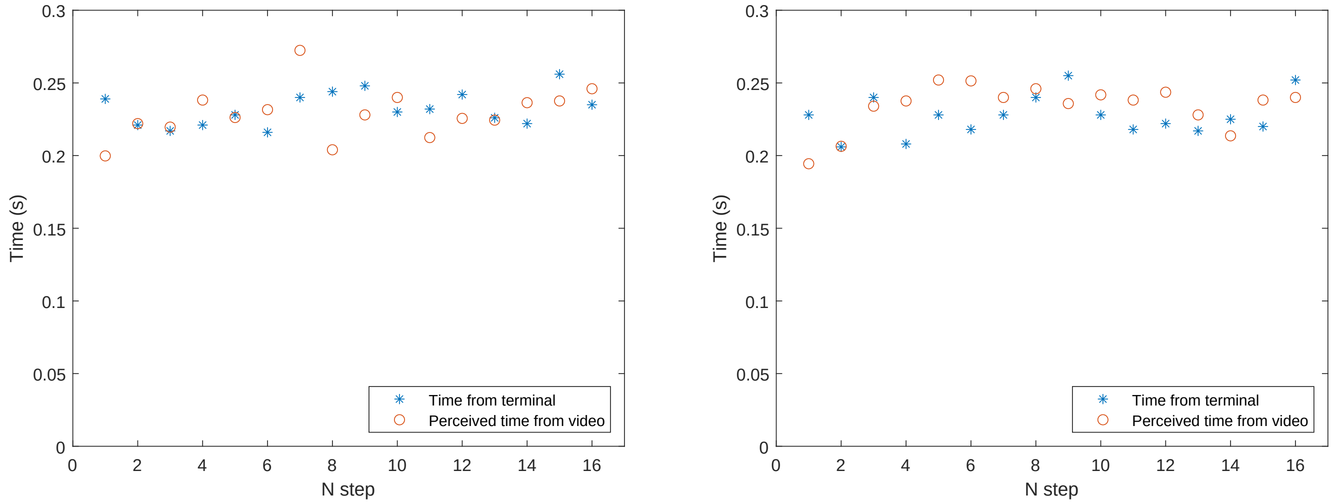
The ground time was calculated and compared to the perceived ground time for four different cases. The difference between the tests was that the data rate of the accelerometer was changed. The system was implemented using a data rate for the accelerometer of 50 Hz, 100 Hz, 200 Hz, and 400 Hz.

Plots of the calculated ground time, outputted to a terminal and the perceived ground time are presented in Figure 4.1. The times and the difference are also presented in Table 4.1. In the table, the difference in percent is always positive and the average error rate is based on the positive numbers. Note that the average difference in time can be a bit misleading as the positive and negative differences can compensate each other.



(a) Data rate of 50 Hz.

(b) Data rate of 100 Hz.



(c) Data rate of 200 Hz.

(d) Data rate of 400 Hz.

Figure 4.1: Plots of the calculated and perceived ground time from the tests using one accelerometer

Table 4.1: Tables of the data points of ground time for different data frequencies

Time from terminal (s)	Perceived time from video (s)	Difference (s)	Difference (%)
0.255	0.270	-0.015	5.6
0.235	0.254	-0.019	7.5
0.216	0.258	-0.042	16.3
0.235	0.258	-0.023	8.9
0.216	0.246	-0.030	12.2
0.235	0.256	-0.021	8.2
0.235	0.264	-0.029	11.0
0.216	0.248	-0.032	12.9
0.235	0.258	-0.023	8.9
0.235	0.262	-0.027	10.3
0.235	0.268	-0.033	12.3
0.215	0.254	-0.039	15.4
0.215	0.246	-0.031	12.6
Average Difference		-0.028	10.9

(a) 50 Hz

Time from terminal (s)	Perceived time from video (s)	Difference (s)	Difference (%)
0.206	0.228	-0.022	9.6
0.234	0.238	-0.004	1.7
0.225	0.252	-0.027	10.7
0.205	0.218	-0.013	6.0
0.206	0.224	-0.018	8.0
0.225	0.224	0.001	0.4
0.225	0.224	0.001	0.4
0.225	0.232	-0.007	3.0
0.245	0.242	0.003	1.2
0.234	0.244	-0.01	4.1
0.235	0.238	-0.003	1.3
Average Difference		-0.009	4.2

(b) 100 Hz

Time from terminal (s)	Perceived time from video (s)	Difference (s)	Difference (%)
0.239	0.200	0.039	19.5
0.221	0.222	-0.001	0.5
0.217	0.220	-0.003	1.4
0.221	0.238	-0.017	7.1
0.228	0.226	0.002	0.9
0.216	0.232	-0.016	6.9
0.240	0.272	-0.032	11.8
0.244	0.204	0.040	19.6
0.248	0.228	0.020	8.8
0.230	0.240	-0.010	4.2
0.232	0.212	0.020	9.4
0.242	0.226	0.016	7.1
0.226	0.224	0.002	0.9
0.222	0.236	-0.014	5.9
0.256	0.238	0.018	7.6
0.235	0.246	-0.011	4.5
Average Difference		0.001	6.4

(c) 200 Hz

Time from terminal (s)	Perceived time from video (s)	Difference (s)	Difference (%)
0.228	0.194	0.034	17.5
0.206	0.206	0.000	0.0
0.240	0.234	0.006	2.6
0.208	0.238	-0.030	12.6
0.228	0.252	-0.024	9.5
0.218	0.251	-0.033	13.1
0.228	0.240	-0.012	5.0
0.240	0.246	-0.006	2.4
0.255	0.236	0.019	8.1
0.228	0.242	-0.014	5.8
0.218	0.238	-0.02	8.4
0.222	0.244	-0.022	9.0
0.217	0.228	-0.011	4.8
0.225	0.214	0.011	5.1
0.220	0.238	-0.018	7.6
0.252	0.240	0.012	5.0
Average Difference		-0.007	7.3

(d) 400 Hz

When the data rate was set at 50 Hz, the ground time calculated in the system was consistently lower than the time perceived from the videotape as seen in Figure 4.1a. According to Table 4.1a, the average error is about 10.9 %.

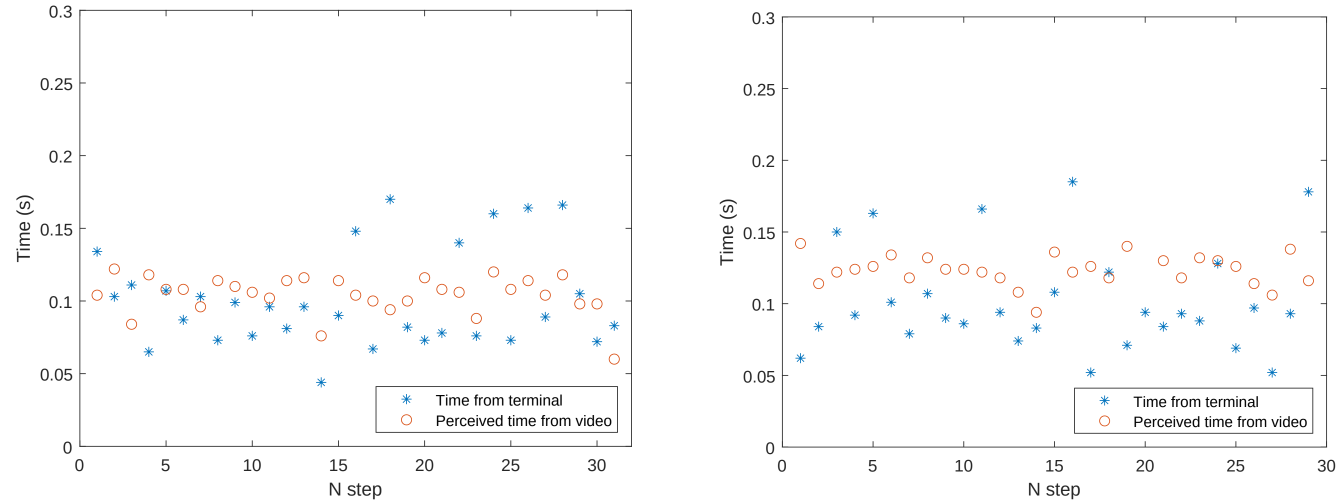
The test performed with a data rate of 100 Hz gives the most correct results as seen in Figure 4.1b. With an average difference of 4.2 % according to Table 4.2b there is not a noticeable consistent error in any direction that can bee seen on Figure 4.1b.

Both the 200 Hz and the 400 Hz are similar to the 100 Hz but as can be seen on Figure 4.1c and Figure 4.1d the values are more scattered but the average difference in seconds is close to zero according to table 4.1c and Table 4.1d. The 200 Hz had an average error of 6.4 % and the 400 Hz had an average error of 7.3 %.

Another thing that was observed during this test was that the calculated ground time appeared on the computer screen before the foot had landed on the ground again, about 0.15 – 0.2 s after the foot off event.

4.1.2 Measuring air time with both accelerometers implemented

The tests conducted with both accelerometers implemented showed reasonable results with an update frequency of 50 Hz and 100 Hz. These results will be showed below, and the tests where an update frequency of 200 Hz respective 400 Hz was used will not be presented as the conclusion of them not working could be drawn.



(a) Data rate of 50 Hz.

(b) Data rate of 100 Hz.

Figure 4.2: Plots of the calculated and perceived ground time from the tests using one accelerometer

Table 4.2: Tables of the data points of Air time for the two different data frequencies

Time from terminal (s)	Perceived time from video (s)	Difference (s)	Difference (%)
0.062	0.142	-0.08	56.3
0.084	0.114	-0.03	26.3
0.15	0.122	0.028	23
0.092	0.124	-0.032	25.8
0.163	0.126	0.037	29.4
0.101	0.134	-0.033	24.6
0.079	0.118	-0.039	33.1
0.107	0.132	-0.025	18.9
0.09	0.124	-0.034	27.4
0.086	0.124	-0.038	30.6
0.166	0.122	0.044	36.1
0.094	0.118	-0.024	20.3
0.074	0.108	-0.034	31.5
0.083	0.094	-0.011	11.7
0.108	0.136	-0.028	20.6
0.185	0.122	0.063	51.6
0.052	0.126	-0.074	58.7
0.122	0.118	0.004	3.4
0.071	0.14	-0.069	49.3
0.094	0.352	-0.258	73.3
0.084	0.13	-0.046	35.4
0.093	0.118	-0.025	21.2
0.088	0.132	-0.044	33.3
0.128	0.13	-0.002	1.5
0.069	0.126	-0.057	45.2
0.097	0.114	-0.017	14.9
0.052	0.106	-0.054	50.9
0.093	0.138	-0.045	32.6
0.178	0.116	0.062	53.4
Average Difference		0.029	27.7

(a) Data rate of 50 Hz

Time from terminal (s)	Perceived time from video (s)	Difference (s)	Difference (%)
0.206	0.228	-0.022	9.6
0.234	0.238	-0.004	1.7
0.225	0.252	-0.027	10.7
0.205	0.218	-0.013	6.0
0.206	0.224	-0.018	8.0
0.225	0.224	0.001	0.4
0.225	0.224	0.001	0.4
0.225	0.232	-0.007	3.0
0.245	0.242	0.003	1.2
0.234	0.244	-0.01	4.1
0.235	0.238	-0.003	1.3
0.206	0.228	-0.022	9.6
0.234	0.238	-0.004	1.7
0.225	0.252	-0.027	10.7
0.205	0.218	-0.013	6.0
0.206	0.224	-0.018	8.0
0.225	0.224	0.001	0.4
0.225	0.224	0.001	0.4
0.225	0.232	-0.007	3.0
0.245	0.242	0.003	1.2
0.234	0.244	-0.01	4.1
0.235	0.238	-0.003	1.3
0.206	0.228	-0.022	9.6
0.234	0.238	-0.004	1.7
0.225	0.252	-0.027	10.7
0.205	0.218	-0.013	6.0
0.206	0.224	-0.018	8.0
0.225	0.224	0.001	0.4
0.225	0.224	0.001	0.4
0.225	0.232	-0.007	3.0
0.245	0.242	0.003	1.2
0.234	0.244	-0.01	4.1
0.235	0.238	-0.003	1.3
0.206	0.228	-0.022	9.6
0.234	0.238	-0.004	1.7
0.225	0.252	-0.027	10.7
0.205	0.218	-0.013	6.0
0.206	0.224	-0.018	8.0
0.225	0.224	0.001	0.4
0.225	0.224	0.001	0.4
0.225	0.232	-0.007	3.0
0.245	0.242	0.003	1.2
0.234	0.244	-0.01	4.1
0.235	0.238	-0.003	1.3
Average Difference		-0.009	4.2

(b) Data rate of 100 Hz

4.1.3 Measuring ground time for one foot using one gyroscope

It was obvious from the start of the test that the implemented algorithm using the gyroscope did not work as the output values varied significantly even when the runner was running at a constant speed. When the calculated values were somewhat consistent, they were unrealistically large and nowhere near the correct values.

4.2 Power consumption

In this chapter, the power consumption of the system for calculating air time using accelerometers will be presented. A power bank was used to provide power to the system during testing. This was done to increase mobility and therefore make the tests easier to perform. It also proves that even the non optimized prototype could be run using a portable battery.

4.2.1 Refresh rates of sensor vs power consumption

The power consumption for the whole system depending on the update frequency is presented in Table 4.3. The system was tested for power consumption at 50 Hz and 100 Hz as these were the only frequencies that would give an output when measuring air time. A small button cell battery with 660 mW h [28] would power the microcontroller for over 8 hours excluding the losses that would come from a voltage regulator.

The microcontroller enters sleep mode after an update of the values and exits the sleep mode after an interrupt. The interrupts from the accelerometers will therefore affect how often the microcontroller exits the sleep mode. The sleep mode is the least disruptive power-saving tool and will keep all registers and states.

The two IMUs were responsible for a large portion of the power consumption. The combined power consumption of both IMUs is 39.79 mW. And at a current of 12.13 mA.

Table 4.3: Power consumption

Update frequency (Hz)	Current (mA)	Voltage (V)	Power consumption (mW)
50	24.43	3.28	80.13
100	24.64	3.28	80.82

4.2.2 Clock frequency for microcontroller vs power consumption

The clock frequency of the microcontroller was changed using software. Changing the clock frequency did not have any measurable effect on power consumption. The range of tested frequencies was from 84 MHz down to 5.25 MHz.

5

Discussion

In this section, the results presented earlier will be discussed. Possible reasons behind the results and problems during testing will be brought up. The function of the system and how it could be implemented to make a final product will also be assessed.

5.1 Gyroscope

It was decided after the first ground time tests of the gyroscope to not try to implement the air time measuring algorithm. This decision was taken because the function of the implemented part of the algorithm did not meet the expectations. Therefore an algorithm for air time based on the algorithm would not have been able to generate a reasonable value for the air time. The reason that we could not make the ground time algorithm function with the gyroscopes was not known. One problem with the algorithm was that it did not have a clear starting point like the accelerometer-based algorithm has, where a high peak could be easily found with an adjustable threshold. The algorithm used was also vulnerable to fluctuations and outliers, as a stray value could be determined at an extreme value even though the value was not reasonable. This could be corrected with some kind of filter but that would require additional processing power.

As the gyroscope algorithm relied on local extreme points it was hard to know what part of the algorithm was functioning without having a plot of the gyroscope values synced to a slow-motion video. This was not possible to do with the test equipment that was available. A plot could be generated but with a delay, that made it impossible to sync as the gait events are very short and the delay varies, which would make it impossible to match the plot to reality.

5.2 Uncertainties

There were several unforeseen problems that made the implementation and evaluation difficult. There are also several things that could possibly have gone wrong during the project, some which might have not even been noticed.

5.2.1 Problems during testing

The original plan for the project involved more detailed testing of the final algorithm and implementation. These tests were planned to be performed at a company that works with motion control and gait analysis primarily using cameras. A test subject would use the system in the test environment and the results could be compared to the "correct" results. This way of testing the function would most likely be more accurate than the method that was used in the project. Unfortunately, the Corona pandemic spread over the world and many companies closed down or permitted their staff which meant that no such test could be carried out.

The tests relied heavily on the author's perception and judgement which could make the tests vulnerable to confirmation bias. In order to prevent the authors from choosing to detect events at a time that match the output to terminal, the event detection performed by the authors was always conducted before looking at the results from the terminal. However, the event times could still be affected by earlier observations and event times that seem familiar or time differences that feels consistent could have been favored over times that would have been chosen with no bias.

The tests were performed with a recreational runner. The final product should work for all types of runners. But it might have been easier to perform the tests and evaluation if a professional runner was used as the gait cycle would be more consistent.

5.2.2 Possible errors that have not been detected

There are many things in this project that might not work as intended. One risky assumption that was made early on was that the accelerometer and gyroscope had been correctly implemented and worked as intended. It is quite hard to verify the correctness of the sensors as there were no correct reference values available. Multiple sensors were tested that all gave the same output so any possible errors should be in the code. The assumptions that the sensors worked as they should was done based on the fact that the data outputted seemed reasonable and there were no prominent reason to believe that the hardware would be faulty.

When the processor pins were connected to the sensor pins through a longer cable for the first time the system stopped working. The wires were then rearranged and the SDA and SCL wires were placed as far apart in the cable as possible, which led to the system working as before. The fact that the system seems to be quite vulnerable to how the wires are placed relative each other might suggest that the wires could possibly still impact each other even though the system gives a reasonable output.

One thing that has not been tested during this project is the ability to detect the FO and FS events. The ground time and air time has been tested and in these tests, only the time passed between the events has been tested which means that a correct result could not rule out the possibility that the event detection was off. The event detection could be equally erroneous for both events. An incorrect result could also mean that the detection works well for one of the events but not for the other one.

Lower sampling frequency should logically give worse results. Optimally, an event should happen as close before sampling as possible for best possible resolution. With longer time between the samples, the uncertainty of when an event has occurred increases as can be seen in Figure 5.1

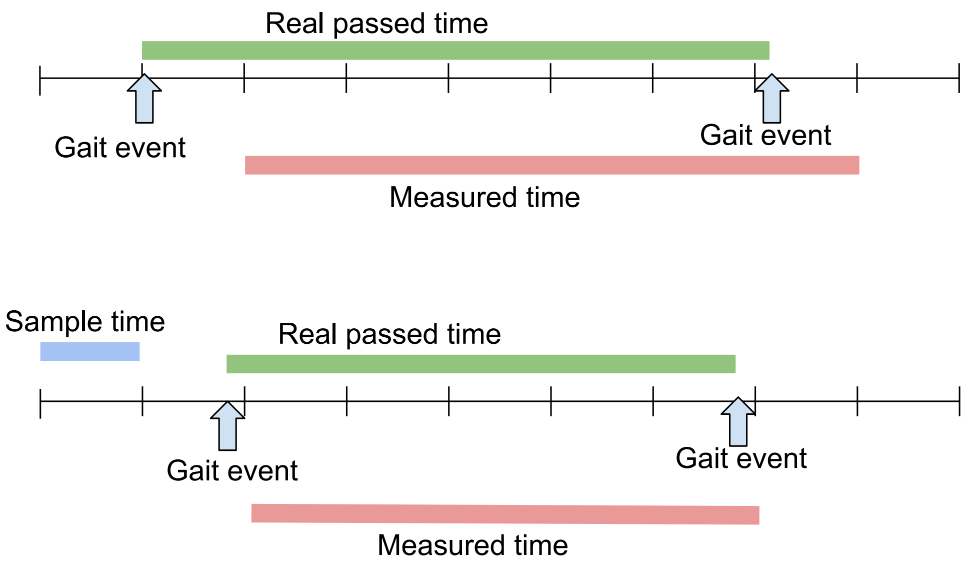


Figure 5.1: Example of how mismatched sampling can affect the results

With an update frequency of 50 Hz, the runner in our tests are in the air for about 5 samples and when the frequency is increased to 100 Hz for about 10 samples. An event that occur immediately after one sampling but is not registered until the next sampling could therefore lead to an error in the air time of up to 20% respectively 10%.

5.3 Function of the system

Only the algorithm using the accelerometer will be discussed here as the results from the gyroscope did not show any evidence of working.

The ability to measure ground time for a runner was tested with varying results depending on the update frequency of the accelerometer. All of the updating frequencies could be run without errors. One reason for the varying results could be that the author's perception varied when they judged if the different events had occurred.

5.3.1 Ground time

When the ability to measure ground time was tested, the test with the lowest update frequency had the most error. This would not be considered very surprising as a higher update frequency can provide a more exact timing to the processor. The fact that the errors were consistent in sign and also fairly consistent in amplitude would suggest that this update frequency could be a good choice if something that would compensate for the errors could be implemented. The plots show a high precision but low accuracy. In the future, an algorithm that compensated for this could be implemented and more thoroughly tested, with more steps and different speeds to determine whether this could be a good data rate for the accelerometer.

The remaining three tests looked more like each other and became more correct as the time between value updates were lower. There is no obvious reason for this and it might just be due to the fact that the tests are not very precise and the accuracy is a coincidence.

The high consistency and precision for the results with an update frequency of 50 Hz suggest that this would be the best option even though the accuracy is not that high.

5.3.2 Air time

The results from the air time tests should have the same errors as the ground time tests, both algorithms are based on the same gait event detection algorithms as the ground time algorithm. The relative errors are larger because of the much shorter air time compared to ground time.

The accuracy of both the 50 Hz and 100 Hz refresh rates seems to be high as can be seen in Figure 4.2. Both of them might have a slight offset where they are centered around a lower value with some outliers with higher values. Both 50 Hz and 100 Hz refresh rates have a low precision in relative terms. The variation between all the air times has higher precision than the precision from the algorithm.

The results from the air time tests show that the higher refresh rates do not improve the function. That would suggest that the function of the algorithm is the limiting factor for improved precision and accuracy.

There are some improvements that could have been implemented without fundamentally changing the algorithm. The most obvious parameter to change is the threshold for initial contact. Another parameter that could be changed is the foot off region of interest. Reducing the region of interest could improve the function by minimizing the chance of detecting a minimum that does not correspond to the FO. The downside of reducing the region could be that it would also reduce the range of different lengths between gait events that could come from different running techniques. It is also more important that the region of interest starts and ends at the correct time when the window is smaller.

All of the discussed parameters have been changed during testing and evaluation. To be able to ensure that the parameters are at the optimal value, better test equipment would be needed. For example, it would be useful to have some system that could synchronise a plot of acceleration values with gait events. The values for these parameters need to be correct before more advanced changes to the algorithm are implemented, otherwise more advanced changes could make performance worse.

5.3.3 Higher refresh rates

When two accelerometers were implemented, the algorithm would not work for higher refresh rates than 100 Hz. We could not figure out the exact reason for this. One theory is that when the refresh rates were high and not in sync, the interrupts from each accelerometer could come too close to each other. If one of the IMUs were updating while the other interrupt came in it could cancel the update or be ignored depending on priority. If it cancelled the update, then no more updates would work. If it was ignored, the interrupt flag would not be reset, which would also stop future updates.

5.4 Power efficiency optimization

The power consumption of a final product could be optimized. Even with no optimizations the micro-controller and IMUs could be run a few hours using a battery. During the tests, power was provided with a power bank.

That would be considered enough for most running applications. With reduced power consumption the battery could be smaller or allow for multiple running sessions. There are a few parts that would need special attention for reducing power consumption as they contribute to a majority of the power consumption, this will be discussed below.

One part of the system that was not included in the measurement of power consumption was the Bluetooth module. There are several reasons for this, the first reason is that it was not part of the scope to optimize for the power consumption of the transmission, therefore the Bluetooth was not chosen according to its power consumption, instead it was chosen for easy implementation. Another reason is that the Bluetooth was not essential for determining the function of the system, it was to simplify the test process.

5.4.1 Power consumption of microcontroller

When the clock frequency of the micro controller were changed, no difference in power consumption could be measured. This could be because the clock frequency were changed using software as opposed to changing the physical oscillator. Another reason for the clock frequency to have no effect on power consumption could be that the power consumption is already dominated by static power consumption. Reducing the dynamic power consumption would therefore not have any significant effect.

5.4.2 Power consumption for IMUs

Both the accelerometer and gyroscope that was used was advertised as "low power" by the manufacturers. In Figure 3.1 it is clear that the output frequency of the accelerometer affects the power dissipation to a high extent. In reality, this difference was negligible as the current of 12.13 mA of the IMU is far higher than the current advertised in Figure 3.1. The reason for this is that there are much more losses outside of just the current consumption of the accelerometer circuit board. The IMU is designed in a way that you have to power both the accelerometer and the gyroscope even if only one of them is used. There are also other components and traces on the IMU circuit board like resistors and capacitors that could increase power consumption. The long cables connecting the IMUs to the microcontroller could also be a major contribution to increased power consumption. The conductors were kept thin to increase mobility as there were five conductors to each IMU.

A final product would reduce or eliminate some of these problems. A final product would have a dedicated circuit board that would only use the components needed for function of the product. It could be possible to have the microcontroller next to the IMU and greatly reduce the length of the cables.

5.4.3 Refresh rates vs Power consumption

The power consumption did vary with the refresh rates of the accelerometer. The consumption was slightly higher with a refresh rate of 100 Hz than 50 Hz. But the difference as can be seen in Table 4.3 is only 69 mW. This difference would not be significant enough to use a lower refresh rate even if the higher refresh rate is only slightly better. If the power consumption of the system could be reduced by reducing overhead power consumption, the difference might be larger in relative terms.

There is no clear way to tell if the 100 Hz or 50 Hz updating frequencies has better accuracy and precision. The clear choice is therefore the 50 Hz updating frequencies as the power consumption is lower with no performance penalties. To be able to make a better choice of updating frequencies the algorithm needs to be improved.

5.5 Further research

The measurement of air time is the first of many features that could be implemented to reach a fully functioning device that can perform gait analysis. To continue the work performed in the project, usage of other important data, such as pressure or angle of the foot could be implemented. To be able to do this both accelerometer and gyroscope would probably be needed.

Some sort of user application could be developed that can provide the runner with updates about running technique and possibly analyze this data to log information about fitness level or how the runner answer to fatigue. A user application is probably a requirement for the product being desirable for runners that run in their spare time. The application could also help runners that suffer from previous injuries or other health problems and recognize returning symptoms before the runner does.

A final product could also have a gait event detection algorithm that would be adapted to the user. This could be done by having different threshold and foot off window depending on parameters like fitness level, weight or height. It could also use some kind of self calibration. Future research have to be carried out in order to determine the correlations between the parameters and thresholds.

5.5.1 Developing a real product

In this prototype, two IMUs and one microcontroller have been used, the IMUs have been connected to the microcontroller through a cable. For a product, users might not want a cable taped to the legs when running. To avoid the cable, the data should probably be sent wireless between, at least one of the IMUs, and the microcontroller. The microcontroller should probably be mounted together with one of the IMUs in order to minimize things that the runner needs to keep track of.

In this project, the microcontroller has been mounted on a development board with several other parts, which would not be needed in a finished product. The final product would be smaller than the hardware used in the project. The gyroscope ended up not being used so maybe the IMUs could be switched out as well but as the gyroscope probably will be needed if more features were to be developed, we decided to keep it.

The current prototype has not been optimized in terms of size. The final product could be considerably smaller than the prototype. The development board that was used measured 82.5x70mm, the real product would likely only use a handful of the components from the development board, for example, the microcontroller chip with the size of only 14x14mm.

The long cables that were used was needed because of the microcontroller development board was too large and heavy to be worn near the IMUs on the feet. With a reduced size, the microcontroller could be located at one foot. For this to work, the problem of transmitting data from the other IMU has to be solved.

If the algorithm was optimized higher updating frequencies than 100 Hz would be interesting for more accurate detection of gait events. A real-time operating system with deadlines would probably be needed to be able to have higher updating frequencies.

5.5.2 Further usage

In the case that this product could be further developed with more features and a user-friendly interface to present the data, the product could be used by not only runners but also other people who want to keep track of their movements, some people might be interested in how many steps they have taken and how many stairs they have climbed during the day.

A potential product could be used in race walking contests, one of the rules in the race walking is that the competitor can not lift his/her toes on one foot until he/she has made contact between the ground and the heel of the other foot. The product that has been developed in this product could be used to help the judges determine whether this rule has been broken by the competitors wearing the system. According to the international association of athletics federations rules, the judging should be solemnly based on the human eye judgment and no aids are allowed, but these rules might change in the future.

Wearable sensors that track the movements that are relevant in other areas can possibly be an aid for many training athletes such as swimmers or rowers who might not be aware of their movements being asymmetric or not ideal for some reason.

6

Conclusion

The developed system outputs a value that is somewhat close to the real air time. The fact that the air time is calculated soon after the runner has landed with a reasonable value gives us the impression that the air time is measured. The resolution, however, is not as good as we had hoped.

As the results do not improve when the update frequency of the sensors is increased, we can draw the conclusion that the the system is somewhat constricted by the algorithm. The algorithm recognizes the foot off and foot strike events.

In order to finalize a product that can measure air time in real time, the algorithm and its threshold values need to be tweaked. More tests, preferably with several different runners needs to be conducted in order to find better values to use in the algorithm.

With a better working algorithm, it becomes more interesting to increase the update frequency as the results should be noticeable in the test results.

The power consumption of the sensors is higher than stated in the data sheets. In the tests, a power bank was used which worked fine. A small battery would be enough power supply for many running sessions. The weight and size of the system is probably small and light enough for the usual jogger who wants to exercise. A professional runner who competes would probably not want to wear the equipment in a competition but maybe during training.

Bibliography

[1] J. L. Roper, E. M. Harding, D. Doerfler, J. G. Dexter, L. Kravitz, J. S. Dufek, and C. M. Mermier, “The effects of gait retraining in runners with patellofemoral pain: A randomized trial,” *Clinical Biomechanics*, vol. 35, pp. 14 – 22, 2016.

[2] Y. Thijs, D. De Clercq, P. Roosen, and E. Witvrouw, “Gait-related intrinsic risk factors for patellofemoral pain in novice recreational runners,” *British Journal of Sports Medicine*, vol. 42, no. 6, pp. 466–471, 2008.

[3] K. Tomida, S. Sonoda, S. Hirano, A. Suzuki, G. Tanino, K. Kawakami, E. Saitoh, and H. Kagaya, “Randomized controlled trial of gait training using gait exercise assist robot (gear) in stroke patients with hemiplegia,” *Journal of Stroke and Cerebrovascular Diseases*, vol. 28, no. 9, pp. 2421 – 2428, 2019.

[4] A. Muro-de-la Herran, B. Garcia-Zapirain, and A. Mendez-Zorrilla, “Gait analysis methods: An overview of wearable and non-wearable systems, highlighting clinical applications,” *Sensors*, vol. 14, no. 2, pp. 3362–3394, 2014.

[5] L. Bell, “Running tech: What is a gait analysis and why should every runner have one?.” *Forbes*, 2018-09-30. [Online]. Available: https://www.forbes.com/sites/leebelltech/2018/09/30/running-tech-what-is-a-gait-analysis-and-why-should-every-runner-have-one. [Accessed: 2020-02-06].

[6] “What are the various types of gait analysis?.” *Tekscan*, 2016. [Online]. Available: https://www.tekscan.com/blog/medical/what-are-various-types-gait-analysis. [Accessed: 2020-02-06].

[7] L. Wang, T. Tan, H. Ning, and W. Hu, “Silhouette analysis-based gait recognition for human identification,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, pp. 1505–1518, Dec 2003.

[8] M. Prakash and G. Singaravel, “An approach for prevention of privacy breach and information leakage in sensitive data mining,” *Computers & Electrical Engineering*, vol. 45, pp. 134 – 140, 2015.

[9] *7 kap. 1 §, Sekretesslag (1980:100)*. S, 1992. [Online]. Available: https://www.riksdagen.se/sv/dokument-lagar/dokument/svensk-forfattningssamling/sekretesslag-1980100$_sfs - 1980 - 100.[Accessed : 2020 - 02 - 06]$.

[10] S. A. Dugan and K. P. Bhat, "Biomechanics and analysis of running gait," *Physical Medicine and Rehabilitation Clinics*, vol. 16, no. 3, pp. 603–621, 2005.

[11] A. R. Anwary, H. Yu, and M. Vassallo, "An automatic gait feature extraction method for identifying gait asymmetry using wearable sensors," *Sensors*, vol. 18, no. 2, 2018.

[12] D.-K. Chew, K. J.-H. Ngoh, D. Gouwanda, and A. A. Gopalai, "Estimating running spatial and temporal parameters using an inertial sensor," *Sports Engineering*, vol. 21, pp. 115–122, 2018.

[13] J. M. Jasiewicz, J. H. Allum, J. W. Middleton, A. Barriskill, P. Condie, B. Purcell, and R. C. T. Li, "Gait event detection using linear accelerometers or angular velocity transducers in able-bodied and spinal-cord injured individuals," *Gait & Posture*, vol. 24, no. 4, pp. 502 – 509, 2006.

[14] M. M. Skelly and H. J. Chizeck, "Real-time gait event detection for paraplegic fes walking," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 9, pp. 59–68, March 2001.

[15] J. Leitch, J. Stebbins, G. Paolini, and A. B. Zavatsky, "Identifying gait events without a force plate during running: A comparison of methods," *Gait & Posture*, vol. 33, no. 1, pp. 130 – 132, 2011.

[16] D. L. Streiner and G. R. Norman, ""precision" and "accuracy": Two terms that are neither," *Journal of Clinical Epidemiology*, vol. 59, no. 4, pp. 327 – 330, 2006.

[17] "Sparkfun 6 degrees of freedom imu digital combo board - itg3200/adxl345." *Sparkfun*, 2015. [Online]. https://www.sparkfun.com/products/retired/10121. [Accessed: 2020-07-23].

[18] "Digital accelerometer adxl345." *Sparkfun*, 2009. [Online]. Available: https://www.sparkfun.com/datasheets/Sensors/Accelerometer/ADXL345.pdf. [Accessed: 2020-07-23].

[19] "Accelerometer, gyro and imu buying guide." *Sparkfun*, 2020. [Online]. Available: https://www.sparkfun.com/pages/accel$_{g}yro_{g}uide.[Accessed: 2020 - 01 - 28]$.

[20] B. Earl, "Programming and calibration for adxl345." *Adafruit*, 2020. [Online]. https://learn.adafruit.com/adxl345-digital-accelerometer/programming. [Accessed: 2020-01-28].

[21] "Itg-3200 product specification revision 1.4." *Sparkfun*, 2010-03-30. [Online]. Available: https://www.sparkfun.com/datasheets/Sensors/Gyro/PS-ITG-3200-00-01.4.pdf. [Accessed: 2020-07-23].

[22] F. Leens, "An introduction to i2c and spi protocols," *IEEE Instrumentation Measurement Magazine*, vol. 12, pp. 8–13, February 2009.

[23] "Hc-05 - bluetooth module." *Components 101*, 2018-03-10. [Online]. Available: https://components101.com/wireless/hc-05-bluetooth-module. [Accessed: 2020-06-02].

[24] J. K. Lee and E. J. Park, "Quasi real-time gait event detection using shank-attached

gyroscopes," *Medical & Biological Engineering & Computing*, vol. 49, pp. 707 – 712, 2011.

[25] S. Mo and D. H. Chow, "Accuracy of three methods in gait event detection during overground running," *Gait & Posture*, vol. 59, pp. 93 – 98, 2018.

[26] C. Strohrmann, H. Harms, C. Kappeler-Setz, and G. Troster, "Monitoring kinematic changes with fatigue in running using body-worn sensors," *IEEE Transactions on Information Technology in Biomedicine*, vol. 16, pp. 983–990, Sep. 2012.

[27] J. Mercer, B. Bates, J. Dufek, and A. Hreljac, "Characteristics of shock attenuation during fatigued running," *Journal of Sports Sciences*, vol. 21, no. 11, pp. 911–919, 2003. PMID: 14626370.

[28] "Batterilagret knappcell 3v 2032." *Batterilagret*, 2020. [Online]. https://www.batterilagret.se/p1535/knappcell-3v-2032. [Accessed: 2020-01-28].