МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ АВІАЦІЙНИЙ УНІВЕРСИТЕТ

Кафедра авіаційних комп’ютерно-інтегрованих комплексів

ДОПУСТИТИ ДО ЗАХИСТУ

Завідувач кафедри

Синєглазов В.М.

“ ____ ” __________2021.

ДИПЛОМНАРОБОТА

(ПОЯСНЮВАЛЬНА ЗАПИСКА)

ВИПУСКНИКА ОСВІТНЬО-КВАЛІФІКАЦІЙНОГО РІВНЯ

“БАКАЛАВР”

Тема: Навігаційна система числення шляху на основі мобільного телефону

Виконав: **Лазаревський О.А.**

Керівник: **к.т.н.** **Мухіна М. П.**

Нормоконтролер: к.т.н. **Тупіцин М. Ф.**

Київ – 2021

EDUCATION AND SCIENCE MINISTRY OF UKRAINE

NATIONAL AVIATION UNIVERSITY

COMPUTER-INTEGRATED COMPLEXES DEPARTMENT

ADMIT TO DEFENSE

Head of department

Viktor M. Sineglazov

“____” ______________ 2021.

BACHELOR WORK

(EXPLANATORY NOTES)

Topic: Navigation dead reckoning system based on a mobile phone

Done by: **Lazarevskiy O.A.**

Supervised by: **Mukhina M.P.**

Normcontrolled by: **Tupitsyn M. F.**

Kyiv 2021

НАЦІОНАЛЬНИЙ АВІАЦІЙНИЙ УНІВЕРСИТЕТ

Факультет аеронавігації, електроніки та телекомунікацій

Кафедра авіаційних комп’ютерно-інтегрованих комплексів

Освітній ступінь бакалавр

Спеціальність: 151 " Автоматизація та комп'ютерно-інтегровані технології"

ЗАТВЕРДЖУЮ

Завідувач кафедри

Синєглазов В.М.

“ ____ ” __________2021 р.

ЗАВДАННЯ

на виконання дипломної роботи студента

Лазаревського Олександра Андрійовича

1. **Тема проекту (роботи):** “Навігаційна система числення шляху на основі мобільного телефону”
2. **Термін виконання проекту (роботи):** з 10.05.2021 р. до 11.06.2021 р.
3. **Вихідні данні до проекту (роботи):** числення шляху на основі мобільного телефону, навігаційна система, алгоритм пошуку шагів, середовище Matlab.
4. **Зміст пояснювальної записки (перелік питань, що підлягають розробці):** 1.Актуальність навігаційна система числення шляху на основі мобільного телефону; 2. Огляд існуючих методів; 3. Огляд теоретичної інформації з приводу рішення задачі пошуку шагів; 4. Розробка навігаційної системи числення шляху на основі мобільного телефону.
5. **Перелік обов’язкового графічного матеріалу:** 1. Блок-схема числення шляху пішохода 2. Положення мобільного телефону у просторі; 2. Вікно розробленої системи пошуку пройденого шляху; 3. Фази ходи людини; 4. Таблиця порівняння кількості шагів щодо реального та виміряної кількості за алгоритм пошуку шагів; 5. Структурна схема розробленої програми.

6. **Календарний план-графік**

№ пор.	Завдання	Термін виконання	Відмітка про виконання
1.	Отримання завдання	10.05.2021 – 11.05.2021	
2.	Формування мети та основних завдань дослідження	12.05.2021 – 13.05.2021	
3.	Аналіз існуючих методів	14.05.2021 – 19.05.2021	
4.	Теоретичний розгляд рішення задачі	20.05.2021 – 25.05.2021	
5.	Розробка структури навігаційної системи числення шляху	25.05.2021 – 30.05.2021	
6.	Розробка програмного забезпечення навігаційної системи числення шляху за методом пошуку піків	30.05.2021 – 05.06.2021	
7.	Оформлення пояснювальної записки	05.06.2021 – 07.06.2021	
8.	Підготовка презентації та роздаткового матеріалу	08.06.2021 – 11.06.2021	

7. **Дата видачі завдання:** “10” травня 2021 р.

Керівник дипломної роботи ____________________ Мухіна М.П.

(підпис керівника) (П.І.Б.)

Завдання прийняла до виконання ____________________ Лазаревський О.А.

(підпис випускника) (П.І.Б.)

NATIONAL AVIATION UNIVERSITY

Faculty of aeronavigation, electronics and telecommunications

Department of Aviation Computer Integrated Complexes

Educational level bachelor

Specialty: 151 "Automation and computer-integrated technologies"

APPROVED

Head of Department

Sineglazov V. M.

"____" ___________2021

TASK

For the student's thesis

Lazarevskiy Alexander Andreevich

1. **Theme of the project:** "Navigation dead reckoning system based on a mobile phone "
2. **The term of the project (work):** from May 10, 2021 until June 11, 2021
3. **Output data to the project (work):** dead reckoning system based on mobile phone, navigation system, step find algorithm, Matlab environment.
4. **Contents of the explanatory note (list of questions to be developed):** 1. Relevance of the navigation dead reckoning system based on a mobile phone; 2. Review of existing methods; 3. Review of theoretical information about the solution of the problem of finding steps; 4. Development of a navigation dead reckoning system based on a mobile phone.
5. **List of compulsory graphic material:**

1. Block diagram of the calculation of the pedestrian dead reckoning 2. The position of the mobile phone in space; 2. Window of the developed system of search of the passed way; 3. Phases of human walking; 4. Table comparing the number of steps relative to the actual and measured number according to the algorithm for finding steps; 5. Block diagram of the developed program.

6. Planned schedule:

№	Task	Execution term	Execution mark
1.	Task	10.05.2021 – 11.05.2021	
2.	Purpose formation and describing the main research tasks	12.05.2021 – 13.05.2021	
3.	Analysis of existing methods	14.05.2021 – 19.05.2021	
4.	Analysis of existing systems	20.05.2021 – 25.05.2021	
5.	Development of the structure of the navigation system of the way	25.05.2021 – 30.05.2021	
6.	Development of software for navigation system of number calculation by the method of finding peaks	30.05.2021 – 05.06.2021	
7.	Making an explanatory note	05.06.2021 – 07.06.2021	
8.	Preparation of presentation and handouts	08.06.2021 – 11.06.2021	

7. Date of task receiving: "10" May 2021

Diploma thesis supervisor Mukhina M.P.

(signature)

Issued task accepted Lazarevskiy O.A.

(signature)

РЕФЕРАТ

Пояснювальна записка до дипломної роботи «Навігаційна система числення шляху на основі мобільного телефону»: с., 44 рис., 7 табл., 25 літературних джерела.

Об'єкт дослідження: система числення шляху пішохода.

Мета роботи: вдосконалення методів пошуку шагів та розробки нової системи, що здатна з порівняно невеликими обчислювальними затратами здійснювати побудову шляху.

Для досягнення цієї мети необхідно розв'язати наступні завдання:

- проаналізувати існуючі методи пошуку шагів;
- проаналізувати існуючі апаратні засоби, придатні для реалізації розглянутих методів;
- на основі проведеного аналізу здійснити вибір методу який би дозволив здійснювати розрахунок маршруту з мінімальними обчислювальними затратами;
- розробити програмне та апаратне забезпечення реалізації обраного методу;
- провести експериментальне дослідження роботи розробленої системи.

Предмет дослідження: розробка методу навігаційної системи числення шляху на основі мобільного телефону.

Методи дослідження: теоретична фізика, теорія коливань, теорія обробки даних з магнітометра.

ІНЕРЦІАЛЬНА НАВІГАЦІЯ; ЗНАХОДЖЕННЯ ПІКІВ КРОКУ ЛЮДИНИ; МОДУЛЬ ВЕКТОРА ПРИСКОРЕННЯ; ФІЛЬТРАЦИЯ; ТРИАНГУЛЯЦІЯ; ГЛОБАЛЬНЫЙ МАРШРУТ.

ABSTRACT

Explanatory note to the thesis "Navigation dead reckoning system based on a mobile phone": p., 44 figures, 7 tables, 25 literary resources.

The object of research: pedestrian dead reckoning system.

The purpose of the work: improving the methods of finding steps and developing a new system that is able to build a path with relatively low computational costs.

To achieve this purpose, it must be solved the following tasks:

- analyze existing methods of finding steps;
- to analyze the existing hardware suitable for the implementation of the considered methods;
- on the basis of the conducted analysis to make a choice of a method which would allow to carry out calculation of a route with the minimum settlement expenses;
- to develop software and hardware for the implementation of the selected method;
- to conduct an experimental study of the developed system.

Subject of research: development of a method of navigation dead reckoning system based on a mobile phone.

Methods of research: theoretical physics, theory of oscillations, theory of data processing from a magnetometer.

INERTIAL NAVIGATION; FINDING PEAKS OF PEOPLE'S STEP; ACCELERATION VECTOR MODULE; FILTRATION; TRIANGULATION; GLOBAL ROUTE.

CONTENT

GLOSSARY

PDR – Pedestrian dead reckoning

LCS – Local coordinate system

GLCS – Global coordinate system

MEMS – Microelectromechanical sensor

IFR – Inertial frame of reference

CMO – Center of mass of an object

GCS – Geographic coordinate system

CS – Coordinate system

GLONASS – Global Navigation Satellite System

GPS – Global Positioning System

ZC – Zero-crossing

ACF – Autocorrelation function

IMU – Inertial measurement unit

GUI – Graphical user interface

DS – Double support

SS – Single support

INTRODUCTION

In the modern dynamically developing world, a person occupies a leading position, in such conditions the market for complex automated integrated systems of enterprises and business institutions of various profiles, as well as devices that provide communication and always inevitably be near us, is developing.

The devices that surround us everywhere have not only a set of standard programs that the developer has introduced there. For example, people who use standard GPS navigation methods never thought about how to navigate in a shopping center and a store, or an enterprise.

All of these organizations can be of very different sizes with a variety of construction schemes, ranging from small businesses with a few dozen people to large corporations with tens of thousands of employees.

The system of inertial individual navigation solves such problems, such systems are designed to solve the problems of both the enterprise as a whole and the level of each individual.

The purpose of this work was to develop an inertial pedestrian system for counting the steps taken, as well as orientation in space without external data.

The first section of the explanatory note describes the relevance of the problem of dead reckoning for pedestrians, as well as its features.

The second section of the note describes the scope of the methods and examines their effectiveness.

The third section describes the algorithm of the developed program, where there is information about its constituent methods used in this work.

The fourth section is the documentation of the software. It contains information about software components and their interaction, as well as describes the methodology for the user to work with the software system.

CHAPTER 1. RELEVANCE OF THE PROBLEM OF PEDESTRIAN DEAD RECKONING

A pedestrian does not always have the opportunity to receive a satellite signal, sometimes it can be deliberately blocked or lost due to obstacles in urban areas, lack of satellite coverage of a specific area or incorrect signal triangulation.

1.1. Dead reckoning problem

Pedestrian dead reckoning (PDR) belongs to the class of individual navigation tasks [1]. A common solution in mobile phones is to use satellite navigation (GPS, GLONASS, Galileo etc.). But also signal GPS can be muted, the problem of PDR is of interest for user localization in premises such as large stores, city shopping centers, etc., where signal can be muted.

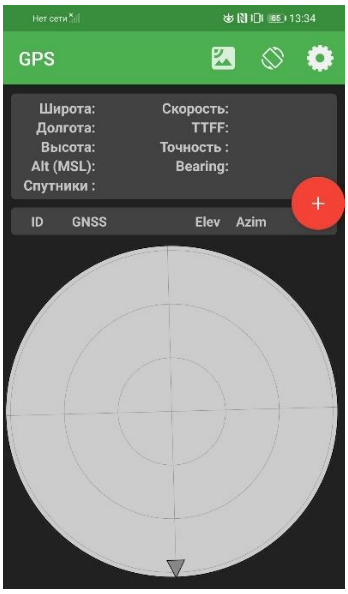


Fig 1.1. Muted GPS signal

All modern smartphones have internal hardware based on micro electromechanical sensors (MEMS) which includes a standard set of inertial measuring unit (IMU): accelerometers, gyroscopes, magnetometers and pressure sensor (optional).

Accelerometers can be used to detect steps and later to calculate length. But it is sensitive to walking speed, road slope, which must be considered when calculating. Because of this, any inertial determination method suffers from a cumulative error, that is, methods PDR accumulate error over time as the location is calculated based on the previous result.

Standard indoor personal positioning with an error greater than 10 m is still challenging to explore. It is well known that the key to solving this problem is to combine the various positioning techniques in more detail in CHAPTER 2.

1.2. Limitation of use pedestrian dead reckoning

As in any other number system, there are limitations:

- Positioning the smartphone in a person's hands;
- Gyro drift over long distances;
- The location of the axes of reading data in the sensors of the smartphone;
- Availability of Hard and Soft iron.

This limitation can be circumvented, for example, for positioning the smartphone as well as the location of the axes of the sensors inside the smartphone, there is a method for transforming rotation matrices in space [2], since an object is usually defined in a local coordinate system (LCS) convenient for its description, and its position in space - in the global coordinate system (GLCS).

At small distances, the gyroscope drift is not considered due to its small effect, for calculations at long distances, in order to avoid such a drift in the attitude assessment, it is necessary to consider other sources of input data to correct the results obtained as a result of the integration of the gyroscope data. The solution lies in data integration, from the orientation sensor of the magnetometer and accelerometer, and data from the gyroscope.

Magnetic measurements will be distorted. These misstatements are believed to fall into one of two categories; hard or soft iron. Solid iron distortion is created by objects that create a magnetic field. For example, a loudspeaker or a piece of magnetized iron will cause hard iron distortion.

If a piece of magnetic material is physically attached to the same frame of reference as the sensor, then this type of hard iron distortion will cause a constant bias at the sensor output. Soft iron distortions are considered abnormalities or changes in the existing magnetic field.

These distortions will stretch or distort the magnetic field depending on which direction the field is acting in relative to the sensor. This type of distortion is commonly caused by metals such as nickel and iron. In most cases, hard iron distortion will contribute much more to the total uncorrected error than soft iron.

It is possible to exclude the influence of distortions of both hard and soft iron on the outputs of the magnetometer. Using the calibration model [3] to correct the distortion of hard and soft iron.

1.3. Block diagram of pedestrian dead reckoning

Today the quality of inertial sensors built into smartphones is sufficient for solving tasks without inertial pedestrian navigation.

Dead reckoning for pedestrians has a general structural diagram [4] shown in Fig. 1.2. And the latest MEMS performance improvements, combined with new algorithms to reduce sensor errors using flexible signals, push the boundaries of existing inertial pedestrian navigation solutions.

The method of dead reckoning of pedestrians contains orientation angles, including heading, that are estimated by integrating the angular velocities read by the gyroscope. With this approach, gyroscope errors propagate at a cubic speed. With inexpensive handheld devices, the ability to reduce sensor deflection and noise is limited.

Regarding magnetic fields, updating the geo-magnetic heading cannot be easily and often applied in urban and interior spaces, because the Earth's magnetic field is severely disturbed by the surrounding artificial fields.

Another source of error relates to the parameterization of the rotation between the mobile device and the navigation frame.

Since the hand performs rapid movements throughout three-dimensional space without indicating any direction, this problem occurs frequently.

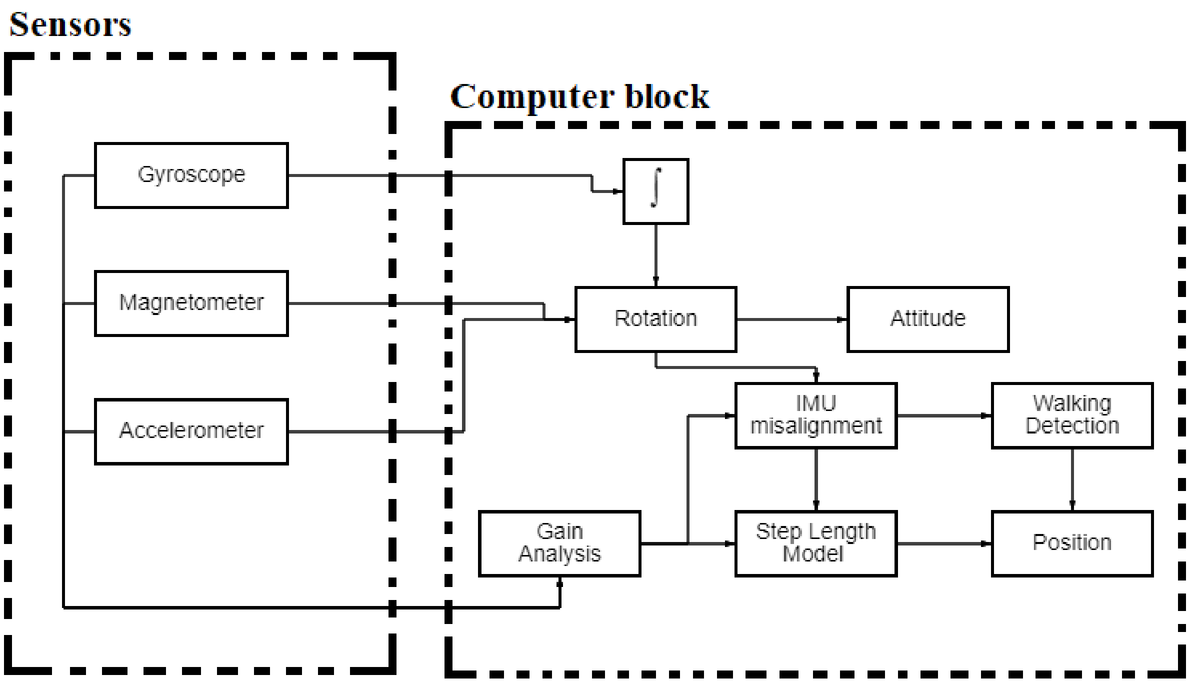


Fig. 1.2. Block diagram of PDR

1.4. Mathematical model of pedestrian dead reckoning

The mathematical model of the state of a pedestrian is based on a nonlinear particle filter [5]. Moving the user has a state vector s_k, the process of the equation of state is described in such a nonlinear form that estimates the parameters included in the state vector:

$$s_k = [x^T \; V^T \; q^T \; \varepsilon_a^T \; \varepsilon_\omega^T]_k^T \tag{1.1}$$

Where x - coordinate vector, V - velocity vector, q - rotation quaternion, ε - vector of displacement of readings of accelerometer and gyroscope.

$$x_k = x_{k-1} + V_{k-1} + C(q_{k-1})(\varepsilon_{\mathrm{a}_{\mathrm{k}-1}})\frac{T^2}{2} \tag{1.2}$$

$$V_k = V_{k-1} + C(q_{k-1})(\varepsilon_{\mathrm{a}_{\mathrm{k}-1}})T \tag{1.3}$$

$$C(q) = \begin{bmatrix} q_1^2 + q_2^2 - q_3^2 - q_4^2 & 2(q_2q_3 - q_1q_4) & 2(q_1q_3 + q_2q_4) \\ 2(q_2q_3 + q_1q_4) & q_1^2 + q_3^2 - q_2^2 - q_4^2 & 2(q_3q_4 - q_1q_2) \\ 2(q_2q_4 - q_1q_3) & 2(q_1q_2 + q_3q_4) & q_1^2 + q_4^2 - q_2^2 - q_3^2 \end{bmatrix} \tag{1.4}$$

A human movement model is used as dynamic models.

The idea is that when considering a uniform movement in it, one can distinguish the fundamental harmonic characteristic of the coordinates of all points of the human body - in time with the steps. Additional information to improve the accuracy of systems of restrictions imposed on the movement of body parts, for example, maintaining an upright position.

1.5. Human step model

Usually [6] the transition between the phases of walking Fig. 1.3, is accompanied by a corresponding change in acceleration. Therefore, most of the methods for detecting and counting steps are based on processing the magnitude of the acceleration vector.

In Figure 1.3, a diagram for studying the gait cycle is shown and has been widely used to study and simulate human gait. Accordingly, the subject performs a walking movement, alternating a double support (DS) phase with a single support (SS) phase, defined as follows:

- DS: the period during which both feet touch the floor. Both the beginning and the end of the stance phase are considered a double support period.
- SS: the period begins when the opposite leg is raised for the swing phase

For reference, in healthy subjects, DS is typically 20% of the normal gait cycle compared to 80% of SS. The time it takes to support the two limbs decreases as the walking speed increases. Walking is different from running because the latter lacks the DS period.

In Figure 1.3, walking motion can be simulated by examining the transitions between DS and SS states. And, therefore, several gait parameters can be obtained:

- Step length: One step is the action performed from one Single Support State to another Single Support State. For each step calculate the time length, the distance moved by the subject and the arm swing range (the arm swing is an interesting parameter to be studied in the gait assessment).
- Stride length: One stride is the action performed from one Double Support State to another Double Support State. For each step calculate the time length, the distance moved by the subject.
- Velocity: with the information of the timestamp and the distance it is possible to know the velocity of each step and stride.
- Cadence or walking rate: is calculated in steps per minute.

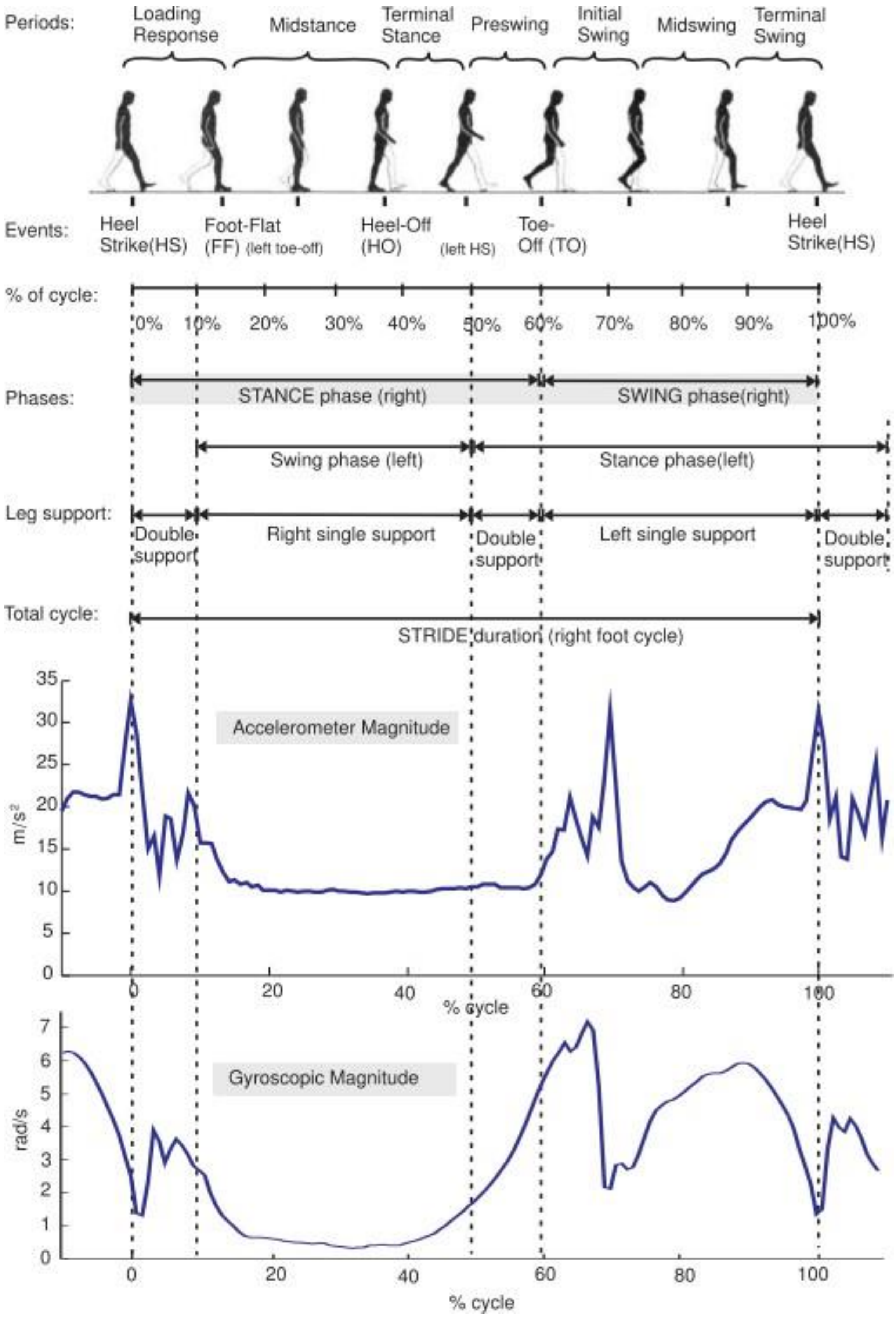


Fig. 1.3. Phases of walking

1.6. Coordinate systems

There are many coordinate systems. For the diploma, 2 coordinate systems were used: Inertial frame of reference (IFR) and geographic coordinates.

1.6.1. Inertial reference system

Inertial frame of reference — frame of reference in which all free bodies move rectilinearly and uniformly or are at rest [7].

IRS in relation to which space is homogeneous and isotropic, and time is homogeneous.

Taking Earth as IFR Fig. 1.4, despite its approximate nature, is widespread in navigation. IFR compiled according to the following algorithm. The center of the earth is selected as the point O - the origin of coordinates in accordance with its adopted model. The z-axis coincides with the earth's axis of rotation. The x and y axes are in the equatorial plane. It should be noted that such a system does not participate in the rotation of the Earth.

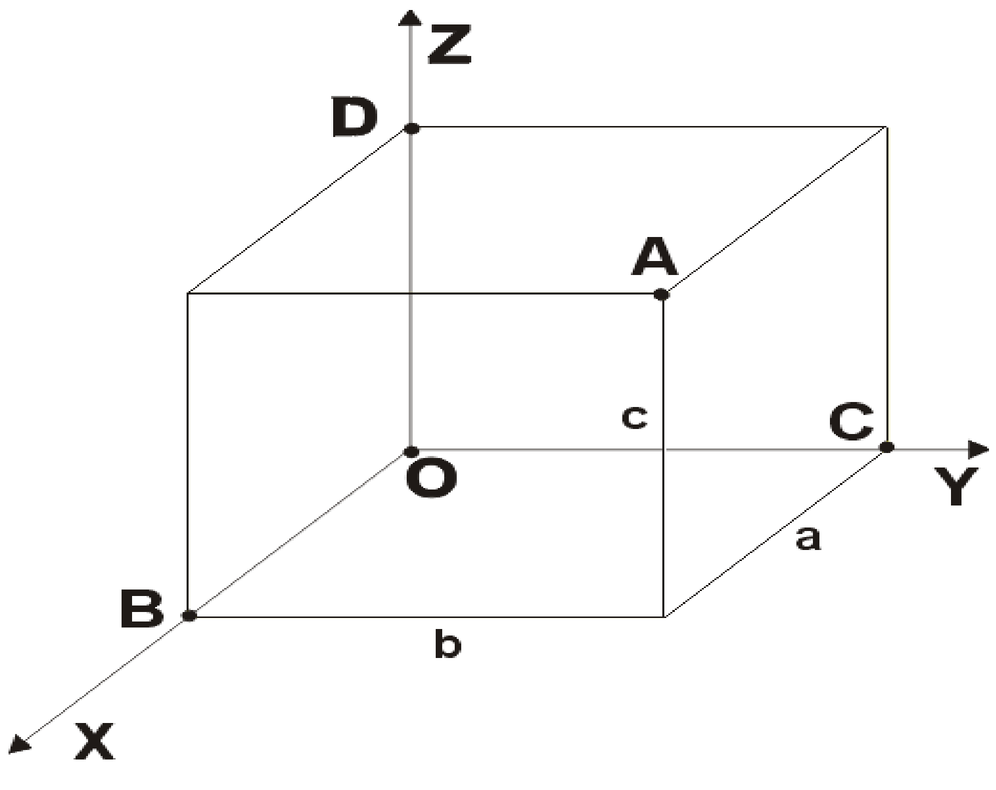


Fig. 1.4. Inertial rectangular coordinate system in space

1.6.2. Geographical coordinates

Geographic coordinates are a generalized concept of geodetic and astronomical coordinates, when the deviation of the plumb line is not considered [8]. In other words, when determining the geographic coordinates, the Earth is taken as a sphere Fig. 1.5, not an ellipsoid of revolution. Geographic coordinates define the position of a point on the earth's surface or, more broadly, in a geographic envelope. Geographic coordinates are based on the spherical principle. Similar coordinates are used for other planets, as well as on the celestial sphere.

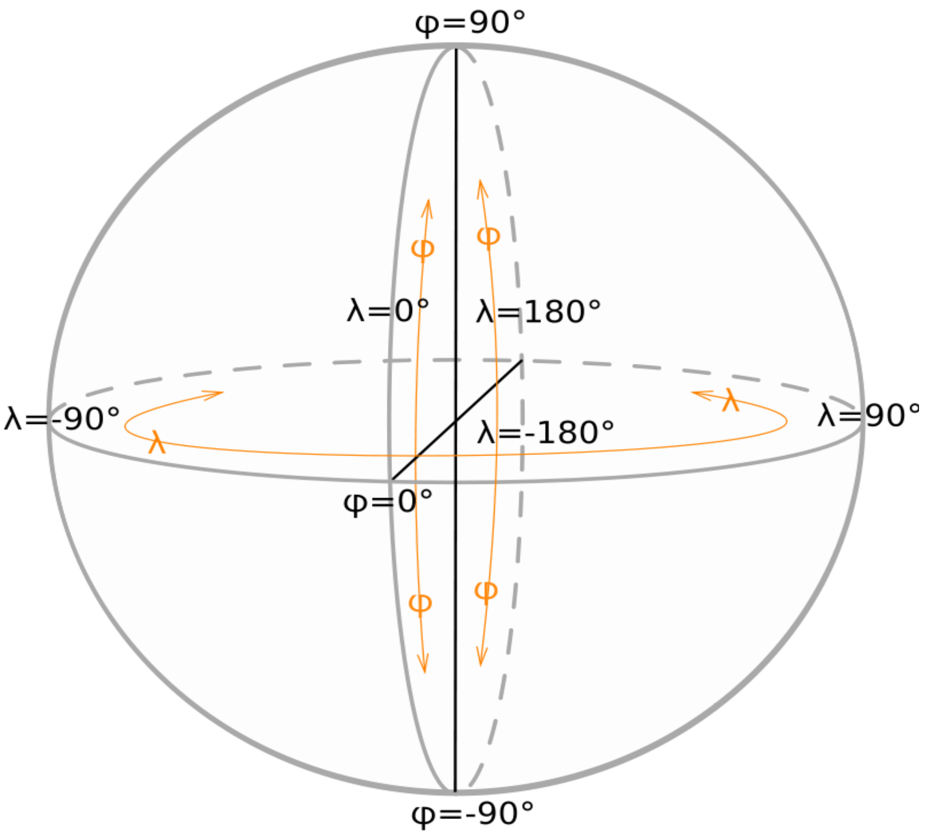


Fig. 1.5. Geographic coordinates on a sphere

In navigation, the center of mass of the object (CMO) is selected as the origin of the coordinate system. The transition of the origin of coordinates from an inertial coordinate system to a geographic one (that is, from O_i in O_g) is carried out based on the values of latitude and longitude. Geographic coordinate system center coordinates O_g in inertial values take on values (when calculating according to the spherical model of the Earth):

$$
\begin{aligned}
X_{og} &= (R+h)\cos(\varphi)\cos(Ut+\lambda) \\
Y_{og} &= (R+h)\cos(\varphi)\sin(Ut+\lambda) \\
Z_{og} &= (R+h)\sin(\varphi)
\end{aligned}
\tag{1.5}
$$

where R — radius of the earth, U — angular velocity of rotation of the earth, h — height above sea level, φ — latitude, λ — longitude, t — time.

The orientation of the axes in the geographic coordinate system (GCS) is selected according to the scheme:

X-axis (also known as E-axis) - east-facing axis.

Y-axis (another designation - N-axis) - axis pointing north.

Z-axis (another designation is the Up-axis) - the axis directed vertically upwards.

The main disadvantage in the practical application of GCS in navigation is the large values of the angular velocity of this system at high latitudes, increasing up to infinity at the pole. Therefore, instead of GCS, a semi-free in azimuth CS is used.

Semi-free coordinate system in azimuth, in reality, all calculations are carried out in this system, and then, to provide output information, the coordinates are converted to GCS.

Accordingly, the system also has an initial position, it is carried out according to the formula:

$$
\begin{aligned}
N &= Y_w\cos(\varepsilon) + X_w\sin(\varepsilon) \\
E &= -Y_w\sin(\varepsilon) + X_w cos(\varepsilon).
\end{aligned}
\tag{1.6}
$$

CHAPTER 2. OVERVIEW OF METHODS FOR COUNTING STEPS AND CALCULATING PEDESTRIAN DIRECTION

Due to the possibility of using GPS, today inertial navigation systems are not often found among ordinary users, mainly to engage in this practice at enterprises.

There are both large, bespoke systems and conventional systems that can be purchased, usually by subscription. In addition, their source code is in private access, which means that as a result of errors, can be need to contact the support department, which will take a lot of time, or the need to change the system and add something, which will be virtually impossible.

As a result of independent implementation of the system, the developer has full control over all processes and can choose tools based on the task at hand, as well as modernize and expand the application.

2.1. Method of pedestrian dead reckoning for smartphones

During the writing of the project, several methods of step detection and course calculation were considered to determine the user's location position.

At the moment there are many methods to determine the steps, the most popular are Table 2.1, of them these 5 methods:

- Threshold detection method
- Threshold dynamic detection method
- Zero-crossing method
- Peak detection method
- Autocorrelation

As well as several methods for determining the course Table 2.2 using a gyroscope and using a magnetometer:

- Component of vertical angular velocity
- Signed magnitude angular velocity
- Calculation of the course based on the magnetometer

Table 2.1 Existing methods for finding a step

	Extensibility	The need for filtration	Performance
Threshold detection	-	-	+
Threshold dynamic detection	+	+	+
Zero-crossing	+	+	+
Peak detection	+	+	+
Autocorrelation	+	-	-

Table 2.2 Existing methods of finding a course

	Extensibility	Accumulation of error	Performance
Component of vertical angular velocity	+	+	+
Signed magnitude angular velocity	+	+	-

Calculation of the course based on the magnetometer	+	-	+

2.1.1. Step counting method based on threshold detection

Threshold detection method – this is the approach using thresholds [9], which is the simplest solution for finding steps in a time series of values taken from accelerometer sensors. The signal received from the sensors has a waveform and judges whether the signal from the sensors meets some predetermined values. Which needs to be changed depending on the device that the person uses, as well as if someone goes up, down or runs, or just walks at a different pace. Then there are many options if testing this method with multiple users. How thresholds will vary greatly between devices, so the frequency of measurements will also have a big impact. When using a constant frequency that detects one person's walking at a normal pace, there is no guarantee that another person's walking will be detected. Also, the phone can be in different positions in the hands, in the pocket of trousers or shorts, or fastened to a belt. This gives a lot of redundant data and little information. Since manufacturers use [10] different versions of accelerometers, so if software works on one phone, there is no guarantee that it will work on another device, even of the same model. Although the approach using thresholds is the simplest, it is really difficult to choose one optimal threshold for all cases, especially when the smartphone is used without space restrictions.

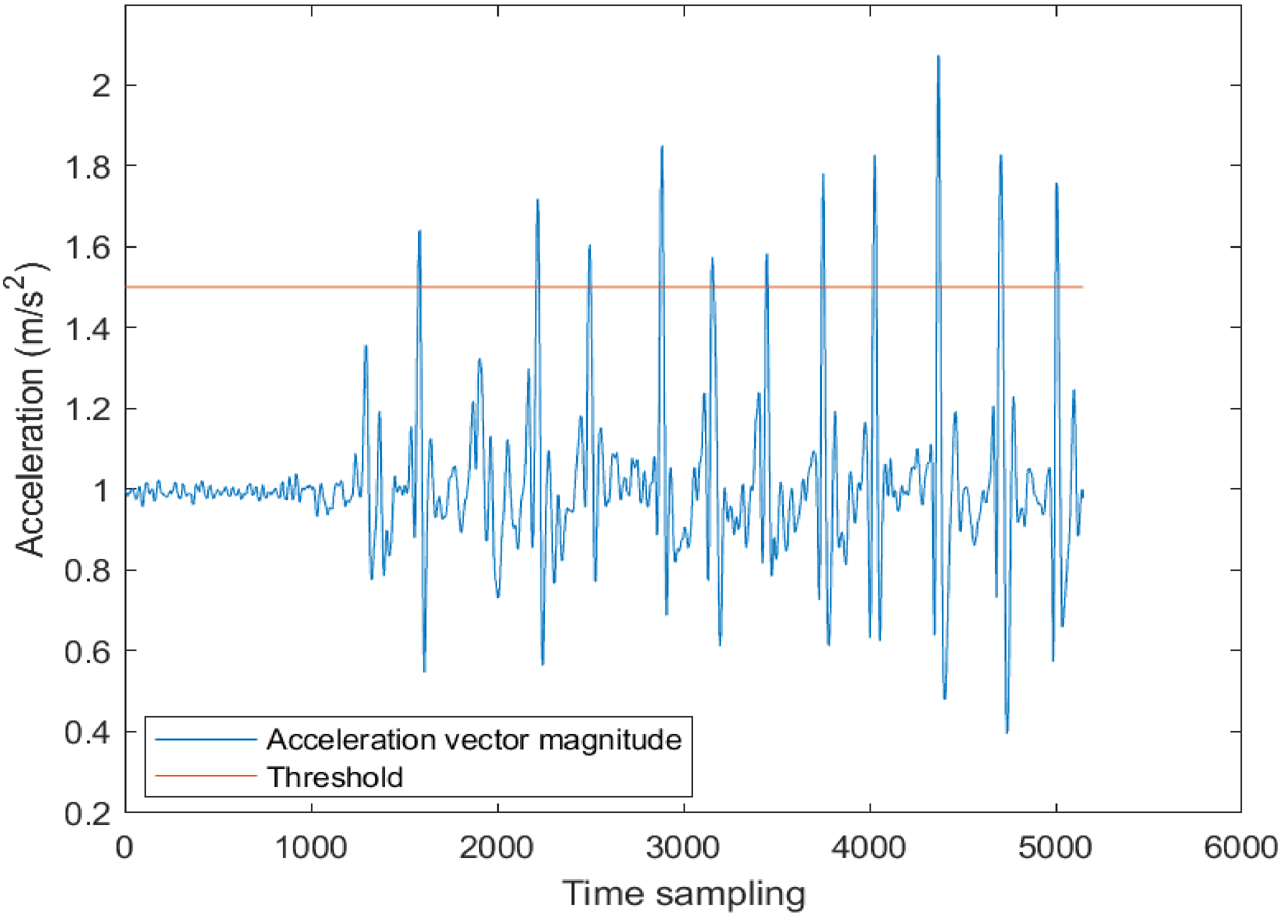


Fig. 2.1. Evaluation threshold for counting steps

Disadvantages:

- Pace of walking
- Frequency of data recording
- Sensor model
- Being in a fixed plane
- Constant threshold

Advantages:

- Simple algorithm
- Works fast

The easiest and fastest method for finding steps, but has many disadvantages

2.1.2. Step counting method based on dynamic detection threshold

Dynamic detection threshold method — this is an approach using a threshold dynamic step detection method [16], which is a more advanced version of the threshold method.

Dynamic change in the threshold allows to more specifically record a person's steps, since the method is designed for the fact that a person can change his pace of walking, switch to running and make sudden stops.

There are several steps to determine the step:

- Smoothing
- Dynamic threshold detection
- Algorithm for detecting steps
- Periodicity

Smoothing: the signal does not always look smooth, having various additional oscillations and noise, this is due to the inaccuracy of data acquisition, as well as signal digitization. Signal smoothing will help to fix this problem.

Smooth signal — this is an easy way to average nearby values to remove some of the noise. For example, can replace each sample with the mean of the current sample, the previous sample, and the sample after it. Remove unnecessarily noisy signal areas. To observe the difference in effect, it is necessary to use different anti-aliasing windows. As the smoothing window increases, the signal will look cleaner and more visually pleasing, but care must be taken when using a window that is too large to smooth out the steps that need to be detected first.

Dynamic detection threshold: given the smoothed data, the next problem is to determine when a step occurs. For example, by extracting the peak point of each step, one can simply count the steps. However, the problem is that there is no fixed threshold that can be used, since the threshold depends on the orientation of the user's phone and the accelerometer in it. So, it is necessary to use a dynamic threshold detection scheme for step detection. The first step in this algorithm is to track the axis (x, y or z), along which the maximum acceleration occurs. While gravity only acts on one axis with the force the rest can be ignored and just focus on that particular axis for this algorithm. This axis tracks the minimum and maximum acceleration levels in the selections window. In other words, the maximum and minimum values of triaxial acceleration are continuously measured every 50 counts. The average value is called the dynamic threshold. For the next 50 samples, this threshold level is used to determine whether steps have been taken. Since it is updated every 50 samples, the threshold is dynamic. The maximum, minimum and (dynamic) threshold values for the z-axis are shown in Fig. 2.2.

Step detection algorithm: Given the dynamic detection threshold, the step detection algorithm can work by finding threshold crossings in the downstream (or upstream) direction. For example, looking at Figure 2.2, it is not difficult to see that each step involves crossing the threshold (purple line) in a downward direction with a significant change in acceleration in the negative direction.

In other words, a step is defined as occurring if there is a negative slope of the acceleration curve when the acceleration curve crosses below the dynamic threshold. Possibility to improve the step detection algorithm.

Periodicity: the step counter calculates steps in the x-axis, y-axis, or z-axis, depending on which axis has the greatest change in acceleration. If the acceleration changes are too small, the step counter will discard them. The step counter can work well with this algorithm, but sometimes it seems overly sensitive. When the pedometer vibrates very quickly or very slowly for a reason other than walking or running, the step counter will also take that as a step. Such unacceptable vibrations must be discarded in order to find the true rhythmic steps.

In order to avoid this, it is necessary to look at the time interval between any two steps. Assuming people can run at five steps per second and walk at one step every one second. Thus, the interval between two permissible steps is in the range

[0.2 to 1.0 s], all steps with intervals outside the time window can be discarded to reduce the number of error sources.

Disadvantages:

- Smoothing required, possible loss of signal data
- Being in a fixed plane
- More time consuming

Advantages:

- Dynamic threshold
- Ability to change the pace
- Ability to change the speed of data writing
- Extensibility of the step detection algorithm

The algorithm is more time consuming, but more accurate than the conventional thresholding method.

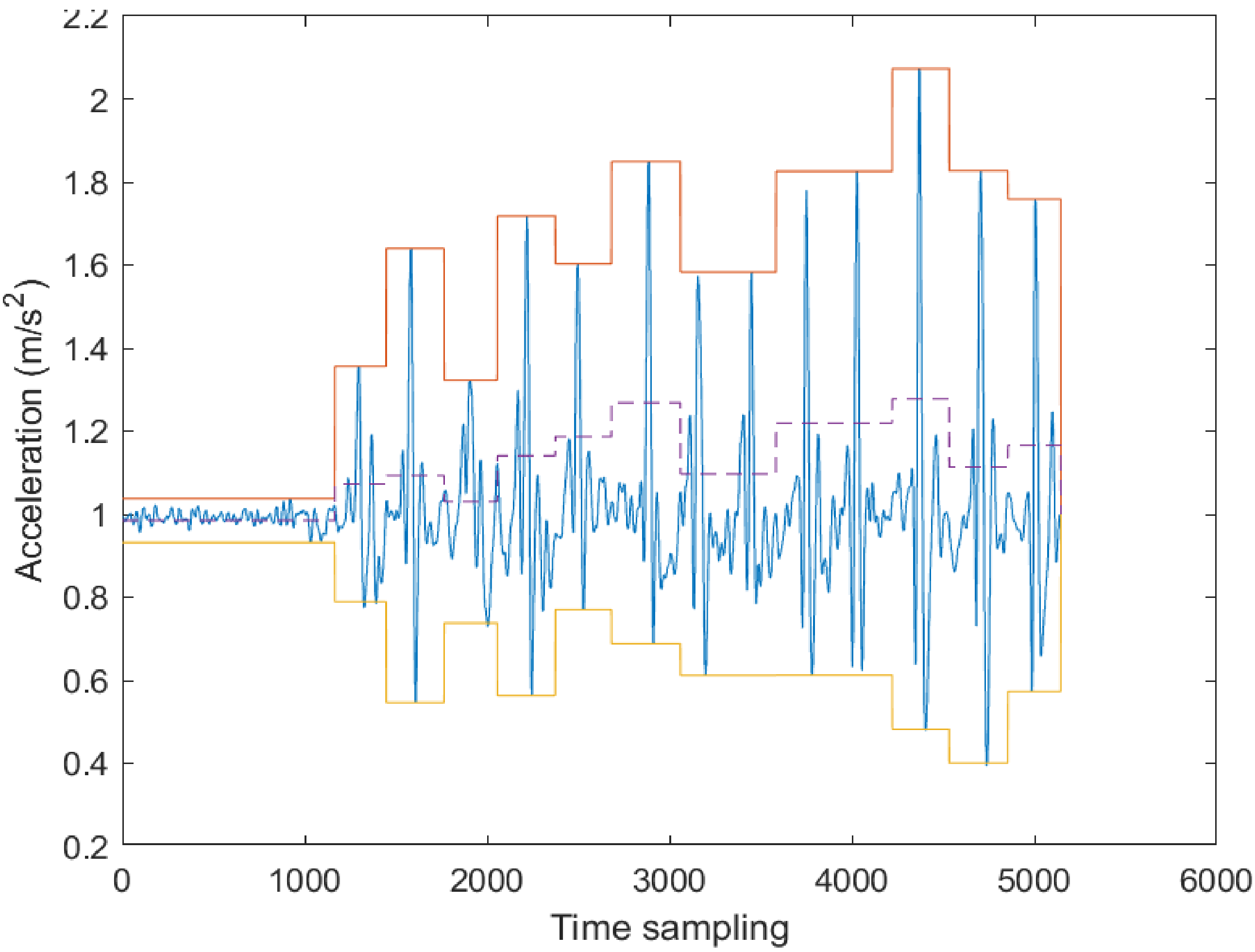


Fig. 2.2. Dynamic detection threshold

2.1.3. Step counting method based on zero-crossing

Zero-crossing (ZC) – this approach counts the steps [9], detecting the number of zero points in the sensory data that are prone to sensory corruption and usually require pre-filtering and anti-aliasing of the original sensory data.

Zero-crossing in Fig. 2.3 shows the data of the acceleration module when the user started walking after a short stop interval. From the moment the user begins to walk, a repeating pattern, that is, a signal crossing the zero mark once in a negative direction, followed by the same action, but in a positive direction, can be observed. This phenomenon is called ZC and can be used to count steps [17]. However, there is one critical issue when using ZC. As mentioned earlier, the raw acceleration data contains noise, even after using the noise reduction technique. Such noise can cross zero in the signal register, even when the user is at rest.

The zero-crossing boundary is used to solve this problem (BZC – boundary zero-crossing). Boundary zero-crossing can be thought of as a filtering mechanism that helps the system consider only those zero crossings that are the result of actual body movement.

This is achieved by using artificial boundaries on either side of the zero point, which is called a range. In Fig. 2.3, BZC range is marked with red lines. Zero-crossing is counted only if the signal came from outside this range, otherwise it is ignored. In Fig. 2.3, the points at which the zero crossing is considered are shown in green.

The zero-crossing approach looks for periods inherent in the cyclical nature of walking, using acceleration values or angular velocities, and can provide better performance, for example, with vertical accelerations, but are degraded if the smartphone is not firmly attached to the human body.

Disadvantages:

- Smoothing required, possible loss of signal data
- Being in a fixed plane
- Constant detection limit
- Additional data processing required

Advantages:

- Ability to change the pace
- Ability to change the frequency of data recording
- Extensibility of the step detection algorithm

This algorithm is best used in tandem with other algorithms, to increase their accuracy, independently using it gives a large error in measurements.

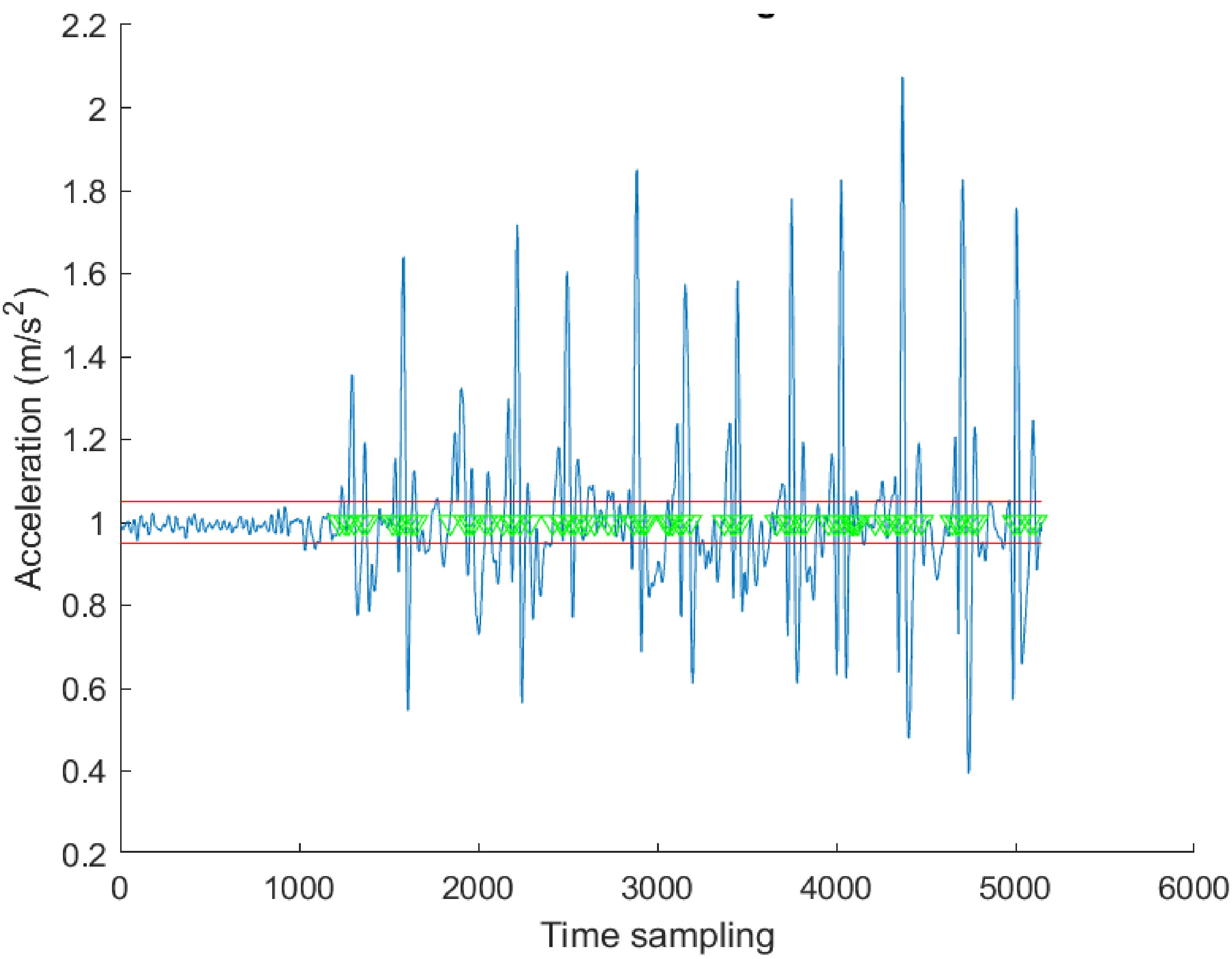


Fig. 2.3. Zero-crossing method

2.1.4. Step counting method based on peak detection

Peak detection – task in time series analysis and signal processing [9]. Common approaches to peak detection include the use of smoothing and matching a known peak shape to a time series. The peak detection approach estimates steps based on the number of peaks given by the captured data sequence and does not rely on predetermined thresholds but suffers from interference peaks due to environmental noise and random interference. To get rid of noise and interference from the environment, it is necessary to use low-pass filtering to remove the interference [18]. The boundaries of the peaks are looked for by determining the increase and decrease, the vector of the acceleration modulus. Thus, the points of all possible peaks are determined, which will subsequently be processed. And also, to limit the time intervals between two peaks in accordance with the model of human behavior, as well as in the frequency of dynamic detection of the threshold, in order to avoid this, it is necessary to look at the time interval between any two steps. Comparing how far the peaks are from each other, it is considered that when overcoming the gap, these are already two or more peaks that are subject to subsequent processing, but if there are several peaks in the time interval, then the main one is highlighted, the criterion for highlighting the main one is the acceleration vector module, that is, the magnitude of the peak.

Assuming people can run at five steps per second and walk at one step every one second. Thus, the interval between two allowable steps is in the range [from 0.2 to 1.0 s] with a time sample of 100 Hertz, this value will be [from 20 to 100-time samples], all steps at intervals outside the time window can be discarded to reduce sources of error and to reduce erroneous judgment. Likewise, finding the device does not have to be in a certain plane, since the vector module is used for acceleration along 3 axes, this gives freedom of action in use, a device for which acceleration data is measured.

Due to the use of the method, the peak detection is obtained, to find more accurate data about the step of a person. In contrast to the intersection ZC, where both the border and the range itself are located near areas with increased noise. Peak detection method Fig. 2.4. avoids most of the problems of other methods, and also loses to them in data processing speed. To improve this algorithm, it is possible to apply more filters, for example, a filter to reduce jitter during acceleration, as well as the vertical acceleration data that is used to determine the steps required combining sensors.

Disadvantages:

- Additional data processing required
- Necessary filtering of false peaks

Advantages:

- Ability to change the pace
- 3-axis freedom
- Smoothing does not lead to data loss
- Ability to change the frequency of data recording
- Extensibility of the step detection algorithm
- Speed

This algorithm proved to be the most reliable, convenient to use, while having one of the best accuracies in comparison with other algorithms, the advantage of the algorithm in the absence of constant detection boundaries, as in the methods: threshold values, dynamic detection threshold and ZC. It also has a high data processing speed and great extensibility of the algorithm and methods of its use.

Also, combining with other methods helps to unleash their potential.

In the course of the work, this particular algorithm for finding steps was used, due to its speed, as well as the accuracy of counting steps, which is of great importance for the subsequent construction of the route, the algorithm effectively counts steps even when using the phone arbitrarily while walking. The disadvantages of the method are completely covered by its advantages.

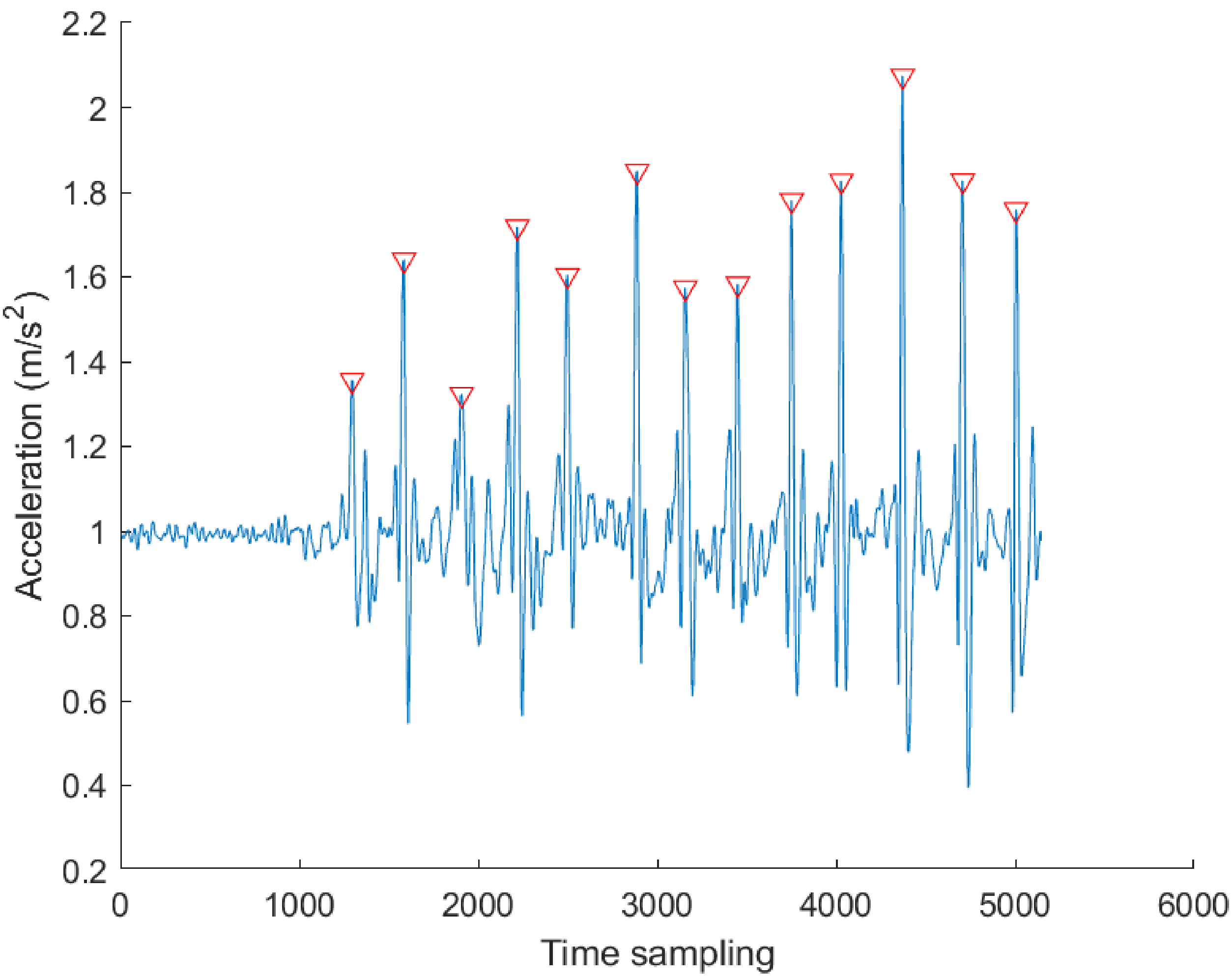


Fig. 2.4. Peak detection

2.1.5. Step counting method based on autocorrelation

Autocorrelation — it is an approach that defines cyclical periods directly in the time domain through autocorrelation estimation [9] and allows good performance at relatively low cost compared to frequency domain approaches. By examining the fact that the accelerometer signal is a pattern, which repeats whenever the user takes a new step, that is, the user's movement is repeated. Therefore, by considering an accelerometer signal in the time domain, autocorrelation can be used to measure the degree of similarity between a given time interval and its inverse version in successive time intervals [19]. Using this principle, steps can be detected by analyzing the maximum results of this operation.

An autocorrelation algorithm for counting steps using smartphones, which is correlation based.

Model used:

$$y_t = \varepsilon_t - 0.5\varepsilon_{t-1} + 0.4\varepsilon_{t-2} \tag{2.1}$$

where ε_t is Gaussian with mean 0 and variance 1, y_t- random process.

The autocorrelation function measures the correlation between y_t and y_{t+k}, where $k = 0, \ldots, K$.

According to [20], autocorrelation for lag k is equal to

$$r_k = \frac{c_k}{c_0} \tag{2.2}$$

Where

$$c_k = \frac{1}{T}\sum_{t=1}^{T-k}(y_t - \bar{y})(y_{t+k} - \bar{y}) \tag{2.3}$$

c_0 - sample variance of a time series.

Let's pretend that q — this is the lag, beyond which the theoretical autocorrelation function (ACF) is actually equal to 0. Then the estimated standard error of autocorrelation at lag is $k > q$ equal to

$$SE(r_k) = \sqrt{\frac{1}{T}\left(1 + 2\sum_{j=1}^{q} r_j^2\right)} \qquad (2.4)$$

If the series is random, then the standard error decreases to $\frac{1}{\sqrt{T}}$

In Fig. 2.5. an example of autocorrelation of an acceleration signal is presented. The top graph displays the acceleration from the sensor (raw), and the bottom graph shows the autocorrelation of the same signal.

Uses only Y and Z linear acceleration. X linear acceleration is not counted because it is more affected by phone shaking. In the first stage, data is collected to obtain the horizontal components of the perceived linear acceleration values, as well as to detect possible triggers of periodic oscillations. In the second step, correlation is used to identify possible correlated segments. These correlated segments can be thought of as user steps.

Disadvantages:

- Additional data processing required
- Possible data loss during correlation
- Speed

Advantages:

- Ability to change the pace
- Ability to change the speed of data writing
- Extensibility of the step detection algorithm
- Repeatability of the step pattern

This algorithm is unnecessarily complicated for a future system, due to its cumbersomeness and non-linearity, although this method gives one of the smallest errors in measuring the number of steps.

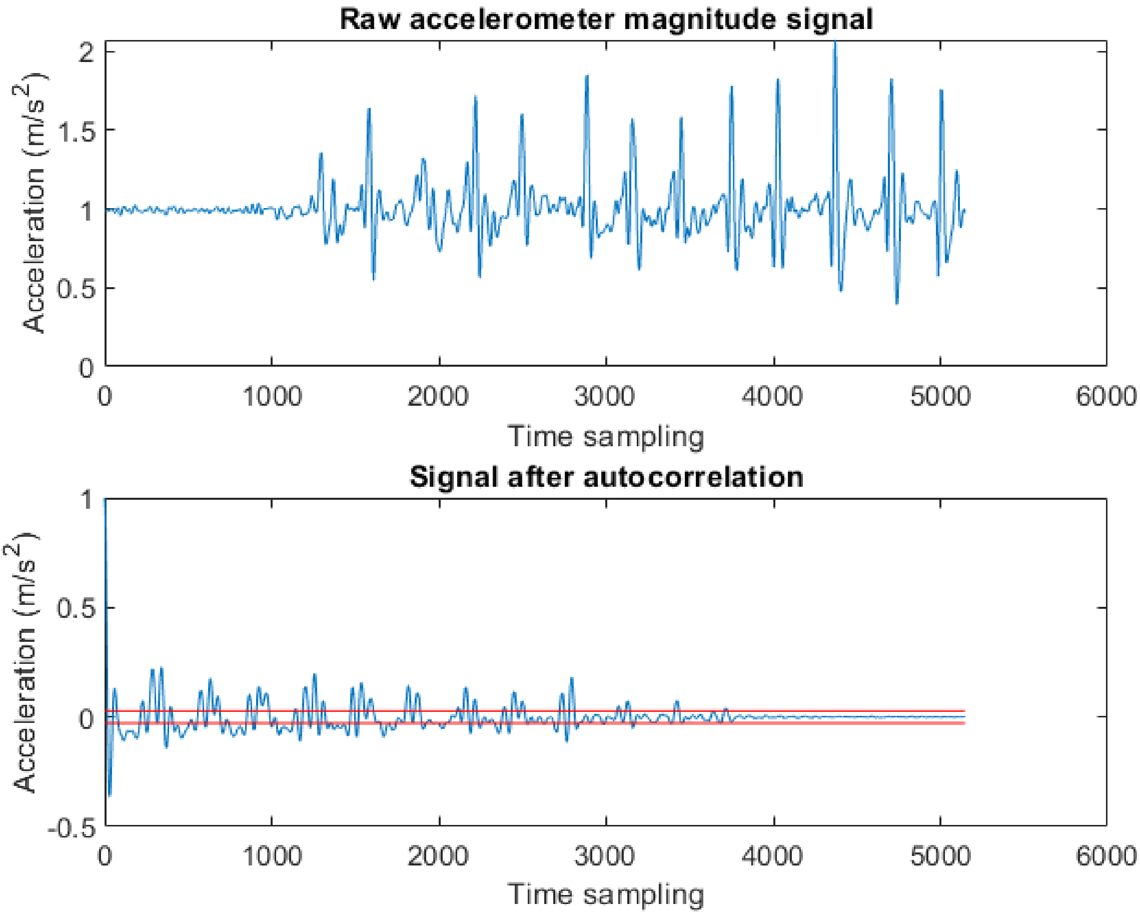


Fig. 2.5. Step detection based on autocorrelation

2.1.6. Method of determining direction based on the vertical angular velocity component

Vertical angular velocity component [24] – this method is based on calculating the heading with a gyroscope using the vertical component of the angular velocity vector, which can be extracted using the estimated direction of gravity. Denoting this vertical projection of the gyroscope through ω_v, its calculation at any time can be performed as follows:

$$\omega_v(t) = \hat{\gamma}(t)^T \omega(t) \tag{2.5}$$

where $\omega(t)$, a three-dimensional vector of angular velocities, should be obtained after appropriate filtering of the original gyroscope measurements in order to reduce the influence of random noise and high frequency interference. Based on the right-hand rule, ω_v measures the speed at which a pedestrian is turning in the horizontal plane, where positive (negative) speeds indicate a turn to the right (left). Thus, one of the approaches to calculating the course change is direct integration $\omega_v(t)$.

$$\psi(t) - \psi_s = \int_{t_s}^{t_f} \omega_v(\tau)\, d\tau \tag{2.6}$$

where $[t_s, t_f]$ - time interval of interest (e.g. start and end of data recording), and $\psi(t) - \psi_s$- heading angle in radians in time relative to its initial value (namely, cumulative heading change).

Disadvantages:

- Additional data processing required
- Filtration required
- Strict fixation to avoid unnecessary hesitation

Advantages:

- Simple method
- Speed

The fastest method for finding directions, but it has many drawbacks, and also has a cumulative error that very badly affects the data over time.

2.1.7. Method of determining direction based on the signed magnitude angular velocity

Signed magnitude angular velocity [24] – this is another approach to this problem - to use the magnitude of the angular velocity vector.

$$\omega_m(t) = \sqrt{\omega_x^2(t) + \omega_y^2(t) + \omega_z^2(t)} \qquad (2.7)$$

When ignoring measurement errors, the magnitude of the vectors does not depend on the frames in which they are measured, therefore, ω_m is independent of sensor orientation. In practice, however, systematic errors such as bias result in the measured value ω_m will have some orientation dependence.

Although the rate of interest is disturbed by other angular rates due to pedestrian movement, it is expected that the corresponding $\omega(t)$ will eliminate most of these, so the filtered measurement should approximate true horizontal rate of turn.

To distinguish between left and right turns, it is necessary to determine the sign $\omega_m(t)$ for each moment in time. This can be achieved by determining which axis (x, y or z) of the accelerometer is most dominant and using the gyroscope sign on that axis to determine the direction of rotation. This method can be unreliable in situations where two different axes are equally dominant (for example, when the smartphone is tilted 45 °, the force of gravity manifests itself in the same way both on their axis and on the z-axis). A different approach will help to avoid this, in order to distinguish between a turn to the left and right, by observing the sign $\omega_v(t)$.

$$\omega_{sm}(t) \equiv sgn\big(\omega_v(t)\big) * \omega_m(t) \qquad (2.8)$$

Where $sgn(\cdot)$ defined as ± 1 for positive / negative inputs and 0 otherwise. Finally, the calculation of the heading using this approach is performed as in the equation (6), but with replacement ω_v on ω_{sm}.

Disadvantages:

- Additional data processing required
- Filtration required
- Accumulation errors

Advantages:

- 3-axis freedom

More advanced than the vertical angular velocity component, but also has a cumulative error that will inevitably accumulate over time.

2.1.8. Method of determining the course based on the calculation of data from a magnetometer

Magnetometer-based heading calculation - A magnetometer is a 3-axis sensor that measures the external magnetic field, which is a combination of the Earth's geomagnetic field and local magnetic disturbances (e.g. due to ferromagnetic materials).

The magnetometer is characterized by hard-iron and soft-iron distortions. Hard-iron distortion [23] — it is an additive effect when a DC component is added to the measured field. This could be due to, for example, the action of a permanent magnet or the sensor's own zero offset. Soft-iron distortion is a multiplicative effect that reflects a change in direction and / or weakening of the magnetic induction vector.

This effect can be caused by the presence of a metal object in the immediate vicinity of the magnetometer or by intrinsic distortions of the sensor - an error of the scale factor or a sag of its sensitivity axis.

To determine the direction angle, the collected data on the strength of the earth's magnetic field in 3-axes, it is possible to calculate the direction using the four-quadrant arctangent from the data from the sensor, converting the force into an angle in radians corresponding to the coordinates of the direction of walking of the person. Then converted to a degree measure [0°,360°].

4-quadrant arctangent, returns values in a closed range $[-\pi, \pi]$ based on the values of the earth's magnetic force.

Disadvantages:

- Hard-iron and soft-iron
- The earth's magnetic field can change over time

Advantages:

- 3-axis freedom
- Has no cumulative error
- More accurate
- Fast method

This method is best suited for developing a dead reckoning system for a pedestrian. Since its implementation does not require deepening into quaternions and rotation matrices, which simplifies the calculation of the course, and also does not have a cumulative error.

2.2. Description of data collection

Sensors such as an accelerometer were used to collect data (accelerometer bmi160) 3-axis sensor with low power consumption [21] — it is a low noise 16-bit IMU, designed for mobile applications such as augmented reality or indoor navigation, providing highly accurate sensor data and real-time sensor data. Low current consumption BMI160 allows continuous operation in battery-powered devices. This sensor features a customizable on-chip interrupt engine that provides motion-based gesture recognition and contextual awareness.

Table 2.3 Technical characteristics of the accelerometer

Parameter	Technical data
Dimensions	2,5x3,0x0,8 mm^3
Temperature range	-40 ... + 85 ° C
Supply voltage	1.2 ... 3.6 V
Consumption current	20 mkA
Sensitivity	± 2 г: 16384 μT / g

A 3-axis magnetometer was also used for data collection. AK09918 (magnetometer mag-akm09918) — it is a 3-axis electronic compass IC with highly sensitive Hall sensor [22]. Small body AK09918 includes magnetic sensors for detecting terrestrial magnetism along the X, Y and Z axes, a sensor control circuit, a signal amplifier circuit and an arithmetic circuit for processing the signal from each sensor. There is also a self-test function. Thanks to its compact size and thin packaging, it is suitable for use in smartphones when navigating the map to realize the pedestrian navigation function.

Table 2.4 Technical characteristics of the magnetometer

Parameter	Technical data
Dimensions	0,76x0,76x0,76 mm^3
Temperature range	-30 ... + 85 ° C
Supply voltage	1.65 ... 1.95 V
Consumption current	4 mA
Sensitivity	0,15 μT

Since gravitational acceleration is involved in the data, then the average acceleration vector is approximately equal to the gravitational acceleration vector. Thus, discarding acceleration vectors that are too large and too small in magnitude in order to reduce the number of vectors responsible for the acceleration of a moving person, and then getting the average vector from the remaining, can be determine what is the direction of the Z-axis of the person in the phone's coordinate system. Then, a rotation matrix so that the Z-axis of the phone Fig. 2.6. coincided with the Z-axis of the person Fig. 2.7., And change the data of the accelerometer and magnetometer. When writing data, the Z axes coincided, the need to complete the rotation matrix did not appear.

The data were recorded in a time sample with a data collection frequency close to 100 Hertz [Hz], that is, if the sample length was 60000, its approximate time can be easily calculated using the formula $t = \frac{N}{f}$, which equals 600 seconds or 10 minutes.

Where t — this time is in seconds, N – sample length, f – data recording frequency.

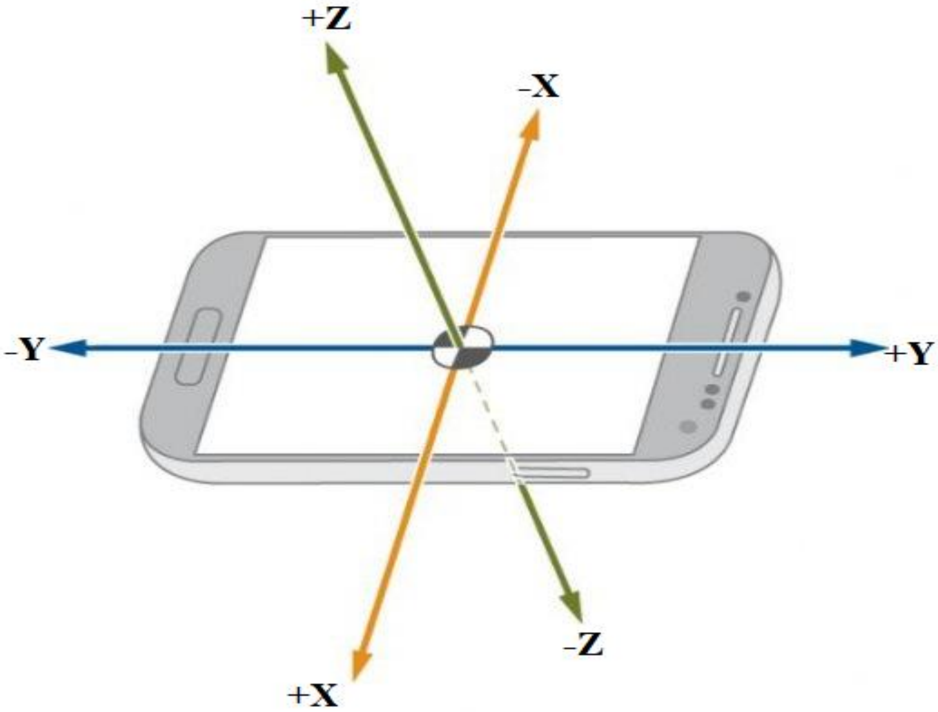


Fig. 2.6. Spatial axes of sensors in the phone

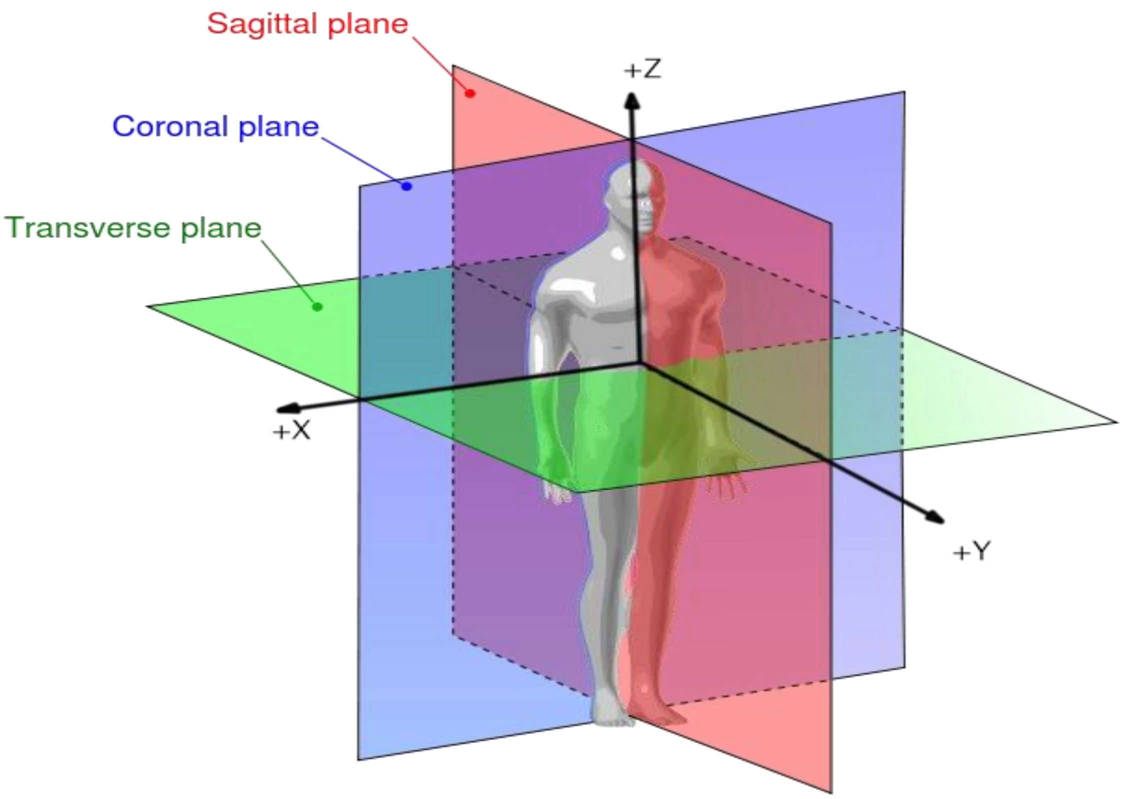


Fig. 2.7. Human axes in space

2.3. Conclusion

Methods for detecting a person's step were considered, and each method was described from a negative and positive side for this work. A specific method was chosen, namely finding the step peaks, which is more suitable for this job than the other methods described.

Methods for finding directions are also considered. Of which one was chosen, the calculation of the course based on the magnetometer, due to the large advantage of this method over the others, since it does not have a cumulative error.

The programs that will be needed to implement algorithms and write an application are considered.

CHAPTER 3. OPERATING ALGORITHM FOR COUNTING STEPS USING THE PEAK DETECTION METHOD AND CALCULATING PEDESTRIAN DIRECTION USING A MAGNETOMETER

3.1. Data processing algorithm

The developed algorithm for finding the steps and direction of the pedestrian:

1. Writing data to the device
2. Loading data into the processing environment
   a. Processing of the received data
   b. Accelerometer
   c. Magnetometer
3. Calculation of the magnitude of the acceleration vector
4. Filtration of low peak acceleration fluctuations
5. Determination of all possible peaks
6. Finding groups of peaks and false peaks
7. Determination of the main peaks in groups
8. Dividing peaks into steps
9. Calculation direction
10. Route calculation
11. Layering the route on the global map
12. Comparison of GPS pedestrian path and pedestrian path using the phone's internal sensors

3.1.1. Writing data to the device

Data is taken from the internal MEMS sensors of the phone: Accelerometer, GPS, magnetometer.

Fig. 3.1. Data recording

All data is taken from 3-axis digital sensors, accelerometers measure the change in linear motion, applying the principle of capacitive detection, the principle of detection is based on recording the change in capacitance between sensitive elements. Magnetometer 3-axis digital geomagnetic sensor that meets the requirements of compass applications. Provides spatial orientation and motion vectors with hardware-adapted sensor fusion software. Using together with an inertial measuring unit, it is possible to determine the route of a pedestrian. GPS sensors measure position, latitude, longitude, as well as the speed and determine the number of visible GPS satellites at a given time of recording. This is done with at least four satellites so that the phone can determine its position from several directions, which allows the smartphone to determine its x, y coordinates on the earth's surface.

3.1.2. Loading data into the processing environment

After the data has been taken, further processing of this data is necessary; for this, it is necessary to load it into the processing environment. Using the MATLAB environment, whose internal functions allow loading and processing of data. More about Fig. 3.2, in the Appendix A.

time	gFx	gFy	gFz	wx	wy	wz	Bx	By	Bz	Latitude	Longitude	Speedms
Number	Number	Number	Number	Number	Number	Number	Number	Number	Number	Number	Number	Number
time	gFx	gFy	gFz	wx	wy	wz	Bx	By	Bz	Latitude	Longitude	Speed (m/s)
3.8180	0.0202	0.1829	0.9541	-0.0820	-0.0564	-0.0277	-17.9375	-21.0625	-35.5625	46.6621	32.6398	0
3.8210	0.0145	0.1840	0.9649	-0.0820	-0.0564	-0.0277	-17.9375	-21.0625	-35.5625	46.6621	32.6398	0
3.8230	0.0145	0.1840	0.9649	-0.0703	-0.0724	-0.0170	-17.9375	-21.0625	-35.5625	46.6621	32.6398	0
3.8270	0.0145	0.1840	0.9649	-0.0703	-0.0724	-0.0170	-18.3125	-20.9375	-35.5000	46.6621	32.6398	0
3.8280	0.0183	0.1860	0.9636	-0.0703	-0.0724	-0.0170	-18.3125	-20.9375	-35.5000	46.6621	32.6398	0
3.8280	0.0183	0.1860	0.9636	-0.0490	-0.0852	0.0053	-18.3125	-20.9375	-35.5000	46.6621	32.6398	0
3.8290	0.0183	0.1860	0.9636	-0.0490	-0.0852	0.0053	-18.3125	-20.9375	-35.5000	46.6621	32.6398	0

Fig. 3.2. Loading data

After loading the array of all data, it is necessary to distribute them separately, each in its own variable. For post-processing, filtering, step counting and pathfinding.

3.1.3. Processing of incoming data

a) Initially, it is necessary to calculate the module of the acceleration vector for a complete display of the data taken from the phone, which has the form of an oscillatory system in time, since the accelerometer sensor moves all the time when walking, while the gravity force acts on 1 of its 3 axes all the time, during the movement this vector shifts, further, to simplify calculations, filters are superimposed, such as filtering low peak fluctuations, acceleration. And all data will be displayed in the MATLAB environment.

b) To calculate the direction, it is necessary to calculate the heading direction of the pedestrian's movement. Using a magnetometer, it is the strength of the magnetic field that induces the earth's magnetic field on these sensors. Then calculate the direction matrix for the future use in finding the path of the pedestrian.

3.1.4. Calculation of the magnitude of the acceleration vector

It is necessary to find the module of the acceleration vector for all data in time for a general idea of the human step. Calculate the magnitude acceleration vector using the following formula.

$$\mathrm{A}_m = \sqrt{\mathrm{A}_\mathrm{x}^2 + A_y^2 + A_z^2} \qquad (3.1)$$

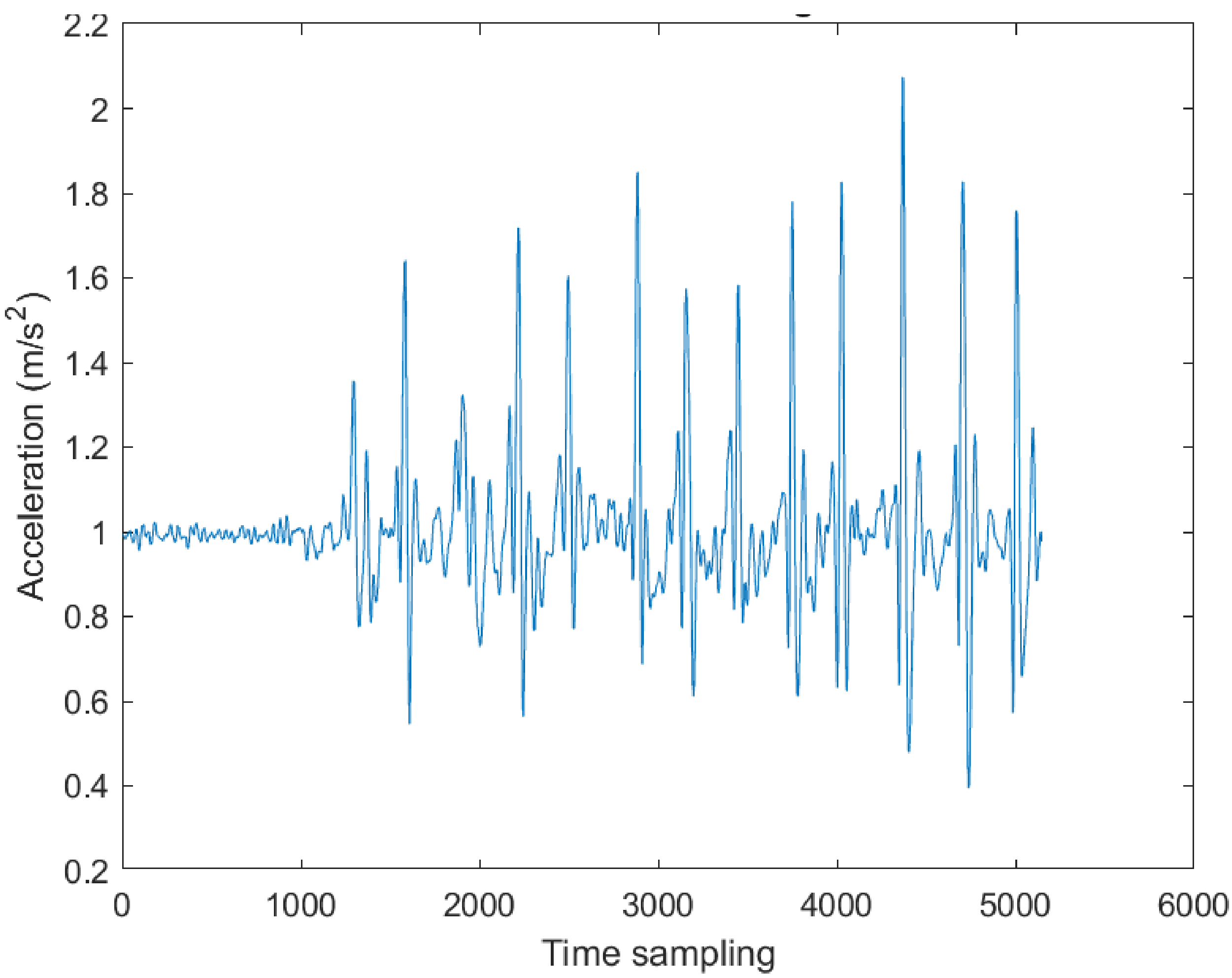


Fig. 3.3. The magnitude of the acceleration vector

3.1.5. Filtering low peak fluctuations Acceleration

To simplify the reckoning of steps, it is necessary to remove the minimum acceleration peaks. The minimum threshold that needs to be filtered can be obtained by analyzing data from 0 steps, that is, when the phone is on a flat surface without hesitation at rest. In the process of research, it was experimentally obtained that the required filtration height is in the range up to $A_{filter} = 1.2\frac{m}{s^2}$.

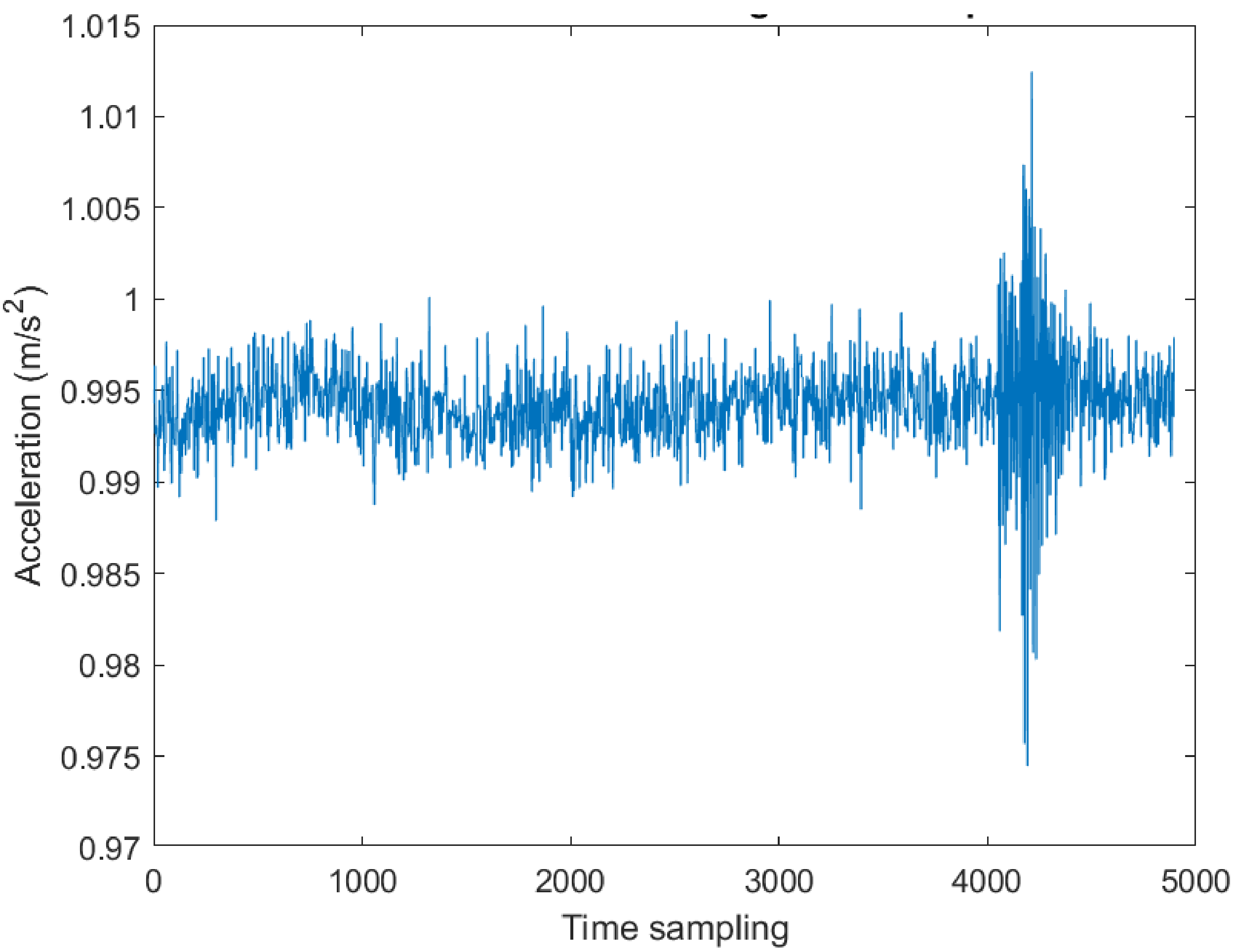


Fig. 3.4. Acceleration vector magnitude 0 step

If $A_m(i) \leq A_{filter}$ this condition is satisfied then $A_m(i) = 1$ after which the acceleration magnitude vector takes the form of a truncated vector, since this algorithm uses a detector that determines the peaks of the step, then do not need the bottom of the truncated vector, as for example, for the ZC algorithm.

Therefore, the informational content when cutting the lower boundary nowhere disappears, as it was when using the algorithm, the vectors were cured through 0. After applying filters to the data of the acceleration vector, it looks like Fig. 3.5.

In Fig. 3.5. Shown are all peaks of a person's steps in the future, which fall under 2 filtering 1 - determination of all possible peaks, 2 - determination of the main peaks in groups.

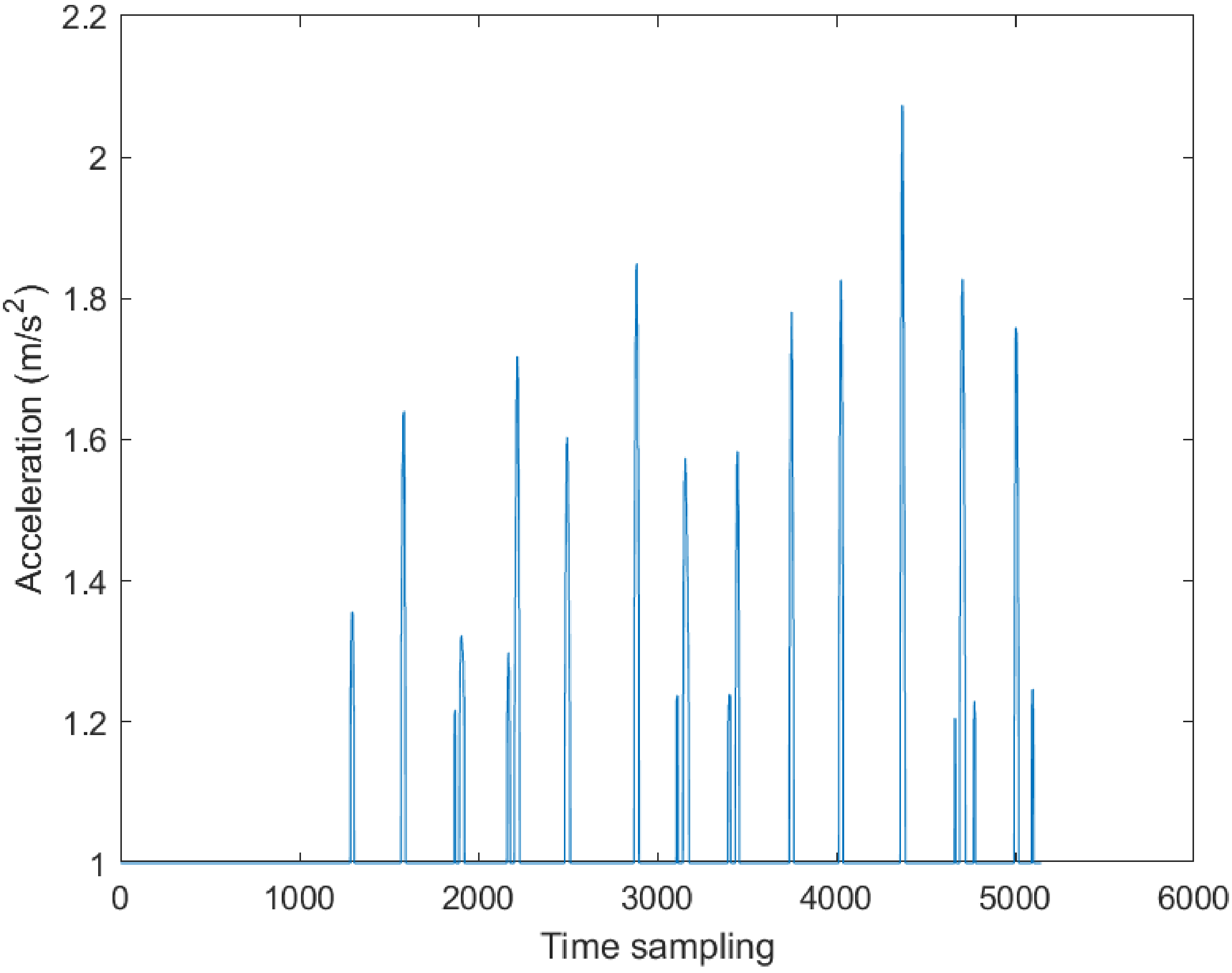


Fig. 3.5 Minimum Peak Threshold

3.1.6. Determination of all possible peaks

At this stage, it is necessary to find all possible peaks and their location in the timeline if the condition, $A_m(i+1) - A_m(i) < 0$ this means that the current value is the end of the peak. Thus, the points of all possible peaks are determined, it is necessary to write the data on the peaks into a matrix of data of the height of the peaks and finding the peak in time, for further processing of the data and simplification of actions on them.

In this case, the initial data array changes slightly and looks like Fig. 3.6.

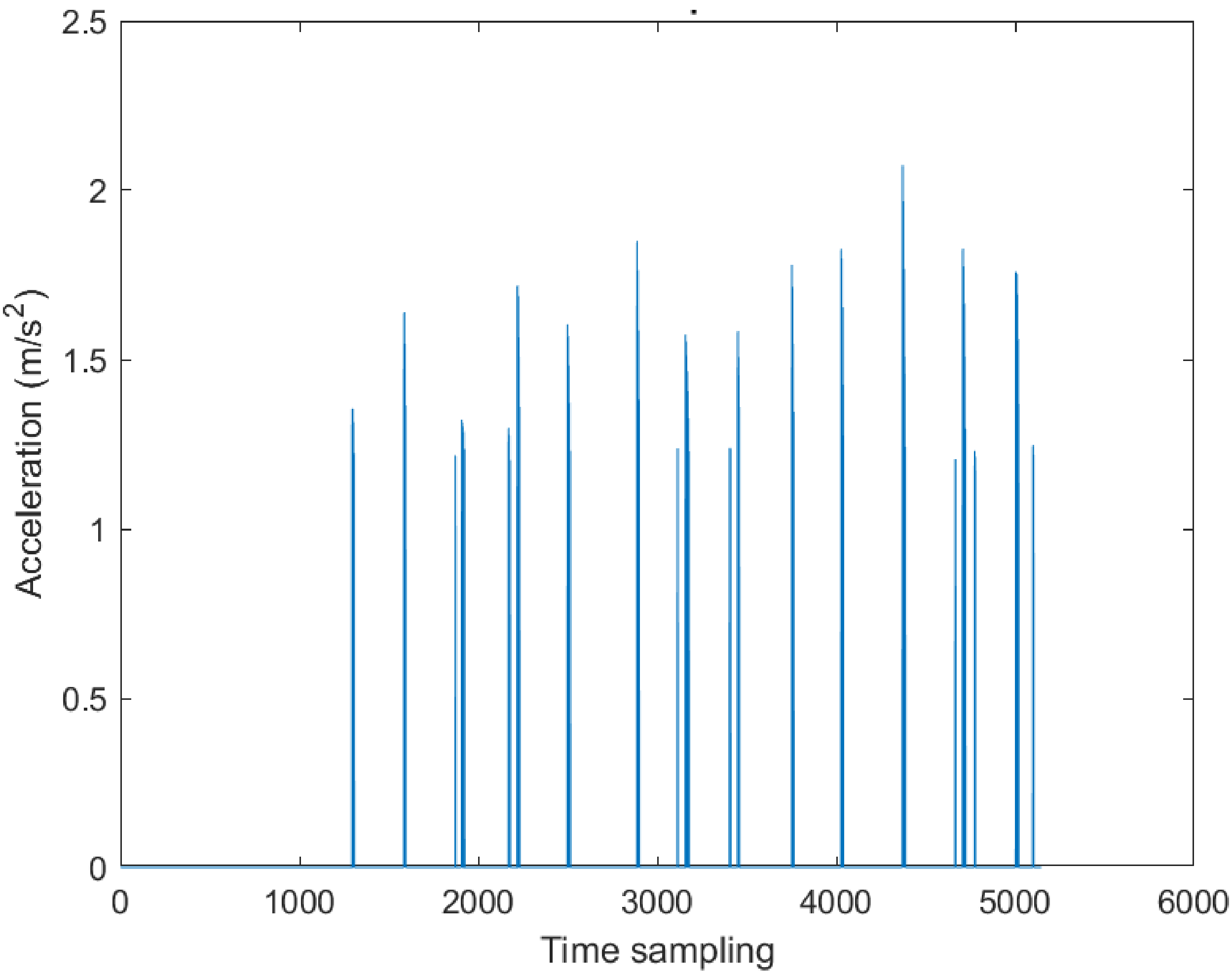


Fig. 3.6. All found peaks

3.1.7. Finding groups of peaks and false peaks

Next, a peak distance filter is used to help identify false peaks that are part of the step, and also need to find areas where the peaks are separated depending on how far they are from each other. If the distance of the found peaks is greater than the preset filtering threshold described in paragraph **3.2,** $A_{Local\ Piks\ Step}$, it splits into different peaks of steps.

The remaining peaks and the remaining peaks also need to be filtered from the group of peaks of one step and select its main peak.

For this data filtering, it is necessary with respect to the pops that form the.

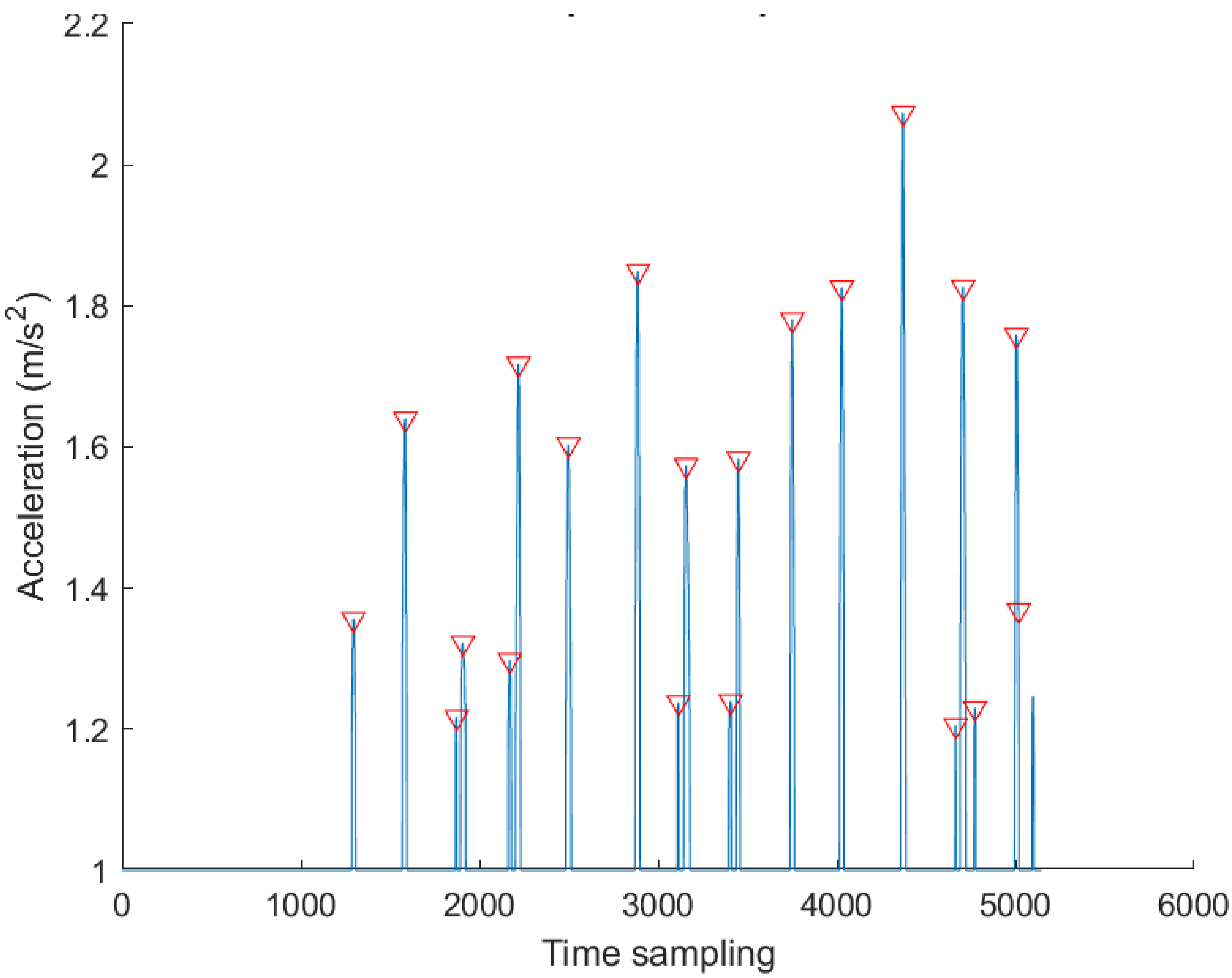


Fig. 3.7. Found unprocessed peaks

3.1.8. Determination of the main peaks in groups

As seen from Fig. 3.7, some peaks have false peaks located near the main peaks, they do not fall under filtering 5, since their height is much greater than the filtering threshold, in order to remove unnecessary false under peaks, it is necessary to find the main peak in the subgroups of these peaks.

That is, if there is a group of peaks in which one or more peaks are much lower in amplitude than another peak, then this peak is not included in the general data, this means that the peak is only a component of the step, and not the step itself, after that stop taking it into account as the peak of the step. The filtering criterion also serves as a division into groups. **3.2,** $A_{Local\ Distance\ Peaks}$ at a distance of peaks located at a certain distance from each other.

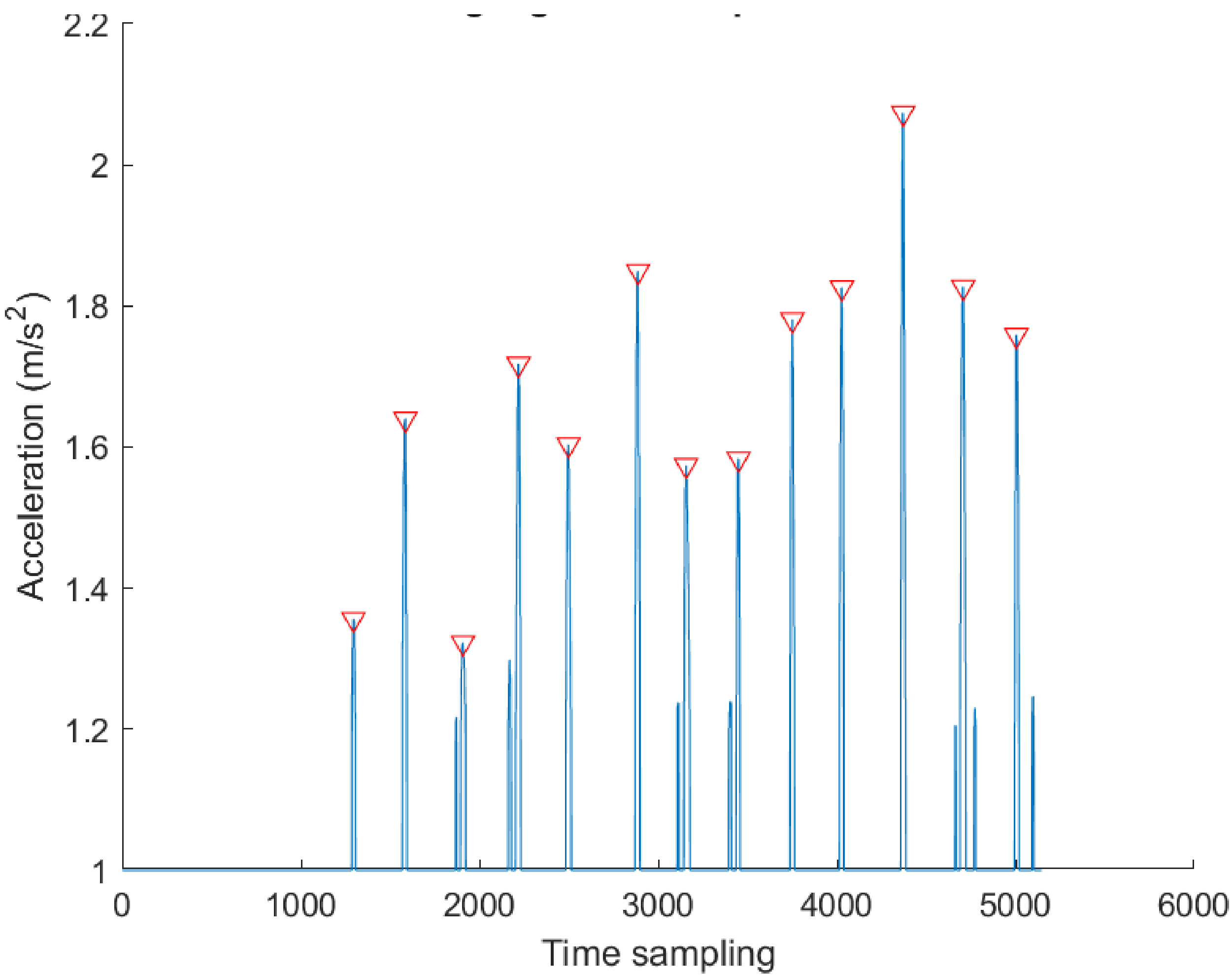


Fig. 3.8. Highlighted main peaks

3.1.9. Splitting peaks into steps

The last step in the search for the number of steps is to leave only the peaks of steps that were filtered in the previous steps. Also, to miscalculate the direction of the course, it is necessary to record the positions of the peaks found, since in subsequent steps, the step will be superimposed on the current course and a route map of the distance traveled will be drawn.

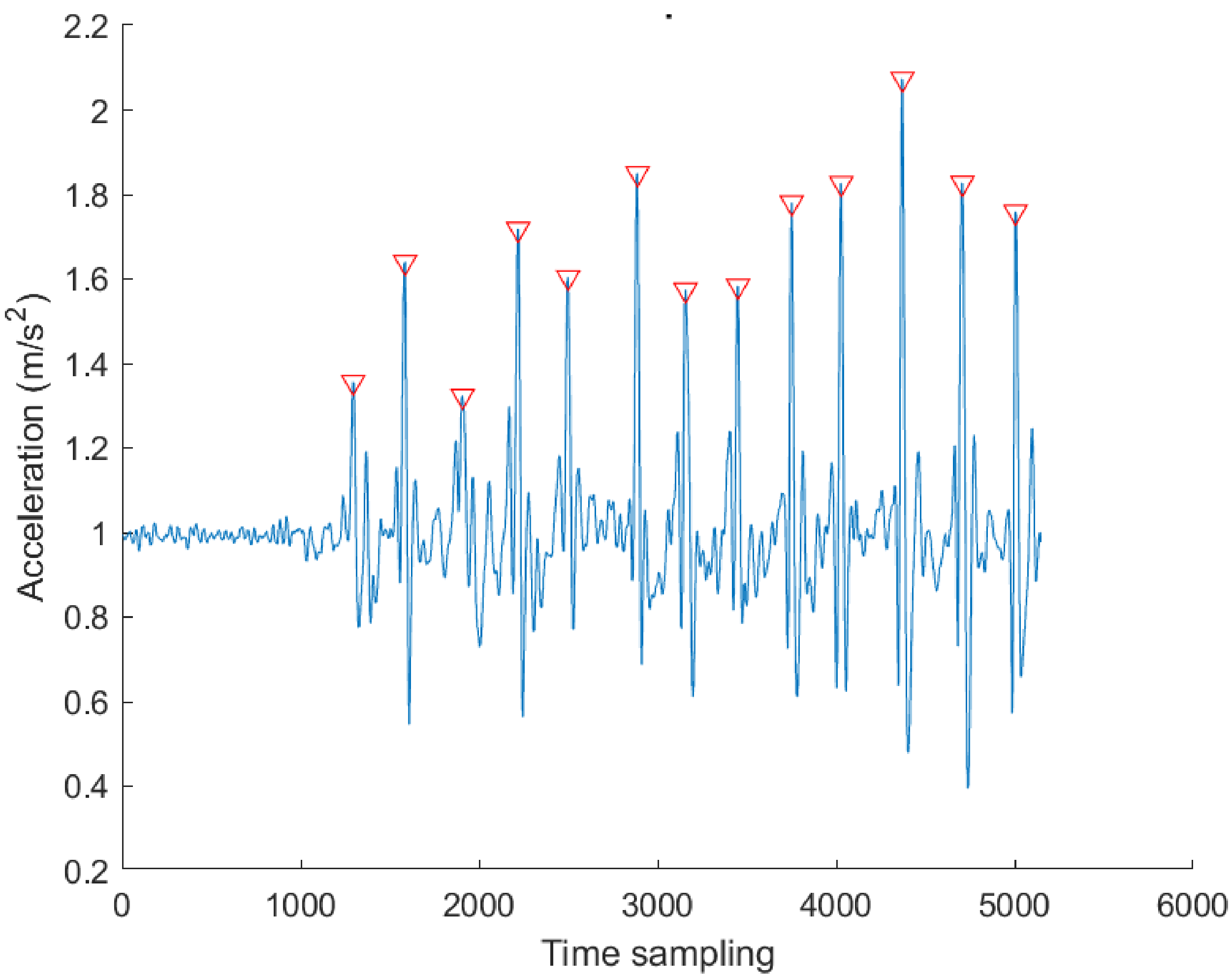


Fig. 3.9. All steps

3.1.10. Calculation direction

Calculation of direction, i.e. angle Θ considers the data collected from the magnetometer built into the device, knowing the magnetic force of the earth along two axes, it is possible to calculate the direction using the four-quadrant arctangent from the data from the sensor, converting the force into an angle in radians corresponding to the coordinates of the direction of walking of a person. Then converted to a degree measure $[0^\circ, 360^\circ]$.

4-quadrant arctangent, returns values in the closed interval $[-\pi, \pi]$ based on the earth's magnetic force values.

If the device can calculate the direction using its internal sensors, the advantage of this data is transmitted.

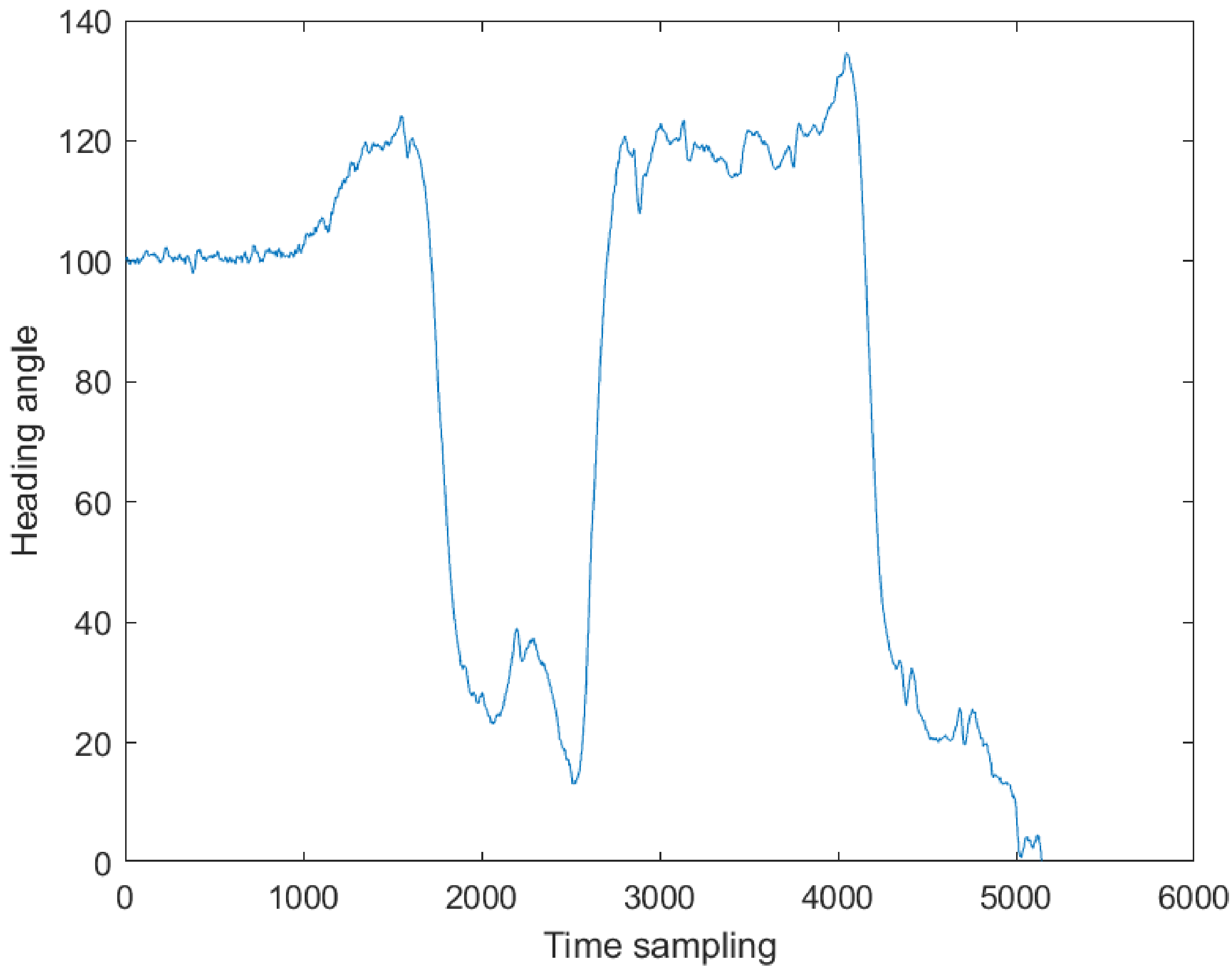


Fig. 3.10. Heading angle

3.1.11. Heading calculation

Using the angles, the direction and the pre-calculated number of steps, as well as their position relative to the timeline, the route that the pedestrian traveled during the data recording is built. Using the transition from a rectangular coordinate system to an great circus coordinate system, as well as changes in the lengths of parallels depending on the latitude at which the test object is located, according to the standards, which significantly increases the display accuracy. If the user cannot receive a GPS signal, he has the opportunity to indicate his approximate coordinates, since the inertial system will only show the distance traveled relative to the initial reference point in meters, while the world direction is preserved relative to the coordinate axes where the positive X-axis is the East direction, and the positive Y-axis is the North, West and South directions, respectively, opposite.

$$X(i+1) = cos(\Theta(i) + 90) * l + X(i)$$
$$Y(i+1) = \sin(\Theta(i) + 90) * l + Y(i) \quad (3.2)$$

Where l – this is the step length, Θ – angle direction, (X, Y) – coordinates.

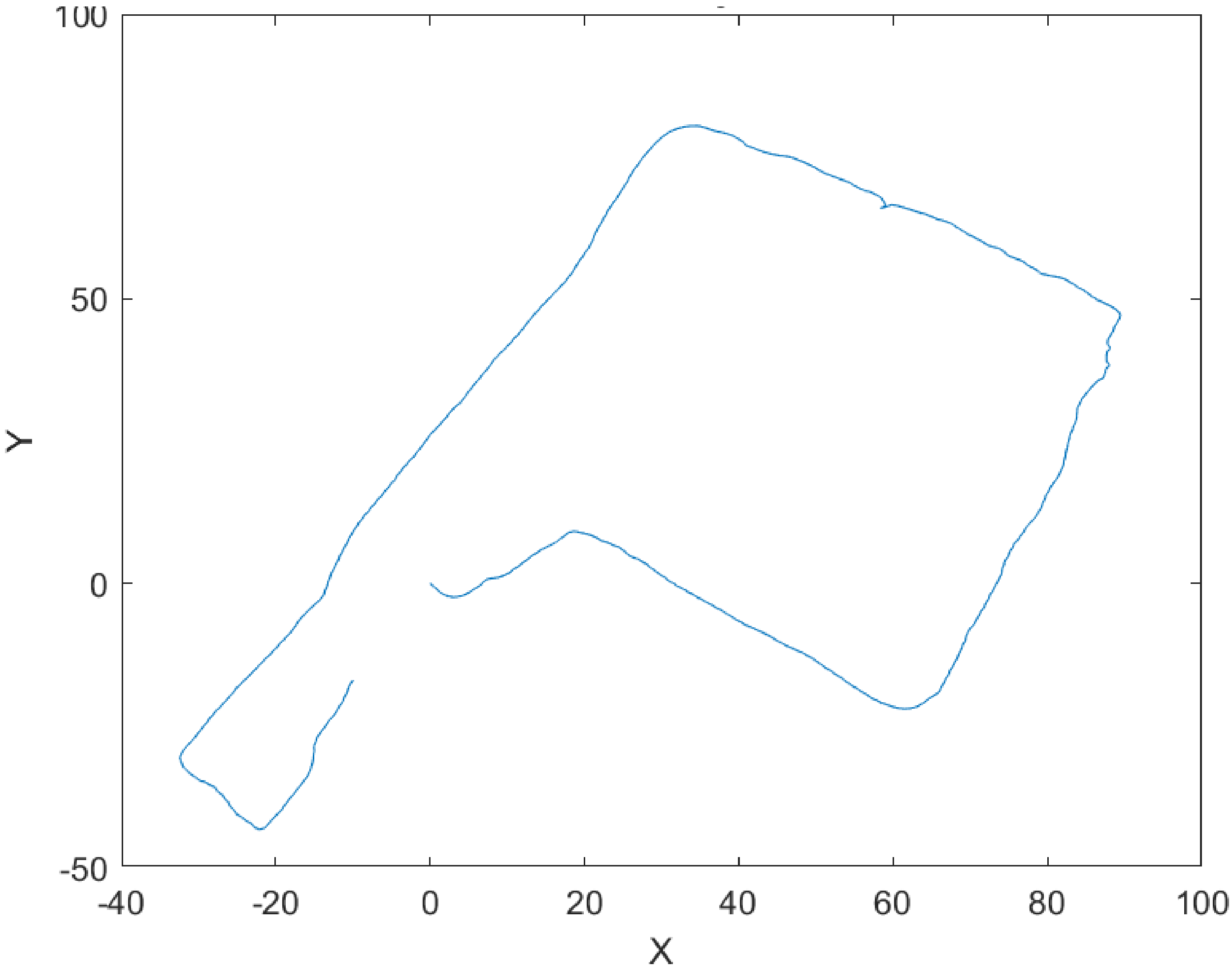


Fig. 3.11 Pathway

3.1.12. Layering a route onto a global map

The resulting route must be overlaid on a map of the earth in order for the pedestrian to be able to navigate the terrain. This requires GPS coordinates from the start of data recording or manual input from the user, considering the geographic coordinate system and the transition to it from the usual coordinate system. Since Latitude is not constant, it must be recalculated depending on longitude to obtain the most accurate values.

$$lat(i+1) = lat(i) + \frac{cos(\Theta(i)) * l * 360}{40000000}$$
$$lon(i+1) = lon(i) + \frac{-sin(\Theta(i)) * l * 360}{40075696 * cos(lat(i))} \tag{3.3}$$

Where (lat, lon) – geographical coordinates.

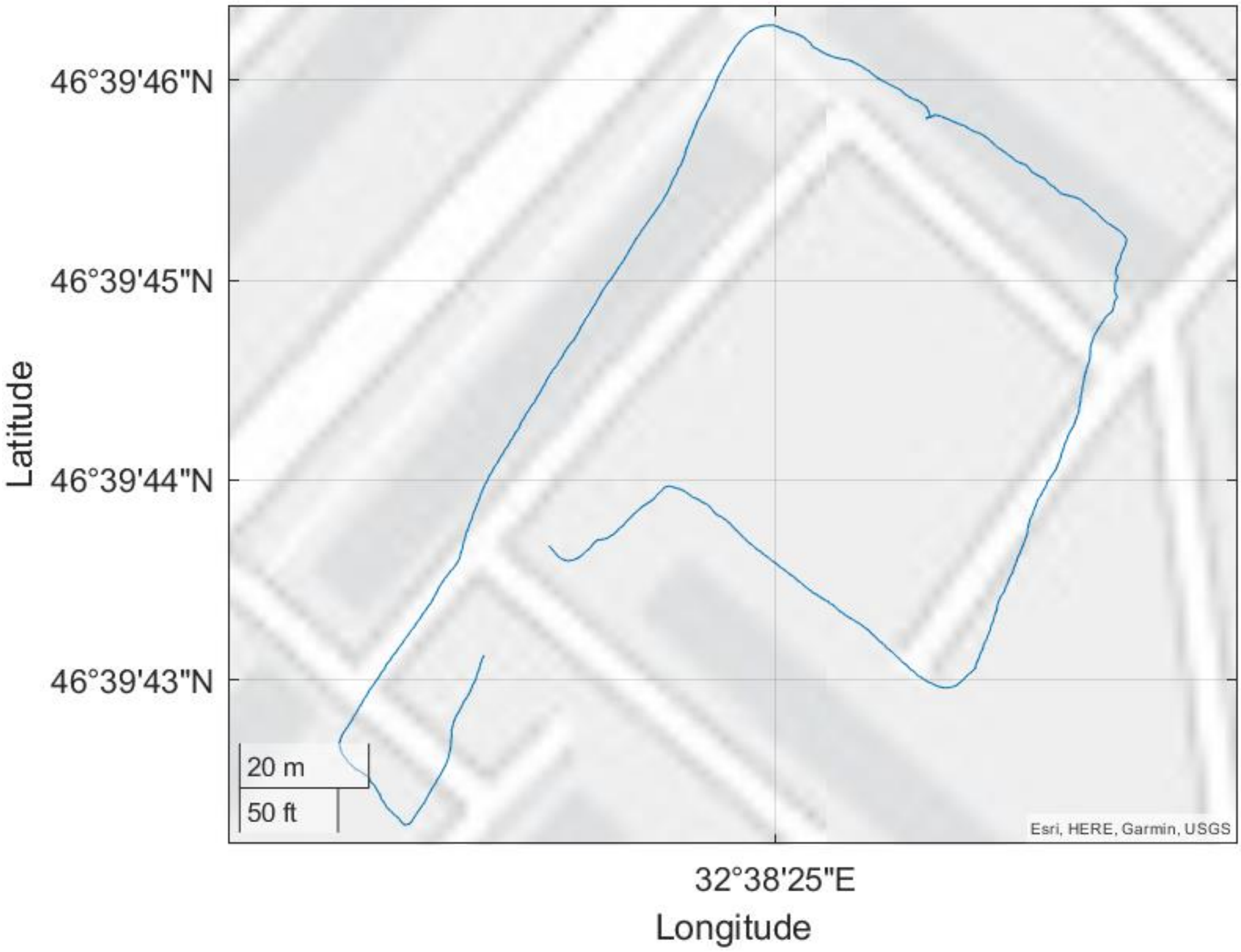


Fig. 3.12. The plotted route on the earth map

3.1.13. Comparison of GPS pedestrian path and pedestrian path using internal phone sensors

After receiving the route using inertial sensors, it is necessary to check its accuracy and correctness in relation to GPS data. Since the data from GPS can have an error of 15 meters with the best connection with the satellite, then the route constant with the help of inertial sensors should have relatively the same accuracy.

In Fig. 3.13, 2 paths are plotted in blue, the path of the GPS signal, in red, the path built using the sensors of the phone. As seen in Fig. 3.13, error is in the norm of permissible.

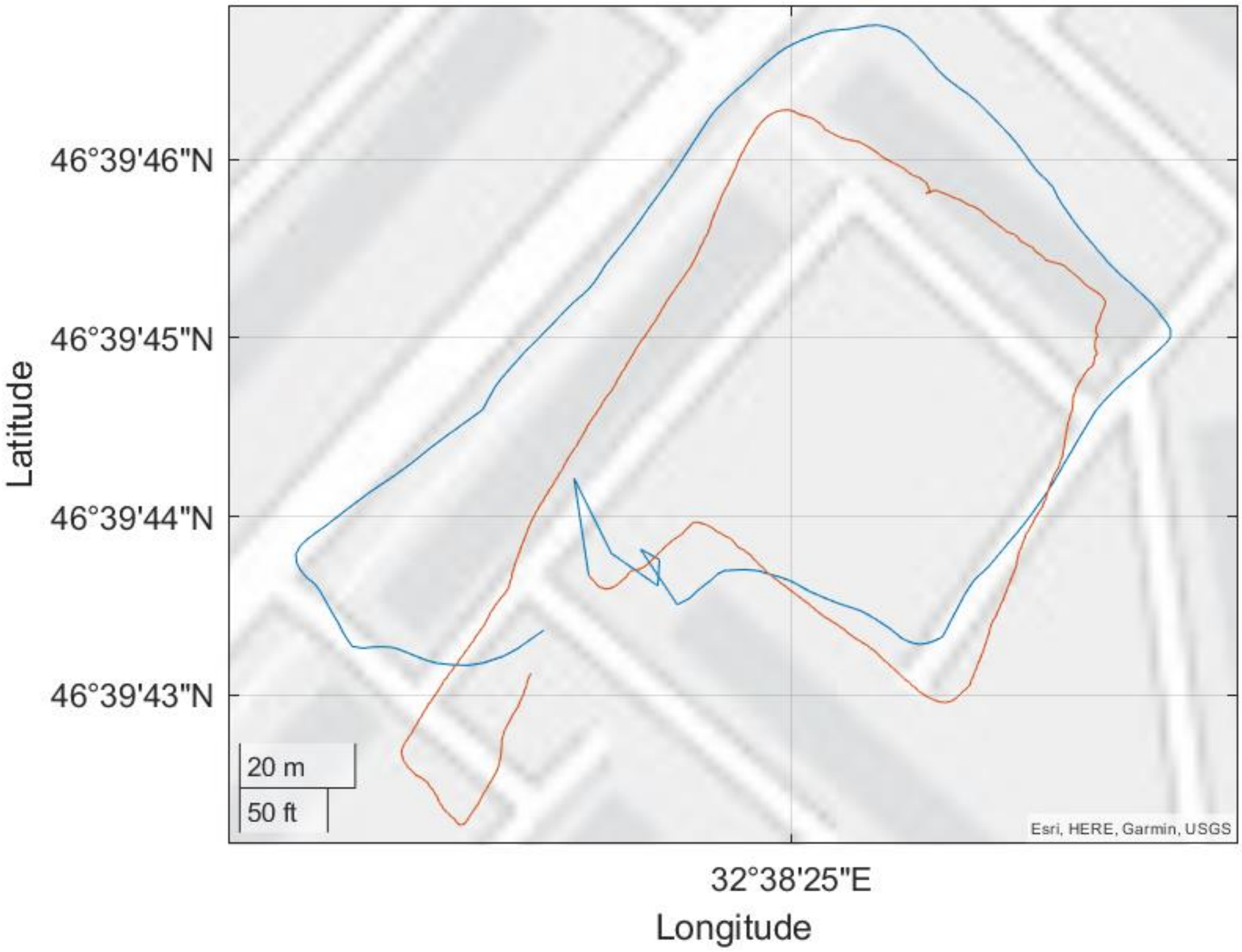


Fig. 3.13. Comparison of the pedestrian path

3.2. Filtration coefficients

In the process of analyzing different time intervals, it was determined that each of the recorded data differs in the frequency of data collection, depending on the system load, the settings of the data collection application, and the device used by the pedestrian to read the data.

Finding steps by peaks is a method that was used to determine at what point in time a person took a step. To do this, several conditions must be met so that the sensors that pick up the signal always work with the same frequency, but after the tests it was found that it is not always possible to achieve this for this, one of the filtering coefficients is how far the steps are from each other in the time scale, more precisely in the frequency scale, that is, the total amount of all data over time. The factor is scaled depending on the characteristic of the step, the time between steps, as well as the time of the step itself. There is also a filtration coefficient A_{filter} which serves as a criterion for filtering out the minimum peaks to simplify the counting of steps, the minimum threshold that needs to be filtered can be obtained by analyzing data with 0 steps, that is, when the phone is on a flat surface without hesitation at rest. In the process of research, it was experimentally obtained that the required filtration height is in the range up to $A_{filter} = 1.2\frac{m}{s^2}$.

The coefficients for finding the peaks of steps themselves, as well as the distance between them $A_{Local\ Peaks\ Step}$ and $A_{Local\ Distance\ Peaks}$ respectively. They are located in relation to the length of the peak relative to Fig 3.14, as well as the distance between the peaks in Fig 3.15, for each specific case, different values of these coefficients are used, which makes it possible to achieve high accuracy in counting in steps, as well as building a route that depends on steps.

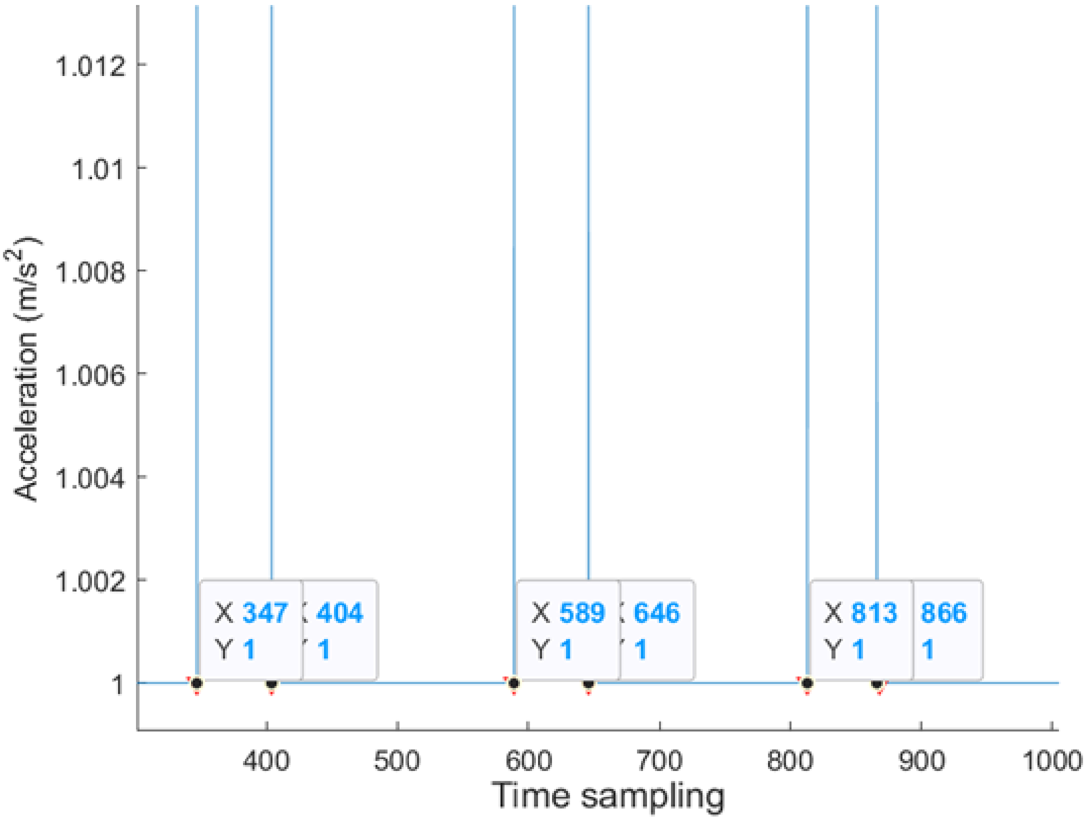


Figure 3.14. Time step period

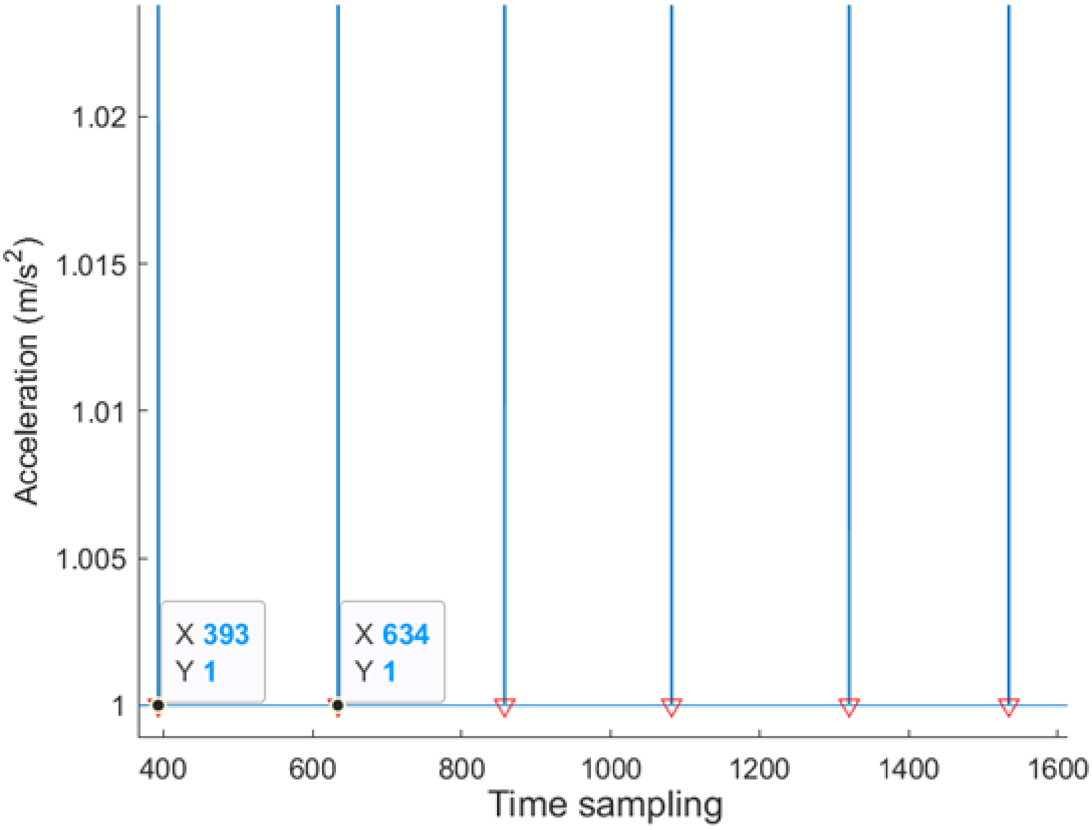


Fig 3.15. Length of time between steps

3.3. Comparison of results

In the course of the study, data from several people of different genders were collected, which can now be compared with respect to the data obtained after performing the filters and the actual readings of the number of steps. The table contains such parameters as the brand of the phone, and the gender of the person who measured the data, as well as the number of steps taken by him. After that, the data obtained by means of calculations is entered, and after that the measurement error is considered.

Table 3.1 Calculated number of steps and measurement error

№	Phone Brand	Gender (M / F)	Actual number of steps	Calculated number of steps	Error, %
1	Honor 8X	M	0	0	0
2	Honor 8X	M	13	13	0
3	Honor 8X	M	80	80	0
4	Honor 8X	M	130	128	1.5385
5	Honor 8X	M	150	149	0.6666
6	Honor 8X	M	198	199	0.5050
7	Honor 8X	M	208	209	0.4807
8	Honor 8X	M	506	502	0.7905
9	Honor 8X	M	1053	1056	0.2849
10	Iphone	F	100	98	2
11	Iphone	F	200	197	1.5
12	Iphone	F	300	296	1.3333
13	Iphone	F	400	404	1

Figure 3.16, shows the actual number of steps and the calculated number of steps. As can see, the discrepancies are almost insignificant, even for long distances, so it can be argued that this method of finding steps does not have a cumulative error, that is, it does not accumulate error over time, since the step pattern is basically the same during walking. The method will also work when a person is resting, or finding him at a traffic light is not mobile as it will be active A_{filter} filtration.

Figure 3.17, the calculated error of the algorithm is shown, as can see it does not exceed 2% of the actual values, which is a good indicator for inertial calculation.

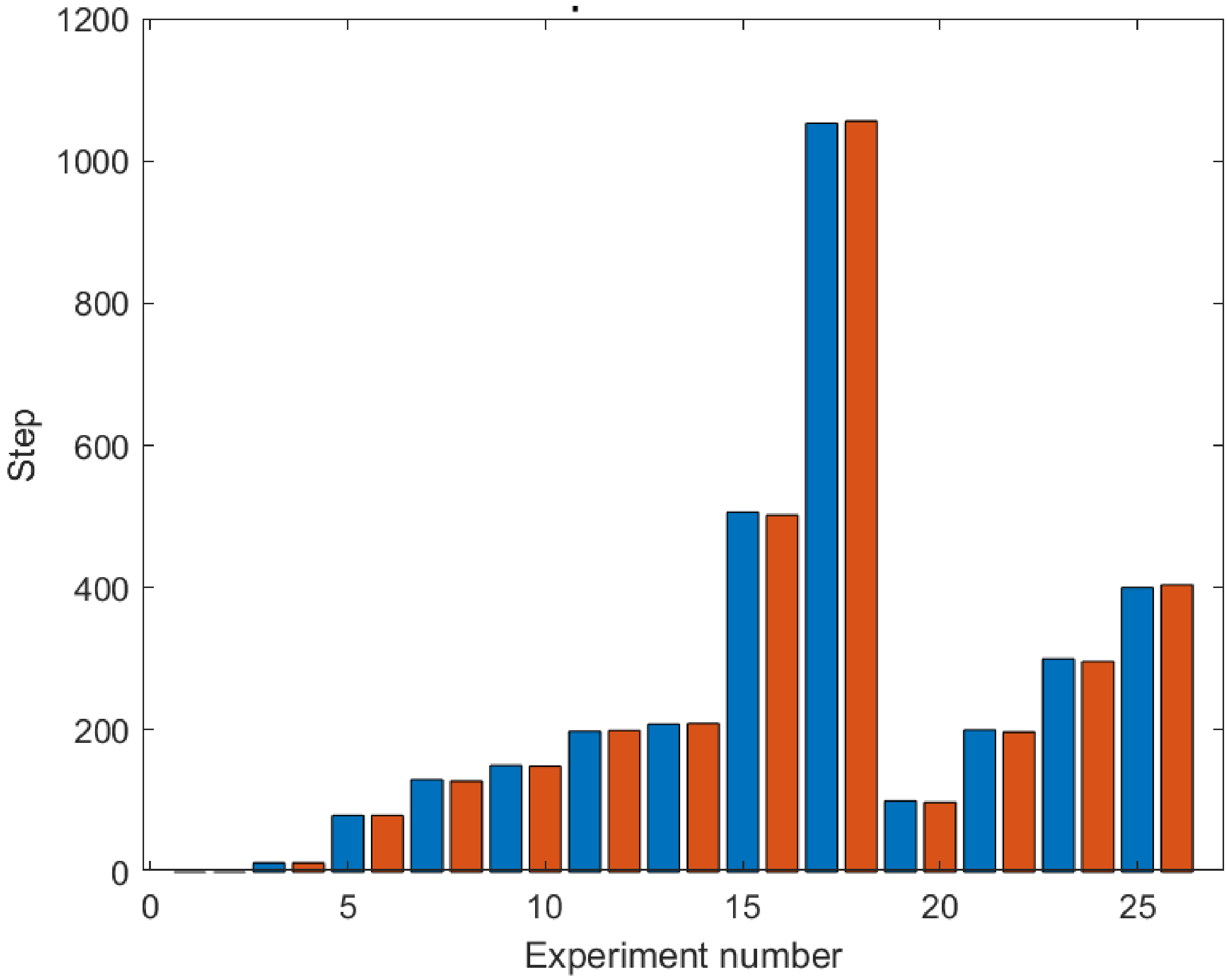


Figure 3.16. Comparison of results

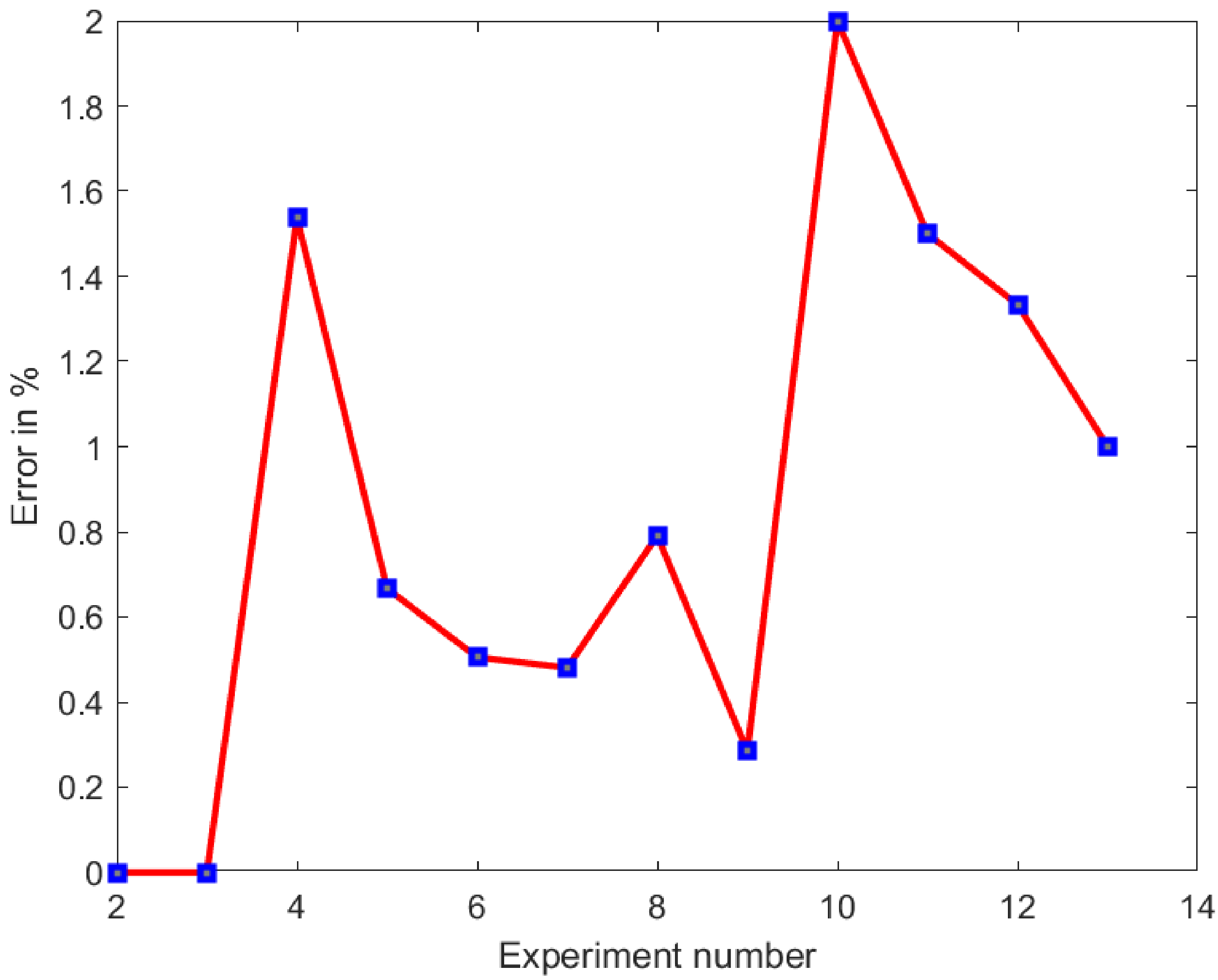


Fig 3.17. Calculation error

3.4. Software development

Algorithm-based **3.1,** a GUI application was developed based on MATLAB, App_PDR. The program follows all points of the above algorithms 1-13. The system in Fig. 3.18, represented by a mobile phone from which data is taken, a computer on which the main processing takes place, and the App_PDR application, which processes all data, and also displays data about the user's location.

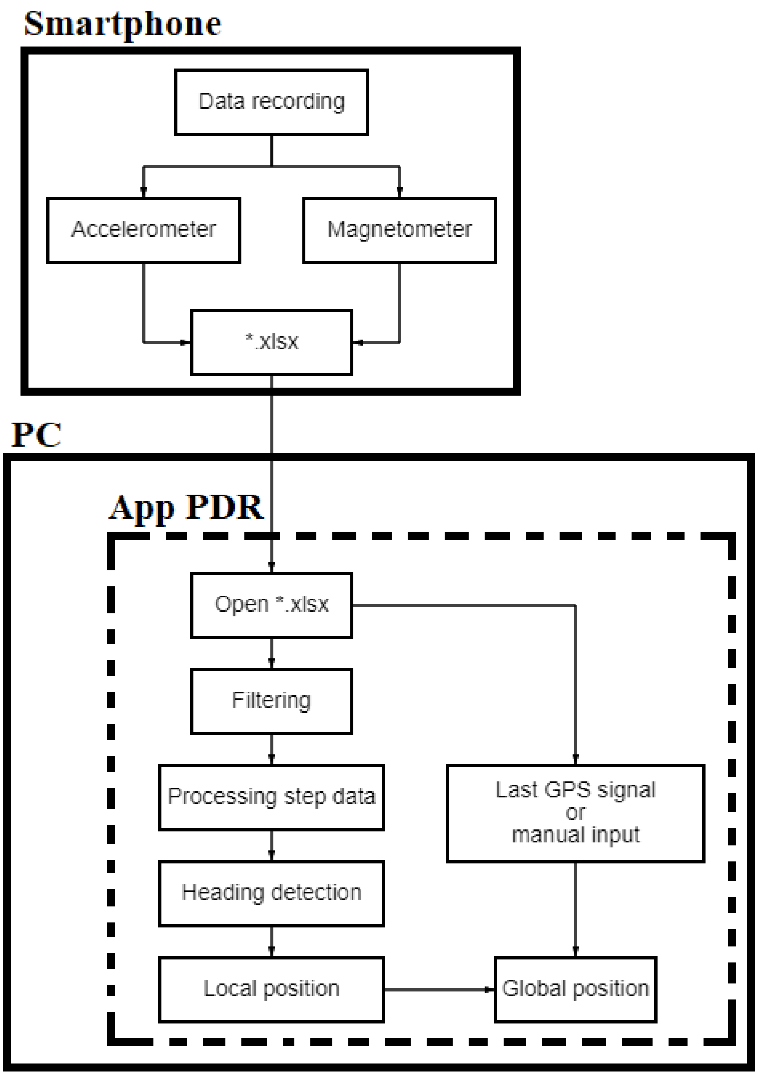


Fig. 3.18. Inertial navigation system diagram ☹

In Fig. 3.19 1-2 points of the algorithm are shown, loading data into the application for its next processing.

To download data, users need to click the “Loading data” button and select the file in the pop-up window and click “Open”.

In the absence of data or the wrong format, the program will display an error in the “Name xlsx” window and ask to enter correct data from the user.

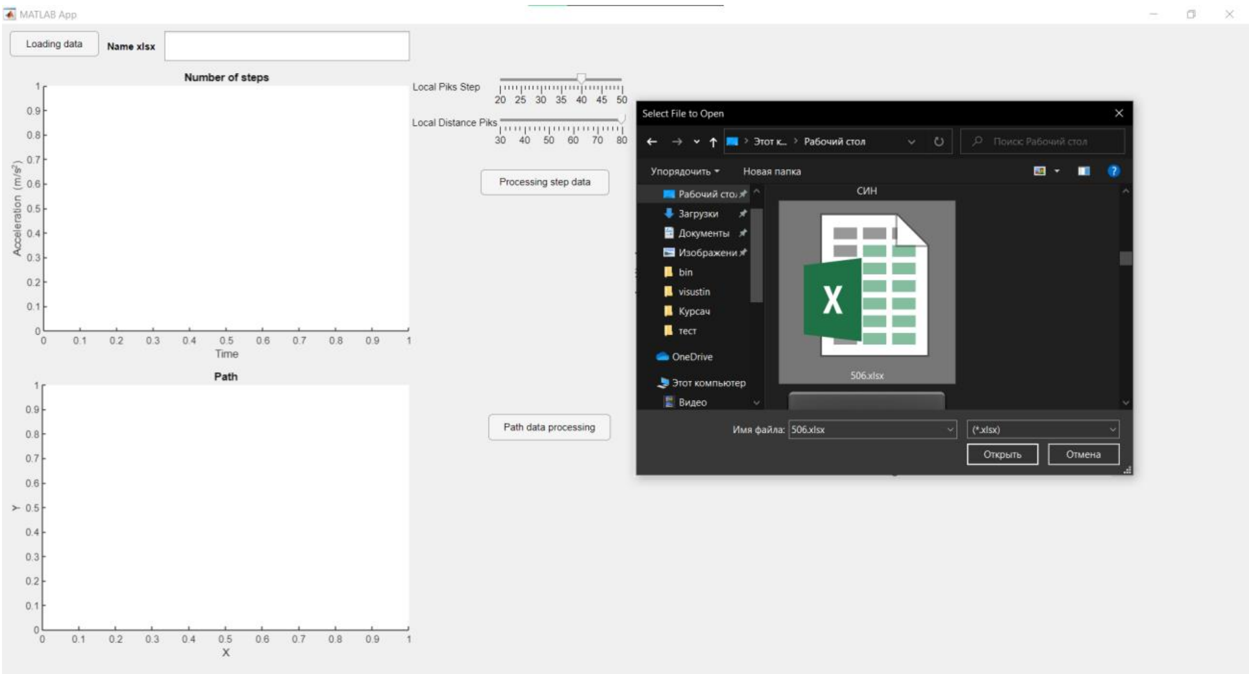


Fig. 3.19. Data loading window

In Fig. 3.20 shows 3 (a), 4-9 points of the algorithm, processing the received data from the accelerometer, calculating the module of the acceleration vector, filtering low peak oscillations acceleration, determining all possible peaks, finding groups of peaks and false peaks, determining the main peaks in groups, separating peaks steps. And data output to the graph.

The block diagram of the part of the program that is responsible for the "Processing step data" button is shown in two figures. Fig. 3.22 (a, b).

Odds settings $A_{Local\ Peaks\ Step}$ and $A_{Local\ Distance\ Peaks}$ occur in manual mode, for this need to move the sliders on the slider.

Click on the "Processing step data" button, after which the processing of steps begins, separately from calculating the path, if the user's ultimate goal is to find out only their number, then the user can stop at this.

In Fig. 3.21 3 (b), 10-13 points of the algorithm are shown, processing the received data from the magnetometer, calculating the direction, calculating the route, layering the route on the global map and comparing the GPS pedestrian path and the pedestrian path.

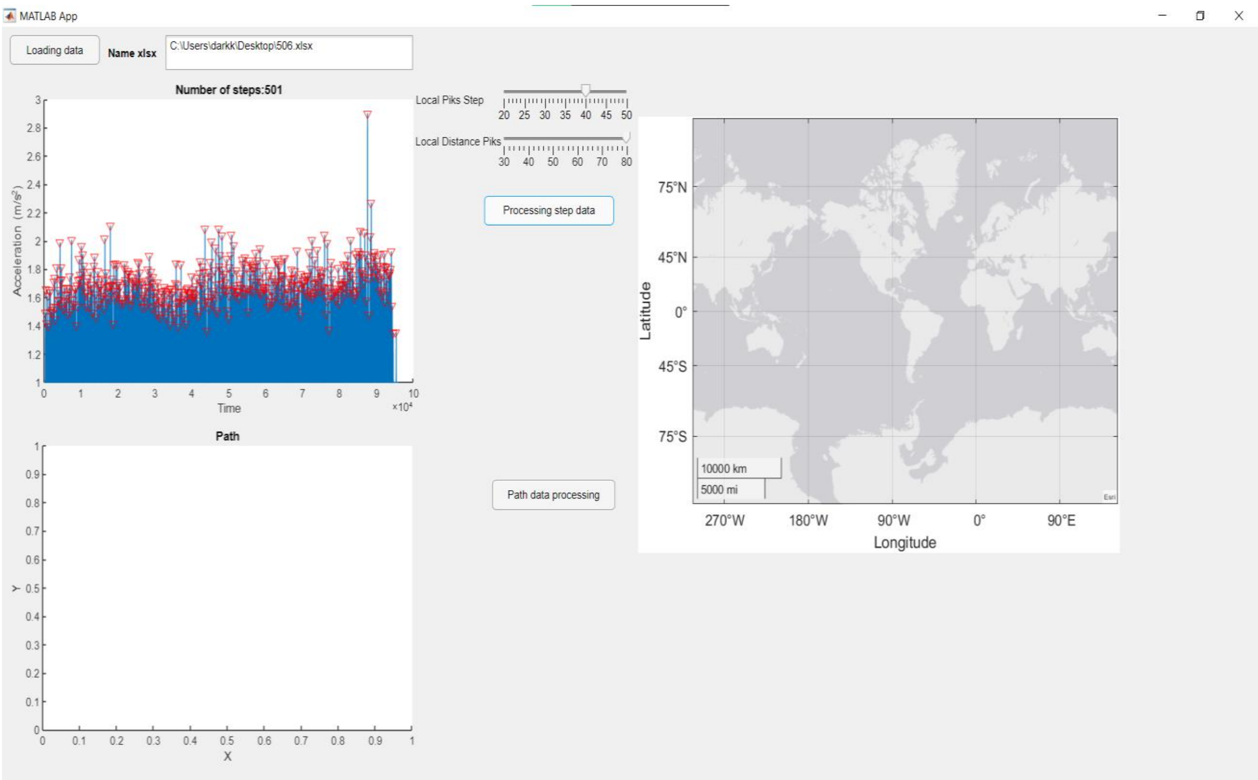


Fig. 3.20. Step data processing window

After calculating these steps, the user has the opportunity to click on the "Path data processing" button to process the data on the route and the distance traveled, and a new window for selecting coordinates appears.

The block diagram of the part of the program that is responsible for the “Path data processing” button is shown in Fig. 3.22 (c).

The user can observe his position relative to the data from the GPS, if any, together with the inertial path, if there is no data from the GPS, then the user can set the initial approximate coordinates himself relative to which the inertial traversed path will be built, or view only the path relative to the point from which he started walking observing the world direction relative to the coordinate axes where the positive X-axis is the direction of the East, and the positive Y-axis is the direction to the North, West and South, respectively, opposite. Using this application allows the user to find out his position in cases when communication with the satellite is not available or is blocked, as well as the ability to navigate in buildings or even caves where communication cannot be.

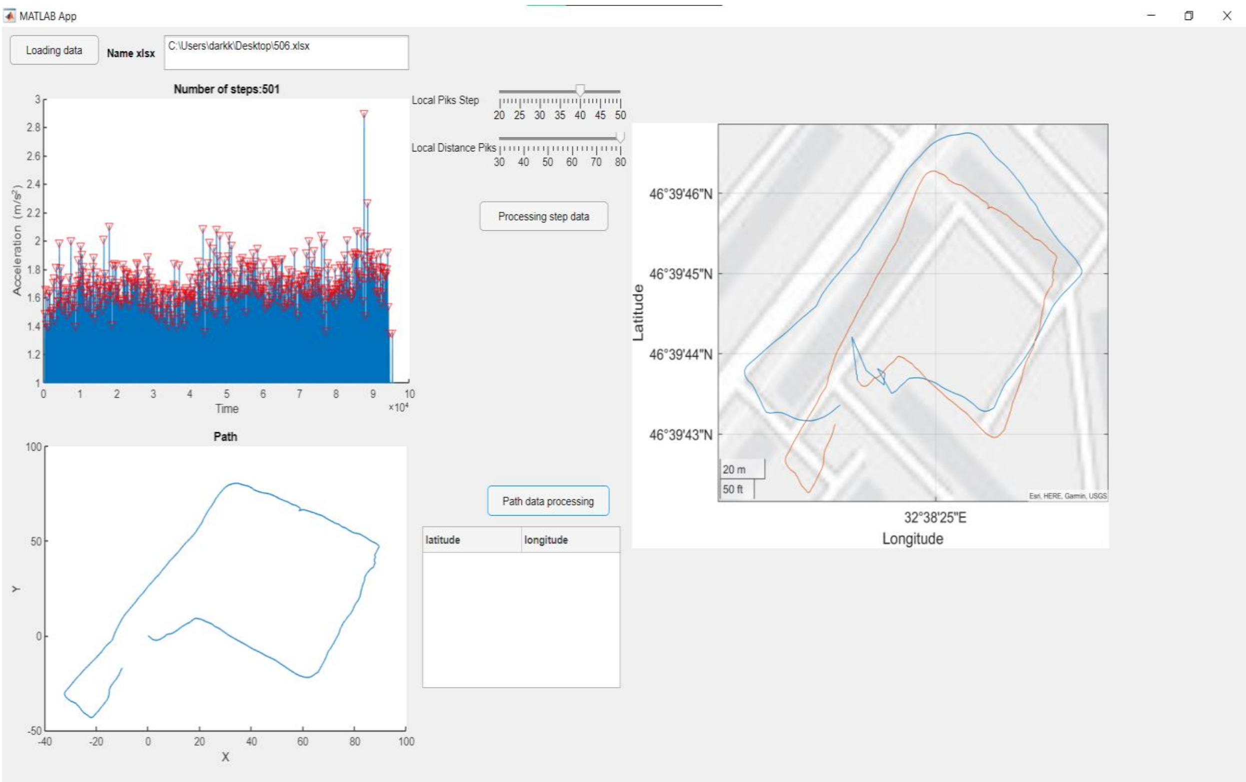


Fig. 3.21. Route processing window

The data processing costs are shown in Table 2. The load on the system is not significant, the time spent by the program depends on the selected data type, as well as the calculation of the global route, the length of the data array itself has almost no effect on performance.

Table 3.2 Elapsed data processing time

№	Number of steps	Amount of data	Elapsed time	Calculating a global route	Expansion file
1	0	4895	2.552359	+	.xlsx
2	13	5146	2.769038	+	.xlsx
3	80	5948	0.408425	-	.xlsx
4	130	9682	0.408125	-	.xlsx
5	150	12462	0.409768	-	.xlsx
6	198	35123	2.803006	+	.xlsx

7	208	38710	2.534432	+	.xlsx
8	506	95921	2.714745	+	.xlsx
9	1053	72803	0.483744	-	.xlsx
10	100	5885	3.456651	+	.csv
11	200	12607	3.055105	+	.csv
12	300	22222	3.980366	+	.csv
13	400	28081	3.046078	+	.csv

In Fig. 3.22 (a, b, c) shows a block diagram of the developed program following algorithms 1-13 with its cycles, branches and filtering. The diagram makes it easier to understand how the program works.

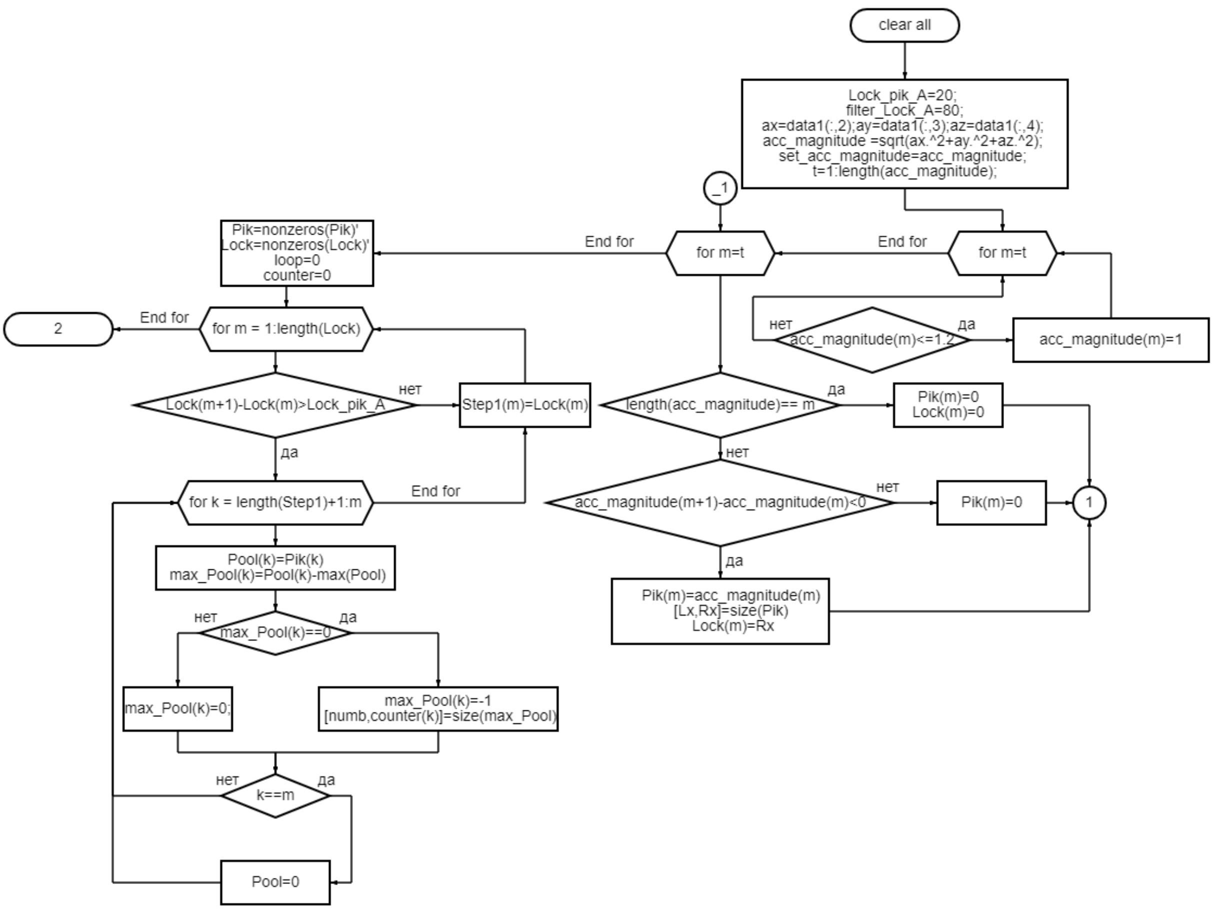


Fig. 3.22. Block diagram of the program (a)

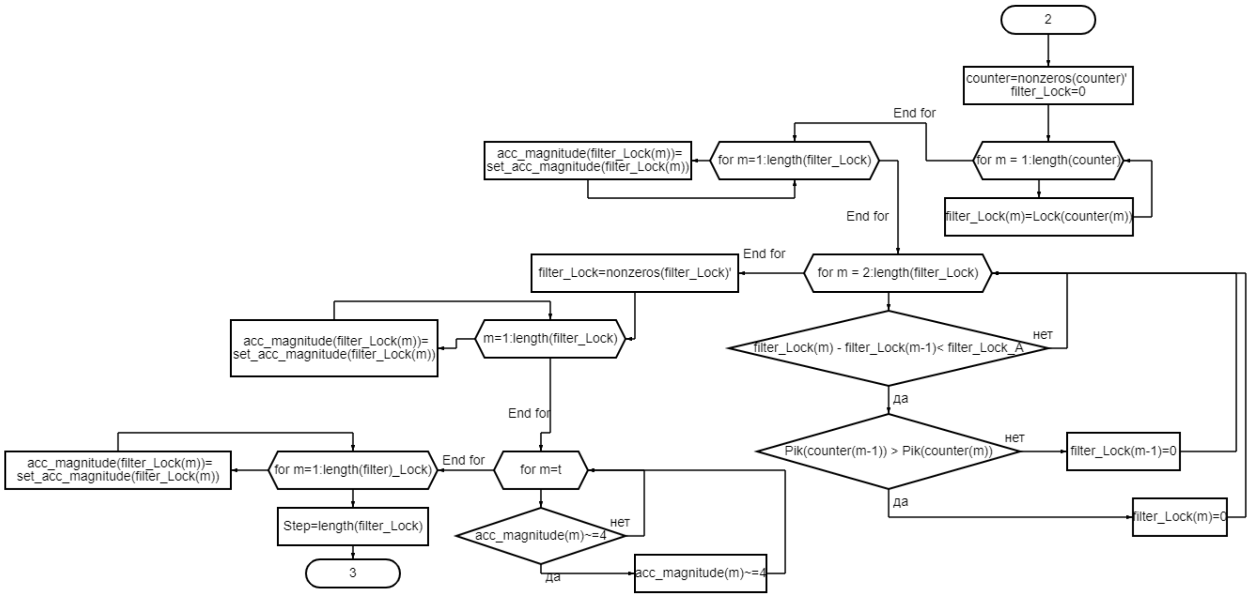


Fig. 3.22. Block diagram of the program (b)

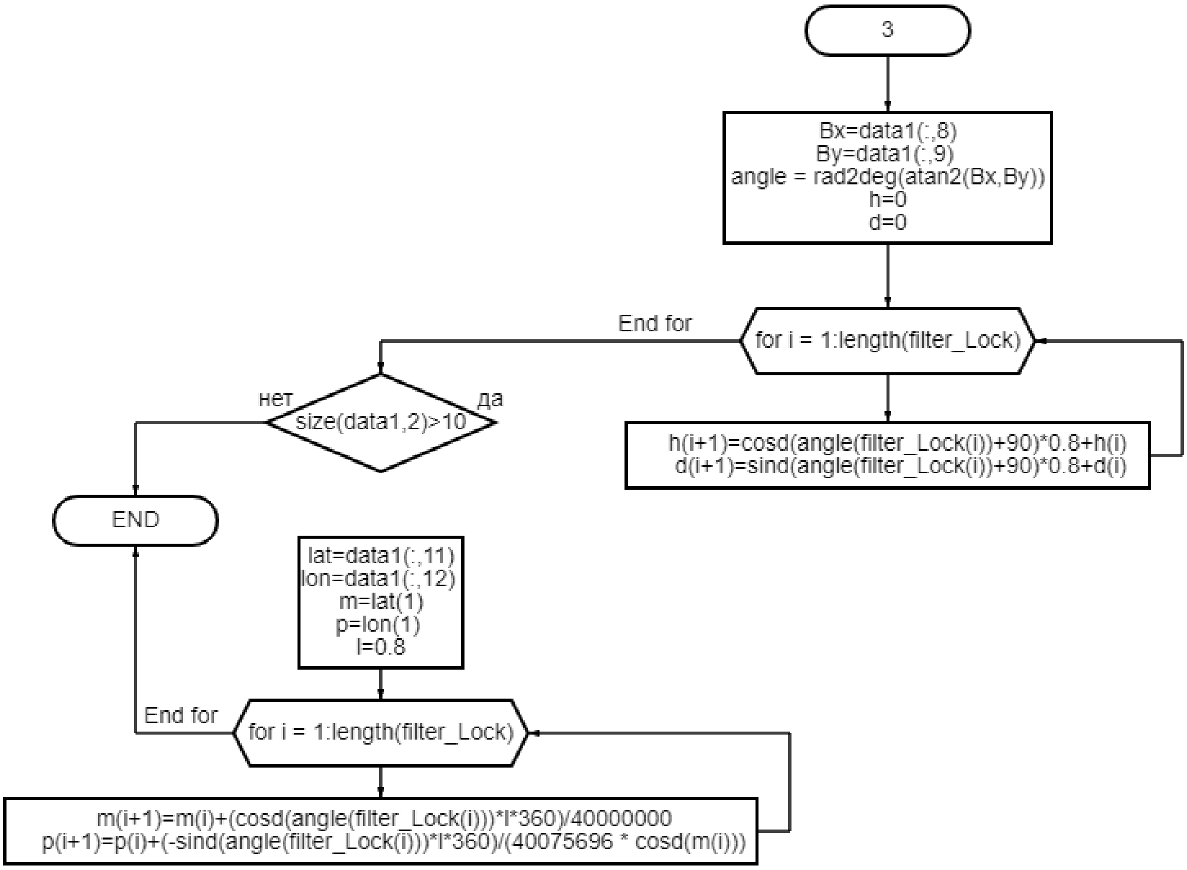


Fig. 3.22. Block diagram of the program (c)

3.5. Conclusions

At the time of writing the diploma, a program was developed with its own structure, as well as a GUI that helps in inertial navigation.

During the development of the software product, such a development environment and tools were used:

1. MATLAB
2. Microsoft Excel
3. Physics Toolbox Sensor Suite

Technologies:

- Object-oriented programming language MATLAB.

Method:

- Finding the peaks
- Calculation of the course based on the magnetometer

The choice of these tools is due to the fact that they have a high productive relationship, which makes it easy to combine into a single system.

The target platform was chosen by MATLAB, how much it is cross-platform, therefore applications developed on its basis can be used on most operating systems.

In the course of the work, the algorithm for finding steps was used, due to its speed, as well as the accuracy of counting steps, which is of great importance for the subsequent construction of the route. The algorithm effectively counts steps even if the phone is used arbitrarily while walking.

As well as an algorithm for calculating the course based on a magnetometer, which complements the system in its speed, as well as the accuracy of measuring the angle of the user's direction, and also allows to hold the device in any convenient position for the user.

CHAPTER 4. DEVELOPMENT TOOLS

4.1. Programs used in development

At the time of writing the project, the following set of techniques, methods, techniques, as well as a set of tools (compilers, application / system libraries, etc.) were used by the developer to create the program code of the Program that meets the specified requirements.

4.1.1. Application package MATLAB

MATLAB - a package of applied programs [11] for solving technical computing problems. The package is used by over a million engineers and scientists, it runs on most modern operating systems, including Linux, macOS, and Windows.

Language programming MATLAB is a high-level interpreted programming language that includes matrix-based data structures, a wide range of functions, an integrated development environment, object-oriented capabilities and interfaces to programs written in other programming languages.

Programs written in MATLAB, there are two types - functions and scripts. Functions have input and output arguments, as well as their own workspace for storing intermediate calculation results and variables. Scripts share a common workspace. Both scripts and functions are saved as text files and compiled to machine code dynamically.

The main feature of the MATLAB language is its extensive capabilities for working with matrices.

MATLAB provides convenient tools for developing algorithms, including high-level ones using object-oriented programming concepts. It contains all the necessary IDE tools, including a debugger and profiler.

For MATLAB, it is possible to create special toolkits that extend its functionality. Toolkits are collections of functions and objects written in MATLAB for solving a specific class of problems.

As part of the MATLAB package Fig. 4.1, there are a large number of functions for building graphs, including three-dimensional, visual data analysis and creating animated videos. The built-in development environment allows to create graphical user interfaces with various controls such as buttons, input fields, and others.

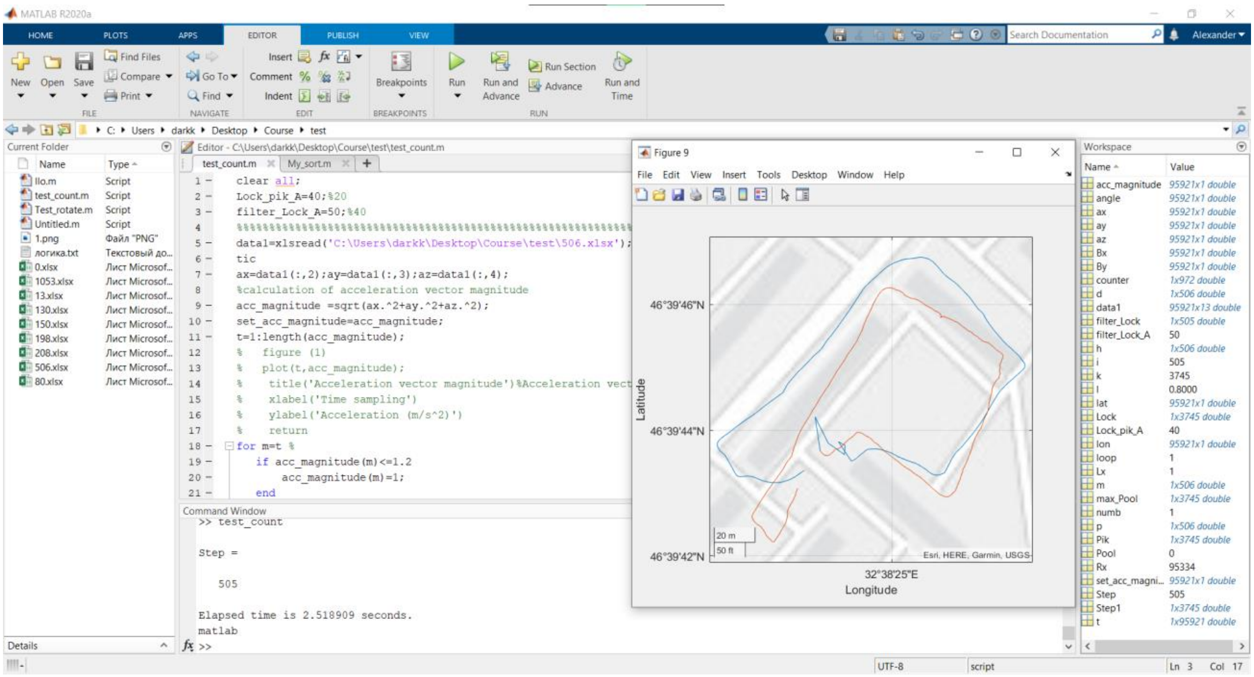


Fig. 4.1. MATLAB window

MATLAB programs, both console and GUI, can be compiled using the MATLAB Compiler module into MATLAB independent executable applications.

4.1.2. Integrated software for creating applications App Designer

App Designer - allows to create professional applications [25] without being a professional software developer. Using visual components to lay out a graphical user interface (GUI) design, and use the built-in editor to quickly program its behavior.

Drag visual components onto the design canvas and use the alignment hints to get an accurate layout. Figure 4.2. The app builder automatically generates object-oriented code that defines the layout and design of the app.

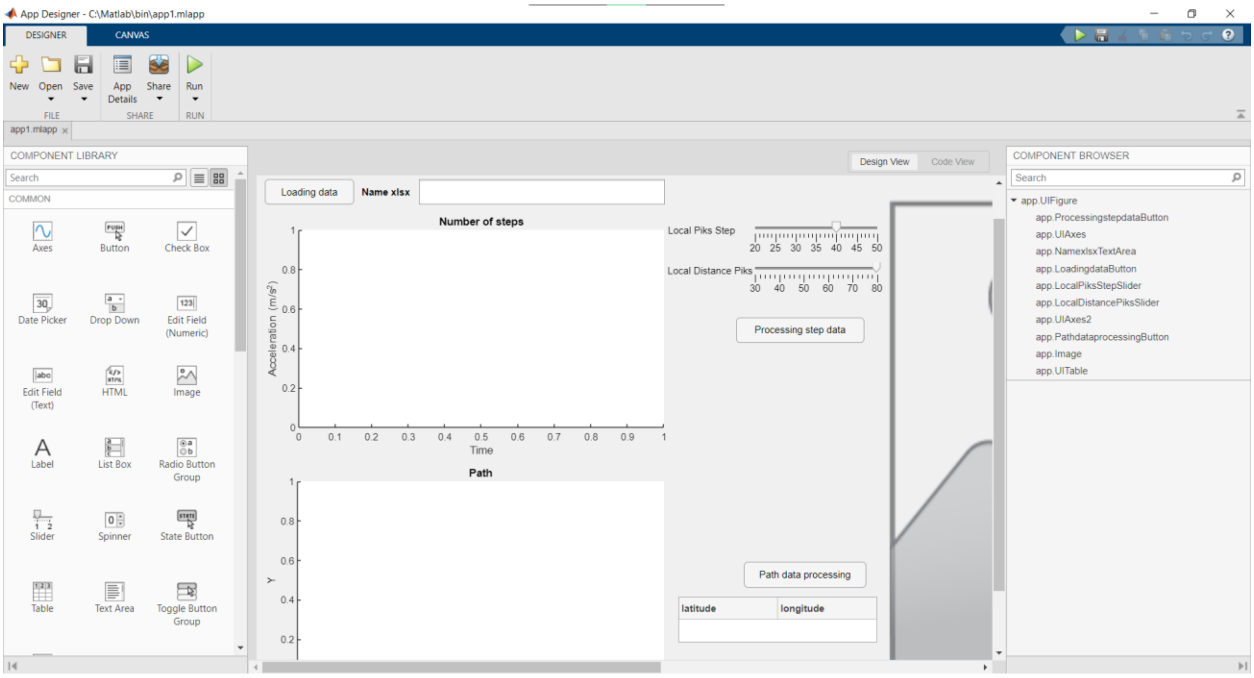


Fig. 4.2. App Designer

Use the integrated version of the MATLAB editor to define the behavior of the application. The app designer can automatically check for coding problems using a code analyzer. Can view warnings and error messages in code as write and modify applications based on those messages.

Create standalone applications with MATLAB Compiler to share with others royalty-free. Can also package applications as interactive web applications and share them using MATLAB Web App Server. End users can run web applications directly from the browser without installing any or additional software.

4.1.3. Application for removing data from the phone Physics Toolbox Sensor Suite (version 2021.04.19)

Physics Toolbox — This application is useful [12] for engineering students, science and industry workers. It uses the sensors of the mobile device to collect, record and export data to a .csv (comma separated value) file that can be shared.

The data can be displayed in the form of a graph with dependence Fig. 4.3, from elapsed time or displayed numerically. Users can export data for further analysis in a spreadsheet or graphing in dedicated software.

Fig. 4.3. Physics Toolbox

In multi recording mode, the user can select one or more sensors to record data at the same time. Files can be easily renamed before exporting or saving to the device, which makes it easy to organize storage and access to data.

Additional features include the ability to record data relative to the current or elapsed time, select a separator in the .csv file (comma or semicolon), change the line width on the graph, change the frequency of data collection from the sensor and control the screen turning on throughout the measurement time.

Sensors used: linear accelerometer - acceleration (x, y, z), gyroscope - radial velocity (x, y, z), magnetometer - magnetic field intensity (x, y, z), compass - magnetic field direction and tilt angle, GPS - latitude, longitude, altitude, speed, direction, number of satellites available to the user.

4.1.4. Program used for database tables Microsoft Excel

Microsoft Excel – is a spreadsheet program [15] created by Microsoft for Microsoft Windows, Windows NT, and Mac OS, as well as Android, iOS, and

Windows Phone. It provides the capabilities of economic and statistical calculations, graphical tools and, with the exception of Excel 2008 for Mac OS X,

macro programming language VBA (Visual Basic for Application). Microsoft Excel is part of Microsoft Office.

4.1.5. Extending used database tables .csv and .xlsx

.csv (Comma-Separated Values) – it is a text format for representing tabular data [13]. A table line corresponds to a text line that contains one or more fields, separated by commas. Fig. 4.4. The CSV format is not fully standardized. The idea of using commas to separate fields is obvious, but this approach creates problems if the original table data contains commas or newlines. A possible solution to the problem of commas and line breaks is to enclose the data in quotation marks, but the original data may contain quotation marks. In addition, the term "CSV" can refer to similar formats that are delimited by a tab (TSV) character or semicolon. Many applications that work with the CSV format allow to select the separator character and the quotation mark.

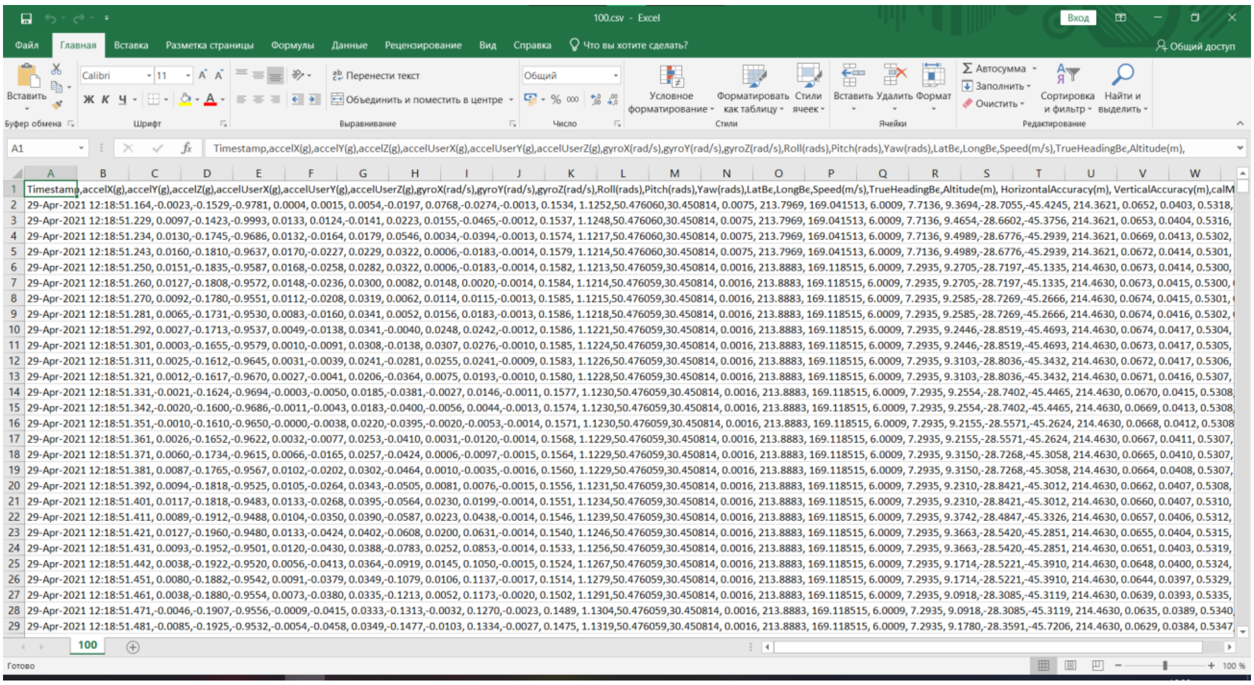
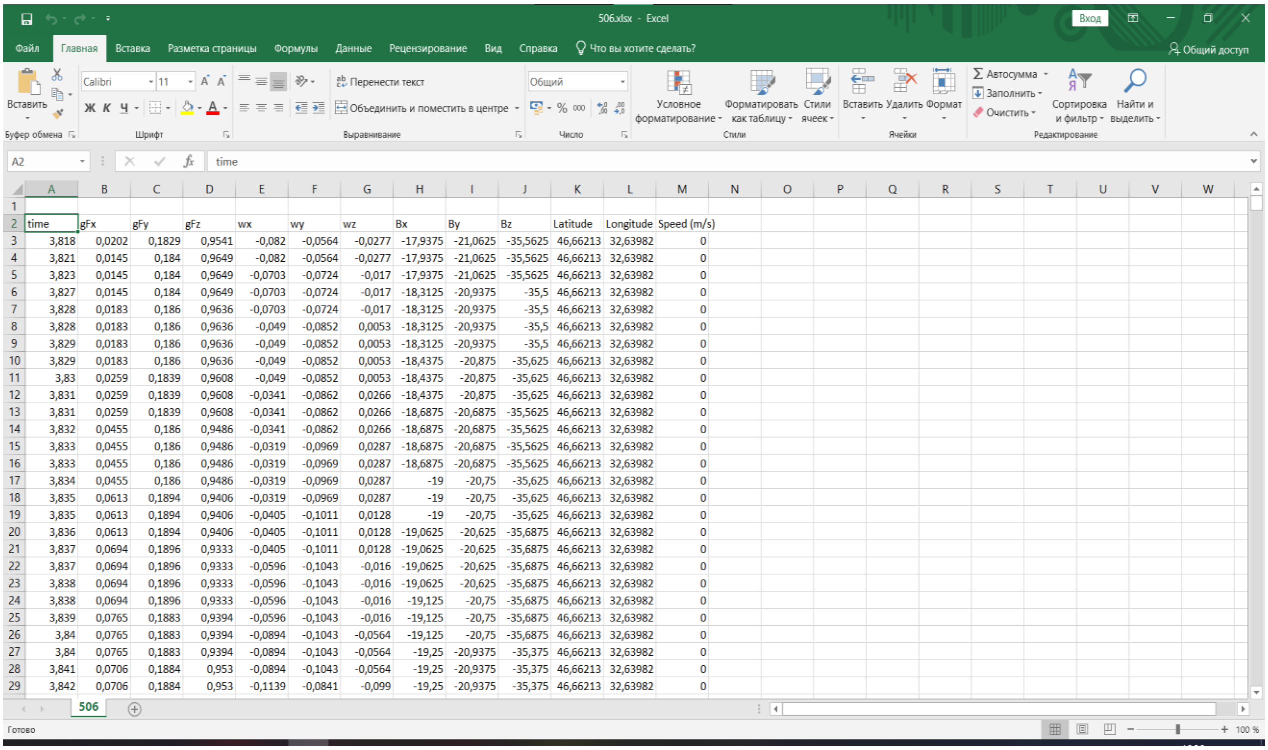


Fig. 4.4. Database tables .csv

The CSV file format is supported by almost all spreadsheets and database management systems, including LibreOffice Calc, Gnumeric, Emacs, Microsoft Excel, Numbers, CSVed, KSpread, Google Docs. CSV import and export is possible in many engineering packages such as ANSYS and LabVIEW.

.xlsx is a series of file formats [14] for storing electronic documents of office suites - in particular, Microsoft Office. The format is a zip archive containing XML text, graphics and other data. 4.5, which were previously stored in binary formats DOC, XLS, etc. The format was originally created as a replacement for the legacy binary document format used by Microsoft Office applications up to and including Office 2003, and a competitor to the standardized OpenDocument right before that. In 2006, the Office Open XML format was declared a free and open format by Ecma International. It is the default format for Microsoft Office 2007 and later applications.



time	gFx	gFy	gFz	wx	wy	wz	Bx	By	Bz	Latitude	Longitude	Speed (m/s)
3,818	0,0202	0,1829	0,9541	-0,082	-0,0564	-0,0277	-17,9375	-21,0625	-35,5625	46,66213	32,63982	0
3,821	0,0145	0,184	0,9649	-0,082	-0,0564	-0,0277	-17,9375	-21,0625	-35,5625	46,66213	32,63982	0
3,823	0,0145	0,184	0,9649	-0,0703	-0,0724	-0,017	-17,9375	-21,0625	-35,5625	46,66213	32,63982	0
3,827	0,0145	0,184	0,9649	-0,0703	-0,0724	-0,017	-18,3125	-20,9375	-35,5	46,66213	32,63982	0
3,828	0,0183	0,186	0,9636	-0,0703	-0,0724	-0,017	-18,3125	-20,9375	-35,5	46,66213	32,63982	0
3,828	0,0183	0,186	0,9636	-0,049	-0,0852	0,0053	-18,3125	-20,9375	-35,5	46,66213	32,63982	0
3,829	0,0183	0,186	0,9636	-0,049	-0,0852	0,0053	-18,3125	-20,9375	-35,5	46,66213	32,63982	0
3,829	0,0183	0,186	0,9636	-0,049	-0,0852	0,0053	-18,4375	-20,875	-35,625	46,66213	32,63982	0
3,83	0,0259	0,1839	0,9608	-0,049	-0,0852	0,0053	-18,4375	-20,875	-35,625	46,66213	32,63982	0
3,831	0,0259	0,1839	0,9608	-0,0341	-0,0862	0,0266	-18,4375	-20,875	-35,625	46,66213	32,63982	0
3,831	0,0259	0,1839	0,9608	-0,0341	-0,0862	0,0266	-18,6875	-20,6875	-35,5625	46,66213	32,63982	0
3,832	0,0455	0,186	0,9486	-0,0341	-0,0862	0,0266	-18,6875	-20,6875	-35,5625	46,66213	32,63982	0
3,833	0,0455	0,186	0,9486	-0,0319	-0,0969	0,0287	-18,6875	-20,6875	-35,5625	46,66213	32,63982	0
3,833	0,0455	0,186	0,9486	-0,0319	-0,0969	0,0287	-18,6875	-20,6875	-35,5625	46,66213	32,63982	0
3,834	0,0455	0,186	0,9486	-0,0319	-0,0969	0,0287	-19	-20,75	-35,625	46,66213	32,63982	0
3,835	0,0613	0,1894	0,9406	-0,0319	-0,0969	0,0287	-19	-20,75	-35,625	46,66213	32,63982	0
3,835	0,0613	0,1894	0,9406	-0,0405	-0,1011	0,0128	-19	-20,75	-35,625	46,66213	32,63982	0
3,836	0,0613	0,1894	0,9406	-0,0405	-0,1011	0,0128	-19,0625	-20,625	-35,6875	46,66213	32,63982	0
3,837	0,0694	0,1896	0,9333	-0,0405	-0,1011	0,0128	-19,0625	-20,625	-35,6875	46,66213	32,63982	0
3,837	0,0694	0,1896	0,9333	-0,0596	-0,1043	-0,016	-19,0625	-20,625	-35,6875	46,66213	32,63982	0
3,838	0,0694	0,1896	0,9333	-0,0596	-0,1043	-0,016	-19,0625	-20,625	-35,6875	46,66213	32,63982	0
3,838	0,0694	0,1896	0,9333	-0,0596	-0,1043	-0,016	-19,125	-20,75	-35,6875	46,66213	32,63982	0
3,839	0,0765	0,1883	0,9394	-0,0596	-0,1043	-0,016	-19,125	-20,75	-35,6875	46,66213	32,63982	0
3,84	0,0765	0,1883	0,9394	-0,0894	-0,1043	-0,0564	-19,125	-20,75	-35,6875	46,66213	32,63982	0
3,84	0,0765	0,1883	0,9394	-0,0894	-0,1043	-0,0564	-19,25	-20,9375	-35,375	46,66213	32,63982	0
3,841	0,0706	0,1884	0,953	-0,0894	-0,1043	-0,0564	-19,25	-20,9375	-35,375	46,66213	32,63982	0
3,842	0,0706	0,1884	0,953	-0,1139	-0,0841	-0,099	-19,25	-20,9375	-35,375	46,66213	32,63982	0

Fig. 4.5. Database tables .xlsx

4.2. The system used in the development

The implemented application for obtaining an inertial route was developed on Windows 10 Pro (Version 10.0.19042 Build 19042), with system specifications, CPU Intel (R) Core (TM) i3-7130U 2.70GHz, Memory 12.0 GB 2133GHz, GPU Intel (R) HD Graphics 620, in the integrated MATLAB App Designer.

For the system to work, there must be an additional library and a configured system on the computer.

4.2.1. Installation and system requirements

For the user to work with the developed software system, the user needs the minimum hardware power (Table 4.1).

Table 4.1 Minimum hardware power

Device	Characteristic
CPU	Pentium III, Pentium IV, Pentium M or AMD Athlon, Athlon XP
GPU	ATI Radeon HD 3450 (256 Mb), MSI NVIDIA GeForce GT 330 768MB
Memory	256Мб
OC	Windows XP/2000/NT/Vista

Supported hardware architectures:

- win-x86
- win/linux/amd64

And the presence of a library:

- mdmcrrt9_8.dll

4.2.2. Scenarios for user interaction with the system

The users of the system are pedestrians in a certain region. The user interface is implemented using MATLAB App Designer.

The first thing the user sees in front of him is the window of the executable program Fig. 4.6, where he has the ability to download the data and process it.

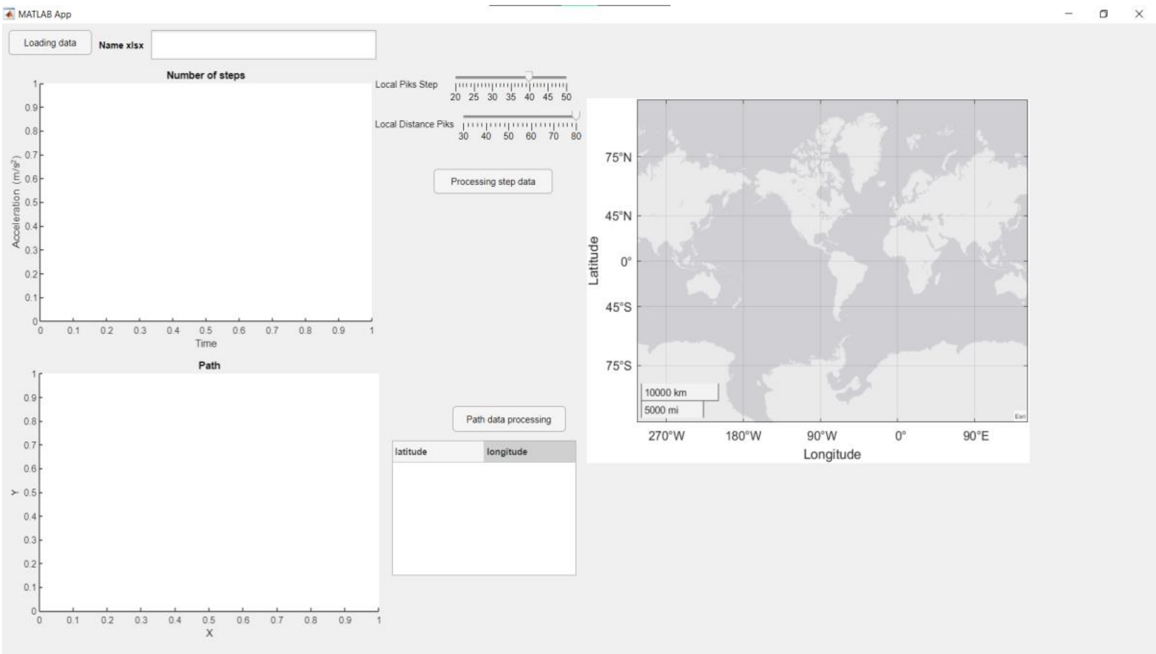


Fig. 4.6. Executable program window

In the following screenshot Fig. 4.7, the user data is loaded into the application (already saved on the device .xlsx or .csv file).

To download data, users need to click the “Loading data” button and select the file in the pop-up window and click “Open”.

In the absence of data or in the wrong format, the program will display an error in the “Name xlsx” window and ask to enter correct data from the user.

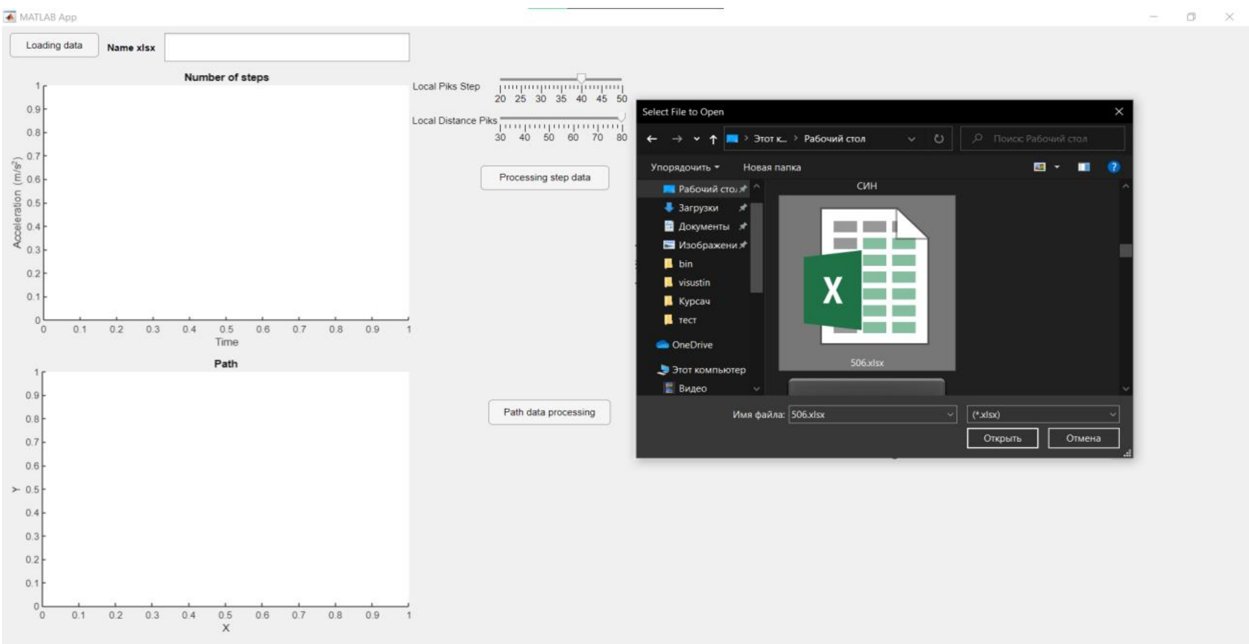


Fig. 4.7. Data loading window

On the next slide Fig. 4.8 shows an example of data processing that the user has entered into the system, must click on the "Processing step data" button and the program will process the data.

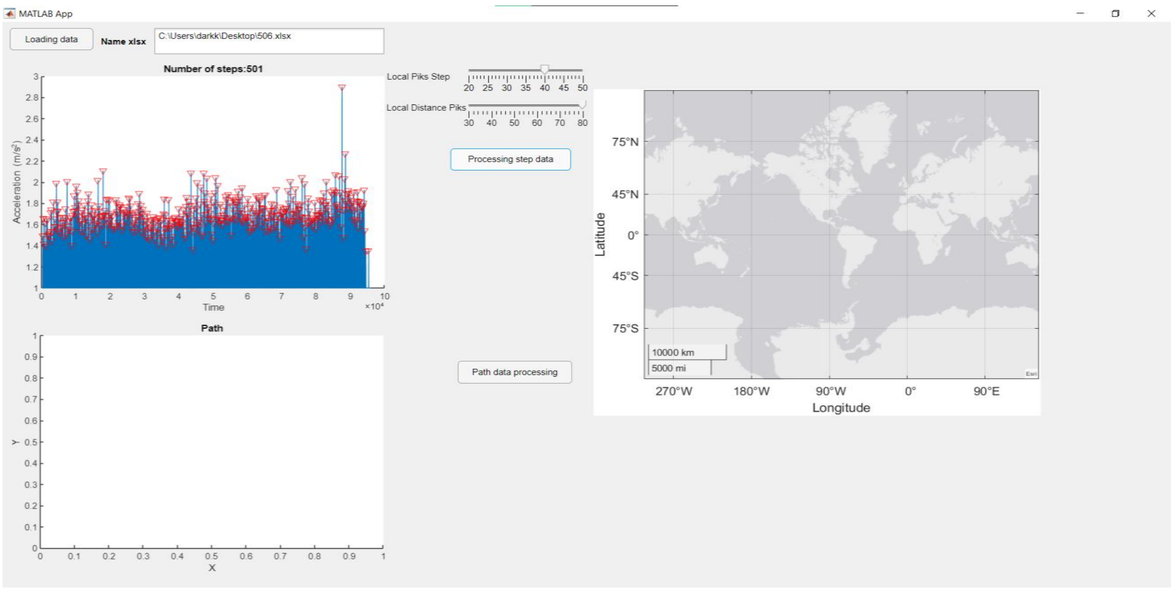


Fig. 4.8. Window for processing these steps

On the last slide Fig. 4.9, the results of counting steps and the path traversed by the user are presented. This is done by the last button in the center of the “Path data processing” screen, which processes the user's data and provides him with the route he has traveled and information about the number of steps.

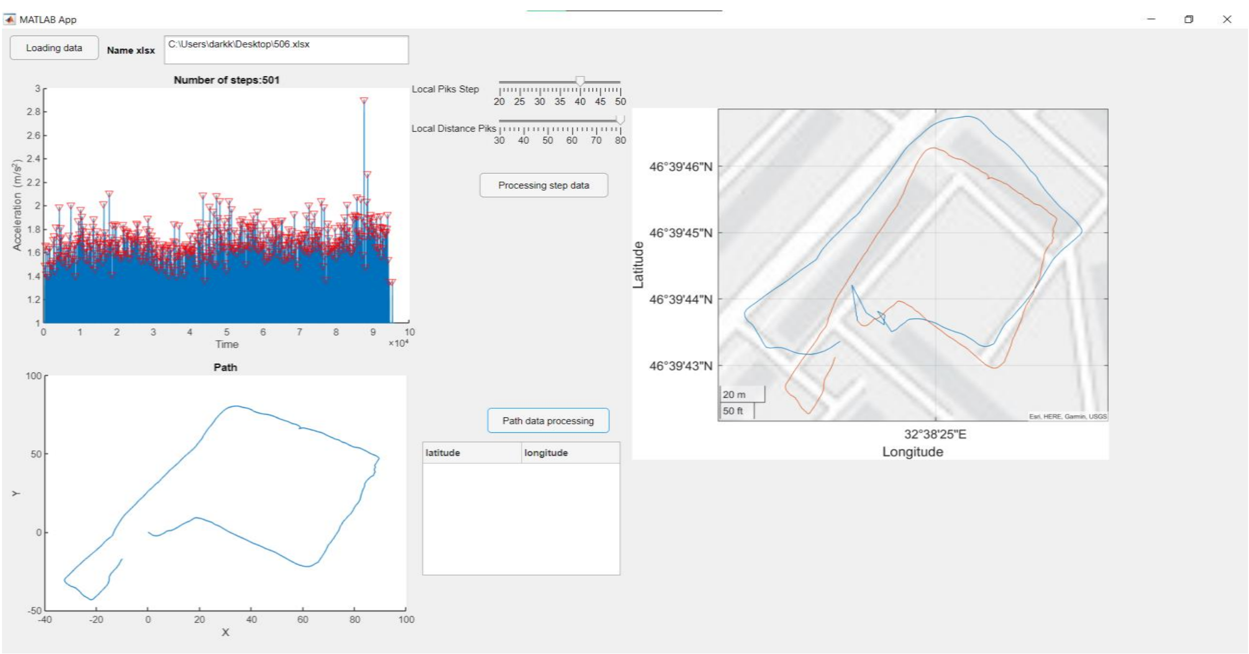


Fig. 4.9. Route processing window

CONCLUSION

The work analyzes the basic algorithms for the pedestrian dead reckoning using inertial sensors located in a smartphone.

The positive and negative sides of various methods for determining a person's step for use in real life are considered. In the process of work, the algorithm for finding steps by peaks, modernized by filters, was used, which increased the accuracy of counting steps and has an important role in building a route.

Methods for finding directions were presented. A heading calculation algorithm based on a magnetometer was chosen, which complements the system in terms of speed and accuracy by measuring the user's heading angle, and allows the device to be held in any position convenient for the user.

In the course of this work, a system was successfully developed that counts a person's steps, and also shows the path he traveled, based on a mathematical model of a person's step and an inertial system developed on the basis of the MATLAB platform. The system consists of an application and an interface based on MATLAB App Designer.

The users of the system are ordinary pedestrians. The input information is the data from the accelerometer and magnetometer received from the user. The initial information is the result of the performed data calculation in the application.

Among the main functions of the software are:

- provides data on the number of steps taken;
- provides the ability to find the traversed route in the global coordinate system.

REFERENCES

1. S. Beauregard and H. Haas, “Pedestrian dead reckoning: A basis for personal positioning,” in Proceedings of the 3rd Workshop on Positioning, Navigation and Communication, March 2006, pp. 27 -35.

2. Матрицы поворота, углы Эйлера и кватернионы (Rotation matrices, Euler angles and quaternions) – Modeling and recognition of 2D/3D images. *Modeling and recognition of 2D/3D images*.

URL: https://api-2d3d-cad.com/euler_angles_quaternions/ (date of access: 10.06.2021).

3. Magnetometer errors calibration.

URL: https://www.vectornav.com/resources/magnetometer-errors-calibration (date of access: 10.06.2021).

4. Renaudin V., Combettes C. (PDF) magnetic, acceleration fields and gyroscope quaternion (magyq)-based attitude estimation with smartphone sensors for indoor pedestrian navigation. *ResearchGate*.

URL: https://www.researchgate.net/publication/269037643_Magnetic_Acceleration_Fields_and_Gyroscope_Quaternion_MAGYQ-Based_Attitude_Estimation_with_Smartphone_Sensors_for_Indoor_Pedestrian_Navigation (date of access: 10.06.2021).

5. URL:https://findpatent.ru/patent/273/2736876.html (date of access: 10.06.2021).

6. Speed-Dependent Modulation of Muscle Activity Based on Muscle Synergies during Treadmill Walking. *Frontiers*.

URL: https://doi.org/10.3389/fnhum.2018.00004 (date of access: 10.06.2021).

7. Contributors to Wikimedia projects. Инерциальная система отсчёта – Википедия. *Википедия – свободная энциклопедия*.

URL: https://ru.wikipedia.org/wiki/Инерциальная_система_отсчёта (date of access: 10.06.2021).

8. Contributors to Wikimedia projects. Географические координаты – Википедия. *Википедия – свободная энциклопедия*.

URL: https://ru.wikipedia.org/wiki/Географические_координаты (date of access: 10.06.2021).

9. A Novel Walking Detection and Step Counting Algorithm Using Unconstrained Smartphones. *PubMed Central (PMC)*.

URL: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5796454/ (date of access: 10.06.2021).

10. Считайте шаги с помощью акселерометра в android - CodeRoad. *Вопросы - CodeRoad*.

URL: https://coderoad.ru/11241208/Считайте-шаги-с-помощью-акселерометра-в-android (date of access: 10.06.2021).

11. Contributors to Wikimedia projects. MATLAB – Википедия. *Википедия – свободная энциклопедия*.

URL: https://ru.wikipedia.org/wiki/MATLAB#:~:text=Язык%20MATLAB%20является%20высокоуровневым%20интерпретируемым,написанным%20на%20других%20языках%20программирования. (date of access: 10.06.2021).

12. Vieyra Software. *vieyra-software*.

URL: https://www.vieyrasoftware.net (date of access: 10.06.2021).

13. Contributors to Wikimedia projects. CSV – Википедия. *Википедия – свободная энциклопедия*.

URL: https://ru.wikipedia.org/wiki/CSV#Спецификация (date of access: 10.06.2021).

14. Contributors to Wikimedia projects. Office Open XML – Википедия. *Википедия – свободная энциклопедия*.

URL: https://ru.wikipedia.org/wiki/Office_Open_XML (date of access: 10.06.2021).

15. Contributors to Wikimedia projects. Microsoft Excel – Википедия. *Википедия – свободная энциклопедия*.

URL: https://ru.wikipedia.org/wiki/Microsoft_Excel (date of access: 10.06.2021).

16. *Computer Science | Academics | WPI*.

URL: http://web.cs.wpi.edu/~emmanuel/courses/cs528/F17/slides/papers/deepak_ganesan_pedometer.pdf (date of access: 10.06.2021).

17. (PDF) Step counting on smartphones using advanced zero-crossing and linear regression / J. Seo et al. *ResearchGate*.

URL: https://www.researchgate.net/publication/282680705_Step_counting_on_smartphones_using_advanced_zero-crossing_and_linear_regression (date of access: 10.06.2021).

18. Palshikar G. (PDF) Simple Algorithms for Peak Detection in Time-Series. *ResearchGate*.

URL: https://www.researchgate.net/publication/228853276_Simple_Algorithms_for_Peak_Detection_in_Time-Series (date of access: 10.06.2021).

19. M.-S. Pan and H.-W. Lin, “A step counting algorithm for smartphone users: Design and implementation,” IEEE Sensors Journal, vol. 15, pp. 2296–2305, 04 2015.

20. Box, G. E. P., G. M. Jenkins, and G. C. Reinsel. *Time Series Analysis: Forecasting and Control*. 3rd ed. Englewood Cliffs, NJ: Prentice Hall, 1994.

21. BMA456. *Bosch Sensortec*.

URL: https://www.bosch-sensortec.com/products/motion-sensors/imus/bmi160/ (date of access: 10.06.2021).

22. https://www.digikey.com/catalog/en/partgroup/ak09918/70323 (date of access: 10.06.2021).

23. Малафеева Д. МЭМС акселерометры, магнитометры и углы ориентации. *Все публикации подряд / Хабр*.

URL: https://habr.com/ru/post/491476/ (date of access: 10.06.2021).

24. Manos A., Klein I., Hazan T. (PDF) Gravity-Based Methods for Heading Computation in Pedestrian Dead Reckoning. *ResearchGate*.

URL: https://www.researchgate.net/publication/331613480_Gravity-Based_Methods_for_Heading_Computation_in_Pedestrian_Dead_Reckoning (date of access: 10.06.2021).

25. MATLAB App Designer. *MathWorks - Makers of MATLAB and Simulink - MATLAB & Simulink*.

URL: https://www.mathworks.com/products/matlab/app-designer. (date of access: 10.06.2021).

APPENDIX

Appendix A. Units of measure the data loading tables

Data that was obtained from the MATLAB environment.

time	gFx	gFy	gFz	wx	wy	wz	Bx	By	Bz	Latitude	Longitude	Speedms
Number	Number	Number	Number	Number	Number	Number	Number	Number	Number	Number	Number	Number
time	gFx	gFy	gFz	wx	wy	wz	Bx	By	Bz	Latitude	Longitude	Speed (m/s)
3.8180	0.0202	0.1829	0.9541	-0.0820	-0.0564	-0.0277	-17.9375	-21.0625	-35.5625	46.6621	32.6398	0
3.8210	0.0145	0.1840	0.9649	-0.0820	-0.0564	-0.0277	-17.9375	-21.0625	-35.5625	46.6621	32.6398	0
3.8230	0.0145	0.1840	0.9649	-0.0703	-0.0724	-0.0170	-17.9375	-21.0625	-35.5625	46.6621	32.6398	0
3.8270	0.0145	0.1840	0.9649	-0.0703	-0.0724	-0.0170	-18.3125	-20.9375	-35.5000	46.6621	32.6398	0
3.8280	0.0183	0.1860	0.9636	-0.0703	-0.0724	-0.0170	-18.3125	-20.9375	-35.5000	46.6621	32.6398	0
3.8280	0.0183	0.1860	0.9636	-0.0490	-0.0852	0.0053	-18.3125	-20.9375	-35.5000	46.6621	32.6398	0
3.8290	0.0183	0.1860	0.9636	-0.0490	-0.0852	0.0053	-18.3125	-20.9375	-35.5000	46.6621	32.6398	0

Fig. A.1. Loading data

Where column time - time in seconds [s];

gFx, gFy, gFz - is the acceleration of an object in three-dimensional space along 3 axes x, y, z is measured in [$\frac{m}{s^2}$];

wx, wy, wz - is the angular velocity of an object in three-dimensional space along 3 axes x, y, z, measured in [$\frac{rad}{s}$];

Bx, By, Bz - this is the magnetic field of an object in three-dimensional space along 3 axes x, y, z is measured in [μT];

Latitude and Longitude - it is a coordinate system associated with positions on Earth and is measured in [degree °];

Speedms – is the speed of the object in measured in [$\frac{m}{s}$].