*I.J. Intelligent Systems and Applications,* **2020, 4, 1-12**
Published Online August 2020 in MECS (http://www.mecs-press.org/)
DOI: 10.5815/ijisa.2020.04.01

# An Algorithm for Detecting the Minimal Sample Frequency for Tracking a Preset Motion Scenario

**Dmytro V. Fedasyuk**
Lviv Polytechnic National University/Software Department, Lviv, 79013, Ukraine
E-mail: fedasyuk@gmail.com

**Tetyana A. Marusenkova**
Lviv Polytechnic National University/Software Department, Lviv, 79013, Ukraine
E-mail: tetyana.marus@gmail.com

Received: 26 November 2019; Accepted: 26 January 2020; Published: 08 August 2020

**Abstract:** Inertial sensors are used for human motion capture in a wide range of applications. Some kinds of human motion can be tracked by inertial sensors incorporated in smartphones or smartwatches. However, the latter can scarcely be used if misclassification of user activities is highly undesirable. In this case electronics and embedded software engineers should design, implement and verify their own human motion capture embedded systems, and oftentimes they have to do so from scratch. One of the issues the engineers should face is selection of suitable components, primarily accelerometers, gyroscopes and magnetometers, after thorough examination of commercially available items. Among technical characteristics of inertial sensors their sample frequency determines whether the sensor will be able to capture a specific motion kind or not. We propose a novel algorithm that allows the researcher or embedded software engineer to calculate the minimal sample frequency sufficient for tracking a prescribed motion scenario without significant signal losses. The algorithm utilizes the Poisson equation for motion of a triaxial rigid body, the Shoemake's algorithm for interpolating quaternions on the unit hypersphere, and the frequency analysis of a discrete-time signal. One can use the proposed algorithm as an argument for acceptance or rejection of a gyroscope when selecting hardware components for a human motion tracking system.

**Index Terms:** Sample frequency, motion scenario, MEMS gyroscope, MEMS inertial sensor, Shoemake's algorithm.

## 1. Introduction

Modern inertial sensors manufactured using MEMS (Micro-Electro-Mechanical Systems) technology have numerous advantages, including small size, light weight and low power consumption, which makes them wearable. Moreover, such sensors are self-containing, i.e., one does not need any external tools for tracking the position and attitude of an object supplied with such sensors, both indoors and outdoors. The latter property coupled with the possibility to embed MEMS inertial sensors into clothes and footwear gives rise to a variety of applications, primarily in human motion capture systems. Constant tracking motions and position of aging or convalescent people in their natural living environment helps understand the cause of many diseases and thus provides valuable information for medical diagnosis and/or feedback on the treatment efficacy. Among the common neurological problems that affect human motor functions are Parkinson disease, multiple sclerosis, stroke, traumatic brain injury, and spinal cord injuries [1]. If a patient is supposed to perform exercises on their own, a human motion capture system can replace surveillance from medical personnel. Fall detection systems are another popular research subject, since they can mitigate the fall consequences by reporting a fall straightaway after it occurred and calling for immediate medical help. Inertial sensors are helpful in analysis of sport activities for performance assessment and injury prevention [2]. Another vast application area relates to classification of an employee's movements as correct or incorrect. A lot of recent research results have been published on gait analysis and classification of human activities. Besides, human motion capture is used in human-machine interaction, user authentication [3], robotics and telemedicine [4]. The appearance of MEMS inertial sensors became a decent means to overcome the inherent limitations of other motion capture systems – optical, structured light, acoustic, magnetic and mechanical ones. Commercial optical systems such as Vicon or Optotrack are considered to be the gold standard in human motion capture due to their high accuracy. The benefits of inertial sensors are corroborated by the variety of commercially available IMUs and IMU-based systems manufactured by Invensense and Microstrain (the USA), Trivisio (Germany), and XSens (The Netherlands) [5].

 

However, MEMS inertial sensors have their limitations [6]. Most researchers focus on filtering algorithms to process noisy measurements. However, the performance of any filtering algorithm is highly dependent on the discretization time, especially for non-linear filtering problems. That is why the selection of sensors with a suitable sample frequency is of key importance, since it will be extremely difficult to extract meaningful information out of seldom and faulty sensor readings. Surprisingly, the problem is not paid the attention it actually deserves. Our aim is to develop an algorithm for calculation of the minimal sample frequency suitable for tracking a specific rotational motion. The algorithm would help the embedded software engineer choose the right gyroscope when designing a motion tracking system from scratch.

## 2. Related Work

A lot has been published on tracking human motion. One can distinguish several main research directions. A large research area concerns usage of smartphones for recognition of their owners' basic daily activities. Modern smartphones are supplied with various sensors such as 3D accelerometers and magnetometers, compasses, proximity sensors, GPS, digital cameras and microphones. Such equipment coupled with the total ubiquity of smartphones make the latter useful for scientific and clinical research, for instance in healthcare and physical activity monitoring. Ref. [7] studied the applicability of smartphones for clinical motion research. The authors showed that smartphones could be used to assess range of motion and joint angle measurement for postural and gait control. They benchmarked performance of different smartphone sensors against an Xsens product. Ref. [8] proposed a human fall monitoring system comprised by a portable sensor unit including a 3D accelerometer, a 3D gyroscope and a 3D magnetometer and a cell phone. However, with cell phones, researchers face the following problems. Firstly, movement of a phone depends on where it is placed on the owner's body. Secondly, people carry their cell phones in different orientations, thus the gravity affects accelerometer readings differently. Thirdly, the variety of hardware models and operating systems cause challenges to activity recognition. Finally, energy-efficiency is always an issue with cell phones. Hence, the main research topics in this field are light methods for user-, orientation- and hardware-independent activity recognition.

Smartphones are not an option in industrial context, where occasional misclassifications of human activities are intolerable. Thus a plenty of specially designed devices appeared. In contrast to smartphones, such devices can afford more complicated recognition algorithms because they are not that restricted in battery consumption and computational complexity. One of the main research topics in this case is the required amount and location of sensors and suitable recognition methods depending on complexity, periodicity, and dynamicity of activities to be analyzed. In [9] a human body motion tracking system was introduced. Ref. [10] proposed a small-sized cheap-to-manufacture measurement system suitable for spatial orientation of the foot, detection of gait cycle phases, assessment of motor activity, etc. The system implements algorithms for determination of the orientation of objects within three-dimensional space using an integrated triaxial MEMS system, magnetometer, and Madgwick's AHRS sensor fusion algorithm. Ref. [11] evaluated the accuracy of postural human motion tracking based on miniature inertial sensors. Ref. [1] demonstrated that wearable inertial sensors can be used to develop metrics for objective measurements of tremor and dyskinesia in individuals suffering from Parkinson's disease, whereas the authors of [12] developed an inertial ambulatory monitoring system that provides a complete motor assessment of tremor, bradykinesia, and hypokinesia. Ref [13] presented a novel interactive method for recognizing handwritten words using inertial sensors included into smartwatches.

Ref. [14] introduced three strategies to measure motions of classical cross-country skiing, ski mountaineering, alpine ski racing and outdoor walking over long distances (several kilometers). These strategies found an articulate implementation in a system for alpine ski racing, which is highly dynamic, i.e., characterized by fast direction changes, high speeds and the absence of static or slow phases. The proposed methods rely only on inertial sensors and magnetometers. Nevertheless, they provide position, attitude and speed information with an accuracy close to the "gold standards". For each activity specific biomechanical constraints and movement dynamics were exploited. The authors emphasize the role of the sample frequency. They pinpoint two error types for signal sampling: signal losses caused by an insufficiently high sampling frequency and inadequate low-pass filters at analogue-to-digital conversion. Signal losses are especially likely to occur for rapidly changing movements such as foot movements during gait. The IMU's sampling frequency used in [14] was 500 Hz, which the authors found to be sufficient for their gyroscopes to measure all movements accurately. Ref. [15] discussed fusion of GNSS with data from inertial and magnetic sensors in order to analyze performance in alpine ski racing. The authors sampled magnetometer data at 125 Hz and data from 3D accelerometers and gyroscopes at 500 Hz. The same sample frequencies were used in the experimental setup of [16]. In their experiments two inertial sensors were attached to the right shank and thigh of a skier. Ref. [17], among other valuable dissertation results, discusses the influence of accelerometer sample frequency on the recognition rate when classifying different swimming styles (two different sensor locations were considered). Besides, in order to reduce battery consumption, the researcher exploits the characteristics of long-term activities: if one was cycling for a while, it is most likely that the same person will be still cycling at the next moment. In his thesis [17] Pekka Siirtola took an original 50 Hz, formed 5 Hz, 10 Hz and 25 Hz signals by picking each $10^{th}$, $5^{th}$ or $2^{nd}$ point correspondingly and recorded the recognition rate of three periodic activities – freestyle, breaststroke and backstroke swimming – for each

 

sample frequency. He revealed that the recognition rate did not suffer much from the sample frequency reduction. The role of the underlying hardware is proven by the results obtained by the same author. The recognition rate of five typical human activities (walking, running, cycling, driving a car and idling) turned out to be slightly higher for a Symbian smartphone than for an Android smartphone.

According to [18], a sample frequency 100 Hz is enough for capturing active human motion in sufficient detail. Ref. [19] presented an implementation of a system for 3D attitude measurement and estimation using a magnetic and inertial measurement unit and a Kalman-filter based sensor fusion algorithm. They stated the used sample frequency 200 Hz. The researchers benchmarked their results against traditional optical motion capture systems. Ref. [20] investigated the impact of the amount, location and type of inertial sensors on the accuracy of activity and posture recognition. The authors used Xsens-MTx sensors, each equipped with a 3-axis accelerometer, 3D gyroscope and 3D magnetometer, with the sampling frequency 6 Hz. According to [21] and [22] sampling frequencies exceeding 50 Hz should be used. However, in [23] experiments at the frequency of 2 Hz proved to be successful – eleven activities were distinguished with the average recognition rate of 85%.

Ref. [24] studied the influence of different sampling rates on the recognition accuracy of ten daily activities. The research results prove that the reduction of sampling frequency from 100 Hz to 5 Hz affects different activities in a different way. The worst effect was observed for a walking activity. Ref. [25] introduced a system that utilizes a wearable device and an effective quaternion algorithm for timely fall detection. The system distinguishes a fall from normal everyday activities and alarms the caregivers. In accordance with [25] 100 Hz is a proper sample frequency for human fall detection. However, [26] sampled signals of inertial sensors at 10 Hz. The reason was that the authors made their fall detection system upon a combination of inertial sensors and a location system Ubisense. The latter has a limited sample rate and the authors preferred to synchronize all the readings. Wireless IMUs were used in [27] in combination with the Kinect sensor whose sampling frequency is too low for capturing fast movements (25–35 Hz) and cannot be controlled by the user. Such a combination was used for skeleton tracking. Ref. [28] studied fall detection using a smartwatch. Accelerometer data were obtained at 50 Hz.

Ref. [29] deals with a choice of a MEMS accelerometer suitable for a specific application. However, the author does not take into account a motion scenario to be captured. Ref. [30] studied the applicability of some IMUs for tracking specific motion kinds. The issues stem from the fact that the static and dynamic accuracy of an IMU dictate whether or not the IMU is suitable for a specific application, and the manufacturer of an inertial sensor often does not specify for which motions their dynamic accuracy specification is valid.

Ref. [31] is one of the fundamental works related to the subject of inertial sensors. The authors mentioned that high sample frequencies are required for capturing high-frequency signals but did not investigated the issue on their own.

Literature study shows that most researchers focus mainly on mathematical processing of inertial sensor data. Primarily they modify, adapt, adjust and apply data fusion algorithms (Kalman, extended Kalman, unscented Kalman, particle filters, Madgwick filter and others) or activity classification algorithms (linear and quadratic discrimination analysis, naïve Bayes, J48, Random forest, etc). Relatively few research works consider the characteristics of the underlying inertial sensors including their sample frequencies. Such works analyze the effect of the sample frequency on the motion capture (for instance, on human activity recognition rate). However, they do not try to find out which sample frequency would be enough for successful motion capture and do not establish dependencies between sample frequencies and expected motion trajectories. Moreover, our literature study shows that there are significant discrepancies in the opinions of different authors on suitable sample frequencies, even for similar applications. All the researchers admit the significance of the sample frequency. Many of them study its effect on the performance of the algorithms they use to process raw sensor data. However, to the best knowledge of the authors, there is a lack of studies aimed at the opposite task – calculation of the sample frequency based on the expected motion trajectories.

Nevertheless, an insufficient sampling frequency means signal losses that can scarcely be compensated for by pure mathematical approaches. Our work is aimed at developing an algorithm for discovering the minimal gyroscope sample frequency sufficient for tracking an expected motion scenario.

The work lies within complex research into development of MIS-IDE (Measurement Inertial System Integrated Development Environment) [32] that has been conducted by a team of scholars (including the authors) within a series of commercial projects, which somewhat restrict disclosure of details. MIS-IDE is intended for enhancing the efficacy of synthesizing firmware of navigation systems based on IMU sensors. The scope of the complex work covers developing: models of simulating signals from IMU sensors in accordance with some preset motion scenario, hardware-software tools for testing IMU sensors; algorithms for parametric analysis and synthesis of IMU sensor models, mathematical models for synthesis of noise and measurement errors, algorithms for selection of the most suitable measurement modes, firmware for microprocessors inside embedded systems based on IMU sensors.

## 3. Mathematical Background

To achieve our goal, we propose to combine the following mathematical apparatus: 1) formal definition of angular velocity, since it is what a gyroscope measures; 2) algorithms for description of rotational motion; and 3) ways of frequency analysis of a signal.

Any motion of a rigid body can be represented as a combination of translational and rotational motions, each of them considered independently. The former can be characterized by linear accelerations provided by an accelerometer whereas the latter – by measurements from a gyroscope, i.e. by angular velocities.

*3.1. Angular Velocity*

A modern MEMS gyroscope measures the angular rate vector in the body coordinate frame. As is well known, the angular rate vector in the inertial coordinate frame is defined as

$$\omega = \lim_{\Delta t \to 0} \frac{\Delta\varphi(t, \Delta t)}{\Delta t} e(t, \Delta t) \tag{1}$$

where $e(t, \Delta t)$ is the rotation axis and $\Delta\varphi(t, \Delta t)$ is the rotation angle. A movement from position $E(t)$ to position $E(t+\Delta t)$ corresponds to quaternion

$$\delta\Lambda = \cos\frac{\Delta\varphi(t, \Delta t)}{2} + e(t, \Delta t)\sin\frac{\Delta\varphi(t, \Delta t)}{2} \tag{2}$$

Taking into account the formula

$$\Delta\Lambda = \Lambda(t + \Delta t) - \Lambda(t) = \delta\Lambda \circ \Lambda(t) - \Lambda(t) = (\Delta\Lambda - 1) \circ \Lambda(t) \tag{3}$$

one obtains

$$\dot{\Lambda} = \lim_{\Delta t \to 0} \frac{\Delta\Lambda}{\Delta t} = \lim_{\Delta t \to 0} \frac{\delta\Lambda - 1}{\Delta t} \circ \Lambda(t) = \frac{\omega \circ \Lambda(t)}{2} \tag{4}$$

Equation (4) is called the Poisson equation.

The classic formula for the angular rate vector expressed using quaternions looks like

$$\omega = 2\dot{\Lambda} \circ \bar{\Lambda} \tag{5}$$

However, when appropriate, one can replace a quaternion derivative by the difference between two quaternions describing the body attitude in two points divided by $\Delta t$ provided that $\Delta t \to 0$, where $\Delta t$ is the time elapsed during moving between the two points.

Equations (4) and (5) can only be used if all the quaternions are expressed in the same coordinate frame.

Oftentimes, sensors measure the projections of the angular velocity vector onto the axes of a coordinate system assigned with the body, i.e., the angular velocity is expressed in the body frame. Since the angular velocity and quaternions in (5) are given in different frames, before using (5) one should perform coordinate system transformation first:

$$\omega = \sum_{k=1}^{3} \omega_k^* e_k = \Lambda \circ \left( \sum_{k=1}^{3} \omega_k^* i_k \right) \circ \bar{\Lambda} = \Lambda \circ \omega^* \circ \bar{\Lambda} \tag{6}$$

where $\omega^*$ is a vector in the inertial frame whose components are the angular velocity projections onto the axes of the body frame assigned to a gyroscope.

*3.2. Motion Trajectory*

Since linear accelerations are measured by accelerometers and angular velocities are measured by gyroscopes, we consider translational and rotational movements independently. I.e., we evaluate the applicability of a specific gyroscope for tracking rotational movements only. Therefore, in the case of a gyroscope, we are interested in curves on a sphere, since one point of the rigid body is fixed and all the other particles in it rotate about this fixed point. The body holds its shape during motion.

In some rare occasions, rotational movements can be defined by an analytical function. However, it is typically not the case when it comes to human motion, which is pretty unpredictable.

One is most likely interested in a way to construct a smooth curve, preferably differentiable, upon only a few points (called key points). A well-known way of approximating smooth motion between two points on a spherical surface is SLERP – Spherical Linear Interpolation also known as great arc in-betweening. It is supposed that two

 

quaternions, $q_1$ and $q_2$, describe the initial and final positions on a hypersphere. To find in-between quaternions using the SLERP, one applies the following formula:

$$Slerp(q_1, q_2; t) = \frac{\sin(1-t)\Theta}{\sin\Theta} q_1 + \frac{\sin(t\Theta)}{\sin\Theta} q_2 \tag{7}$$

where $\Theta$ is the angle between quaternions $q_1$ and $q_2$. The angle between two quaternions is calculated as the arccosine of their scalar product. It means that the quaternions should be normalized in order to ensure that their dot product does not exceed 1. The more values of $t$ are considered, the more in-between quaternions will be obtained.

An equivalent SLERP formula is

$$Slerp(q_1, q_2; t) = q_1(q_1^{-1}, q_2)^t \tag{8}$$

Here the conjugates and the inverses are the same since we deal with unit quaternions.

The SLERP generates the shortest path on the hypersphere between $q_1$ and $q_2$. Being a perfect interpolation method for a pair of points, the SLERP is not that good when more than two key points are involved. It is possible to apply the SLERP to several points in a pair-wise fashion. However, the obtained curve loses its smoothness at the key points. Each SLERP curve taken separately is smooth, but several curves do not join together seamlessly. In a key point the curve seems to change its direction and angular velocity abruptly. In the case of multiple control points, the Shoemake's algorithm [33] generates a curve running through all the points without any abrupt changes.

The Shoemake's algorithm makes use of "spherical Bezier curves", i.e. adopts Bezier's approach to construction of smooth parametrized curves. The idea is to connect a pair of two consecutive control points by a Bezier curve imitation instead of an arc segment. All the curves should be pieced together seamlessly by lining up the segments of each curve at its ends.

Shoemake proposed a reasonable compromise between two conflicting arc segments provided by SLERP, by "aiming for a point halfway between where the incoming segment would proceed, and where the outgoing segment must arrive".

Let's suppose we have three quaternions, $q_{n-1}$, $q_n$ and $q_{n+1}$ describing three consecutive key points.

Many Bezier curves can be built between a pair of points, depending on which control points we have chosen for the curve. The Shoemake's algorithm assumes that the control points for any curve between key points are chosen taking into account the neighboring key points. Two operations over quaternions are introduced by Shoemake in his original work [32]:

$$Double(p, q) = 2(p \cdot q)q - p \tag{9}$$

$$Bisect(p, q) = \frac{p+q}{\|p+q\|} \tag{10}$$

First, two auxiliary quaternions, $a_n$ and $b_n$, are added to control motion through the joints. They are computed as follows:

$$a_n = Bisect(Double(q_{n-1}, q_n), q_{n+1}) \tag{11}$$

$$b_n = Double(a_n, q) \tag{12}$$

Then we spherically interpolate by proportion $t$ between $q_n$ and $a_n$, $a_n$ and $b_{n+1}$, $b_{n+1}$ and $q_{n+1}$, thus obtaining three new quaternions:

$$p_1^{(1)} = Slerp(q_n, a_n; t) \tag{13}$$

$$p_2^{(1)} = Slerp(a_n, b_{n+1}; t) \tag{14}$$

$$p_3^{(1)} = Slerp(b_{n+1}, q_{n+1}; t) \tag{15}$$

Interpolation between consecutive quaternions $p_1^{(1)}$ and $p_2^{(1)}$ and $p_2^{(1)}$ and $p_3^{(1)}$ gives two more quaternions:

 

$$p_1^{(2)} = Slerp(p_1^{(1)}, p_2^{(1)}; t) \tag{16}$$

$$p_2^{(2)} = Slerp(p_2^{(1)}, p_3^{(1)}; t) \tag{17}$$

Finally, interpolation between $p_1^{(2)}$ and $p_2^{(2)}$ gives the sought-for quaternion, $q_{n+t}$:

$$q_{n+t} = Slerp(p_1^{(2)}, p_2^{(2)}; t) \tag{18}$$

Let us emphasize that the result curve will only contain $q_n$, $q_{n+1}$ and $q_{n+t}$. All the interim interpolation quaternions $p_1^{(1)}$, $p_2^{(1)}$, $p_3^{(1)}$, $p_1^{(2)}$ and $p_2^{(2)}$, as well as $a_n$ and $b_{n+1}$, are auxiliary and never included into the result curve.

Then the previous steps should be repeated for different values of parameter $t$, $t \in [0,1]$ producing new quaternions $q_{n+t}$. The step can be chosen at our discretion. The smaller the steps, the smoother the result curve and the more resources consumed.

Now that all the needed quaternions are computed and saved, together they represent a smooth curve proved to be differentiable as well. Initially Shoemake developed his algorithm for smooth computer animation, an area he worked in himself. However, the Shoemake's algorithm can serve other purposes as well, and the calculated points do not always need visualization.

One can consider interpolation parameter *t* as a time parameter. On the other hand, nothing prevents us from using the Shoemake's algorithm for forming an array of quaternions, which comprise a nice smooth curve, having only a few key points. An object being tracked may move along the curve with the same angular velocity or it may have accelerations.

Since the SLERP is used to get all the points on the hypersphere, we can calculate the total length of the curve as the sum of arc segments between pairs of consecutive quaternions. To do so, we compute the angle between two consecutive quaternions as the cosine of their dot product. The smaller the angle is, the shorter the arc segment will be. Its length can be defined as the percent of the complete circle length, which is 2π.

### *3.3. Sample Frequency*

In accordance with the well-known Shannon-Nyquist theorem, the sampling frequency for a signal should not be less than the doubled maximum frequency present in this signal. The minimum allowed sample frequency is called the Nyquist frequency. It is considered a sufficient frequency for reconstruction of the original signal without significant losses, which does not necessarily mean a required frequency.

In order to perform a signal analysis in the frequency domain, the Fourier transform is usually used along with its modifications.

The Fourier transform decomposes an arbitrary signal existing for the whole time axis into a sum of sinusoids of different frequencies. The Fourier transform $\hat{f}(w)$ of function $f(x)$ and its inverse are computed as follows:

$$\hat{f}(w) = \int_{-\infty}^{\infty} f(x) e^{-2\pi i x w} dw \tag{19}$$

$$f(x) = \int_{-\infty}^{\infty} \hat{f}(w) e^{2\pi i w x} dw \tag{20}$$

Graphically, the Fourier transform can be shown as a diagram that displays the amplitude and frequency of each of the determined sinusoids.

If a signal to be analyzed in the frequency domain has been already subjected to time-quantization, then it is processed using the discrete-time Fourier transform (DTFT). If we have a signal only defined for some finite interval $[t_1, t_2]$ instead of the whole time axis, we assume the signal has a zero value on $[-\infty, t_1)$ and $(t_2, \infty]$, which can be achieved by applying a window function. The latter leads to spectral leakages, however. The output of the DTFT is continuous in frequency and periodic, which makes it inconvenient for automated processing.

In order to use computers for calculating the frequency spectrum of a discrete-time signal, the discrete Fourier transform (DFT) is used. The DFT is defined as:

$$Y(k) = \sum_{j=1}^{N} X(j) W_N^{(j-1)(k-1)}, W_N = e^{\frac{-2\pi i}{N}} \tag{21}$$

 

The DFT is an invertible transform, whose inverse is calculated as:

$$X(j) = \frac{1}{N}\sum_{k=1}^{N} Y(k) W_N^{-(j-1)(k-1)} \tag{22}$$

The fast Fourier transform (FFT) efficiently computes the DFT, whose computational complexity is $O(n^2)$. The FFT reduces the computational complexity to $O(n \log n)$.

Hence, the FFT is what is typically used for automated frequency analysis of signals. There are multiple implementations of this algorithm, including those available in professional tools for scientific and engineer calculations. The FFT returns a vector of complex numbers that contain spectral characteristics for the corresponding frequencies. The complex number modulus of each vector item represents the magnitude whereas its argument denotes the phase. The resulting vector contains no explicit data about the frequencies themselves. Moreover, if the sampling frequency was not specified before the FFT, one will not be able to map the results to real physical frequencies. This means that it is impossible to take a discrete-time signal, perform its FFT and pinpoint the maximum frequency present in this signal. If the original signal contained frequencies exceeding the actual sample frequency, higher frequencies will be mapped into lower ones. This phenomenon is called aliasing.

Along with the sample frequency, the FFT requires the block length to be specified, i.e., the number of samples included into a fragment being analyzed. For the FFT the block length is $2^N$ ($N$ is some natural number). The bigger the FFT block length, the more accurate the resulting frequency spectrum but the slower the whole computation process. On the contrary, the smaller the FFT block length is, the faster the FFT can be accomplished. The DFT assumes that the measured signal is an integer number of periods, so the two endpoints of the time waveform are interpreted as if they were connected together. In practice this assumption is unlikely to be fulfilled, i.e. the signal values at the ends of each block do not coincide. These discontinuities between signal blocks cause spectral leakages. A spectral leakage is an appearance of parasite frequencies actually absent from the original signal. In order to fight discontinuities and avoid spectral leakages, one applies a window function. A window function is a function that has zero values everywhere outside some specific interval. Multiplying a signal being investigated by a window function, one obtains a modified signal, zeroed everywhere outside the specified interval. Thus the signal values at the ends of blocks are forced to be equal. On the other hand, use of windows means the loss of signal information, since the parts of the signal in the middle of each window matter more than the signal at the boundaries of the window.

The inverse FFT allows us to reconstruct a frequency-domain signal in the time domain.

## 4. Algorithm Description

In order to define the minimum allowable sample frequency suitable for tracking a prescribed motion scenario, we propose to piece together all the mathematical apparatus described in the previous section. In a nutshell, the algorithm is depicted in Fig. 1.

It is comprised by the following detailed steps.

Step 1. Prescribing a motion scenario. As has been mentioned before, the user, who is an embedded software or electronics engineer, should be given a possibility to set only a few key points upon which a collection of quaternions describing a smooth curve will be calculated (using the Shoemake's algorithm). An object can move along the curve faster or slower. Hence, the same motion trajectory can be captured without significant losses by the average gyroscope if the object moves slowly enough. Moreover, the object can move with a constant velocity or a changing one. To complicate the case even more, motion can be stopped abruptly in the middle of the trajectory and then renewed with some different angular velocity, i.e., the angular velocity may not be continuous. In this work we consider two options for the user. The first one is to set the time period for which the object passed the whole trajectory (the object's velocity is supposed to be constant). The second one is that the user assigns time stamps for each key point. The Shoemake's algorithm provides us with a collection of points, and the velocities/time stamps can be assigned afterwards. In practice, as one can infer from the above-stated considerations, the user may require a possibility to assign angular velocity vectors to trajectory key points. This issue will be covered in our next paper.

The result of this algorithm step is $Q$, an array of quaternions, obtained by the Shoemake's algorithm.

Step 2. Calculation of the angular velocity vector for the motion trajectory. Now that we have a smooth curve and time periods associated with the whole curve or its parts, we can make use of (4) and (5). We replace the quaternion derivative, $\dot{\Lambda}$, by the corresponding limit $\lim_{\Delta t \to 0} \frac{\Delta \Lambda}{\Delta t}$. In order to calculate all the limits we take the differences of all the consecutive quaternions in $Q$ and divide them by $dt$, which is obtained as the whole time period divided by the number of points visited during this period. For our array of quaternions $Q$, we calculate their conjugates $Q^*$ (or inverses, which is the same for unit quaternions). Using (6) with derivatives replaced by limits, we obtain an array of quaternions, $\Omega$.

 

This array describes the angular velocity vector in each point of the curve represented by *Q* in respect to some fixed frame. The result of Step 2 is array Ω.

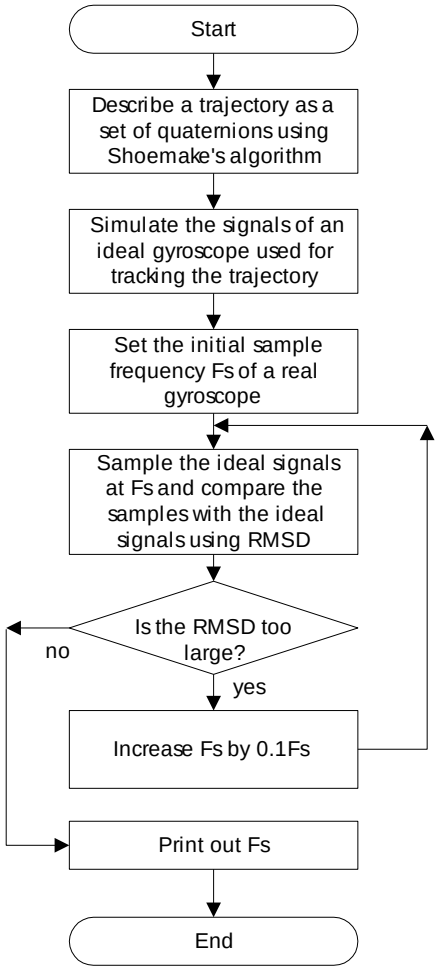


Fig.1. The flow-chart of the algorithm

Step 3. Expressing the angular velocity vector in the body frame coordinates. Next, we represent the same array of angular velocity vectors in the body frame, making use of (6). In this way we get a new array, $\Omega^*$ that contains the angular velocity vector projections onto the axes assigned with an ideal gyroscope. We can consider each angle velocity vector projection individually, i.e., we extract $\Omega_x^*$, $\Omega_Y^*$ and $\Omega_Z^*$ out of array $\Omega^*$. Thus we obtain three signals that might have been measured by an ideal gyroscope if it could have been used for tracking the prescribed motion scenario in real life. Step 3 results in arrays $\Omega_x^*$, $\Omega_Y^*$ and $\Omega_Z^*$.

Step 4. Frequency analysis of the angular velocity vector components. Taking into account the above-stated properties of the FFT and inverse FFT, a suitable sample frequency will be determined iteratively as follows. First, we start with some low initial sample frequency *Fs*, which is still sufficient for tracking human motion. As our literature review has shown, the least sample frequency ever claimed for motion capture is 2 Hz. The discrete-time signal $\Omega_x^*$, which simulates the readings of the ideal gyroscope along the axis *X* in the body frame, is subjected to the FFT at sample frequency *Fs*. The obtained result is transformed back into the time domain using the inverse FFT. Thus we obtain some discrete-time signal $S_1^*$, which reproduces $\Omega_x^*$ to some extent. In order to measure the discrepancies between $\Omega_x^*$ and $S_1^*$, we first perform the linear interpolation of the latter. That is, the consecutive discrete-time points are connected with straight line segments. After that we calculate the root-mean-square deviation (RMSD):

$$RMSD = \frac{\sqrt{\sum_{t=1}^{T}(x_{1,t} - x_{2,t})^2}}{T} \tag{23}$$

Usually, the concept of the RMSD relates to statistical disciplines and implies computation of the differences between sample values and the mean, i.e. benchmarking them against some "standard". Here we use the RMSD to compare differences between the corresponding points of two signals. One can easily figure out that the higher the sample frequency is, the closer two signals will be and, correspondingly, the less RMSD will be achieved.

Some acceptable error should be set in advance. If the RMSD exceeds the threshold, then we increase sample frequency *Fs*. The process is repeated until such a sample frequency is found that the two signals, $\Omega_x^*$ and $S_1^*$, are close enough for the acceptable RMSD value to be reached. Step 4 should be taken for two other gyroscope dimensions, *Y* and *Z*. It is quite possible that the same device has different technical characteristics along different axes. Thus the axes are not to be perplexed – one should make their conclusions about each axis individually.

Step 5. Final decision. If there are several motion trajectories to be captured by the same gyroscope, the worst case should be chosen. I.e., steps 1-4 are to be applied to each trajectory individually, which results in several values of the minimum sample frequency. The highest sample frequency should be chosen.

## 5. Approbation

Verification of the proposed algorithm has been performed in several steps. Firstly, we generated signals using MIS_IDE. It is built upon module GY-80 (Fig. 2), which incorporates a 3D digital gyroscope L3G4200D. The latter can provide measurements at a sample frequency 100, 200, 400 or 800 Hz. The resolution of analogue-to-digital conversion is 16 bit. One switches between operation modes using corresponding registers.

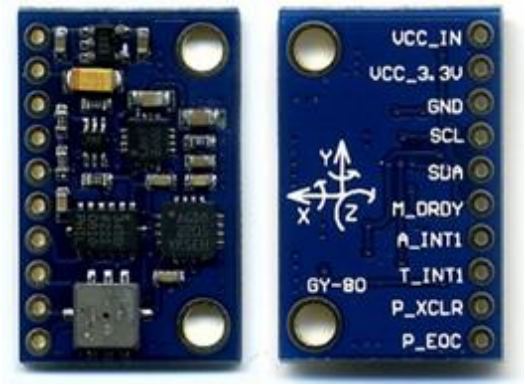


Fig.2. IMU module GY-80

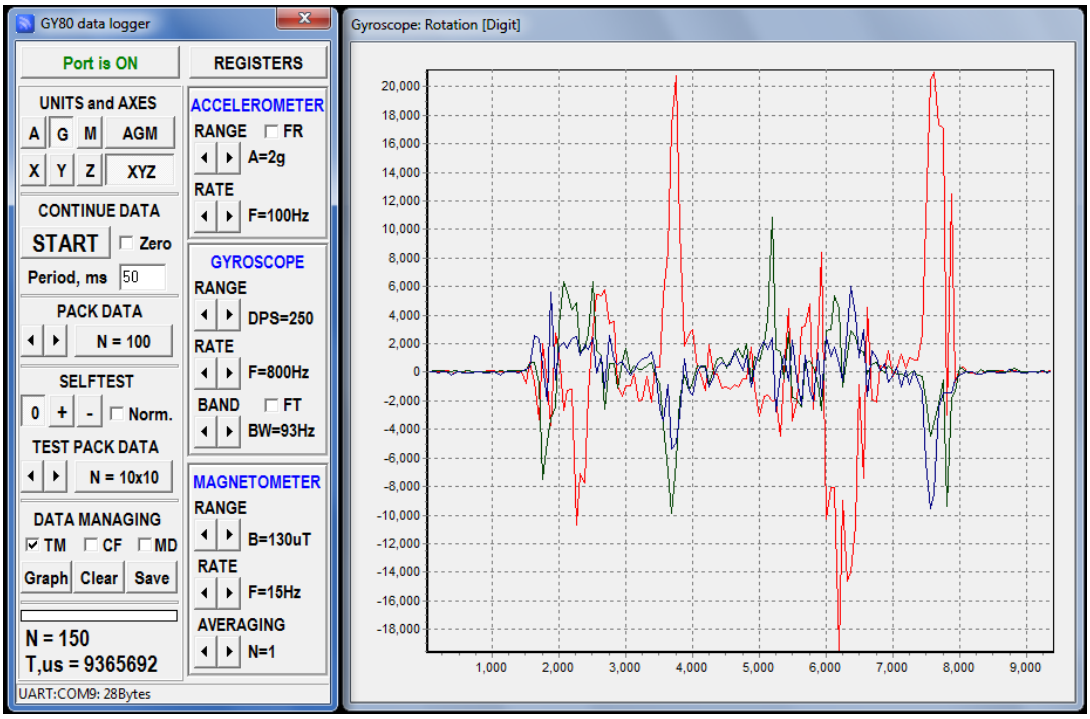


Fig.3. Gyroscope signals corresponding to sharp motions sampled at 800 Hz

Having gripped the module with fingers we performed quick chaotic motions with the wrist, when the rest of the arm was kept steady. The sample frequency was set to 800 Hz. Undoubtedly this experiment cannot pretend to accuracy since there were no precise mechanisms applied for registering motion. However, it suffices to illustrate the significance of selecting the right sample rate for tracking motion. Fig. 3 depicts sample gyroscope signals obtained during quick motions.

It can be easily seen that if one reduced the sample rate to 200 Hz, some peaks would be irreproducible, which highlights the role of the properly chosen sample rate. In a steady position, the module produced small noise close to zero. The sampled signals inevitably contain noise, mostly bias drift. Slow motions were reflected by smoother curves of the sampled signals.

Then, we integrated the sampled signal to obtain the curves. The curves have their imperfections due to integrating noise. Then we picked up points corresponding to equally spaced time moments. Upon these points trajectories have been constructed. Since the user is interested in a visual, interactive drag-and-drop tool for building a trajectory, we used our own software module [34]. The module was developed primarily with the purpose to integrate it into MIS-IDE. It is based on OpenGL and allows the user to either select a well-known motion scenario (for instance, a bouncing ball trajectory), or set a customized scenario. The basic idea of the module is that any embedded software engineer can prescribe a motion curve even if they are not familiar with OpenGL programming. The tool has not been integrated in MIS-IDE yet, however, it allows the user to dump all the data into a file.

At the next step we exported curve data into MATLAB and first calculated angular velocities in the inertial frame, then performed coordinate transformation in order to obtain the angular velocity vector in the body frame. Thus we simulated signals that might have been provided with an ideal gyroscopes used to track the preset trajectory. It worth mentioning that the simulated angular velocities never matched the real gyroscope signals. This can be ascribed to several reasons. The main reason is the fact that the curves were built upon several dozens of key points, thus, not surprisingly, some information from the real signals has been lost. When increasing the amount of points the module for prescribing a motion scenario slowed down drastically, which indicates the need to investigate better into the influence of the key point amount on the module performance. Another reason is that the curves obtained by integration did not reflect the real motion. Despite the fact that the exact correspondence has never been achieved, in general the simulated signals resembled the shape of the real signals.

Then these signals were subjected to the FFT with the sample frequency increased iteratively. In each iteration, the sample frequency was increased by 10% of its previous value. The frequency-domain signal was then transformed back into the time-domain using the inverse FFT. The discrepancies between each signal before the FFT and the signal after the inverse FFT were estimated using the RMSD. Formula (23) can only be applied to sequences of the same length. That is why the points missing in the signal, restored by the inverse FFT, were simulated by the linear interpolation. All the calculations were performed programmatically. Besides, we plotted the interim results for visual examination as well.

The approbation procedure cannot pretend to be an exact proof. Nevertheless, it demonstrated that 1) even a sophisticated trajectory can be rendered by the Shoemake's algorithm; 2) smooth movements required smaller sample frequencies than quick and chaotic ones; 3) inappropriate sample rates for chaotic motion led to irrevocable loss of information.

## 6. Conclusion

The above-stated algorithm has several aspects that need further explanation and investigation. Firstly, a question arises, under which conditions a replacement of a quaternion derivative by the corresponding limit is appropriate. Secondly, it is not completely clear how to assign time to the difference of two consecutive quaternions for the limit calculation. The problem is that if the user specifies key points and the time period spent on the whole trajectory, it does not mean that an object moves with a constant velocity. On the contrary, it can move along some part of its trajectory literally in no time, then slow down abruptly and renew its way with a much lesser velocity. The user may need an option of assigning angular velocities to different parts of a motion trajectory. Instead of limits, derivatives can be used. However, in this case analytical functions are needed. Calculation of analytical expressions for the derivatives is a way of further improvement of our algorithm. Thirdly, other algorithms (SQUAD, SPRING) might have been investigated along with the Shoemake's algorithm. Finally, the FFT introduces spectral leakages. Instead of a "brute force" search for a minimum sample frequency, a more efficient search algorithm might have been implemented. On the one hand, if the sample frequency is increased in small steps, it consumes computational resources. On the other hand, if the steps are not small enough, some gyroscopes may be erroneously rejected.

These issues comprise our future work scope.

Nevertheless, the proposed algorithm is applicable for figuring out a sample rate sufficient for tracking a prescribed motion and thus contributes to further filtering performance.

 

## References


[1] M. El-Gohary, "Joint Angle Tracking with Inertial Sensors", Ph.D. dissertation, Portland State University, Portland, USA, 2013. doi: 10.15760/etd.661.

[2] A. Bulling, U. Blanke and B. Schiele, "A tutorial on human activity recognition using body-worn inertial sensors", ACM Computing Surveys, vol. 46, no. 3, pp. 1-33, 2014. doi: 10.1145/2499621.

[3] T. Ngo, Y. Makihara, H. Nagahara, Y. Mukaigawa and Y. Yagi, "Similar gait action recognition using an inertial sensor", Pattern Recognition, vol. 48, no. 4, pp. 1289-1301, 2015. doi: 10.1016/j.patcog.2014.10.012.

[4] C. Chen, R. Jafari and N. Kehtarnavaz, "A survey of depth and inertial sensor fusion for human action recognition", Multimedia Tools and Applications, vol. 76, no. 3, pp. 4405-4425, 2015. doi: 10.1007/s11042-015-3177-1.

[5] A. Filippeschi, N. Schmitz, M. Miezal, G. Bleser, E. Ruffaldi and D. Stricker, "Survey of Motion Tracking Methods Based on Inertial Sensors: A Focus on Upper Limb Human Motion", Sensors, vol. 17, no. 6, p. 1257, 2017. doi: 10.3390/s17061257.

[6] S. Daroogheha, T. Lasky and B. Ravani, "Position Measurement Under Uncertainty Using Magnetic Field Sensing", IEEE Transactions on Magnetics, vol. 54, no. 12, pp. 1-8, 2018. doi: 10.1109/tmag.2018.2873158.

[7] Q. Mourcou, A. Fleury, C. Franco, F. Klopcic and N. Vuillerme, "Performance Evaluation of Smartphone Inertial Sensors Measurement for Range of Motion", Sensors, vol. 15, no. 9, pp. 23168-23187, 2015. doi: 10.3390/s150923168.

[8] A. Mao, X. Ma, Y. He and J. Luo, "Highly Portable, Sensor-Based System for Human Fall Monitoring", Sensors, vol. 17, no. 9, p. 2096, 2017. doi: 10.3390/s17092096.

[9] X. Chen, "Human Motion Analysis with Wearable Inertial Sensors", PhD dissertation, University of Tennessee, Knoxville, USA, 2013.

[10] K. Różanowski, J. Lewandowski and M. Placha, "Determination of the Orientation of an Object within a 3D Space. Implementation and Example Applications", International Journal of Microelectronics and Computer Science, vol. 6, no. 4, pp. 124 - 129, 2015.

[11] D. Dinu, M. Fayolas, M. Jacquet, E. Leguy, J. Slavinski and N. Houel, "Accuracy of Postural Human-motion Tracking Using Miniature Inertial Sensors", Procedia Engineering, vol. 147, pp. 655-658, 2016. doi: 10.1016/j.proeng.2016.06.266.

[12] D. Zwartjes, T. Heida, J. van Vugt, J. Geelen and P. Veltink, "Ambulatory Monitoring of Activities and Motor Symptoms in Parkinson's Disease", IEEE Transactions on Biomedical Engineering, vol. 57, no. 11, pp. 2778–2786, 2010.

[13] Y. Li, K. Yao and G. Zweig, "Feedback-based handwriting recognition from inertial sensor data for wearable devices", in 2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Brisbane, QLD, Australia, 2015. doi: 10.1109/icassp.2015.7178375.

[14] B. Fasel, "Drift reduction for inertial sensor based orientation and position estimation in the presence of high dynamic variability during competitive skiing and daily-life walking", PhD. dissertation, EPFL, Lausanne, Switzerland, 2017. doi: 10.5075/epfl-thesis-7803.

[15] B. Fasel, J. Spörri and K. Aminian, "Improving the accuracy of low-cost GNSS by fusion with inertial and magnetic sensors in alpine ski racing", in 34rd Conference on Biomechanics in Sports 2016, Tsukuba, Japan, 2016.

[16] B. Fasel, M. Gilgien, J. Spörri and K. Aminian, "A new training assessment method for alpine ski racing: estimating center of mass trajectory by fusing inertial sensors with periodically available position anchor points", Frontiers in Physiology, vol. 9, 2018. doi: 10.3389/fphys.2018.01203.

[17] P. Siirtola, "Recognizing human activities based on wearable inertial measurements. Methods and applications", PhD dissertation, University of Oulu, Oulu, Finland, 2015.

[18] R. Aylward, "Sensemble: A Wireless Inertial Sensor System for Interactive Dance and Collective Motion Analysis", Master of Science thesis, Massachusetts Institute of Technology, Cambridge, USA, 2006.

[19] Z. Tang, M. Sekine, T. Tamura, N. Tanaka, M. Yoshida and W. Chen, "Measurement and Estimation of 3D Orientation using Magnetic and Inertial Sensors", Advanced Biomedical Engineering, vol. 4, pp. 135-143, 2015. doi: 10.14326/abe.4.135.

[20] H. Gjoreski and M. Gams, "Activity/Posture Recognition using Wearable Sensors Placed on Different Body Locations", in The Fourteenth International Conference on Artificial Intelligence and Soft Computing, 2011. doi: 10.2316/p.2011.716-067.

[21] P. Gupta and T. Dallas, "Feature Selection and Activity Recognition System Using a Single Triaxial Accelerometer", IEEE Transactions on Biomedical Engineering, vol. 61, no. 6, pp. 1780-1786, 2014. doi: 10.1109/tbme.2014.2307069.

[22] J. Guiry, P. van de Ven, J. Nelson, L. Warmerdam and H. Riper, "Activity recognition with smartphone support", Medical Engineering & Physics, vol. 36, no. 6, pp. 670-675, 2014. doi: 10.1016/j.medengphy.2014.02.009.

[23] Y. Liang, X. Zhou, Z. Yu and B. Guo, "Energy-Efficient Motion Related Activity Recognition on Mobile Devices for Pervasive Healthcare", Mobile Networks and Applications, vol. 19, no. 3, pp. 303-317, 2013. doi: 10.1007/s11036-013-0448-9.

[24] Z. Yan, V. Subbaraju, D. Chakraborty, A. Misra and K. Aberer, "Energy-Efficient Continuous Activity Recognition on Mobile Phones: An Activity-Adaptive Approach", in 2012 16th International Symposium on Wearable Computers, Newcastle, 2012. doi: 10.1109/iswc.2012.23.

[25] F. Wu, H. Zhao, Y. Zhao and H. Zhong, "Development of a Wearable-Sensor-Based Fall Detection System", International Journal of Telemedicine and Applications, vol. 2015, pp. 1-11, 2015. doi: 10.1155/2015/576364.

[26] H. Gjoreski, M. Gams and M. Luštrek, "Context-based fall detection and activity recognition using inertial and location sensors", Journal of Ambient Intelligence and Smart Environments, vol. 6, no. 4, pp. 419-433, 2014.

[27] F. Destelle, A. Ahmadi, N. O'Connor, K. Moran, A. Chatzitofis, D. Zarpalas, and P. Daras, "Low-cost accurate skeleton tracking based on fusion of Kinect and wearable inertial sensors", in 22nd European Signal Processing Conference (EUSIPCO), Lisbon, Portugal, 2014. doi: 10.5281/zenodo.44122.

[28] M. Gjoreski, H. Gjoreski, M. Luštrek and M. Gams, "How Accurately Can Your Wrist Device Recognize Daily Activities and Detect Falls?", Sensors, vol. 16, no. 6, p. 800, 2016. doi: 10.3390/s16060800.

[29] C. Murphy, "Choosing the Most Suitable MEMS Accelerometer for Your Application—Part 2", Analog Dialogue, 2017.


 

[30] L. Landry, "Inertial Sensor Characterization for Inertial Navigation and Human Motion Tracking Applications", Master's Thesis, Naval Postgraduate School, Monterey, USA. 2012.
[31] M. Kok, J. Hol and T. Schön, "Using Inertial Sensors for Position and Orientation Estimation", Foundations and Trends® in Signal Processing, vol. 11, no. 1-2, pp. 1-153, 2017. doi: 10.1561/2000000094.
[32] D. Fedasyuk, R. Holyaka and T. Marusenkova, "Method of Analyzing Dynamic Characteristics of MEMS Gyroscopes in Test Measurement Mode", in 2019 9th International Conference on Advanced Computer Information Technologies (ACIT), pp. 157–160, 2019. doi: 10.1109/acitt.2019.8780058.
[33] K. Shoemake, "Animating Rotation with Quaternion Curves", SIGGRAPH, vol. 19, no 3, pp. 245-254,1985.
[34] Y. Stets and T. Marusenkova, "The module for prescribing motion scenarios in hardware-software tools for inertial sensors assessment" in *Int. Youth Science Forum «LITTERIS ET ARTIBUS»*, Lviv, Ukraine, Nov. 2019.

## Authors' Profiles

**Dmytro V. Fedasyuk** was born in 1955. He graduated from Lviv Polytechnic National University in 1977 as an expert in radio-frequency engineering. He started his career in State Enterprise "Scientific-Research Institute For Metrology of Measurement And Control Systems" ("Systema"). Three years later he returned to Lviv Polytechnic National University in order to proceed with both scientific and teaching activities. There he gradually evolved from a junior researcher to the Vice-Rector for Undergraduate Education.

In 1985 he defended his Ph.D. thesis "Automated methods for modeling and analysis of thermo-electrical characteristics of microelectronic units for radio-electronic hardware CAD" in the city named today St. Petersburg. In 2000 he became a doctor of science after defending his work "Automation of thermal design of microelectronic systems" in Lviv. In 2002 he was promoted to the academic rank of professor. In 2004 he became the head of Software Department, a relatively recent department of the same university and left this position in 2015 to focus on his vice-rector role. After working as a vice-rector for twelve years, he concentrated completely on his scientific and educational activities. His main fields of interest are mathematical modeling; modeling and analysis of thermo-electrical processes in microelectronic systems, Internet technologies, software design.

Prof. Fedasyuk's scientific contribution is contained in two monographs and over 150 works published in a wide range of scientific journals included those well-known all over the world. He attended conferences in Germany, France, Italy, Poland, Hungary and many other countries. Under his supervision, a number of Ph.D. students have successfully got their degrees. Prof. Fedasyuk is a Member of IEEE, a Member of Shevchenko Science Society, a Member of UFI in Ukraine and a co-editor of several scientific journals.

**Tetyana A. Marusenkova** was born in 1982. She received the master degree in the Institute of Computer Science and Informational Technologies of Lviv Polytechnic National University in 2005. After working as a software engineer in several IT companies, she came back to the University in 2009 and joined Electronic Devices Department as a postgraduate student. In 2013 she got Ph.D. degree having defended thesis "Semiconductor magnetic sensors based on split Hall structures".

In 2011 she joined Software Department of Lviv Polytechnic National University as a teacher. In 2014 she started working in the team built of teachers and students of Software Department in order to develop embedded systems in collaboration with Dinamica Generale S.p.A., an Italian company that specializes on innovative electronic solutions and sensors for agriculture. In 2018 she was promoted to the academic rank of Associate Professor. She is a co-author of over 70 papers and proceedings, two monographs and two textbooks. Her main interests are mathematical modeling, MEMS inertial sensors, data fusion algorithms, electromagnetic tracking and embedded systems.

**How to cite this paper:** Dmytro V. Fedasyuk, Tetyana A. Marusenkova, "An Algorithm for Detecting the Minimal Sample Frequency for Tracking a Preset Motion Scenario", International Journal of Intelligent Systems and Applications(IJISA), Vol.12, No.4, pp.1-12, 2020. DOI: 10.5815/ijisa.2020.04.01

 