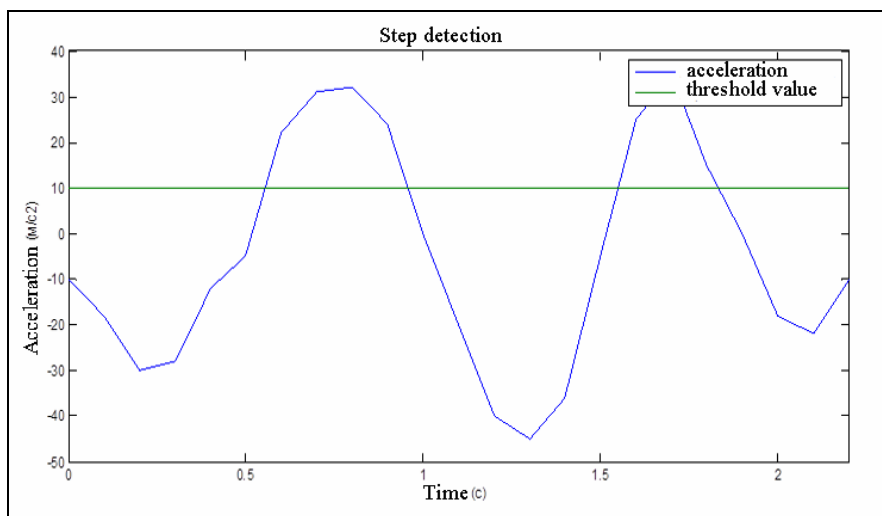


## **Determining the location of objects using inertial sensors in mobile devices**

At the present time the increase of interest to the opportunity of determining the location of some object is observed. The information about the location enables to increase quality of service for the clients. The tendencies of mobile device development constantly bring to a great amount of new functions and capabilities in mobile devices. One of such results is modernization of a great number of new devices with inertial sensors which are widely used in different infotainment applications. However there is a possibility to use this hardware to solve the problems of navigation indoor which become more and more urgent because of the expansion in the number of building projects with difficult infrastructure. The present work objective is the investigation of accuracy of determining the location of objects with inertial sensors in mobile devices [Saleh 2010: 49–52].

To track the location of an object with the help of a mobile device the algorithm of step detection based on the threshold is used. The point is that a step has occurred whenever a value above the threshold value is followed by a value below the threshold.



**Pic.1. Visualization of step detection algorithm**

The function which helps to determine that according to the sensor and the given value a step has occurred is the following:

$$step(i,T) = \begin{cases} 1, & \text{if } a(i) < T \text{ and } a(i-1) > T \\ 0, & \text{else} \end{cases}$$

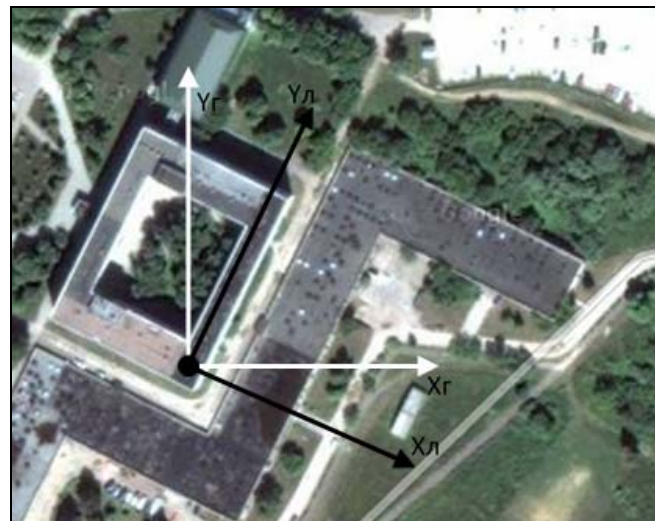
where  $a(i)$  – is a readout of the accelerometer.

Assuming the number of steps is known, determining the distance moved and consequently the new location of the object is a matter of determining of movement given by each step. Mathematically, for  $n$  steps with lengths  $d_1, d_2, \dots, d_n$  the total distance moved  $D$  can be calculated by:

$$D = \sum_{i=1}^n d_i.$$

Determining the step length by looking at the accelerometer output is difficult. It is much easier to estimate the length of a step by a constant  $d$  corresponding to the average step length of the current user of the system. Though there are two problems with this: first, the average step length must be known. It can be calculated, for example, by having each user walk a fixed distance once, and divide this distance with the number of steps. Second, the stride of a person is anything but constant [Grewal, Weill, Andrews 2000].

As a range for performing the main experiment the corridor of the second block of Vladimir State University was chosen. Its location in the global system of axes is depicted in picture 2. Axis direction is shown by the lines of white colour.



**Pic. 2. Position of local system of axes**

The local system of axes referred to the positioning area is shown in the picture by the lines of black colour. The system is fixed in such a way on purpose,

according to the experiments the movement ran from the southern point of the building to the eastern one. The result of using such a coordinate system is that the movement was strictly in the positive direction of  $Y$ -axis that simplifies appraisal of calculations for accuracy.

During the experiment a person moved in the way said above holding the phone on his palm in front of him in such a way that  $Y$ -axis of the phone coincided with the  $Y$ -axis of the local coordinate system, he tried to keep up this position while walking the whole distance. The recording started before the moment of movement and was over after the full stop. The real corridor extension and consequently the distance covered are 66 meters.

During the experiment two files with data containing accelerometer and gyroscope outputs respectively were recorded. The recording duration was 51.7 seconds. The number of recorded gyroscope value was 3455, and accelerometer one was 2588 that shows the average recording frequency equal to approximately 67 Hz and 50 Hz respectively.

All sensors have measures of inaccuracy limiting measurement accuracy of real value. The gyroscope value can be presented in the following way [Hansson, Tufvesson 2011]:

$$\omega_i = K_\omega \omega_{true} + b_\omega + N_i + R_i,$$

where  $\omega_{true}$  is the true value that is desired,  $K_\omega$  is a scale factor,  $b_\omega$  is a constant bias in the sensor,  $N_i$  is a random noise in the sensor and  $R_i$  a model for bias instability. Similarly for the accelerometer measurements  $a_i$ :

$$a_i = K_a a_{true} + b_a + N_i + R_i.$$

Error indexes were calculated for our sensors:

$$K_\omega = (K_{\omega,x}, K_{\omega,y}, K_{\omega,z})^T = (1,037 \ 0,989 \ 0,$$

$$K_a = (K_{a,x}, K_{a,y}, K_{a,z})^T = (0.975 \ 0.985 \ 0.9367)^T,$$

$$b_\omega = (b_{\omega,x}, b_{\omega,y}, b_{\omega,z})^T = (0.00636, -0.00473, -0.00051)^T,$$

$$b_a = (b_{a,x}, b_{a,y}, b_{a,z})^T = (0.29685, -0.3553, -0.02$$

If the measured sensor value (for a gyroscope ) is:

$$\omega_i = K_\omega \omega_{true},$$

the real value is respectively:

$$\omega_i = K_\omega \omega_{true}.$$

The formula for an accelerometer is similar.

$$\omega_{true} = \frac{\omega_i - b_\omega}{K_\omega}$$

If value measured by sensors is equal (for a gyroscope):

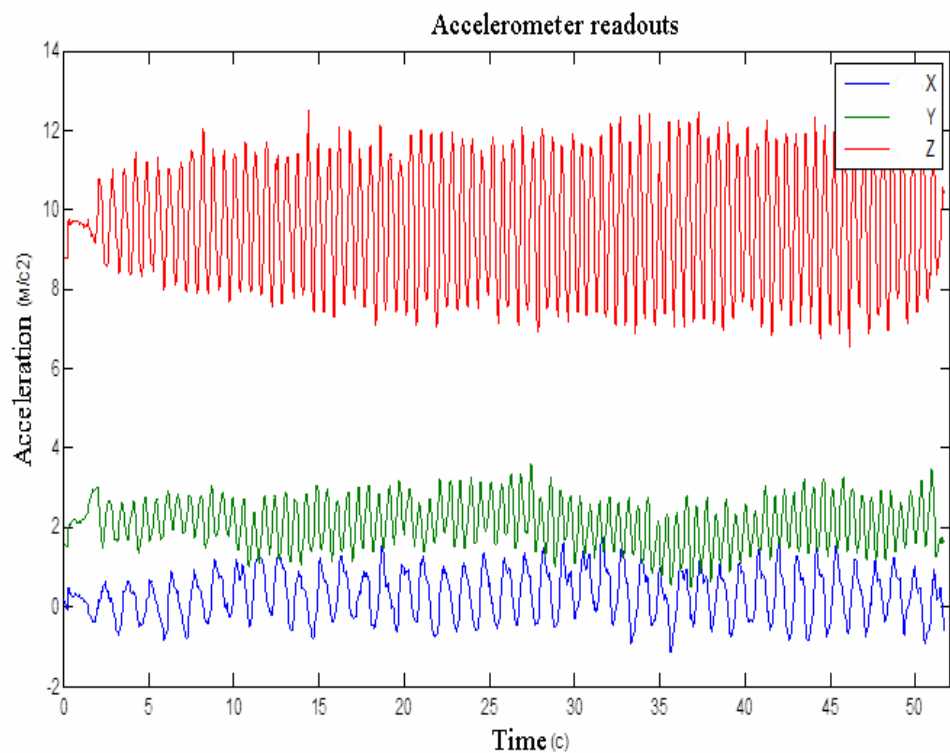
$$\omega_i = K_\omega \omega_{HCT},$$

the true value is respectively equal:

$$\omega_{HCT} = \frac{\omega_i}{K_\omega}.$$

For the accelerometer the formula is similar.

The next stage of processing is smoothing, the aim of which is to get rid of abrupt readouts. Smoothing was done by the method of a floating average value with the interval width equal to 11 [Internet]. As the result the data is as shown in picture 3.



**Pic. 3. The diagram of accelerometer readouts after processing**

For the implementation of our model three classes were worked out:

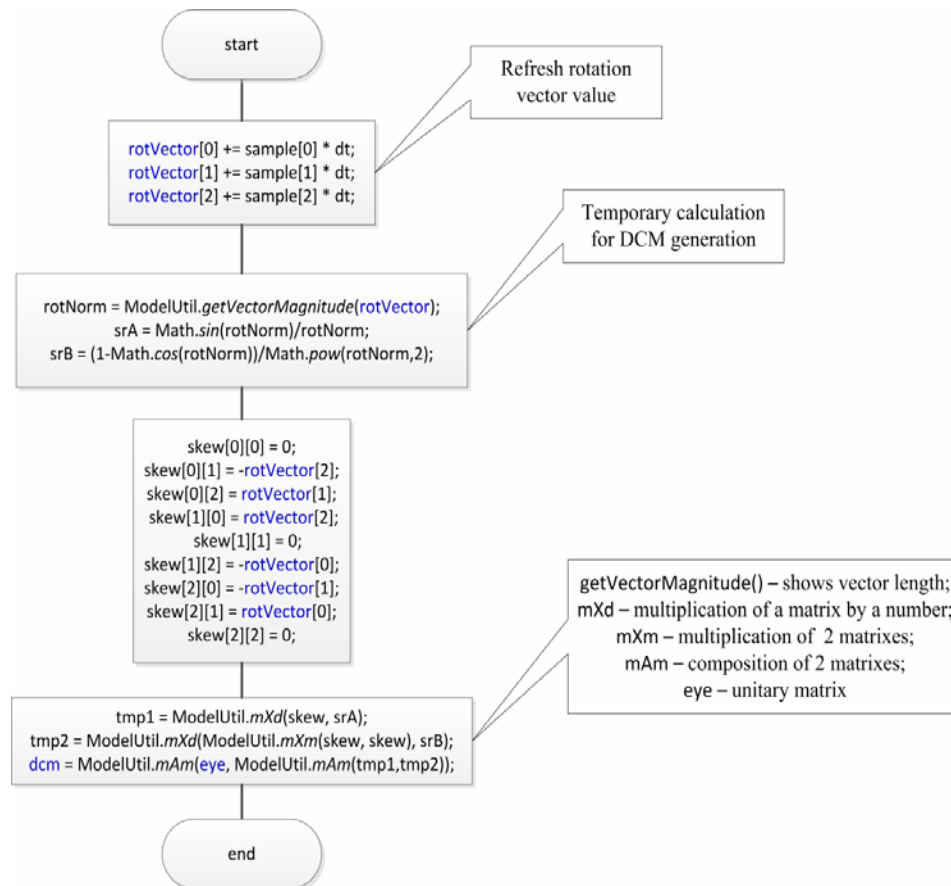
*DataSample.java* – is a sensor measurement, it contains the same margins as a line in the file and the algorithm of the line conversion to the number format.

*InsModelUtil.java* – keeps different functions needed for the work of an algorithm. They are: vector magnitude finding, multiplication of two matrixes, multiplication of a matrix by a vector, multiplication of a matrix by a number.

*InsModel.java* – implementation of the algorithm based on direction cosines. It has two main methods:

***updateAtt(double[] sample, double dt)***

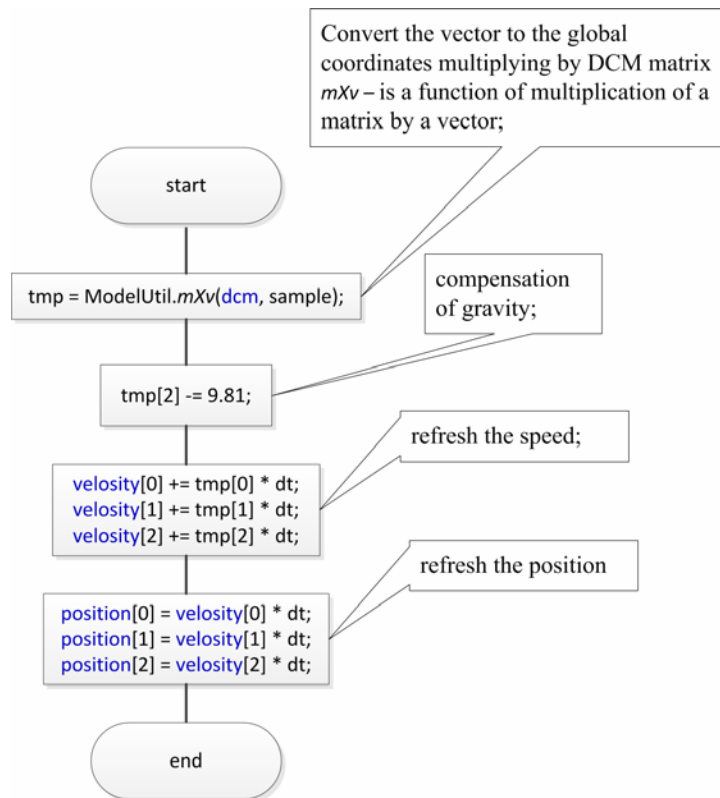
It takes gyroscope outputs and follows the following steps to recalculate the orientation (pic. 4):



**Pic. 4. Realization of the algorithm of orientation tracking**

***updatePos(double[] sample, double dt)***

It takes accelerometer outputs and refreshes the position (pic. 5):



**Pic. 5. Realization of the algorithm of position tracking**

Besides these classes realizing an algorithm the program has a mechanism of reading data from files, it imitates data streams going from real sensors.

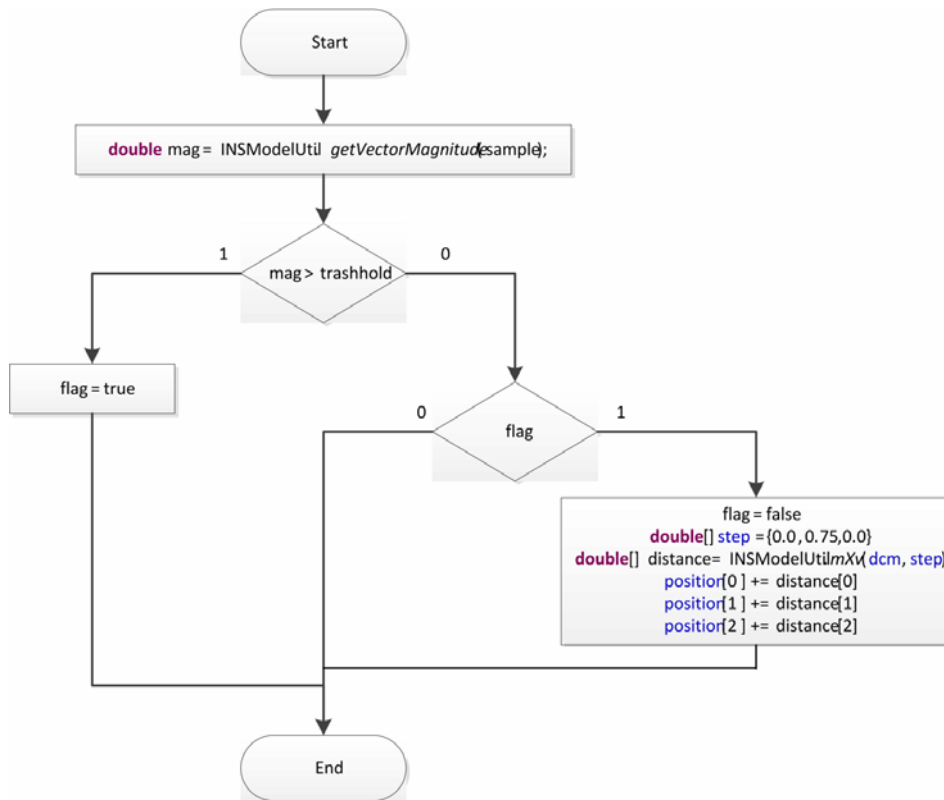
The model with the realization of the step detection algorithm consists in changing only a part of the movement tracking model. The algorithm to follow with every new accelerometer output is shown in picture 6.

As the result of the experiment of walking a fixed distance and counting the number of steps made the stride length is found to be equal to 0.75 m.

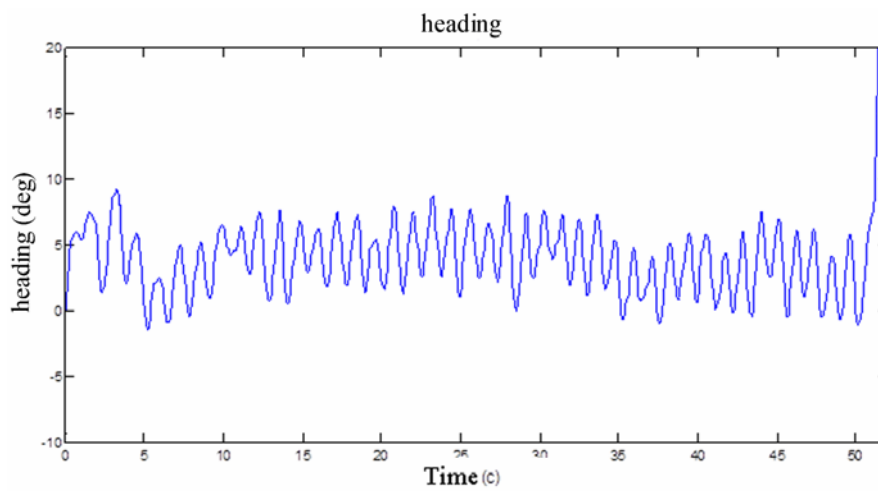
To estimate the work of the model a number of diagrams reflecting some change of calculated model indexes with the time is depicted.

The course diagram is shown in picture 7. Course is an angle of system rotation around Z-axis or to put it otherwise it is a system orientation in the plane XY. At given conditions of experimentation this index is determinative for system orientation tracking.

The diagram shows that the course value changes in the manner of a sinusoid within the range of 0–10 degrees, however, the average value remains almost invariable during the whole walk that corresponds to reality. Vibrations can be explained by the nature of system motion.

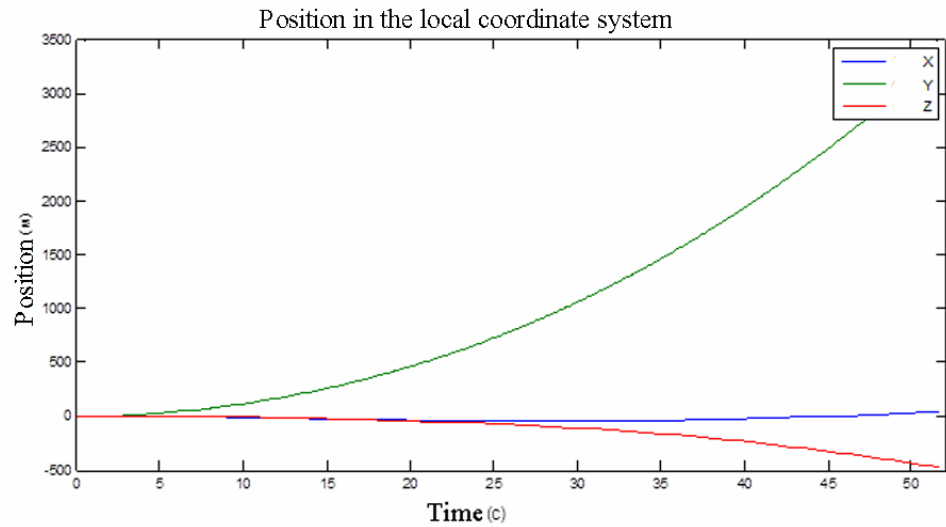


**Pic. 6. Realization of the step detection algorithm**



**Pic. 7. The diagram of course change**

According to the accelerometer outputs the diagram of the object position on three axes is shown in picture 8 (the position in the local coordinate system).



**Pic. 8. The diagram of the position change**

Without using the algorithm of step detection at the moment of a full stop the indexes had the following value (table 1):

**Table 1**

**Final outputs of the model without using the step detection algorithm**

Axis	Speed	Position, m
<i>X</i>	7.312	42.663
<i>Y</i>	136.949	3332.214
<i>Z</i>	-24.990	-475.173

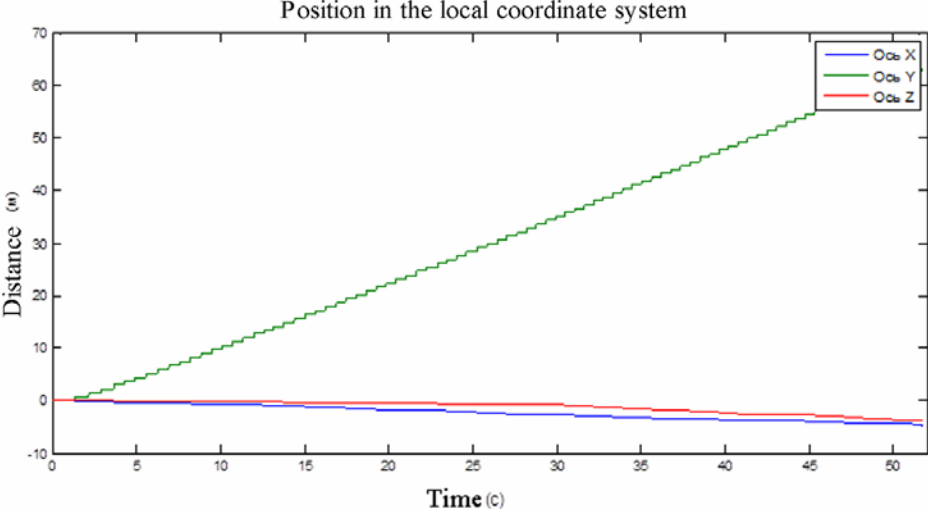
The same data were used when calculating using the step detection algorithm. As the orientation tracking algorithm has become invariable, so it has no sense to give the course diagram and it completely corresponds to the derived one (pic. 7).

The diagram of the position change on three axes is given in the picture 9.

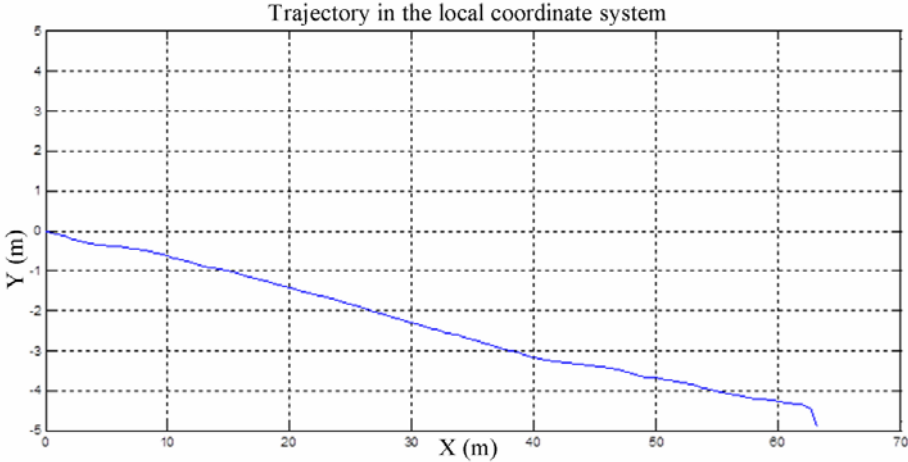
One can clearly see in the picture that there is discrete behavior of position change caused by the specificity of the algorithm. However, the position change approximately corresponds to the real one. To interpret the derived result better let us turn to the path based on the calculated indexes (pic. 10).



The path almost fully corresponds to the real one with the exception of the fact that there is a deviation angle that appears because the local coordinate system does not ideally coincide with the direction of the corridor where the walk took place and the local system is the starting orientation of the phone but it changed a little bit at the start of recording. At the moment of full stop the model indexes were as follows (table 2):



**Pic. 9. The diagram of the position change with the use of the step detection algorithm**



**Pic. 10. The motion trajectory**

**Table 2****The final model outputs without using the step detection algorithm**

Axis	Position, m
<i>X</i>	- 4.943
<i>Y</i>	63.227
<i>Z</i>	- 3.633

The number of steps is 88.

### Conclusion

Having run the experiments with the model using the data got while walking along a straight 66 meter corridor the results are the following.

The shift from the starting course has occurred to be in the limits of five degrees and that is an appropriate index even if we omit inaccurate alignment of the starting orientation. It is evidence of compatibility of gyroscopes set in the phone to solve this kind of problems.

Distance calculation based on inertial navigation has given deplorable results, the final value is 3366.2 m, that is 51 times as much as the real value. Such a great error is caused by first of all accelerometers with poor accuracy set in the device; they are not able to give accurate data concerning acceleration influencing the device.

However, the further research of possibility of using accelerometers shows that they are good as motion sensors especially they can precisely detect the number of steps made. Using even the simplest detection algorithm the calculated distance is 63.524 m, that is 96% from the real one and goes into the acceptable error 5%.

Within the frame of this work the research of position-finding capabilities of phone sensors was done. For the further development of this area usage of improved orientation and stride length algorithms instead of given in the present work can be offered.

### Literature

Grewal M.S., Weill L.R., Andrews A.P. (2000), *Global Positioning Systems, Inertial Navigation, and Integration*, ISBN-13: 978-0-470-04190-1 – John Wiley & Sons. – 416 c.

Hansson A., Tufvesson L. (2011), *Using Sensor Equipped Smartphones To Localize WiFi Access Points. master thesis*, Lund University.

Internet: Google Inc. Android Developers – SensorManager, May 2012. URL <http://developer.android.com/reference/android/hardware/SensorManager.html>

Saleh H.M. (2010), *Positioning of technical objects using high-sensitivity inertial sensors* / H.M. Saleh, Y.V. Novozhilov // Vestnik of Kostroma State University named after N.A. Nekrasov. – Kostroma. – T. 16. – P. 49–52.

### **Abstract**

The article is devoted to questions of development of inertial navigation system for handheld devices and development of navigation algorithms on the basis of using accelerometers and gyroscopes in a mobile phone to track the movement of a walking person.

**Key words:** mobile device, inertial sensor, inertial navigation system, accelerometer, gyroscope.

### **Определение местоположения объектов с использованием инерционных датчиков в мобильных устройствах**

#### **Резюме**

Статья посвящена разработке инерциальной навигационной системы для мобильных устройств и навигационных алгоритмов на основе использования акселерометров и гироскопов в мобильном телефоне для отслеживания перемещения человека.

**Ключевые слова:** мобильное устройство, инерционный датчик, инерциальная навигационная система, акселерометр, гироскоп.

### **Ustalanie lokalizacji obiektów wykorzystujących czujniki inercyjne w urządzeniach mobilnych**

#### **Streszczenie**

Artykuł poświęcony jest rozwojowi systemu nawigacji inercyjnej dla urządzeń przenośnych oraz opracowaniu algorytmów nawigacji w oparciu o użycie akcelerometrów i żyroskopów wbudowanych w telefon komórkowy służących do śledzenia ruchu osoby pieszej.

**Słowa kluczowe:** urządzenia mobilne, czujnik inercyjny, inercyjny system nawigacji, akcelerometr, żyroskop.