An Increase in Estimation Accuracy Position Determination of Inertial Measurement Units

Ladislav Beran[a], Pavel Chmelar and Lubos Rejfek

Department of Electrical Engineering, Faculty of Electrical Engineering and Informatics, University of Pardubice,Pardubice, Czech Republic

Abstract. This paper deals with an increase in measurement accuracy of the Inertial Measurement Units (IMU). In the Inertial Navigation Systems (INS) a fusion of gyroscopes, accelerometers and in some cases magnetometers are typically used. The typical problem of cheap IMU is non-stationary offset and high level of noise. The next problem of IMU is a problem with a bumpy floor. For this case it is necessary to a have high quality chassis to eliminate additional noise. Also, it is possible to eliminate this noise by using some algorithm, but results are still poor. These properties lead to the inaccurate position estimation in the integration process. Even a small offset error leads to a big mistake in position determination and grows quickly with a time. This research is focused on the elimination of these poor properties and increase of accuracy of position estimation using Kalman Filtration.

1 Introduction

Precise navigation of mobile robots in unknown areas is common issue in much research. For the robot navigation in exterior the Global Navigation Satellite System (GNSS)is commonly used. For better results in outdoors is also possible to use the Satellite Based Augmentation Systems (SBAS). In Europe the European Geostationary Navigation Overlay Service (EGNOS)is usually used, while in US Wide Area Augmentation System (WAAS)is used. The benefit of using GNSS navigation systems is the achieved accuracy, typically better than 15 meters. Our research is focused on the autonomous navigation of a robotics platform in indoor spaces like ruins, mines, etc. The GNSS navigation system is not suitable for indoor areas because in the buildings navigation does not have a signal received from satellites or base stations. For these cases, it is necessary to develop an alternative navigation system.

The navigation of a robotics platform in indoor spaces, a modern navigation system called visual odometry can be used. This system is known from the eighties, but massive penetration is connected with fast Digital Signal Processors (DSP). This system is based on capturing a scene by the camera. The algorithm calculates interesting points from an actual and previous captured scene. The algorithm searches pairs of interesting points and afterwards calculate a distances between these pairs, called pixel distance. Also it is necessary to use some type of transformation to transform pixel distances to real coordinates. The distance estimation is performed using the common elevation angle of a sensing camera, the size of a sensing chip and focal length. This problem is described in detail in papers [1], [2], [3] and [4].

An alternative navigation system is called Inertial Navigation System (INS). This system uses accelerometers and gyroscopes. High level of noise and non-stationary offset is typical for cheap measurement units. These properties lead to the inaccurate acceleration estimation in the integration process. This is typical especially for low-cost accelerometers. Even a small offset error leads to a big mistake in position determination and grows quickly with time.Normally, the high level of noise is possible to eliminate using advanced digital filters [5], but in this case we need a special filter like Kalman filter. In our research we are focused on the eliminating a wrong position estimation using a Direct Cosine Matrix, Odometrymeasurement and Kalman filter [6]. The final device will be used as fusion of a navigation of robotics platform with an optical measurement device [7], [8], [9].

2 Rotation formalism

2.1 Direct cosine matrix (DCM)

The Direct Cosine Matrix provides transformation vectors in Euclidean space around an origin by an angle θ. In practical use it is necessary to have some rotation formalism to provide a rotation around three independent axes called *Yaw*, *Pitch*, *Roll* (in some cases called α, β and γ). These axes are shown in Figure 1.

Basic clockwise rotation around an x-axis by an angle γ in three dimension space can be written in a matrix form (1). Similar equations (2), (3) exists for an y-axis and an z-axis.

[a] Corresponding author: ladislav.beran@student.upce.cz

$$R_x(\gamma) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\gamma & -\sin\gamma \\ 0 & \sin\gamma & \cos\gamma \end{bmatrix} \quad (1)$$

$$R_y(\beta) = \begin{bmatrix} \cos\beta & 0 & \sin\beta \\ 0 & 1 & 0 \\ -\sin\beta & 0 & \cos\beta \end{bmatrix} \quad (2)$$

$$R_z(\alpha) = \begin{bmatrix} \cos\alpha & -\sin\alpha & 0 \\ \sin\alpha & \cos\alpha & 0. \\ 0 & 0 & 1 \end{bmatrix} \quad (3)$$

where R_n is the rotation matrix defined on an axis and α, β and γ are angles of clockwise rotation.

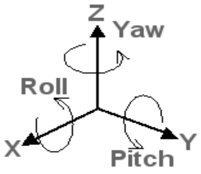

Figure 1. Euler angles.

A final three-dimension rotation matrix is created by a simple matrix multiplication (4).

$$\mathrm{R} = \mathrm{R_x}(\gamma)\mathrm{R_y}(\beta)\mathrm{R_z}(\alpha) \quad (4)$$

2.2 Position estimation correction using DCM

The correction of position estimation using DCM by simple matrix multiplication is provided using (4).

The corrected results using DCM transformation was presented in [1]. Precision of results was poor and it is necessary to increase precision to use this system for fusion with a higher precision navigation system based on visual odometry. In this time, we are testing a suitability of increasing anestimation accuracy position determination of IMU only on direct path. The DCM correction will be implemented to the final version of robotics platform navigation – navigation of robotics platform using a fusion of visual odometry, Inertial Measurement Unit and optical odometers at every wheel.

2.3 Position estimation correction using moving average filters

The moving average filters are usually used to eliminate the extreme deviations in a data series. The diagram of fifth order Moving Average Filter (MVA) is shown inFigure 2.

In our research we tried to use this type of filtration to eliminate extreme deviations in our measured data using IMU. The results of using this type of filtration are shown inFigure 6.

$$y_k = \sum_{i=-\frac{M-1}{2}}^{\frac{M-1}{2}} (A_i x_{k+i}) \quad (5)$$

where M is an order of filter, Ai are coefficients of filter, x_{k+i} are input data, y_k are filtered output data

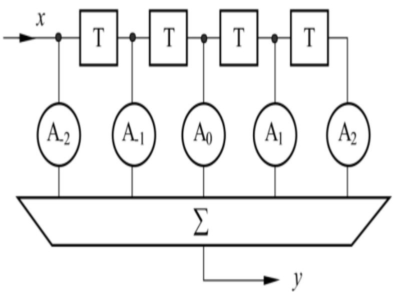

Figure 2. The moving average filter for fifth order [7].

3 Kalman filter and it´s Implementation to our system

The Kalman filter is a modern, effective and robust method for solving problems of dynamic systems with random signals. The implementation and using of Kalman Filter is described in papers [10], [11]. The Kalman filter is established using equation (6) and (7).

$$\boldsymbol{x}(t+1) = \boldsymbol{A} \cdot \boldsymbol{x}(t) + \boldsymbol{B} \cdot \boldsymbol{u}(t) + \boldsymbol{v}(t) \quad (6)$$

$$\boldsymbol{y}(t) = \boldsymbol{C} \cdot \boldsymbol{x}(t) + \boldsymbol{D} \cdot \boldsymbol{u}(t) + \boldsymbol{e}(t) \quad (7)$$

where ***x(t)*** is state vector in time *t*, ***y(t)*** is a vector with measurements in time *t*, ***u(t)***is thedrive state vector, ***v(t)*** is process noise vector, ***e(t)*** is the measurement noise vector, ***A,B,C,D*** are matrixes,which providesbehaviour of system.

In our case, a model of traveled distance we can establish using equation (8).

$$\mathrm{x(t)} = \mathrm{x_0} + \mathrm{v} \cdot \mathrm{t} + \frac{\mathrm{a \cdot t^2}}{2\cdot} \quad (8)$$

where *x(t)* is a distance in time, x_0 is an initial distance, *v* is a speed, *a* is an acceleration, *t* is a time.

Using equation (9) we can establish a time-dependent system of equations.

$$\begin{bmatrix} x_{m1} \\ x_{m2} \\ \vdots \\ x_{mN} \end{bmatrix} = \begin{bmatrix} x_0 \\ x_0 \\ \vdots \\ x_0 \end{bmatrix} + \begin{bmatrix} v \\ v \\ \vdots \\ v \end{bmatrix} \cdot \begin{bmatrix} t_1 \\ t_2 \\ \vdots \\ t_N \end{bmatrix} + \frac{a}{2} \cdot \begin{bmatrix} t_1^2 \\ t_2^2 \\ \vdots \\ t_N^2 \end{bmatrix}. \quad (9)$$

where $\boldsymbol{x_m}$ is a vector of measured results, $\boldsymbol{x_0}$ is a vector containing the initial distance (before measurement),***v*** is a vectorcontaining theinitialspeed (before measurement)

The vector $\mathrm{x_m}$ is a vector which contains measured distances according to time and it is reference to our system.

$$y_e(t) = y_e(t-1) + \frac{a(t)}{2} t^2 \quad (10)$$

where $y_e(t)$isan actual estimated path using acceleration from accelerometers, $y_e(t\text{-}1)$is a previous value of estimated path, a(t) is time-depend acceleration acquired from accelerometers.

In our case we can rewrite an equation (6) and (7) to easier form, because theinformation about the influence of the control signalshave not been assumed.

$$\boldsymbol{x}(t+1) = \boldsymbol{A} \cdot \boldsymbol{x}(t) + \boldsymbol{v}(t) \quad (11)$$

$$\boldsymbol{y}(t) = \boldsymbol{C} \cdot \boldsymbol{x}(t) + \boldsymbol{e}(t) \tag{12}$$

The behaviour of a system is defined by the simple state vector. This vector is created using measured distance and calculated speed.

$$\mathrm{x(t)} = \begin{bmatrix} \mathrm{x(t)} \\ \mathrm{s(t)} \end{bmatrix} \tag{13}$$

where *s(t)* is calculated speed - derivation a distance by time.

$$\mathrm{x(t+1)} = \begin{bmatrix} \mathrm{x(t+1)} \\ \mathrm{s(t+1)} \end{bmatrix} = \begin{bmatrix} \mathrm{x(t) + s(t) \cdot T} \\ \mathrm{s(t)} \end{bmatrix} =$$

$$= \begin{bmatrix} 1 & \mathrm{T} \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \mathrm{x(t)} \\ \mathrm{s(t)} \end{bmatrix} = \mathrm{A \cdot x(t)} \tag{14}$$

$$\mathrm{y(t)} = \begin{bmatrix} 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} \mathrm{x(t)} \\ \mathrm{s(t)} \end{bmatrix} = \mathrm{C \cdot x(t)} \tag{15}$$

The last variables are ***v(t)*** and ***e(t)***. The noise process vector ***v(t)*** have two elements because the state vector ***x(t)*** also have two elements (measured distance *x(t)* and it´s derivation–speed *s(t)*).

$$\mathrm{v(t)} = \begin{bmatrix} \mathrm{v_x(t)} \\ \mathrm{v_s(t)} \end{bmatrix} \tag{16}$$

$$\mathrm{e(t)} = \begin{bmatrix} \mathrm{e_y(t)} \end{bmatrix} \tag{17}$$

The Kalman filter usingVarianceMatrix of noise of process signed Q and Variance Matrix of noise of measurement signed R. The both matrixes are possible to establish using equations (16), (17), (18) and (19)

$$\mathrm{Q} = \mathrm{var}\big(\mathrm{v(t)}\big) = \begin{bmatrix} \mathrm{var(v_x)} & \mathrm{cov(v_x, v_S)} \\ \mathrm{cov(v_s, v_x)} & \mathrm{var(v_s)} \end{bmatrix} \tag{18}$$

$$\mathrm{R} = \mathrm{var}\big(\mathrm{e(t)}\big) = \begin{bmatrix} \mathrm{var(e_y)} \end{bmatrix} \tag{19}$$

The Kalman filter minimalizes the mean value of trace of the correlation matrix.

$$\mathrm{P(t)} = \mathrm{E}\left[\big(\mathrm{x(t)} - \hat{\mathrm{x}}\mathrm{(t)}\big) \cdot \big(\mathrm{x(t)} - \hat{\mathrm{x}}\mathrm{(t)}\big)^{\mathrm{T}}\right] \rightarrow \mathrm{MIN} \tag{20}$$

where *E[...]* is mean value, $\hat{x}(t)$ is an optimal result (the best estimation value of *x(t)*)

In the Kalman filter is recalculation step and predictive step. In the recalculation part, the estimations of the state vector and a correlation matrix of error are increasing to the time *t-1*.

$$\mathrm{K} = \mathrm{P(t-1) \cdot C^T \cdot (C \cdot P(t-1) \cdot C^T + R)^{-1}} \tag{21}$$

$$\mathrm{P(t)} = \mathrm{P(t-1) - K \cdot C \cdot P(t-1)} \tag{22}$$

$$\hat{\mathrm{x}}\mathrm{(t)} = \hat{\mathrm{x}}\mathrm{(t-1)} - \mathrm{K} \cdot (\mathrm{y(t)} - \mathrm{C} \cdot \hat{\mathrm{x}}\mathrm{(t-1)}) \tag{23}$$

$$\boldsymbol{P}(t+1) = \boldsymbol{A} \cdot \boldsymbol{P}(t) \cdot \boldsymbol{A}^T + \boldsymbol{Q} \tag{24}$$

$$\hat{\boldsymbol{x}}(t+1) = A \cdot \hat{x}(t) \tag{25}$$

Equations (21) to (25) provide a recursive solution of a model which is described by the equations (11) and (12).

The last step of The Kalman filter design is setting of initial values. The best estimated value $\hat{\mathrm{x}}\mathrm{(t)}$ is equal to zero, because the initial speed and distance is also equal to zero.

$$\hat{\mathrm{x}}(\mathrm{t}=0) = \begin{bmatrix} \mathrm{x(0)} \\ \mathrm{s(0)} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \tag{26}$$

4 Propossed approach

In our research we are using a wireless-controlled robotics platform. On this platform is fitted a cheap Inertial Measurement CHR-6Dand user-made optical odometer. The parameters of the optical odometer are shown at Table 1. In Table 2 the chassis parameters are shown.

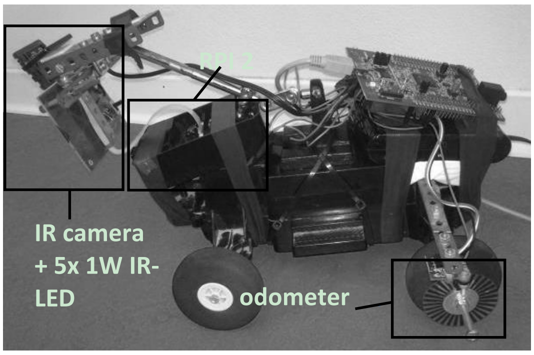

Figure 3. Our robotics platform.

Table 1. Optical encoder wheel parameters.

Optical encoder wheel parameters	
Parameter	**Value**
Outside Radius	60mm
Inner Radius	45mm
Pixel length (at outside radius)	2,94mm
no. of steps	64 steps
bits	1 bit

Table 2. Chassis parameters of our robotics platform.

Robotics chassis configuration	
Parameter	**value**
Nr. of Wheels	4
Wheel diameter	58 mm
Wheel circumference	182,21 mm
Chassis type	Automobile type - controllable front part

The IMU is based on accelerometer ADXL335 and gyroscopes LP510AL for roll and pitch axes and LY510ALH for yaw axis. All parameters of IMU are written in Table 3.The odometer is based on an infrared light barrier and a 64 step encoder wheel. The parameters of optical wheel are written inTable 1. The wheel rotation is captured by a STM32F4Discovery development kit. The measured data from the odometer is transferred to the Raspberry PI 2 via a serial link every 0.25 seconds. In this time, data from odometer and IMU are transferred via a wired connection, but in near future this connection will be upgraded to the Bluetooth connection. In this time, the Raspberry Pi 2 is used for recording data stream. Then, the captured data is processed in Matlab software. In the future, all algorithms will be implemented at this platform. In Figure 3 our robotics platform is shown.

Table 3. Inertial measurement unit parameters.

IMU POLOLU CHR-6D	
Parameter	**value**
Gyroscopes LP510AL and LY510 ALH	300°/full scale
ADXL335	16bit, ±3g
	300mV/g

5 Position estimation

The position estimation of the path using IMU is calculated by equations (27) and (28).

$$\mathrm{v(t)} = \mathrm{v(t-1)} + \mathrm{a(t)} \cdot \mathrm{Ts} \tag{27}$$

$$\mathrm{s_a(t)} = \mathrm{s_a(t-1)} + \mathrm{v(t)} \cdot \mathrm{Ts} \tag{28}$$

where $S_a(t)$ is the calculated distance using data from accelerometers, *v(t)* is the calculated speed, *a(t)* is measured acceleration, *Ts* is sampling frequency.

The position estimation of path using optical odometer is calculated by equation (28).

$$\mathrm{s_o(t)} = \mathrm{s(t-1)} + \mathrm{c(t)} \cdot \frac{\mathrm{W_c}}{\mathrm{N}} \tag{29}$$

where *so(t)* is calculated distance using data from the odometer, *c(t)* is count of impulses measured by odometer in defined time, *Wc* is a wheel circumference (in our case 182,21 mm), *N* is a number of impulses(steps) in one rotation of a wheel (in our case 64 steps).

The maximal error caused by resolution of odometer wheel can be calculated using equation (30) and (31).

$$\alpha_l = \frac{360°}{\mathrm{N}} \tag{30}$$

$$\mathrm{e_o} = \pm\frac{\alpha_l}{2} \cdot \frac{\mathrm{W_c}}{360} \tag{31}$$

where e_o is maximal error caused by the resolution of the odometer wheel; α_l is an angle covered in one step; *Wc* is wheel circumference, *N* is a number of steps on the odometer wheel (in our case 64 steps).

6 Results and discussion

At present, in our solution we are using post-processing calculations in Matlab software, but in future all calculations will be implemented into a powerful microcontroller, like Raspberry Pi 2 or equal.The captured data is calculated in pair in Matlab software. The first data array contains a number of counted impulses from the odometer which depends on time; the second data array contains acceleration in depends on time; the third data array contains angular velocity also in depends on time. In this time we are using only a first and second data array for position estimation. The third data array (measured angular velocities) will be used in the following research.

At the Figure 4, the original measured data is shown. This measured data contains a lot of noise caused by the bumpy floor. For this case it is necessary to use some type of filtration. The result using a moving average filter of the tenth order is shown in Figure 6. Using of digital filters is described in detail in [7].

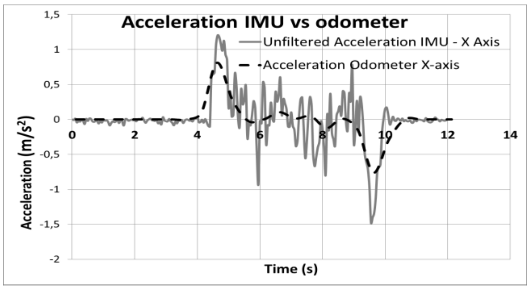

Figure 4. Original acceleration measured using inertial measurement unit vs calculated from odometer.

In Table 4 and Table 5, estimated path length using an odometer and an IMU is shown. In the table results calculated from filtered data using a moving average filter is shown. For better readability, the first five measurements from fifteen are shown in the tables. Totally, each path had fifteen repetitions. The statistical values are calculated from all measurements.

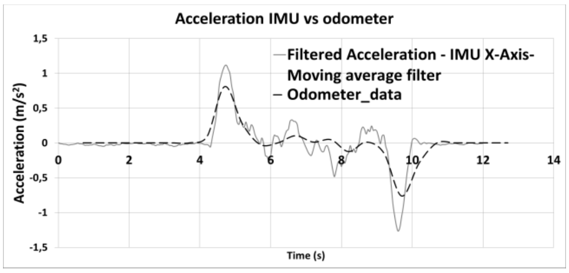

Figure 5. Filtered acceleration measured by IMU using Moving average filter.

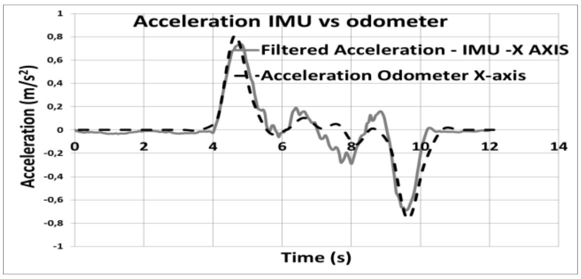

Figure 6. Filtered acceleration measured by IMU using Kalman filter.

At the Figure 6, the filtered data using Kalman filter is shown.

In Table 6 the filtered data using MVA is shown. InTable 7, the filtered data using the Kalman Filter is shown. From the tables it is possible to find that, the Kalman filter rapidly increase a precision compared with unfiltered data. The results using MVA has better precision than unfiltered data but worse than parameters in case of using the Kalman Filter.

Table 4. Estimated path length using odometer.

Real Path length (m)	Estimated Path Length Using Odometer						
	Measure number					Aver. (m)	STD. DEV (m)
	1	2	3	4	5		
3	2,99	3,02	2,98	3,01	3,00	3,005	0,013
4	4,08	3,96	3,98	4,01	4,02	3,992	0,041
5	5,09	5,02	5,03	4,94	5,02	5,070	0,061
10	10,01	9,93	10,15	9,98	10,10	10,041	0,080

Table 5. Estimated path length using IMU.

Real Path length (m)	Estimated Path Length Using IMU						
	Measure number					Aver. (m)	STD. DEV (m)
	1	2	3	4	5		
3	3,25	3,15	2,91	3,25	3,09	3,1268	0,108
4	4,28	3,95	3,95	4,01	3,90	4,135	0,118
5	5,19	5,27	5,16	4,91	4,92	5,219	0,165
10	10,58	10,25	10,36	9,51	10,13	10,391	0,476

From Table 4 and Table 5 it is possible to recognize that, the odometer has more precise measurement than IMU.

Table 6. An estimated path using IMU – MVA.

Real Path length (m)	Estimated Path Length Using IMU – MVA						
	Measure number					Aver. (m)	STD. DEV (m)
	1	2	3	4	5		
3	3,12	3,11	2,95	3,14	3,09	3,0722	0,078
4	4,17	3,94	3,93	3,95	3,90	4,085	0,082
5	5,12	5,18	5,07	4,94	4,94	5,180	0,127
10	10,31	10,17	10,28	9,79	10,10	10,173	0,131

Table 7. An estimated path using IMU - Kalman filter.

Real Path length (m)	Estimated Path Length Using IMU – Kalman Filter						
	Measure number					Aver. (m)	STD. DEV (m)
	1	2	3	4	5		
3	3,07	3,09	2,97	3,11	3,05	3,051	0,053
4	4,1	3,99	3,97	4,00	3,97	4,078	0,098
5	5,09	5,16	5,12	4,95	4,94	5,091	0,117
10	10,27	10,17	10,20	9,81	10,09	10,191	0,228

7 Conclusion

In this paper an increasing estimation accuracy position determination of Inertial Measurement Units was described. This paper is focused on the implementation of the Kalman Filter to our robotics platform. Firstly this research used an alternative to Kalman filtration – Average Moving Filter (AMF). Results were better compared with unfiltered data, but precision was still poor. For this case it is necessary to implement predictive filtration -Kalman filter.

The results recorded using the Kalman filter has very well precision. The main problem of the Kalman filter is that, it is necessary to have a precision reference signal – for this case we are using an Odometer.

The odometer has the best precision.The disadvantage of the odometer is that, measuring only path length (impulses) and does not care about curves etc. For this case, it is necessary to have gyroscopes for recording a rotation of robotics platform. Also a big disadvantage of the odometer is wheel slip. In case of a slippery floor, the odometer counts impulses, but the position of robotics platform is the same. The disadvantage of IMU is a big noise in measured signal and lower precision than the odometer. The IMU have problems with additional noise and have a lower precision than the odometer.

Also big disadvantageof IMU is a problem with bumpy floor. In case of bumpy floor, it is impossible to measure acceleration and rotation using IMU without high quality chassis. To eliminating this problem we are using a fusion of Odometer and IMU.

In this time, we are testing suitability of our theory only on a direct path. The DCM correction will be implemented to final version of a robotics platform – navigation of robotics platform using a fusion of visual odometry, Inertial Measurement Unit and mechanical odometers at every wheel. A very big advantage of our visual odometry is precision of position estimation [12]. The final device will be used as a fusion of navigation of the robotics platformwith an optical measurement device to create a 3D space map of dangerous spaces like a ruins, abandoned mines etc. [7], [8], [9]

Acknowledgment

The research was supported by the Internal Grant Agency of University of Pardubice, the project No. SGSFEI_2016_022.

References

1. L. Beran, P. Chmelar, and M. Dobrovolny, Navigation of robotic platform with using Inertial Measurement Unit and Direct Cosine Matrix, 56thInternation-al Symposium Electronics in Marine(Elmar), **56**, 87-90 (2014)
2. L. Beran, P. Chmelar, and L. Rejfek, Navigation of Robotics Platform using Monocular Visual Odometry, 25th International Conference Radioelektronika, **25**, 213-216 (2015)

3. J. Campbell et.al, A Robust Visual Odometry and Precipice Detection System Using Consumer-grade Monocular Vision, Robotics and Automation. ICRA 2005, 3421-3427, (2005)
4. CH. Yang, M. Maimone, and L. Matthies, "Visual Odometry on the Mars Exploration Rovers," Systems, Man and Cybernetics, **1**, 903 (2005)
5. L. Rejfek,Z. Mosna, D. Kouba, J. Boska, and D. Buresova, Application of digital filters to check quality of the automatically scaled ionograms, Journal of Electrical Engineering, ISSN: 1335-3632, **66**, 3 164-168 (2015)
6. J. Borenstein and L. Feng , Measurement and Correction of Systematic Odometry Errors in Mobile Robots, IEEE Transactions on Robotics and Automation, **12**, 6 869-880, (1996)
7. P. Chmelar, L. Beran, N. Kudriavtseva , The Laser Color Detection for 3D Range Scanning Using Gaussian Mixture Model, 25th International Conference Radioelektronika 2015, **25**, 248 253 (2015)
8. P. ChmelarAnd M. Dobrovolny, The Fusion of Ultrasonic and Optical Measurement Devices for Autonomous Mapping. 2013, 23rd International Conference Radioelektronika, **23**, 292-296 (2013)
9. P. Chmelar, L. Beran, N. Kudriavtseva., "Projection of point cloud for basic object detection," 56th International Symposium (ELMAR 2014), **56**, 11-14 (2014)
10. B. Gersdorf and U. Frese, A Kalman Filter for Odometry using a Wheel Mounted Inertial Sensor. 10th International Conference on Informatics in Control, Automation and Robotics, **10**, (2013)
11. J.Z. Sasiadek and P. Hartana, Sensor data fusion using Kalman filter, in, FUSION 2000. Proceedings of the Third International Conference, **3**, (2000)
12. L. Beran, P. Chmelar, and L. Rejfek, Navigation of Robotics Platform Using Advanced Image Processing Navigation Methods, V. Eccomas Thematic Conference on Computional Vision and Medical Image Processing(VIPIMAGE), **5**, 341-346 (2015)