Localization of the Wheeled Mobile Robot based on Multi-Sensor Data Fusion

Submitted: 4 March 2015; accepted 26th May 2015

Piotr Jaroszek, Maciej Trojnacki

DOI: 10.14313/JAMRIS_3-2015/26

Abstract:
The paper presents a method of localization of a mobile robot which relies on aggregation of data from several sensors. A review of the state of the art regarding methods of localization of ground mobile robots is presented. An overview of design of the four-wheeled mobile robot used for the research is given. The way of representation of robot environment in the form of maps is described. The localization algorithm which uses the Monte Carlo localization method is described. The simulation environment and results of simulation investigations are discussed. The measurement and control equipment of the robot is described and the obtained results of experimental investigations are presented. The obtained results of simulation and experimental investigations confirm the validity of the developed robot localization method. They are the foundation of further research, where additional sensors supporting the localization process could be used.

Keywords: *wheeled mobile robot, localization, environment map, laser scanner, Inertial Measurement Unit, odometry, data fusion, Monte Carlo localization method*

1. Introduction

One of the elements necessary for autonomous realization of robot motion is its localization in the global reference system [1]. The problem of robot localization is widely studied in scientific publications [2]–[4], because of its key significance. The knowledge of position and orientation of an autonomous robot is fundamental for its proper and reliable operation.

Artificial intelligence algorithms based on which the control is determined, must have the largest possible knowledge of the state in which the controlled object currently is. For many such algorithms, correct localization is essential, as for example in the global path planning problem [5] or the local motion planning problem [6].

Robot localization can be considered from various points of view, the most straightforward of which is the position tracking [7]. In this situation, the initial position and orientation of the robot are known and the only objective is determination of position in particular time instants based for example on robot odometry. Simultaneously, compensation for errors which accumulate in time is required. Another, more difficult, problem is the global localization problem [8], where robot initial pose is not known. The localization process must therefore deduce the initial robot state and then determine the correct position and orientation during its motion.

The most difficult of localization problems is the kidnapped robot problem [9], where in addition to the problems introduced by the global localization, the robot state can suddenly change in a way unrelated to robot actions. One may say that the robot has been "kidnapped" during operation and placed in some other unknown place.

For execution of the robot localization task, the probabilistic methods are widely applied [10]–[13]. It is due to the fact that all information gathered by robot sensor systems is burdened with some level of error. Minimization of influence of those errors on the localization accuracy can be achieved with probabilistic methods, whose task is to create the representation of probability that describes the robot localization in the best way. Based on the chosen probability function representation, the estimation of position and orientation is carried out.

The probabilistic methods are most often based on the recursive Bayesian filter principle, which is the principal tool in the task of estimation of the unknown probability distribution function. This task is made recursively in time using both measurements from sensors and information about actions undertaken by the robot. The task is executed in two basic phases: prediction phase and update (correction) phase.

The inherent element of the prediction phase is the robot motion model, by means of which the transformation of the probability function before and after single movement step is described. Most often this model is based on modification of the Gaussian distribution after prediction.

In turn, update of position and orientation of the robot is made based on measurements from the robot sensors. By means of the adopted observation model, update of the probability distribution function takes place by minimization of uncertainty caused by errors accumulated in the preceding steps.

The best known methods of probabilistic localization are the Kalman filter methods [12], Markov localization [14] and particle filters [15]. They differ mainly in terms of ways of representation of the probability distribution function, and each one of them has certain advantages and drawbacks as compared to others.

One of the newest methods based on the probabilistic concept is the Simultaneous Localization and Mapping (SLAM) [16]. This task is difficult because of

the nature of the problem, since the map is required for localization and conversely the information about localization is required to construct the map.

A key to efficient localization is also the appropriate model of the robot environment. The most often used solution is the introduction of the map defined adequately to the given problem.

The amount of information delivered by robot sensors can directly affect the effectiveness of its localization. Taking into account the presence of noise in sensor signals, uncertainties and limited applicability of sensors themselves, it is beneficial to apply a larger number of different types of sensors. The data fusion is a tool that allows to combine knowledge from various sources to maximize their usefulness [17]. In this way, in the considered cases it is possible to improve the quality of the results, by simultaneously allowing smaller amount or worse quality of the required input data.

The aim of the present work is robot global localization on the known environment map, based on the aggregation of sensor data. For the task of localization robot odometry (encoders), inertial sensors and the laser scanner will be used. The proposed localization algorithm will use the Monte Carlo localization method and the hybrid representation of the environment.

2. Four-wheeled Mobile Robot

The object of research, on which the developed localization method will be investigated, is the PIAP SCOUT mobile robot [18].

In Fig. 1a the commercial version of the robot is shown. It was designed for quick reconnaissance of places with difficult access, i.e., vehicle chassis, places under seats in means of transportation, narrow rooms and ventilation ducts. The robot is manufactured in various versions, differing mainly in type of equipment installed on-board, which makes it suitable for specialized tasks.

The robot locomotion system is hybrid. It consists of tracks and non-steered wheels which operate simultaneously. Two rear wheels are driven independently by DC motors equipped with gear units and encoders, which enable measurement of angular velocity of spin of the driving wheels. The drive from the rear wheels is transmitted to the front wheels via two toothed belts or tracks.

a) b)

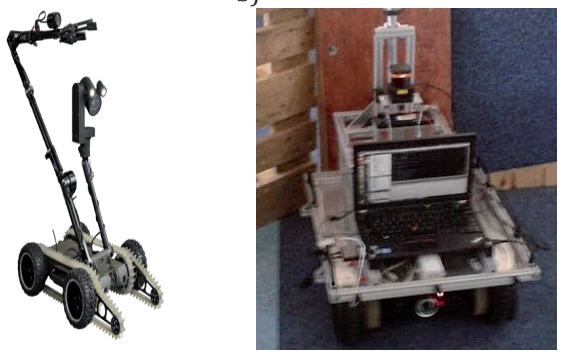


Fig. 1. PIAP SCOUT wheeled mobile robot: a – commercial version [18], b – version adapted to experimental investigations

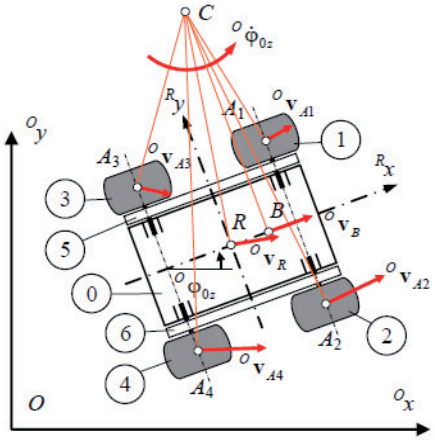


Fig. 2. Kinematic structure of the robot

During investigations the manipulator, camera and auxiliary front tracks were dismantled. The robot has been adapted to realization of experimental research by installing additional frame to mount the necessary equipment (Fig. 1b).

On the frame were installed: the laptop computer, the Inertial Measurement Unit and the 2D laser scanner.

The kinematic structure of the robot is illustrated in Fig. 2 [19].

The following basic robot components can be distinguished: 0 – mobile platform (robot body with additional frame), 1-4 – wheels, 5-6 – tracks. The mobile platform of the robot is approximately 0.5 m by 0.5 m (length x width), and its mass is about 22 kg.

Because the robot is equipped with non-steered wheels, large wheel slips occur during its turning (particularly large during pivot turning). For this reason, in a general case it is not possible to determine robot motion parameters based only on the encoder data. As a result the robot localization is much more difficult.

As far as the robot localization is concerned, actual robot state vector has the form:

$$\mathbf{x} = [{}^{O}x_R,\ {}^{O}y_R,\ {}^{O}\varphi_{0z}]^T, \tag{1}$$

where: ${}^{O}x_R$, ${}^{O}y_R$ and ${}^{O}\varphi_{0z}$ are respectively actual coordinates of the point R of the robot and its heading in the global (stationary) reference system $\{O\}$.

By analogy, the prediction of the robot state vector can be written as:

$$\hat{\mathbf{x}} = [{}^{O}\hat{x}_R,\ {}^{O}\hat{y}_R,\ {}^{O}\hat{\varphi}_{0z}]^T, \tag{2}$$

where elements of this vector are counterparts of the elements of the $\mathbf{x}$ vector.

In turn, the error of the robot state vector will be denoted as:

$$\tilde{\mathbf{x}} = [{}^{O}\tilde{x}_R,\ {}^{O}\tilde{y}_R,\ {}^{O}\tilde{\varphi}_{0z}]^T = \mathbf{x} - \hat{\mathbf{x}}, \tag{3}$$

where the error of robot position (Euclidean norm) and error of heading will be respectively equal to ${}^{O}\tilde{r}_R$ and ${}^{O}\tilde{\varphi}_{0z}$, and

$$ {}^{O}\tilde{r}_R = \sqrt{({}^{O}\tilde{x}_R)^2 + ({}^{O}\tilde{y}_R)^2} . \quad (4) $$

The space of possible robot states is represented by the collection of particles **S** of which every *i*-th particle is described by the state vector:

$$ \mathbf{x}_i = [{}^{O}x_{Ri}, {}^{O}y_{Ri}, {}^{O}\varphi_{0zi}]^T, \quad (5) $$

where: ${}^{O}x_{Ri}$, ${}^{O}y_{Ri}$, ${}^{O}\varphi_{0zi}$ are respectively possible coordinates of the point *R* and its heading in the global coordinate system {*O*} for the *i*-th particle.

In turn, the prediction of the state vector of the *i*-th particle in the successive step of the algorithm has the form:

$$ \hat{\mathbf{x}}_i = [{}^{O}\hat{x}_{Ri}, {}^{O}\hat{y}_{Ri}, {}^{O}\hat{\varphi}_{0zi}]^T, \quad (6) $$

where the elements of the vector are the counterparts of the vector $\mathbf{x}_i$.

3. Environment Representation

Robot environment can be [20]: observable where both map and robot in each time instant can be uniquely defined, or partly observable, where neither map nor robot can be uniquely determined in each time instant. Partial non-observability makes localization task more difficult, because significant increase of robot admissible state space is necessary in this case.

The environment can be also [20]: deterministic, where result of the actions performed by the robot is certain and completely defined or stochastic, where result of actions performed by the robot is uncertain. For stochastic environments additional probabilistic coefficients have to be introduced.

Additionally, robot environment can be [20]: static, where the map of environment does not change during robot operation, or dynamic, where the environment map can change. Dynamic environments are characterized by the presence of moving obstacles, which must be properly recognized and appropriate actions must be undertaken to check their influence on localization.

In the present work the environment has hybrid representation in the form of the 2-dimensional static deterministic and fully observable map. It consists of two layers, the first of which is the map of features. In this layer, the walls (obstacles) are represented by segments of known start and end coordinates in the global coordinate system {*O*}. The second layer has the form of grid occupancy map. The algorithm uses one of the two layers (environment representations) in the way so as to minimize the time of computations. In case of parts of the environment that consist of straight segments, the algorithm uses the map of features, and in case of complex shapes of the environment, the grid occupancy map is used.

For the purpose of simulation research, the first layer of the map is created in AutoCAD and then it is converted to the .bmp format in order to obtain the second layer of the map.

In the case of experimental research, the spatial scanning technique using the 2D laser scanner mounted on the robot is used to obtain the second map layer, and after subsequent conversion of the point cloud into lines, the first layer is obtained.

4. Localization Based on Multi-sensor Data Fusion

The theme of the work is wheeled mobile robot localization using data aggregation from the sensors. To this end, the probabilistic method based on the particle system, that is Monte Carlo localization, will be used. It relies on recursive algorithm which uses the Bayesian theory for posteriori estimation of the distribution.

Main features of the Monte Carlo localization are:

- capability of using independent sensor data from multiple devices, also burdened with large errors,
- flexibility of adaptation of the number of particles and map complexity to the computing power of the computer,
- the estimate of robot current state is represented by a multimodal probability density function, which enables global localization,
- total computation cost of the algorithm is concentrated around places where the largest probability of robot occurrence exists.

The Monte Carlo algorithm consists of 3 main phases:

- prediction,
- update of weights,
- resampling.

The phases are preceded by the initialization process, which generates *M* particles on the map and assigns an identical normalized weight $\bar{w}_i$ to each *i*-th particle.

In the prediction phase, the algorithm predicts the $\hat{\mathrm{x}}_i$ state for each *i*-th particle from the set **S** using the motion model.

In the subsequent weight update phase, the weight w_i is calculated for each *i*-th particle from the **S** set. After calculation of weights, the process of their normalization takes place, that is, determination of weights $\bar{w}_i$ such that the sum of weights of all particles is equal to 1.

In the last phase, resampling of the **S** set takes place. Particles from the **S** set are drawn with replacement proportionally to the value of the weights (particles with large weights are chosen more often) and put into new **S'** set. Analogy to the survival of the fittest is noticeable. Particles with large weights are chosen more frequently so in the map concentrations are created where larger probability of robot position exists. After sampling is completed, the **S'** set becomes the **S** set for the next calculation loop. In this phase, algorithm of reduction of the number of particles based on the Effective Sample Size (ESS) coefficient can be additionally introduced [21]. In the case, when nearly whole population of particles will be close to the unknown robot state **x**, this coefficient can be used for gradual removal of particles from the remaining region of the map.

Process of robot localization was carried out on PIAP premises using the available mobile platform and the following sensors:

- HOKUYO UTM-30LX 2D laser scanner and environment map,
- STM iNEMOv2 sensors module, of which the accelerometer was used,
- encoders mounted on the shafts of robot drives.

The laser scanner with the environment map has a primary function in the process of determination of weights of the particles in the Monte Carlo localization method by comparing the real readouts from selected laser beams with virtual measurements of each of the particles.

The measurements using accelerometer have a corrective function in cases of robot motion with wheel slip.

Measurement from encoders is in turn used in the phase of prediction, in the robot motion model.

The iterative process of the method takes place in parallel, on two planes:

- robot control with sensor data reading and
- execution of localization process using the Monte Carlo method.

The robot control and sensor data reading plane is responsible for providing information in the form of distance, speed and accelerometer readouts. During experiment the operator moves the robot by tilting the joystick axis. Sensor readouts are made by means of programmatic real-time layers.

The amount of particles in the Monte Carlo localization method was experimentally chosen so as to ensure quick and accurate localization in the given environment. One of available resampling algorithms was implemented and a module for optimization of speed of algorithm execution was added.

4.1. Prediction Phase – Robot Motion Model

The motion model is used for prediction of the $\hat{\mathbf{x}}_i$ state vector for each *i*-th particle depending on the previous estimated state vector $\mathbf{x}_i$ and the action an identical for all particles. In the work, for prediction of robot motion speed its odometry and accelerometer were used.

From robot odometry, based on known angular speeds of spin of the driven wheels $\dot{\theta}_3, \dot{\theta}_4$ obtained based on the encoder indications, are calculated: robot longitudinal velocity ${}^R v_{ex}$ and angular velocity of turning ${}^R\omega_e$ and the transverse velocity is assumed ${}^R v_{ey} = 0$, where all those velocities are expressed in the moving coordinate system {*R*} associated with the robot. Those velocities, on the assumption of zero wheel slip, are determined from the following relationships [19]:

$$ {}^R v_{ex} = (\dot{\theta}_3 + \dot{\theta}_4) r/2, \; {}^R v_{ey} = 0, \tag{7} $$

$$ {}^R \omega_e = (\dot{\theta}_4 - \dot{\theta}_3) r/W, \tag{8} $$

where: *r* – geometric radius of wheel (the same for all wheels), *W* – wheel track (distance between geometric centers of left- and right-hand side wheels).

Because of existence of large wheel slips during robot turning, which was pointed out, for instance, in the work [19], those relationships are then burdened with large errors, but thanks to use of the Monte Carlo localization method, despite those errors localization of the robot is possible with relatively high accuracy.

Robot motion velocities can be also determined based on the ${}^R a_{ax}$ and ${}^R a_{ay}$ accelerations read from accelerometer, based on the following relationships:

$$ {}^R v_{ax}(t) = {}^R v_{ax}(t-\Delta t) + \left({}^R a_{ax}(t) - {}^R a_{ax}(t-\Delta t) \right) \Delta t \tag{9} $$

$$ {}^R v_{ay}(t) = {}^R v_{ay}(t-\Delta t) + \left({}^R a_{ay}(t) - {}^R a_{ay}(t-\Delta t) \right) \Delta t \tag{10} $$

where *t* denotes time instant of prediction, and Δt is the step time.

Similarly as previously, it is assumed that the ground on which the robot moves is horizontal, so the influence of the gravitational acceleration on the accelerometer indications is neglected. Additionally, the centripetal (normal) acceleration occurring during robot turning and Coriolis acceleration associated with Earth rotation (which is small) are not taken into account.

Next, the robot longitudinal velocity is calculated as a fusion of longitudinal velocities obtained from robot odometry and accelerometer according to the formula:

$$ {}^R v_{fx} = \mu_e \, {}^R v_{ex} + \mu_a \, {}^R v_{ax}, \tag{11} $$

where the following weights were assumed: $\mu_e = 0.7$, $\mu_a = 0.3$.

In turn, robot motion velocities in the global reference system, on the assumption of robot motion on a horizontal ground, can be determined based on the relationship:

$$ {}^O\mathbf{v} = \begin{bmatrix} {}^O v_{Rx} \\ {}^O v_{Ry} \\ {}^O \omega_0 \end{bmatrix} = \begin{bmatrix} {}^R v_{fx} \cos({}^O\varphi_{0z}) - {}^R v_{ay} \sin({}^O\varphi_{0z}) \\ {}^R v_{fx} \sin({}^O\varphi_{0z}) + {}^R v_{ay} \cos({}^O\varphi_{0z}) \\ {}^R \omega_e \end{bmatrix}. \tag{12} $$

Based on it, from the robot motion model prediction of the state vector of *i*-th particle $\hat{\mathbf{x}}_i$ is determined on the basis of the relationship [8]:

$$ \hat{\mathbf{x}}_i = \mathbf{x}_i + {}^O\mathbf{v} \, \Delta t + \mathbf{x}_{ni} \tag{13} $$

where $\mathbf{x}_{ni}$ is declared noise (uncertainty) of motion, which is determined in each iteration for every particle separately.

The noise is defined by standard deviations σ_{trans} and σ_{drift}, respectively for longitudinal and lateral directions of robot motion.

The motion model described above is applied for each *i*-th particle from the **S** set.

4.2. Weight Update Phase

To each *i*-th particle from the **S** set must be assigned appropriate weight w_i, proportional to the level of probability that the given measurement is equal to the real value. The particles, from the point of view of which the measurement is close to real values should be given larger weights than those, whose measurements differ significantly.

To this end, for the laser scanner the sensor model based on the Gauss distribution is used, that is:

$$w_i = \prod_{k=0}^{n}\left(\frac{1}{\sqrt{2\pi}\sigma_0}\exp\left(\frac{-(o_{laser,k}-o_{p,k,i})^2}{2\sigma_0^2}\right)\right) \quad (14)$$

where: $o_{laser,k}$ – measurement from the k-th laser scanner beam, $k = 0, ..., n$, $o_{p,k,i}$ – indication "from the point of view" of the i-th particle, σ_0^2 – experimentally chosen variance describing the sensor noise.

In case of the laser scanner, one obtains measurements from n equally spaced beams. Therefore each of the particles also contains n indications.

As far as the variance σ_0^2 is concerned, its small value results in narrow Gauss function, so only the particles of measurements very close to real obtain significant weights. For large values of σ_0^2, this distribution is wider, so the differences between weights are smaller.

After calculation of the weights of all particles, additional normalization of weights is necessary. Without it, the distribution is not the probability distribution. To this end, all weights are divided by the following normalization constant:

$$\vartheta = \sum_{i=0}^{M} w_i \quad (15)$$

This constant is the sum of all original weights. Finally, weights of particles are given values according to the following formula:

$$\overline{w}_i = \frac{w_i}{\vartheta}. \quad (16)$$

4.3. Resampling Phase

Resampling consists in selection of particles from the **S** set proportionally to their weights and creation of the new set **S'**, which will be used in the successive iteration. It is the so called draw with replacement, which means that elements not selected (because of small weight) will be replaced with elements with higher weights. This selection results in keeping in the memory only those places, which are probable from the point of view of localization. After performing the estimation of robot localization, the created **S'** set is exchanged with the **S** set for successive iteration.

Next point of iteration is appropriate estimation of robot position based on the cloud of points. To this end, the following two principal methods are used:

- maximum weight of the particle and
- superposition proportional to weights of particles.

The first method is trivial, since the particle with the largest weight is assumed as the position of the robot (the result of the localization). This solution has the disadvantage of "jumping" behavior in successive steps of the iteration.

The second method relies on estimation of position using both weight and position of each particle. Keeping in mind that sum of weights of all particles is equal to one; it is possible to carry out superposition of all coordinates and orientations of particles proportionally to their weights, using the following relationship:

$$\hat{\mathbf{x}} = \left[\sum_{i=0}^{M}({}^{O}\hat{x}_{Ri}\overline{w}_i), \sum_{i=0}^{M}({}^{O}\hat{y}_{Ri}\overline{w}_i), \sum_{i=0}^{M}({}^{O}\hat{\varphi}_{0zi}\overline{w}_i)\right]^T \quad (17)$$

where $\hat{\mathbf{x}} = \left[{}^{O}\hat{x}_R, {}^{O}\hat{y}_R, {}^{O}\hat{\varphi}_{0z}\right]^T$ is the eventual prediction of the robot state vector, obtained as a result of data fusion.

Next, exchange of the **S** and **S'** sets is performed, and execution of the successive iteration in the next time step.

Finally, optimization of the number of particles is done, which allows reduction of time of computation in successive iterations.

Because of lack of information on the initial state of the robot (global localization), the Monte Carlo localization requires a large number of particles only at the very beginning of the process. For the task without the problem of the kidnapped robot, keeping large number of particles is less important with time, because the localization problem smoothly transitions to the problem of position tracking. Therefore, it is possible to reduce the number of particles then.

To this end, the mechanism reducing the number of particles, when the algorithm discovers that the problem of tracking is more pronounced than the problem of localization, has been introduced. The measuring quantity ESS (Effective Sample Size) was introduced, which informs about the level of closeness of all weights of particles to each other, which is calculated based on the relationship [21]:

$$ESS = \left(\sum_{i=1}^{M}\overline{w}_i\right)^2 / \sum_{i=1}^{M}\overline{w}_i^2. \quad (18)$$

It means that at the time instant, when most of the particles will be concentrated in one place, their number will be reduced.

Reduction of the number of particles by the imposed percentage η takes place at the moment of satisfaction of the condition:

$$ESS > \tau M, \quad (19)$$

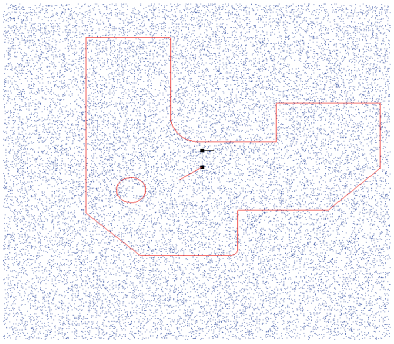
where: $\tau \in (0, 1)$ is the experimentally chosen index of rate of reduction of the number of particles (0 – very high rate of reduction, 1 – no reduction).

5. Simulation Research

Within the present work multiple investigations in the prepared simulation environment were carried out. On the ideal map (Fig. 3a) created in two variants, by means of lines and in the occupancy map, a series of localization processes was conducted for the simulated robot. In the figure, the map contour is marked red and particular particles – blue.

The developed application was also able to show simulated beams of distance readouts from chosen particles in the form of green lines (Fig. 3b).

a)



b)

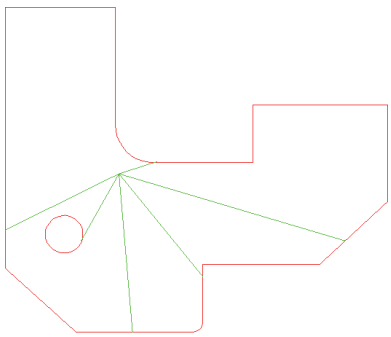


Fig. 3. Robot environment map: a –visible particles, b – example of simulated beams

In the simulation research, the following properties were checked:

- algorithm speed for various number of particles *M* (from 2000 to 20 000) and number of lines constructing the map (70 and 140),
- algorithm speed for various number of particles *M* (from 2000 to 20 000) and length of the radius in the case of the bitmap of 1000 x 1000 pixels,
- accuracy of localization in the 25-th step of iteration for identical sequence of robot motion (for initial number of particles $M = 10\,000$ and $M = 20\,000$),
- influence of value of variance of sensor noise σ_0^2 on the localization process.

In the simulation, robot position and its error in the map coordinate system, and also all performed simulations of distance measurement were represented in generic units.

Additionally, in every trial the initial pose of the robot and its control were the same.

In Fig. 4 simulation results were presented where time of simulation *T* dependency on the number of particles *M*, for two methods of representation of environment of the robot and for two chosen levels of complexity of those maps, was investigated. The vector map was investigated for the case of 70 and 140 lines, whereas the raster map of the size 1000 x 1000 units for maximum length of the beam equal to respectively 500 and 1000 units.

From the obtained results it follows that the time of computation decreases more or less linearly as a function of the number of particles used for robot localization. The time necessary for calculation of the algorithm for the map based on lines can be noticeably smaller. However, geometric complexity of the environment can cause significant increase of the number of lines necessary for map construction, which is not a problem when using a discrete map of occupation based on a bitmap. It can be also noticed that for small numbers of particles, the level of complexity of the map becomes unimportant. This information is, however, of little use in practical applications, where the common trend is to increase the number of particles.

In the following figures, results of simulations of robot localization are shown for various numbers of particles *M* and various values of variance of sensor noise σ_0^2 as a function of the iteration step.

On the basis of the obtained results, it is possible to evaluate the accuracy of robot localization depending on the used number of particles *M* and the value of σ_0, which is one of the method settings and has large influence on the results.

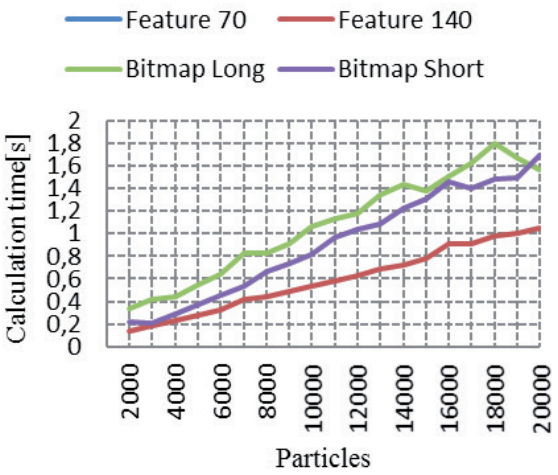


Fig. 4. Dependency of time of computation on the number of particles for two levels of complexity of the map

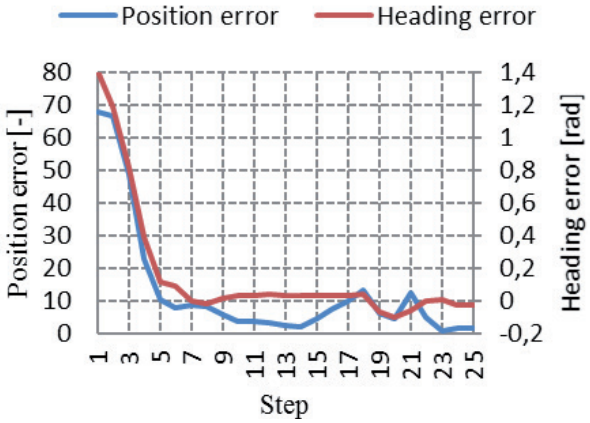


Fig. 5. Position ${}^{O}\tilde{r}_R$ and heading ${}^{O}\tilde{\varphi}_{Oz}$ errors for $M = 10\,000$ and $\sigma_0 = 70$

In the case of standard deviation $\sigma_0 = 70$ convergence of position of the simulated robot with its estimate in the initial steps of the algorithm slightly increased when the number of particles was increased from $M = 5000$ to $M = 10\,000$ (Figs. 5-6).

In turn, in the case of analysis of the problem of robot localization for the standard deviation $\sigma_0 = 200$

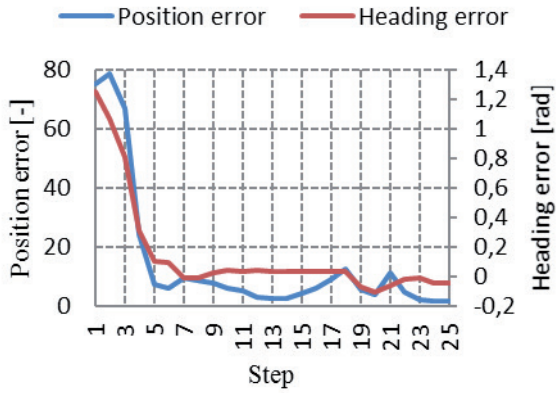


Fig. 6. Position ${}^{\circ}\tilde{r}_R$ and heading ${}^{\circ}\tilde{\varphi}_{0z}$ errors for M = 5 000 and σ_0 = 70

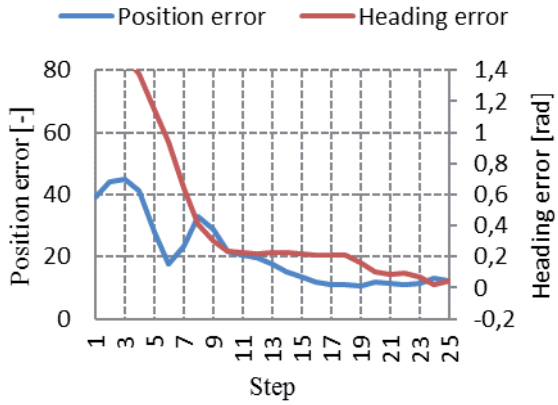


Fig. 7. Position ${}^{\circ}\tilde{r}_R$ and heading ${}^{\circ}\tilde{\varphi}_{0z}$ errors for M = 10 000 and σ_0 = 200

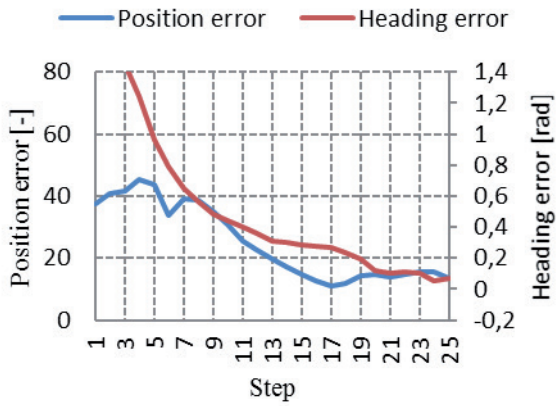


Fig. 8. Position ${}^{\circ}\tilde{r}_R$ and heading ${}^{\circ}\tilde{\varphi}_{0z}$ errors for M = 5 000 and σ_0 = 200

and identical number of particles (Figs. 7-8), the initial error of heading is significantly greater and gradually tends to zero. On the other hand, the position error is smaller right from the beginning and ultimately reaches smaller value.

In the case of significant reduction of the parameter, that is, to σ_0 = 10 and for M = 20 000 (Fig. 9), the correct localization of the robot was achieved already after completion of the first step.

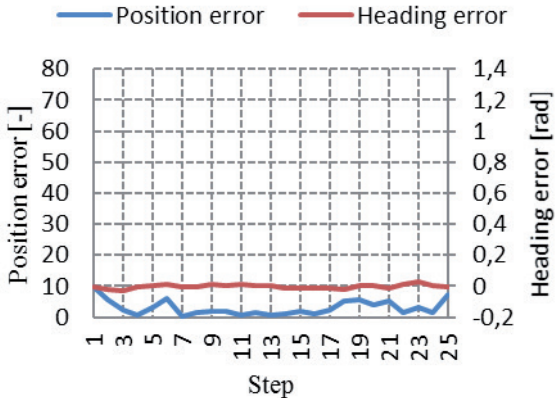


Fig. 9. Position ${}^{\circ}\tilde{r}_R$ and heading ${}^{\circ}\tilde{\varphi}_{0z}$ errors for M = 20 000 and σ_0 = 10

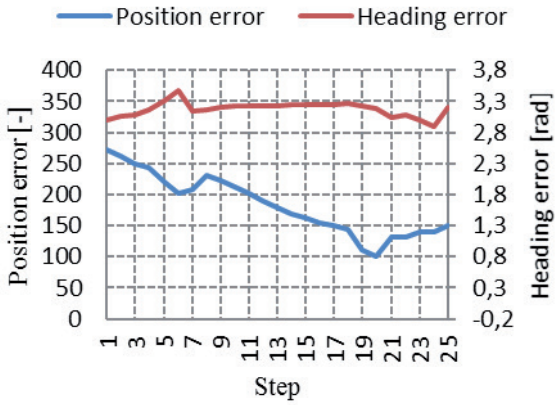


Fig. 10. Position ${}^{\circ}\tilde{r}_R$ and heading ${}^{\circ}\tilde{\varphi}_{0z}$ errors M = 10 000 and σ_0 = 10

Therefore, decreasing the value of σ_0 resulted in significant increase of convergence for the case of large number of particles. However, it turned out to be critical for the case in which M = 10 000 and σ_0 = 10 (Fig. 10), where the error was large, and in the case of heading, error convergence to zero is not visible.

It means that large number of particles in the simulation with small number of the σ_0 parameter can significantly increase the speed and accuracy and simultaneously involves risk of failure of the localization task as a whole.

The presented results clearly point to experimental character of selection of number of particles M and the parameter σ_0 for the given map and used sensors.

6. Experimental Investigations

Experimental investigations concerning localization of the PIAP SCOUT robot were carried out in the research environment, whose schematic diagram is shown in Fig. 11.

The research platform consists of three main components:

- the PIAP SCOUT mobile robot controlled via CAN bus,
- the laptop with software running on Ubuntu operating system with Xenomai real-time framework,

- the system of sensors communicating with the laptop via USB and CAN interfaces.

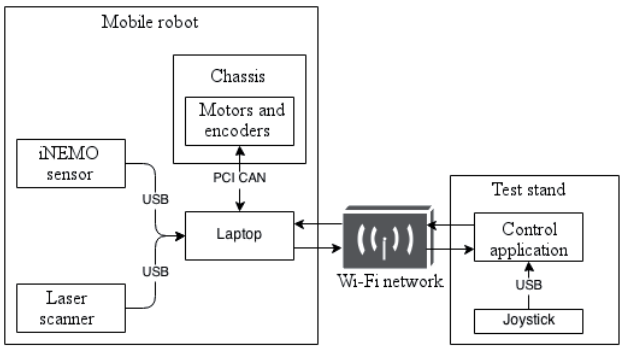


Fig. 11. Schematic diagram of experimental environment

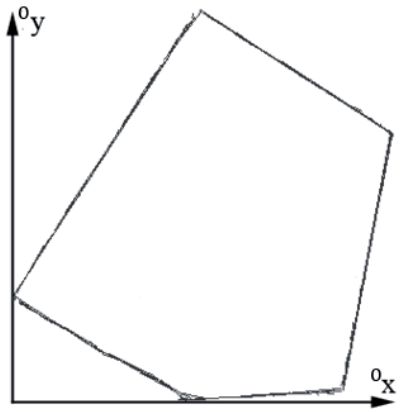


Fig. 12. Robot environment map in the experimental investigations

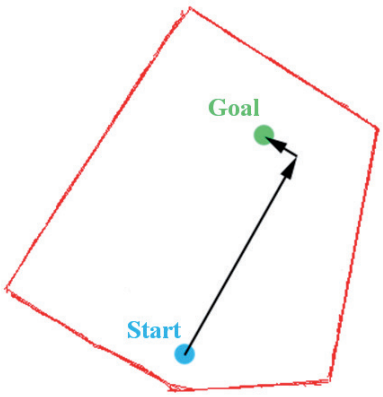


Fig. 13. Schematic illustration of robot desired movement in the experiment 1

Robot control takes place by means of the control panel, using joystick. Both systems, which are of the robot and of the control panel, were connected using the Wi-Fi network.

Experimental investigations of robot localization using the proposed method were performed using the PIAP SCOUT robot described in point 2. Robot environment was the 9 m^2 area made out of the part of the room. The robot environment map with the adopted coordinate system is shown in Fig. 12.

The map used with the algorithm was created using the space scanning technique. Raw laser measurements were used to build a 1000 x 1000 points occupancy map.

Next, the layer of the map obtained in this way was linearized using one of the available algorithms, and as a result the second layer of the vector map consisting of 120 lines was obtained.

The conducted experimental research involved a series of experiments of localization using the earlier described Monte Carlo method, with aggregation of sensor data taken into account.

At first, the experimental standard deviations were chosen: σ_{trans} = 0.1 m, σ_{drift} = 0.09 rad, σ_0 = 50 m. Next, the proper experimental research was done.

During investigations, after measurement of the initial robot pose, the operator using joystick made maneuvers aimed at moving the robot into new location. The task of the algorithm was robot localization during motion and after its completion. After finishing the control process, the final robot pose was measured and also its estimate as a result of localization algorithm operation was determined. During algorithm execution, several quantities were recorded, for instance, values of predicted robot velocities and times required for computation at the given number of particles.

In the present work, results of two selected experiments are shown.

Experiment 1

The first experiment consisted in driving the robot forwards, then making 90 degrees counter-clockwise pivot turn and driving forwards again. In Fig. 13 the desired robot motion is shown in a schematic way.

In turn, in Fig. 14 and in Table 1, accuracy of the final robot pose estimation is illustrated, whereas in Fig. 23 selected corresponding steps of the algorithm are shown.

As a result of the experiment also the following quantities were recorded:

- the number of active particles M involved in the localization process (Fig. 15),
- duration of algorithm computation T (Fig. 16)
- estimated robot velocities (Fig. 17).

Experiment 2

The second experiment was a natural follow up to the first one and it consisted in robot returning to the starting point from the first experiment. The desired motion of the robot is shown in Fig. 18.

The accuracy of the final estimation of the robot pose are illustrated in Fig. 19 and in Table 2, whereas in Fig. 24 are shown selected intermediate steps of the algorithm execution. Similarly as in the previous experiment, the following quantities were recorded during experiment:

- number of the active particles M involved in the localization process (Fig. 20),
- duration of computations of the algorithm T (Fig. 21),
- estimated robot velocities (Fig. 22).

As a result of the conducted experiments with a map of little geometric complexity, relatively good accuracy of robot localization was achieved (Tables 1-2).

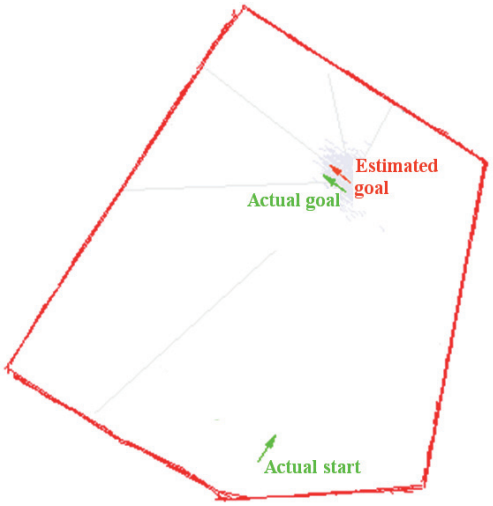


Fig. 14. Results of robot localization in the experiment 1

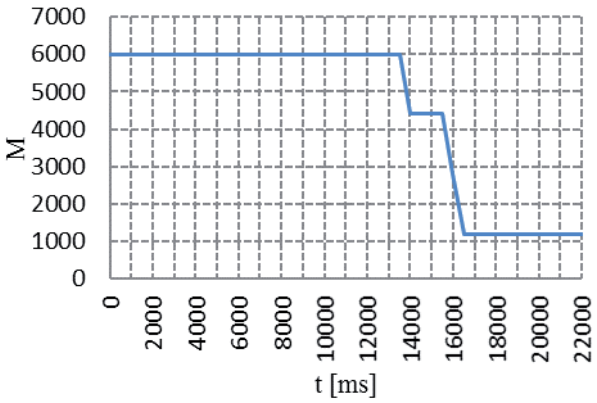


Fig. 15. Change of the number of particles – experiment 1

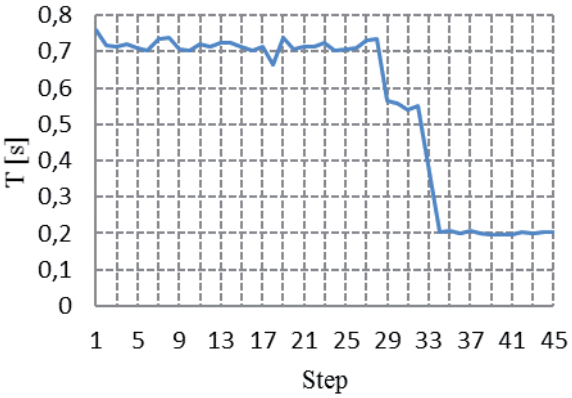


Fig. 16. Time of computation for successive steps of the algorithm – experiment 1

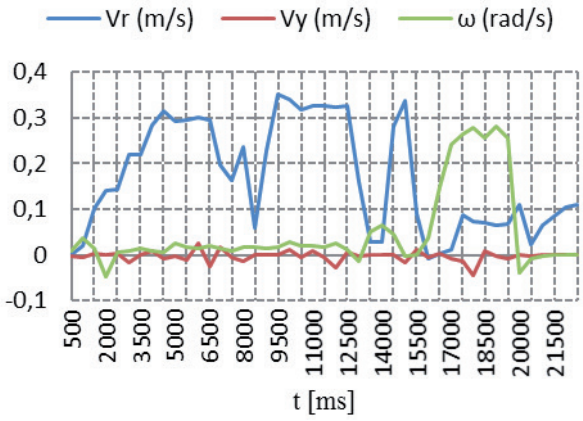


Fig. 17. Estimated velocities of robot motion – experiment 1

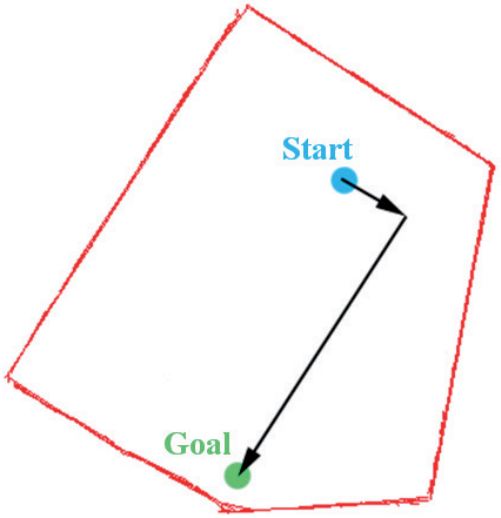


Fig. 18. Schematic illustration of robot desired movement in the experiment 2

Despite larger difficulties in localization in uniform spaces or spaces containing repeating elements, the results of this quality can be considered attractive from the point of view of mobile robotics.

At the beginning of each experiment (Figs. 23-24), relatively large uncertainty of estimation of position can be observed in the form of scattered distribution of particles all over the map. This dis-

Tab. 1. Accuracy of localization in the final position for the experiment 1

Unit	[m]	[m]	[deg]
Actual pose	${}^{0}x_R = 2.23$	${}^{0}y_R = 2.03$	${}^{0}\varphi_{0z} = 142.1$
Estimated pose	${}^{0}\hat{x}_R = 2.25$	${}^{0}\hat{y}_R = 2.09$	${}^{0}\hat{\varphi}_{0z} = 139.4$
Estimation error	${}^{0}\tilde{x}_R = -0.02$	${}^{0}\tilde{y}_R = -0.06$	${}^{0}\tilde{\varphi}_{0z} = +2.7$

Tab. 2. Accuracy of localization in the final position for the experiment 2

Unit	[m]	[m]	[deg]
Actual pose	${}^{0}x_R = 1.14$	${}^{0}y_R = 0.64$	${}^{0}\varphi_{0z} = 44.21$
Estimated pose	${}^{0}\hat{x}_R = 0.91$	${}^{0}\hat{y}_R = 0.57$	${}^{0}\hat{\varphi}_{0z} = 55.85$
Error of estimation	${}^{0}\tilde{x}_R = +0.23$	${}^{0}\tilde{y}_R = +0.07$	${}^{0}\tilde{\varphi}_{0z} = -11.64$

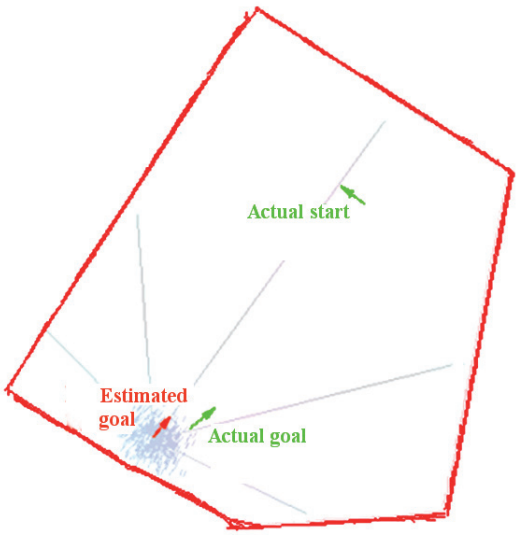


Fig. 19. Results of robot localization in the experiment 2

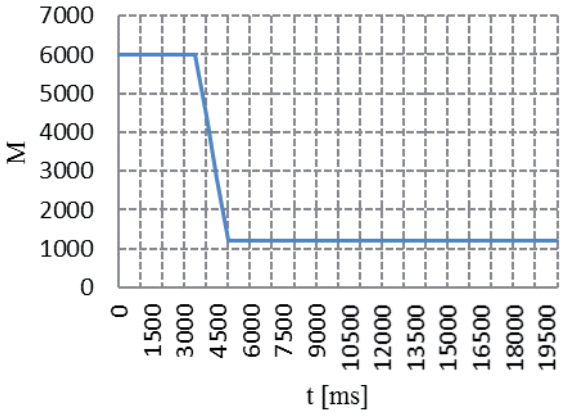


Fig. 20. Change of the number of particles – experiment 2

tribution underwent consolidation in the successive steps, which resulted in more and more accurate localization. During every experiment, the algorithm gradually reduced the number of particles used for localization depending on the ESS value (Effective Sample Size). Moreover, the results of investigations for selected steps of algorithm operation illustrated in Figs. 23-24 show consistency of estimation of robot pose during robot motion with desired movements presented in Figs. 13 and 18. In the figures also the visualized laser beams drawn from the estimated robot position can be noticed, which are consistent with the environment map. It is the evidence of correctness of the localization process (position tracking) during experiment, and not only after its completion.

The time of computations required by the algorithm was dependent proportionally on the number of particles and the level of map complexity (that is, on the number of lines required to its representation for the vector map or the maximum length of the ray for the raster map). One can notice, however, that the mentioned time was significantly different in case of the feature map in comparison to the results of simulation investigations. The explanation is lack of operation of the real-time layer during simulation investigations and absence of the associated latencies (acceleration integration, laser measurements, etc.). The real-time system, in order to achieve high accuracy, operates in the top priority mode with respect to any application in the user space of the Ubuntu system, so its influence on the efficiency is significant.

In Figs. 15-16 and Figs. 20-21 it is possible to notice a distinct fall in the required time of computation for the steps where the number of particles was reduced.

Linear and angular velocities of the robot exhibit high local variability, because of measurement inaccuracy and instability of robot motion during turning maneuvers (Figs. 17 and 22). Moreover, the drift of error of acceleration integration is also significant. This justifies the supposition that those measurements were burdened with quite significant errors, which, however, was not a problem for robot localization with relatively good accuracy. It was possible due to the fact that the Monte Carlo localization algorithms are robust to this kind of errors, because they themselves operate on the principle of introduction of noise to motion and sensor models.

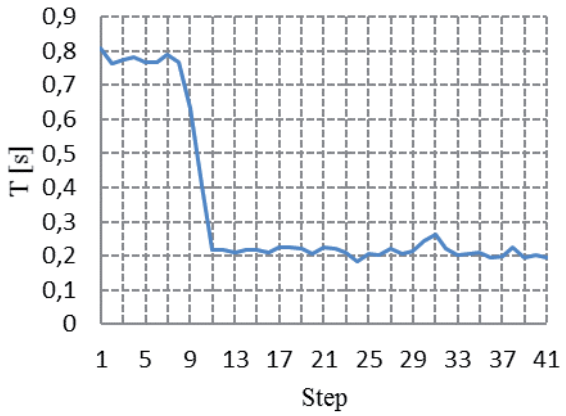


Fig. 21. Time of computation for successive steps of the algorithm – experiment 2

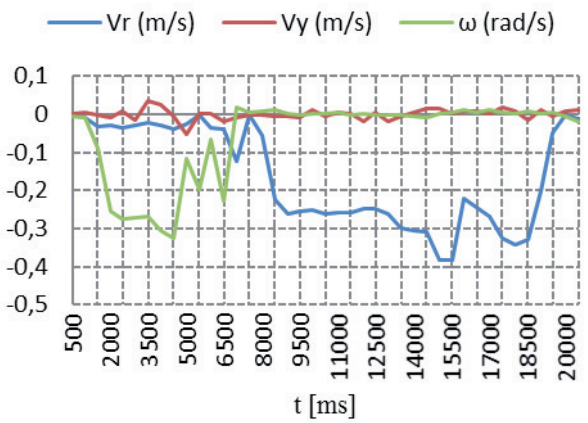


Fig. 22. Estimated velocities of robot motion – experiment 2

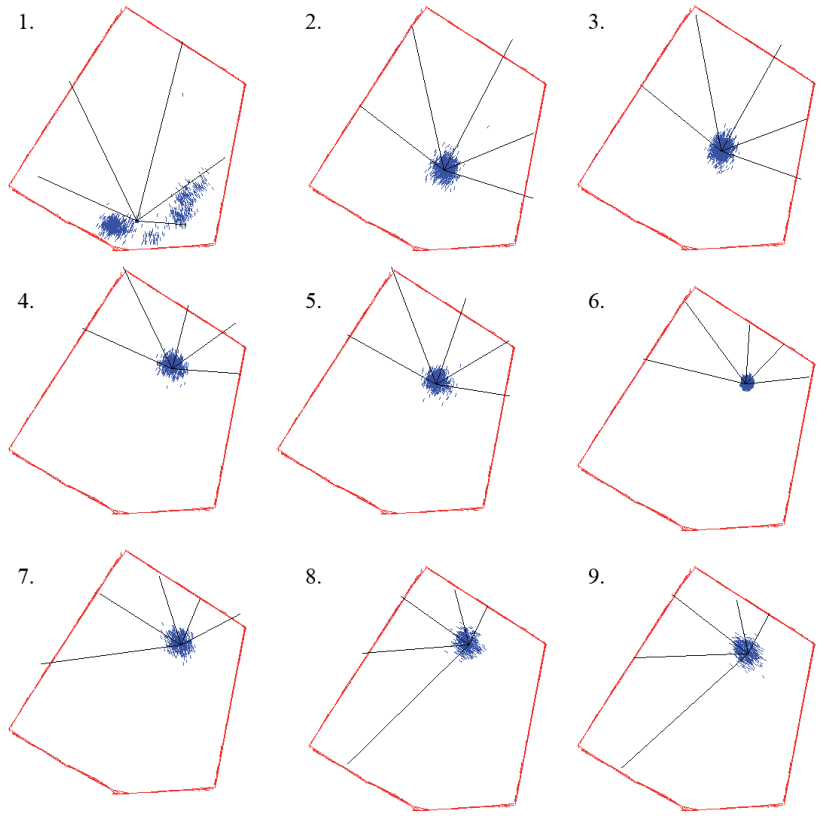


Fig. 23. Selected steps of localization – experiment 1

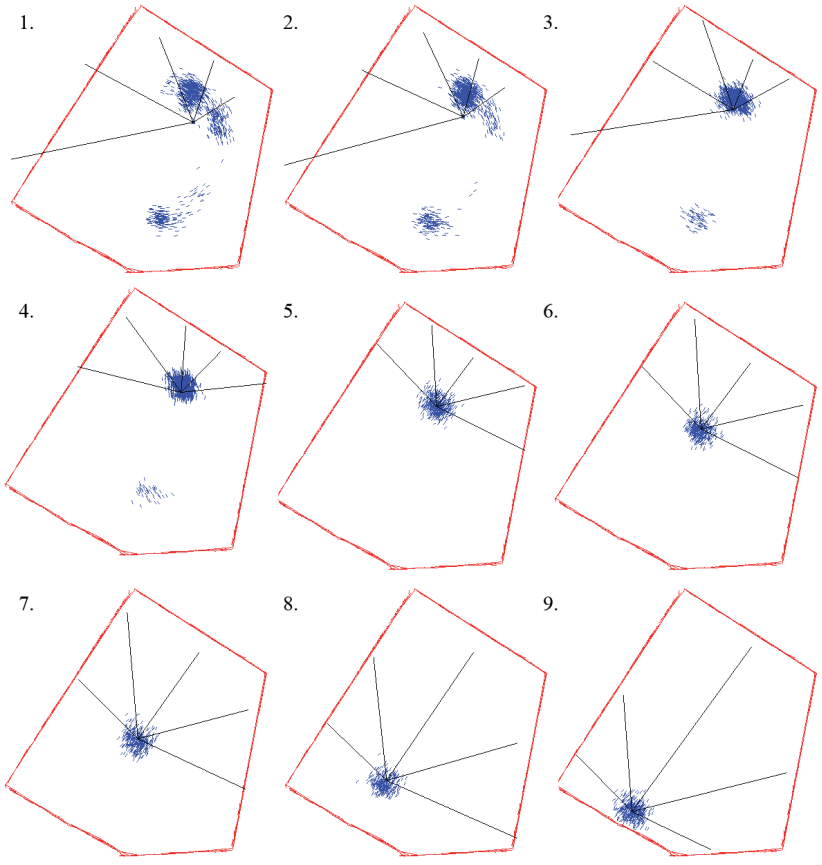


Fig. 24. Selected steps of localization – experiment 2

7. Conclusions and Future Works

The proposed algorithm of localization works quickly and effectively on the real mobile robot. The results of localization are satisfactory, and because of speed of algorithm operation its use in real time is possible. Thanks to data aggregation, distance sensors, accelerometers and encoders fulfill their role as components gathering information required for assigning weights to particles and prediction of robot motion.

In order to further develop the presented method and increase the accuracy of prediction of robot motion, the following works are planned:

- enhancement of fusion of data from the sensors with data from the gyroscope in order to improve accuracy of prediction of robot heading and/or with the global navigation satellite system (GNSS) [22] in case of robot motion in open terrain.
- addition of the third spatial dimension of the environment, which will enable localization in the environment of variable surface inclination.

ACKNOWLEDGEMENTS

The work has been realized as a part of the project entitled "Dynamics modeling of four-wheeled mobile robot and tracking control of its motion with limitation of wheels slip". The project is financed from the means of National Science Centre of Poland granted on the basis of decision number DEC-2011/03/B/ST7/02532.

AUTHORS

Piotr Jaroszek* – Industrial Research Institute for Automation and Measurements (PIAP), Warsaw, 02-486, Poland.
E-mail: pjaroszek@piap.pl,

Maciej Trojnacki – Industrial Research Institute for Automation and Measurements (PIAP), Warsaw, 02-486, Poland.
E-mail: mtrojnacki@piap.pl.

*Corresponding author

REFERENCES

[1] Siegwart R., *Introduction to Autonomous Mobile Robots*, MIT Press, 2004.

[2] Cox I. J., Wilfong G. T., ed., *Autonomous Robot Vehicles*, New York, NY, USA: Springer-Verlag New York, Inc., 1990. DOI: 10.1007/978-1-4613-8997-2.

[3] Fukuda T., Ito S., Oota N., Arai F., Abe Y., Tanaka K., Tanaka Y., „Navigation system based on ceiling landmark recognition for autonomous mobile robot". In: *International Conference on Industrial Electronics, Control, and Instrumentation, 1993. Proceedings of the IECON '93*, vol. 3, 1993, 1466–1471. DOI: 10.1109/IECON.1993.339287.

[4] Hertzberg J., Kirchner F., „Landmark-based autonomous navigation in sewerage pipes". In: , *Proceedings of the First Euromicro Workshop on Advanced Mobile Robot,* 1996, 68–73. DOI: 10.1109/EURBOT.1996.551883.

[5] Jaroszek P., *Globalne planowanie ścieżki* (*Global Path Planning)*, Bachelor of Engineering Thesis, Warsaw University of Technology, Warsaw, 2012. (In Polish)

[6] Bartoszek J., Trojnacki M., Bigaj P., „Simulation of semiautonomy mode for ibis mobile robot with analysis of sensor failure tolerance", *Journal of Automation, Mobile Robotics, & Intelligent System.*, Vol. 5, No. 4, ss. 3–10, 2011.

[7] Borenstein J., *Navigating Mobile Robots: Systems and Techniques*. Wellesley, Mass: A K Peters Ltd, 1996.

[8] Yamauchi B., Schultz A., Adams W., „Mobile robot exploration and map-building with continuous localization". In: *1998 IEEE International Conference on Robotics and Automation. Proceedings*, 1998, vol. 4, 3715–3720. DOI: 10.1109/ROBOT.1998.681416.

[9] H. M. Choset, *Principles of Robot Motion: Theory, Algorithms, and Implementation*. MIT Press, 2005.

[10] Burgard D., Fox, D. Hennig, i T. Schmidt, „Estimating the Absolute Position of a Mobile Robot Using Position Probability Grids", 1996. .

[11] Cassandra A., Kaelbling L. P., Kurien J., „Acting under uncertainty: discrete Bayesian models for mobile-robot navigation", w *Proceedings of the 1996 IEEE/RSJ International Conference on Intelligent Robots and Systems '96, IROS 96*, 1996, vol. 2, 963–972.

[12] Kalman R., „A New Approach to Linear Filtering and Prediction Problems", *Trans. ASME – J. Basic Eng.*, no. 82 (Series D), 1960, 35–45.

[13] Thrun S., „Bayesian Landmark Learning for Mobile Robot Localization", *Machine Learning,* vol 33, no. 1, October, 1998, 41–76.

[14] Burgard W., Derr A., Fox D., Cremers A., „Integrating global position estimation and position tracking for mobile robots: the dynamic Markov localization approach". In: *1998 IEEE/RSJ International Conference on Intelligent Robots and Systems,. Proceedings*, vol. 2, 730–735. DOI: 10.1109/IROS.1998.727279.

[15] Thrun S., Fox D., Burgard W., Dellaert F., *Robust Monte Carlo Localization for Mobile Robots*, *Artificial Intelligence,* vol. 128, no. 1–2, 2001, 99–141. DOI: 10.1016/S0004-3702(01)00069-8.

[16] Bedkowski J., Maslowski A., De Cubber G., „Real time 3D localization and mapping for USAR robotic application", *Industrial Robot: An International Journal*, vol. 39, no. 5, 2012, 464–474.

[17] Mandic D. P., Obradovic D., Kuh A., et al. „Data Fusion for Modern Engineering Applications: An Overview". In: *Artificial Neural Networks: Formal Models and Their Applications – ICANN 2005*, ed. W. Duch, J. Kacprzyk, E. Oja, and S. Zadrożny, Springer Berlin Heidelberg, 2005, 715–721.

[18] „PIAP – producer of EOD equipment, EOD robots and surveillance robots". [Online]. http://anti-terrorism.eu/en/.

[19] Trojnacki M., *Dynamics modeling of wheeled mobile robots,* PIAP Publ. House, Warsaw 2013.

[20] Russell S. J., Norvig P., *Artificial Intelligence: A odern Approach*, Prentice Hall, 2010.

[21] Robert C. P., *Introducing Monte Carlo Methods with R*, 2010 ed., New York: Springer Verlag, 2009. DOI: 10.1007/978-1-4419-1576-4.

[22] Perski A., Wieczynski A., Baczynska M., et al., „Odbiorniki GNSS w praktyce inżynierskiej. Badania stacjonarne", *Pomiary Automatyka Robot.*, vol. 17, no. 4, 2013, 64–77. (In Polish)