UDC 007.5:519.2

A. Kudriashov, T. Buratowski, M. Giergiel
AGH University of Science and Technology

ROBOT'S POSE ESTIMATION IN ENVIRONMENT EXPLORATION PROCESS WITH SLAM AND LASER TECHNIQUES

In this paper autonomous mobile robot system for exploration purposes in a rough and hard-accessed terrain has been presented. The main focus of the research was on robot's pose estimation during simultaneous localization and mapping (SLAM) process that is based only on laser and wheel odometry information, when robot is moving through unknown environment and the destination goals are outside of discovered area. For SLAM solution algorithms based on environment laser scan and precise localization calculation – probabilistic methods by Adaptive Monte Carlo localization (AMCL) is applied and compared with Extended Kalman Filter (EKF) solution. Mobile robot platform is based on 2-wheeled differential drive prototype with ARM-architecture controller which uses linear and radial velocities input for motion. All algorithms and sub-systems are implemented with Robot Operation System (ROS) navigation stack framework.

Keywords: *AMCL, EKF, exploration, localization, mobile robot, ROS, SLAM.*

А. Кудряшов, Т. Буратовський, М. Гергель

ОЦІНКА РОБОТА В ПРОЦЕСІ РОЗВІДКИ НАВКОЛИШНЬОГО СЕРЕДОВИЩА КОНТАКТНИМ ТА ЛАЗЕРНИМ МЕТОДАМИ

У даній статті представлена автономна мобільна роботизовано система для розвідки твердого та важкодоступного рельєфу. У дослідженні основну увагу було приділено оцінки робота в процесі одночасної локалізації та картографії (SLAM), яка базується на інформації що сприйнята лазером коли робот переміщується по невідомій поверхні а цілі призначенн знаходяться за межами відомої області. Для алгоритмів рішення (SLAM), що базуються на лазерному скануванні середовища та точному розрахунку локалізації , застосовуються імовірні методи адаптивної локалізації Монте-Карло (AMCL) та порівнюються з розширеним рішенням фільтра Калмана (EKF). Мобільна платформа заснована на двоколісному прототипі з диференціальним приводом та контролером архітектури ARM, який використовує лінійні і радіальні швидкості що вводяться для руху. Всі алгоритми і підсистеми реалізовані з використанням стека навігаційної стековой системи Robot Operation System (ROS).

Ключові слова: *AMCL, EKF, розвідка, локалізація, мобільний робот, ROS, SLAM.*

А. Кудряшов, Т. Буратовський, М. Гергель

ОЦЕНКА РАБОТА В ПРОЦЕССЕ РАЗВЕДКИ ОКРУЖАЮЩЕЙ СРЕДЫ КОНТАКТНЫМ И ЛАЗЕРНЫМ МЕТОДОМ

В этой статье представлена автономная мобильная роботизированная система для разведки на грубом и труднодоступном рельефе. Основное внимание в исследовании было уделено оценке позы робота в процессе одновременной локализации и картографии (SLAM), которая основана только на информации о лазерной и одометрии лазера, когда робот перемещается по неизвестной среде, а цели назначения находятся за пределами обнаруженной области. Для алгоритмов решения SLAM, основанных на лазерном сканировании среды и точном расчете локализации, применяются вероятностные методы с помощью адаптивной локализации Монте-Карло (AMCL) и сравнивается с расширенным решением фильтра Калмана (EKF). Мобильная робостроительная платформа основана на двухколесном прототипе дифференциального привода с контроллером ARM-архитектуры, который использует линейные и радиальные скорости, вводимые для движения. Все алгоритмы и подсистемы реализованы с использованием стека навигационной стековой системы Robot Operation System (ROS).

Ключевые слова: *AMCL, EKF, разведка, локализация, мобильный робот, ROS, SLAM.*

Introduction. For any robotic tasks it is very important to have precise interesting information gathered as well as clearly defined subject pose. Mobile robots, especially wheeled mobile robot, like the presented platform, are commonly used in inspection. Often, when target area is hard-accessed or might injure human's health, possibility of an autonomous work is required. Both environment map and localization respected to some coordinate system are needed, so it seems to be some kind of 'the chicken or the egg' dilemma: map is required for self-localize, but to build a map, robot's pose must be known [8]. So the solution for this might be simultaneous localization estimating and map creating. That kind of algorithms are commonly named as SLAM (simultaneous localization and mapping).

SLAM is a method which solves problem of map building with landmark measurements and localization when the subject is moving. It is commonly used in robotics and autonomous systems.

Snapshot map creation is possible by using many different measuring systems, like: ultrasonic range sensor, radars, lidars, vision systems etc. The results might be very precise, but it's possible only when the subject (robot) is in a static state. While the subject is moving, localization and current measures are changing. To be able to extend map it is required to merge with previous results. But this is not an easy task. SLAM is a method which helps to solve this issue by giving knowledge of current localization the map.

For general understanding of SLAM problem and its solution, the full process might be defined by four vectors: controls *U*, observations *Z*, map *m* and robots pose *X* [7, 9]:

$$\begin{gathered} U_{1:t} = \{U_1, U_2, U_3 \ldots, U_t\} \\ Z_{1:t} = \{Z_1, Z_2, Z_3 \ldots, Z_t\} \\ m \\ X_{0:t} = \{X_0, X_1, X_2 \ldots, X_t\} \end{gathered} \tag{1}$$

Where *U*, *Z* are given values and *m*, *X* are calculated or estimated. Robot pose during its moving according to the mobile robot's kinematics is defined by position and orientation $X = [x \quad y \quad \theta]^T$ with respect to coordinates frame.

Probabilistic estimation of robot's pose and the map for every step is named as Probability Density Function presented as **[7,** 8]:

$$p(X_{0:t}, m | Z_{1:t}, U_{1:t}) \tag{2}$$

Equation (2) is known as a Full SLAM and almost never used in online mobile systems, in practice the most recent pose is calculated by recursive integration once at the time (3), this algorithm is called Online SLAM [9]:

$$p(X_{t+1}, m | Z_{1:t+1}, U_{1:t+1}) = \int_{X_0} \ldots \int_{X_{t-1}} p(X_{0:t+1}, m | Z_{1:t+1}, U_{1:t+1}) dX_t \ldots dX_0 \tag{3}$$

Mobile robot's pose *X* estimation in robotics literature is usually called as solving robot localization problem. A ate when robot does not know its initial pose regarding to environment (map) is named the global localization problem [10] or self-localization [8] For solving this problem there are two major academic communities: a Kalman filter school, which mostly use variations of Extended Kalman Filter (EKF) and a particular filters school that created an Adaptive Monte-Carlo Localization (AMCL) filter.

An Extended Kalman Filter (EKF) localization

The basis for the EKF SLAM method is to describe the vehicle motion in the following way [1, **7]**:

$$p(X_t, m | Z_{t-1}, U_t) \Leftrightarrow X_t = f(X_{t-1}, U_t) + w_t \tag{4}$$

Robot's kinematic model $f()$ is used together with additive, zero mean uncorrelated Gaussian motion disturbances w_t with a covariance Q_t. In that assumptions the observations Z_t shall be described by geometry observation model $h()$ with additive, zero mean uncorrelated Gaussian observation errors v_t with covariance R_t **[7]:**

$$p(Z_t | X_t, m) \Leftrightarrow z(t) = h(X_t, m) + v_t \tag{5}$$

One of many implementations of EKF SLAM process could be shown as on fig. 1, where Extended Kalman Filter is used together with laser scanning. After getting laser measure results of an environment state the initial map is creating. Then method checks by Odometry change if localization is changed, in the meanwhile new laser scan are putting into Kalman filter and comparing between them-selves. If odometry is empty - most intensive points are choosing as current map and it's looping again. When new odometry information is coming then algorithm merging current and previous localization together with a laser scan results at the same time and updating global environment picture (fig. 1).

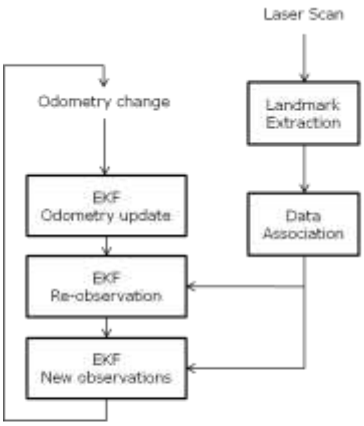

Fig. 1. - **Example EKF-SLAM block diagram**

In [2] environment mapping tests with EKF-SLAM algorithm has been prepared and self-localization problem by using wheel and IMU odometry information combined with estimated robot's path has been solved (fig. 2).

Fig. 2. - **Robot's path respect to wheel odometry information (green) and estimated by EKF (red) [2].**

Adaptive Monte-Carlo Localization

As mentioned before, the second most often used algorithm for robot pose estimating is Adaptive Monte-Carlo Localization (AMCL). AMCL is a particular filter, variant of Markov localization family. Markov localization uses Bayes rule to update beliefs - probability distribution of robot's position when it's moving or getting information from sensors [4, 5]. According to the authors of MCL, in contrast to other filters, MCL [4]:

- is able to represent multi-modal distributions which might help with robot's global localization.
- drastically reduces the amount of memory required compared to grid-based Markov localization, and it can integrate measurements at a considerably higher frequency
- is more accurate than Markov localization with a fixed cell size, as the state represented in the samples not discretized.
- is easiest to implement.

Robot's pose according to Bayes rule and MCL is defined by Probability Density Function in following way [6]:

$$p(X_t|Z_{1:t},U_{0:t-1}) = \alpha \cdot p(Z_t|X_t)\int p(X_t|U_{t-1},X_{t-1}) \cdot p(X_{t-1})dX_{t-1} \tag{6}$$

Where α is normalization constant that is used to ensure that $p(X_t|Z_{1:t},U_{0:t-1})$ is one over all X_t. MCL is generally done by two steps: prediction and correction. Prediction $p(X_t|U_{t-1},X_{t-1})$ is given from robot's previous control U_{t-1} and X_{t-1} pose and correction $p(Z_t|X_t)$ that is taken from observation Z_t on position X_t.

The results of pose estimation for predefined known environment with laser sensors and sense-less systems might be found in many works like [4]:

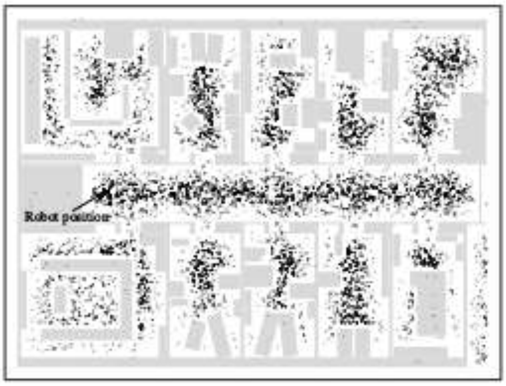

Fig. 3. - **Robot's pose estimation by AMCL – initial state [4].**

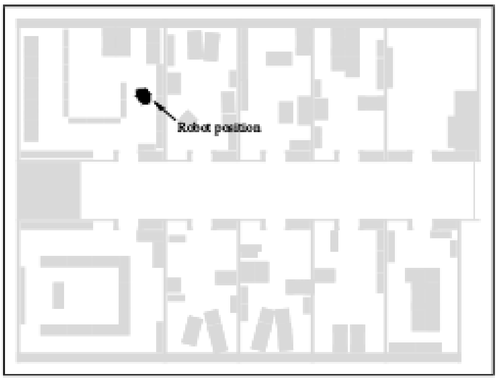

Fig. 4. - **Robot's pose estimation by AMCL – after calculating [4].**

Autonomous Mobile Robots platform prototype

For this paper Mobile robot platform is based on 2-wheeled differential type mobile robot described in details in [3] with sensor system modification from infrared and ultrasonic sensors to lidar [2] and image camera vision system.

Fig. 5. - **Autonomous mobile robot prototype during exploration tests.**

As control system is used ROS Navigation stack, which consists of several different applications connected together:

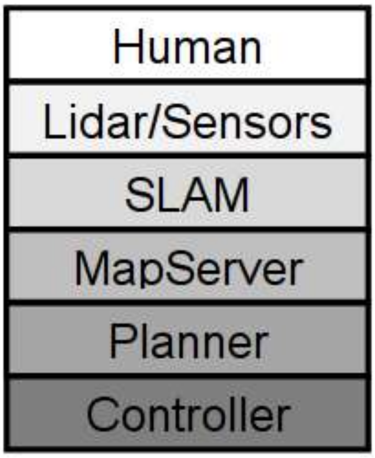

Fig. 6. - **ROS Navigation Stack diagram**

Human control in exploration mode is limited to enabling and disabling robot move processes, and in normal autonomic mode to setting navigation goal required by planners.

Map is calculated based on laser scans. The used sensor is Hokuyo UTM-30LX-EW lidar (LIght Detection And Ranging), with distance resolution 10÷30mm and 0.1÷30m of guaranteed distance range (max is up to 60m), and angular specifications: angular range — 270°, angular resolution - 0.25°.

As trajectory planners for autonomous exploration work was chosen: navfn::NavfnROS and base_local_planner::TrajectoryPlannerROS.

Developed drive controller is supporting navigation stack message system. Planers generate ROS ‘geometry_msgs/Twist’ message which consists of linear and angular velocity vectors of robot frame and by solving kinematic equation (X) it is possible to efficiently control mobile robot.

Kinematic model

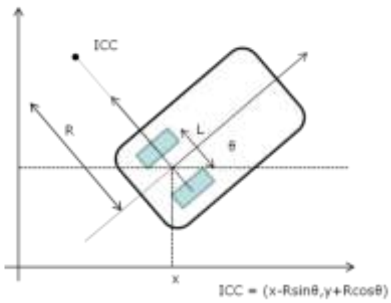

Fig. 7.- **Differential drive mobile robot model**

Kinematic model (fig. 7.) of that kind of mobile robots is required for control applications and can be defined as:

$$w_r(t) = \frac{v_r(t)}{R + \frac{L}{2}} \tag{7}$$

$$w_l(t) = \frac{v_l(t)}{R - \frac{L}{2}} \tag{8}$$

$$v(t) = w(t)R = \frac{1}{2}(v_r(t) + v_l(t)) \tag{9}$$

$$\begin{bmatrix} v_x(t) \\ v_y(t) \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} r/2 & r/2 \\ 0 & 0 \\ -r/L & r/L \end{bmatrix} \begin{bmatrix} w_l(t) \\ w_r(t) \end{bmatrix} \tag{10}$$

Robot's linear $\dot{x}(t)$, $\dot{y}(t)$ and angular $\dot{\theta}(t)$ velocities in world frame might be calculated from following:

$$\begin{aligned} \dot{x}(t) &= v(t)\cos\theta(t) \\ \dot{y}(t) &= v(t)\sin\theta(t) \\ \dot{\theta}(t) &= w(t) \end{aligned} \tag{11}$$

By compiling (10) and (11) together we can find forward kinematics solution for 2-wheeled differential type mobile robot:

$$\begin{bmatrix} v_x(t) \\ v_y(t) \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \cos\theta & 0 \\ \sin\theta & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v(t) \\ w(t) \end{bmatrix} = \begin{bmatrix} v(t)\cos\theta & 0 \\ v(t)\sin\theta & 0 \\ w(t) & 1 \end{bmatrix} \begin{bmatrix} v_l(t) \\ v_r(t) \end{bmatrix} =$$

$$= \begin{bmatrix} \frac{1}{2}(v_r + v_l)\cos\theta \\ \frac{1}{2}(v_r + v_l)\cos\theta \\ (v_r + v_l)/L \end{bmatrix} = \begin{bmatrix} \frac{1}{2}\cos\theta & \frac{1}{2}\cos\theta \\ \frac{1}{2}\sin\theta & \frac{1}{2}\sin\theta \\ -1/L & 1/L \end{bmatrix} \begin{bmatrix} v_r \\ v_l \end{bmatrix} \tag{12}$$

AMCL-SLAM Exploration tests

For research purpose testing environment inspected by SLAM process with manual control (without using planners) was prepared. On the environment map (fig 8) five different-size and -form objects might be found which make impossible to scan the whole field without exploration. Trajectory of robot's motion collected from wheel odometry is shown on fig 9.

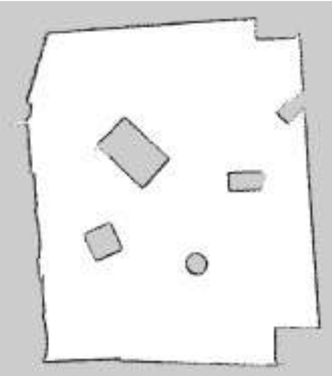

Fig. 8. - **Environment map compiled by Gmapping pure SLAM process.**

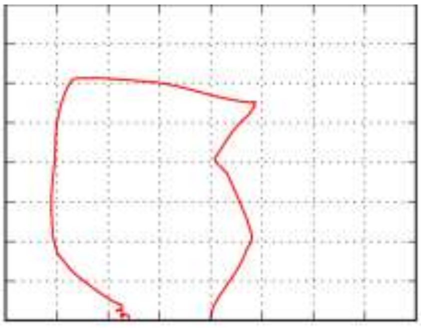

Fig. 9. - **Robot's trajectory taken from wheel odometry during Gmapping pure SLAM process**

Results of simultaneous localization and dynamic laser mapping of undiscovered environment by AMCL algorithm for pose estimation could be found below. As initial pose is taken only from laser observation, robot's localization probability array is much dispersed (fig. 10).

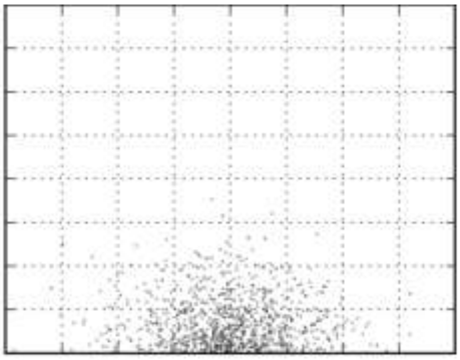

Fig. 10. - **Probability of robot's pose after AMCL-SLAM initialization with only first observation information**

After navigation goal is set, planners are compiling trajectory and giving information to robot's controller with linear and angular velocities related to world's frame. On fig. 11 robot's estimated by AMCL and taken from wheel odometry trajectories are shown. The result of pose estimation as array of poses compared to wheel odometry is on fig 12. As it is not hard to discover, during exploration when motion information was aggregated, dispersion of robot's pose probability is decreasing.

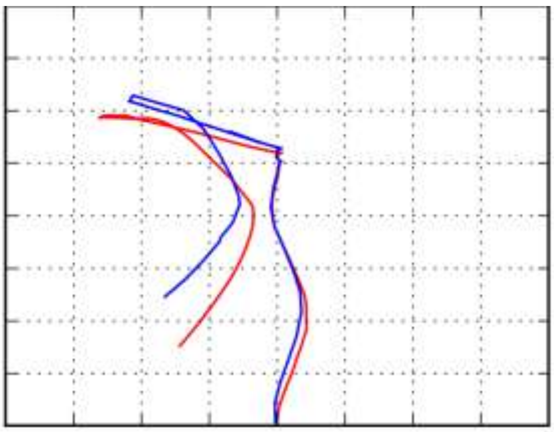

Fig. 11. - **Robot's trajectory taken from wheel odometry (red) and AMCL estimation (blue) during autonomous exploration SLAM process**

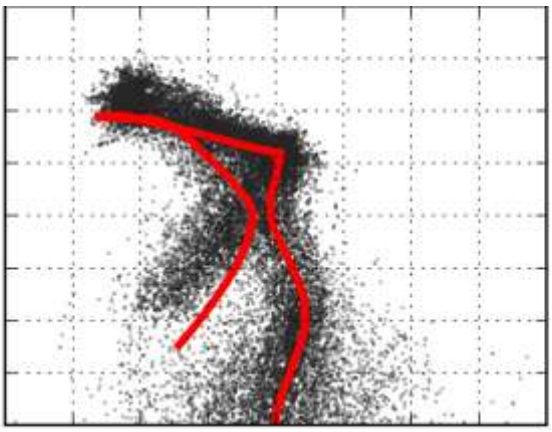

Fig. 12. - **Probability of robot's pose during AMCL-SLAM autonomous exploration together with trajectory taken from wheels odometry (red).**

The map compiled in autonomous exploration work by AMCL-SLAM is shown on fig 13:

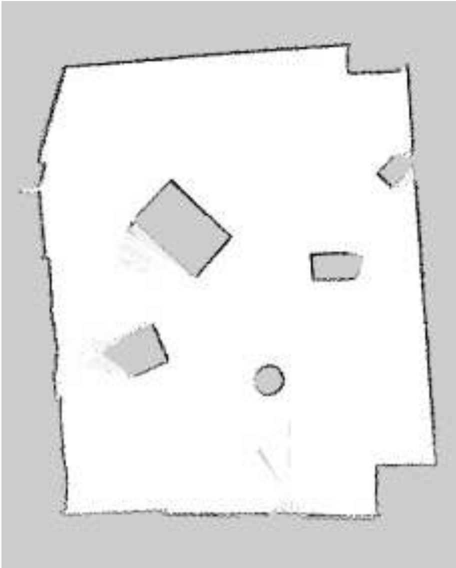

Fig. 13. - **Environment map compiled AMCL-SLAM autonomous exploration process.**

Another test with wheels skidding errors was prepared. As can be seen, in this case pose estimation help to find real robot's global localization quickly. Due to environmental errors from skidding trajectory from only wheels odometry was dropped outside real testing field.

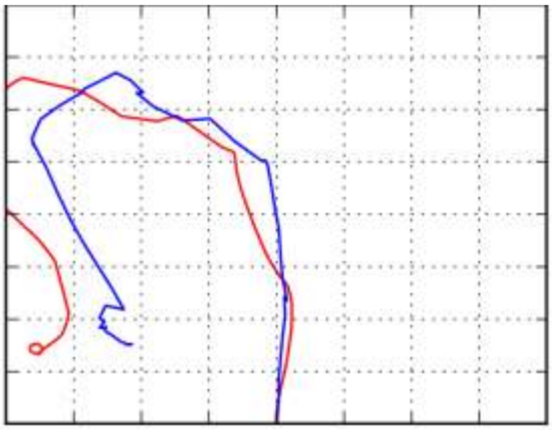

Fig. 14. - **Robot's trajectory taken from wheel odometry (red) and AMCL estimation (blue) during autonomous exploration SLAM process with environmental errors**

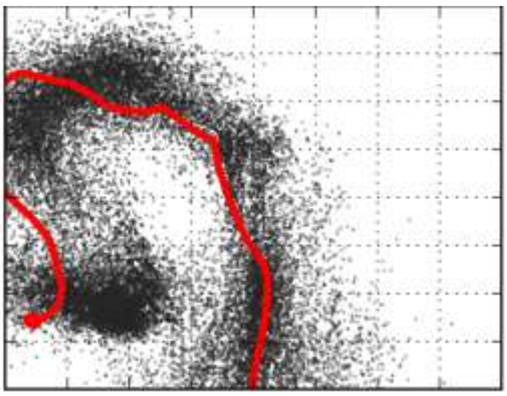

Fig. 15.- **Probability of robot's pose together with trajectory taken from wheels odometry (red) during AMCL-SLAM autonomous exploration with environmental errors.**

The resulted map in this experiment looks almost the same as without any issues (fig. 16):

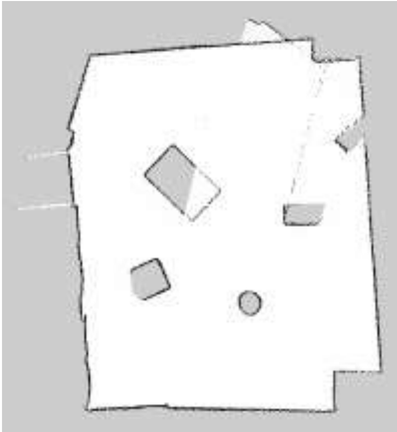

Fig. 16. - **Environment map compiled AMCL-SLAM autonomous exploration process with skidding errors**.

Summarize

According to current research presented above it can be assumed that efficient pose estimation is mandatory for a correct mobile robot work. Wheel odometry shows us a path of robot from robot's wheel point of view, but it might consist many errors even in ideal indoor solutions [3]**.** That is why it generally cannot be used as only one source of robot's pose for different terrain. Using such estimators like an Extended Kalman filter with different sources of sensor information, like IMU, GPS, laser/ultrasonic measurements, increases precision of localization calculation. However, EKF doesn't solve the problem of global localization at initial state and cannot be used for the solution of 'the kidnapping problem'. EKF is mostly focused on tracking estimating issue, where it has very strong results and usage. For self-localization or global localization respectively to some frame with coordinate system, like map, at the initial state, particular filters like Markov filter or its evolution - Adaptive Monte Carlo Localization might be a better choice.

According to results presented in this paper it is possible to say that AMCL is a strong concurrent for EKF in SLAM processes which definitely could be used to efficient elimination of common localization errors during simultaneous mapping and localization process. Extending mobile robot platform with AMCL-SLAM algorithms increase resistance of localization errors which positively impact on map building in rough and undiscovered terrain. On fig. 16 we can find a map which is created when odometry information was disturbed by wheels skidding. Map is almost the same as in undisturbed process and better than in the situation when AMCL or EKF are not used. That might confirm the previous assumptions regarding AMCL-SLAM for exploration process and open a road to better pose estimation researches for SLAM approaches in rough terrain.

Bibliography:

1.Afanasyev, I., Sagitov, A., & Magid, E. (2015, October). ROS-based SLAM for a Gazebo-simulated mobile robot in image-based 3D model of indoor environment. In International Conference on Advanced Concepts for Intelligent Vision Systems (pp. 273-283). Springer International Publishing.

2.Buratowski, T., Ciszewski, M., Giergiel, M., Kudriashov, A., & Mitka, Ł. (2017). Robot z laserowym czujnikiem odległości do budowy map 2D. Modelowanie inżynierskie. 61, 27-33.

3.Buratowski, T., Dąbrowski, B., Uhl, T., & Kwosek, J. (2015). The precise odometry navigation for the group of robots. Schedae Informaticae, 19, 99-111.

4.Fox, D., Burgard, W., Dellaert, F., & Thrun, S. (1999). Monte carlo localization: Efficient position estimation for mobile robots. AAAI/IAAI, 1999(343-349), 2-2.

5.Nourbakhsh, I., Powers, R., & Birchfield, S. (1995). DERVISH an office-navigating robot. AI magazine, 16(2), 53.

6.Pfaff, P., Burgard, W., & Fox, D. (2006). Robust monte-carlo localization using adaptive likelihood models. In European robotics symposium 2006 (pp. 181-194). Springer Berlin/Heidelberg.

7.Riisgard, S., & Blas, M. R. (2011). SLAM for Dummies-A Tutorial Approach to Simultaneous Localization and Mapping.

8.Skrzypczyński, P. (2009). Simultaneous localization and mapping: A feature-based probabilistic approach. International Journal of Applied Mathematics and Computer Science, 19(4), 575-588.

9.Stachniss, C. (2014). Introduction to robot mapping. http://ais.informatik.uni-freiburg.de/teaching/ws12/mapping/pdf/slam01-intro.pdf [access 24.04.2017]

10. Thrun, S. (2000). Probabilistic algorithms in robotics. Ai Magazine, 21(4), 93.

Стаття надійшла до редакції 03.05.2017