Localization of holonomous mobile robot HOLBOS using extended Kalman filter (EKF) and robotic vision

Conference Paper · November 2013

DOI: 10.1109/IECON.2013.6699806

Localization of holonomous mobile robot HOLBOS using extended Kalman filter (EKF) and robotic vision

Jasmin Velagić, *Member, IEEE*, Admir Kaknjo, *Member,* Nedim Osmić, *Member, IEEE,* Muhidin Hujdur*, Member, IEEE and* Faruk Dautović*, Member, IEEE*

Abstract—**Determining the position of a mobile robot in every time instant from sensor data is the fundamental problem in mobile robotics. This paper considers a localization of holonomous mobile robot solved in this paper using two different approaches: odometry localization and landmark based localization. In both cases the robot is placed in known environment with landmarks whose coordinates were also known. Detecting the landmarks was done by using the Microsoft Kinect camera. For odometry localization four encoders were used. Data acquired from encoders and camera is fused together employing extended Kalman filter in order to get more accurate estimation of position and orientation. Obtained experimental results prove that using encoders without any additional measurements is not enough for getting reliable estimation of robot's position. Odometry localization produced an error that accumulates over time, while in the case of landmark based localization, the error is kept inside acceptable limits.**

I. Introduction

Mobile robots are designed for a specific purpose and for performing various predefined tasks. There are robots acting as a guides in museums [1][2], warehouse robots for getting and placing equipment in a warehouse [3], explorer robots that are used in a hostile and harsh environments instead of humans (in space), etc. In order to be able to accomplish the specific tasks mentioned robots have to provide autonomous navigation capabilities. In order to the robot can navigate autonomously, set of basic problems have to be solved earlier. These problems can be represented by three fundamental statements in mobile robotics [4]:

1) The robot has to know where it is in order to make useful decisions.
2) In order to do some tasks the robot has to know where it is going.
3) Once the robot knows where it is and where it has to go it has to decide on how to get there.

One of the first localization approaches was based on odometery and accelerometer measurements, which is known as "Dead Reckoning" [7]. However, this approach is prone to errors because of the disturbances that affect measurements and hence the estimation of robot's location can be inaccurate. Dead reckoning algorithm represents deterministic algorithm, because it does not make any assumptions about the disturbances in the system. In order to increase accuracy and precision in determining the robot's location, it is necessary to use some of the probabilistic localization algorithms. The main idea of probabilistic techniques in robotics is explicit representation of uncertainty using the probability theory. In localization, this can be achieved employing a distribution function in representing the position and orientation of mobile robots. Some of probabilistic algorithms that can be used in localization are: Monte Carlo algorithm [8], [9], [10], particle filter localization [11], as well as localization based on extended Kalman filter [8], [12], [13].

This paper deals with determining the position of mobile robot in every time instant. In order to determine the location of the mobile robot, it is necessary to know the map of the environment in which the robot operates. Although techniques for simultaneous localization and mapping (SLAM) exist, here it is assumed that map of environment is a priori known. Considering the fact that the robot used in this paper was constructed to be used in warehouse, i.e. in static environment, the request for the map of environment does not represent significant restriction for localization application. Moreover, this robot can be used for implementation and demonstration of various control and localization algorithms [5].

One of the main features of our custom made robot is its capability to move in any direction, regardless of its orientation, which is possible with the help of special mecanum wheels that are used in robot construction [6]. The heading and the course of the robot do not need to achieve same values and it not has any restrictions on their movements. The robot is equipped with the encoder sensors mounted on every wheel. These sensors provided the odometry measurements that were used both in odometry localization and landmark based localization. Besides encoders, the robot is equipped with Microsoft Kinect camera which represents CCD camera and 3D depth sensor combined together in one device. The possibility of acquiring the image of environment together with the distance of every point on the image from the camera, made this device very popular in robotic applications. Here, the camera was used to get the necessary measurement data that are used in correction phase of EKF algorithm. After acquiring an image of the environment it is possible to detect the landmarks and calculate the updated location of the robot. The robot that was used in the paper is shown in Figure 1.

The paper is organized as follows. Localization algorithms that are implemented in the paper are described in Section II. Mathematical models of holonomous mobile robot and of camera are represented in Section III. Section IV describes process of detecting the landmarks from robot's environment while Section V deals with the experimental results.

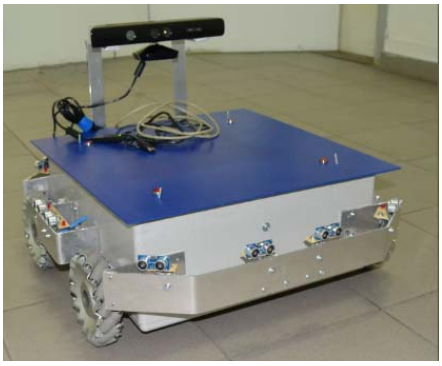

Figure 1. Holonomous mobile robot HOLBOS

II. LOCALIZATION ALGORITHMS

Localization is the problem of estimating a robot's coordinates in an external reference frame from sensor data, using a map of the environment [8]. In typical indoor environment with robot moving on a flat surface, localization represents estimation of position (x, y) and orientation (θ) of the robot. Techniques that were used in this paper are odometry localization and localization based on landmarks and using an EKF.

A. *Odometry localization*

The main requirements for odometry localization are measurements of angular velocities of the robot's wheels. These measurements are usually collected periodically from encoders that are mounted on a robot's wheels. Based on these measurements and by using mathematical model of the robot, it is possible to get robot's coordinates at every time instant. To be able to locate the robot in this way, initial location of the robot must also be provided. Robot's position and orientation is then calculated by using equations (1)-(12).

B. *Landmark based localization*

Landmarks represent features in robot's environment which can be detected by sensors on the robot. In general, landmarks have fixed and known position relative to the robot. In this paper, passive round shaped landmarks that were coloured in unique colours, so they could be detected by the robot were used. For detection of the landmarks and the distance measurement from robot to every landmark, Microsoft Kinect camera was used. Information collected in this way can be fused with odometry measurements in order to improve the accuracy of robot's location. To fuse the data acquired from various sensors, extended Kalman filter algorithm was used.

C. *Extended Kalman filter*

There are two steps in extended Kalman filter algorithm: prediction and correction step. Prediction step is similar to odometry localization procedure. Inputs to the system, in form of wheels angular velocities are measured and mathematical model of the system is then used to predict the states of the system (coordinates of the robot) in next time instant k. In correction step, additional measurements are collected and used to correct the values calculated in prediction step. Main requirements for extended Kalman filter are state and measurement equations for system:

$$x_t = g(u_t, x_{t-1}) + \varepsilon_t \tag{1}$$

$$y_t = h(x_t) + \delta_t \tag{2}$$

Extended Kalman filter algorithm that was used in the paper can be written as [8]:

$$\bar{x}_t = g(u_t, x_{t-1}) \tag{3}$$

$$\bar{P}_t = G_t P_{t-1} G_t^T + R_t \tag{4}$$

$$K_t = \bar{P}_t H_t^T (H_t \bar{P}_t H_t^T + Q_t)^{-1} \tag{5}$$

$$x_t = \bar{x}_t + K_t (y_t - h(\bar{x}_t)) \tag{6}$$

$$P_t = (I - K_t H_t) \bar{P}_t \tag{7}$$

Equations (3) and (4) represent prediction step. In equation (5) Kalman gain is calculated, while equations (6) and (7) represent correction step of EKF algorithm.

III. MODELLING OF THE SYSTEM

As it was already mentioned in introduction, both odometry localization and localization based on robotic vision use mathematical model of robot in order to calculate current position and orientation of the robot. For that reason, mathematical model that was used for algorithms implementation in this paper will be stated in this section. State equations, in form of kinematic model of the robot, will be derived first. In the second part of the section, measurement equations will be stated.

A. *Kinematic modelling of holonomous mobile robot*

As a starting point for model derivation, the results published in [14] were used.

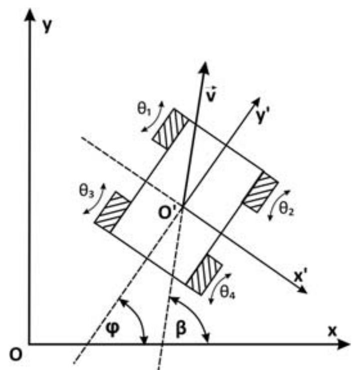

Figure 2. Global and local coordinate system of the robot [14]

Kinematic model of holonomous mobile robot in global coordinate system xOy (Figure 2.) can be expressed using the following equation:

$$x(k+1) = x(k) + v_x \Delta t \tag{8}$$

$$y(k+1) = y(k) + v_y \Delta t \tag{9}$$

$$\varphi(k+1) = \varphi(k) + \omega \Delta t \tag{10}$$

The next step in model deriving is to determine the x and y components of the linear velocity as well as the angular velocity of the robot in every time instant k. The starting point for determining those expressions is the equation for linear velocity of every wheel in coordinate system that is attached to the robot ($x'O'y'$ frame in Figure 2).

The linear velocity of each wheel in coordinate system $x'O'y'$ can be expressed as [15]:

$$v_i^{'} = \operatorname{sgn}(\dot{\theta}_i) K_i r \dot{\theta}_i \tag{11}$$

where K_i is the wheel coefficient, r is diameter of the wheel and $\dot{\theta}_i$ is angular velocity of the wheel

It is now possible to decompose linear velocity of the wheels (11) along x' and y' axis of the $x'O'y'$ coordinate system, which yields:

$$v_{ix}^{'} = \operatorname{sgn}(\dot{\theta}_i) K_i r \dot{\theta}_i \cos(\alpha) \tag{12}$$

$$v_{ix}^{'} = \operatorname{sgn}(\dot{\theta}_i) K_i r \dot{\theta}_i \sin(\alpha) \tag{13}$$

where α represents the angle of the rollers relative to the axis of rotation of the wheel.

The total linear velocity components of mobile robot along x' and y' axis of the $x'O'y'$ coordinate system are sums of velocity components for all wheels. The total linear velocities along x' and y' axis can be expressed as:

$$v_x^{'} = \sum_{i=1}^{4} v_{ix}^{'} = \sum_{i=1}^{4} \operatorname{sgn}(\dot{\theta}_i) K_i r \dot{\theta}_i \cos(\alpha) \tag{14}$$

$$v_y^{'} = \sum_{i=1}^{4} v_{iy}^{'} = \sum_{i=1}^{4} \operatorname{sgn}(\dot{\theta}_i) K_i r \dot{\theta}_i \sin(\alpha) \tag{15}$$

Equations (14) and (15) are written relative to the origin of coordinate system $x'O'y'$. Now it is necessary to transform those equations into equations relative to origin of global coordinate system xOy. Linear velocities along x and y axis in global coordinate system xOy can be written as:

$$v_x = \sin(\varphi) \sum_{i=1}^{4} v_{ix}^{'} + \cos(\varphi) \sum_{i=1}^{4} v_{iy}^{'} \tag{16}$$

$$v_y = \cos(\varphi) \sum_{i=1}^{4} v_{ix}^{'} + \sin(\varphi) \sum_{i=1}^{4} v_{iy}^{'} \tag{17}$$

After linear velocities, it is necessary to determine angular velocity of the robot. Considering the fact that all components that contribute to angular velocity have to be taken into consideration, it is convenient to split all the wheels of the robot into four groups, as shown in Figure 3.

The robot has nonzero angular velocity if there is any difference in velocity between the wheels on left and on the right side (if the robot is split along y axis), or if there is any difference in velocity between the wheels on top and on bottom side (if the robot is split along x axis). Consequently, contribution to angular velocity from every group of wheels is described as follows:

$$\begin{aligned} v_{yL} &= v_{1y} + v_{3y} & v_{xD} &= v_{3x} - v_{4x} \\ v_{yR} &= v_{2y} + v_{4y} & v_{xD} &= -v_{1x} + v_{2x} \\ \Delta\theta_y &= \frac{v_{yL} - v_{yR}}{Dx} & \Delta\theta_x &= \frac{v_{yD} - v_{yU}}{Dy} \end{aligned} \tag{18}$$

$$\omega = \Delta\theta = \Delta\theta_x + \Delta\theta_y = \frac{v_{yL} - v_{yR}}{Dx} + \frac{v_{yD} - v_{yU}}{Dy} \tag{19}$$

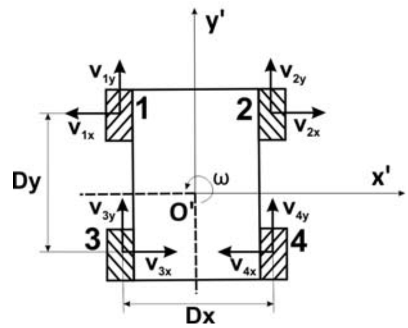

Figure 3.Wheel velocities

With angular velocity of the robot derived, all needed components of the kinematic model are determined and can be used in algorithm implementation.

B. Measurement equations

Additional measurements in the systems are coordinates of every detected landmark on the image together with the distance from camera to every landmark. Measurement vector can be written as $y = [u\ v\ d]^T$, where:

- u is horizontal coordinate of landmark on the image,
- v is vertical coordinate of landmark on the image,
- d distance from the camera to landmark.

Measurement equations in EKF algorithm are used to calculate expected values of measurement, based on robot's coordinates that were calculated in prediction step. Camera assembly, together with robot coordinates and coordinate systems are shown in Figure 4. The camera is mounted on a robot in the way that optical axis of the camera is horizontal. Angle φ represents orientation of the robot, while θ represents the angle between the camera and robot. In this case θ angle is equal to zero, since the camera axis is aligned with robot heading.

Robot is moving in $X_sO_sY_s$ plane of the global coordinate system, shown in Figure 4. The position of the robot is $x = [x_R\ y_R\ \theta_R]^T$ at every time instant k. The translation from camera to robot's coordinates is given by $T = [T_x\ T_y\ T_z]^T$.

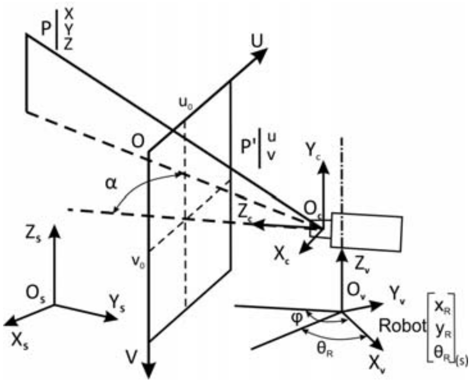

Figure 4.Camera assembly [15]

The equations that provide image coordinates $\begin{bmatrix} u & v \end{bmatrix}$ for every scene point $\begin{bmatrix} x & y & z \end{bmatrix}$ are [15]:

$$u = \alpha_u \frac{\left(\left(x - x_R\right)\sin(\varphi) - \left(y - y_R\right)cos(\varphi) + T_y\right)}{\left(\left(x - x_R\right)cos(\varphi) + \left(y - y_R\right)\sin(\varphi) - T_x\right)} + u_0 \quad (20)$$

$$v = \alpha_v \frac{\left(T_z - z\right)}{\left(\left(x - x_R\right)cos(\varphi) + \left(y - y_R\right)\sin(\varphi) - T_x\right)} + v_0 \quad (21)$$

where $\alpha_u, \alpha_v, u_{0,} v_0$ are camera model parameters which can be obtained as in [16]. The last equation in measurement equations vector provides distance from the camera to every detected landmark:

$$d = \sqrt{(x_R - x_0)^2 + (y_R - y_0)^2} \quad (22)$$

where $\begin{bmatrix} x_0 & y_0 \end{bmatrix}$ represents the coordinates of the detected landmark in robot's environment. Equations (20), (21) and (22) are measurement equations that were used for realization of EKF algorithm. In the next section, landmark detection procedure will be described.

IV. LANDMARKS DETECTION

Landmark detection is a process that was conducted in several steps. First step in the process is an image acquisition. Images are acquired and processed in real time. One of the images that presents the robot's environment is shown in Figure 6. The landmarks are placed in the lower part of the image, near the floor on which the robot is moving on. Therefore it was not necessary to process the whole image, but only the part with the landmarks. The part of the image that was being processed is shown in Figure 7.

The steps in image processing are conducted in following order:

- Image segmentation,
- Connected components labelling,
- BLOB analysis,
- Colour identification.

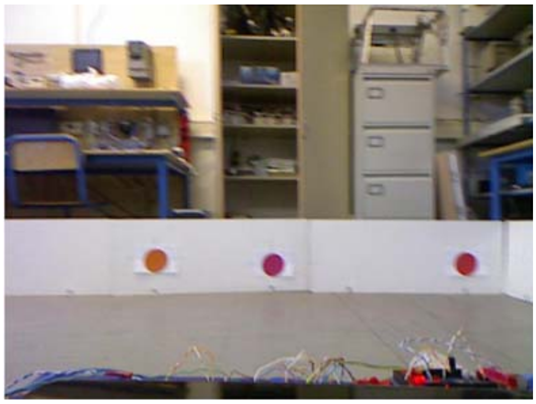

Figure 5. Image acquired from camera

Figure 6. Part of image that was processed

A. Image segmentation

Image segmentation is conducted in order to extract the parts on the image that are of interest [17]. Since the image shown in Figure 6 is relatively simple (coloured landmarks on white background), one of the simplest segmentation methods is applied – thresholding method. In order to use this segmentation method, colour image had to be converted into a grayscale image. All the pixels in image that are greater of threshold value are considered as background, while the pixels that have a value smaller then threshold are considered as objects. As a result of this method, binary image of an environment is obtained (Figure 7.).

Figure 7. Binary image – the result of image segmentation

B. Connected components labelling

The connected components labelling (CCL) represents the process of assigning the unique label to every group of pixels in binary image. There are several algorithms for labelling the components: two pass and multiple pass algorithms [18]. The program package Matlab already have a 'bwlabel' function that conducts the labelling of connected components.

C. BLOB Analysis

Methods that are used to get additional information about every group of pixels in the image belong to an area that is called BLOB analysis. This analysis is conducted when there is interest in classifying the pixel groups based on their properties. Analysis also can provide statistical data, such as the size, number, location, etc. of all groups of pixels.

Properties that are used in this paper are:

- 'Area' – total number of pixels in a region.
- 'Centroid' – coordinates of centre of mass of a region.

- 'Extent' – scalar value which represents the ratio of a number of white pixels and total numbers of pixels in a box that surrounds the group.

Figure 8. shows the image with all analysed groups of pixels. Blue dots in the Figure 8. represent the centre of mass of every group of pixels. The rectangles surrounding the groups of pixels are not necessary for the landmark detection, but are drawn just to denote an area for every group.

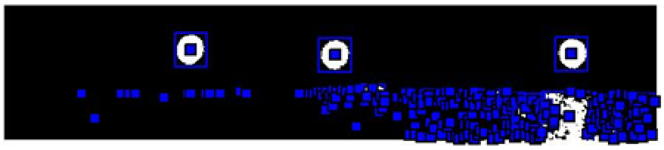

Figure 8.Analysed groups of pixels

As it can be seen in Figure 8, there are three landmarks and many other groups of pixels that are denoted which belong to the background of the image. The landmarks are extracted based on known minimum area and extent values. In order for a group of pixels to be denoted as a landmark, it has to have certain area and extent value. After extracting the true landmarks from the image, the coordinates of all detected landmarks are accessible. Figure 9 shows the image with detected landmarks.

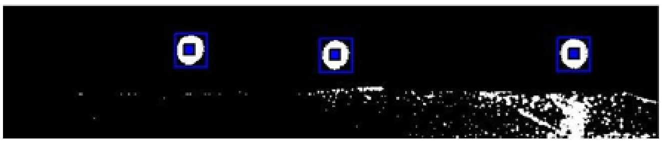

Figure 9. Detected groups of pixels after elimination of all non landmark groups

D. *Colour identification*

After extracting the parts of the image which belong to landmarks, the next step is to detect the colours of each landmark. The procedure that was used in the paper is to calculate the median value of colour pixels inside detected regions. Since Red Green Blue (RGB) values for every landmark are known, it is necessary to compare median value of detected landmark to the colour of every landmark. If the colour of detected landmark is satisfactorily close to colour of any of the landmarks than the closest colour is assigned as true colour for detected landmark. Figure 10. shows an image with landmarks coloured in median colour values.

Figure 10. Landmarks with median values of colour

V. Experimental Results

In order to demonstrate the work of localization algorithms, two experiments were conducted. In every experiment the robot is moving on a flat surface in an environment with dimensions of 414 x 310 cm. In first experiment, the robot is moving along a square shaped path in counter-clockwise direction. The paths of the robot consist of three forward movements and two rotations. The robot was programmed to go in one direction for certain amount of time, and then to rotate for some time in counter-clockwise direction. This pattern was repeated three times first experiment. Before the robot started moving, the coordinates of the robot were measured. Also, after every robot's maneuver, the coordinates were measured and recorded for future comparison.

In second experiment, the robot moves sideways to the left side. In these two experiments, holonomous properties of the robot were demonstrated.

Two graphs will be shown for every experiment: robot trajectory graph and coordinate errors graph. Measured and calculated coordinates of the robot in key points of its trajectory will be shown in the tables after the graphs.

A. *Experiment I*

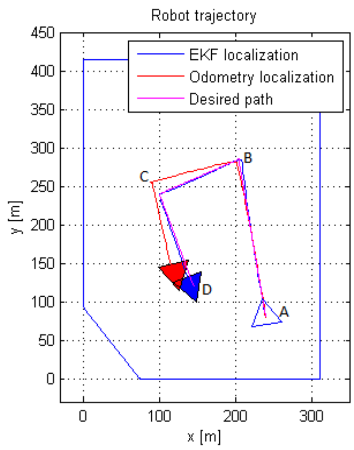

Figure 11. Robot trajectory

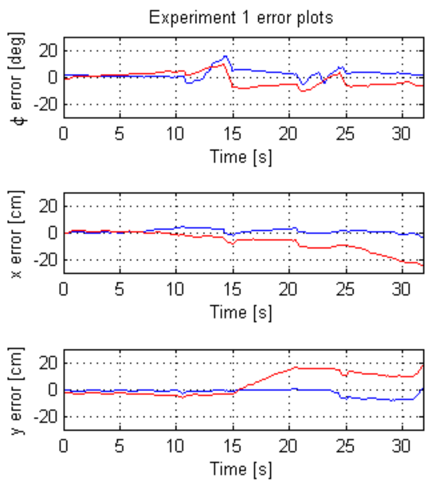

Figure 12. Coordinates errors

TABLE I. Values from odometry localization

Coordinates	Values from odometry localization			
	A	***B***	***C***	***D***
x [cm]	239.1	200.30	89.71	120.30
y [cm]	81.82	281.90	255.70	137.74
φ [deg]	105.00	195.00	280.35	282.43

TABLE II. VALUES FROM LANDMARK LOCALIZATION

Coordinates	Values from landmark based localization			
	A	*B*	*C*	*D*
x [cm]	239.1	206.3	99.88	140.80
y [cm]	81.82	284.70	239.4	121.40
φ [deg]	105.00	208.00	289.12	291.72

TABLE III. VALUES FROM ODOMETRY LOCALIZATION

Coordinates	Values from odometry localization			
	A	*B*	*C*	*D*
x [cm]	239.1	200.30	89.71	120.30
y [cm]	81.82	281.90	255.70	137.74
φ [deg]	105.00	195.00	280.35	282.43

B. Experiment II

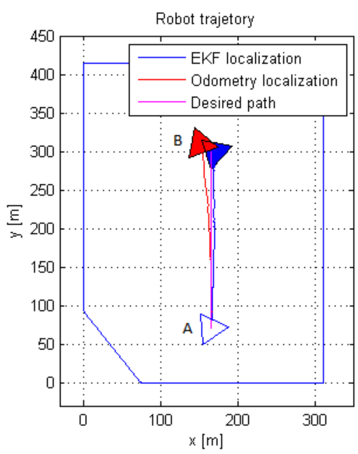

Figure 13. Robot trajectory

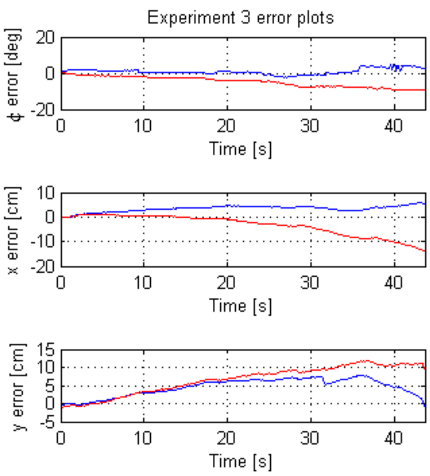

Figure 14. Coordinates errors

TABLE IV. MEASURED COORDINATES VALUES

Coordinates	Measured coordinate values	
	A	*B*
x [cm]	165	168
y [cm]	70	300
φ [deg]	0	15

TABLE V. VALUES FROM LOCALIZATION ALGORITHMS

Coordinates	Odometry		EKF	
	A	*B*	*A*	*B*
x [cm]	165	151.45	165	170.29
y [cm]	70	309.39	70	298.94
φ [deg]	0	-9.68	0	18.41

REFERENCES

[1] W. Burgard, A. B. Cremers, D. Fox, D. Hahnel, G. Lakemeyer, D. Shulz, W. Steiner, and S. Thrun, „The Interactive Museum Tour – Guide Robot“, Proceeding of the Fifteenth National Conference on Artificial Intelligence, Madison, Wisconsin, 1998.

[2] S. Thrun, M. Beetz, M. Bennewitz, W. Burgard, A. B. Cremers, F. Dellaert, D. Fox, D. Hahnel, C. Rosenberg, N. Roy, J. Schulte, D. Schulz, „Probabilistic Algorithms and the Interactive Museum Tour – Guide Robot Minerva“, The international journal of robotics research, November 2000..

[3] M. Flipse, „Altering and Improving Kiva, Some suggestions for improvement of the current Kiva system“, unpublished.

[4] R. Negeborn, „Robot Localization and Kalman Filters, On finding your position in a noisy world“, Utrecht University, September 1, 2003.

[5] J. Velagic, A. Kaknjo, F. Dautovic, M. Hujdur and N. Osmic, "Design and Physical Implementation of Holonomous Mobile Robot - Holbos," in book: *Interdisciplinary Mechatronics: Research Development* (Eds. M. Habib & J. P. Davim), Wiley, 2013.

[6] M. Blackwell, “The URANUS Mobile Robot”, tech. Report CMU-R-TR-91-06, Robotics Institute, Carnegie Mellon University, Pittsburgh, September 1990.

[7] Y. Fuke and E. Krotkov, „Dead Reckoning for a Lunar Rover on Uneven Terrain“, The Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, Proceedings of the 1996 IEEE, International Conference on Robotics and Automation, Minneappolis, Minnesota, April 1996.

[8] S. Thrun, W. Burgard, D.Fox, “Probabilistic Robotics (Intelligent Robotics & Autonomous Agents Series), MIT Press, 2005.”

[9] F. Dellaert, D. Fox, W. Burgard, S. Thrun, „Monte Carlo Localization for Mobile Robots“,Dept. of Comput. Sci., Carnegie Mellon Univ., Pittsburgh, PA, USA, Robotics and Automation, 1999. Proceedings. 1999 IEEE International Converence on Robotics and Automation.

[10] S. Thrun, D. Fox, W. Bugard, and F. Dellaert, „Robust Monte Carlo Localization for Mobile Robots“, School of Computer Science, Carnegie Mellon University, Pittsburgh, April 2000..

[11] H. Andreasson, A. Treptow and T. Duckett, „Localization for Mobile Robots using Panoramic Vision, Local Features and Particle Filter“, Proceedings of the 2005 IEEE International Conference on Robotics and Automation, Barcelona, Spain, April 2005.

[12] J. M. Drumheller, „Mobile Robot Localisation Using Sonar“, Massachusetts Institute of Technology, Artificial Intelligence Laboratory, January 1985.

[13] Tobias Einsele, „Localization in Indoor Environments Using a Panoramic Laser Range Finder“, Ph.D. dissertation Lehrstuhl fur Realzeit-ComputerSysteme, Technischen Universitat Munchen.

[14] N.Tlale, M. De Villiers, “Kinematics and Dynamics Modelling of a Mecanum Wheeled Mobile Platform”,15th International Conference on Mechatronics and Machine Vision in Practice, 2008. M2VIP 2008.

[15] F. Chenavier, J.L. Crowley, “Position estimation for a mobile robot using vision and odometry”, 1992 IEEE International Conference on Robotics and Automation, 1992. Proceedings.

[16] A. Fetic, D. Juric, D. Osmankovic, “The procedure of camera calibration using Camera Calibration Toolbox for MATLAB”, MIPRO, 2012 Proceedings of the 35th International Convention

[17] S.Raut, M. Raghuvanshi, R. Dharaskar, A. Raut “Image Segmentation – A state Of Art Survey for Prediction, Conference on Advanced Computer Control, 2009. ICACC ‘09”.

[18] C. Grana, D. Borghesani, R. Cucchiara, “Fast block based connected components labeling”, 16th IEEE International Conference on Image Processing (ICIP), 2009.