

Extended Kalman Filter Sensor Fusion in Practice for Mobile Robot Localization

Alaa Aldeen Housein, Gao Xingyu*, Weiming Li, Yang Huang

School of Electrical and Mechanical Engineering, Guilin University of Electronic Technology, Guilin, China

Abstract—Self-driving vehicles and autonomously guided robots could be very beneficial to today's civilization. However, the mobile robot's position must be accurately known, which referred as the localization with the task of tracking the dynamic position, in order for the robot to be active and useful. This paper presents a robot localization method with a known starting location by a real-time reconstructed environment model that represented as an occupancy grid map. The extended Kalman filter (EKF) is formulated as a nonlinear model-based estimator for fuse Odometry and a LIDAR range finder sensor. Because the occupancy grid map for the area is provided, just the inaccuracies of the LIDAR range finder will be considered. The experimental results on the “turtlebot” robot using robot operating system (ROS) show a significant improvement in the pose of the robot using the Kalman filter compared with sample Odometry. This paper also establishes the framework for using a Kalman filter for state estimation, providing all relevant mathematical equations for differential drive robot, this technique can be used to a variety of mobile robots.

Keywords—Autonomous navigation; kalman filter; self-driving vehicle; simultaneous localization and mapping; occupancy grid map; ROS

I. INTRODUCTION

Recently, mobile robots have been used to perform specialized tasks in a number of industries, including services, rescue, military, disaster relief, unmanned defense vehicles, and so on. Localization of mobile robots is a hard problem that many academics are seeking to address via the development of innovative techniques. Researchers have added more sources in order to build a powerful localization approach [1]. Odometry is one of the most important techniques to tackle the posture tracking problem, it uses encoder data to track the motion progress from a specified beginning position. This technique tracks motion from a known beginning position using encoder data, the encoded data is sent to the central processor, which uses a geometric equation to update the robot's position [2, 3]. Due to a variety of factors such as wheel slippage, ground roughness, and varying wheel diameters, this approach has accumulative errors. So, under severe conditions, solely utilizing Odometry for localization virtually never results in an accurate state, and it becomes more active when other sources of sensing are used. Stereo, LIDAR range finder, sonar, compass, gyro, and GPS are the most often utilized extra sensors [4]. Building a robot from the scratch is expensive and time consuming, so working in an environment that guarantees theoretical study and practical implementation will be quiet helpful. ROS is an open source meta operating system that provides hardware abstraction,

control implementation of commonly used functionalities, tools and libraries for building, writing, and running code for simulated and real robots after installing necessary drivers. The significance of this research rests in the framework it provides for fusing several sensors with a Kalman filter for robot localization, with the experimental results emphasizing notable reduction of errors in robot position. The paper is structured as follow: in section III the motion model of two wheeled robot was derived. Section IV presents Kalman filter pose tracking design. Section V discusses Kalman filter implementation results, followed by conclusion in section VI.

II. LITERATURE REVIEW

Iraj Hassanzadeh and Mehdi Abedinpour implemented an augmented unscented and extended Kalman filter for position tracking using a differential drive mobile robot with encoder readings, assuming real measurements are available. The work showed an improvement in pose tracking using this technique with the unscented filter outperforming the extended one [5]. Jaeyong Park and Sukgyu Lee investigated a mobile robot SLAM (simultaneous localization and mapping) technique based on EKF extended Kalman filter, with an additional extended Kalman filter used to enhance robot heading accuracy, because the robot's kinematic model was unclear due to the rough surface, its heading was deviated as it drove across uneven terrain. They proposed a method for correcting uncertain robot postures utilizing an extra extended Kalman filter on a simulation-based test [6]. The Kalman filter was applied on a Pioneer 2DX mobile robot by Edouard Ivanjko, Mario Vasak, and Ivan Petrovic, combining data from wheel encoders and a sonar sensor. The experimental result indicates that the kalman filter reduces posture tracking errors when compared to using Odometry alone [7]. Yusuke Misono and Yoshitaka Goto have implemented outdoor SLAM using Kalman filter in an outdoor environment using electric wheelchair mobile robot provided by LIDAR range finder and GPS the experimental results reveal a significant improvement of self-localization vehicle estimate by the SLAM algorithm compared with dead reckoning [8]. On a laser range finder-equipped robot platform, Angga Rusdinar, and Sungshin Kim investigated simultaneous localization and mapping with a particle filter. The experiment's findings showed that the suggested particle filter can improve map building performance and mobile robot localization accuracy inside indoor buildings [9]. Mohammed Faisal, Mansour Alsulaiman, and Ramdane Hedjar Hassan developed a localization method to reduce error accumulation in the dead-reckoning approach, since dead-reckoning is reliant on encoder information, they use an additional sensing source, the proposed localization

*Corresponding Author.

system uses the extended Kalman filter in combination with infrared sensors to enhance the mobility robots' localization, by rectifying errors in the robot's position and address the issue of dead-reckoning, with the working area's walls serving as references (landmarks)[10]. Yi-Xiang Wang and Ching-Lung Chang investigated SLAM under the robot operating system (ROS) utilizing a laser range finder (LIDAR), an Inertial Measurement Unit (IMU), an odometer, and an Ultra-wideband (UWB), and fused all of the above sensors using Extended Kalman filter. Experiment findings demonstrate that the mobile robot's average error distance in the system is restricted to 10cm [11]. For city navigation, Zanwu Xia and Si Tang developed a new technique to improve the accuracy of high definition (HD) maps in order to improve the localization of self-driving cars. The research focused on extracting the factors that have a high impact on the global map, such as feature sufficiency, layout, local similarity, and map representation quality. The Kalman filter was used to combine data from LIDAR, IMU, and GNSS systems. The experimental results show a reduction in accumulative errors [12].

In this study, we focus on the problem of indoor robot localization, which involves determining the position of the robot x , y , and its orientation Θ . The idea of the Kalman filter is to reduce the errors in both the mechanical model of the robot and the sensor readings. Kalman Filter is designed to deal with linear systems, but most nontrivial systems are nonlinear. Therefore, a new modified technique called extended Kalman filter (EKF) has been developed. This paper aims to deal with the uncertainties of a mobile robot by fusing Odometry and LIDAR range finder in the dead-reckoning method. Three steps are encoded in the method proposed here; the prediction step depends on the motion equation of the robot platform. Data acquisition by measurement sensors in the workspace, which are used to correct the robot position calculated in the motion prediction step. Update step that corrects the sum of motion uncertainty and measurement uncertainty, Fig. 1.

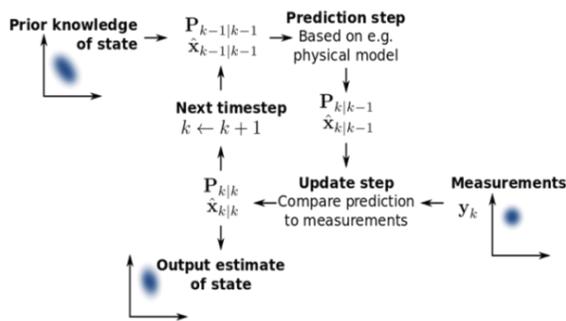


Fig. 1. Kalman Filter Estimator.

III. MOTION MODEL

Regardless of whether robot position has to be adjusted or not, a kinematic model of robot motion exists that is dependent on the degree of freedom available to it. For example, a wheeled mobile robot without a manipulating arm (our robot platform) has three degrees of freedom (displacement along X-axis, displacement along Y-axis, and the orientation around Z-axis). Fly robot for instance has six

degree of freedom (beside to displacement along orthogonal axis's X Y Z, there is orientation around each axis $\Theta_x \Theta_y \Theta_z$). In the experiment, a two-wheeled mobile robot was used. Each wheel has encoders placed, besides the passive caster wheel for stability. The driving wheels are individually controlled. The following relationships define the mobile robot's kinematic model (Fig. 2).

$$x_{k+1} = x_k + d_k \cos \theta_k \quad (1)$$

$$y_{k+1} = y_k + d_k \sin \theta_k \quad (2)$$

$$\theta_{k+1} = \theta_k + \Delta \theta_k \quad (3)$$

$$d_k = v_{t,k} \cdot T \quad (4)$$

$$\Delta \theta_k = w_k \cdot T \quad (5)$$

$$v_{t,k} = \frac{v_{l,k} + v_{r,k}}{2} = \frac{w_{l,k} \cdot R + w_{r,k} \cdot R}{2} \quad (6)$$

$$w_k = \frac{v_{r,k} - v_{l,k}}{b} = \frac{w_{r,k} \cdot R - w_{l,k} \cdot R}{b} \quad (7)$$

Where x_k and y_k are the center gravity of robot platform ; d_k travel distance between two successive time interval $k+1$ and k ; $v_{t,k}$ mobile robot's translational velocity ; T is the sampling time ; θ_k the robot's heading with the X-axis; $\Delta \theta_k$ rotational angle of robot between $k+1$ and k time steps ; $v_{l,k}$ and $v_{r,k}$ the left and right wheel's respective linear velocities ; $w_{l,k}$ and $w_{r,k}$ the left and right wheel's respective angular velocities ; The two driving wheels have a radius of R ; b axle length or robot. The radius of both driving wheels, in sampling, is assumed to be equal. We add three more variables to (6) and (7) to account for sampling errors caused by not knowing the exact radius and axle length.

$$v_{t,k} = \frac{v_{l,k} + v_{r,k}}{2} = \frac{w_{l,k} \cdot R + w_{r,k} \cdot R}{2} \quad (8)$$

$$w_k = \frac{v_{r,k} - v_{l,k}}{b} = \frac{w_{r,k} \cdot R - w_{l,k} \cdot R}{b} \quad (9)$$

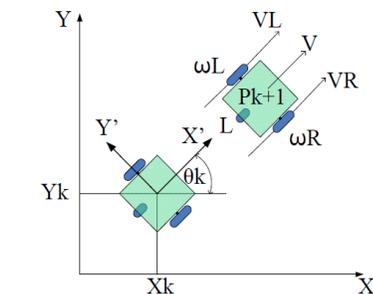
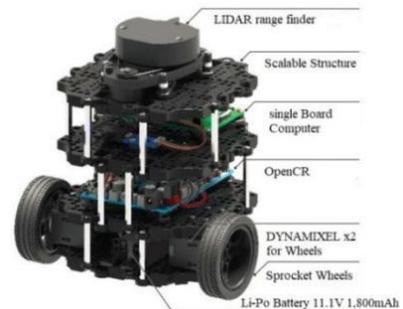


Fig. 2. Real Robot and its Kinematic Model.

The uncertainty of the exact wheel radius is represented by k_1 and k_2 , while the uncertainty of the exact axle length is represented by k_3 . In [13, 14, 15, 16] the systematic error correction approach is described in depth, as well as how the parameter values were calculated. Equations (1) to (7) describe the fundamental concept of Odometry position tracking.

IV. LOCATION TRACKING WITH KALMAN FILTER

Although there are many specific techniques for estimating the state of the system from, a set of measurements, most of these do not explicitly consider the noisy nature of measurements. This noise is typically described by statistics, which leads to have to use stochastic methods to tackle the problem [17, 18]. This section describes the stochastic state estimation process, initially, the basics of Bayesian filtering are presented, providing a brief mathematical derivation of how it is possible to make an estimate of the state. A mathematical treatment of the Kalman Algorithm for localization is then presented.

The difficulty in utilizing sensor fusion to localize a mobile robot is balancing the uncertainty of the state (x , y and Θ) with the LIDAR range measurement (robot's output) to achieve an optimum estimation of the posture. Kalman filter requires that the state random variables have Gaussian probability distributions that are adequately characterized by the mean and covariance [19, 20]. Time Update and Measurement Update are the two major stages in calculating the optimum state estimate. The state prediction is generated using the motion model based on the previous value and the control input value. The output forecasts are calculated using the measurement model based on the outcomes of the time update. The anticipated state mean and covariance are then adjusted by reducing the state covariance using the difference between

Expected and measured output.

A. Bayesian Filtering

A Bayesian filter is a mathematical tool that estimates the development of the system's state given the available data [21]. This tool necessitates:

The analytical knowledge of the transition function f_t and the stochastic knowledge of the noise of the state v_t

- The analytical knowledge of the output function h_t and the stochastic knowledge of the observation noise w_t
- The realization of the output of the system $z_{1:t}$ at time t .

Having this data, the Bayesian filter is able to estimate the function probability density. Once the state estimation problem has been formalized as a Bayesian filtering problem, we have to find a mathematical formulation that allows us to return to the probability density function $p(x_{0:t}|z_{1:t})$ using our system knowledge and observations. First, we have to model the system's internal evolution as well as how it manifests itself through its observable outputs. Two conditional distributions are introduced for this purpose. The first is observation model $p(z_t|x_t)$, represents the density of the measurement z_t given the system state x_t . The second

function $p(x_t|x_{t-1})$ is an evolution model that represents how the system develops over time. Using the Bayes rule and the Markovian chain assumption, it is possible to obtain an a posteriori probability density of the state incrementally. Because data arrives in stages over time, a recursive formulation of Bayes' rule known as the Bayes Filter is used. The a posteriori probability of the state is gradually refined in this formulation as measurements arrive.

$$p(x_{0:t+1}|z_{1:t+1}) = \frac{p(z_{t+1}|x_{0:t+1}, z_{1:t})p(x_{0:t+1}|z_{1:t})}{p(z_{t+1}|z_{1:t})} \quad (10)$$

$$= \frac{p(z_{t+1}|x_{t+1})p(x_{0:t+1}|z_{1:t})}{p(z_{t+1}|z_{1:t})} = \frac{p(z_{t+1}|x_{t+1})p(x_{t+1}|x_t)}{p(z_{t+1}|z_{1:t})} p(x_{0:t}|z_{1:t}) \quad (11)$$

If we are interested in the iterative distribution the equation becomes:

$$p(x_{t+1}|z_{1:t+1}) = \frac{p(z_{t+1}|x_{t+1})p(x_{t+1}|z_{1:t})}{p(z_{t+1}|z_{1:t})} \quad (12)$$

$$= \frac{p(z_{t+1}|x_{t+1})p(x_{t+1}|x_t)}{p(z_{t+1}|z_{1:t})} p(x_{0:t}|z_{1:t}) \quad (13)$$

$$= \frac{p(z_{t+1}|x_{t+1}) \int p(x_{t+1}|x_t)p(x_t|z_{1:t})dx_t}{\int p(z_{t+1}|z_{1:t}, x_{t+1})p(x_{t+1}|z_{1:t})dx_{t+1}} \quad (14)$$

$$= \eta p(z_{t+1}|x_{t+1}) \int p(x_{t+1}|x_t)p(x_t|z_{1:t})dx_t \quad (15)$$

Where η is a normalization factor to ensure that equation (15) represents a probability density function. Usually, the evaluation of this equation takes place in two steps. In the prediction phase, the x_{t+1} state is calculated starting from the x_t state, through the application of the transition model. Subsequently, the z_t observation is incorporated into the previously calculated probability density function, through the updating phase. The relationship between the present state and the prior state is depicted by the motion model. The location of the mobile robot in a global coordinate frame is defined as the state vector; $X_k = [x_k, y_k, \Theta_k]^T$, where, k indicates the sampling moment. The probability distribution is assumed to be Gaussian so that the state variable is fully described by a 3×3 covariance matrix P_k and the state expected value \hat{x}_k (Mean estimated value). The control input u_k represents the motion command that moves the mobile robot from step k to step $k + 1$, since u_k can be expressed as; $u_k = [d_k, \Delta\Theta_k]^T$ denotes translation by distance d_k followed by a rotation angle $\Delta\Theta_k$. The state vector is calculated using the current state vector and the current control input by the state transition function $f(\cdot)$; $x_{k+1} = f(x_k, u_k, V_k)$. Since $V_k = [v_{1,k}, v_{2,k}]^T$ depicts process noise (also Gaussian with zero mean) $E(V_k) = [0,0]^T$, according to equations (1), (2), and (3) the state transition function is obtained.

$$f(x_k, u_k, V_k) = \begin{pmatrix} x_k + (d_k + v_{1,k}) \cdot \cos(\Theta_k + \Delta\Theta_k + v_{2,k}) \\ y_k + (d_k + v_{1,k}) \cdot \sin(\Theta_k + \Delta\Theta_k + v_{2,k}) \\ \Theta_k + \Delta\Theta_k + v_{2,k} \end{pmatrix} \quad (16)$$

Two independent error sources, translational and angular, were used to model the process noise covariance Q_k . The expression for Q_k is:

$$Q_k = \begin{pmatrix} \sigma_D^2 & 0 \\ 0 & \Delta\theta_k^2 \sigma_{\Delta\theta}^2 \end{pmatrix} \quad (17)$$

Where σ_D^2 and $\sigma_{\Delta\theta}^2$ are the variances of translation d_k and rotation $\Delta\theta_k$ respectively.

B. Measurement Model

Measurement function $h_i(X, p_i)$ determines the distance between the obstacles and robot's LIDAR (Fig. 3).

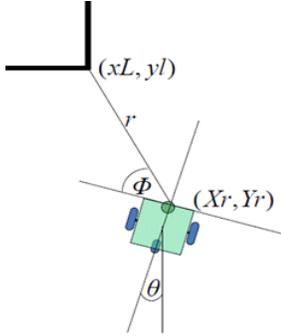


Fig. 3. Robot Pose from the Landmark.

In the world model, $p_i = (x_i, y_i)$ represents the point (occupied cell identified by the LIDAR). The LIDAR model utilizes distance readings that are linked to the causative obstacle.

$$z_{i,k} = h_i(X_k, p_i) + w_{i,k} \quad (18)$$

The measurement noise is represented by $w_{i,k}$. Because all distance measurements are utilized in parallel for the distance measurement value, the distance measurements $z_{i,k}$ are a single measurement vector. And r_k^i and ϕ_k^i components of the measurement form a diagonal matrix R_k .

C. EKF-based Pose Tracking

This paper describes EKF, as sensor fusion-based method for robot posture tracking. More a thorough discussion of the EKF localization method can be found in [22]. At time k the values of the control input vector u_{k-1} generated by wheel encoder are supplied to the equations that will be mentioned in this section, and the first-time update is performed to get the prediction estimate, and when fresh LIDAR measurements are received, these predictions are adjusted. The prior mean \hat{X}_k^- is calculated by using the nonlinear Odometry function to propagate the predicted state.

$$\hat{X}_k^- = f(\hat{X}_{k-1}, u_{k-1}, E\{V_{k-1}\}) \quad (19)$$

The anticipated state covariance P_k^- is calculated by propagating the state covariance through a linearized system form.

$$P_k^- = \nabla f_X P_{k-1} \nabla f_X^T + \nabla f_u Q_k \nabla f_u^T \quad (20)$$

$\nabla f_X = \nabla f_X(\hat{X}_{k-1}, u_{k-1}, E\{V_{k-1}\})$ denotes the Jacobian of function $f(\cdot)$ with respect to the state X_k which can be computed as follows.

$$\nabla f_X = \begin{pmatrix} 1 & 0 & -(d_k + v_{1,k}) \sin(\theta_k + \Delta\theta_k + v_{2,k}) \\ 0 & 1 & (d_k + v_{1,k}) \cos(\theta_k + \Delta\theta_k + v_{2,k}) \\ 0 & 0 & 1 \end{pmatrix} \quad (21)$$

$\nabla f_u = \nabla f_u(\hat{X}_{k-1}, u_{k-1}, E\{V_{k-1}\})$ is the Jacobian of $f(\cdot)$ with respect to the control input u . It is important to note that when (9) and (10) are employed, the mean and covariance are only true to the first order of the associated Taylor expansion [23]. If no fresh LIDAR measurements are available at time k , or if all are discarded, there is no measurement update, and the estimated mean and covariance are assigned the anticipated values.

$$\begin{cases} \hat{x}_k = \hat{x}_k^- \\ P_k = P_k^- \end{cases} \quad (22)$$

Otherwise, a measurement update occurs, in which the initial forecasts of the approved LIDAR readings in $\hat{z}_{i,k}^-$ with the i -th component are as follows:

$$\hat{z}_{i,k}^- = h(\hat{X}_k^-, p_i) + E\{w_{i,k}\} \quad (23)$$

In time step k , the state estimate and covariance are calculated as follows:

$$\begin{cases} \hat{X}_k = \hat{X}_k^- + K_k(z_k - \hat{z}_k^-) \\ P_k = (I - K_k \nabla h_X) P_k^- \end{cases} \quad (24)$$

Since Z_k reflects actual LIDAR readings, where ∇h_X is the Jacobian matrix and we can obtain by calculating the derivative of measurement function with respect to state X_k :

$$\begin{aligned} \nabla h_X = \frac{\delta h_i(X_k, p_i)}{\delta X_k} &= \begin{pmatrix} \frac{\delta r_k^i}{\delta \hat{X}_{k,x}} & \frac{\delta r_k^i}{\delta \hat{X}_{k,y}} & \frac{\delta r_k^i}{\delta \hat{X}_{k,\theta}} \\ \frac{\delta \phi_k^i}{\delta \hat{X}_{k,x}} & \frac{\delta \phi_k^i}{\delta \hat{X}_{k,y}} & \frac{\delta \phi_k^i}{\delta \hat{X}_{k,\theta}} \end{pmatrix} \\ &= \begin{pmatrix} -\frac{p_{i,x} - \hat{x}_{k,x}}{\sqrt{q}} & -\frac{p_{i,y} - \hat{x}_{k,y}}{\sqrt{q}} & 0 \\ \frac{p_{i,y} - \hat{x}_{k,y}}{q} & -\frac{p_{i,x} - \hat{x}_{k,x}}{q} & -1 \end{pmatrix} \end{aligned} \quad (25)$$

Since:

$$q = (p_{i,x} - \hat{x}_{k,x})^2 + (p_{i,y} - \hat{x}_{k,y})^2 \quad (26)$$

And K_k is the Kalman filter which can be computed as follows:

$$K_k = P_k^- \nabla h_X^T (\nabla h_X P_k^- \nabla h_X^T + R_k)^{-1} \quad (27)$$

By implementation of previous equations, we can see the diverges in robot location in case of using Kalman filter (blue color) compared with sample Odometry (red color), as shown in Fig. 4.

After we have seen the diverges in robot pose using kalam filter compared with sample odometry and deriving the necessary equations, We will go further, to see the effect of using kalman filter in reducing estimated error the following figure (Fig. 5) depicts the implementation of EKF on a differential drive mobile robot (turtelbot) using ROS.

VI. CONCLUSION

Extended Kalman filters are developed and tested as mobile robot posture monitoring methods. Odometry and sensor fusion LIDAR-based sensors are shown to significantly improve mobile robot localization, and the experimental results reveal substantial differences when utilizing sample Odometry and the EKF method. The algorithm is tested with different motion and measurement noise covariance, generated trajectories and a varying number of landmarks. This technique produces significant results in a limited environment and with a small number of Landmarks. However, it is clear that this approach is insufficient in the case of a large number of Landmarks, which would definitely result in an increase in the size of the state space.

ACKNOWLEDGMENT

This work was supported in part by the Guangxi Innovation Driven Development Science and Technology Project (grant # Guike AA18118002-3), in part by the Guangxi Key Lab of Manufacturing System and Advanced Manufacturing Technology (grant # 19-050-44-005Z), in part by Guangxi Science and Technology Base Talent Project (grant # Guike-AD19245141), and in part by Guangxi Natural Science Foundation Program (grant # 2020GXNSF BA297157).

REFERENCES

- [1] Borenstein, J. & Feng, L. (1996). Measurement and correction of systematic Odometry errors in mobile robots. *IEEE Transactions in Robotics and Automation*, 12(2), 869-880.
- [2] C. Tarin Sauer, H. Brugger, E.P. Hofer & B. Tibken.(2002). Odometry error correction by sensor fusion for autonomous mobile robot navigation. *IEEE Instrumentation and Measurement Technology Conference*, Cat. No.01CH 37188. August 07, Budapest, Hungary.
- [3] Digiampaolo E. & Martinelli F. (2014). Mobile robot localization using the phase of passive UHF RFID signals. *IEEE Transactions on Industrial Electronics*, 61(1), 365-376. doi:10.1109/TIE.2013.
- [4] Leonardo Marín, Marina Vallés, Ángel Soriano, Ángel Valera & Pedro Albertos (2013). Multi Sensor Fusion Framework for Indoor-Outdoor Localization of Limited Resource Mobile Robots. *Sensor*, 13(10), 14133-14160. doi: 10.3390/s131014133.
- [5] Iraj Hassanzadeh & Mehdi Abedinpour (2008). Design of Augmented Extended Kalman Filter for Differential Drive Mobile Robots. *Journal of Applied Sciences*, 8(16), 2901-2906.
- [6] Jaeyong Park, Sukgyu Lee & Joohyun, Park (2009). Correction Robot pose for SLAM based on Extended Kalman Filter in a Rough Surface Environment. *International Journal of Advanced Robotic Systems*, 6(2), 67-72.
- [7] Edouard Ivanjko, Mario Vasak, & Ivan Petrovic (2005). Kalman Filter Theory Based Mobile Robot Pose Tracking Using Occupancy Grid Maps. *International Conference on Control and Automation (ICCA2005)*, June 27-29, Budapest, Hungary.
- [8] Yusuke Misono & Yoshitaka Goto (2007). Development of Laser Rangefinder-based SLAM Algorithm for Mobile Robot Navigation. *AICE Annual Conference*, September 17-20, Kagwa University, Japan.
- [9] Angga Rusdinar, Jungmin Kim & Sungshin Kim (2010). Error Pose Correction of Mobile Robot for SLAM Problem using Laser Range Finder Based on Particle Filter. *International Conference on Control, Automation and Systems*, Oct. 27-30, Gyeonggi-do, Korea.

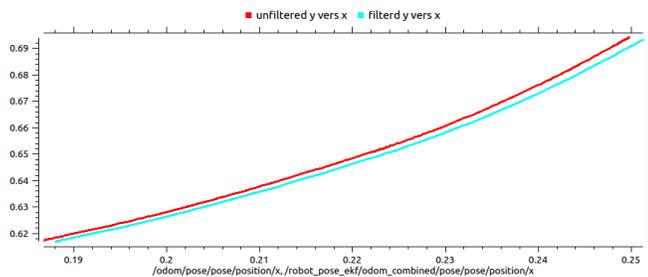


Fig. 4. Robot Ypos vers Xpos.

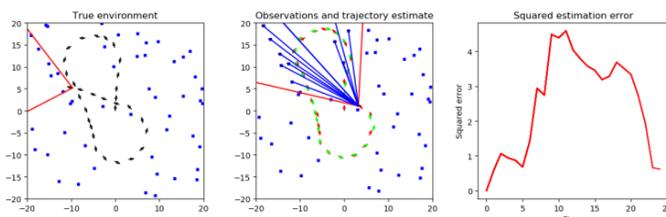


Fig. 5. True Robot Trajectory (Left), Estimated Trajectory (Middle) and Squared Positional Error (Right) using EKF Algorithm.

V. DISCUSSION

In Fig. 5 estimates after the prediction step are shown as red arrows; estimates after the update step are shown as green arrows. Blue dots represent (static) landmarks and the red lines indicate the robot's field of view. Robot state covariance increases during the prediction step and decreases during the update step. The algorithm benefits more from the correction step when the motion covariance is large, as otherwise the prediction step already yields fairly precise estimates. A larger number of landmarks result in a more precise position estimate. Loop closing, e.g., observing landmarks that have already been observed at an earlier stage greatly decreases the covariance of both the robot state and landmark position. This effect is clearly visible in Fig. 5, where loop closure occurs twice: first around step 9 and then around step 20. Both times, the squared estimation error decreased. It's worth noting that the findings displayed above were obtained using ROS in simulated settings for obstacles, LIDAR measurements, linear and angular velocities; hence, with ROS, we can safely utilize the same software for actual robots, as shown in Fig. 6.

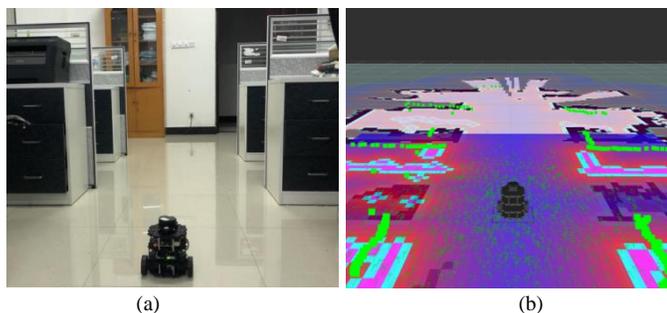


Fig. 6. (a) Robot's Environment in Laboratory (b) Green Squared Dots Represent LIDAR Measurements Readings. Turtlebot Real EKF Implementation under ROS Robot Operating System.

- [10] Mohammed Faisal & Mansour Alsulaiman (2016). Enhancement of mobile robot localization using extended Kalman filter. *Advances in Mechanical Engineering*, 8(11), 1–11. DOI: 10.1177/1687814016680142.
- [11] Yi-Xiang Wang & Ching-Lung Chang (2020). ROS-base Multi-Sensor Fusion for Accuracy Positioning and SLAM System. *International Symposium on Community-centric Systems*. Tokyo, Japan DOI: 10.1109/CcS49175.2020.9231442.
- [12] Zanwu Xia & Si Tang (2019). Robust self-localization system based on multi-sensor information fusion in city environments. *International Conference on Information Technology and Computer Application (ITCA)*. DOI: 10.1109/ITCA49981.2019.00011.
- [13] Ivanjko, E., Petrovi, I. & Peric, N. (2003). An approach to Odometry calibration of differential drive mobile robot. *International conference of electrical drives and power electronics*. Sept. 24-26, pp. 519-523.
- [14] Libal, U. & Plaskonka, J. (2014). Noise sensitivity of selected kinematic path following controllers for a unicycle. *Bulletin of the Polish Academy of Sciences: Technical Sciences*, 62(1), 3-13. DOI: 10.2478/bpasts-2014-0001.
- [15] Kozłowski, K. & Michałek M. (2012). Trajectory tracking control with obstacle avoidance capability for unicycle-like mobile robot. *Bulletin of the Polish Academy of Sciences: Technical Sciences*, 60(3), 537-546. DOI: 10.2478/v10175-012-0066-x.
- [16] Kozłowski, K. & Michałek M. (2019). Trajectory tracking and collision avoidance for the formation of two-wheeled mobile robots. *Bulletin of the Polish Academy of Sciences: Technical Sciences*, 20(19), 5915-924. DOI: 10.24425/bpas.2019.128652.
- [17] Frank E. Schneider & Dennis Wildermuth (2004). Using Extended Kalman Fiter for Relative Localization in a Moving Robot Formation. *International workshop on robot Motion and Control*, June 17-20.
- [18] NAGARAJAN M., BOSCO, J. & KANNAN R. (2017). Implementation of extended Kalman filter-based simultaneous localization and mapping: a point feature approach, *Sādhanā*, 42(9), 1495–1504. doi: 10.1007/s12046-017-0692-y.
- [19] Xifeng Li & Yongle Xie (2013). State Estimation Based on Generalized Gaussian Distributions, *Metrology and Measurement Systems*, 1, 65-76. DOI: 10.2478/mms-2013-0006.
- [20] Jonas Skeivalas, Eimuntas Paršeliūnas, Raimundas Putrimas & Dominykas Šlikas (2020). An influence of the covariance between single orbit parameters on the accuracy of observations of the pseudo ranges and phase differences. *Metrology and Measurement Systems*, 27(1), 131-140. DOI: 10.24425/mms.2020.131721.
- [21] Jeremy Kolansky & Corina Sandu (2013). Real-time parameter estimation study for inertia properties of ground vehicles. *Archive of Mechanical Engineering*, 60, 17-21. DOI: 10.2478/meceng-2013-0001
- [22] Heonmoo Kim & Yosoon Choi (2020). Comparison of Three Location Estimation Methods of an Autonomous Driving Robot for Underground Mines. *Applied Sciences*, 10, 4831. doi:10.3390/s131014133.
- [23] Housein, A. & Gao Xingyu (2021). Simultaneous Localization and Mapping using differential drive mobile robot under ROS, *Intelligent Manufacturing and Automation Technology (MEMAT)*, January 15-17, Guilin, China. doi:10.1088/1742 6596/1820/1/012015.