Trajectory Tracking Control of a Mobile Robot Using Lidar Sensor For Position and Orientation Estimation

Thiago Alves Lima, Marcus Davi do Nascimento Forte
Dr. Fabrício Gonzalez Nogueira, Dr. Bismark Claure Torrico, Adriano Rodrigues de Paula
Departamento de Engenharia Elétrica
Universidade Federal do Ceará, UFC
Fortaleza, Ceara, Brazil
Mr.Thiago.AlvesLima@ieee.org, davi2812@dee.ufc.br
fnogueira@dee.ufc.br, bismark@dee.ufc.br, rodrigues@dee.ufc.br

***Abstract*— In this paper it is investigated the trajectory tracking control of a wheeled mobile robot using a laser rangefinder sensor to estimate the robot position and orientation (posture). This technique is compared with the conventional odometry method where the encoders data are used for posture estimation. The paper also presents the description of the robot as well as its mathematical modeling. Two controllers in cascade were used to implement the tracking control. A kinematics controller generates the references for internal motor speed controllers. Experimental results in an indoor environment are presented for trajectories with no obstacles.**

I. INTRODUCTION

The position and orientation measurement of a nonholonomic mobile robot is an important challenge to overcome when the mobile robot has to follow a desired path. Estimation methods are often used to obtain these information, for example the well-known odometry method, where encoders data are used to calculate the linear and angular velocities and through the kinematics model of the robot is possible to estimate the position and orientation (posture). This simple method works only if the robot is not subjected to disturbances, such as slipping and skidding on the wheels. Therefore, the real world applicability of this technique has some limitations.

Along the last years, an intensive research effort was spent in this area, where many works about sensor fusing were published. In [4], the use of encoders in addition to accelerometer and gyroscope sensors with a Kalman filter was presented. In [11], ultrasonic, encoders and gyroscopes sensors data are fused to estimate a mobile robot localization and avoid obstacles. In [10], the use of accelerometers, gyroscopes, encoders and compass with an extended Kalman filter algorithm is presented. In [9], magnetometers and gyroscope data are fused using a Kalman filter with Fuzzy Compensation to estimate a mobile robot localization.

There are different approaches to solve this problem [3]. One can include the skidding and slipping mathematical model and consider it for different types of terrain. One can also use a heavily adherent tire to mitigate the impact of these disturbances.

This paper proposes the use of a LIDAR (Light Detection and Ranging) system to estimate the posture of the robot in real time and close the path tracking control loop.

This sensor sweeps in a horizontal line and calculates the distances from the laser position and any object met by the beams. While the robot's moves through previously unknown environments, a localization and mapping algorithm can estimate, with high rate and high precision, the robot's posture.

The estimated posture can be used as feedback in the control loop even if the robot skids or slips, since the laser rangefinder will measure different distances, and therefore the 2D localization algorithm will detect a new posture, even in the presence of such disturbances.

Using this approach to estimate the robot posture, one can use traditional path tracking control strategies, such as the one in [2].

In this work, we show the advantages of using the LIDAR data to estimate postures instead of encoders' odometry to implement the posture control loop of a mobile robot. The performance of the two strategies was evaluated through tests with different reference paths. All tests were carried out in an indoor environment where the robot's wheels are subjected to slipping.

Authorized licensed use limited to: UNIVERSIDADE FEDERAL DO CEARA. Downloaded on March 29,2022 at 18:42:50 UTC from IEEE Xplore. Restrictions apply.

This work is organized as follows. In Section II the mobile robot used for the experiments is described. Section III explains the mathematical modeling of the robot. Section IV brings a brief explanation about the use of laser data to estimate the robot posture with a 2D simultaneous localization and mapping (SLAM) system. The controller architecture is presented in Section V. Section VI presents experimental tests and results. Finally, the conclusion is presented in Section VI.

II. MOBILE ROBOT DESCRIPTION

The real mobile robot used for the experiments is shown in Fig 1. It is a wheeled differential drive mobile robot, which is prepared for the testing of control techniques.

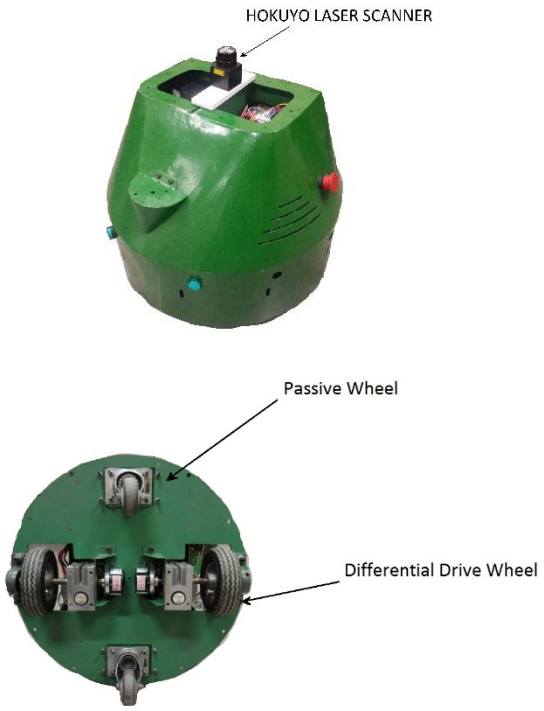

Figure 1. Robot front and under view.

Geared DC motors drive the two differential drive wheels while the two smaller fixed wheels are the responsible for the equilibrium of the robot. The distance L between the two differential drive wheels is 0.4 m and the radius of each wheel is 0.08 m.

An ARM Cortex M4 microcontroller based board (FRDM-K64F) from NXP embeds the posture control strategy. The speed control of the DC motors is performed by an electronic driver (HDC2450, Roboteq) which communicates with the microcontroller through a RS-232 serial protocol. A PID controller internal to the motors' driver is responsible for closing the velocity control loop, using feedback provided by encoders.

A Raspberry Pi 2B is used to communicate with a Hokuyo Laser Rangefinder. The Raspberry Pi reads the data from the sensor and computes the robot new posture using localization algorithms (section 3). At each sample time the laser data is processed and sends the robot current posture to the microcontroller using a high-speed serial communication.

The robot transmits and receives input/output data through a radio-frequency module that communicates with the microcontroller. Another microcontroller connected to a supervisor computer receives the transmitted data through another radio-frequency module, saves it in file, and plots it in real time.

The robot electric system is supplied by two series 24 V 50 AH rechargeable batteries.

III. MODELING OF THE MOBILE ROBOT

Figure 3 shows a diagram block with the components mentioned in section II. The robot's architecture along with its symbols is shown in Fig. 2.

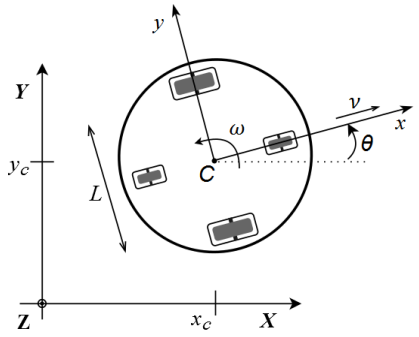

Figure 2. Robot architecture and symbols

Its geometric center and center of gravity supposedly coincide at point C. The inverse kinematic model may be obtained by observing the robot architecture in Fig. 2 as follows:

$$\begin{bmatrix} \dot{x}_c \\ \dot{y}_c \\ \dot{\theta}_c \end{bmatrix} = \begin{bmatrix} \cos\theta & 0 \\ sen\ \theta & 0 \\ 0 & 1 \end{bmatrix} . \begin{bmatrix} v \\ \omega \end{bmatrix} \quad (1)$$

where v and ω are tangential and angular velocities of the robot platform. One can also express each wheel velocity equation, as follows:

$$v_L = v - \frac{\omega L}{2}; v_R = v + \frac{\omega L}{2} \quad (2)$$

Given a reference trajectory ($x_r(t)$, $y_r(t)$, $\theta_r(t)$) defined in a time interval $t \in [0, T]$ the feedforward control law can be derived. The necessary robot inputs are calculated from the reference trajectory as follows:

$$v_r(t) = \pm\sqrt{\dot{x}_r^2(t) + \dot{y}_r^2(t)} \quad (3)$$

Authorized licensed use limited to: UNIVERSIDADE FEDERAL DO CEARA. Downloaded on March 29,2022 at 18:42:50 UTC from IEEE Xplore. Restrictions apply.

Where the ± sign is dependent on the desired movement direction (+ forward and – backward). The angular velocity is also calculated,

$$\omega_r(t) = \frac{\dot{x}_r(t)\,\ddot{y}_r(t) - \dot{y}_r(t)\ddot{x}_r(t)}{\dot{x}_r^2(t) + \dot{y}_r^2(t)} \quad (4)$$

as well as the orientation at each point:

$$\theta_r(t) = \tan^{-1}(\dot{y}_r(t), \dot{x}_r(t)) + k\pi \quad (5)$$

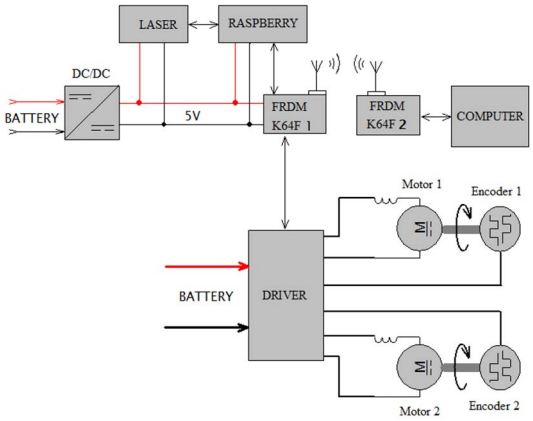

Figure 3. Simplified mobile robot block diagram

IV. 2D SLAM SYSTEM

Laser estimated postures are more reliable when compared to encoder estimation, as the first measures the robot displacement even if the robot movement was caused by disturbances. Such is the case when the robot slips, skids or is pushed away by someone. Even in these situations, the robot is capable of knowing its position and orientation in space. This capability makes the robot's position controller able to find its way back to the desired path.

Furthermore, the high scan rate and accuracy of modern LIDAR systems and the high performance microcomputers allow the use of laser data for posture estimation in real time for embedded control of mobile robots.

Laser scan matching techniques have been widely used for mapping and posture estimation in the last years [5], [6], [7], [8]. It is important to note that the process of aligning laser scans to obtain a robot's posture can be done with two main techniques. In the first one, the points obtained after each scan are compared directly with previously obtained scans in order to determine how much the robot has moved and rotated. A second and less computer consuming two-step approach can be used instead. A slower back-end step is performed to build a grid map of the environment. Meanwhile, a fast front-end step for posture estimation compares new scans with the map learnt so far, and thus the robot posture is estimated with real time closing loop applicability.

A SLAM strategy based on the second approach was used in this work. The Hector_mapping package available in the Robot Operating System (ROS) and developed in [1] was used in the experiments in order to obtain the posture estimation through the LIDAR system.

Suppose that the SLAM system just started, and that there are no previous information about the environment. The world coordinates will be set as shown in figure 2, where the center of the laser becomes the origin of the Cartesian plan and (x,y,θ)=(0,0,0). The x axis points into the yaw direction of the platform at startup. The laser rangefinder sensor has a field of view of 240 degrees.

As the robot moves, a grid map of the environment is built in parallel to the scan matching algorithm that aligns new beam endpoints with the map learnt so far, implicitly performing the matching with all previous scans. Thus, a new posture for the robot is published and is used to close the robot's path tracking control loop, as shown in figure 6.

The 2D slam system is shown in figure 5. The LIDAR system (Hokuyo laser rangefinder) acquires new distance points. A pre-processing routine maps the scan-endpoints into the coordinates of the grid map. The scan matching and mapping routines are interconnected as explained before.

It is important to note that the environment mapping in this work is used exclusively with the aim of finding the robot's relative posture. However, the grid map can be used for future applications such as obstacle avoidance and path planning.

The preprocessing, scan matching and mapping steps are performed in a Raspberry Pi 2B.

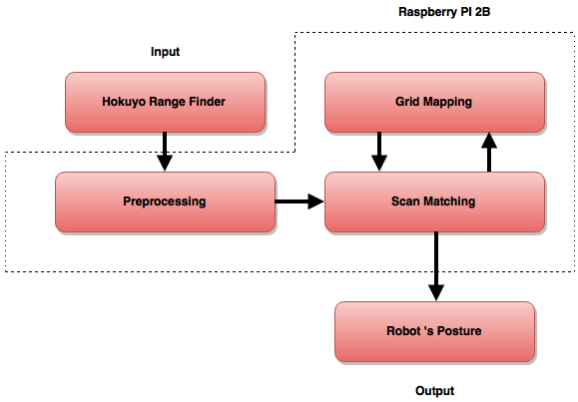

Figure 4. 2D Slam System.

V. CONTROLLER DESIGN

The path tracking control scheme developed in this work is shown in Fig 6. There are two controllers working in cascade. One of them is the posture loop, in which reference position and orientation are compared with the feedback posture obtained with the 2D Slam system presented in figure 5. The kinematics controller then generates the reference linear and angular velocities for right and left motors in order to bring the robot to the desired posture.

Authorized licensed use limited to: UNIVERSIDADE FEDERAL DO CEARA. Downloaded on March 29,2022 at 18:42:50 UTC from IEEE Xplore. Restrictions apply.

The other controller is the motor control inside the mobile robot block. The speed of the DC motors are then controlled through PID controllers that are already included in the electronic driver. The PIDs were adjusted with the following gains: proportional = 5.0, integral = 10.0 and derivative = 0.0. The speed feedback is obtained through incremental encoders coupled to the shaft.

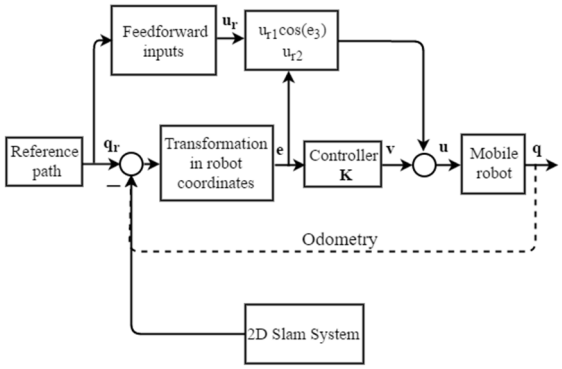

Figure 5. Path tracking control scheme.

It is important to note that the dashed line shows the conventional method used to estimate the robot's posture, which is not used in our control system. As mentioned in IV, we introduced a new block, which converts measured distances into our posture vector [x, y, θ] as feedback.

According to [2], the reference trajectory is defined as the vector $P_{ref} = (x_{ref}, y_{ref}, \theta_{ref})$. The posture error in the mobile robot reference is given by:

$$\begin{bmatrix} e_1 \\ e_2 \\ e_3 \end{bmatrix} = \begin{bmatrix} cos\theta & sen\theta & 0 \\ -sen\theta & cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} . \begin{bmatrix} X_r - x \\ Y_r - y \\ \theta_r - \theta \end{bmatrix} \quad (6)$$

From Equation (6) the error model is defined as the following system of equations:

$$\begin{bmatrix} \dot{e}_1 \\ \dot{e}_2 \\ \dot{e}_3 \end{bmatrix} = \begin{bmatrix} \cos e_3 & 0 \\ \text{sen } e_3 & 0 \\ 0 & 1 \end{bmatrix} . \begin{bmatrix} v_r \\ \omega_r \end{bmatrix} + \begin{bmatrix} -1 & e_2 \\ 0 & -e_1 \\ 0 & -1 \end{bmatrix} . \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} \quad (7)$$

Then, the robot inputs (trajectory tracking controller outputs) are defined as:

$$u_1 = v_r \cos e_3 - v_1 \qquad u_2 = \omega_r - v_2 \quad (8)$$

where v_r and ω_r are the feedforward tangential and angular velocities, while v_1 and v_2 are the the closed loop outputs. The error model is linearized and the following feedback control law is obtained [2]:

$$\begin{bmatrix} v_1 \\ v_2 \end{bmatrix} = \begin{bmatrix} -k_1 & 0 & 0 \\ 0 & -sign(v_r)k_2 & -k3 \end{bmatrix} . \begin{bmatrix} e_1 \\ e_2 \\ e_3 \end{bmatrix} \quad (9)$$

In order to tune the controller gains a desired relative damping ζ and natural frequency ω_n are chosen by the user. The closed-loop response of the error signals are related to these specifications [2]. Then, the controller gains are calculated in real time through the given equations:

$$k_1 = k_3 = 2\zeta\omega_n(t) \qquad k_2 = g.|v_r(t)| \qquad \omega_n(t) = \sqrt{\omega_r(t)^2 + gv_r(t)^2} \quad (10)$$

Stability analysis and proof were presented in [2].

VI. EXPERIMENTAL TESTS AND RESULTS

The specified parameters for the controller are shown in the Table 1 below. The selected values were chosen so that the robot frame drives smoothly along the reference path. The "g" variable represents a proportional gain, which multiplies both the reference velocity and the calculated error, as mentioned in [2]. The "ζ" variable represents the damping gain that multiplies the natural frequency of the desired characteristic polynomial plant, which is assumed to be of third order [3].

Table 1. Robot initial errors for circumference.

Variable	Value
ζ	0.6
g	40

The robot shown in Fig 1 was used for experimental tests. As a first test, the robot posture was obtained using the encoders data integrated over time using the robot's inverted kinematics model.

The reference robot path was a circumference. This is a good reference path, since the robot needs to change its position and orientation at every point of the trajectory, adding a good level of complexity. The robot performed two laps.

Table 2. Robot initial errors for circumference.

Initial Errors		
x [m]	*y [m]*	*θ [rad]*
0	0.5	-pi

Authorized licensed use limited to: UNIVERSIDADE FEDERAL DO CEARA. Downloaded on March 29,2022 at 18:42:50 UTC from IEEE Xplore. Restrictions apply.

In the first test the desired reference (dashed red line in Fig. 6) was set to a circle and the robot had its initial posture [x=0, y=0, θ=pi] far from the reference [x=0, y=0.5, θ=0]. The blue line is the real position during the path following. It is notable that the blue curve is very far from the reference path due to the slipping of the wheels, an effect that the encoders cannot detect, which means that the odometry presents a steady state error. Note also that the encoders' data shows that the robot is following the desired path, which implies that the algorithm "thinks" that it is following without errors, although it is not what really happens.

In a second test, the SLAM algorithm cited in section IV provided the estimated pose for the kinematics controller. The initial conditions were the same from Table 2. The resulting trajectory curve can be seen as a blue line in Fig 7. The real trajectory follows the one of the reference path and the controller reduces the posture error to zero. Fig 8 shows x, y and θ errors for the second test.

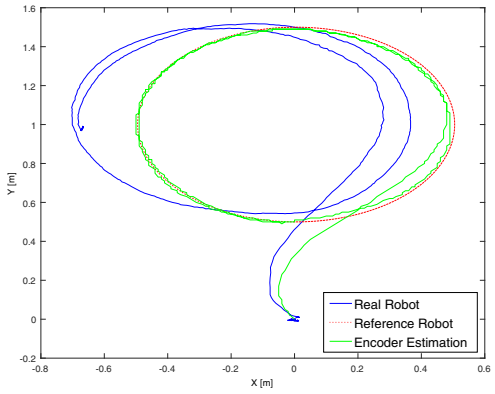

Figure 6. Robot trajectory using encoder estimated poses.

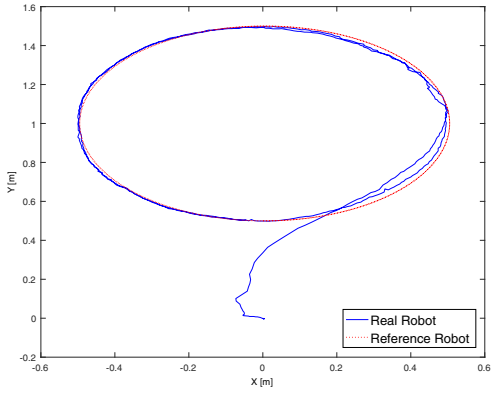

Figure 7. Robot trajectory using laser estimated postures.

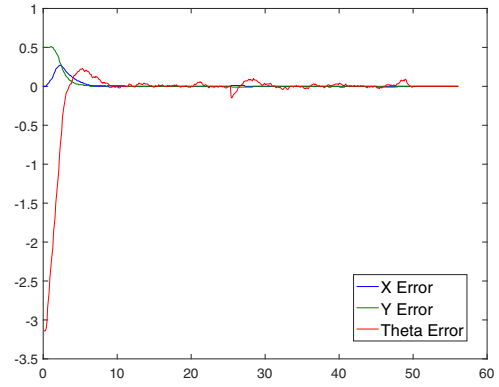

Figure 8. Robot trajectory errors using laser estimated postures.

In a third test the reference trajectory was switched to a square of sizes equal to 1.5 meters. The reference had its initial posture as [x=0, y=0.5, θ=0]. Once again, the encoder estimation was used for posture feedback. Initial errors are listed in table 3, and the same gains of the first two tests were used. The robot trajectory was plotted in Fig. 9. The wheel's slipping and skidding is very notable, since the robot had to do a sharper curve in order to follow the desired trajectory. The red curve shows the encoder estimated position, the dashed line shows the reference path and the real robot curve is shown by the blue curve.

In a forth test, the same trajectory was followed, but using the SLAM algorithm. The trajectory curve and errors were plotted in Fig. 10 and Fig 11, respectively. The real robot curve is show by a blue curve, and the desire path as a dashed red line.

Table 3. Robot initial errors for square.

Initial Errors		
x [m]	*y [m]*	*θ [rad]*
0	0.5	0

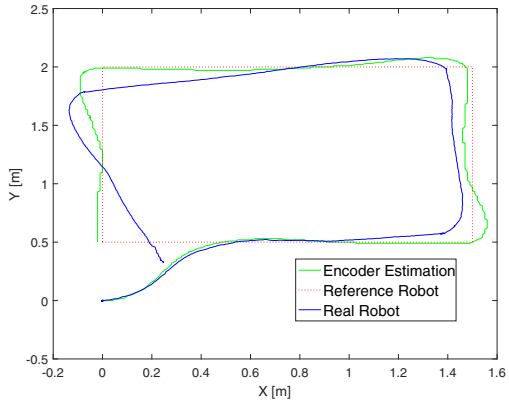

Figure 9. Robot trajectory using encoder estimated postures.

Authorized licensed use limited to: UNIVERSIDADE FEDERAL DO CEARA. Downloaded on March 29,2022 at 18:42:50 UTC from IEEE Xplore. Restrictions apply.

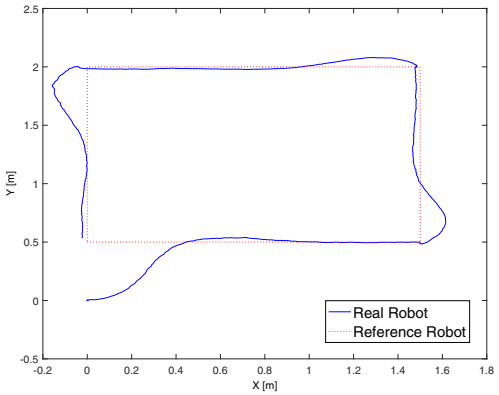

Figure 10. Robot trajectory using laser estimated postures.

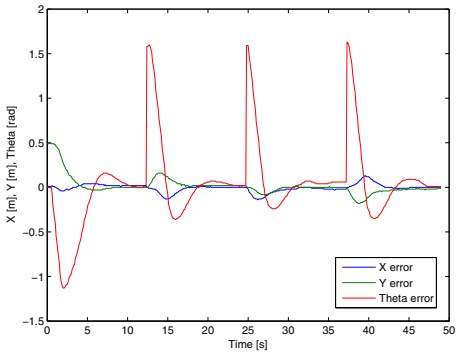

Figure 11. Robot trajectory errors using laser estimated postures.

VII. CONCLUSION

In this paper are showed the main disadvantages of using dead reckoning as a feedback for a robot's posture control loop. In addition, it was proposed the use of a laser sensor to estimate the robot`s posture and use it for the kinematic controller.

The proposed scheme results were very effective, as the robot's experiments were not affected by any slipping or skidding. We also believe that a laser sensor can be used to provide feedback in many other control strategies applied to a mobile robot trajectory control, such as in predictive or adaptive control models. In addition, it should be very simple to adapt any existing control strategy to the proposed model by simply removing the feedback acquired by encoders or other sensors and inserting a processing unit that is able to receive data from a laser rangefinder sensor and output it in terms of the robot's posture.

ACKNOWLEDGEMENTS

The authors acknowledge the support from CNPQ, CAPES, and FUNCAP.

REFERENCES

[1] Kohlbrecher, S., Von Stryk, O., Meyer, J. and Klingauf, U., 2011, November. A flexible and scala-ble slam system with full 3d motion estimation. In Safety, Security, and Rescue Robotics (SSRR), 2011 IEEE International Symposium on (pp. 155-160). IEEE.

[2] Klančar, G., Matko, D. and Blažič, S., 2005, June. Mobile robot control on a reference path. In Intelligent Control, 2005. Proceedings of the 2005 IEEE International Symposium on, Mediterrean Conference on Control and Automation (pp. 1343-1348). IEEE.

[3] Wang, D. and Low, C.B., 2008. Modeling and analysis of skidding and slipping in wheeled mobile robots: control design perspective. Robotics, IEEE Transactions on, 24(3), pp.676-687.

[4] Fuke, Y. and Krotkov, E., 1996, April. Dead reck-oning for a lunar rover on uneven terrain. In Robotics and Automation, 1996. Proceedings., 1996 IEEE International Conference on (Vol. 1, pp. 411-416). IEEE.

[5] Zhang, J. and Singh, S., 2014, July. Loam: Lidar odometry and mapping in real-time. In Robotics: Science and Systems Conference (RSS) (pp. 109-111).

[6] Gonzalez, J., Stentz, A. and Ollero, A., 1995. A mobile robot iconic position estimator using a radial laser scanner. Journal of Intelligent and Robotic Systems, 13(2), pp.161-179.

[7] Dissanayake, G., Durrant-Whyte, H. and Bailey, T., 2000. A computationally efficient solution to the simultaneous localisation and map building (SLAM) problem. In Robotics and Automation, 2000. Proceedings. ICRA'00. IEEE International Conference on (Vol. 2, pp. 1009-1014). IEEE.

[8] Eliazar, A. and Parr, R., 2003, August. DP-SLAM: Fast, robust simultaneous localization and mapping without predetermined landmarks. In IJCAI (Vol. 3, pp. 1135-1142).

[9] H. Y. Chung, C. C. Hou and Y. S. Chen, "Indoor Intelligent Mobile Robot Localization Using Fuzzy Compensation and Kalman Filter to Fuse the Data of Gyroscope and Magnetometer," in IEEE Transac-tions on Industrial Electronics, vol. 62, no. 10, pp. 6436-6447, Oct. 2015.

[10] L. Marín, M. Vallés, Á Soriano, Á Valera and P. Albertos, "Event-Based Localization in Ackermann Steering Limited Resource Mobile Robots," in IEEE/ASME Transactions on Mechatronics, vol. 19, no. 4, pp. 1171-1182, Aug. 2014.

[11] K. Srinivasan and J. Gu, "Multiple Sensor Fusion in Mobile Robot Localization," 2007 Canadian Conference on Electrical and Computer Engineering, Vancouver, BC, 2007, pp. 1207-1210.

Authorized licensed use limited to: UNIVERSIDADE FEDERAL DO CEARA. Downloaded on March 29,2022 at 18:42:50 UTC from IEEE Xplore. Restrictions apply.