Available online at www.sciencedirect.com

ScienceDirect

IFAC PapersOnLine 51-4 (2018) 364–369

Reference Tracking of a Nonholonomic Mobile Robot using Sensor Fusion Techniques and Linear Control

Marcus D. N. Forte * **Wilkley B. Correia** *
Fabrício G. Nogueira * **Bismark C. Torrico** *

* *Electrical Engineering Department, Universidade Federal do Ceará, Av. Mister Hull - Pici - 60455-760*

e-mails: davi2812@dee.ufc.br wilkley@dee.ufc.br fnogueira@dee.ufc.br bismark@dee.ufc.br

Abstract: This paper presents the control design of a nonholonomic mobile robot with differential drive using control strategies on a linearized space state error model. In this case, a diagonal multi-variable model is obtained for which a decentralized PI controller may be designed. In this paper, PI tuning is performed through a LQR problem whose feedback gains are set as the proportional gain of the PI controller. The use of an Inertial Measuring Unit (IMU) allows for a precise posture feedback by using a Kalman Filter on the output of the sensors. For such purpose, a Sensor Fusion technique is also needed in order to combine multiple sensor output so that physical limitations of each sensor may be compensated.

Keywords: Sensor fusion, Mobile robots, Inertial measurement units, Linear control systems, Robot kinematics

1. INTRODUCTION

Control of a mobile robot is commonly referred to as the vast application of robotics, since its often applied for exploration, mapping and observation of certain environments. Control of nonholonomic mobile robots is particularly a challenging application as its movement restrictions must be accounted for the control design. Its system normally consists of three variables to model its posture, namely its position (x, y) and its angle (θ) related to a global reference frame, along with two control input variables, namely, the linear and angular velocities (v, ω). From the perspective of control theory, such problem can be approached as a multi-variable system.

The feedback information of the robot posture is normally obtained through estimation of the current position using both linear and angular velocities obtained from encoder inputs. This simple method, commonly referred to as odometry, works well if the robot is not subjected to disturbances such as skidding and slipping, limiting therefore, its real world application.

An IMU is a device that measures forces, angular rates and eventually magnetic forces with respect to its body frame. It usually consists of accelerometers, gyroscopes and magnetometers embedded into a single monolithic integrated circuit or board. The main issue with IMUs is that the raw data acquired from such sensors are subjected to measurement noise and/or bias, which must be filtered in order to get meaningful information for an application. The use of inertial and vision sensors for posture estimation has been the object of study over the recent past decades. In Corke et al. (2007) the combination of vision data for mobile robot is detailed. In Chenavier and Crowley (1992) a positioning strategy using vision and odometry is used. In Lee et al. (2012) odometry, inertial and vision fusion strategy is used as a localization technique.

Recent works make use of advanced sensors and techniques such as GPS(Global Positioning System) and in-door localization methods to track a mobile robot position. A GPS sensor fused with inertial data to localize and control a mobile robot along a desired trajectory is considered in Khatib et al. (2015). Fusion of inertial and ultrasonic sensors are used to estimate a mobile robot localization in Dobrev et al. (2016). GPS localization has proven its usefulness under condition of clear sky, while alternatives must be studied for mobile robots navigating in-doors.

Basic control of a mobile robot can be made with a point-to-point approach (classical control) when the path between the states is not important (Klancar et al., 2005). Alternatively, one may apply a "virtual robot" to follow trajectory as a reference frame of the real robot (Klancar et al., 2005). The latter is the one commonly used since it is helpful for obstacle avoidance. Although it leads to a nonlinear control law, it became a popular controller for reference trajectory tracking. In Normey-Rico et al. (2001) a robust PI approach is introduced for mobile robot reference tracking, as an alternative to the virtual robot.

This paper proposes a modification in the virtual robot approach in order to consider a linear primary controller to minimize the error between real robot position and a virtual robot frame for reference tracking. To accomplish such a task, a linearized model based on the posture error is considered for the primary controller. In addition, data fusion from gyroscope and a digital compass is considered

10.1016/j.ifacol.2018.06.092

for a Kalman filter in order to estimate mobile robot angular velocity which is essential to compute the robot posture.

2. ROBOT MODELING

2.1 Inverse Kinematics

The robot architecture, along with its symbols, is shown in Fig. 1. The variables x_c, y_c represent the mobile robot

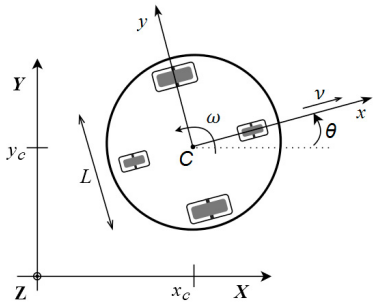

Fig. 1. Robot Architecture

center of mass. ω is the angular velocity, v is the linear velocity. L is the distance between the motorized wheels. θ is the mobile robot global heading. x, y represent the mobile robot body reference frame.
For the robot it is assumed that both geometric center C and gravity center are coincident. The inverse kinematics can be obtained by observing the robot architecture in Fig. 1 as follows:

$$\begin{bmatrix}\dot{x}\\ \dot{y}\\ \dot{\theta}\end{bmatrix} = \begin{bmatrix}\cos\theta & 0\\ \sin\theta & 0\\ 0 & 1\end{bmatrix} \cdot \begin{bmatrix}v\\ \omega\end{bmatrix}, \tag{1}$$

whose discretized model is given by

$$\begin{bmatrix}x\\ y\\ \theta\end{bmatrix}_{k+1} = \begin{bmatrix}x\\ x\\ \theta\end{bmatrix}_{k} + T_s \cdot \begin{bmatrix}\cos\theta & 0\\ \sin\theta & 0\\ 0 & 1\end{bmatrix} \cdot \begin{bmatrix}v\\ \omega\end{bmatrix}_{k}, \tag{2}$$

where v and ω are tangential and angular velocities of the robot platform, respectively, and T_s is the discrete-time implementation sampling time. The right and left wheel velocities are expressed by

$$v_L = v - \frac{\omega L}{2}; v_R = v + \frac{\omega L}{2}. \tag{3}$$

Given a reference trajectory $(x_r(t), y_r(t), \theta_r(t))$ defined in a time interval $T \in [0, T]$ a feed-forward control law can be derived

$$v_r(t) = \pm\sqrt{\dot{x}_r^2(t) + \dot{y}_r^2(t)} \tag{4}$$

and

$$\omega_r(t) = \frac{\dot{x}_r(t)\ddot{y}_r(t) - \dot{y}_r(t)\ddot{x}_r(t)}{\dot{x}_r^2(t) + \dot{y}_r^2(t)}, \tag{5}$$

with the angles at each point defines as:

$$\theta_r(t) = \text{atan2}(\dot{y}_r(t), \dot{x}_{r(t)}) + k\pi, \tag{6}$$

where k defines a desired direction (0 for forward and 1 for backward). These control laws, however, only drive the mobile robot to the correct path when there are no disturbances (skidding, slipping) and errors in the initial state.

2.2 Controller Design

The system can be converted into a regulation problem when an error expression is explicitly defined. Let p_r be the reference trajectory states $p_r = [x_r\ y_r\ \theta_r]^T$ and p be the real robot posture $p = [x\ y\ \theta]^T$. The error state $p_e = [e_1\ e_2\ e_3]^T$ is then written as:

$$\mathbf{p_e} = \mathbf{p_r} - \mathbf{p} \quad or \quad \begin{bmatrix}e_1\\ e_2\\ e_3\end{bmatrix} = \begin{bmatrix}x_r - x\\ y_r - y\\ \theta_r - \theta\end{bmatrix}. \tag{7}$$

The error frame is expressed in the real robot frame using a rotation matrix to convert from the navigation frame (whose coordinates are path coordinates: X_r, Y_r) to the body frame (x and y from Fig. 1).

$$\begin{bmatrix}e_1\\ e_2\\ e_3\end{bmatrix} = \begin{bmatrix}\cos\theta & \sin\theta & 0\\ -\sin\theta & \cos\theta & 0\\ 0 & 0 & 1\end{bmatrix} \cdot \begin{bmatrix}x_r - x\\ y_r - y\\ \theta_r - \theta\end{bmatrix}. \tag{8}$$

Considering the robot kinematics from Eq. (1) and taking the derivative of Eq. (8), one obtains the error model (Klancar et al., 2005):

$$\begin{bmatrix}\dot{e}_1\\ \dot{e}_2\\ \dot{e}_3\end{bmatrix} = \begin{bmatrix}\cos e_3 & 0\\ \sin e_3 & 0\\ 0 & 1\end{bmatrix} \cdot \begin{bmatrix}u_{r1}\\ u_{r2}\end{bmatrix} + \begin{bmatrix}-1 & e_2\\ 0 & -e_1\\ 0 & -1\end{bmatrix} \cdot \begin{bmatrix}u_1\\ u_2\end{bmatrix}. \tag{9}$$

Rewriting the control inputs as

$$\begin{aligned} u_1 &= u_{r1}\cos e_3 - v_1\\ u_2 &= u_{r2} - v_2 \end{aligned} \tag{10}$$

Eq. (9) can now be expressed as

$$\begin{aligned} \begin{bmatrix}\dot{e}_1\\ \dot{e}_2\\ \dot{e}_3\end{bmatrix} &= \begin{bmatrix}0 & u_2 & 0\\ -u_2 & 0 & 0\\ 0 & 0 & 0\end{bmatrix} \cdot \begin{bmatrix}e_1\\ e_2\\ e_3\end{bmatrix} +\\ &+ \begin{bmatrix}0\\ \sin e_3\\ 0\end{bmatrix} \cdot u_{r1} + \begin{bmatrix}1 & 0\\ 0 & 0\\ 0 & 1\end{bmatrix} \cdot \begin{bmatrix}v_1\\ v_2\end{bmatrix} \end{aligned} \tag{11}$$

By considering a discrete-time implementation (2) the posture of the mobile robot from does not changes abruptly between consecutive sampling time T_s. Thus, variations δe on errors may be taken as small as $\delta e_1 = \delta e_2 = \delta e_3 \approx 0$ as well as velocity changes $\delta v_1 = \delta v_2 \approx 0$. These considerations applied to Eq. (11) lead to the linearized model

$$\delta\dot{e} = \begin{bmatrix}0 & u_{r2} & 0\\ -u_{r2} & 0 & u_{r1}\\ 0 & 0 & 0\end{bmatrix} \cdot \delta e + \begin{bmatrix}1 & 0\\ 0 & 0\\ 0 & 1\end{bmatrix} \cdot \delta v \tag{12}$$

If posture (x, y, θ) is the output named ξ, one takes the linear state-space model

$$\dot{\delta e} = \bar{A}\delta e + \bar{B}\delta v \tag{13}$$

$$\xi = \bar{C}\delta e \tag{14}$$

where

$$\bar{A} = \begin{bmatrix}0 & u_{r2} & 0\\ -u_{r2} & 0 & u_{r1}\\ 0 & 0 & 0\end{bmatrix};\ \bar{B} = \begin{bmatrix}1 & 0\\ 0 & 0\\ 0 & 1\end{bmatrix} \text{ and } \bar{C} = \begin{bmatrix}1 & 0 & 0\\ 0 & 1 & 0\\ 0 & 0 & 1\end{bmatrix}.$$

System described by Eqs. (13) and (14) $[\bar{A}\ \bar{B}]$ has full rank of controllability if either u_{r1} or u_{r2} are non-zero which is a sufficient condition if reference inputs u_{r1} and u_{r2} are constant. For this system state δe is related to the non-linear states from (1) when subtracted from the reference frame $p_r = [x_r\ y_r\ \theta_r]^T$. The linearized input vector δv is related to the mobile robot velocities vector $[v\ \omega]^T$.

Therefore, one may consider a linear control law such as

$$u = -K\delta e, \tag{15}$$

where K is a state static gain which may be obtained from the solution of the linear quadratic regulator problem associated with the linear system given by Eqs. (13) and (14), where $u = [v\ \omega]^T$.

Nevertheless, if integral action is not taken into account, state feedback control law is not able to guarantee itself null steady-state error for reference path tracking. Therefore, feedforward control law, from the virtual robot approach in Klancar et al. (2005), must be included, leading to the overall control block diagram shown in Figure 2.

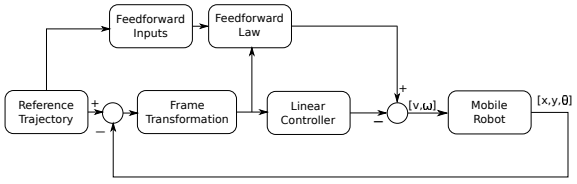

Fig. 2. Linear Control Scheme

In a case of reference tack of line shaped, or with small curvature, one may consider $\omega_r = 0$ in Eq. (12), whose dimensional analysis of the LQR solution leads to

$$K = \begin{bmatrix} k_1 & 0 & 0 \\ 0 & k_2 & k_3 \end{bmatrix}. \tag{16}$$

Consequently, a realization of such reduced state space linear system, i. e. if $\omega_r = 0$ in Eqs. (13) and (14) lead to the multi-variable linear system

$$\xi(s) = \begin{bmatrix} \frac{1}{s} & 0 \\ 0 & \frac{1}{s^2} \\ 0 & \frac{1}{s} \end{bmatrix} \delta e(s), \tag{17}$$

whose output x depend exclusively on v while y and θ depend exclusively on ω. So that, model in Eq. (17) exhibits decoupling between output and input variables. In this case it is suitable to consider a decentralized controller, such as PID, for each transfer function, leading to the following set of open loop control systems:

$$L_1 = C_1(s) \cdot \frac{1}{s} \tag{18}$$

$$L_2 = C_2(s) \cdot \frac{1}{s^2} \tag{19}$$

$$L_3 = C_3(s) \cdot \frac{1}{s}. \tag{20}$$

In this case, a set of proportional controllers $C_1(s)$, $C_2(s)$ and $C_3(s)$ in Eqs. (18) to (20) along with feed-forward control law assures correctness of the path trajectory tracking.

However, Eq. (16) exhibits a set of gains whose values can be used as the desired proportional controllers, which allows to write $C_1(s) = k_1, C_2(s) = k_2$, and $C_3(s) = k_3$.

3. SENSOR FUSION MODEL

The accelerometer measures linear acceleration with respect to earth's frame; gyroscope measures angular velocities of a body and magnetometer senses the magnetic field around its body, including Earth's magnetic field.

The angular velocity is an important variable with which one can estimate a mobile robot heading angle (yaw angle) with respect to its body or a navigational frame, according to (1). Gyroscopes provide such data, however, along with an unknown bias intrinsic to its construction limitation. It is also expected to generate an integration error when estimating the heading angle due to noise. Heavy noise is also present in the magnetometer readings as of its sensitivity

By combining the sensor data provided by the magnetometer with the angular velocities from the gyroscope, one can eliminate both bias and noise issues.

Let $\theta(t)$ be

$$\theta_{gyro}(t) = \theta_0 + \int \omega_{gyro}(t)\ dt \tag{21}$$

which can also be expressed in terms of magnetometer data

$$\begin{aligned} X_H = x_{mag}\cos\beta + y_{mag}\sin\alpha\sin\beta + \\ + z_{mag}\sin\beta\cos\alpha \end{aligned} \tag{22}$$

$$Y_H = y_{mag}\cos\alpha + z_{mag}\sin\alpha \tag{23}$$

$$\theta_{mag}(t) = atan2(-Y_H, X_H) \tag{24}$$

where $x_{mag}, y_{mag}, z_{mag}$ are the magnetometer calibrated outputs, α, β are the roll and pitch rotations, respectively. The sensor fusion can be implemented to accurately estimate the mobile robot yaw angle. For such purpose, consider the following discrete model for the gyroscope data

$$\omega_{gyro} = \omega + b + \eta_g \tag{25}$$

$$\theta_k = \theta_{k-1} + \omega_{gyro} \cdot T_s \tag{26}$$

where b is the sensor bias, η_g is the gyroscope measurement noise and T_s is the sensor sampling time. The magnetometer discrete model is expressed by

$$\theta_k = \theta_{k-1} + \eta_m \tag{27}$$

where η_m is the magnetometer measuring noise and θ_k is computed by (24). The estimation models are given as follows:

$$\begin{bmatrix} \theta \\ b \end{bmatrix}_k = \begin{bmatrix} 1 & -T_s \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \theta \\ b \end{bmatrix}_{k-1} + \begin{bmatrix} T_s \\ 0 \end{bmatrix} \omega_{gyro} + \eta_g' \tag{28}$$

$$= A_f x_{k-1} + B_f u_k + \eta_g' \tag{29}$$

$$y_k = [1\ 0] \begin{bmatrix} \theta \\ b \end{bmatrix}_{k-1} + \eta_m \tag{30}$$

$$= C_f x_k + \eta_m \tag{31}$$

Note that the gyroscope measurement noise was removed from (25) and used as a Wiener process noise in (28) (Brown and Hwang, 1997).

The system expressed in (28) and (30) is a linear system which can be used as a model to compute an optimal estimator for its states, namely the Kalman Filter, with $\hat{x}$ being the estimated state, A_f, B_f and C_f are the system model matrices taken to solve the Kalman filter optimal estimator.Therefore, Kalman filter gain K is solved as usual by considering covariance matrix Q_n as the process noise covariance and R_n is the output covariance noise, given respectively by

$$Q_n = q. \begin{bmatrix} \frac{T_s^2}{2} & 0 \\ 0 & T_s \end{bmatrix}, \qquad R_n = Var(\eta_m),$$

where q is a tuning parameter. Measurement noise variance is obtained experimentally from data collected through a

FRDM-STBC-AGM01 board from NXP.
The linear velocity obtained from encoders are used to estimate x and y from (8) using numerical integration in order to compute the error states. $\hat{\theta}$ is the output of the Kalman filter. Both estimations compose the fusion technique.
The schematic for this control strategy is shown in Fig. 3.

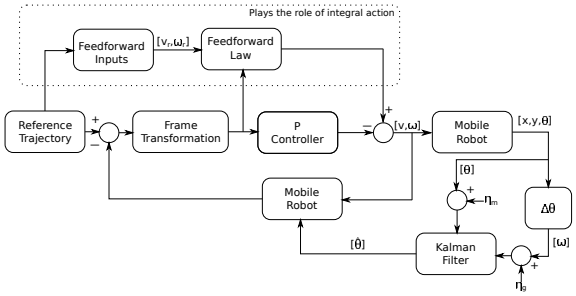

Fig. 3. Mobile Robot Control Scheme

4. SIMULATION RESULTS

This section presents the sensor and control model simulations. The gyroscope and magnetometer measuring data for a given angular velocity is shown in Fig. 4. Notice the

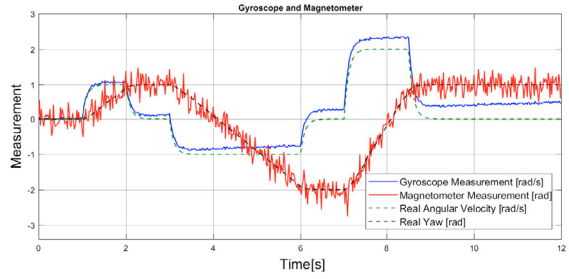

Fig. 4. Raw Sensor Measurements

noisy nature of the magnetometer and the time-varying biased output of the gyroscope. This data is used to estimate the mobile robot yaw angle. In Fig. 5 the Kalman filter is implemented with its gain computed for the steady-state. The gyroscope raw biased data is integrated for comparison. Fig. 7 presents an actual trajectory tracking

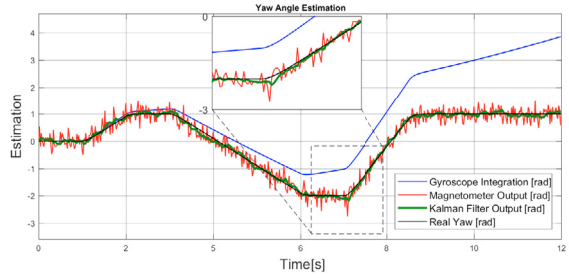

Fig. 5. Filtered Sensor Measurements

contemplating both the sensor model and the mobile robot inverse kinematics. For this trajectory, two interconnected L shapes trajectory paths with constant linear velocity $v = 0.1m/s$ were considered. The actual reference starts on position $[x\ y\ \theta] = [0.5\ 0.5\ 0]$. An angular disturbance of 0.5 rad was applied at $T = 10s$. The total trajectory tracking lasts for $T_{simulation} = 60s$.
The controller was designed as a LQR problem with the following weighting matrices

$$Q = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 50 & 0 \\ 0 & 0 & 0 \end{bmatrix}, R = I,$$

with $q = 0.01$ and $R_n = 0.01$ for the Kalman filter design purpose and $T_s = 0.02s$. Note that entries for Q suggest that e_1 and e_2 are taken into account for LQR optimization while e_3 is not. States e_1 and e_2 are associated with the mobile robot position (x, y), whose error is supposed to be minimized by the proposed controller. Comparison results for different Q weighting matrix is shown in Fig. 6 for a square shaped curve. Such result exhibits the intuitive nature for gain tuning from Q entries. Therefore, under a PI controller perspective, computed proportional gains are given by

$$K = \begin{bmatrix} 1.0 & 0 & 0 \\ 0 & 7.07 & 1.19 \end{bmatrix}. \quad (32)$$

It is worth to note that weight for e_2 must be higher than

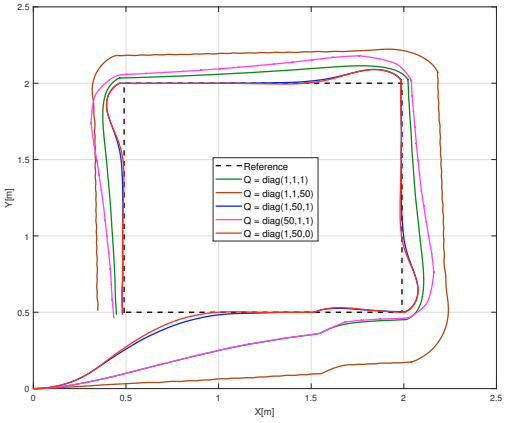

Fig. 6. Q tuning

that for e_1 in order to assure optimization for the reference trajectory distance error. Therefore, by tuning e_2 one acts in order to reduce both maximum overshoot and settling time in a case of abrupt change in the direction of the reference path as it is readily noticed in Fig. 6. In this case, the proposed strategy makes tuning procedure more intuitive than the nonlinear one.
Simulations for comparison purpose have also been carried out to the well-know nonlinear control strategy (Klancar et al., 2005), whose tuning is performed by a gain and a damping factor.

The odometry estimation was not able to detect any angular disturbance, whose drift drove the robot away from the reference path. For this simulation, yaw estimation from filtered data (Kalman Filter) was used as the actual mobile robot angle, instead of an encoder estimated angle.
A saturation velocity of 0.4 m/s and 0.8 rad/s was used to account for real mobile robot constraints as shown in Fig. 8. In Fig. 9 the state error variables are shown. For a second test, a circular trajectory was to be tracked.

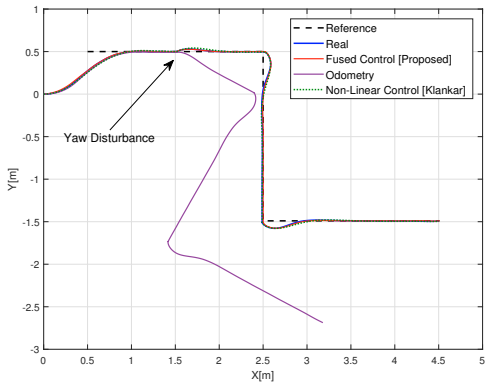

Fig. 7. Line Trajectory Tracking

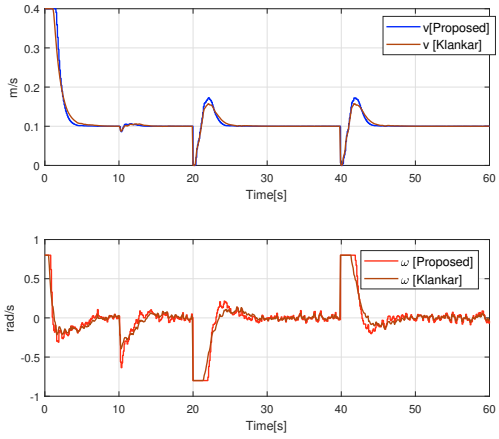

Fig. 8. Mobile Robot Velocities

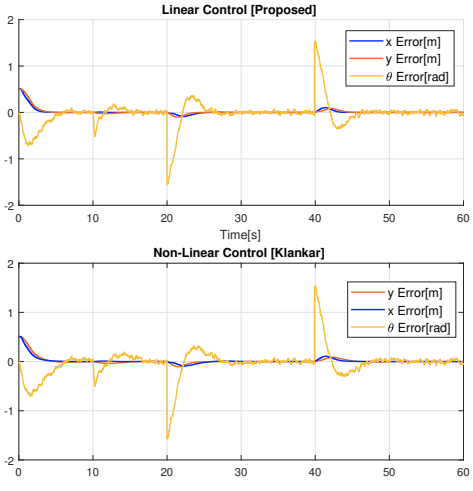

Fig. 9. Navigational Frame Errors

The trajectory consisted of two circles, each to be tracked with opposite angular velocities. An interesting result has occurred when the feed-forward input u_{r2} of (12) was set to zero even though the mobile robot was to be subjected to angular velocity, as the mobile robot was successfully able to perform a circular trajectory tracking. The same feed-forward input u_{r1}, weighting matrices for LQ and Kalman filter design from the first experiment were used.

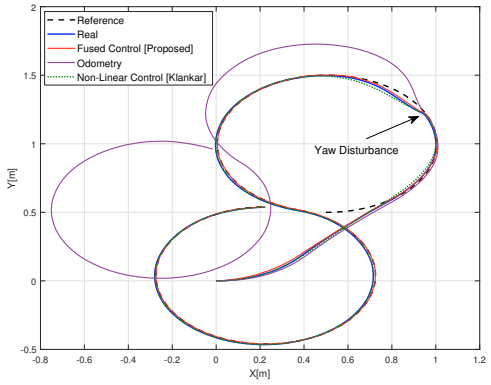

Fig. 10. Line Trajectory Tracking

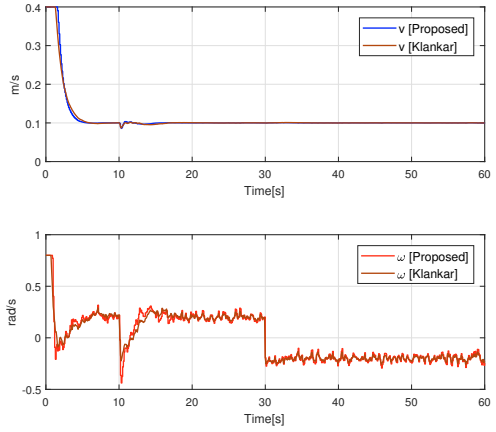

Fig. 11. Mobile Robot Velocities

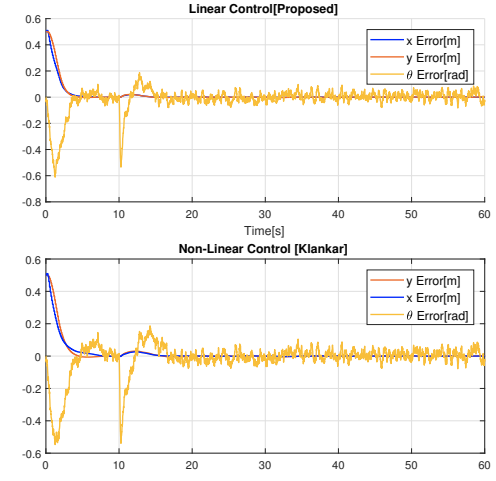

Fig. 12. Navigational Frame Errors

5. EXPERIMENTAL RESULTS

The proposed control strategy was used in a real differential driven mobile robot. Due to terrain slipping and

skidding, it is expected that the experiment behaves differently than the simulations. Fig. 13 to Fig. 15 show the obtained results for $Q = diag(1, 50, 0)$.

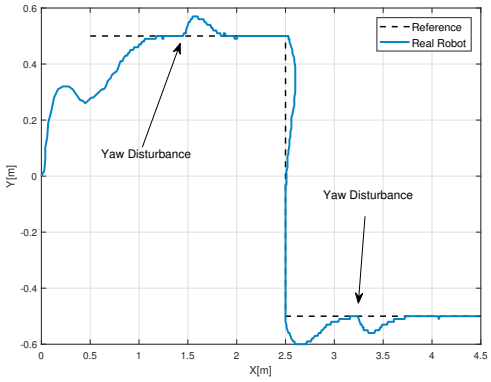

Fig. 13. Line trajectory tracking

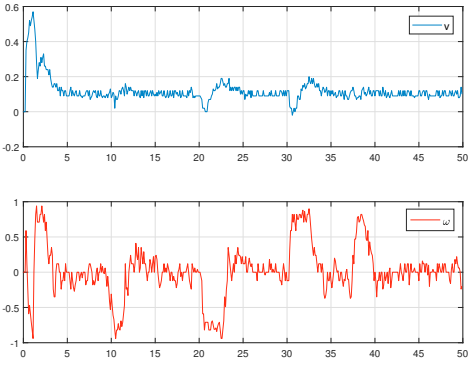

Fig. 14. Mobile Robot Velocities

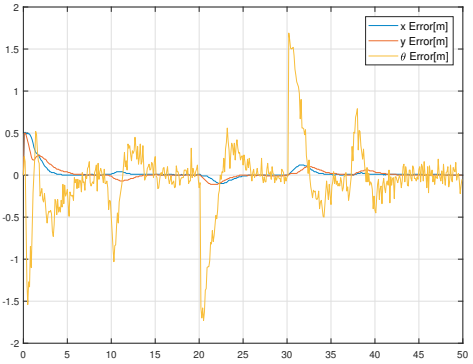

Fig. 15. Navigational Frame Errors

6. CONCLUSION

This paper has presented a linear controller to be applied in a nonholonomic mobile robot reference tracking as an alternative to the commonly used nonlinear controller. By considering the linearized system, the controller is set as simple gains to form a decentralized P controller. Feedforward action from the nonlinear approach is preserved in order to play the role of integral action in a zero steady-state error sense for reference tracking. Tuning of the P controllers comes from a LQR problem applied to the linearized robot model. Feedback loop is closed trough fusion data from both gyroscope and magnetometer in order to get an accurate value of the heading angle. Nevertheless, because of the measurements from magnetometer are noisy it must go trough a Kalman filter in order to provide online filtered information. By applying LQR procedure to tune a decentralized P controller one may suggest that an optimal state feedback can be further exploited. Future work are intended to use information from the linear accelerometer for translational disturbances rejection purposes (robot pushing or sliding).

7. ACKNOWLEDGMENT

The authors acknowledge the support from CNPQ, CAPES, and FUNCAP, through the project grant number 88887.114030 /2015-01.

REFERENCES

Brown, R. and Hwang, P. (1997). *Introduction to random signals and applied Kalman filtering: with MATLAB exercises and solutions*. Number v. 1 in Introduction to Random Signals and Applied Kalman Filtering: With MATLAB Exercises and Solutions. Wiley.

Chenavier, F. and Crowley, J.L. (1992). Position estimation for a mobile robot using vision and odometry. In *Proceedings 1992 IEEE International Conference on Robotics and Automation*, 2588–2593 vol.3. doi: 10.1109/ROBOT.1992.220052.

Corke, P., Lobo, J., and Dias, J. (2007). An introduction to inertial and visual sensing. 26, 519–535.

Dobrev, Y., Flores, S., and Vossiek, M. (2016). Multi-modal sensor fusion for indoor mobile robot pose estimation. In *2016 IEEE/ION Position, Location and Navigation Symposium (PLANS)*, 553–556. doi: 10.1109/PLANS.2016.7479745.

Khatib, E.I.A., Jaradat, M.A., Abdel-Hafez, M., and Roigari, M. (2015). Multiple sensor fusion for mobile robot localization and navigation using the extended kalman filter. In *2015 10th International Symposium on Mechatronics and its Applications (ISMA)*, 1–5. doi: 10.1109/ISMA.2015.7373480.

Klancar, G., Matko, D., and Blazic, S. (2005). Mobile robot control on a reference path. In *Proceedings of the 2005 IEEE International Symposium on, Mediterrean Conference on Control and Automation Intelligent Control, 2005.*, 1343–1348. doi:10.1109/.2005.1467211.

Lee, T.J., Bahn, W., m. Jang, B., Song, H.J., and i. Dan Cho, D. (2012). A new localization method for mobile robot by data fusion of vision sensor data and motion sensor data. In *2012 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, 723–728. doi:10.1109/ROBIO.2012.6491053.

Normey-Rico, J.E., Alcalá, I., Gómez-Ortega, J., and Camacho, E.F. (2001). Mobile robot path tracking using a robust pid controller. *Control Engineering Practice*, 9(11), 1209 – 1214. doi:https://doi.org/10.1016/S0967-0661(01)00066-1. PID Control.