

LPV CONTROLLER DESIGN FOR TERRESTRIAL MOBILE ROBOT WITH TILT COMPENSATION

CLAUDIO G. M. FILHO^{1*}, MARCUS D. N. FORTE^{1*}, FABRÍCIO G. NOGUEIRA^{1*}, WILKLEY B. CORREIA^{1*}, BISMARCK C. TORRICO^{1*}

**Campus do Pici – Caixa Postal 6001 CEP 60455-760 – Fortaleza-CE*

Emails: claudio@dee.ufc.br, davi2812@dee.ufc.br, fnogueira@dee.ufc.br, wilkley@dee.ufc.br, bismark@dee.ufc.br

Abstract— This work presents an LPV (Linear Parameter Varying) controller design for the trajectory tracking problem of nonholonomic wheeled mobile robots (NWMR). The scheduling varying parameters are defined as the feedforward velocities which the robot has to track according to a desired trajectory. The proposed model also includes the robot's motion over a tilted environment. A tilt-compensation scheme is used to maintain the ground speed of the robot if the robot is subjected to an inclination. The compensation is computed on the feedforward velocities. For practical purposes, the inclination angle is measured by an Inertial Measuring Unit. Thus, a sensor fusion model is proposed to fuse data from an accelerometer and a gyroscope. Simulation results are presented in order to evaluate the performance of the controller, sensor fusion and tilt-compensation models.

Keywords— LPV Control, Mobile Robot Control, Sensor Fusion

Resumo— Este trabalho descreve um controlador LPV (Linear Parameter Varying) projetado para o problema de seguimento de trajetória de robôs móveis não holonômicos. Os parâmetros variantes são definidos como sendo as velocidades de *feedforward* que o robô deverá seguir para o seguimento de uma trajetória desejada. O trabalho proposto também inclui um modelo de navegação de robôs não-holonômicos para planos inclinados. Um esquema de compensação de inclinação é usado para manutenção da velocidade planar (*Ground Speed*) caso o robô esteja se movimentando em um plano inclinado. A compensação é realizada para as velocidades de *feedforward*. Para aplicações práticas, o ângulo de inclinação é medido por uma IMU (*Inertial Measuring Unit*). Portanto, um modelo de fusão sensorial é proposto fundir dados de um acelerômetro e um giroscópio. Resultados de simulação são apresentados para validação do desempenho do controlador, da fusão e da compensação de inclinação.

Palavras-chave— Controle LPV, Controle de Robô Móvel, Fusão Sensorial

1 INTRODUCTION

Research on autonomous vehicles, robots or automated-guided vehicles (AGV) have increased over the last years as the technology further facilitated embedded solution of complex control systems and algorithms. For such applications, it is useful the study of different control strategies which would better suit a specific problem. Terrestrial autonomous vehicles, for example, may have its trajectory generated by a path planning algorithm based on some objective, e.g., exploration and obstacle avoidance. The trajectory tracking of the planned path is a classical mobile robot problem that has been studied over the past decades. Path planning in uncontrolled environments have shown to be a challenging task depending on the constraints of the autonomous vehicles, which further motivates the study of control strategies for the trajectory tracking problem.

Many different strategies for trajectory tracking of autonomous vehicles have been developed over the last years. In (Ramírez-Martínez et al., 2014) an adaptive controller is proposed based on the changing motion of a non-holonomic wheeled mobile robot (NWMR). In (Bui et al., 2013) a Lyapunov-based controller is proposed for trajectory tracking of an AGV using a laser sensor to follow a desired trajectory. In (Fan et al., 2012) a straight-line following algorithm is proposed us-

ing PID controller and Fuzzy control. A Linear Parameter Varying (LPV) controller for a four-wheeled omnidirectional vehicle is proposed in (Rotondo et al., 2015) as well as a fault-tolerant scheme that maintain system stability in a case of a faulty actuator.

Trajectory tracking of a NWMR is a non-linear problem that may be tackled using either non-linear (Klanar et al., 2005) or linear (Forte et al., 2018) controllers over a linearized operating point. A Lyapunov-based stability study is performed in (Tzafestas, 2014), where it is proved that a simple set of positive error state feedback gains may guarantee system stability. The main problem, however, lies in maintaining a desired closed-loop performance. Nonetheless, if the non-linear system is modeled as an LPV model, an LPV controller is able to guarantee a desired performance over a space of operating points (Boyd et al., 1994), (Nogueira et al., 2018).

Yet another practical problem arises when one needs to correctly estimate the pose of the mobile robot. Some works propose strategies that increase localization precision based on vision sensors (Alatise and Hancke, 2017), Global Positioning System (GPS) (Skobeleva et al., 2016), or Light Detection and Ranging (LiDAR) systems (Lima et al., 2016). Most works use the effective sensor fusion techniques and Inertial Measuring Units (IMUs) to combine sensory data from

different sensors so that the limitation of individual sensors are compensated. The work in (Forte et al., 2018), for example, proposes a sensor fusion technique to estimate the heading of a NWMR using a magnetometer and a gyroscope which copes with the heavily noisy magnetic measurements and biased gyroscopic measurements.

In this paper it is considered a three-dimensional model of a NWMR able to include the robot's movement over a tilted terrain. A LPV-based control scheme driven to a NWMR subject to variations on reference velocities, which may be caused by both reference changes and tilt compensation. The former from the forward portion control law the latter from pitch variations when it moves over irregular terrain. For such purpose, an inertial measurement unit (IMU) under a sensor fusion model framework is applied to accurately estimate the actual pitch angle.

2 MOBILE ROBOT KINEMATIC MODEL

Traditional NWMR forward kinematic models take in linear and angular velocities and describe motion over a navigational plane XY , through the well-established model (Kanayama et al., 1990)

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

where v and ω are linear velocities and angular velocities with respect to the body frame of the robot, and $[\dot{x} \ \dot{y} \ \dot{\theta}]$ are velocities on the navigational frame. Such model can be extended to take into account the pitch angle θ so inclination may be included. Such approach is useful if a NWMR is intended to navigate through irregular terrains so climbing moves (up and down) occur naturally as the mobile follows a desired path. Using an inertial to navigational frame rotation matrix, one describes the kinematic motion with respect to a linear velocity on the body frame by

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{bmatrix} = \begin{bmatrix} \cos \psi & -\sin \psi & 0 \\ \sin \psi & \cos \psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos \theta & 0 & -\sin \theta \\ 0 & 1 & 0 \\ \sin \theta & 0 & \cos \theta \end{bmatrix} \begin{bmatrix} v \\ 0 \\ 0 \end{bmatrix} \quad (2)$$

$$= \begin{bmatrix} \cos \psi \cos \theta & -\sin \psi & -\cos \psi \sin \theta \\ \sin \psi \cos \theta & \cos \psi & -\sin \psi \sin \theta \\ \sin \theta & 0 & \cos \theta \end{bmatrix} \begin{bmatrix} v \\ 0 \\ 0 \end{bmatrix} \quad (3)$$

$$= \begin{bmatrix} \cos \psi \cos \theta \\ \sin \psi \cos \theta \\ \sin \theta \end{bmatrix} \begin{bmatrix} v \\ 0 \\ 0 \end{bmatrix} \quad (4)$$

Fig. 1 shows the angles associated with the studied NWMR kinematic model. Note that pitch angle θ for rotation does not follow traditional aeronautics direction, but the opposite one instead, so

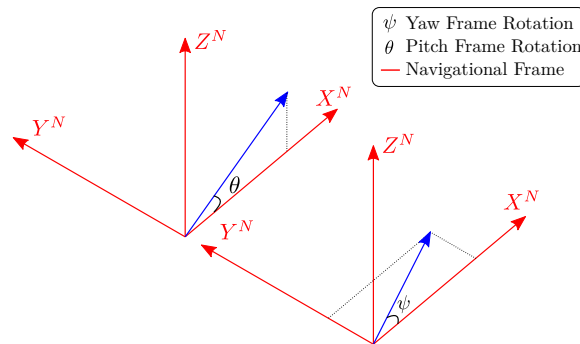


Figure 1: Robot Frame Rotation

one should take its transpose in order to get the model in Eq. (2).

The angular velocity is simply included in (4) to describe mobile robot yaw motion. The complete model is given by

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \cos \psi \cos \theta & 0 \\ \sin \psi \cos \theta & 0 \\ \sin \theta & 0 \\ 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} v \\ \omega \end{bmatrix} \quad (5)$$

Model (5) describes three-dimensional motion of a NWMR subjected to pitch and yaw rotations. Although the new state z appears explicitly in the model, it should not affect any trajectory tracking algorithm used for (1). This is simply due to the fact that there is no point in tracking z reference as the NWMR is physically unable to follow it.

Reference tracking controllers are commonly designed using (1) aiming to obtain a nonlinear control law. Performance tracking from linear control specs are reachable for a linear model obtained by taking body frame referenced errors for the XY -plane (Klančar et al., 2005). The same principle is extended herein to include pitch variations leading to the extended error model

$$e = \begin{bmatrix} e_1 \\ e_2 \\ e_3 \end{bmatrix} = \begin{bmatrix} \cos \psi & \sin \psi & 0 \\ -\sin \psi & \cos \psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x_r - x \\ y_r - y \\ \psi_r - \psi \end{bmatrix}, \quad (6)$$

where $p_r = [x_r \ y_r \ \theta_r]^T$ are reference trajectory coordinates and $p = [x \ y \ \theta]^T$ are current robot coordinates. Taking time derivative of (6) one gets

$$\begin{bmatrix} \dot{e}_1 \\ \dot{e}_2 \\ \dot{e}_3 \end{bmatrix} = \begin{bmatrix} \text{cose}_3 & 0 \\ \text{sine}_3 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v_r \\ \omega_r \end{bmatrix} + \begin{bmatrix} -1 & e_2 \\ 0 & -e_1 \\ 0 & -1 \end{bmatrix} \begin{bmatrix} u_v \\ u_\omega \end{bmatrix}. \quad (7)$$

A commonly used control input velocities which assures stability (Kanayama et al., 1990), (Tzafestas, 2014) of (7) is given by

$$\begin{aligned} u_v &= v_r \text{cose}_3 - v_v \\ u_\omega &= \omega_r - v_\omega \end{aligned} \quad (8)$$

where v_r, ω_r are feedforward velocities and v_v, v_ω are velocities from a closed-loop controller. Using

inputs (8) on (7) yields following non-linear model

$$\begin{bmatrix} \dot{e}_1 \\ \dot{e}_2 \\ \dot{e}_3 \end{bmatrix} = \begin{bmatrix} 0 & u_\omega & 0 \\ -u_\omega & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} e_1 \\ e_2 \\ e_3 \end{bmatrix} + \begin{bmatrix} 0 \\ \sin e_3 \\ 0 \end{bmatrix} u_v + \begin{bmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v_v \\ v_\omega \end{bmatrix}. \quad (9)$$

Finally, linearizing system (9) over operating point $\dot{e}_1 = \dot{e}_2 = \dot{e}_3 = v_v = v_\omega = 0$ leads to

$$\dot{\bar{e}} = \begin{bmatrix} 0 & \omega_r & 0 \\ -\omega_r & 0 & v_r \\ 0 & 0 & 0 \end{bmatrix} \bar{e} + \begin{bmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v_v \\ v_\omega \end{bmatrix}, \quad (10)$$

where ω_r and v_r are feedforward linear and angular velocities from inverse kinematics and v and ω are linear and angular velocities of the mobile robot. The general reference tracking scheme of a NWMR is shown in Figure 2. It can be readily

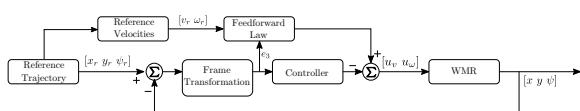


Figure 2: Reference Tracking Scheme

seen from (10) that different feedforward velocities lead to different models. Linear controller from (Forte et al., 2018) is suitable for a constant set of velocities but may exhibit lack of robustness if variations in feedforward inputs are needed. So that, natural choices include either non-linear or LPV controllers which are capable of keeping a desired performance over a set of operating conditions. This paper follows the LPV approach to properly schedule the linear velocity, as becomes clear in the controller design at the next section.

2.1 Tilt Compensation

Feedforward inputs v_r, ω_r are obtained from the feedforward loop according to the desired reference tracking from the path planning layer. Often simple planning rules do not consider elevation which may lead to velocities change in the lower control layer as tilt variations happen. Therefore, in order to track the XY -plane linear velocity from the feedforward control law it is mandatory to monitor the pitch angle from the mobile robot, so that feedforward inputs may be updated. For such goal it is proposed in this paper control law

$$v_r = \frac{v_{r0}}{\cos \theta} \quad (11)$$

where v_{r0} is the base feedforward velocity given by the reference trajectory block. Proposed feedforward law seamlessly compensates for irregularity on terrain as if the robot were navigating over a perfect plane, i.e., it is able to keep constant XY -plane velocity (ground speed).

3 LPV CONTROLLER DESIGN

The discussion in the previous section clearly provokes changes in linear reference velocity. So that, it is desirable the controller to cover all possible feedforward velocities as irregularities on terrain appear. For this goal, a polytopic LPV approach was chosen with polytopes defined by matrix (10) at each possible operating points.

The LPV system is defined by

$$\begin{aligned} \dot{\bar{e}} &= A(p(t))\bar{e} + Bu = \\ &= \begin{bmatrix} 0 & \omega_r & 0 \\ -\omega_r & 0 & p(t) \\ 0 & 0 & 0 \end{bmatrix} \bar{e} + \begin{bmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix}, \quad (12) \end{aligned}$$

where $p(t) \in [v_{rmin} \ v_{rmax}]$ defines the minimum and maximum possible feedforward velocities. The objective is to find an LPV controller $K(p(t))$ that guarantees both stability and performance over the polytopic region. Moreover, although there is no assumption that ω_r varies, the robot does exhibit no problems when performing non-straight trajectories.

This problem can be approached using Lyapunov stability, linear matrix inequalities (LMIs) and convex optimization tools. Given a system closed-loop $\dot{x} = (A_n - BK)x$, the existence of a quadratic function on the form

$$V(x) = x^T W x > 0, \quad (13)$$

the satisfaction of

$$\dot{V}(x) = x^T [(A_n - BK)^T W + W(A_n - BK)] x < 0, \quad (14)$$

where W is a semi-definite-positive matrix, is a sufficient condition that assures stability of the system (Boyd et al., 1994). The polytopic problem is stated when A_n refers to each vertex of the polytopic region, whose solution is to find a single matrix W that satisfies (13) and (14) of each vertex.

Inequalities (13) and (14) can be rewritten in an LMI form as

$$W > 0 \quad (15)$$

$$A_n W + W A_n - B Y - Y^T B^T < 0 \quad (16)$$

with $K = YW^{-1}$. The convex optimization solvers would compute W and Y which would yield the desired controller, placing the closed-loop poles within the left-half plane. Solving the aforementioned inequalities alone, however, would simply return stabilizing state feedback gains. It is preferable to limit the region where the closed-loop poles are placed using the \mathcal{D} -Stability so that some performance criteria are achieved.

3.1 \mathcal{D} -Stability

The \mathcal{D} -Stability concept consists in defining a custom stability region instead of the entire complex

left-half-plane in order to assure desired closed-loop behavior. The chosen regions are described in an LMI form as

$$A_n W + W A_n^T - B Y - Y^T B^T - 2hW < 0 \quad (17)$$

$$\begin{bmatrix} \sin \phi \mathcal{A}_1 & \cos \phi \mathcal{A}_2 \\ \cos \phi \mathcal{A}_3 & \sin \phi \mathcal{A}_4 \end{bmatrix} < 0 \quad (18)$$

$$\begin{bmatrix} -rW & W A_n^T - Y^T B^T \\ A_n W - B Y & -rW \end{bmatrix} < 0 \quad (19)$$

where

$$\mathcal{A}_1 = (A_n W + W A_n^T - B Y - Y^T B^T)$$

$$\mathcal{A}_2 = (A_n W - W A_n^T - B Y + Y^T B^T)$$

$$\mathcal{A}_3 = (W A_n^T - A_n W + B Y - Y^T B^T)$$

$$\mathcal{A}_4 = (A_n W + W A_n^T - B Y - Y^T B^T).$$

LMI (17) defines the region to the left of vertical line $\sigma = h$. LMI (18) defines a conic section with its vertex at the origin of complex plane and internal angle of 2ϕ . Finally, LMI (19) defines a circular region centered at the origin with a radius of r . The aforementioned LMIs effectively substitute inequality (16) in order to define the \mathcal{D} -Stability region.

Since it is a polytopic problem, \mathcal{D} -Stability has to be solved for each vertex. The LPV controller is computed by using the affine relation $Y = Y_0 + \gamma Y_1$, and consequently, $K(\gamma) = K_0 + \gamma K_1$, where $K_0 = Y_0 W^{-1}$ and $K_1 = Y_1 W^{-1}$. The problem was modeled using YALMIP parser from (Lofberg, 2004) and solved using the semi-definite programming solver SeDuMi 1.3 from (Sturm, 1999). The proposed controller diagram scheme is shown in Figure 3.

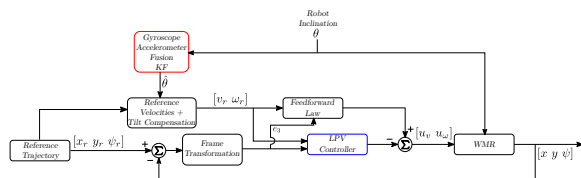


Figure 3: Proposed Scheme

4 FUSION MODEL FOR PITCH ESTIMATION

The estimation of an object's pitch angle may be obtained using accelerometers or gyroscopes. Since accelerometer sensors are able to detect the acceleration of gravity along its axes, it is possible to use such effect to compute either the roll or pitch angles based on the distribution of the gravity vector along the body axes of an IMU. This behavior can be modeled using similar rotation matrices to the ones used for deriving the inclination model of the mobile robot. Let \vec{g} be the gravity vector defined on an inertial frame, the

accelerometer output a_m of a rotated IMU device due to gravity is given by (note the gravity does not affect yaw rotation, thus it is not used)

$$\begin{aligned} a_m &= R_I^B(\phi) R_I^T(\theta) \vec{g} \\ &= \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \phi & \sin \phi \\ 0 & -\sin \phi & \cos \phi \end{bmatrix} \begin{bmatrix} \cos \theta & 0 & -\sin \theta \\ 0 & 1 & 0 \\ \sin \theta & 0 & \cos \theta \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ g \end{bmatrix} \\ &= \begin{bmatrix} -\sin \theta \\ \cos \theta \sin \phi \\ \cos \theta \cos \phi \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ g \end{bmatrix} = \begin{bmatrix} A_x \\ A_y \\ A_z \end{bmatrix} \end{aligned} \quad (20)$$

The computation of pitch angle θ can be achieved using

$$\theta_a = \text{atan2}(-A_x, \sqrt{A_y^2 + A_z^2}). \quad (21)$$

Accelerometer measurement vector a_m includes unwanted influences such as Coriolis effect, components due to the acceleration of the IMU device, heavy white-Gaussian noise and bias, depending on the quality of the IMU. Solely relying on (21) would compromise any system which uses θ i.e. the controller and pose estimation (5).

Gyroscopes, on the other hand, normally measures angular velocities without much noise. The pitch angle can be computed using simple integration

$$\theta_g = \int \omega_y. \quad (22)$$

However, gyroscopes present a time-varying bias intrinsic to its construction. Integration of a signal with a bias would corrupt any estimation after a while.

This motivates the use of a sensor fusion model that would combine the advantages of both accelerometer and gyroscope. The fusion model is similar to the yaw estimation presented in (Forte et al., 2018), but with the measurement angle coming from the accelerometer. The model consists in estimating the time-varying bias from the gyroscope and removing it from the integration based on the accelerometer measurement. The state-space discrete-time model is described by

$$\begin{bmatrix} \theta \\ b_g \end{bmatrix}_k = \begin{bmatrix} 1 & -T_s \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \theta \\ b_g \end{bmatrix}_{k-1} + \begin{bmatrix} T_s \\ 0 \end{bmatrix} \omega_y + \eta'_g \quad (23)$$

$$y_k = \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} \theta \\ b_g \end{bmatrix}_{k-1} + \eta_a \quad (24)$$

where T_s is the sampling time, b_g is the estimated gyroscope bias, ω_y is the angular velocity around the pitch axis, η'_g is the process noise associated with the gyroscope measurement and η_a is the measurement noise from accelerometer.

Kalman filtering is commonly used on sensor fusion since the stochastic properties of the problem are embedded into the optimal estimator. Thus, it is possible to include the noise properties of a sensor into the filter's model by com-

puting the variance of its noise. Its algorithm requires the covariance matrices associated with the model (23), (24) as well as the model itself. The covariance matrices used are the same as in (Forte et al., 2018), with

$$Q_n = q \begin{bmatrix} \frac{T_s^2}{2} & 0 \\ 0 & T_s \end{bmatrix}, R_n = Var(\eta_a) \quad (25)$$

where Q_n is the process covariance matrix and R_n is the measurement noise covariance with q as a tuning parameter adjust the filter's noise rejection and bandwidth. The value used for the current study is $q = 0.01$.

5 SIMULATION RESULTS

Simulations were performed for the proposed LPV controller and compared with the well known non-linear one (Klancar et al., 2005) with tuning gains $g_k = 40$ and $\zeta = 0.6$ and the recently proposed linear controller from (Forte et al., 2018). All the controllers in this study have the nonlinear portion control law modified to include (11) so the pitch angle information is fed back. For such analysis, two different trajectory tracking paths with tilt variations were considered.

The LPV controller was designed so that closed-loop poles are within the \mathcal{D} -region defined by $h = -1.7$, $r = 5$ and $\phi = 25^\circ$. The chosen interval for the parameter is $p(t) \in [0.2 \ 0.5]$. Computed LPV controller is given by

$$K(p(t)) = \begin{bmatrix} 2.85 & 0 & 0 \\ 0 & 41.54 & 4.66 \end{bmatrix} + \quad (26)$$

$$p(t) \begin{bmatrix} 0 & 0 & 0 \\ 0 & -4.37 & 7.38 \end{bmatrix}.$$

The first trajectory consists of four straight lines with an inclination of $\theta = \frac{\pi}{6}$ upward, followed by another of $\theta = -\frac{\pi}{6}$ downward. Base feedforward velocities are varied throughout the reference tracking. Fig. 4 shows the top-down view of the pose of the robot and Fig. 5 shows the three-dimensional pose. Feedforward velocities with tilt-compensation are presented in Fig. 6. Control inputs are shown in Fig. 7, pose errors are shown in Fig. 8 and the sensor fusion estimation of robot's pitch is shown in Fig. 9. Note that the robot feedforward velocities belong to the polytope of the LPV controller. It can be seen that the LPV controller is superior to the linear controller as the latter is designed considering only a single set of feedforward velocities. The non-linear controller also exhibited similar performance compared to LPV controller.

For the second simulation, two straight lines and a curved trajectory were tracked. The same controllers from first simulation were used. Figure 10 and 11 show the trajectory, and Fig. 12 shows

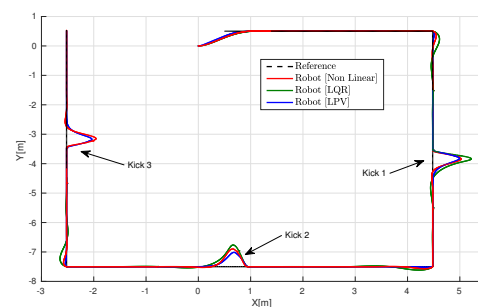


Figure 4: Trajectory Tracking Top-Down View

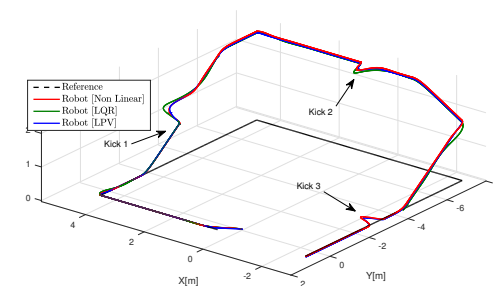


Figure 5: Trajectory Tracking 3D View

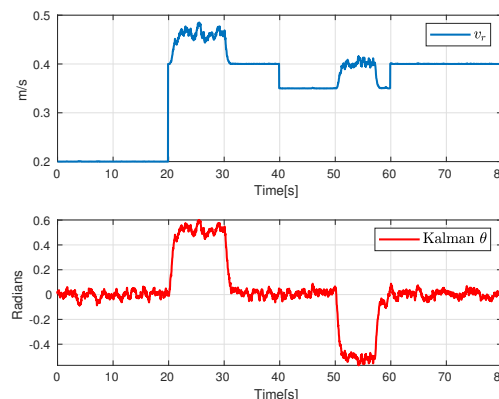


Figure 6: Feedforward Reference Velocities

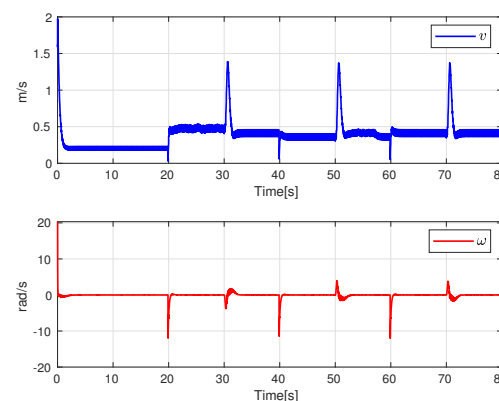


Figure 7: Control Inputs

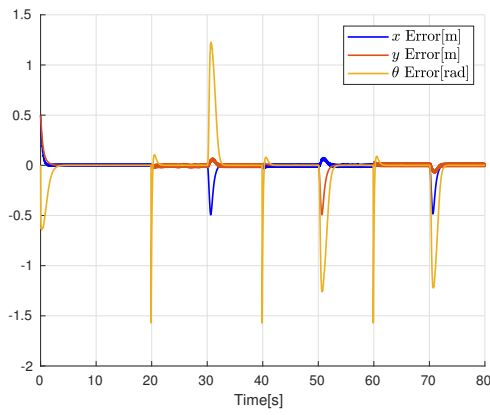


Figure 8: Pose Errors

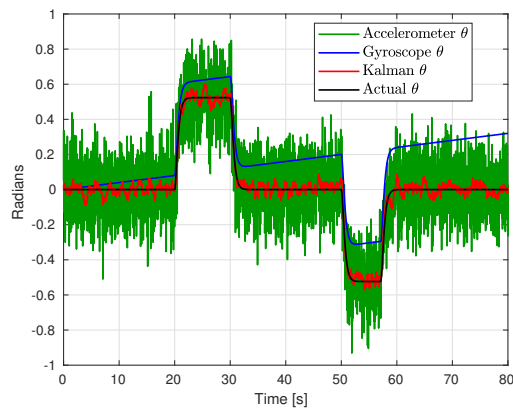


Figure 9: Robot Pitch Estimation

the pose errors. Note that a similar behavior is achieved when compared to the previous trajectory. The LPV controller is also able to tackle curved trajectories successfully. There is no need to include feedforward angular velocities on model (12) as long as small angular velocities are used.

6 CONCLUSIONS

The paper presented a NWMM model that considers motion over tilted planes. A tilt-compensation scheme is used to maintain a constant robot's ground speed with respect to its navigational frame of reference in case of trajectory tracking over a tilted surface. An LPV controller is designed so that an interval of feedforward velocities may be used while maintaining a desired closed-loop performance. The model is validated by introducing disturbance on the pose of the robot and by observing its rejection.

The proposed controller is compared to both a classical non-linear and a linear schemes. While the non-linear controller may be tuned to achieve faster disturbance rejection, its tuning rules are not as intuitive as the linear and LPV controllers, whose performance are tuned using the

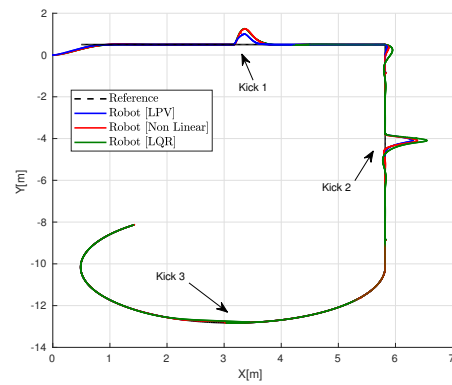


Figure 10: Trajectory Tracking Top-Down

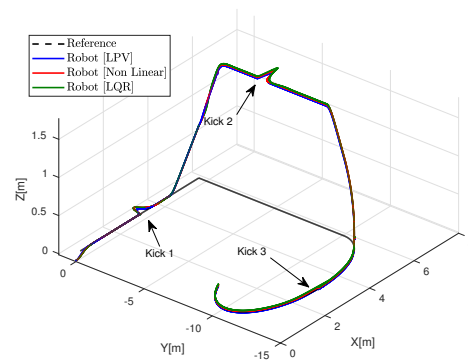


Figure 11: Trajectory Tracking 3D

\mathcal{D} -Stability Region.

The curves showed the faster response of the LPV controller and that it is not affected by the different settings of feedforward velocities as long as they are within the specified polytope.

Future work will study different NWMM so that further non-linearities are included within an LPV model to increase the controller's robustness, as well as the LPV synthesis to merge controller design with NWMM model in order to cope with uncertainties on the robot's dimensions.

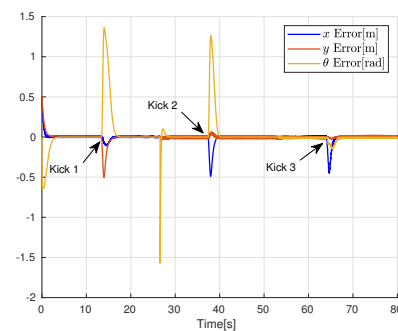


Figure 12: Pose Errors

ACKNOWLEDGMENT

This work has been fully supported by Energia Pecém along with the Program of Research and Technical Development from ANEEL (The Brazilian National Electrical Energy Agency) under the grant number PD-07267-0016/2018, for what authors are thankful.

References

- Alatise, M. B. and Hancke, G. P. (2017). Pose estimation of a mobile robot based on fusion of imu data and vision data using an extended kalman filter, *Sensors* **17**(10).
- Boyd, S., El Ghaoui, L., Feron, E. and Balakrishnan, V. (1994). *Linear Matrix Inequalities in System and Control Theory*, Vol. 15 of *Studies in Applied Mathematics*, SIAM, Philadelphia, PA.
- Bui, T. L., Doan, P. T., Kim, H. K. and Kim, S. B. (2013). Trajectory tracking controller design for agv using laser sensor based positioning system, *2013 9th Asian Control Conference (ASCC)*, pp. 1–5.
- Fan, K., Yang, Q., Li, P. and Yan, W. (2012). On the design of agv obstacle avoidance system based on fuzzy-pid dual-mode controller, *2012 IEEE Conference on Control, Systems Industrial Informatics*, pp. 54–58.
- Forte, M. D., Correia, W. B., Nogueira, F. G. and Torrico, B. C. (2018). Reference tracking of a nonholonomic mobile robot using sensor fusion techniques and linear control, *IFAC-PapersOnLine* **51**(4): 364 – 369. 3rd IFAC Conference on Advances in Proportional-Integral-Derivative Control PID 2018.
- Kanayama, Y., Kimura, Y., Miyazaki, F. and Noguchi, T. (1990). A stable tracking control method for an autonomous mobile robot, *Proceedings., IEEE International Conference on Robotics and Automation*, pp. 384–389 vol.1.
- Klancar, G., Matko, D. and Blazic, S. (2005). Mobile robot control on a reference path, *Proceedings of the 2005 IEEE International Symposium on, Mediterrean Conference on Control and Automation Intelligent Control, 2005.*, pp. 1343–1348.
- Lima, T. A., do Nascimento Forte, M. D., Nogueira, F. G., Torrico, B. C. and de Paula, A. R. (2016). Trajectory tracking control of a mobile robot using lidar sensor for position and orientation estimation, *2016 12th IEEE International Conference on Industry Applications (INDUSCON)*, pp. 1–6.
- Lofberg, J. (2004). Yalmip : a toolbox for modeling and optimization in matlab, *2004 IEEE International Conference on Robotics and Automation (IEEE Cat. No.04CH37508)*, pp. 284–289.
- Nogueira, F. G., Junior, W. B., da Costa Junior, C. T. and Lana, J. J. (2018). Lpv-based power system stabilizer: Identification, control and field tests, *Control Engineering Practice* **72**: 53 – 67.
- Ramírez-Martínez, O. L., Martínez-García, E. A., Mohan, R. E. and Sheba, J. K. (2014). Mobile robot adaptive trajectory control: Non-linear path model inverse transformation for model reference, *2014 13th International Conference on Control Automation Robotics Vision (ICARCV)*, pp. 877–881.
- Rotondo, D., Puig, V., Nejjari, F. and Romera, J. (2015). A fault-hiding approach for the switching quasi-lpv fault-tolerant control of a four-wheeled omnidirectional mobile robot, *IEEE Transactions on Industrial Electronics* **62**(6): 3932–3944.
- Skobeleva, A., Ugrinovskii, V. and Petersen, I. (2016). Extended kalman filter for indoor and outdoor localization of a wheeled mobile robot, *2016 Australian Control Conference (AuCC)*, pp. 212–216.
- Sturm, J. F. (1999). Using sedumi 1.02, a matlab toolbox for optimization over symmetric cones, *Optimization Methods and Software* **11**(1-4): 625–653.
- Tzafestas, S. G. (2014). 2 - mobile robot kinematics, in S. G. Tzafestas (ed.), *Introduction to Mobile Robot Control*, Elsevier, Oxford, pp. 31 – 67.