Autonomous Vehicle State Estimation using a LPV Kalman Filter and SLAM

Shivam Chaubey[1] and Vicenç Puig[1]

Abstract— **This paper presents an optimal approach for state estimation and Simultaneous Localization and Mapping (SLAM) correction using Kalman gain obtained via Linear Matrix Inequality (LMI). The technique utilizes a Linear Parameter Varying (LPV) represention of the system, which allows to model the complex non-linear dynamics in a way that linearization is not required for the estimator or controller design. In addition, the LPV polytopic representation is exploited to obtain a real-time Kalman gain, avoiding expensive optimization of LMIs at every step. The estimation schema is integrated with a Non-linear Model Predictive Control (NMPC) in charge of controlling the vehicle. For the demonstration, the approach is tested in the simulation and for the practical validity, a small-scale autonomous car is used.**

I. Introduction

To solve the issue of traffic accidents and congestion, autonomous vehicles provide a promising solution. Research work on this technology has immensely progressed from perception algorithms to vehicle control algorithms. The perception stack exploits the sensor measurements to provide the vehicle state as well as environmental information. The correction of these measurements is very important task to allow the vehicle to navigate safely within the environment. On the other hand, the planner and the controller use this information to plan the motion and achieve the desired trajectory while maintaining some objectives such as speed, time to reach the destination, and comfort. For such reasons, the measurement filtering and the obtaining of unmeasured states are paramount (e.g., lateral velocity in many cases can not be measured directly, which is important for lateral control). The advanced controllers (as e.g. NMPC) require a full state observation, which can be achieved using a state estimator in the loop for the full-body control.

The state estimation of an autonomous vehicle involves an algorithm that fuses the raw information provided by the sensors, which is affected by noise, and the system model, which is affected by some uncertainty represented as a disturbance. The estimation algorithm considers measurement model, vehicle model, sensor uncertainty, and model perturbation to estimate the correct states. The standard approach is based on some form of Kalman filter, originally developed by R.E Kalman [1]. Kalman filter provides an optimal estimate in the least square sense assuming that the model perfectly matches with the real system and considering that the noises/disturbances are Gaussian and their covariances are known. The first assumption is not always true, especially in nonlinear systems with complex dynamics. A variant named Extended Kalman Filter (EKF) for the nonlinear systems has been developed, which provides the solution by linearizing the system and the measurement model around the current states. The linearization is based on a Taylor series expansion, rejecting higher-order terms. The linearization process does not preserve the random variable distributions of the states and the measurements as Gaussian, which means that the optimality condition is not guaranteed. For the systems with a complex nonlinear model such as vehicle, the assumption of a small range of operation for a linear model acquired by linearizing the nonlinear model is not valid. For all these reasons, new methods for extending Kalman filters to non-linear systems are required.

The LPV framework offers a systematic approach to obtain a multi-model system from the original mathematical model of the system, see as e.g. for more details [2]. The main feature of this approach is that it allows representing the model of the nonlinear system as a set of linear system models and the overall LPV model of the system is achieved by the polytopic combination of these linear system models. This allows extending the LMI design procedures for control and estimation to the non-linear systems represented in polytopic LPV form. Moreover, there exist a systematic procedures for generating the LPV model for the non-linear model [3] and approximating it in a polytopic way [4].

In this paper, the state estimation of an autonomous vehicle is developed using an approach alternative to the use of EKF. Moreover, this approach will be combined with an imprecise SLAM algorithm that provides a rough pose estimation. Other dynamic states such as longitudinal velocity, angular velocity will be directly measured from the sensors. Then, the rough pose obtained from the SLAM algorithm, direct measurements from sensors will be corrected using the LPV Kalman filter design. A similar work [5] is done to improve the approximation error by utilizing Takagi-Sugeno model which is analogous to LPV form. However, this approach considers a limited and known number of landmarks and neither ensures optimality. Another work [6] has shown the performance in the simulation with different application.

The structure of the paper is the following: Section II describes the proposed approach and presents the autonomous vehicle considered as a case study. Section III presents the proposed LPV Kalman filter for the state estimation. Section IV presents the simulation and experimental results using the considered case study. Finally, Section V draws the main conclusions of the paper and proposes future research paths.

[1]S. Chaubey and V. Puig are with the Institute of Robotics and Industrial Informatics (IRI-UPC), Barcelona, Spain. E-mail: vicenc.puig@upc.edu

II. Proposed Approach

Figure 1 presents the overall architecture and integration of different modules for the autonomy. The proposed pipeline also include an NMPC controller, details can be found in [7]. To generalize the SLAM algorithm, i.e. free from number of

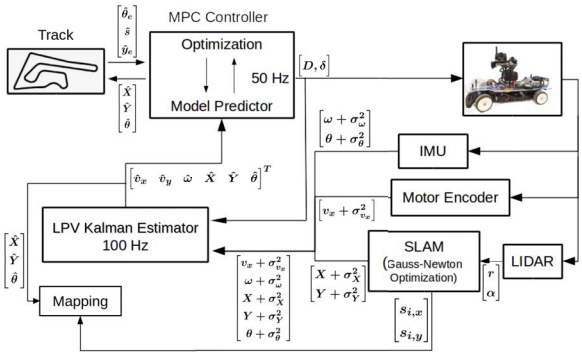


Fig. 1: Architecture of the software modules, including information flow and a conceptual overview of the interconnections.

landmarks and environment, the rough pose estimation is based on LIDAR scan end-point matching by exploiting the Gauss-Newton optimization process. This matching approach allows to estimate the pose without any prior information of the environment or landmarks. The scan end-points matching is similar to the work done in the paper [8] and due to the limitation of pages, the SLAM algorithm is not presented here since it is not the focus of this work. In this development, a LIDAR's scan end-points are exploited for observing the environment. Instead, a camera can also be used for the rough localization using either feature matching technique [9] or deep learning technique [10]. Then, to correct the rough pose from the SLAM and acquired sensor measurements, a LPV modeling approach is considered to design an optimal and stable Kalman estimator.

A. Considered Autonomous Vehicle

To validate the proposed approach, a case based on a small scale autonomous car[1] is used. The states of the car have to be estimated using available sensors and further this information will be exploited by full-body controller. The dynamic states, such as longitudinal velocity can be roughly estimated using the radius and RPM of the rear wheel measured by the motor encoder. Angular velocity can be measured by an IMU sensor. The rough position of the vehicle can be obtained using the scan end-point matching algorithm and orientation from IMU.

As the estimator is based on a system model the dynamics of the system are derived using bicycle model [11]. The vehicle model includes kinematic as well as dynamic equations represented by

$$
\begin{aligned}
\dot{v_x} &= \frac{1}{m}(F_{rx} - F_{flat}\sin(\delta) + m v_y \omega) \\
\dot{v_y} &= \frac{1}{m}(F_{flat}\cos(\delta) + F_{ry} - m v_x \omega) \\
\dot{\omega} &= \frac{1}{I_z}(l_f F_{flat}\cos(\delta) - l_r F_{ry}) \\
\dot{X} &= v_x cos(\theta) - v_y sin(\theta) \\
\dot{Y} &= v_x sin(\theta) + v_y cos(\theta) \\
\dot{\theta} &= \omega
\end{aligned}
\tag{1}
$$

where there are six states $[v_x, v_y, \omega, X, Y, \theta]$ of the vehicle. The lateral velocity v_y can not be measured and will be estimated. F_{rx} is the longitudinal force in the rear-wheel which consists of force from the motor, drive-line resistance, and drag force. F_{flat} and F_{ry} are the lateral tire force in the front and rear wheel respectively, which is obtained by simplifying Pacejka tire model [12]. Longitudinal force on the front wheel is considered to be negligible since no brake nor torque is applied on it. The forces are given by

$$F_{rx} = (C_{m0} - C_{m_1} v_x)D - C_0 v_x - C_1 - \frac{C_D A \rho v_x^2}{2}$$

$$F_{flat} = 2C_{af}\left(\delta - \arctan\left(\frac{v_y + l_f\dot{\theta}}{v_x}\right)\right)$$

$$F_{ry} = -2C_{ar}\arctan\left(\frac{v_y - l_r\dot{\theta}}{v_x}\right)$$

where δ and D are two control inputs, steering angle and duty cycle respectively. Some of the system parameters are obtained by physical measurement, and the remaining ones by least-squares optimization. The obtained values and description of the parameters are listed in Table I.

Parameters	Values	Description
m	$2.424\ kg$	Mass of the vehicle
l_f	$0.1377\ m$	Distance from CoG to front wheel
l_r	$0.1203\ m$	Distance from CoG to rear wheel
ρ	$1.225\ kg/m^3$	Air density
C_{m0}	$9.4685\ N$	Motor parameter 1
C_{m1}	$0.6672\ kg/s$	Motor parameter 2
C_0	$2.6104\ kg/s$	Resistive driveline parameter
C_1	$-0.00213\ N$	Static friction force
$C_D A$	$0.466\ m^2$	Coefficient of drag multiplied with area
C_{af}	$1.2354\ N/rad$	Front wheel cornering stiffness
C_{ar}	$1.4532\ N/rad$	Rear wheel cornering stiffness
I_z	$0.02\ kg.m^2$	Moment of inertia

TABLE I: Vehicle model parameters.

1) Construction of LPV model: The goal is to derive a LPV model from the nonlinear vehicle dynamics (1) as if the response of the LPV model exactly matches with the response of a nonlinear system within a specified domain with the same input u. To form the LPV model the varying nonlinear terms in the equations are set as varying parameters or scheduling variables. The set of scheduling variables is represented in Table II. These are the state and control

[1]This car is being used in the SEAT autonomous driving challenge (http://carnetbarcelona.com/index.php/2017/07/19/carnet-presents-the-seat-autonomous-driving-challenge/)

variables on which the nonlinear dynamics are dependent and by embedding these terms inside the matrix the LPV form (3) is obtained.

Variables $(\vartheta_i(t))$	$\min(\vartheta_i(t)) = \underline{\vartheta_i}$	$\max(\vartheta_i(t)) = \overline{\vartheta_i}$
v_x	-5.0	5.0
v_y	-3.0	3.0
ω	-1.5	1.5
θ	$-\pi$	0
δ	-0.35	0.35

TABLE II: Scheduling variables i.e. states or inputs on which the system matrix $A(\vartheta)$ and $B(\vartheta)$ are dependent. The maximum and minimum value of these variables can define any system vertices or polytopic systems.

The LPV model can be represented in polytopic form, using often referred as a bounding box method [2], which will be used to formulate the LMIs to obtain offline gain in the Section III. Additionally, the polytopic representation will also allow to compute the online gain using interpolation technique without optimizing the LMIs at each time step.

$$\underbrace{\begin{bmatrix} \dot{v_x} \\ \dot{v_y} \\ \dot{\omega} \\ \dot{X} \\ \dot{Y} \\ \dot{\theta} \end{bmatrix}}_{\dot{x}} = \underbrace{\begin{bmatrix} A_{11} & A_{12} & 0 & 0 & 0 & 0 \\ A_{21} & A_{22} & 0 & 0 & 0 & 0 \\ A_{31} & 0 & 0 & 0 & 0 & 0 \\ A_{41} & A_{42} & 0 & 0 & 0 & 0 \\ A_{51} & A_{52} & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix}}_{A(\vartheta)} \underbrace{\begin{bmatrix} v_x \\ v_y \\ \omega \\ X \\ Y \\ \theta \end{bmatrix}}_{x} + \underbrace{\begin{bmatrix} B_{11} & B_{12} \\ 0 & B_{22} \\ 0 & B_{32} \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}}_{B(\vartheta)} \underbrace{\begin{bmatrix} D \\ \delta \end{bmatrix}}_{u} \quad (3)$$

The polytopic representation of the LPV form is given by,

$$\dot{x}(t) = \sum_{i=1}^{32} h_i(\vartheta(t))\{A_i x(t) + B_i u(t)\} \quad (4)$$

where,

$$h_i(\vartheta(t)) = \mu_{1j}(\vartheta_1(t)) \cdot \mu_{2j}(\vartheta_2(t)) \cdot \mu_{3j}(\vartheta_3(t)) \cdot \mu_{4j}(\vartheta_4(t)) \cdot \mu_{5j}(\vartheta_5(t)), \quad j \in [1,2] \quad (5)$$

and $\vartheta(t) = [\vartheta_1(t), \vartheta_2(t), \vartheta_3(t), \vartheta_4(t), \vartheta_5(t)]$. The term $\mu_{ij}(\vartheta_i(t))$ is the interpolation function of $\vartheta_i(t)$ and $h_i(\vartheta(t))$ is the weight applied to each corresponding system vertex. The scheduling variables can be further represented as a set of interpolation function as follows:

$$\vartheta_i(t) = \mu_{i1}(\vartheta_i(t)) \cdot \underline{\vartheta_i}) + \mu_{i2}(\vartheta_i(t)) \cdot \overline{\vartheta_i} \quad (6)$$

$$\mu_{i1}(\vartheta_i(t)) + \mu_{i2}(\vartheta_i(t)) = 1, \quad i \in [1,2,...,5] \quad (7)$$

where $\underline{\vartheta_i}$, $\overline{\vartheta_i}$ are the minimum and maximum scheduling variable respectively and the $\mu_{ij}(\vartheta_i(t))$ are two interpolation function for each scheduling variables being in total 32 combinations.

2) Measurement model: The rough position of the vehicle is obtained from the SLAM algorithm [8] and other information is directly obtained from various sensors listed in Table III. The discretized state-space measurement model for the system is defined by $y = Cx_k + Du_k + E_v v_k$, where $v_k \in R^{n_y}$ is the measurement noise and $n_y = 5$, E_v is the distribution matrix with appropriate dimension. $D = 0_{5\times 2}$ is taken as there is no interaction from control. C matrix relates the system state to the output measurement y.

$$E_v = \begin{bmatrix} 1&0&0&0&0 \\ 0&1&0&0&0 \\ 0&0&1&0&0 \\ 0&0&0&1&0 \\ 0&0&0&0&1 \end{bmatrix}, \qquad C = \begin{bmatrix} 1&0&0&0&0&0 \\ 0&0&1&0&0&0 \\ 0&0&0&1&0&0 \\ 0&0&0&0&1&0 \\ 0&0&0&0&0&1 \end{bmatrix} \quad (8)$$

The vector $v_k = [\sigma^2_{v_x}, \sigma^2_{\omega}, \sigma^2_{X}, \sigma^2_{Y}, \sigma^2_{\theta}]^T$ represents the noise of each sensor with the variances listed in Table III.

States	Sensors	Variance (σ^2)
v_x	Motor encoder	0.04
ω	IMU	0.0187
X	SLAM	0.0225
Y	SLAM	0.0225
θ	IMU	0.01

TABLE III: Dedicated sensors or algorithm to measure or estimate rough information and their corresponding noise.

3) Observability analysis: Before designing the estimator, the observability of the equation (3) needs to be analysed for singularity. The equation consist of $n_x = 6$ state variables and the observability matrix defined by

$$\mathcal{O} = \left[\, C, CA_d, CA_d^2, \cdots, CA_d^{n_x-1} \,\right]^T \quad (9)$$

If the rank of observability is equal to n_x, then the system is observable. To check the observability the equation (3) is discretized using Euler approximation $A_d = I + A \cdot \Delta t_s$. During the analysis, at certain state the rank of observability matrix $\mathcal{O}$ reaches to singularity, precisely when any of the $v_x = 0, v_y = 0$ and $\delta = 0$. To resolve this issue whenever this state variable attain this value a bias $\epsilon = 0.0001$ is added to this variable.

III. Linear Parameter Varying Kalman Filter Design

For the development of the proposed estimator design, first in Section III-A LMIs for the LPV Kalman filter design are formulated. Second, in Section III-B the measurement noise and system perturbation matrix are provided. Finally, the implementation of the proposed approach and its improvement is discussed in Section III-C.

A. LMI Design Procedure

The following Kalman filter to obtain the $\hat{x}$ estimation is required,

$$\dot{\hat{x}}(t) = (A(\vartheta) - L(\vartheta)C)\hat{x} + (B(\vartheta) - L(\vartheta)D)u + L(\vartheta)y \quad (10)$$

where $A(\vartheta), B(\vartheta)$ can be obtained by using LPV model (3), $L(\vartheta)$ is the online Kalman gain and $D = 0_{5\times 2}$. The above continuous Kalman estimator is discretized for implementation on a real-time system. Now, the aim is to find the optimal $L(\vartheta)$ which converges to the estimation ground truth in presence of sensor noise and system disturbance. To obtain an optimal Kalman gain $L(\vartheta)$, first, LMIs are offline optimized to obtain gain L^{off}. Second, the obtained L^{off} is interpolated in real-time to obtain $L(\vartheta)$ by exploiting relation (4). To formulate LMIs for discretized system, the

continuous LPV model equation (4) is discretized using Euler approximation with sampling time Δt_s:

$$x(k+1) = \sum_{i=1}^{32} h_i(\vartheta(t))\{\underbrace{(I + A_i\Delta t_s)}_{A_{di}} x(k) + \underbrace{B_i\Delta t_s}_{B_{di}} u(k)\} \tag{11}$$

The following LMI optimization problem should be solved at the vertices of the model (11) to design LPV Kalman filter [13]

$$\begin{bmatrix} \gamma I & I \\ I & Y \end{bmatrix} > 0 \tag{12}$$

$$\begin{bmatrix} -Y & YA_d - W^T C_d & YH^T & W^T \\ A_d^T Y - C_d^T W & -Y & 0 & 0 \\ HY & 0 & -I & 0 \\ W & 0 & 0 & -R^{-1} \end{bmatrix} < 0 \tag{13}$$

such that the optimal gain can be found by $L = (WY^{-1})^T$. The solution involves optimization at each time step, due to the varying system matrices (A_d, B_d), which is computationally expensive and slow. Instead, we exploit the method developed in Section II-A.1. Any nonlinear system can be represented in the form (11) which means the offline gain can be found at the system vertices of these systems and later on interpolated online using LPV interpolation function (5) for the LPV model (3). So, for the 5 scheduling variables, 32 system matrices are obtained resulting in 32 LMIs and corresponding offline gains for each system vertex. The gain obtained from solving LMIs will be called $L_i^{off} \in [1, \cdots, 32]$.

B. Disturbance and noise matrix

The disturbance of the vehicle model and sensor noise is modeled by Gaussian distribution whose mean is set to zero and variance is obtained experimentally. The disturbance in the model is considered to be higher than the measurement noise to compensate for uncaptured dynamics during parameter estimation. Following weights were set for the disturbance (Q) and noise (R) matrices respectively:

$$Q = \text{diag}(0.15, 0.05, 0.15, 0.25, 0.25, 0.1) \tag{14}$$

$$R = \text{diag}(0.04, 0.0187, 0.0225, 0.0225, 0.01) \tag{15}$$

C. Switching estimator design

During the experiment phase the scheduling variable $\theta \in [-\pi, 0]$ does not yields a better approximation of nonlinear function. This can be seen by substituting the values in element A_{41}, A_{42}, A_{51} and A_{52} in equation (3) and taking only these subelements:

$$A_{4:5,1:2}(\theta) = \begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix} \tag{16}$$

$$A_{4:5,1:2}(-\pi) = \begin{bmatrix} -1 & 0 \\ 0 & -1 \end{bmatrix}, \quad A_{4:5,1:2}(0) = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \tag{17}$$

The above polytopic systems are true considering the minimum and maximum vertices in cos function but it is not valid for sin function. To accurately model this effect the θ scheduling variable is chosen for each quadrant $\vartheta_{4j} = [0, \frac{\pi}{2}, -\pi, \frac{-\pi}{2}], \quad j \in [1, .., 4]$. This improvement provides the system vertex to reach all the possible values:

$$A_{4:5,1:2}(0) = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad A_{4:5,1:2}(-\pi) = \begin{bmatrix} -1 & 0 \\ 0 & -1 \end{bmatrix} \tag{18}$$

$$A_{4:5,1:2}\left(\frac{\pi}{2}\right) = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, \quad A_{4:5,1:2}\left(\frac{-\pi}{2}\right) = \begin{bmatrix} 0 & -1 \\ -1 & 0 \end{bmatrix} \tag{19}$$

For the implementation, every four sections of the quadrant are considered and the LMIs for all the quadrant are optimized to obtain offline gain $L_{ij}^{off}, \ j \in [1, .., 4]$. Then, particular offline gain is applied according to the region in which the previous yaw estimate lies.

IV. Results

To validate the estimator performance experiments are performed in simulator as well a real environment. First, the estimator is tested using manual control input and once it is proved to be working an NMPC controller is utilized to follow a track. The vehicle is 41 cm long, 21 cm wide, and the track has 50 cm width. The setting of the controller is tuned to complete two laps while achieving a longitudinal velocity of 0.8 m/s, keeping the vehicle inside the track and closer to the center track, shown as dotted line in Figure 2. The objective of the validation is to i) estimate full states correctly, including unmeasured state v_y, in the simulator as well as real vehicle, ii) validate real-time performance. For the simulator, the NRMSE evaluation metric is used to compare the error between the simulated and estimated state, which can be computed by

$$NRMSE = \frac{\sqrt{\sum_{i=1}^{N}(x - \hat{x})^2}}{x_{\max} - x_{\min}} \tag{20}$$

A. Simulation

The vehicle simulator is developed using the non-linear dynamics obtained in Section II-A. The perturbations and noises are injected into the vehicle model and sensor model respectively to simulate the real world. The perturbation and noises are kept a little higher than the actual one to ensure the estimator work even in the worst cases. By analyzing Figure 2, the estimated position is very close to the simulator state of the vehicle. The NRMSE error for all the estimated states is presented in Table IV. Rest of the estimated states are compared in Figure 3. The lateral velocity (v_y) which can not be measured directly has a $NRMSE$ value of 0.0323 (Figure 3b). The errors of all the dynamic estimated states are within a certain range which indicates the estimator is fully working to be tested on the real vehicle.

States	v_x	v_y	ω	X	Y	θ
NRMSE	0.0781	0.0323	0.0913	0.0157	0.0175	0.0198

TABLE IV: NRMSE evaluation of estimated states

B. Real Experiment

Note that the $RMSE$ error is not used here due to a lack of ground truth measurement. The validation for this experiment is done visually. The estimated X-Y position for the real

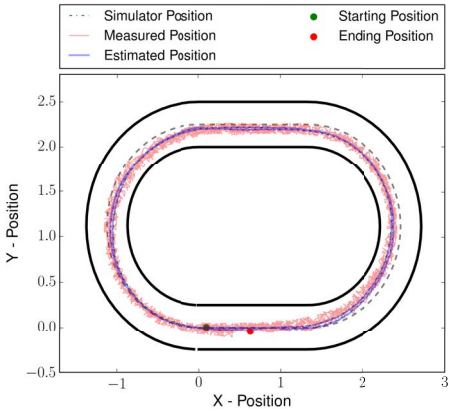
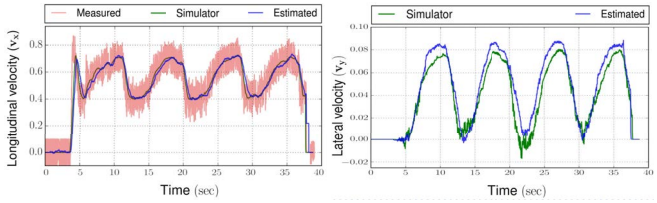
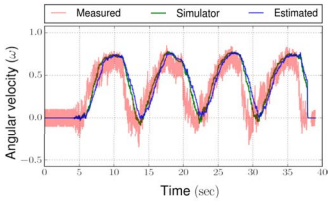


Fig. 2: Estimated trajectory in the presence of imprecise SLAM and sensor measurement during the simulation.

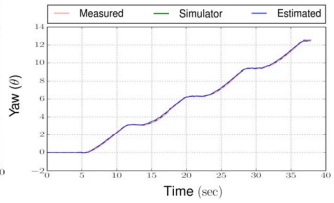


(a) Longitudinal velocity

(b) Lateral velocity



(c) Angular velocity



(d) Yaw

Fig. 3: Estimated states on vehicle simulator

experiment is shown in Figure 4. Some snapshots for visual validation are shown in Figure 5 and compared from Figure 4, it can be noticed that the estimated position of the vehicle matches with the real position of the vehicle in the snapshots. For properly substantiated illustration, the media `https://youtu.be/Oey2ZxsxlnY` shows the estimated states and NMPC controller performance. The media validates the performance of the estimator design. Additional estimated states are shown in Figure 6. The velocity obtained from the motor encoder is very noisy and inaccurate (see Figure 6a), but the LPV Kalman filter can provide a quite clean estimation. Similarly happens with the angular velocity. The lateral velocity was not measured and is successfully estimated. The final corrected map after completing the laps is shown in Figure 7b. Figure 7a represents the environment which includes fixtures, obstacles, and free space. There are some unoccupied cells particularly when the LIDAR scan is hindered by the fixtures present in the environment. For this reason, the error between ground truth and map is only calculated for properly scanned area i.e. the fixture occlusion region is excluded (see Figure 7a). The Intersection-Over-

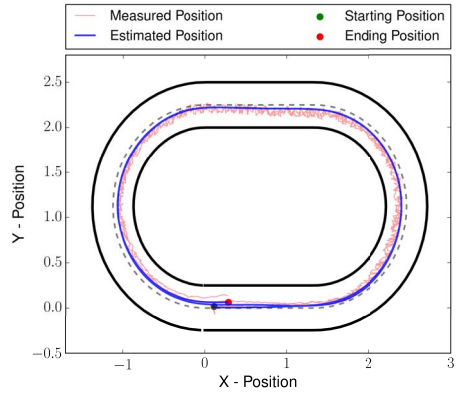
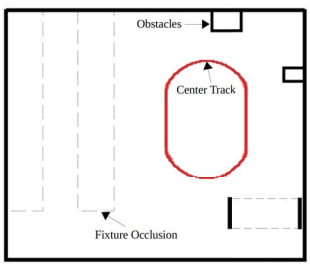


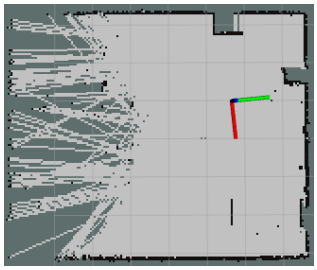
Fig. 4: Estimated trajectory in the presence of imprecise SLAM and sensor measurement during the real experiment.

Fig. 5: Snapshots of the vehicle following the track.

Union (IOU) score of 0.9388 is obtained between ground truth and corrected map.



(a) Environment Representation



(b) Obtained map

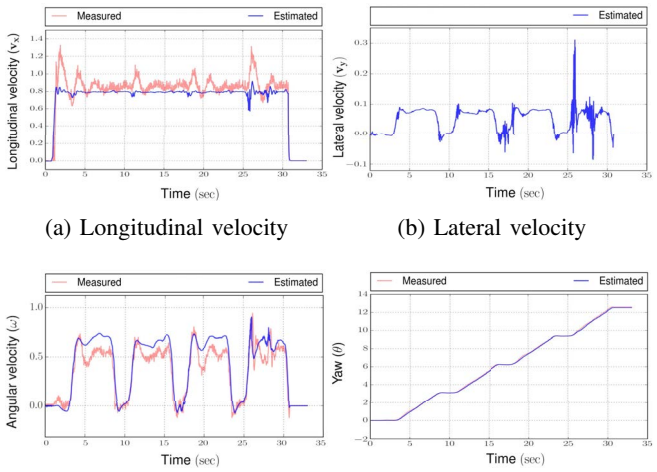
Fig. 7: The map formed during the trajectory following.

C. Simulated vs Real Experiment

It is worth noting that the time taken for completing two laps in the simulator and real experiments are 38 sec and 32 sec respectively. The real experiment completed the lap faster than the simulation due to strictly maintaining a longitudinal velocity of 0.8 m/s. There are discrepancies between some estimated states which are due to different control input in simulation and real experiment phase or mismatch of simulator model from the real vehicle model.

V. Conclusions

This paper has presented an approach for state estimation of an autonomous vehicle using a LPV Kalman Filter and imprecise SLAM. Further, this technique can also correct the



(a) Longitudinal velocity (b) Lateral velocity

(c) Angular velocity (d) Yaw

Fig. 6: Estimated states on real vehicle

map obtained from the noisy LIDAR scan end-points matching. The proposed approach provides a stable and optimal solution in presence of model disturbance and measurement noises at a high update rate of 100 Hz for state estimation and simultaneously correction of obtained map. The update rate can go much faster than 100Hz, the only limitation is the update rate of the sensor measurements. The computational cost of the proposed approach is fairly low if compared with EKF SLAM which depends on the number of landmarks. The result produced motivates the usage of the LPV based state estimation and mapping in the field of autonomous vehicle and robotics. The proposed estimation technique depends on the model of the system and the measurement model with certain disturbance and noise respectively. This requirement is sometimes hard to full fill, and also for some system the parameters changes within a certain range, for example, tire coefficient of vehicle. For such kind of system, online model learning techniques need to be incorporated [14, 15].

Acknowledgements

This work has been co-financed by the Spanish State Research Agency (AEI) and the European Regional Development Fund (ERFD) through the project SaCoAV (ref. MINECO PID2020-114244RB-I00), by the European Regional Development Fund of the European Union in the framework of the ERDF Operational Program of Catalonia 2014-2020 (ref. 001-P-001643 Looming Factory) and by the DGR of Generalitat de Catalunya (SAC group ref. 2017/SGR/482).

References

[1] R. E. Kalman. “A New Approach to Linear Filtering and Prediction Problems”. In: *Journal of Basic Engineering* 82.1 (Mar. 1960). ISSN: 0021-9223.

[2] D. Rotondo. “Advances in gain-scheduling and fault tolerant control techniques”. Springer, 2017.

[3] A. Kwiatkowski, M.-T. Boll, and H. Werner. “Automated Generation and Assessment of Affine LPV Models”. In: *Proceedings of the 45th IEEE Conference on Decision and Control*. 2006, pp. 6690–6695.

[4] R. Tóth. “Modeling and identification of linear parameter-varying systems”. Vol. 403. Springer, 2010.

[5] C. D. Pathirana. “Fuzzy Kalman Filtering of the Slam Problem Using Pseudo-Linear Models with Two-Sensor Data Association”. In: *International Journal of Systems Signal Control and Engineering Applications* (Jan. 2008).

[6] D. Rotondo, V. Reppa, V. Puig, and F. Nejjari. “Adaptive Observer for Switching Linear Parameter-Varying (LPV) Systems”. In: *IFAC Proceedings Volumes* 47.3 (2014). 19th IFAC World Congress, pp. 1471–1476. ISSN: 1474-6670.

[7] E. Alcalá, V. Puig, J. Quevedo, and U. Rosolia. “Autonomous racing using Linear Parameter Varying-Model Predictive Control (LPV-MPC)”. In: *Control Engineering Practice* 95 (2020), p. 104270. ISSN: 0967-0661.

[8] S. Kohlbrecher, J. Meyer, O. von Stryk, and U. Klingauf. “A Flexible and Scalable SLAM System with Full 3D Motion Estimation”. In: *Proc. IEEE International Symposium on Safety, Security and Rescue Robotics (SSRR)*. IEEE. Nov. 2011.

[9] J. Hartmann, J. H. Klüssendorff, and E. Maehle. “A comparison of feature descriptors for visual SLAM”. In: *2013 European Conference on Mobile Robots*. 2013, pp. 56–61.

[10] C. Chen, B. Wang, C. X. Lu, N. Trigoni, and A. Markham. “A Survey on Deep Learning for Localization and Mapping: Towards the Age of Spatial Machine Intelligence”. In: *CoRR* abs/2006.12567 (2020). arXiv: 2006.12567.

[11] R. Rajmani. “Vehicle dynamics and control.” In: vol. 2. 2012. ISBN: New York, Springer.

[12] H. B. Pacejka and E. Bakker. “The Magic Formula Tyre Model”. In: *Vehicle System Dynamics* 21.sup001 (1992), pp. 1–18. eprint: https://doi.org/10.1080/00423119208969994.

[13] E. Ostertag. “Mono-and multivariable control and estimation: linear, quadratic and LMI methods”. Springer Science & Business Media, 2011.

[14] D. Masti and A. Bemporad. “Learning nonlinear state–space models using autoencoders”. In: *Automatica* 129 (2021), p. 109666. ISSN: 0005-1098.

[15] H. S. A. Yu, D. Yao, C. Zimmer, M. Toussaint, and D. Nguyen-Tuong. “Active Learning in Gaussian Process State Space Model”. In: *CoRR* abs/2108.00819 (2021). arXiv: 2108.00819.