



HAL
open science

Coordinated PSO-PID based longitudinal control with LPV-MPC based lateral control for autonomous vehicles

Yassine Kebbati, Naïma Aït Oufroukh, Vincent Vigneron, Dalil Ichalal

► To cite this version:

Yassine Kebbati, Naïma Aït Oufroukh, Vincent Vigneron, Dalil Ichalal. Coordinated PSO-PID based longitudinal control with LPV-MPC based lateral control for autonomous vehicles. 20th European Control Conference (ECC 2022), Jul 2022, London, United Kingdom. pp.518–523, 10.23919/ECC55457.2022.9838192 . hal-03749480

HAL Id: hal-03749480

<https://univ-evry.hal.science/hal-03749480v1>

Submitted on 10 Aug 2022

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Coordinated PSO-PID based longitudinal control with LPV-MPC based lateral control for autonomous vehicles

Yassine Kebbati^{1*}, Naima Ait-Oufroukh¹, Vincent Vigneron¹ and Dalil Ichalal¹

Abstract—Autonomous driving is achieved by controlling the coupled nonlinear longitudinal and lateral vehicle dynamics. Longitudinal control greatly affects lateral dynamics and must preserve lateral stability conditions, while lateral controllers must take into account actuator limits and ride comfort. This work deals with the coordinated longitudinal and lateral control for autonomous driving. An improved particle swarm optimized PID (PSO-PID) is proposed to handle the task of speed tracking based on nonlinear longitudinal dynamics. An enhanced linear parameter varying model predictive controller (LPV-MPC) is also designed to control lateral dynamics, the latter is formulated with an adaptive LPV model in which the tire cornering stiffness coefficients are estimated by a recursive estimator. The proposed LPV-MPC is enhanced with an improved cost function to provide better performance and stability. Matlab/Carsim co-simulations are carried out to validate the proposed controllers.

Index Terms—Autonomous driving, Particle swarm optimization, Model predictive control, Adaptive control.

I. INTRODUCTION

Autonomous vehicle technology has recently been the subject of significant research and development, due to its numerous advantages compared to conventional vehicles. Among its advantages are a reduced accident and fatality rate, better and smoother traffic, energy and time efficiency. Automated driving systems consist of several modules that work together in a coordinated manner. First, there is a perception module made up of a variety of sensors, including cameras, LIDAR, RADAR, GPS and IMU. Several algorithms are used by this module to detect the environment and extract relevant information. Second, a planning and decision-making module (speed and path planners) that exploits the perceived information to make decisions and plan future actions such as changes in vehicle speed and direction. Then comes the control system that executes the decisions taken by the previous module. Automatic control is the final step and one of the most important tasks in autonomous driving systems. The latter is divided into longitudinal and lateral control, where longitudinal control ensures precise speed tracking and lateral control handles the steering task. Extensive research has been done to address both tasks and reasonable progress has been made in the field. For instance, paper [1] developed an MPC controller for low velocity tracking in advanced driver-assistance systems (ADAS); it has been compared to PID and neuro-fuzzy inference system (ANFIS) controllers. Simorgh *et al.* [2] worked on adaptive PID control for speed regulation systems; they used the inverse model theory to adapt the PID gains. Good speed

tracking has been obtained under adverse trajectory conditions and aerodynamic effects, but this approach depends on the reference model which is generally less accurate due to linearizations and simplified assumptions. Other works exploit the advantage of Artificial Intelligence to improve classical controllers such as Nie *et al.* [3] who used radial basis function to adapt the PID gains online (RBF-PID), which improved the tracking accuracy and driving comfort. In a similar way, Jin *et al.* [4] developed a fuzzy-PID for maintaining low speeds while driving downhill. Kebbati *et al.* [5] developed a self-adaptive PID controller for the speed tracking task. They used genetic algorithms and neural networks to tune and optimize the PID gains for better tracking and disturbance rejection.

Several papers worked also on lateral control. In [6], an adaptive MPC controller has been designed for automatic lane-keeping. The developed controller was able to handle steering offsets learned from measured data by using a membership function approach. Overall, the results showed significant improvement provided that the vehicle lateral dynamics were perfectly known. Paper [7] also addressed lane-keeping, where an LPV-MPC was developed for steering control, and a dual rate extended Kalman filter (DREKF) was used for state estimation. The simulation results demonstrated acceptable performance. Likewise, Yang *et al.* [8] addressed the longitudinal control by designing a sliding mode controller (SMC) with a conditional integrator. They coupled it with an LPV-MPC for lateral guidance and then with an active disturbance rejection controller (ADRC). SMC suffers from the chattering phenomenon. ADRC showed good robustness against parametric uncertainties, but LPV-MPC achieved much more accurate tracking performance. Nevertheless, most of the previously mentioned papers only consider constant longitudinal speeds, while variable speeds greatly affect lateral control performance. Study [9] developed an adaptive MPC for the path tracking task; they proposed an improved PSO algorithm to optimize the MPC parameters and used a lookup table approach to adapt these parameters online. Despite the optimal results for time-varying longitudinal velocities, this strategy cannot cover all possible situations and the lookup table method requires certain approximations which reduce the overall control precision. However, the same authors improved their controller design in [10] by using neural networks and adaptive neuro-fuzzy inference systems to learn the optimal MPC parameters and replace the lookup table approach. The idea was to adapt the MPC controller to varying working conditions and external disturbances. The obtained results showed significant tracking

*Corresponding author: yassine.kebbati@univ-evry.fr

¹IBISC-EA4526, univ Evry, universit  Paris-Saclay, France

improvements, but this approach still requires long offline optimizations.

However, research on coupled longitudinal and lateral control is less abundant. Attia *et al.* [11] developed a combined lateral and longitudinal control strategy for autonomous driving. Nonlinear model predictive control (NLMPC) was used to manage lateral guidance and longitudinal control was based on Lyapunov theory. The control strategy achieved good results, but NLMPC requires very expensive computations. Although nowadays it can be applied in real-time, it requires expensive hardware. Chebly *et al.* [12] also addressed coupled lateral and longitudinal control. A multi-body formalism was used to accurately model the four-wheeled vehicle, and the control laws were developed based on Lyapunov function, then on the immersion and invariance with sliding mode approach. Robust speed and trajectory tracking were achieved with both methods. The immersion and invariance based controller showed better performance than its rival since it is less model-dependent. However, the design of this type of controllers is a difficult task and its practical implementation is not guaranteed.

The contributions of this work are threefold; First, an improved LPV-MPC with a simple and precise prediction model is developed for handling the vehicle lateral dynamics. Second, recursive least squares algorithm (RLS) is used to estimate the cornering stiffness coefficients online and adapt the MPC prediction model iteratively. Third, an improved PSO algorithm is used to optimize a PID controller (PSO-PID) for handling the longitudinal dynamics which is coordinated with the lateral controller while ensuring lateral stability. This paper is organized as follows: Section II presents the modeling of the longitudinal and lateral dynamics of the vehicle and the parameters estimation with recursive least squares algorithm. The design of the coordinated longitudinal and lateral control is detailed in section III. Section IV presents and analyzes the obtained results. Finally, section V concludes the work and gives some perspectives.

II. VEHICLE MODELING

The vehicle dynamics are highly nonlinear, where the longitudinal and lateral dynamics are coupled. In this paper a nonlinear longitudinal model that accounts for nonlinear power-train and tire longitudinal dynamics is developed and coupled with a linear parameter varying bicycle model for the lateral dynamics.

A. Longitudinal Dynamics Modeling

The longitudinal dynamics of a vehicle consist of several subsystems that represent the chassis, the engine, the transmission, the wheels, the tires and the brakes. Vehicle body dynamics can be modeled by the following equation [5]:

$$F_t = m \frac{dv}{dt} + \frac{1}{2} \rho C_d A_f (v + v_w)^2 + mg C_r \cos \theta + mg \sin \theta \quad (1)$$

where F_t , m , v and v_w are the drive force, the vehicle mass, the vehicle velocity and the wind velocity respectively. A_f , C_d and C_r are the vehicle cross sectional area, the drag and rolling resistance coefficients respectively. g , ρ and θ

represent the gravity, air density and road elevation angle. The propulsion system of electric vehicles is composed of the motor and the transmission. The motor can be modeled statically using the motor efficiency map and the transmission is modeled as a single speed gearbox as in equations (2) and (3) respectively [5]:

$$\begin{cases} T_e &= T_{e\text{-ref}} \\ I_b &= \frac{T_e \omega_g}{u_b \eta^k} \end{cases} \quad (2)$$

$$\begin{cases} T_g &= k_g T_e \\ \omega_g &= k_g \omega_e \end{cases} \quad (3)$$

with $T_{e\text{-ref}}$, T_e and T_g being the engine reference torque (obtained from the efficiency map), the engine torque and the gear torque respectively. Similarly, ω_e and ω_g are the engine and the gear angular speeds. k_g , I_b and u_b represent the gear ratio, the current and the voltage drawn from the battery. η is the motor efficiency where the exponent $k = 1$ if $T_e \omega_g \geq 0$, otherwise $k = 0$. The wheels can be modeled by equations (4) under the assumption of negligible slip [5], and the tires are modeled by the magic formula given in (5):

$$\begin{cases} F &= \frac{T - B_d \omega_w + I_w \dot{\omega}_w}{R_w} \\ \omega_w &= \frac{1}{R_w} v \end{cases} \quad (4)$$

$$F_i = F_z D \sin(C \tan^{-1}[Bk - E[Bk - \tan^{-1}(Bk)]]) \quad (5)$$

where F and T are either the drive or the braking force and torque respectively, ω_w , R_w and v represent the wheel angular velocity, the wheel radius and the vehicle linear velocity. I_w and B_d are the wheel moment of inertia and damping coefficient. Equation (5) is the Pacejka tire model [13] where F_i can be either the longitudinal or the lateral tire force, F_z is the normal tire force, k is the slip angle and $\{B, C, D, E\}$ are experimental fitting coefficients. The torque of the brake system is governed by equation (6) where B_a and R_m are the actuator diameter and the brake pad mean radius. P and f_b are the brake pressure and the friction coefficient. These subsystems are modeled using the the powertrain-blockset of MATLAB.

$$T_b = \frac{f_b P \pi B_a^2 R_m}{2} \quad (6)$$

B. Lateral Dynamics Modeling

The vehicle lateral dynamics are modeled using an LPV version of the standard bicycle model where the motion accounts for translation on the y-axis and rotation around the z-axis. By applying Euler-Newton's formalism to the model in (Fig. 1), using small angle approximations and linearized tire model, the full model can be summarized in the following equations [9]:

$$\begin{cases} m(\dot{v}_y + v_x \dot{\psi}) &= 2[c_f(\delta - \frac{v_y + b\dot{\psi}}{v_x}) + c_r \frac{a\dot{\psi} - v_y}{v_x}] \\ I_z \ddot{\psi} &= 2[bc_f(\delta - \frac{v_y + b\dot{\psi}}{v_x}) - ac_r \frac{b\dot{\psi} - v_y}{v_x}] \end{cases} \quad (7)$$

where v_x/y , ψ and δ are the longitudinal/lateral velocities, the heading angle and the steering angle respectively. c_f/r ,

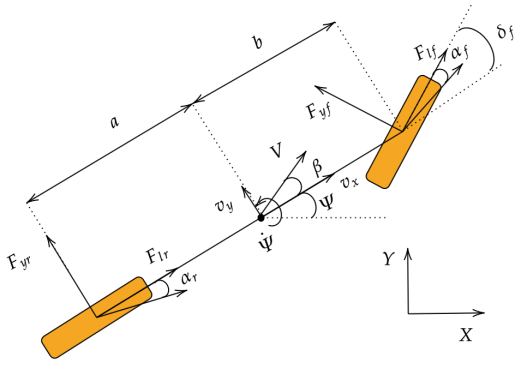


Fig. 1: Bicycle dynamic model

a and b represent the cornering stiffness coefficients of front/rear tires and the distances from rear and front wheels to the vehicle gravity center (GC), and I_z is the vehicle inertia moment. The corresponding affine LPV state space representation is given by:

$$\begin{cases} \dot{X} &= A_d(\rho)X + B_d(\rho)u \\ Y &= CX \end{cases} \quad (8)$$

where $X = [\dot{y} \ \psi \ \dot{\psi} \ y]^T$, $Y = [y \ \psi]^T$ and $u = \delta_f$ represent the state vector, the output vector and the control signal respectively. The system matrices A and B depend on an LPV scheduling vector $\rho(t) = [v_x \ \frac{c_f}{v_x} \ \frac{c_r}{v_x} \ c_f]^T$. All the parameters are bounded and the system matrices are affine with respect to the parameters of the scheduling vector, the matrices are defined by:

$$A = \begin{bmatrix} a_{11}(\rho) & 0 & a_{13}(\rho) & 0 \\ 0 & 0 & 0 & 1 \\ a_{31}(\rho) & 0 & a_{33}(\rho) & 0 \\ 1 & a_{42}(\rho) & 0 & 0 \end{bmatrix}, \quad B = \begin{bmatrix} b_{11}(\rho) \\ 0 \\ b_{13}(\rho) \\ 0 \end{bmatrix}, \quad C = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \end{bmatrix}.$$

where the terms of A and B are defined as follows:

$$\begin{aligned} a_{11}(\rho) &= -2 \frac{C_f + C_r}{m v_x}, \quad a_{13}(\rho) = -v_x - \frac{b C_f - a C_r}{m v_x}, \\ a_{31}(\rho) &= -2 \frac{b C_f - a C_r}{I_z v_x}, \quad a_{33}(\rho) = -2 \frac{C_f b^2 + C_r a^2}{I_z v_x}, \\ a_{42}(\rho) &= v_x, \quad b_{11}(\rho) = 2 \frac{C_f}{m}, \quad b_{13}(\rho) = 2 \frac{C_f b}{I_z}. \end{aligned}$$

Parameters c_f and c_r are obtained by using a recursive least squares (RLS) estimator. The estimator uses the linear tire cornering stiffness equations:

$$\begin{cases} F_f &= c_f \alpha_f \\ F_r &= c_r \alpha_r \end{cases} \quad (9)$$

The RLS algorithm given in equation (10) uses the above mentioned equations as a parameter identification form where $z(k)$ is the measured value, $\phi(k)$ is a regression vector, $\theta(k)$ represents the estimated parameters and $e(k)$ is the identification error. These are defined in equations (11):

$$z(k) = \phi^T(k)\theta(k) + e(k) \quad (10)$$

$$\begin{cases} z(k) &= [F_f \ F_r]^T \\ \phi^T(k) &= \begin{bmatrix} \alpha_f & 0 \\ 0 & \alpha_r \end{bmatrix} \\ \theta^T(k) &= [c_f \ c_r]^T \end{cases} \quad (11)$$

The values of $z(k)$ and $\phi(k)$ are obtained from the vehicle simulator (Carsim) and are fed to the RLS estimator, but in reality the lateral forces and side-slip angle are very hard to measure and they are usually estimated. The estimation process of these parameters is well studied in the literature [14], [15] and is not in the scope of this study. Table (I) summarizes the parameters used for both models.

TABLE I: Model parameters.

| | | | | | |
|--------|------------------------|-------|----------------------|--------|----------------------|
| m | 1575kg | C_d | 0.29 | a | 1.6m |
| C_r | 0.007 | k_g | 3.4 | b | 1.2m |
| A_f | 1.6m ² | R_w | 0.329m | I_z | 2875N.m |
| ρ | 1.222kg/m ³ | R_m | 0.1778m | I_w | 0.8kg.m ² |
| f_b | 0.9 | B_d | 0.001/s ² | η | 95% |

III. CONTROLLER DESIGN

The coupled lateral and longitudinal control strategy must preserve the lateral stability when performing lateral maneuvers. In this regard, both road geometry and lateral dynamics are to be considered [11]. The road information is used by the speed planner to estimate the maximum allowable longitudinal speed for the steering maneuver, the cruise speed has to be reduced in function of the curvature and must respect the following condition:

$$v_{\max} = \sqrt{\frac{g}{\rho_r} \frac{\phi_r + \mu}{1 - \phi_r \mu}} \quad (12)$$

with g , μ , ϕ_r and ρ_r being the gravity, the adhesion coefficient, the camber angle and the curvature respectively. This criterion is used as a predictive measure to guarantee admissible speeds before initiating lateral maneuvers. However, additional criteria related to lateral dynamics must be validated and in particular, the side slip angle (β) must obey criterion (13):

$$\beta \leq 10 - \frac{7v_x^2}{40} \quad (13)$$

The latter can be reformulated in terms of steering angle and implemented as a constraint on the control signal. According to [16], the side slip angle can be expressed by:

$$\beta = \arctan \left(\frac{b \tan \delta_r + a \tan \delta_f}{a + b} \right) \quad (14)$$

where the rear wheel steering angle can be set to zero as only the front wheels are steerable ($\delta_r = 0$). Hence, using equation (14), criterion (13) can be reformulated into :

$$|\delta_f| \leq \arctan \left(\frac{(a + b) \tan \left(10 - \frac{7v_x^2}{40} \right)}{a} \right) \quad (15)$$

A. PSO-PID Control

For the speed tracking task, an optimized PID controller is proposed. PID control is commonly used in the industry, it achieves good performance whilst being relatively simple to design compared to other methods. The general formula of a PID controller is provided in equation (16):

$$u(t) = K_p e(t) + K_i \int e(\tau) d\tau + K_d \frac{de(t)}{dt} \quad (16)$$

where $e(t)$ is the error between the reference and the output, and $\{K_p, K_i, K_d\}$ are gains for the proportional, integral and derivative actions, which are often hard to tune. Therefore, the three parameters are optimized through an improved particle swarm optimization algorithm (PSO), which is a well established stochastic optimization technique [17], [18]. The algorithm relies on the principle of swarm social behaviour, where a swarm contains multiple particles which represent potential solutions. Each particle has a position p_i and a velocity v_i , and the idea is to find the velocity that moves particles towards the global optimal position. The algorithm is governed by:

$$\begin{cases} v_i(k+1) = \omega v_i(k) + c_1 r_1 (Pb_i(k) - x_i(k)) \\ \quad + c_2 r_2 (Gb(k) - x_i(k)) \\ x_i(k+1) = x_i(k) + v_i(k+1) \end{cases} \quad (17)$$

where ω , c_1 and c_2 are known as inertia weight, cognitive and social accelerations respectively, and $r_{1,2} \in [0, 1]$ are random constants. Gb and Pb are the global best position of the whole swarm and the local best position in a single swarm generation. The inertia weight, social and cognitive accelerations are prefixed constants in the classic algorithm. However, in the improved version of this work they are dynamic according to equations (18,19) which improves the PSO search performance:

$$\omega = \omega_{\min} + \frac{\exp(\omega_{\max} - \lambda_1(\omega_{\max} + \omega_{\min}) \frac{g}{G})}{\lambda_2} \quad (18)$$

$$\begin{cases} c_1(k+1) = c_1(k) + \alpha \\ c_2(k+1) = c_2(k) + \beta \\ \alpha = -2\beta = 0.085 \quad \text{for } \frac{g}{G} \leq 30\% \\ \alpha = \frac{-\beta}{2} = 0.045 \quad \text{for } 30\% \leq \frac{g}{G} \leq 60\% \\ \alpha = \frac{-\beta}{2} = -0.025 \quad \text{for } 60\% \leq \frac{g}{G} \leq 85\% \\ \alpha = -\beta = -0.0025 \quad \text{for } \frac{g}{G} \geq 85\% \end{cases} \quad (19)$$

Parameters g and G are the actual and the last generations respectively. $\lambda_{1,2}$ are constants adjusted to ensure an exponential decrease from ω_{\max} to ω_{\min} which are maximum and minimum inertia weights. The advantage of this PSO improved version compared to other improved versions is that it enhances the overall search capabilities of the algorithm, the exponential decrease of ω accelerates the convergence towards the global best solution. On the other hand, increasing c_1 pulls the particles towards Pb and enhances the exploration phase, while increasing c_2 speeds the convergence towards Gb which enhances the exploitation phase and vice versa. The proposed PSO is used to optimize the PID gains $\{K_p, K_i, K_d\}$ using the longitudinal model

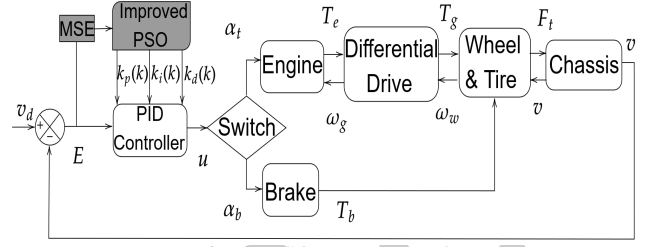


Fig. 2: PSO-PID approach.

discussed in section II. The approach is illustrated in (Fig. 2) where the mean squared error (MSE) is used as the fitness function of the PSO algorithm. The switch module in the figure decides whether the control signal is a braking or an acceleration command based on its sign, the switching logic sends the command either to the braking system or to the engine at a time [3].

B. LPV-MPC Control

The lateral control performs the position and orientation tracking, an LPV-MPC formulation is proposed for this task. Model (8) is discretized using sampling time T_s and used as the MPC prediction model, where $A_d = I + T_s A$ and $B_d = T_s B$. The principle relies on using the prediction model to predict the behaviour of the vehicle along a prediction horizon N_p , then solving the resulting optimization problem along a control horizon N_c . The goal is to find a control sequence that minimizes the error between the predicted vehicle behavior and the desired behaviour. At each iteration, a new adapted instance of the LPV model is used for the MPC predictions which improves the overall prediction accuracy. Moreover, constraints are imposed on the control signal, its rate and system states to account for actuator limits and ride comfort. In addition to optimality, the possibility to handle constraints is a major advantage of model predictive control. The standard optimization problem is given in equations (20-24):

$$\min J = (Y_r - Y_p)^T Q (Y_r - Y_p) + \Delta U^T R \Delta U \quad (20)$$

$$s.t : x(k+1) = A_d(\rho)x(k) + B_d(\rho)\Delta u(k) \quad (21)$$

$$\Delta u_{\min} \leq \Delta u(k) \leq \Delta u_{\max} \quad (22)$$

$$u_{\min} \leq u(k) \leq u_{\max} \quad (23)$$

$$x_{\min} \leq x(k) \leq x_{\max} \quad (24)$$

where ΔU , Y_p and Y_r represent the respective control sequence, the predicted and the reference trajectories, and Q and R are weighting matrices that penalize the tracking error and control effort. The constraints on the control and rate of control signal ensure a comfortable ride and impose actuator limits in addition to the lateral stability criterion given in equation (15). To improve the optimization, an enhanced cost function is proposed by adding a relaxation factor (ϵ) to avoid the infeasibility of strict constraints. An exponential weight (β^{-k}) is also included, the latter decreases with iteration k . This decreasing weight ensures that the first

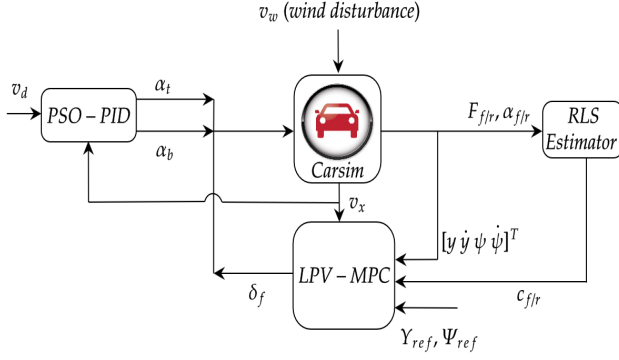


Fig. 3: Structure of the control approach

control input increments are more important in the sequence [19]. Therefore, immediate actions are favoured to penalize immediate tracking errors more than future tracking errors since MPC is based on the receding horizon principle. The cost function is then reformulated into equation (25):

$$J = \sum_{j=1}^{N_p} \beta^{-j} \|y_r(k+j|k) - y_p(k+j|k)\|_Q^2 + \sum_{j=1}^{N_c} \beta^{-j} \|\Delta u(k+j|k)\|_R^2 + \rho \epsilon^2 \quad (25)$$

The LPV-MPC is coded in MATLAB using Yalmip platform and solved using Gurobi optimization solver. The algorithm runs at 20Hz on an Asus Rog G17 with 2.6GHz I7 10750H and 32GB of DRAM. The coordinated control strategy is implemented and tested in Matlab/Carsim co-simulations as shown in Fig. 3, where α_t and α_b represent the throttle and brake signals, v_d , Y_{ref} and Ψ_{ref} are the desired longitudinal velocity, the lateral position and the heading trajectories. The desired velocity is generally obtained via a speed planner and must verify condition (12). The LPV-MPC and PSO parameters are tuned iteratively until the desired performance is obtained. These are given in table (II).

TABLE II: Model parameters.

| N_p | 9 | Q | $diag(35 \ 3.25)$ | R | 1.25 |
|----------------|-----------------|------------------|-------------------|------------------|-------------------|
| ρ | 15 | ϵ | 0.5 | β | 3.5 |
| T_s | 0.1s | Δu_{max} | $\frac{p^i}{12}$ | Δu_{min} | $-\frac{p^i}{12}$ |
| u_{max} | $\frac{p^i}{6}$ | u_{min} | $-\frac{p^i}{6}$ | G | 25 |
| ω_{max} | 1 | ω_{min} | 0.1 | c_{i1} | 2.2 |
| λ_1 | 3 | λ_2 | 30 | c_{i2} | 2.2 |

IV. RESULTS AND DISCUSSION

The proposed control strategy is evaluated for a double lane-change maneuver where the velocity varies between 50 and 65 km/h, and the vehicle is exposed to lateral and longitudinal wind disturbances. The test road is flat asphalt with an adhesion coefficient $\mu = 0.95$. For this test, the PSO optimization generated the PID triplet $\{K_p = 5.52, K_i = 0.0547, K_d = 0.469\}$. The LPV-MPC with regular cost function (20) is compared to the one with the enhanced cost function (25), they are denoted 'MPC' and 'E-MPC' respectively. Fig. 4 shows the wind speed (with variable direction), the speed tracking performance and control signals. The results show smooth speed tracking with adequate

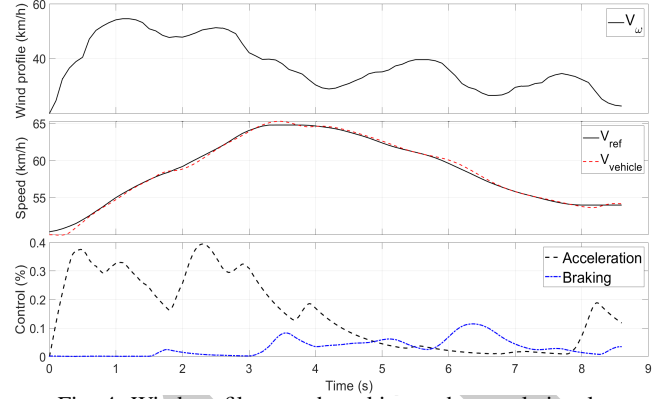


Fig. 4: Wind profile, speed tracking and control signals.

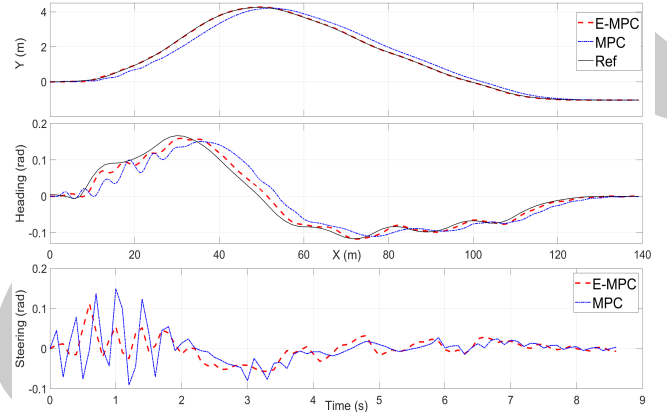


Fig. 5: LPV-MPC tracking performance and steering control.

accuracy, with an $MSE=0.0213$. Fig. 5 shows the lateral tracking performance, the advantage of *E-MPC* is obvious in both position and yaw tracking with smoother steering which results in a more comfortable ride. The accuracy is measured by the MSE value for position and orientation tracking, where *E-MPC* scored $2.118e^{-4}$ against 0.601 for *MPC* with standard cost function. For the orientation tracking, *E-MPC* scored $1.228e^{-4}$ against $6.756e^{-4}$. The highest position tracking error of *E-MPC* does not exceed 5cm.

Furthermore, the effectiveness of *E-MPC* is verified for a general trajectory (see Fig. 6) with different bends and a varying velocity profile. The same road characteristics were maintained and external wind disturbances were also applied. The PSO optimization in this test produced the PID triplet $\{K_p = 1.884, K_i = 0.048, K_d = 0.325\}$. The wind profile, speed tracking performance and control signals are respectively shown in Fig. 7, where the observed performance is similar to the previous test, the tracking accuracy is indeed acceptable ($MSE=0.0187$) and the controls are smooth. The respective lateral tracking performance along with the corresponding steering control are shown in Fig. 8. Despite the challenging trajectory and speed profile with imposed wind disturbances, the proposed LPV-MPC showed good performance overall and was able to track both position and orientation with a resulting MSE of 0.32 and $6.56e^{-4}$ for position and orientation tracking respectively.

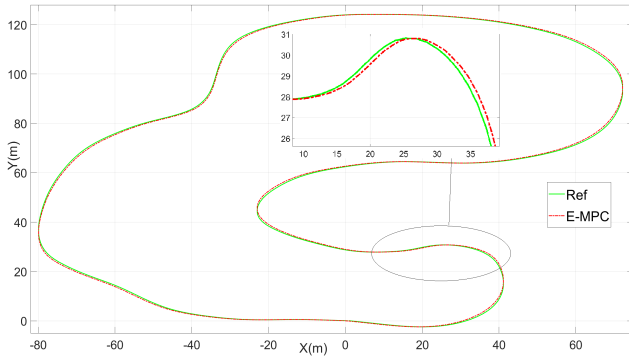


Fig. 6: Trajectory tracking.

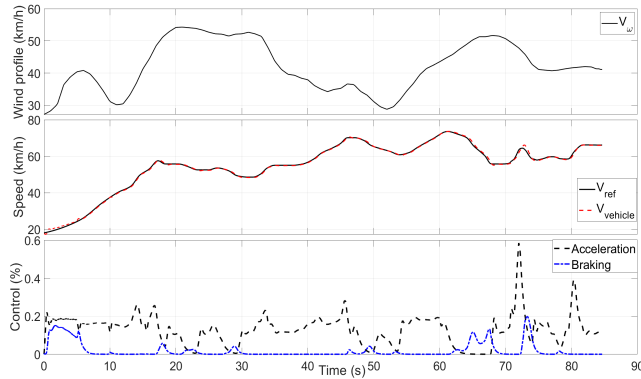


Fig. 7: Wind profile, Speed tracking and control signals.

V. CONCLUSIONS

This paper addressed the coordinated longitudinal and lateral control for autonomous driving. A nonlinear model is used to account for power-train and vehicle longitudinal dynamics, and an optimized PSO-PID is developed to ensure accurate speed tracking, where the PID gains were optimized by a new improved PSO algorithm. An LPV-MPC with an enhanced cost function is designed for the lateral control task, the proposed controller takes into account the varying longitudinal velocity and cornering stiffness coefficients. A recursive least squares estimator is used to estimate the tire cornering stiffness coefficients. The proposed controllers were validated in Matlab/Carsim co-simulations for a double lane-change scenario and a general trajectory tracking problem. The results showed satisfactory performance for both lateral and longitudinal tracking while being relatively robust against wind disturbances. Future research shall address the development of an enhance LPV-MPC that handles both lateral and longitudinal control simultaneously.

REFERENCES

- [1] M. Marcano, J. A. Matute, R. Lattarulo, E. Martí, and J. Pérez, "Low Speed Longitudinal Control Algorithms for Automated Vehicles in Simulation and Real Platforms," *Complexity*, vol. 2018, 2018.
- [2] A. Simorgh, A. Marashian, and A. Razminia, "Adaptive PID Control Design for Longitudinal Velocity Control of Autonomous Vehicles," *Proc. - 2019 6th Int. Conf. Control. Instrum. Autom. ICCIA 2019*, pp. 1–6, 2019.

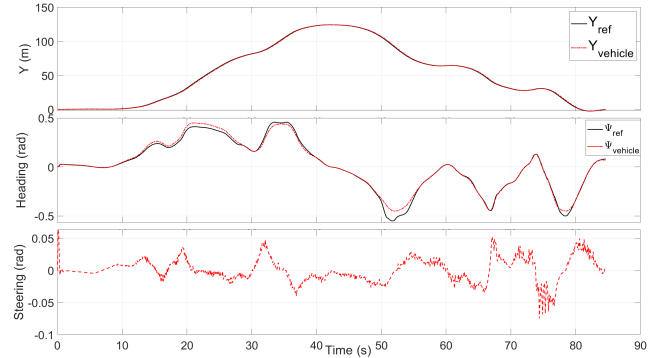


Fig. 8: Tracking performance and Steering control.

- [3] L. Nie, J. Guan, C. Lu, H. Zheng, and Z. Yin, "Longitudinal speed control of autonomous vehicle based on a self-adaptive PID of radial basis function neural network," *IET Intell. Transp. Syst.*, vol. 12, no. 6, pp. 485–494, 2018.
- [4] L. Jin, R. Zhang, B. Tang, and H. Guo, "A Fuzzy-PID scheme for low speed control of a vehicle while going on a downhill road," *Energies*, vol. 13, no. 11, pp. 1–18, 2020.
- [5] Y. Kebbati, N. Ait-Oufroukh, V. Vigneron, D. Ichalal and D. Gruyer, "Optimized self-adaptive PID speed control for autonomous vehicles," *2021 26th International Conference on Automation and Computing (ICAC)*, 2021, pp. 1-6.
- [6] M. Bujarbaruah, X. Zhang, H. E. Tseng, and F. Borrelli, "Adaptive MPC for Autonomous Lane Keeping," 2018, [Online]. Available: <http://arxiv.org/abs/1806.04335>.
- [7] J. M. S. Ducajú, J. J. S. Llobregat, Á. Cuenca, and M. Tomizuka, "Autonomous ground vehicle lane-keeping lpv model-based control: Dual-rate state estimation and comparison of different real-time control strategies," *Sensors*, vol. 21, no. 4, pp. 1–17, 2021.
- [8] X. Yang, L. Xiong, B. Leng, D. Zeng, and G. Zhuo, "Design, validation and comparison of path following controllers for autonomous vehicles," *Sensors (Switzerland)*, vol. 20, no. 21, pp. 1–24, 2020.
- [9] Y. Kebbati, V. Puig, N. Ait-Oufroukh, V. Vigneron and D. Ichalal, "Optimized adaptive MPC for lateral control of autonomous vehicles," *2021 9th International Conference on Control, Mechatronics and Automation (ICCMA)*, 2021, pp. 95-103.
- [10] Y. Kebbati, N. Ait-Oufroukh, V. Vigneron and D. Ichalal, "Neural Network and ANFIS based auto-adaptive MPC for path tracking in autonomous vehicles," *2021 IEEE International Conference on Networking, Sensing and Control (ICNSC)*, 2021, pp. 1-6.
- [11] R. Attia, R. Orjuela, and M. Basset, "Combined longitudinal and lateral control for automated vehicle guidance," *Veh. Syst. Dyn.*, vol. 52, no. 2, pp. 261–279, 2014.
- [12] A. Chebly, R. Talj, and A. Charara, "Coupled longitudinal/lateral controllers for autonomous vehicles navigation, with experimental validation," *Control Eng. Pract.*, vol. 88, no. May, pp. 79–96, 2019.
- [13] H. B. Pacejka, (2008). *Vehicle System Dynamics : International Journal of Vehicle Mechanics and Mobility*. International Journal of Vehicle Mechanics and Mobility, (August 2012), 37–41.
- [14] Y. F. Lian, Y. Zhao, L. L. Hu, and Y. T. Tian (2015). Cornering stiffness and sideslip angle estimation based on simplified lateral dynamic models for four-in-wheel-motor-driven electric vehicles with lateral tire force information. *Int.J Automot. Technol.* 16, 669–683
- [15] G. Baffet, A. Charara, and D. Lechner, (2009). Estimation of vehicle sideslip, tire force and wheel cornering stiffness. *Control Engineering Practice*, 17(11), 1255–1264.
- [16] R. Rajamani, *Vehicle dynamics and control*. Springer Science and Business Media, 2011
- [17] B. Song, Z. Wang, and L. Zou, "An improved PSO algorithm for smooth path planning of mobile robots using continuous high-degree Bezier curve," *Appl. Soft Comput.*, vol. 100, p. 106960, 2021
- [18] Y. Xie and W. Zhang, "A novel tuning method for PID controller based on improved PSO algorithm for unstable plants with time delay," *Chinese Control Conf. CCC*, vol. 2019-July, pp. 4270–4275, 2019
- [19] B. Zhang, C. Zong, G. Chen, and B. Zhang, "Electrical Vehicle Path Tracking Based Model Predictive Control with a Laguerre Function and Exponential Weight," *IEEE Access*, vol. 7, pp. 17082–17097, 2019