2024 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM)
July 15-19, 2024. Boston, MA, USA

Combined Longitudinal-Lateral Dynamic Modeling and Control via an Integrated Physics-Data-Based Approach

Wenpeng Wei[1,2], Zhaoyu Qiu[1], Xiaoyuan Zhu[1], Guodong Yin[1], Tianyi He[3,*]

Abstract— **This paper presents an Integrated Physics-Data-Based (IPDB) modeling and control scheme of the combined longitudinal-lateral vehicle dynamics. A nonlinear bicycle vehicle model is used to derive the linear parameter-varying (LPV) system representation, where four vehicle motion variables are considered as scheduling parameters. Taking advantage of kernels from LPV representation, the combined longitudinal-lateral dynamics are further expressed by the data snapshots of states, inputs, and scheduling parameters, which formulate the IPDB model. After that, the IPDB model is used to design a state-feedback gain-scheduling tracking controller to follow a reference trajectory. For validation purposes, the proposed modeling and control method is implemented on a QCar experimental platform. First, the IPDB model of coupled longitudinal-lateral dynamics is derived from experimental data and is further validated with excellent model accuracy under various driving conditions. Furthermore, an IPDB model-based gain-scheduling controller is synthesized and compared with the baseline Stanley controller in the experiment to track a given trajectory. The experimental results demonstrate that the IPDB model-based controller renders better tracking control performance.**

I. Introduction

The modeling and control of combined longitudinal-lateral vehicle dynamics play an important role in autonomous vehicles, which is essential for achieving optimal vehicle performance, safety, and maneuvering capability. While longitudinal and lateral dynamics are traditionally treated separately [1], [2], the coupling nature of vehicle motions [3] necessitates a holistic understanding and control strategy that addresses both motions simultaneously.

The challenges in modeling and controlling combined longitudinal and lateral dynamics arise from the complex interactions between front/rear tires and roads, and the coupling of nonlinear longitudinal-lateral dynamics under various driving conditions [4]. The traditional model-based approach relies on accurate models to design controllers. Longitudinal velocity, lateral velocity, and yaw rate are usually considered as states in the dynamic model, with considerations of longitudinal and lateral tire forces at the front and rear tires. Continued efforts have been witnessed to obtain models with higher accuracy, which often result in more and more complex model representations, parameter estimation algorithms, and control laws [5].

With the coupled dynamic model, various control methods have been reported to achieve accurate and robust tracking of given reference trajectories. Two loosely coupled low-complexity model-based controls were used for longitudinal and lateral motion in the yielding maneuvers [6]. A dissipative feedback control law was implemented based on the coupled model and was validated in test-bed experiments [7]. Model predictive control (MPC) has been embraced by the coupled longitudinal-lateral model, tire models, and powertrain models [8], [9], [10]. In particular, an envelope-MPC with dynamic objective and safety requirements was proposed and tested to yield improved stability under extreme conditions [11]. It is also demonstrated in the literature that the coupled longitudinal-lateral control can improve the vehicle-following performance in the platoon control than the decoupled control strategy [12].

Nonetheless, the complex nonlinear model often comes with heavy computations in nonlinear model predictive control or complex nonlinear control law. Besides, some of the vehicle parameters, especially the tire frictions, are difficult to estimate or identify, which makes it difficult to obtain accurate models. These limitations bring challenges to the practical implementations of complex nonlinear model-based control in coupled longitudinal-lateral vehicle dynamics.

Data-driven modeling and control provide another alternative to address the coupled longitudinal-lateral dynamics. A simple linear data-driven model from experimental data was found to outperform both linear and nonlinear physical models under real-world driving conditions [13], which sheds light on using such models in real-time control. A neural network (NN) model-based adaptive control was proposed for coupled longitudinal-lateral control in [14]. The controller was synthesized using a proportional and derivative control coupled with an online adaptive neural module to compensate for model mismatch, strong nonlinearities, and coupling effects. The closed-loop stability of the combined control scheme was analyzed using a Lyapunov-based method. Adaptive dynamic programming was proposed in [15] for the data-driven optimal control of coupled longitudinal-lateral vehicle dynamics. However, many of these data-driven methods have only been demonstrated in the simulation scenarios and are limited to certain driving conditions.

In view of these issues, this paper extends the physics-data-based (IPDB) approach in [16], [17] to modeling the

[1]Wenpeng Wei, Zhaoyu Qiu, Xiaoyuan Zhu and Guodong Yin are with School of Mechanical Engineering, Southeast University, Nanjing, 211189, China (e-mails: weiwenpeng@seu.edu.cn, qiuzhaoyu@seu.edu.cn, zhuxy@seu.edu.cn, and ygd@seu.edu.cn).

[2]Wenpeng Wei is also with Key Laboratory of Measurement and Control of Complex Systems of Engineering, Ministry of Education, Southeast University, Nanjing, 210096, China.

[3]Tianyi He is with the Department of Mechanical and Aerospace Engineering, Utah State University, Logan, UT, 84322, USA (e-mail:tianyi.he@usu.edu).

[*] corresponding author.

coupled longitudinal-lateral dynamics and further designs IPDB controller to follow reference trajectories. The novelties of this paper are three-fold: -

1) The IPDB approach has been well demonstrated in the earlier works on lateral dynamics, while in this paper, the IPDB approach is extended to the combined longitudinal and lateral vehicle dynamics, which is more practical in real driving scenarios.
2) The IPDB model is derived from the nonlinear dynamic equation, linear time-varying expressions of tire forces, and embedded into the linear parameter-varying (LPV) representation. Four variables are selected as the scheduling parameters, which produce kernels between states, inputs, and scheduling parameters. The vehicle-related parameters are kept in the system matrices, separated from the measurable vehicle motion variables, and are further expressed by the experimental data.
3) A gain-scheduling path-following controller is synthesized from the IPDB model and is validated in the experimental platform. The proposed IPDB control is compared with the traditional Stanley controller and renders better tracking performance.

The remainder of this paper is organized as follows. Section II presents the details of the IPDB modeling for combined longitudinal and lateral vehicle dynamics. Section III designs an LPV tracking controller using error dynamics based on the proposed IPDB model. In Section IV, the experiment equipment is introduced and the proposed modeling approach and controller are validated. Finally, the conclusions are drawn, and future work is discussed.

II. IPDB Modeling of Integrated Longitudinal-Lateral Dynamics

A. Combined Longitudinal and Lateral Dynamics

The combined longitudinal and lateral vehicle dynamics can be obtained using Newton-Euler law as follows.

$$\begin{aligned} m(\dot{v}_x - v_y\gamma) &= F_f^x cos\delta_f + F_r^x - F_f^y sin\delta_f \\ m(\dot{v}_y + v_x\gamma) &= F_f^x sin\delta_f + F_f^y cos\delta_f + F_r^y \\ I_z\dot{\gamma} &= (F_f^x sin\delta_f + F_f^y cos\delta_f)L_f - F_r^y L_r \end{aligned} \tag{1}$$

where v_x (m/s) denotes longitudinal velocity; v_y (m/s) represents lateral velocity; γ (rad/s) is yaw rate; m (kg) is vehicle mass; δ_f $(^\circ)$ is the steering angle at the front wheel; I $(kg\cdot m^2)$ is yaw moment of inertia; L_f (m) is distance from center of gravity to front axle; L_r (m) is distance from center of gravity to rear axle; F_i^x (N) is longitudinal forces with i denotes 'f' or 'r' representing front or rear tire; and F_i^y (N) is lateral forces.

The longitudinal and lateral forces are modeled to be linearly related to slip ratio or slip angle with time-varying longitudinal or lateral tire stiffness coefficients, as follows.

$$\begin{aligned} F_i^x &\approx \mathscr{C}_i^x(t)\lambda_i, \quad \lambda_i = \tfrac{r_i w_i - v_x}{r_i w_i} \\ F_f^y &\approx \mathscr{C}_f^y(t)\alpha_f, \quad \alpha_f = \delta_f - \tfrac{v_y + L_f\gamma}{v_x} \\ F_r^y &\approx \mathscr{C}_r^y(t)\alpha_r, \quad \alpha_r = \tfrac{L_r\gamma - v_y}{v_x} \end{aligned} \tag{2}$$

where $\mathscr{C}_i^x(t)$ (N) and $\mathscr{C}_f^y(t)$ (N/rad) are time-varying longitudinal and lateral stiffness coefficients, which are nonlinear functions of factors of tire-road interactions; λ_i is wheel slip ratio; w_i (rad/s) is the tire rotational speed; r_i (m) is the effective tire radius; and α_i is the tire slip angle.

The vehicle dynamics (1) follow the general representation of the LPV system as follows.

$$\dot{x} = A(\rho)x + B(\rho)u \tag{3}$$

where $x = \begin{bmatrix} v_x & v_y & \gamma \end{bmatrix}^T \in \mathscr{R}^{n_x}$ denotes system state vector; control input $u^T = [u_1, u_2]$, and $u_1 = \delta_f$ is steering angle, $u_2 = 1$ is an extra '*virtual*' input resulted from the LPV formulation; $\rho \in \mathscr{R}^{n_\rho}$ is the scheduling parameter vector, $\rho = \begin{bmatrix} \gamma & \frac{1}{v_x} & \lambda_f & \delta_f \end{bmatrix}^T$; $A(\rho), B(\rho)$ are dependent on the vehicle-related parameters.

The LPV formulation in this work is innovative from the existing LPV models of longitudinal/lateral dynamics [18] in the following aspects: 1) The vehicle-related parameters $(C_f^x, C_r^x, C_f^y, C_r^y, L_f, L_r, m, I_z)$ are separated from the motion-related variables $(v_x, v_y, \gamma, \delta_f, \lambda_r, \lambda_f)$. The vehicle-related model parameters depend on the tire, road, and vehicle conditions, which are needed in the model but are often hard to estimate or identify. The motion-related variables, however, are measurable by on-board sensors. 2) We keep the vehicle-related parameters in the model matrices and encode the dependency of motion-related variables into the LPV representation. 3) Four measurable motion-related variables are chosen as the scheduling parameters, which renders an affine-dependent LPV model.

By discretizing the system with sampling time T_s, the following discrete-time LPV model can be obtained, where ρ_k is the scheduling parameter vector at time instant k.

$$x_{k+1} = A_k(\rho_k)x_k + B_k(\rho_k)u_k \tag{4}$$

B. IPDB Model Representation

The vehicle-related parameters, such as vehicle mass, yaw moment of inertia, tire stiffnesses, etc., in $A(\rho), B(\rho)$ are subject to change under varying operating conditions, and it is usually challenging to accurately identify these parameters. To overcome these difficulties, this paper adopts the IPDB modeling approach that models the system dynamics with data while preserving the explainable property of physical laws. For more details regarding the IPDB modeling, readers are referred to our previous work [17], [16].

Remember that $A_k(\rho_k)$ and $B_k(\rho_k)$ are affine dependent on the scheduling vector ρ_k, of which the element is ρ_{ik}. Using the Kronecker product, the nonlinear LPV system can be rewritten into (5),

$$\begin{aligned} x_{k+1} &= (A_{0k} + \sum_{i=1}^{4} A_{ik}\rho_{ik})x_k + (B_{0k} + \sum_{i=1}^{4} B_{ik}\rho_{ik})u_k \\ &= \mathscr{A}_k \begin{bmatrix} x_k \\ \rho_k \otimes x_k \end{bmatrix} + \mathscr{B}_k \begin{bmatrix} u_k \\ \rho_k \otimes u_k \end{bmatrix} \end{aligned} \tag{5}$$

where $\mathscr{A}_k$ is the lumped transition matrix $\mathscr{A}_k = [A_{0k}, A_{1k}, A_{2k}, A_{3k}, A_{4k}]$ and $\mathscr{B}_k$ is the lumped input matrix $\mathscr{B}_k = [B_{0k}, B_{1k}, B_{2k}, B_{3k}, B_{4k}]$. The lumped states have additional term $\rho_k \otimes x_k$, which is the Kronecker product kernel between states and scheduling parameter, and lumped inputs have additional term $\rho_k \otimes u_k$, the kernel between inputs and scheduling parameter. These two kernels encode the nonlinearity stemming from the physical laws and LPV formulation.

Assuming that the signal traces with length n_d can be recorded from vehicle onboard sensors, the following matrices can be defined to facilitate the IPDB modeling.

$$
\begin{aligned}
\mathbf{U}_k &= \begin{bmatrix} u^d_{k-n_d} & u^d_{k-n_d+1} & \cdots & u^d_{k-1} \end{bmatrix} \in \mathscr{R}^{n_u \times n_d} \\
\mathbf{X}_k &= \begin{bmatrix} x^d_{k-n_d} & x^d_{k-n_d+1} & \cdots & x^d_{k-1} \end{bmatrix} \in \mathscr{R}^{n_x \times n_d} \\
\mathbf{X}^+_k &= \begin{bmatrix} x^d_{k-n_d+1} & x^d_{k-n_d+2} & \cdots & x^d_{k} \end{bmatrix} \in \mathscr{R}^{n_x \times n_d} \\
\mathbf{U}(\rho_k) &= \begin{bmatrix} \rho^d_{k-n_d} \otimes u^d_{k-n_d} & \cdots & \rho^d_{k-1} \otimes u^d_{k-1} \end{bmatrix} \in \mathscr{R}^{n_\rho n_u \times n_d} \\
\mathbf{X}(\rho_k) &= \begin{bmatrix} \rho^d_{k-n_d} \otimes x^d_{k-n_d} & \cdots & \rho^d_{k-1} \otimes x^d_{k-1} \end{bmatrix} \in \mathscr{R}^{n_\rho n_x \times n_d}
\end{aligned} \tag{6}
$$

Using the linear property with respect to inputs and states in (5), the system matrices $\begin{bmatrix} \mathscr{B}_k & \mathscr{A}_k \end{bmatrix}$ can be extracted from the data snapshots

$$
\begin{bmatrix} \mathscr{B}_k & \mathscr{A}_k \end{bmatrix} = \mathbf{X}^+_k \begin{bmatrix} \mathbf{U}_k \\ \mathbf{U}(\rho_k) \\ \mathbf{X}_k \\ \mathbf{X}(\rho_k) \end{bmatrix}^{\dagger} \tag{7}
$$

As a result, the system evolution can be represented by the data snapshots in (8), which is called the *integrated physics-data-based model*, i.e., the IPDB model.

$$
x_{k+1} = \mathbf{X}^+_k \underbrace{\begin{bmatrix} \mathbf{U}_k \\ \mathbf{U}(\rho_k) \\ \mathbf{X}_k \\ \mathbf{X}(\rho_k) \end{bmatrix}^{\dagger}}_{\Omega_k} \begin{bmatrix} u_k \\ \rho_k \otimes u_k \\ x_k \\ \rho_k \otimes x_k \end{bmatrix} \tag{8}
$$

According to different data snapshot selection strategies, different variants of IPDB models can be obtained. In the sequel, the fixed-window IPDB model is used, which is obtained by selecting data snapshots in a fixed window.

III. Gain-Scheduling Controller Design

In this section, a gain-scheduling LPV controller is designed based on the IPDB model for the combined longitudinal-lateral vehicle dynamics. The tracking error dynamics is first formulated, and the controller is synthesized via the LMI (Linear Matrix Inequality) for H_∞ performance.

A. Tracking Control Scheme

The overall control scheme is shown in Fig.1. The path planner generates a path in x, y directions for the vehicle to follow, and the kinematic controller generates the reference trajectory of the velocity and yaw rate. In this work, the pure pursuit kinematic controller is used. With the IPDB model, the gain-scheduling IPDB controller is designed to steer the vehicle to track the reference trajectory. The vehicle status information is feedback to the dynamic tracking controller with states v_x, v_y, γ and steering angle δ_f, and the vehicle location information x, y, θ are feedback to the kinematic controller to follow the desired path.

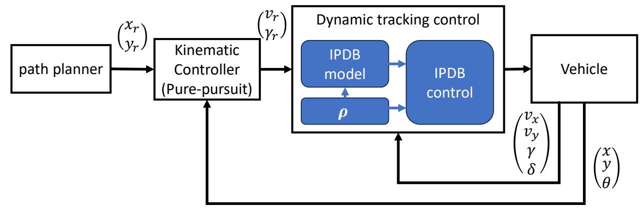

Fig. 1. Scheme of combined longitudinal-lateral vehicle control.

The tracking error dynamics need to be derived to design the tracking controller. Denote $e_k = x_k - r_k$ as the tracking error between states and the references states, $\Delta u_k = u_k - u_{r,k}$ as the error between control inputs and reference input, the tracking error dynamics along the reference trajectory (r_k, ρ_k) can be derived by the first-order approximation as

$$
\begin{aligned}
e_{k+1} &= A_k(\rho_k) e_k + B_k(\rho_k) \Delta u_k + E w_k \\
z_k &= C e_k + D \Delta u_k
\end{aligned} \tag{9}
$$

where w_k denotes the model mismatch or the regression residual in (7) due to data noises, and z_k as the weighted performance output that consists of state tracking error and control inputs.

In addition, $C = [Q^{1/2} I; 0]$ and $D = [0; R^{1/2} I]$ that render $z_k^T z_k = e_k^T Q e_k + \Delta u_k^T R \Delta u_k$ with weighting matrices Q, R. The weighting matrices can be tuned to achieve the desired responses of tracking error and control efforts.

B. Controller Synthesis Conditions

With the IPDB model (9), a gain-scheduling state-feedback controller $\Delta u_k = K(\rho_k) e_k$ is to be found that can stabilize the system and achieve robust performance. The following LMI can be used to synthesize the controller matrix, where $*$ makes the symmetric matrix. Note that, the state-feedback LPV controller synthesis condition has been well studied, and the following LMI is stemmed from the classic work [19]. Hence, the proof is omitted here.

Theorem 1: Given a positive scalar γ_∞, if there exist parameter-dependent matrices $P(\rho_k) > 0$, $F(\rho_k)$ such that the following LMI holds for $\rho_k, \rho_{k+1} \in \mathscr{P}$,

$$
\begin{bmatrix} P(\rho_{k+1}) & \Xi(\rho_k) & E & 0 \\ * & P(\rho_k) & 0 & P(\rho_k) C^T - F(\rho_k)^T D^T \\ * & * & -\gamma_\infty I & 0 \\ * & * & * & -\gamma_\infty I \end{bmatrix} > 0, \tag{10}
$$

where $\Xi(\rho_k) = A_k(\rho_k) P(\rho_k) - B_k(\rho_k) F(\rho_k)$, then the controller $K(\rho_k) = F(\rho_k) P^{-1}(\rho_k)$ can stabilize the system (9) and achieve $\|z_k\|_2 \le \gamma_\infty \|w_k\|_2$.

Remark 1: Note that, the given LMI now is not computationally solvable because the LMI is infinite-dimensional since $\rho_k, \rho_{k+1} \in \mathscr{P}$. S-procedure can be used to address the

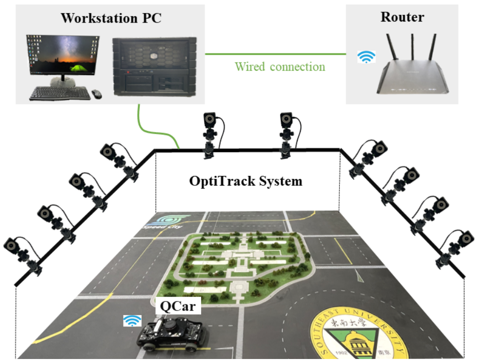

Fig. 2. The experimental setup of the Qcar platform.

parameter-dependent LMI [20]. Alternatively, one can fix the structures of the variables and test the resulting inequalities using the relaxation technique of homogeneous polynomially parameter-dependent solutions [21]. The Robust LMI Parser (ROLMIP) [22] was used to parse the LMIs and solver SeDuMi [23] was used to solve the semi-definite program. One can choose the polynomial degrees of the variables from the trade-off relationship that higher degrees usually lead to less conservative results at the price of heavier computations. Due to the affine dependency of the IPDB model matrices, it is natural that the gain-scheduling controller variables $P(\rho_k), F(\rho_k)$ are selected as affine-dependent as well, i.e. $P(\rho_k) = P_{0k} + \sum_{i=1}^{n} P_{ik}\rho_{ik}$, $F(\rho_k) = F_{0k} + \sum_{i=1}^{n} F_{ik}\rho_{ik}$. If these controller variables are selected as constant matrices, the LMI can be directly solved, leading to a conservative design of a fixed-gain controller K at the sacrifice of control performance.

IV. Experimental Validation and Discussion

A. Experimental Setup

The experiment tests are conducted on the Quanser self-driving car (QCar) platform, which consists of the OptiTrack motion capture system, the delicately designed track, the QCar vehicle, communication devices, and a workstation PC. The configuration of the platform is illustrated in Fig. 2. The OptiTrack motion capture system is equipped with 10 cameras and employed to obtain real-time pose information of the QCar, the QCar velocities/angular velocities are calculated from the pose information. The collected positional data are transmitted to the workstation PC through cables. The control algorithm is first developed in the MATLAB/Simulink environment on the workstation PC, then compiled into C code, and finally executed on an embedded Linux-based system powered by the onboard NVIDIA Jetson TX2 processor with a quad-core ARM Cortex-A57 microcontroller. During the tests, the onboard NVIDIA processor receives the command from the workstation PC and generates appropriate Pulse Width Modulation (PWM) signals. These PWM signals are sent to the drive and steering servo motors to control the motion of the QCar. The proposed IPDB modeling does not require any vehicle parameters, therefore, the vehicle configuration is omitted.

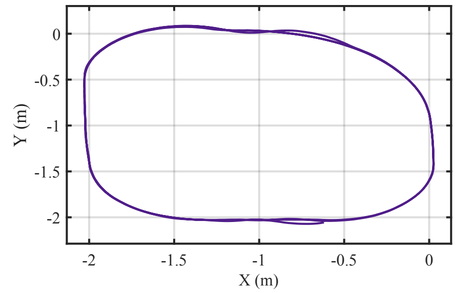

Fig. 3. The path of Qcar in *x*-*y* track.

B. IPDB Model Validation

The IPDB approach to validate the combined longitudinal and lateral vehicle dynamics is presented in this section. The QCar is first controlled to complete a round track to cover various vehicle operations. The global vehicle position information is obtained through the OptiTrack system in real time, and the velocities are calculated from the changing positions. The logged signals are displayed in Fig. 4. In the meantime, the vehicle operation state signals are presented in Fig. 5. The vehicle longitudinal speed is obtained from the global to local coordinate transformation, and the tire slip ratio is calculated by combining the vehicle longitudinal velocity and the wheel rotational speed. The steering angle is converted from the steering motor. In addition, the vehicle heading angle can also be obtained from the OptiTrack system, from which the vehicle yaw rate can be calculated.

At this point, the data matrices described in (6) can be formulated so that the system matrices $[\mathscr{B}_k \quad \mathscr{A}_k]$ in (7) can be identified. The numerical values within the matrices are omitted here, however, the model performances are reported in Fig. 6. The IPDB modeling results are compared to the Optical system measurements, and the error signals between them are also presented. It is obvious that the IPDB approach without knowing the vehicle configuration parameters can reflect the coupled longitudinal and lateral vehicle dynamics response with excellent precision. The Root-Mean-Square-Error (RMSE) for longitudinal velocity, lateral velocity, and yaw rate is calculated to be $0.0074 m/s$, $0.0013 m/s$ and $0.0131 rad/s$, respectively, which is almost trivial. This demonstrates the high accuracy of the IPDB modeling approach.

C. Gain-Scheduling Controller Design and Experimental Results

First, the range of four scheduling parameters are selected based on the experimental data. Yaw rate $\gamma \in [-1,1]$, $\frac{1}{v_x} \in [1,10]$, slip ratio $\lambda_f \in [-0.4,0.4]$, $\delta_f \in [-0.5,0.5]$. Note that, the upper bound of $\frac{1}{v_x}$ is approximated to avoid infinity when the longitudinal velocity is near 0.

The weighting matrices are tuned in experiments and the following matrices are finally taken. $Q = diag(1,1,100)$, $R = 1$, which lead to $C = diag(1,1,10,1)$ and $D = [0,0,0,1]^T$. These matrices are employed in the controller synthesis conditions in Theorem 1 to solve for controller variables

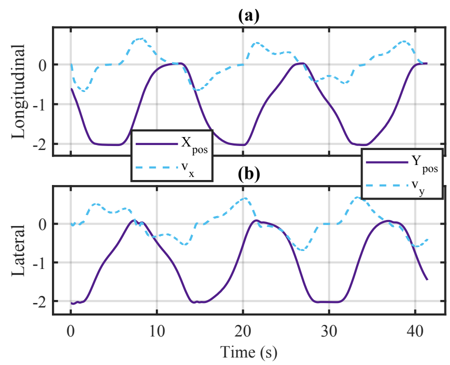

Fig. 4. (a) Longitudinal position and speed; (b) Lateral position and speed.

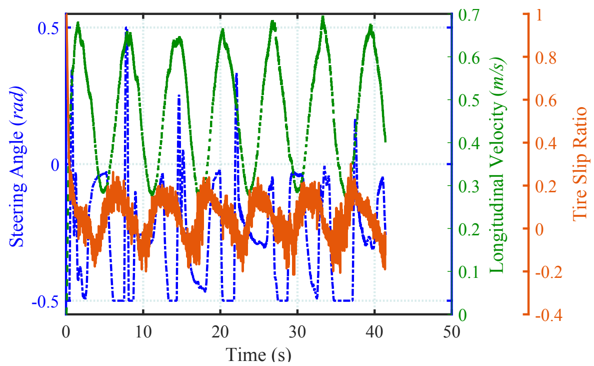

Fig. 5. Signals for IPDB modeling (Steering angle: blue dashed line; Longitudinal speed: green dotted line; Slip ratio: orange solid line)

P_{ik}, F_{ik}. After that, the controller variables were stored and used to online calculate controller gain $K = F(\rho_k)P^{-1}(\rho_k)$ with the real-time measurement of scheduling parameter ρ. The controller computation (inverse and matrix multiplications) was performed in Simulink and the controller gain is transmitted to Qcar by wireless communication.

The proposed controller is implemented on the experimental platform and was compared with a Stanley controller, which is often used in lateral control. The experimental results are plotted in Figs. 7-8, and the numerical analysis of control performance is summarized in Table I.

It can be seen that the proposed IPDB controller achieves similar tracking performance in v_x as the Stanley controller since the longitudinal dynamics are mainly controlled by the wheel speed. However, the proposed control has much better control performance in v_y and γ than the Stanley controller, which achieves better tracking and can reduce the tracking error to a smaller order of magnitude. The underlying reason is that the IPDB controller captures the combined longitudinal-lateral dynamics and schedules controller gains based on vehicle status, rather than the Stanley controller just considers the lateral dynamics without considering coupled longitudinal dynamics. This can be observed as well in the comparisons of control inputs, if one notices that different steering strategies are adopted by two controllers to adjust the pose of the vehicle.

In particular, at the first seconds, see zoom-in figures in Fig. 7, the proposed IPDB-based controller has faster

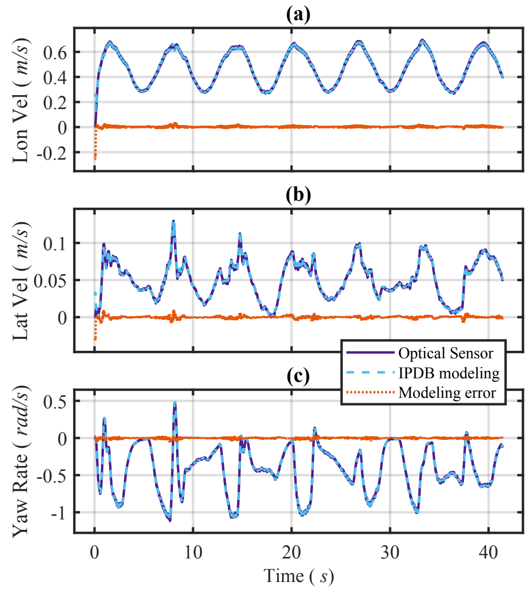

Fig. 6. Combined vehicle dynamics model validation: (a) Longitudinal velocity; (b) Lateral velocity; (c) Yaw rate.

TABLE I
COMPARISON OF CONTROLLER PERFORMANCES.

Singals	RMSE		RRMSE (%)	
	IPDB	Stanley	IPDB	Stanley
$v_x(m/s)$	0.773	0.776	2.45	2.46
$v_y(m/s)$	0.022	0.62	0.062	0.462
$\dot{\gamma}(rad/s)$	0.07	0.158	0.4	0.98
$\gamma(rad)$	0.04	0.13	0.035	0.12

convergence to track the reference trajectory than the Stanley controller. It can also been observed in the yaw rate and lateral velocity tracking during 7-15 seconds. The Stanley controller renders an obvious phase lag in the responses, however, the IPDB-based controller can enable rapid convergence and follow the reference trajectory.

V. CONCLUSION

This paper proposes a new method to model and control the coupled longitudinal and lateral vehicle dynamics using the IPDB approach. Specifically, the nonlinear coupled dynamic model originated from physical principles is derived into an affine-dependent LPV representation, and then the IPDB modeling approach using experimental data is employed to establish the system transformation matrices and reconstruct the coupled vehicle dynamics. On the foundation of the IPDB model, an LPV gain-scheduling tracking controller is designed to steer the vehicle to track a given reference trajectory. The effectiveness of the proposed modeling and control algorithm are validated experimentally on the Quanser QCar platform with an optical measurement system. The IPDB model is validated with excellent modeling accuracy, and the tracking controller designed based on the IPDB model outperforms the baseline Stanley controller

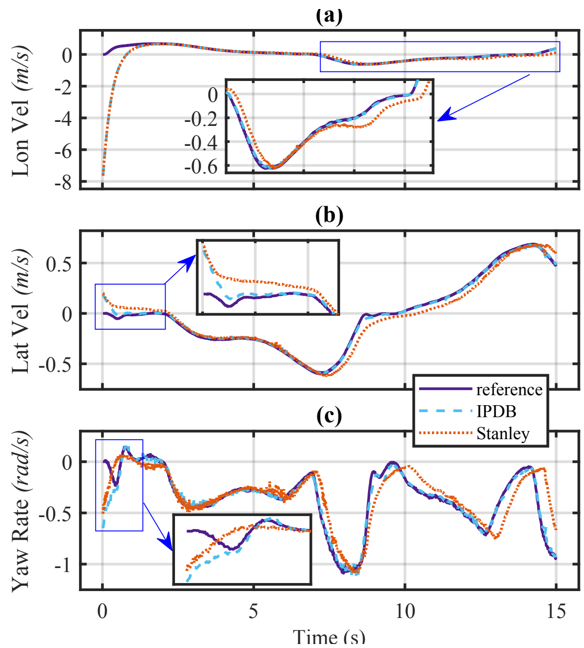

Fig. 7. Tracking performance validation: (a) Longitudinal velocity; (b) Lateral velocity; (c) Yaw rate.

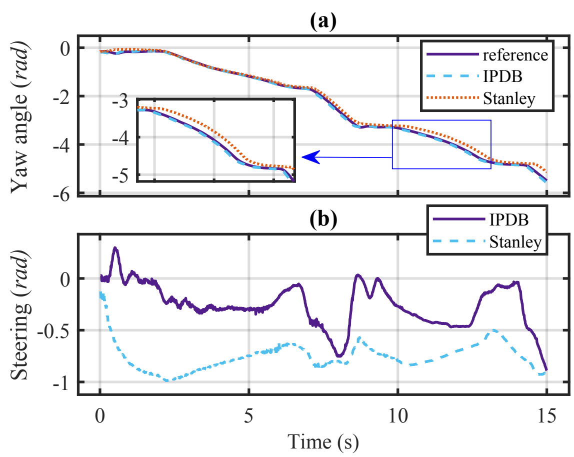

Fig. 8. Tracking performance validation: (a) Yaw angle; (b) Steering input.

in both tracking accuracy and convergence speed.

Acknowledgment

The work fulfilled by Southeast University are supported partially by the National Natural Science Foundation of China under Grant 52025121 and Grant 51975118, and partially by "The Fundamental Research Funds for the Central Universities".

References

[1] L. Xu, Y. Wang, H. Sun, J. Xin, and N. Zheng, "Integrated longitudinal and lateral control for Kuafu-II autonomous vehicle," *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 7, pp. 2032–2041, 2015.

[2] R. Rajamani, *Vehicle dynamics and control*. Springer Science & Business Media, 2011.

[3] W. Wei, H. Dourra, and G. Zhu, "Vehicle tire traction torque estimation using a dual extended kalman filter," *Journal of Dynamic Systems, Measurement, and Control*, vol. 144, no. 3, p. 031004, 2022.

[4] W. Wei, H. Dourra, and G. G. Zhu, "Transfer case clutch torque estimation using an extended kalman filter with unknown input," *IEEE/ASME Transactions on Mechatronics*, vol. 27, no. 5, pp. 2580–2588, 2022.

[5] E. Velenis, P. Tsiotras, C. C. de Wit, and M. Sorine, "Dynamic tire friction models for combined longitudinal and lateral vehicle motion," *Vehicle system dynamics*, vol. 43, no. 1, pp. 3–29, 2005.

[6] J. Nilsson, M. Brännström, J. Fredriksson, and E. Coelingh, "Longitudinal and lateral control for automated yielding maneuvers," *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 5, pp. 1404–1414, 2016.

[7] R. Zhang, Y. Ma, Z. Li, R. Malekian, and M. Á. Sotelo, "Energy dissipation based longitudinal and lateral coupling control for intelligent vehicles," *IEEE Intelligent Transportation Systems Magazine*, vol. 10, no. 2, pp. 121–133, 2018.

[8] R. Attia, R. Orjuela, and M. Basset, "Combined longitudinal and lateral control for automated vehicle guidance," *Vehicle System Dynamics*, vol. 52, no. 2, pp. 261–279, 2014.

[9] L. Ge, Y. Zhao, F. Ma, and K. Guo, "Towards longitudinal and lateral coupling control of autonomous vehicles using offset free MPC," *Control Engineering Practice*, vol. 121, p. 105074, 2022.

[10] Z. Li, P. Wang, H. Liu, Y. Hu, and H. Chen, "Coordinated longitudinal and lateral vehicle stability control based on the combined-slip tire model in the MPC framework," *Mechanical Systems and Signal Processing*, vol. 161, p. 107947, 2021.

[11] Z. Li, H. Chen, H. Liu, P. Wang, and X. Gong, "Integrated longitudinal and lateral vehicle stability control for extreme conditions with safety dynamic requirements analysis," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 10, pp. 19 285–19 298, 2022.

[12] R. Rajamani, H.-S. Tan, B. K. Law, and W.-B. Zhang, "Demonstration of integrated longitudinal and lateral control for the operation of automated vehicles in platoons," *IEEE Transactions on Control Systems Technology*, vol. 8, no. 4, pp. 695–708, 2000.

[13] B. A. H. Vicente, S. S. James, and S. R. Anderson, "Linear system identification versus physical modeling of lateral–longitudinal vehicle dynamics," *IEEE Transactions on Control Systems Technology*, vol. 29, no. 3, pp. 1380–1387, 2020.

[14] S. Kumarawadu and T. T. Lee, "Neuroadaptive combined lateral and longitudinal control of highway vehicles using RBF networks," *IEEE Transactions on Intelligent Transportation Systems*, vol. 7, no. 4, pp. 500–512, 2006.

[15] L. Cui, K. Ozbay, and Z.-P. Jiang, "Combined longitudinal and lateral control of autonomous vehicles based on reinforcement learning," in *2021 American Control Conference (ACC)*. IEEE, 2021, pp. 1929–1934.

[16] W. Wei, W. Zhuang, Q. Gao, G. Yin, and T. He, "Online modeling of lateral vehicle dynamics via recursive integrated physics-data-based method," *IEEE Transactions on Intelligent Vehicles*, pp. 1–9, 2023.

[17] W. Wei, G. Yin, and T. He, "Physics-informed data-based LPV modeling and validations of lateral vehicle dynamics," *IEEE Transactions on Intelligent Vehicles*, pp. 1–10, 2023.

[18] S. Fergani, L. Menhour, O. Sename, L. Dugard, and B. D'Andréa-Novel, "Integrated vehicle control through the coordination of longitudinal/lateral and vertical dynamics controllers: Flatness and LPV/-based design," *International Journal of Robust and Nonlinear Control*, vol. 27, no. 18, pp. 4992–5007, 2017.

[19] P. Gahinet and P. Apkarian, "A linear matrix inequality approach to H_∞ control," *International journal of robust and nonlinear control*, vol. 4, no. 4, pp. 421–448, 1994.

[20] C. W. Scherer, "LPV control and full block multipliers," *Automatica*, vol. 37, no. 3, pp. 361–375, 2001.

[21] R. C. Oliveira and P. L. Peres, "Parameter-dependent LMIs in robust analysis: Characterization of homogeneous polynomially parameter-dependent solutions via LMI relaxations," *IEEE Transactions on Automatic Control*, vol. 52, no. 7, pp. 1334–1340, 2007.

[22] C. M. Agulhari, A. Felipe, R. C. Oliveira, and P. L. Peres, "Algorithm 998: The robust LMI parser—a toolbox to construct LMI conditions for uncertain systems," *ACM Transactions on Mathematical Software (TOMS)*, vol. 45, no. 3, pp. 1–25, 2019.

[23] J. F. Sturm, "Using SeDuMi 1.02, a MATLAB toolbox for optimization over symmetric cones," *Optimization methods and software*, vol. 11, no. 1-4, pp. 625–653, 1999.