Wheel Speed Dynamics and Control for Autonomous Drifting at 1/10th-Scale

Kaijia Chen and Alexander Davydov

***Abstract*— Regulating rear wheel speed is key in enabling autonomous drifting on full-scale vehicles. However, on small-scale cars, the wheel speed dynamics are often neglected because the tires are small and light. We show that this simplification of the vehicle dynamics can result in failure to remain in a drifting maneuver. On a minimally-modified 1/10th-scale car whose hobby-grade electronic speed controller only accepts open-loop PWM throttle commands, we identify a drivetrain time constant of 0.37 s, comparable to the 0.52 s time constant of the unstable drift mode. To address wheel speed control for autonomous drifting at small-scale, we implement a layered controller in which a nonlinear model predictive controller plans steering and wheel speed references, and an onboard PI controller with model-inverse feedforward tracks this wheel speed reference. In our experiments, this layered controller sustained the drift in all five trials while a baseline controller using a static PWM-to-wheel speed map lost the drift in all five trials.**

I. Introduction

Drifting is a highly dynamic vehicle maneuver in which the rear tires operate at or beyond friction saturation, holding the vehicle in a counter-steered state with large sideslip angles and high yaw rates. Beyond its prominence in motorsport, drifting serves as a useful testbed for vehicle control at handling limits, with applications to emergency collision avoidance [1] and loss-of-control recovery on low-friction surfaces [2].

For full-scale vehicles, the rear wheel speed, ω, is a key control input for drifting. When the friction forces in the rear tires are saturated, adjusting the rear wheel speed adjusts the longitudinal slip ratio and trades off the lateral stabilizing force for the longitudinal drive force. This tradeoff can be leveraged to approximately control rear tire friction forces. Goh et al. [3], [4] exploited this mechanism with a low-level wheel speed controller and demonstrated that accounting for the rear wheel speed dynamics is essential for accurate tracking of drifting trajectories. More modern approaches to full-scale autonomous drifting exploit this mechanism by explicitly incorporating wheel speed dynamics into nonlinear model predictive control (NMPC) frameworks [5]–[8]. These wheel speed dynamics arise from a torque balance on the rear axle of the vehicle and can be sensitive to uncertainty in the drive torque input or rear axle moment of inertia.

Small-scale autonomous vehicles (1/10th-scale and smaller) are cost-effective, high-throughput platforms for validating such dynamic control strategies [9]–[12]. However, operating at a small scale introduces two distinct challenges. First, motion evolves at a faster time-scale than in full-scale cars. At the same time, actuator delays and communication latency do not scale down with the size of the vehicle. Second, off-the-shelf hobby-grade electronic speed controllers (ESCs) generally accept only open-loop pulse-width modulation (PWM) duty cycle commands and provide limited motor torque estimation or current feedback.

Authors are with the Department of Mechanical Engineering and the Ken Kennedy Institute, Rice University, Houston, TX, 77005 USA. {kc222, davydov}@rice.edu.

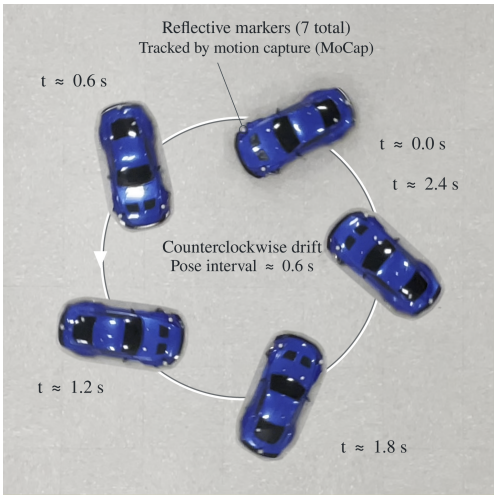


Fig. 1: Overhead drifting trajectory from closed-loop control using the implemented layered control architecture.

One may expect that, at this small scale, wheel speed dynamics can be neglected due to the small size and light nature of the tires and axle. Indeed, existing small-scale drifting controllers neglect these dynamics, either commanding longitudinal force through a static drivetrain model [10] or treating rear wheel speed as a directly assignable control input [12], [13]. To the best of our knowledge, whether this simplification holds for small-scale drifting has not been properly examined.

In this paper, we examine this question on a minimally-modified 1/10th-scale rear-wheel-drive car. Working within the limits of a hobby-grade ESC, without adding custom torque sensors or modifying high-current wiring, we develop a layered controller that explicitly models and regulates rear wheel speed dynamics. Wheel speed dynamics are estimated via a first-order ARX model identified directly from manual driving data. An offboard NMPC plans steering and wheel speed references while enforcing rear-tire slip saturation, and an onboard microcontroller executes high-frequency PI control with model-inverse feedforward based on local encoder feedback.

The primary contributions of this work are as follows.

1) We demonstrate experimentally that transient wheel speed dynamics cannot be neglected for drift control, *even at 1/10th-scale*. On our platform, the identified drivetrain time constant (0.37 s) is comparable to the time constant of the unstable drift mode (0.52 s).
2) Inspired by [3], we develop a layered control architecture that accounts for the wheel speed dynamics. We demonstrate that this control design enables sustained drifting across five trials, whereas a baseline NMPC that statically maps PWM to wheel speed, loses the drift in all five trials.

In particular, we instantiate this control architecture on an off-the-shelf hobby-grade RC vehicle. The implemented approach only requires a wheel encoder, IMU, onboard microcontroller, and a motion capture system.

The remainder of this paper is organized as follows. Section II details the single-track vehicle model and nonlinear tire model. Section III outlines system identification. Section IV describes the proposed control architecture. Section V presents experimental indoor drift performance against a baseline controller, and Section VI concludes and discusses future research directions.

II. Background

A. Single-Track Vehicle Model

We model the car as a rigid body with each axle's tires lumped together, following the single-track model widely used for autonomous drifting [3], [5], [14], [15]. The car is rear-wheel drive, so we treat the front wheels as free-rolling. The vehicle's motion is described by the yaw rate r, speed V, and sideslip angle β between velocity and heading. Its position relative to a path of curvature κ_{ref} is described by the lateral error e and course error $\Delta\phi$.

Newton's laws give the chassis dynamics:

$$\dot{r} = \frac{aF_{yF}\cos\delta - bF_{yR}}{I_z}, \tag{1}$$

$$\dot{V} = \frac{F_{xR}\cos\beta - F_{yF}\sin(\delta-\beta) + F_{yR}\sin\beta}{m}, \tag{2}$$

$$\dot{\beta} = \frac{F_{yF}\cos(\delta-\beta) - F_{xR}\sin\beta + F_{yR}\cos\beta}{mV} - r, \tag{3}$$

where m is the mass, I_z the yaw inertia, a and b the distances from the center of mass to the front and rear axles, and δ the steering angle. The tire forces, modeled in the next subsection, are the front and rear lateral forces F_{yF}, F_{yR} and the rear longitudinal force F_{xR}. Since our car is rear-wheel drive, so we make the assumption that there is no front longitudinal force ($F_{xF} = 0$).

The path errors evolve as

$$\dot{e} = V\sin\Delta\phi, \tag{4}$$

$$\dot{\Delta\phi} = \dot{\beta} + r - \frac{\kappa_{\text{ref}} V\cos\Delta\phi}{1-\kappa_{\text{ref}} e}, \tag{5}$$

where $\dot{\beta}+r$ is the rotation rate of the velocity vector and the last term in (5) is the rotation rate of the path tangent at the closest point. For a circular drift of radius R, $\kappa_{\text{ref}} = \pm 1/R$.

B. Tire Force Modeling

To complete the vehicle model, we express the tire forces as functions of tire slip. Lateral slip is characterized by the slip angle, α, between the wheel heading and its velocity direction, while longitudinal slip is characterized by the slip ratio κ, i.e., the normalized difference between wheel circumferential speed and longitudinal ground speed. Tire forces initially increase with slip before reaching saturation. During drifting, the rear tires operate in this high-slip, saturated regime.

Front tire. The front slip angle, α_F, follows from the chassis kinematics, and the front lateral force is modeled according to the Pacejka Magic Formula [16]:

$$\begin{aligned} \alpha_F &= \tan^{-1}\left(\frac{V\sin\beta + ar}{V\cos\beta}\right) - \delta, \\ q_F &= B_F(\alpha_F - S_h), \\ z_F &= q_F - E_F\big(q_F - \tan^{-1} q_F\big), \\ F_{yF} &= -D_F\sin\big(C_F\tan^{-1} z_F\big), \end{aligned} \tag{6}$$

where B_F, C_F, D_F, E_F set the stiffness, shape, peak, and curvature of the force curve, respectively, and S_h is the horizontal shift which models asymmetries such as a residual steering offset.

Rear tire. With rear wheel speed ω and wheel radius R_w, the rear slip angle, α_R, and rear slip ratio, κ_R are

$$\alpha_R = \tan^{-1}\left(\frac{V\sin\beta - br}{V\cos\beta}\right), \tag{7}$$

$$\kappa_R = \frac{R_w\omega - V\cos\beta}{V\cos\beta}. \tag{8}$$

We assume the rear tire is always saturated during a drift. It then transmits its maximum friction force magnitude $F_{\text{sat}} = \mu F_{zR}$, where μ is the tire-road friction coefficient and $F_{zR} = amg/(a+b)$ is the static rear axle load, directed by the combined slip $\sigma_R = \sqrt{\kappa_R^2 + \tan^2\alpha_R}$:

$$F_{xR} = F_{\text{sat}}\frac{\kappa_R}{\sigma_R}, \qquad F_{yR} = -F_{\text{sat}}\frac{\tan\alpha_R}{\sigma_R}. \tag{9}$$

Because $F_{xR}^2 + F_{yR}^2 = F_{\text{sat}}^2$, friction fixes the magnitude of the rear tire force while the rear wheel speed controls its direction. Increasing ω increases κ_R and rotates the rear tire force toward the longitudinal axis, increasing F_{xR} at the expense of F_{yR}.

III. Identifying Vehicle and Wheel Speed Dynamics

The experimental platform is a Traxxas 4-Tec Drift (model 105237-4) [17], a 1/10th-scale rear-wheel-drive electric RC car with a locked rear axle, a brushed 550-size motor, and a hobby-grade XL-5 ESC that provides no current or torque feedback. The vehicle is controlled through a steering servo and the ESC. An ESP32-S3 microcontroller replaces the RC remote, receives commands from an offboard computer over Wi-Fi, and directly generates the steering and throttle PWM signals. The steering pulse width ranges from $1000-2000$ μs and is mapped linearly to steering angles from -56° to

56° using manual calibration. The throttle pulse width is restricted to $1500 - 1850$ μs, corresponding to neutral and forward operation. An onboard MPU6050 inertial sensor measures yaw rate, and an H2010 optical encoder with an LM393 interface measures wheel speed. A 12-camera OptiTrack motion-capture system tracks the vehicle's position and orientation at 240 Hz. From these measurements, we estimate velocity and sideslip angle and compute lateral and course errors relative to the reference path.

A. System Identification

Tire-model parameters were identified from 20 minutes of manual RC driving, mostly drifting, with Traxxas Stability Management (TSM) [17] enabled. Data were recorded at 100 Hz. Since (1)–(3) are linear in the three tire forces, these forces can be solved for at each sample:

$$\begin{bmatrix} a\cos\delta & -b & 0 \\ -\sin(\delta-\beta) & \sin\beta & \cos\beta \\ \cos(\delta-\beta) & \cos\beta & -\sin\beta \end{bmatrix} \begin{bmatrix} F_{yF} \\ F_{yR} \\ F_{xR} \end{bmatrix} = \begin{bmatrix} I_z\dot{r} \\ m\dot{V} \\ mV(\dot{\beta}+r) \end{bmatrix}.$$

The terms $m\dot{V}$ and $mV(\dot{\beta}+r)$ are the tangential and normal components of the center-of-mass acceleration. We compute them by numerically differentiating the motion-capture position twice and projecting onto the velocity direction, which avoids differentiating β directly. $\dot{r}$ is obtained by numerically differentiating the IMU yaw rate. Each force sample is then paired with its slip quantities from (6)–(8). Table I lists the identified parameters, and Fig. 2 shows the fits.

For the front tire, we group the samples into 90 equal-width slip-angle bins (about 1.2° each) and fit the Magic Formula to the median force in each bin, obtaining $R^2 = 0.987$ on the bin medians (Fig. 2a). The fitted horizontal shift $S_h = -8.52^\circ$ indicates an offset in the zero steering angle.

For the rear axle, we use only samples where the rear tire is saturated. We bin these samples by force direction $\theta = \operatorname{atan2}(F_{yR}, F_{xR})$ in 5° arcs. The 95th-percentile force magnitude in each bin gives one point on the outer envelope of the data, so outliers do not set the limit. A least-squares circle centered at the origin, fitted to these points, gives $F_{\text{sat}} = 6.71$ N (Fig. 2b). This corresponds to an effective friction coefficient $\mu = F_{\text{sat}}/F_{zR} = 0.666$.

The friction circle sets the magnitude of the saturated rear force, and its direction follows the slip direction from (8). To test the direction model separately from magnitude, we take the inverted force magnitude at each sample, apply it along the slip direction, and compare the resulting components with the inverted forces (Fig. 2c,d). Points on the $y = x$ line mean the predicted and inverted components agree. The RMSE is 0.65 N for F_{xR} and 0.67 N for F_{yR}, about 10% of F_{sat}.

B. Wheel Speed Dynamics

Under the saturated rear-tire assumption, wheel speed sets the direction of the rear force through the slip ratio in (9), so predicting the response to throttle requires a model of the wheel speed dynamics. Full-scale drifting controllers model the wheel through a torque balance [15], which requires the drive torque. Our hobby-grade ESC accepts only a PWM command, so we identify the wheel speed response directly from the PWM command and the measured vehicle speed.

Let $p_k = \text{PWM}_k - 1500$ denote the throttle offset in μs and $v_{x,k} = V_k \cos\beta_k$ the longitudinal velocity at a sample instant, k. Using driving data sampled at $\Delta t_c = 0.01$ s, we identify the first-order affine ARX model by least squares:

$$\omega_{k+1} = c_0 + c_\omega \omega_k + c_p p_k + c_v v_{x,k}, \tag{10}$$

with $c_0 = -0.35009$, $c_\omega = 0.97358$, $c_p = 0.0097364$, and $c_v = 0.23414$. The structure is inspired by the rear axle torque balance $I_w\dot{\omega} = \tau_m - R_w F_{xR}$ [15]. The motor, gearbox, rear axle, and wheels act as one inertia, so the model is first order in ω. Linearizing and replacing the unmeasured motor torque with the throttle offset gives (10), where $c_p p_k$ approximates the motor torque, $c_\omega < 1$ captures friction, back-EMF, and wheel-speed-dependent road load, $c_v v_{x,k}$ captures the speed dependence of the road load, and c_0 absorbs constant offsets.

TABLE I: Vehicle and Tire-Model Parameters

Vehicle			*Tire*		
m	1.958	kg	B_F	6.581	rad^{-1}
I_z	0.040	kg m^2	C_F	1.779	–
a	0.1347	m	D_F	3.599	N
b	0.1223	m	E_F	0.719	–
R_w	0.035	m	S_h	-0.1487	rad
			F_{sat}	6.709	N

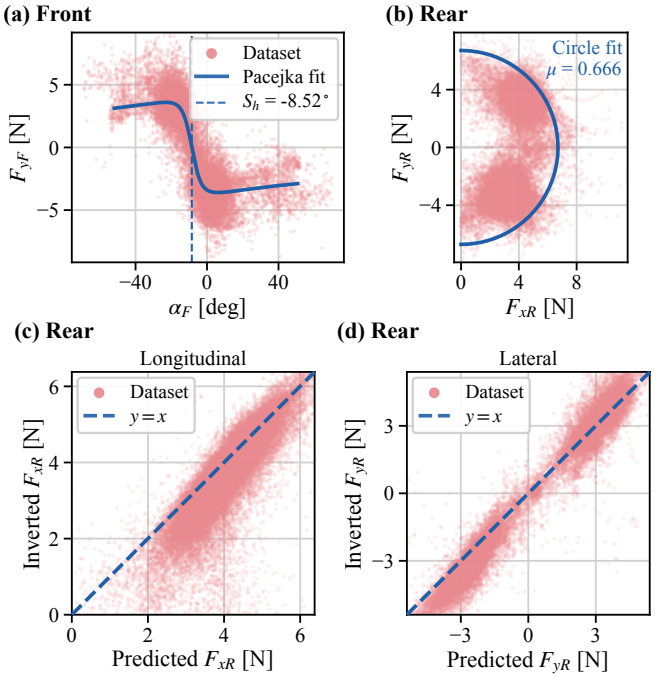


Fig. 2: Tire-model identification. (a) Front Magic Formula fit; (b) rear high-slip forces and fitted friction circle; (c) longitudinal and (d) lateral rear force: inverted force vs. predicted force.

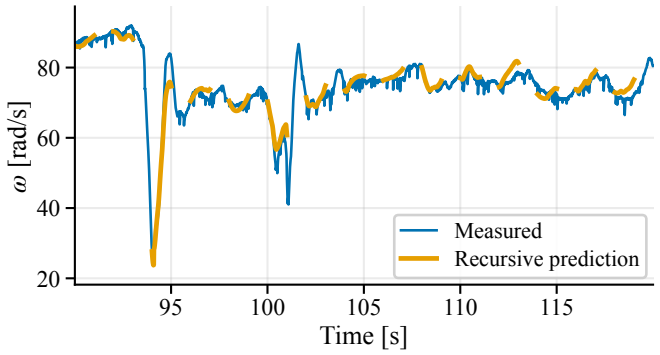


Fig. 3: Measured rear wheel speed and open-loop model predictions over 1 s horizons, initialized from measurements every 2 s.

The equivalent continuous-time model is

$$\dot{\omega} = \frac{\omega_{\mathrm{ss}}(p, v_x) - \omega}{\tau_d}, \quad \omega_{\mathrm{ss}}(p, v_x) = \frac{c_0 + c_p p + c_v v_x}{1 - c_\omega}, \tag{11}$$

with $c_\omega = e^{-\Delta t_c/\tau_d}$. The drivetrain time constant is therefore $\tau_d = -\Delta t_c / \ln c_\omega \approx 0.37$ s. This model reproduces (10) at the sampling instants when p and v_x are held constant over each interval. The identified time constant is not negligible and motivates retaining wheel speed as a dynamic state rather than using a static PWM-to-wheel speed map.

We evaluate the wheel-speed model on a 30 s drifting segment taken from the collected data (Fig. 3). Every 2 s, a rollout starts from the measured wheel speed and runs open loop for 1 s (100 steps), driven by the logged PWM and longitudinal velocity without wheel-speed feedback. Evaluated at the final step of each of the 15 rollouts, the RMSE is 4.44 rad/s, or 6.0% of the mean wheel speed.

IV. Layered Control Architecture Design

Our control architecture follows a layered architecture similar to the one in Goh et al. [3], in which a high-level controller specifies a rear wheel speed reference and a low-level controller tracks it with model-based feedforward and feedback. There are two key differences in our approach. First, our feedforward controller inverts the identified drivetrain model (10) at the PWM-level rather than the axle torque balance. Second, our high-level controller is an NMPC that models the closed-loop wheel speed as a first-order lag in its prediction. Our control architecture is presented in Fig. 4.

Our NMPC is run offboard and our low-level controller is onboard and is a wheel speed PI controller. The NMPC coordinates steering and wheel speed to track the desired drift, while the low-level controller regulates wheel speed using local encoder feedback. This low-level feedback loop runs on the ESP32-S3 at 100 Hz and does not pass through the Wi-Fi link.

A. High-Level Nonlinear MPC

The NMPC predicts vehicle motion using the state vector

$$x = \begin{bmatrix} e & \Delta\phi & \beta & r & V & \delta & \omega & \delta_{\mathrm{ref}} & \omega_{\mathrm{ref}} \end{bmatrix}^\top. \tag{12}$$

The references δ_{ref} and ω_{ref} are the control actions sent to the hardware: a steering-angle command to the servo and a wheel-speed reference to the low-level onboard controller. The states δ and ω are the realized steering angle and rear wheel speed, which enter the tire slips (6)–(8). Since neither actuator responds instantly, each is modeled as a first-order lag,

$$\dot{\delta} = \frac{\delta_{\mathrm{ref}} - \delta}{\tau_s}, \qquad \dot{\omega} = \frac{\omega_{\mathrm{ref}} - \omega}{\tau_\omega}. \tag{13}$$

Wheel speed cannot be set directly; the onboard controller tracks ω_{ref} by adjusting the throttle using the drivetrain model of Section III-B. The lag τ_ω therefore is an estimate for the closed-loop time constant rather than the open-loop drivetrain constant, and it is smaller than the drivetrain time constant $\tau_d \approx 0.37$ s. Modeling the wheel speed this way also helps absorb any mismatch between the identified drivetrain model and the real vehicle's wheel speed dynamics.

The optimization inputs are the reference rates,

$$\nu = \begin{bmatrix} \dot{\delta}_{\mathrm{ref}} & \dot{\omega}_{\mathrm{ref}} \end{bmatrix}^\top, \tag{14}$$

which lets the NMPC penalize and bound how quickly each reference changes. In total, we have 9 states and 2 inputs.

At each update, the controller solves

$$\begin{aligned} \min_{\nu_0,\ldots,\nu_{N-1}} \quad & \sum_{k=0}^{N-1} \left(\|\tilde{x}_k\|_Q^2 + \|\nu_k\|_R^2 \right) + \|\tilde{x}_N\|_P^2 \\ & + w_\sigma \sum_{k=1}^{N} \left[\sigma_{\min} - \sigma_R(x_k) \right]_+^2 \\ \text{s.t.} \quad & x_0 = \hat{x}, \\ & x_{k+1} = f_d(x_k, \nu_k), \quad k \in \{0, \ldots, N-1\}, \\ & |\nu_k| \le \nu_{\max}, \quad k \in \{0, \ldots, N-1\}, \\ & |\delta_{\mathrm{ref},k}| \le \delta_{\max}, \quad k \in \{1, \ldots, N\}, \\ & 0 \le \omega_{\mathrm{ref},k} \le \omega_{\max}, \quad k \in \{1, \ldots, N\}, \end{aligned} \tag{15}$$

where $\tilde{x}_k = x_k - x^\star$ is the deviation from the target drift equilibrium, with $\Delta\phi$ wrapped to $(-\pi, \pi]$, $\|z\|_Q^2 = z^\top Q z$, and $[z]_+ = \max(0, z)$. The bounds on ν_k holds elementwise with $\nu_{\max} = [\dot{\delta}_{\max}, \dot{\omega}_{\max}]^\top$. The weights Q and R are diagonal and P solves the discrete algebraic Riccati equation for the model linearized at $x^\star$. The last term in the objective promotes plans where the vehicle is in a high-slip regime; σ_R is the total rear slip of the tire model, and the penalty is zero unless the prediction shows the rear tires regaining grip. f_d is a fourth-order Runge–Kutta discretization of the model and $\hat{x}$ is the measured state. The references are bounded by $\delta_{\max} = 50°$ and $\omega_{\max} = 140$ rad/s. The rate bounds $\dot{\delta}_{\max} = 500°/\mathrm{s}$ and $\dot{\omega}_{\max} = 100$ rad/s^2 are set conservatively below the steering rate and wheel acceleration measured on the vehicle.

The horizon has $N = 20$ steps of $\Delta t = 0.05$ s. The problem is solved at 50 Hz by direct single shooting, with one Gauss–Newton SQP iteration per update warm-started from the previous solution, in the style of the real-time iteration scheme [18]. The decision variables are the reference rates, held constant over each prediction step, so the planned δ_{ref} and ω_{ref} are piecewise linear in time. Between updates,

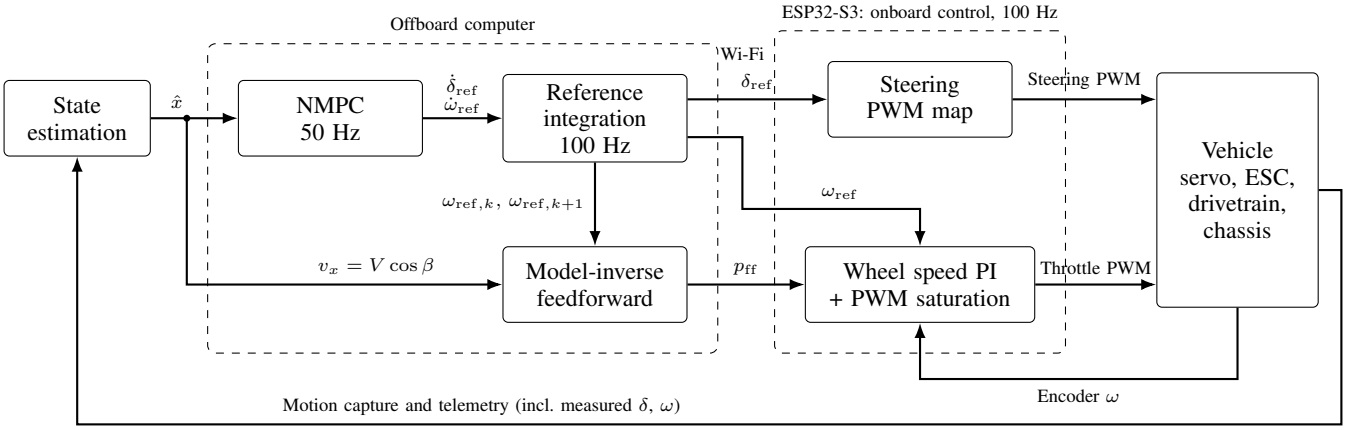


Fig. 4: Schematic of the layered control architecture. The NMPC plans steering and wheel speed reference rates at 50 Hz; these are integrated at 100 Hz into references sent over Wi-Fi. The PWM throttle is the model-inverse feedforward p_{ff}, computed from the reference wheel speed values $\omega_{\mathrm{ref},k}, \omega_{\mathrm{ref},k+1}$ and the estimated longitudinal speed v_x, corrected by an onboard PI on the encoder wheel speed. The NMPC models both the servo and the closed-loop wheel speed dynamics as first-order lags with time constants τ_s, τ_ω, respectively.

the first planned rate is integrated at the control period $\Delta t_c = 0.01$ s, e.g., $\omega_{\mathrm{ref}}(t + \Delta t_c) = \omega_{\mathrm{ref}}(t) + \Delta t_c\, \dot{\omega}_{\mathrm{ref},0}$, so the references ramp smoothly every 10 ms.

B. Low-Level Wheel Speed Controller

The steering reference from NMPC is converted to servo PWM through the calibrated steering map. For the wheel speed, the offboard computer inverts the identified drivetrain model (10) to obtain the feedforward throttle offset

$$p_{\mathrm{ff},k} = \frac{\omega_{\mathrm{ref},k+1} - c_\omega \omega_{\mathrm{ref},k} - c_0 - c_v v_{x,k}}{c_p}, \tag{16}$$

saturated to $[0,\, p_{\max}]$, where $v_{x,k} = V_k \cos\beta_k$. This is the throttle that would carry the wheel speed from $\omega_{\mathrm{ref},k}$ to $\omega_{\mathrm{ref},k+1}$ if the wheel were exactly at the reference. Since the model (10) is identified at the same $\Delta t_c = 0.01$ s period at which the references are updated, no resampling is needed. The wheel speed reference and the feedforward are sent to the ESP32-S3, which adds a PI correction on the measured wheel speed:

$$\begin{aligned} \mathrm{PWM}_k &= 1500 + p_{\mathrm{ff},k} + K_p e_{\omega,k} + K_i z_k, \\ z_{k+1} &= z_k + \Delta t_c e_{\omega,k}, \qquad e_{\omega,k} = \omega_{\mathrm{ref},k} - \omega_k, \end{aligned} \tag{17}$$

with PWM_k saturated to $[1500,\, 1500 + p_{\max}]$ μs. The feedforward supplies the throttle that the reference trajectory nominally requires, and the PI controller corrects the tracking errors caused by model mismatch and disturbances.

V. Experimental Results

We evaluate our layered control architecture on drifting constant-radius trajectories (donuts).

A. Experimental Setup

The reference trajectory is a constant-radius circular drift of radius $R = 0.50$ m at a prescribed speed $V^\star = 1.42$ m/s. Steady circular motion requires $r^\star = V^\star/R = 2.84$ rad/s and zero path-tracking errors, $e^\star = \Delta\phi^\star = 0$. Using the identified tire model, we numerically solve for the remaining states $(\beta^\star, \delta^\star, \omega^\star)$ such that $\dot{\beta} = \dot{r} = \dot{V} = 0$. Among the feasible solutions, we select the counter-steered, high-slip drift equilibrium,

$$\begin{aligned} V^\star = 1.42\ \mathrm{m/s}, \quad r^\star = 2.84\ \mathrm{rad/s}, \quad \beta^\star = -37.21^\circ, \\ \delta^\star = -16.49^\circ, \quad \omega^\star = 72.97\ \mathrm{rad/s}. \end{aligned} \tag{18}$$

The corresponding rear slip ratio is $\kappa_R^\star = 1.258$, so the rear tires operate well inside the saturated regime.

Linearizing the chassis dynamics in (β, r, V) about this equilibrium, with steering and wheel speed held fixed, gives eigenvalues

$$\{\lambda_1, \lambda_2, \lambda_3\} = \{-9.27, -1.30, 1.91\}\ \mathrm{s}^{-1}. \tag{19}$$

The positive eigenvalue indicates that this drift equilibrium is a saddle and open-loop unstable. The time constant for this unstable mode is $1/1.91 = 0.52$ s, which is comparable to the time constant of 0.37 s identified in the modeled wheel speed dynamics (11).

The drift is initiated with an open-loop kick. The PWM throttle is held at 1760 μs while a small turn-in steering command is applied. The rapid acceleration saturates the rear tires, and the steering starts the yaw rotation. Control switches to the NMPC once the yaw rate, sideslip, speed, and rear slip σ_R have all stayed above preset thresholds for a short time, which catches the drift after it develops but before the vehicle spins out. At the switch, the NMPC references are initialized to the commands currently applied, and the reference circle is placed tangent to the measured velocity, so that $e = \Delta\phi = 0$. Because the kick and the switching thresholds are the same in every experiment, all controllers take over from similar initial conditions.

All experiments use the same target drift and controller settings. The weights are designed via Bryson's rule [19, Ch. 20], $Q = \Delta t\, \mathrm{diag}(q_i/\bar{x}_i^2)$ and $R = \Delta t\, \mathrm{diag}(q_j/\bar{\nu}_j^2)$, where $\bar{x}_i$ and $\bar{\nu}_j$ are the maximum acceptable deviations and

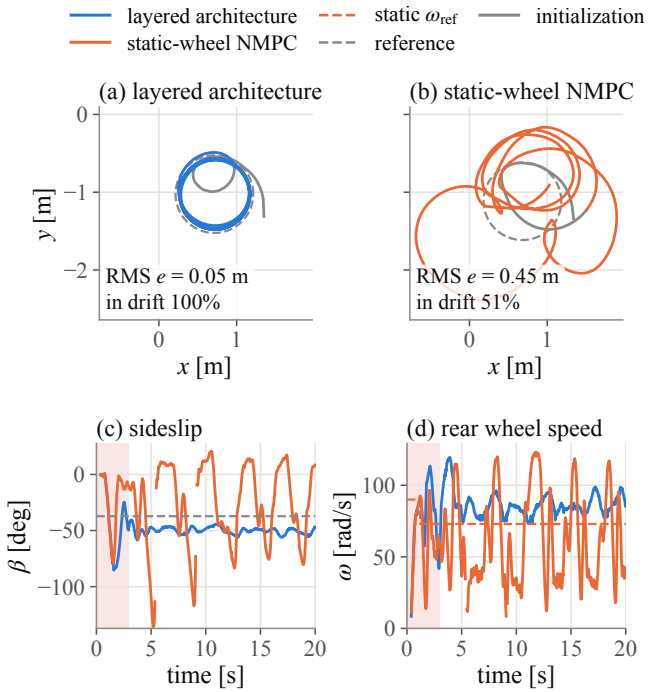


Fig. 5: Layered architecture versus static-wheel NMPC:(a, b) trajectories with RMS path error and time in drift ($|\beta| > 20°$) after initialization; (c) sideslip and (d) rear-wheel speed.

q sets the relative priority:

	e	$\Delta\phi$	β	r	V	δ	ω	δ_{ref}	ω_{ref}	$\dot{\delta}_{\text{ref}}$	$\dot{\omega}_{\text{ref}}$
$\bar{x}$, $\bar{\nu}$	0.1	10°	10°	0.4	0.35	12°	18	12°	18	300°	120
q	1	1	12	3	10	0.5	0.5	1	1	200	4

with lengths in m, speeds in m/s, rates in rad/s or °/s, and $\dot{\omega}_{\text{ref}}$ in rad/s^2. The path errors have low priority so the controller does not give up the drift to track the circle, and the high steering-rate priority suppresses oscillations caused by servo lag and loop delay. The final term in the objective function of (15) uses $w_\sigma = 150$ and $\sigma_{\min} = 0.6$. The actuator lags are $\tau_s = 0.045$ s and $\tau_\omega = 0.05$ s, and the onboard PI gains are $K_p = 6\ \mu\text{s/(rad/s)}$ and $K_i = 0.2\ \mu\text{s/rad}$.

B. Results

We compare the proposed layered architecture against a static-wheel NMPC that neglects the wheel speed dynamics. The static-wheel NMPC uses the same chassis and tire model, cost, horizon, constraints, and kick-start initialization, but removes ω from the state and commands steering and throttle PWM directly. The rear wheel speed in its prediction is given instantaneously by a linear map identified from the same driving data, $\omega = 0.2271\,(\text{PWM} - 1500) + 26.10$ rad/s, which has no dynamics and matches the measured wheel speed with an RMSE of 10.1 rad/s over all samples.

Each controller was tested in five back-to-back trials from the same kick-start, with the battery fully charged before each set. The layered architecture initiated and stabilized the drift in all five trials. After handoff, the vehicle converged to a sustained donut within 5 s and held it for the rest of the

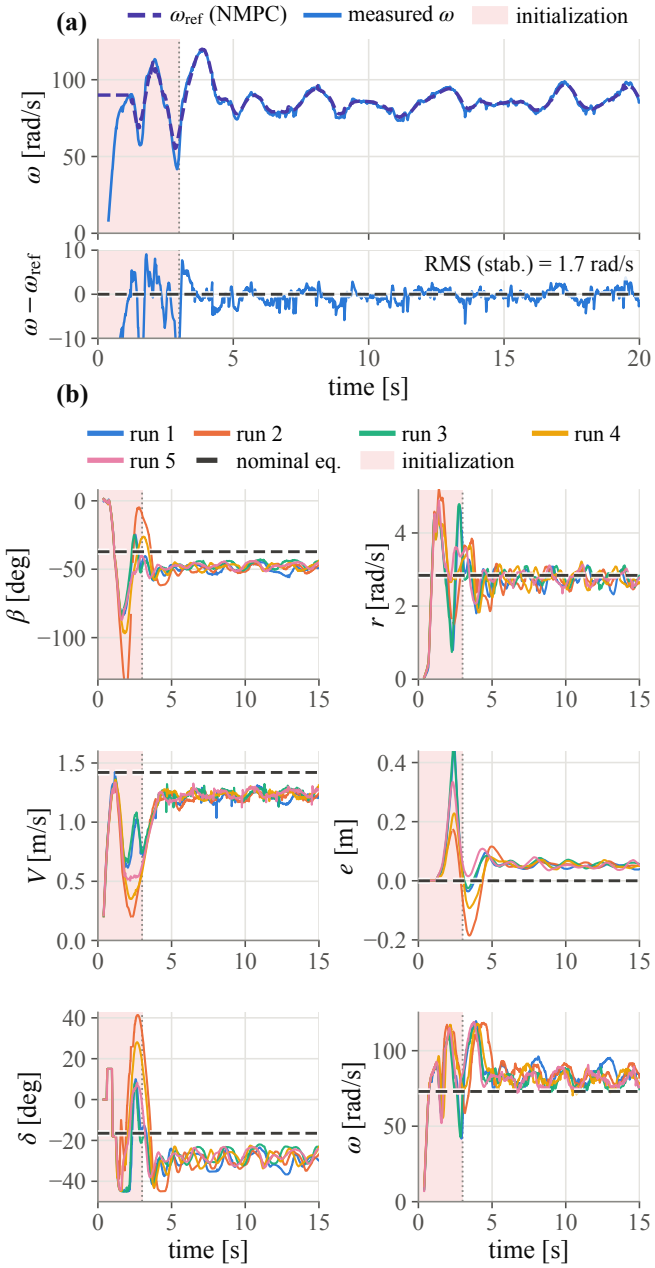


Fig. 6: Layered architecture over five drift experiments. (a) Wheel-speed reference tracking in one run (RMS error 1.7 rad/s after initialization). (b) States and inputs of all runs; dashed lines mark the nominal equilibrium, shading the 3 s initialization.

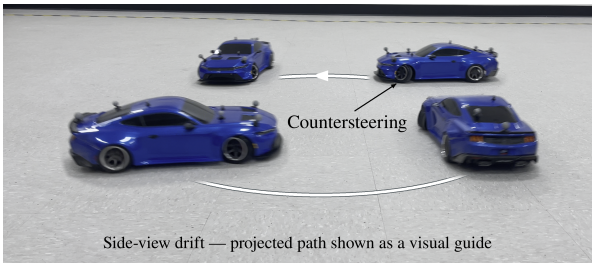


Fig. 7: Composite snapshots of a steady donut drift under the layered architecture, showing front-wheel countersteering. The arc indicates the direction of travel.

TABLE II: Small-scale vs. full-scale drifting platforms (donut).

	Ours (1/10th)	Supra [20]	LC500 [20]
Mass m [kg]	1.96	2048	1476
Wheelbase L [m]	0.257	2.867	2.448
Rear load fraction a/L	0.52	0.47	0.51
Drift speed V [m/s]	1.42	9.0	7.5
Sideslip β [deg]	-37.2	-40	-34
Rear slip ratio κ_R	1.26	1.13	1.04
Yaw rate r [rad/s]	2.84	0.90	1.07

run. The static-wheel NMPC failed in all five trials. After handoff, the vehicle repeatedly lost the drift, with sideslip swinging to positive values and yaw rate dropping toward zero, and it never settled into a steady donut. Because the two controllers share the chassis and tire models, cost, horizon, constraints and initialization, and differ only in how wheel speed is handled, this result indicates that neglecting wheel-speed dynamics and feedback is enough to lose the drift, even at 1/10th-scale. Videos of all trials are available online[1].

C. Comparison With Full-Scale Drifting Platforms

Table II compares our experimental vehicle with the Toyota Supra and Lexus LC500 drifted autonomously in the recent work [20], using the steady first-gear donut of each full-scale vehicle and our reference equilibrium (18).

The mass distribution is comparable ($a/L = 0.52$ vs. 0.47–0.51). The drift itself falls in the same regime: the sideslip (-37° vs. -40° and -34°) and the rear slip ratio (1.26 vs. 1.13 and 1.04) are close to the full-scale values. The small-scale platform therefore reproduces the deep rear-slip, counter-steered equilibrium of full-scale drifting at a fraction of the cost and risk.

VI. Discussion and Future Work

In this paper, we investigated whether the rear wheel speed dynamics can be neglected on a 1/10th-scale vehicle. We found that on our platform, they cannot. The identified drivetrain time constant is $\tau_d = 0.37$ s which is comparable to the time constant of the unstable drift mode (≈ 0.52 s) so a controller which assumes the wheel responds instantly to PWM commands does not account for this lag. In our experiments, the static-wheel NMPC makes this assumption and lost the drift in all five trials. On the other hand, our approach based on the layered control architecture sustained the drift in all five trials, tracking the wheel speed reference with a RMSE of ≈ 1.7 rad/s in a representative run.

Our second takeaway is more practical. The hobby-grade ESC offers neither torque nor current feedback, yet a low-order model identified from a small amount of driving data seems to be sufficient to enable reliable tracking of wheel speed references when combined with a simple PI controller. Our approach needs no extra onboard sensors beyond the wheel encoder and inertial sensor, which should make it straightforward to apply to other hobby-grade platforms.

Future work will investigate closed-loop approaches to drift initiation and transitions between different transient drifting maneuvers. More generally, repeating this study on vehicles at different scales would help test the effect of the wheel speed dynamics as a function of scale.

[1] `https://bit.ly/RC_Drift_Videos`

References

[1] T. Zhao, E. Yurtsever, and G. Rizzoni, "Justifying emergency drift control for automated vehicles," in *IFAC Symposium on Advances in Automotive Control*, vol. 55, no. 24, 2022, pp. 141–148.

[2] N. A. Spielberg, M. Brown, and J. C. Gerdes, "Neural network model predictive motion control applied to automated driving with unknown friction," *IEEE Transactions on Control Systems Technology*, vol. 30, no. 5, pp. 1934–1945, 2021.

[3] J. Y. Goh, T. Goel, and J. C. Gerdes, "Toward automated vehicle control beyond the stability limits: Drifting along a general path," *Journal of Dynamic Systems, Measurement, and Control*, vol. 142, no. 2, p. 021004, 2020.

[4] J. Y. M. Goh, "Automated vehicle control beyond the stability limits," Ph.D. dissertation, Stanford University, 2019.

[5] F. Djeumou, J. Y. M. Goh, U. Topcu, and A. Balachandran, "Autonomous drifting with 3 minutes of data via learned tire models," in *IEEE Conference on Robotics and Automation*, 2023, pp. 968–974.

[6] N. Ding, M. Thompson, J. Dallas, J. Y. M. Goh, and J. Subosits, "Drifting with unknown tires: Learning vehicle models online with neural networks and model predictive control," in *IEEE Intelligent Vehicles Symposium*, 2024, pp. 2545–2552.

[7] T. P. Weber and J. C. Gerdes, "Modeling and control for dynamic drifting trajectories," *IEEE Transactions on Intelligent Vehicles*, vol. 9, no. 2, pp. 3731–3741, 2024.

[8] A. Davydov, F. Djeumou, M. Greiff, M. Suminaka, M. Thompson, J. Subosits, and T. Lew, "First, learn what you don't know: Active information gathering for driving at the limits of handling," *IEEE Robotics and Automation Letters*, vol. 10, no. 11, pp. 11 379–11 386, 2025.

[9] M. Cutler and J. P. How, "Autonomous drifting using simulation-aided reinforcement learning," in *IEEE Conference on Robotics and Automation*, 2016, pp. 5442–5448.

[10] F. Zhang, J. Gonzales, K. Li, and F. Borrelli, "Autonomous drift cornering with mixed open-loop and closed-loop control," in *IFAC World Congress*, 2017, pp. 1916–1922.

[11] M. Baur and L. Bascetta, "An experimentally validated LQR approach to autonomous drifting stabilization," in *European Control Conference*, 2019, pp. 732–737.

[12] B. Yang, Y. Lu, X. Yang, and Y. Mo, "A hierarchical control framework for drift maneuvering of autonomous vehicles," in *International Conference on Robotics and Automation*, 2022, pp. 1387–1393.

[13] Y. Lu, B. Yang, J. Li, Y. Zhou, H. Chen, and Y. Mo, "Consecutive inertia drift of autonomous RC car via primitive-based planning and data-driven control," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2023, pp. 4835–4840.

[14] J. Y. Goh and J. C. Gerdes, "Simultaneous stabilization and tracking of basic automobile drifting trajectories," in *IEEE Intelligent Vehicles Symposium*, 2016, pp. 597–602.

[15] J. Y. M. Goh, M. Thompson, J. Dallas, and A. Balachandran, "Beyond the stable handling limits: Nonlinear model predictive control for highly transient autonomous drifting," *Vehicle System Dynamics*, pp. 1–24, 2024.

[16] H. B. Pacejka, *Tire and Vehicle Dynamics*, 3rd ed. Butterworth-Heinemann, 2012.

[17] Traxxas, *4-Tec Drift Ford Mustang Owner's Manual (Model 105237-4)*, Traxxas, McKinney, TX, USA, 2024, document 105237-4-OM-EN-R00.

[18] M. Diehl, H. G. Bock, and J. P. Schlöder, "A real-time iteration scheme for nonlinear optimization in optimal feedback control," *SIAM Journal on Control and Optimization*, vol. 43, no. 5, pp. 1714–1736, 2005.

[19] J. P. Hespanha, *Linear Systems Theory*, 2nd ed. Princeton University Press, 2018.

[20] F. Djeumou, T. J. Lew, N. Ding, M. Thompson, M. Suminaka, M. Greiff, and J. Subosits, "One model to drift them all: Physics-informed conditional diffusion model for driving at the limits," in *Conference on Robot Learning*, 2024.