

Drifting in the Future: Stabilizing Path Following Drifting on High-Latency Vehicle Systems

Frederik Werner, Till Heintzenberg, Markus Lienkamp, Johannes Betz

Abstract—Autonomously controlling and handling a vehicle at and beyond its stability limit is a mathematically and computationally demanding task. Prior demonstrations of automated drifting have been limited to research platforms with instantaneous torque delivery and independently actuated wheels, leaving their applicability to production vehicles with actuator latencies and mechanically coupled axles uncertain. To overcome these issues, we design a predictor to compensate for powertrain delays, develop a revised control formulation to accommodate higher actuation latencies as well as a differential coupling on the driven axle, and introduce brake-based velocity stabilization. This paper presents the controller framework, the model extensions, and real-world experimental results. We observe that our controller enables a production sports car with a combustion engine to robustly sustain circular and figure-eight drifts, limiting lateral error to 1.1 m and sideslip overshoot to 0.06 rad despite actuator delays exceeding 250 ms, while mitigating oscillations and maintaining stable path and sideslip tracking. In conclusion, our results establish that autonomous drifting is feasible on production-ready vehicles, opening pathways to advanced safety systems capable of stabilizing cars in scenarios where traditional control fails.

I. INTRODUCTION

Conventional safety systems like Electronic Stability Control (ESC) ensure autonomous vehicles operate within stable handling limits. These systems are designed to prevent excursions into open-loop unstable regimes, ensuring controllability for the average driver. However, to surpass expert human drivers in safety and performance, autonomous vehicles must master the full range of vehicle dynamics, especially in unexpected situations [1]. Emergency maneuvers or sudden changes in road friction may push the vehicle beyond traditional stability boundaries, a domain where expert drivers use drifting techniques for control [2].

While earlier research [3]–[7] has demonstrated the viability of autonomous drifting along a path, its validation was performed on a purpose-built electric research vehicle with near-instantaneous torque delivery and independently actuated rear wheels [8]. We found that direct application of this framework to a production vehicle is not possible without adaptations. Vehicles equipped with an internal combustion engine (ICE) present a more complex problem due to actuator delays in the powertrain and mechanical coupling of the rear wheels through a differential, which prohibits independent wheel speed control. This work addresses these challenges by adapting and extending a validated control framework for use on a high-performance, ICE-powered

F. Werner, J. Betz are with the Professorship of Autonomous Vehicle Systems, TUM School of Engineering and Design, Technical University of Munich, 85748 Garching, Germany; Munich Institute of Robotics and Machine Intelligence (MIRMI), corresponding author: frederik.werner@tum.de



Fig. 1: Demonstration of our autonomous drifting control algorithm in a production vehicle performing figure-eight drifting on a surface with mixed wet, damp, and dry conditions. [9]

production sports car. We demonstrate our novel controller on a real-world vehicle and show that we can successfully initiate, sustain, and transition between drifts while accurately tracking dynamic paths, including circles and figure-eight maneuvers. In doing so, this research takes a step toward transferring drift control from research platforms to real-world production vehicles, supporting its potential for future safety and performance systems.

This work extends a previously validated drift control framework [8] to production-ready high-latency vehicles, introducing several novel elements that make autonomous drifting feasible on a conventional ICE sports car. Our main contributions are:

- Design of a state prediction module that compensates for ICE powertrain delays and phase-aligns the steering and torque dynamics, preventing oscillations.
- Development of a revised drift control formulation for a vehicle with high actuation latencies and without independent wheel-torque actuation.
- Integration of brake-based velocity stabilization to counter overshoots during drift transitions and improve path tracking accuracy.

Videos of our experiments are available at: https://github.com/TUM-AVS/ICRA2026_Drifting_in_the_future

II. RELATED WORK

Research on autonomous drifting [7] initially focused on open-loop or equilibrium stabilization using nested control structures [10]–[14]. These methods demonstrated drift stabilization at fixed points but lacked path-following capabilities.

Bárdos et al. [15] proposed a linear-quadratic regulator around a known equilibrium, but their approach did not generalize to varying curvature. Similarly, the model-free strategy by Joa et al. [4] achieved equilibrium control using only measurable signals, omitting tire models and friction estimation.

Goh and Gerdes [8] developed a control architecture capable of maintaining a drift while performing path-following. Their framework imposed desired dynamics on lateral and sideslip errors and used model inversion to compute control inputs. Later works by Goh *et al.* included a nonlinear model predictive controller (MPC) for racetrack drifting [16] and the definition of recoverability envelopes [17], as well as the use of brakes to tackle the under-actuation problem for the stabilization of drifts along a path [18]. His work on MPC showed the capability to handle ICE drivetrains as well, but drawbacks include high computational requirements and sensitivity to model mismatch. In addition, multiple authors focused on MPC variations for autonomous drifting [19], [20] but demonstrated this only in simulations. Recently, Meyjer *et al.* developed an MPC based controller for a production BMW M3 vehicle, but only as a driver aid, not in a fully autonomous setup [21]. Djeumou *et al.* investigated drifting in both specialized drift vehicles and production vehicles using an MPC based on a physics-informed conditional diffusion model [22], [23]. Their approach achieved remarkable improvements in generalization capability while requiring comparatively little training data. However, reliance on a high-performance CPU for execution indicates substantial computational demands.

Learning-based approaches [5], [24]–[29] have also been explored, where agents are trained in simulation environments to acquire drifting skills. These controllers show promising flexibility and adaptability, being able to discover non-intuitive control strategies and operate without requiring detailed vehicle or tire models. However, most published results remain confined to simulation studies, with little evidence of reliable real-world deployment due to challenges in sim-to-real transfer, robustness to model mismatch, and the need for extensive training data.

A. Baseline Controller

Goh *et al.* [8], [30] introduced a control framework for autonomous drifting along trajectories. This work was validated on the MARTY DeLorean test vehicle with independently actuated, electrically powered rear wheels. Their research demonstrated robust sideslip and path tracking during figure-eight drifts with a maximum lateral error of 1.5 m, a maximum heading error of 0.1 rad and a maximum sideslip overshoot of 5° [17]. In addition, the framework is computationally efficient, which makes it a promising candidate for adaptation to production-ready software. We are building on the control framework of Goh *et al.* [8], [30] for path and sideslip tracking which consists of four key modules (Fig. 2):

1) *Imposed Error Dynamics*: Stable dynamics are imposed on the lateral path error and the sideslip tracking

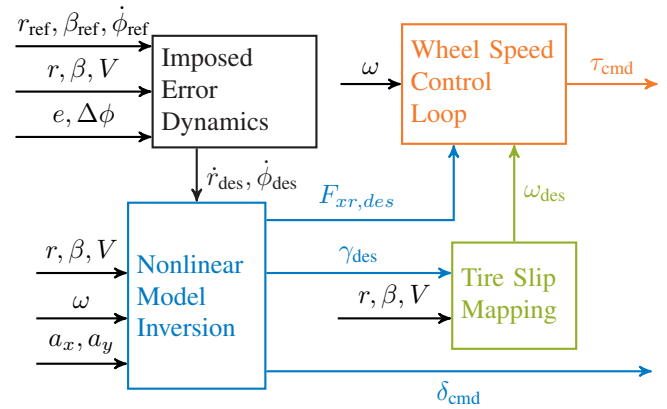


Fig. 2: Internal structure of the drift controller adapted from Goh *et al.* [8]. The controller consists of four main modules explained in Sec. II-A and adapted in Sec. III-A.

error, yielding a desired course angle rate $\dot{\phi}_{des}$ and yaw acceleration \dot{r}_{des} .

2) *Nonlinear Model Inversion*: This module inverts a single-track vehicle model to calculate the steering angle δ_{cmd} and rear tire thrust angle γ_{des} required to produce the desired yaw and course angle rates (\dot{r}_{des} , $\dot{\phi}_{des}$). To handle actuator limits, this calculation is constrained to a feasible set of achievable dynamics, labeled as the *tangent space*. When a desired state derivative is unachievable, the *yaw stability preserving projection* (YSPP) prioritizes sideslip tracking over path tracking to prevent the vehicle from losing control authority and spinning out [17].

A key model assumption described by Eq. 1 considers continuously saturated rear wheels. Because of this assumption and given a constant friction value μ_r , the magnitude of the rear tire force is constant and can be represented by the rear thrust angle γ , which is the angle of the rear tire force vector.

$$F_{xr} = \sqrt{\mu_r^2 F_{zr}^2 - F_{yr}^2} \quad (1)$$

3) *Tire Slip Mapping*: The Tire Slip Mapping Module converts the rear thrust angle into the required wheel speed ω_{des} .

4) *Wheel Speed Control Loop*: Torque commands τ_{cmd} are generated to track the desired wheel speed, taking into account wheel inertia and tire forces. Factoring in wheel and drivetrain inertia accelerates the tracking of the desired rear longitudinal force $F_{xr,des}$. Experiments by Goh *et al.* showed that wheel speed control reduces yaw oscillations and improves path tracking compared to direct calculation of $\tau_{cmd} = R F_{xr,des}$ via the tire rolling radius R [8].

III. METHOD

Our target platform is a Mercedes-AMG GT 63 S 4-Door shown in Fig. 3, equipped with production actuators that accept torque and steering angle commands via the CAN interface, enabling autonomous operation. The vehicle state

is estimated using an Oxford OXTS RT3000 differential INS unit operating at 100 Hz.



Fig. 3: The Mercedes-AMG GT 63 S research vehicle during a drift maneuver. [9]

Compared to the MARTY platform, stabilizing drifts on the AMG poses additional challenges. The primary challenges are the actuator latency of the ICE powertrain and the coupled rear wheels, which prevent independent control of the wheel speed. The following sections outline our adaptations to address these limitations.

A. Adaptations to the Baseline Controller and Integration into a Closed-Loop Simulation

We integrated the baseline controller into our closed-loop simulation framework comprising four main modules, shown in Fig. 4:

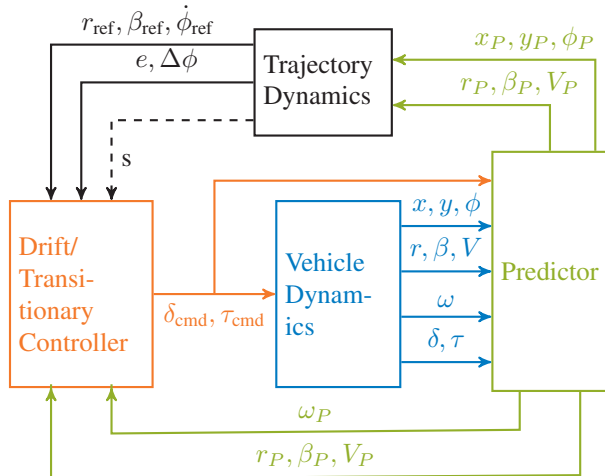


Fig. 4: Overview of the closed-loop control and simulation architecture.

- **Drift/Transitional Controller:** Contains the described drift control algorithm as well as a simple, Stanley-style controller for path tracking in transitional phases like starting from a standstill or return-to-start.
- **Vehicle Dynamics:** A two-track model with a nonlinear tire model is used. The brake, steering and engine subsystems, which are fitted to track data, model the dynamic response of these systems using delay, PT2 and rate-limiting dynamics. The simulation is performed using an ode45 solver with variable step, limited to a maximum of 1 ms. Additionally, we model friction variations and vehicle parameter uncertainty.

- **Predictor:** Estimates the future vehicle state through model forward integration, used as the actual state of the vehicle in the error calculation of the controller.
- **Trajectory Dynamics:** Path matching and reference state interpolation based on an offline-generated trajectory.

The following subsections describe modifications to the original control approach which can be classified as robustification and a more general implementation.

1) *Imposed Error Dynamics:* We made several simplifications for the sake of stability. First, instead of $\dot{\phi}_{\text{ref}} = \kappa_{\text{ref}} \left(\frac{V \cos \Delta \phi}{1 - \kappa_{\text{ref}} e} \right)$, the simplified expression $\dot{\phi}_{\text{ref}} = \kappa_{\text{ref}} V$ is used, where e denotes the lateral error, $\Delta \phi$ the course angle error and κ_{ref} the reference curvature. When path tracking errors become too large, the original formulation can introduce instability.

Second, \dot{r}_{syn} is simplified to $\dot{r}_{\text{syn}} = \dot{r}_{\text{ref}}$. In the derivation of Goh *et al.* there are additional terms that are acknowledged to be negligible [8, p.3].

2) *Tire Slip Mapping:* Since no independent wheel speed control is possible, $\gamma_{\text{RL}} = \gamma_{\text{RR}} = \gamma_{\text{des}}$ cannot be achieved. This is not a problem as long as the thrust angle is achieved for the rear axle as a whole. Eq. 2 defines the computation of the desired wheel speed, where t denotes the track width, and $F_{z,RL}$ and $F_{z,RR}$ represent the vertical loads on the rear left and right tires. Vertical tire loads $F_{z,RL}$ and $F_{z,RR}$ are estimated in real-time using a simple load-transfer model using longitudinal and lateral acceleration (a_x, a_y), vehicle mass, track width, and center-of-gravity height. The third term ensures that the thrust angle is achieved even for cases of uneven load distribution. For instance, if $F_{z,RL}$ becomes zero and the entire load of the rear axle is carried by the right-hand tire, the equation becomes equivalent to the formulation of Goh *et al.* for $\omega_{\text{des},RR}$ [8, p.6].

$$\omega_{\text{des}} = \frac{V \cos \beta}{R} - \frac{V \sin \beta - b r}{R \tan \gamma_{\text{des}}} + \frac{t r}{R} \left(\frac{F_{z,RR}}{F_{z,RL} + F_{z,RR}} - 0.5 \right) \quad (2)$$

3) *Nonlinear Model Inversion (NMI):* Our NMI implementation introduces three key modifications to the baseline for improved accuracy and robustness.

First, to better match our vehicle's behavior, we replaced the Fiala-Brush tire model with a more accurate Pacejka MF5 model for computing the tire forces.

Second, we solve the inversion by formulating an explicit system of nonlinear equations. The system is derived from single-track model equations [8, p.3], where the rear lateral force F_{yr}^I and the rear longitudinal force F_{xr}^{III} are expressed as functions of the steering angle δ . The symbols I_z and m denote the yaw inertia and mass; a and b are the distances from the center of gravity to the front and rear axles; V is the velocity; and β is the sideslip angle. This formulation yields a single equation in terms of δ (Eq. 5), which is solved using

the Newton-Raphson method.

$$F_{yr}^I = -\frac{I_z \dot{r}_{des} - a F_{yf} \cos(\delta)}{b} \quad (3)$$

$$F_{xr}^{III} = \sqrt{\mu_r^2 F_{zr}^2 - F_{yr}^I{}^2} \quad (4)$$

$$f(\delta) = F_{yf} \cos(\beta - \delta) + F_{yr}^I \cos(\beta) - F_{xr}^{III} \sin(\beta) - V m \dot{\phi}_{des} \stackrel{!}{=} 0 \quad (5)$$

If the controller's demand for $(\dot{r}_{des}, \dot{\phi}_{des})$ lies outside the feasible tangent space, we project it onto the space's boundary. The point on the boundary closest to the original demand is chosen, ensuring that commands remain physically achievable while prioritizing sideslip stability. Figure 5 visualizes the tangent space for a certain vehicle state and shows the projection for two different controller demands.

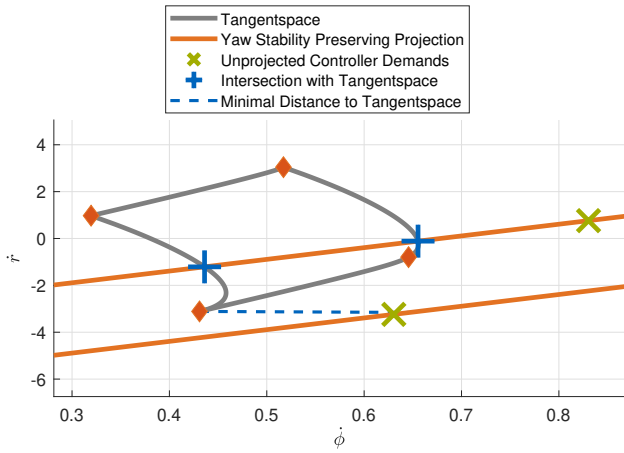


Fig. 5: Projection of controller demands for two cases, where $r = 0.6 \text{ rad s}^{-1}$, $\beta = -0.7 \text{ rad}$, $V = 13.8 \text{ m s}^{-1}$. The controller demand to the upper right gets projected onto the boundary of the tangent space via the yaw stability preserving projection (YSPP). The lower demand gets calculated via the closest distance of its YSPP to the tangent space.

Finally, we introduce a constraint to prevent an intermittently observed drop in sideslip caused by a slow torque response in situations with high rates of $F_{xr,des}$. To mitigate this, we limit the allowable error between the desired and actual wheel speed $\Delta\omega = \omega_{des} - \omega_{act}$, introducing an additional constraint on the rear longitudinal force ($F_{xr,des}$) within the NMI. As a result the steering angle command δ_{cmd} adapts to the slow torque and wheel speed response, stabilizing the sideslip dynamics.

B. Predictor for Actuator Latency Compensation

In contrast to the MARTY vehicle, the internal combustion engine (ICE) and steering system of the test vehicle exhibit significant actuator latencies, as shown in Sec. IV-A. These delays between a command and its execution can induce oscillations and instability, rendering the original control approach ineffective. An idealized simulation with added actuator latencies highlights this issue (Fig. 6), consistent with observations by Joa *et al.* [4].

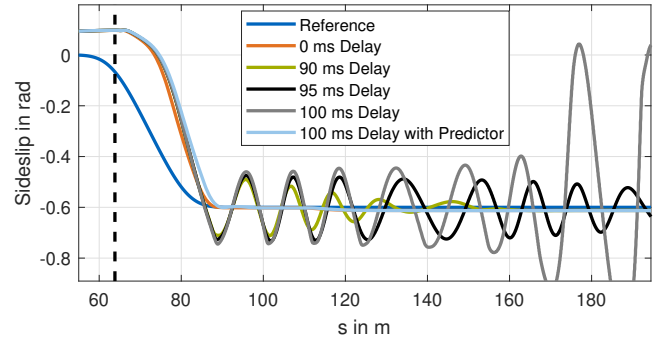


Fig. 6: Sideslip Tracking Performance with varying synthetic latencies on the actuator signals. With rising delay time oscillations start to form. At over 95 ms of actuation latency, the oscillations get catastrophic. The use of the predictor mitigates these effects.

To mitigate this, a state predictor module is implemented to compensate for delays inspired by the state space model forecast by Goh *et al.* [16]. The predictor forecasts the vehicle's state forward in time over a prediction horizon set to approximate the mean actuator latency. The drift controller uses these predicted future states for its calculations, effectively aligning its commands with the vehicle's delayed response.

The predictor uses a two-track vehicle dynamics model to propagate the current state. Its inputs are the currently measured vehicle states (e.g. position x^i, y^i , velocity V^i , yaw rate r^i , wheel speed $\omega_{present}$), actuator states (δ^i, τ^i) and the controller commands (δ_{cmd}, τ_{cmd}) from the previous time steps. By applying the previous commands to the current state, the model simulates the physical delay of the system. A pre-filter estimates the present wheel speed from its delayed measurement before this value is passed to the main predictor model. The predictor structure is shown in Fig. 7.

The dynamics of the predictor are described by the state-space model $\dot{\mathbf{X}} = f_{dyn}(\mathbf{X}, \mathbf{U})$, where the state vector is $\mathbf{X} = [r, \beta, V, \omega, x, y, \psi, \delta, \tau]^T$ and the input is $\mathbf{U} = [\delta_{cmd}, \tau_{cmd}]^T$. The primary state derivatives are given by:

$$\dot{r} = \frac{1}{I_z} \left(a F_{yf} \cos \delta - b F_{yr} + \frac{t}{2} \sin \delta (F_{y,FL} - F_{y,FR}) + \frac{t}{2} (F_{x,RR} - F_{x,RL}) \right) \quad (6)$$

$$\dot{\beta} = \frac{1}{mV} (F_{yf} \cos(\delta - \beta) + F_{yr} \cos \beta - F_{xr} \sin \beta) - r \quad (7)$$

$$\dot{V} = \frac{1}{m} (-F_{yf} \sin(\delta - \beta) + F_{yr} \sin \beta + F_{xr} \cos \beta) \quad (8)$$

The remaining derivatives describe the wheel speed ($\dot{\omega}$), kinematics ($\dot{x}, \dot{y}, \dot{\psi}$) and actuator responses ($\dot{\delta}, \dot{\tau}$). The state propagation over the prediction horizon is performed using a second-order Runge-Kutta numerical integration with a step size of 10 ms over a total horizon of 170 ms. This specific horizon was chosen through empirical ablation and yielded the best tracking and oscillation suppression performance.

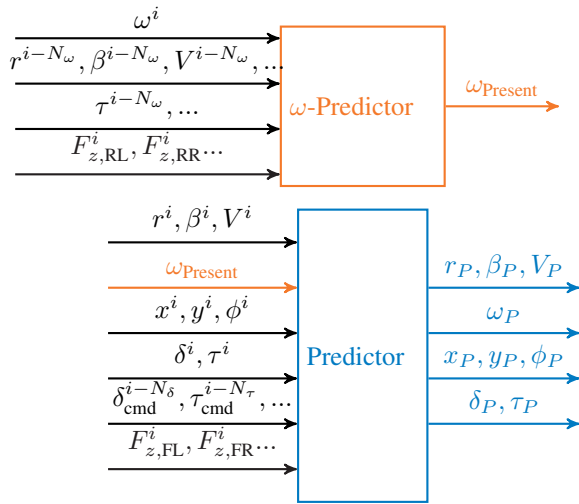


Fig. 7: The predictor structure.

The resulting performance increase due to the predictor is shown in Sec. IV-C.

C. Velocity Control via Supplementary Braking

During dynamic transitions in the figure-eight maneuver, particularly when the sideslip magnitude is low, the vehicle was observed to accelerate. This unwanted increase in speed leads to increased lateral path tracking errors. To counteract this behavior, a supplementary brake controller was introduced to regulate the vehicle's velocity.

The controller is implemented as a simple proportional controller that becomes active when the vehicle's velocity, V , surpasses a predefined target equilibrium speed, V_{eq} . The commanded brake torque, $\tau_{brk,cmd}$, is calculated according to the following control law:

$$\tau_{brk,cmd} = -k_{brk}(V - V_{eq}) \quad \text{for } V > V_{eq} \quad (9)$$

The brake torque command is applied independently and is not considered within the main nonlinear model inversion of the drift controller. Consequently, braking acts as an external disturbance that the primary controller must reject. This affects the yaw and course angle dynamics and, by introducing a longitudinal force at the front axle, reduces the peak available lateral force. To minimize this interference with the primary objectives of sideslip and path tracking, the commanded brake torque was saturated at a relatively low total maximum value of 2000 Nm. The effectiveness of this additional controller is shown in Sec. IV-D

IV. RESULTS

This chapter presents the identification of the vehicle actuator models used in the dynamics model and the predictor. We follow up with a description of our real-world experimental setup and show the effectiveness of our added predictor and velocity controller. We conclude with a comparison of our simulation and real-world experiments and compare our solution with the baseline controller.

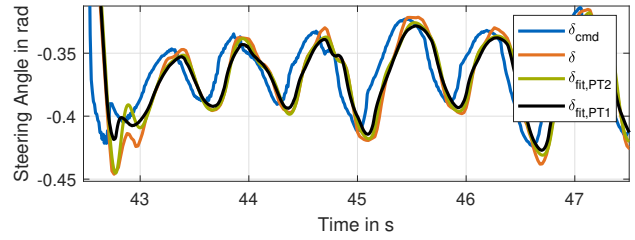


Fig. 8: Identification of the Steering Model and comparison of a PT1 and PT2 based model.

A. Simulation Development and Validation

The controller was developed in a MATLAB/Simulink simulation environment using a two-track vehicle model with a MF5 Pacejka tire model and actuator models. The reference maneuver was a symmetric figure-eight drift. Well-tuned actuator models were necessary to validate the simulation environment for controller development. Initial models proved insufficient, necessitating a data-driven refinement process focused on the steering and the engine.

A simple first-order (PT1) steering model proved inadequate. During rapidly changing steering inputs, the measured steering angle exhibited a distinct overshoot characteristic, indicative of the inertia of the physical steering rack. The steering was modeled using second-order (PT2) dynamics. The parameters for this PT2 model were identified by fitting its response to the measured data. The initial and final fit is shown in Fig. 8. The mean peak-to-peak phase delay between the actual steering angle δ and the steering angle command δ_{cmd} is 140 ms.

Modeling ICE torque delivery is challenging. An initial model with first-order (PT1) dynamics with additional response delay and an engine torque map failed to capture the state-dependent behavior of the engine. Experimental data showed that the rate of torque increase was not constant; it was slow at low torque levels, rose more sharply in a mid-range, and then slowed again at higher output.

To address this, a state-dependent engine torque model based on real-world data was developed. Instead of a single time constant, the model defines the maximum rate of change of the engine torque output τ as a function of the current engine torque. This state-dependent approach is shown in Fig 9. The mean peak-to-peak phase delay between the actual engine torque τ and the commanded engine torque τ_{cmd} is 280 ms.

B. Real-World Experiment Setup

The full control stack was deployed on our Mercedes-AMG GT63 S 4-door test platform with an onboard computer (Vector Informatik VN Control Unit), high-precision localization based on a differential inertial navigation unit (OxTS RT3000 GNSS), and integrated CAN-based actuation. The actuators are Original Equipment Manufacturer (OEM) components and can be accessed via CAN. The vehicle can be set to drift mode, where the active center differential

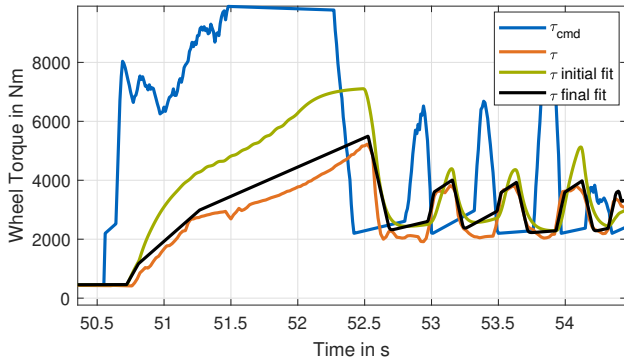


Fig. 9: Identification of the engine model. A sufficient fit could be achieved with a state-dependent parameter map.

decouples the front axle and the electronic limited-slip differential locks the rear axle. The vehicle weighs 2200 kg with a full tank and produces 470 kW and 900 Nm from its 3982 cm³ V8 Biturbo engine. The controller runs at a fixed frequency of 100 Hz.

We conduct experiments on an asphalt surface with mixed conditions, including wet, damp, and dry spots. The vehicle executes figure-eight and circular drift maneuvers. We design the reference trajectories geometrically and set a target slip angle of 0.6 rad. Higher values were possible, but were not reliably achievable above 0.65 rad, because the steering angle command would regularly exceed the steering lock, leading to spin-outs. Transitions between circles follow clothoid curves. The radius of the figure-eight circles is set to 22 m, and the centers of the circles are placed 45 m apart.

C. Predictor Module

Different actuator latencies create phase delays between steering and engine torque. These delays induce divergent oscillations in both simulation and real-world experiments, as shown on a 10 m radius circular trajectory in Fig. 10.

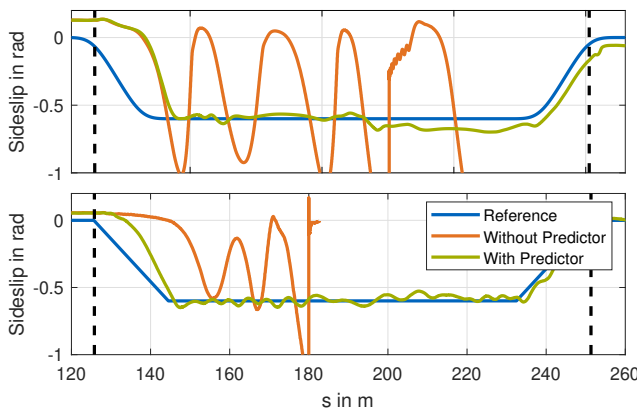


Fig. 10: Sideslip stabilization with and without predictor. Top: Closed-loop simulation. Bottom: Real vehicle.

The baseline controller without predictions is unable to maintain stability. Sideslip oscillations grow in amplitude until a spin-out occurs. Analysis of the yaw rate dynamics

shows the culprit: the delayed actuator response causes the vehicle to consistently overshoot and undershoot the yaw rate target, feeding the instability. The introduction of the state predictor module successfully resolves this instability in simulation as well as in real-world experiments. By providing the controller with a future state, its commands are effectively timed with the vehicle's delayed response.

D. Velocity Control Module

The velocity controller notably improves lateral tracking and mitigates velocity overshoots, particularly during transitions. Without velocity control, the torque command can reach high values to maintain rear tire saturation during periods of low sideslip, resulting in unwanted longitudinal acceleration as the thrust direction aligns with the trajectory.

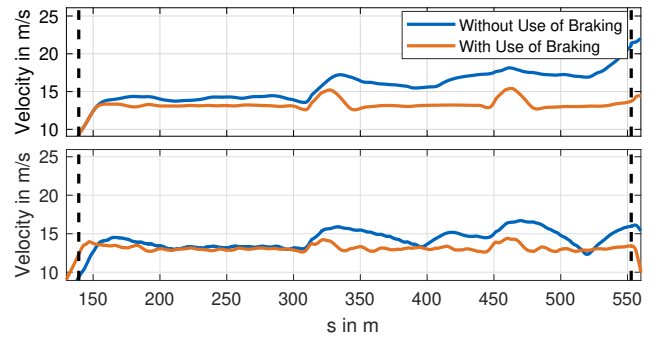


Fig. 11: Comparison of the velocity with and without the additional velocity controller. Especially in the transitions there is a notable speed increase. Top: Closed-loop simulation. Bottom: Real vehicle.

As a consequence, the vehicle enters the next circle at a higher-than-optimal speed, leading to increased lateral deviations due to the controller's prioritization of the sideslip target. This speed-up during the transition is shown in Fig. 11. The incorporation of the velocity controller resolves this issue, and the resulting improvement in velocity overshoots and path-tracking performance can be observed in Fig. 11 & 12.

E. Final Results on the Figure-Eight Trajectory

Our reformulated approach, incorporating the predictor and velocity controller, yielded strong results on our target vehicle. Table I compares the performance of the adapted baseline approach from Goh *et al.* with our reformulated and extended approach, both in simulation and on the real vehicle. Without our extensions, path tracking was unsuccessful and reliably resulted in spin-outs. By contrast, including the predictor and velocity controller enabled robust tracking of a figure-eight trajectory. Experiments repeated on the real vehicle confirmed these results and even outperformed the simulation. This improvement is largely due to the simulation model being intentionally parameterized to be more challenging, with several parameters set to worst-case values within their plausible range. This approach created a demanding simulation environment that reproduced many

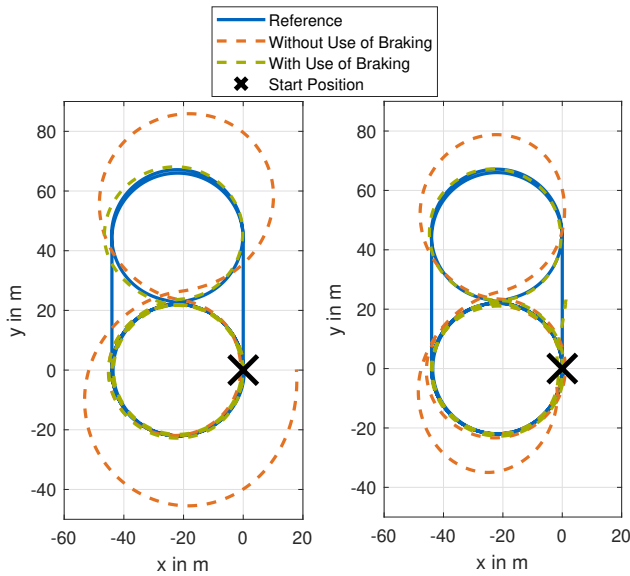


Fig. 12: Path Tracking Performance with and without Speed Control (Predictor already included). The result of the increased velocity is observable in increased lateral error. Left: Closed-loop simulation. Right: Real vehicle.

TABLE I: Evaluation of the baseline controller against our extended approach including predictor and velocity control in our closed-loop simulation, using the identified AMG vehicle model. The third column shows the performance of our extended approach on the real vehicle for comparison. The control parametrization has not been changed between column two and three.

Metric	Sim (Goh)	Sim (Ours)	Real-world (Ours)
Max Sideslip	≥ 1.5 rad	0.64 rad	0.66 rad
Max Lateral Error	≥ 10.0 m	2.4 m	1.1 m
Max Heading Error	≥ 1.5 rad	0.41 rad	0.1 rad

on-track effects, facilitating quick iterations in our development cycles.

The performance on the full figure-eight on the real vehicle is shown in Fig. 13. We compare our performance against the data from Goh *et al.* on the MARTY vehicle platform [17] in Tab. II. Overall we conclude that despite the more challenging vehicle platform we could achieve similar overall performance, even improving the performance in certain metrics like lateral deviation or sideslip overshoot. Also, our velocity variation is slightly narrower (3.0 m s^{-1} vs. 1.8 m s^{-1}). The peaks in the lateral deviation and velocity overshoots occurred during the entry, exit, and transition phases of the figure-eight, which are the most difficult to handle. However there are slight differences in our experimental setup. Due to our lower mechanical steering angle lock at 34° compared to their 38° , we set a slightly lower slip angle target and increased the corner radius. However, we do not expect those differences to invalidate the comparison.

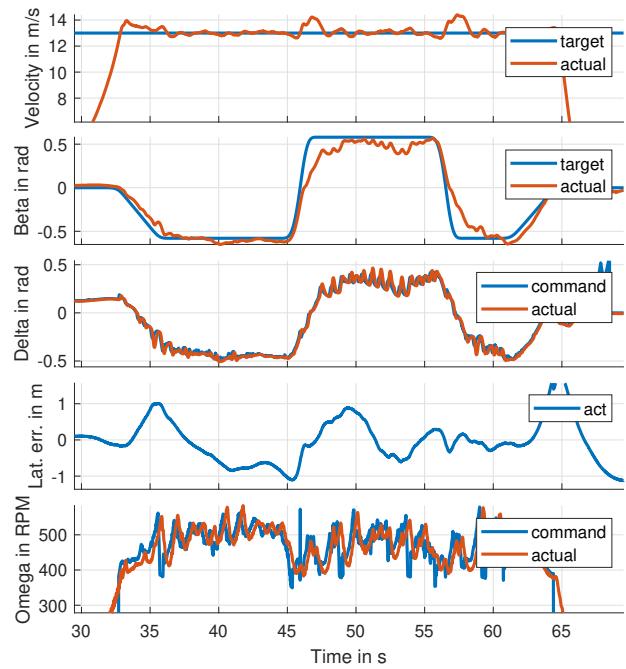


Fig. 13: Final figure-eight performance on the real Mercedes-AMG GT 63 S vehicle.

TABLE II: Comparison of the real-world results achieved by Goh *et al.* [17] on the MARTY vehicle against our results on the AMG vehicle.

Metric	Goh <i>et al.</i> [17]	Ours
Single circle radius	18 m	22 m
Max Sideslip Target	0.7 rad	0.6 rad
Max Sideslip Overshoot	0.08 rad	0.06 rad
Max Abs. Lateral Error	1.4 m	1.1 m
Max Abs. Heading Error	0.11 rad	0.1 rad
Max Abs. Yaw Rate	2 rad s^{-1}	0.95 rad s^{-1}
Min/Max Velocity	$10.8 / 13.8 \text{ m s}^{-1}$	$12.6 / 14.4 \text{ m s}^{-1}$

V. DISCUSSION & CONCLUSION

Our findings highlight that latency compensation is critical for applying drift control to production vehicles. While prior drift controllers perform well under ideal actuation, real-world ICE systems demand predictive behavior to maintain stability.

The predictor module was the most impactful enhancement. Its integration enables the system to anticipate the future vehicle response and maintain phase coherence between model and real dynamics. Our approach extends the original framework to retain drift stability on platforms with significant latency. Finally, on the AMG vehicle, our controller successfully replicated and even slightly improved the tracking performance that the baseline framework demonstrated on the MARTY platform, despite the additional challenges posed by actuator latency. In addition, we were able to avoid the need for independent wheel speed control on the rear axle. Although a nonlinear MPC could theoretically handle prediction and control in a unified framework, our

modular approach is computationally lightweight and more easily interpretable. This makes it suitable for deployment on existing automotive-grade hardware without the model tuning and computational overhead associated with MPC. Unlike learning-based systems, our architecture remains interpretable and does not require large-scale training data.

Although our empirical results validate the effectiveness of the framework, several limitations remain for future work. First, unlike linear delay compensation methods (e.g., Smith predictors), our forward-integration predictor handles the non-linearities of tire limits but lacks a formal mathematical proof of closed-loop stability. Second, our real-world testing was conducted on structured geometric trajectories (circles and figure-eights). Future research will evaluate the controller's robustness on arbitrary, varying-curvature tracks. Furthermore, we note that tire model fidelity and friction estimation still limit performance. Sensor-based adaptation or online learning could further improve robustness [23].

ACKNOWLEDGEMENTS

The authors thank Mercedes-AMG for their support and collaboration in the project. In particular, the authors acknowledge Florian Pejak and Szabolcs Dora for their contributions through discussions and technical support. The authors thank Vuk Boskovic, who supported this work through his Master Thesis. This work was supported by the Munich Institute of Robotics and Machine Intelligence (MIRMI).

REFERENCES

- [1] J. Betz, H. Zheng, A. Liniger, U. Rosolia, P. Karle, M. Behl, V. Krovi, and R. Mangharam, "Autonomous vehicles on the edge: A survey on autonomous vehicle racing," *IEEE Open Journal of Intelligent Transportation Systems*, vol. 3, pp. 458–488, 2022. [Online]. Available: <http://arxiv.org/pdf/2202.07008v1>
- [2] M. Abdulrahim, "On the dynamics of automobile drifting," in *SAE Technical Paper Series*, ser. ANNUAL. SAE International, Apr. 2006. [Online]. Available: <http://dx.doi.org/10.4271/2006-01-1019>
- [3] B. Olofsson, K. Lundahl, K. Berntorp, and L. Nielsen, "An investigation of optimal vehicle maneuvers for different road conditions," *IFAC Proceedings Volumes*, vol. 46, no. 21, pp. 66–71, 2013.
- [4] E. Joa, H. Cha, Y. Hyun, Y. Koh, K. Yi, and J. Park, "A new control approach for automated drifting in consideration of the driving characteristics of an expert human driver," *Control Engineering Practice*, vol. 96, p. 104293, 2020.
- [5] Y. Yin, S. E. Li, K. Li, J. Yang, and F. Ma, "Self-learning drift control of automated vehicles beyond handling limit after rear-end collision," *Transportation Safety and Environment*, vol. 2, no. 2, pp. 97–105, 2020.
- [6] T. P. Weber, R. K. Aggarwal, and J. C. Gerdes, "Human-inspired autonomous racing in low friction environments," *IEEE Transactions on Intelligent Vehicles*, pp. 1–14, 2024.
- [7] Y. Liu, F. Ma, X. Mei, B. Xue, J. Wu, and C. Zhang, "Autonomous drifting like professional racing drivers: A survey," *AppliedMath*, vol. 5, no. 2, p. 33, 2025.
- [8] J. Y. Goh, T. Goel, and J. Christian 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, 2020.
- [9] V. Boskovic, "Autonomous driving drift control – path-following drifting for highperformance production cars," Master's thesis, Technical University Munich, 2024.
- [10] C. Vöser, R. Y. Hindiyeh, and J. C. Gerdes, "Analysis and control of high sideslip manoeuvres," *Vehicle System Dynamics*, vol. 48, no. suppl, pp. 317–336, 2010.
- [11] R. Y. Hindiyeh and J. Christian Gerdes, "A controller framework for autonomous drifting: Design, stability, and experimental validation," *Journal of Dynamic Systems, Measurement, and Control*, vol. 136, no. 5, 2014.
- [12] E. Velenis, D. Katzourakis, E. Frazzoli, P. Tsiotras, and R. Happee, "Steady-state drifting stabilization of rwd vehicles," *Control Engineering Practice*, vol. 19, no. 11, pp. 1363–1376, 2011.
- [13] J. Gonzales, F. Zhang, K. Li, and F. Borrelli, *Autonomous drifting with onboard sensors*. Crc Press, Dec. 2016, p. 133–138.
- [14] M. Acosta, S. Kanarachos, and M. E. Fitzpatrick, "On full magv lateral dynamics exploitation: Autonomous drift control," in *2018 IEEE 15th International Workshop on Advanced Motion Control (AMC)*. IEEE, Mar. 2018. [Online]. Available: <http://dx.doi.org/10.1109/AMC.2019.8371149>
- [15] A. Bardos, A. Domina, V. Tihanyi, Z. Szalay, and L. Palkovics, "Implementation and experimental evaluation of a mimo drifting controller on a test vehicle," in *2020 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2020, pp. 1472–1478.
- [16] 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*, vol. 62, no. 10, pp. 2590–2613, 2024.
- [17] J. Goh, *Automated vehicle control beyond the stability limits*, 2019. [Online]. Available: <https://purl.stanford.edu/th820zb4174>
- [18] T. Goel, J. Y. Goh, and J. C. Gerdes, "Opening new dimensions: Vehicle motion planning and control using brakes while drifting," in *2020 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2020, pp. 560–565.
- [19] H. Dong, H. Yu, and J. Xi, "Real-time model predictive control for simultaneous drift and trajectory tracking of autonomous vehicles," in *2022 6th CAA International Conference on Vehicular Control and Intelligence (CVCI)*. IEEE, Oct. 2022, p. 1–6. [Online]. Available: <http://dx.doi.org/10.1109/CVCI56766.2022.9964790>
- [20] P. Stano, D. Tavernini, U. Montanaro, M. Tufo, G. Fiengo, L. Novella, and A. Sorniotti, "Enhanced active safety through integrated autonomous drifting and direct yaw moment control via nonlinear model predictive control," *IEEE Transactions on Intelligent Vehicles*, vol. 9, no. 2, p. 4172–4190, Feb. 2024. [Online]. Available: <http://dx.doi.org/10.1109/TIV.2023.3340992>
- [21] S. Meijer, A. Bertipaglia, and B. Shyrokaou, "A nonlinear model predictive control for automated drifting with a standard passenger vehicle," 2024. [Online]. Available: <https://arxiv.org/abs/2405.10859>
- [22] 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 *8th Annual Conference on Robot Learning*, 2024. [Online]. Available: <https://openreview.net/forum?id=0gDbEtVrd>
- [23] F. Djeumou, J. Y. Goh, U. Topcu, and A. Balachandran, "Autonomous drifting with 3 minutes of data via learned tire models," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*, 2023, pp. 968–974.
- [24] S. H. Tóth, Z. J. Viharos, B. Árdos, and Z. Szalay, "Sim-to-real application of reinforcement learning agents for autonomous, real vehicle drifting," *Vehicles*, vol. 6, no. 2, p. 781–798, Apr. 2024. [Online]. Available: <http://dx.doi.org/10.3390/vehicles6020037>
- [25] Y. Jiang, X. Xu, X. Zhang, J. Huang, and S. Gao, "Reinforcement learning-based drift control for autonomous vehicles," in *2021 China Automation Congress (CAC)*. IEEE, 2021, pp. 4508–4513.
- [26] B. Zhou, C. Hu, J. Zeng, Z. Li, J. Betz, L. Xie, and H. Su, "Adaptive learning-based model predictive control strategy for drift vehicles," *Robotics and Autonomous Systems*, vol. 188, p. 104941, 2025.
- [27] B. Zhou, Z. Li, L. Xie, H. Su, and J. Betz, "A learning-based planning and control framework for inertia drift vehicles," *arXiv preprint arXiv:2507.05748*, 2025.
- [28] P. Cai, X. Mei, L. Tai, Y. Sun, and M. Liu, "High-speed autonomous drifting with deep reinforcement learning," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 1247–1254, 2020.
- [29] F. Domberg, C. C. Wemmers, H. Patel, and G. Schildbach, "Deep drifting: Autonomous drifting of arbitrary trajectories using deep reinforcement learning," in *2022 International Conference on Robotics and Automation (ICRA)*, 2022, pp. 7753–7759.
- [30] J. Y. Goh and J. C. Gerdes, "Simultaneous stabilization and tracking of basic automobile drifting trajectories," in *2016 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2016, pp. 597–602.