The Simulator-in-the-Loop approach for vehicle dynamics control

Federico Dettù, Simone Formentin and Sergio Matteo Savaresi

Abstract—In vehicle dynamics control, engineering a suitable regulator is a long and costly process. The starting point is usually the design of a nominal controller based on a simple control-oriented model and its testing on a full-fledged simulator. Then, many driving hours are required during the End-of-Line (EoL) tuning phase to calibrate the controller for the real vehicle. Given the recent technological advances, in this paper we consider the pioneering perspective where the simulator can be run on-board in the electronic control unit, so as to calculate the nominal control action in real-time. In this way, it can be shown that, in the EoL phase, we only need to tune a simple compensator of the mismatch between the expected and the measured outputs. The resulting approach not only exploits the already available simulator and nominal controller and significantly simplifies the design process, but also outperforms the state-of-the-art in terms of tracking accuracy and robustness within a challenging active braking control case study.

I. INTRODUCTION

The use of Digital Twins (DTs) - combining software and physical connections to produce a faithful virtual replica of a given system - is revolutionizing state-of-the-art technological solutions in different fields [1].

In the automotive world, DTs are highly exploited for online monitoring and prognostics of vehicle components [1]. Moreover, vehicle dynamics simulators - as faithful replicas of the system - are widely employed at the mechanical design level, i.e., when selecting physical components or when assessing the differences among structural choices [2]. Instead, vehicle dynamics controls are still based on simple control-oriented models, usually capturing the key features of a single maneuver [3]. Indeed, this yields well known issues when implementing the controller on the real platform, having to deal with many unmodeled dynamics. In the industrial practice, this issue is overcome by finely adjusting the controller parameters, during the so-called End-of-Line (EoL) tuning process [4]. Nonetheless, the latter might be a time consuming and costly procedure, especially when considering complex industrial regulators.

In this paper, we take advantage of the most recent technological advances to show how a high-fidelity vehicle simulator can be used in real-time and directly embedded into the Electronic Control Unit (ECU) of the vehicle to enhance the closed-loop performance and significantly simplify the EoL tuning phase. We will denote the resulting architecture Simulator-in-the-loop (SiL) control hereafter. Doing so, the nominal control action could be provided as the one computed on the system simulator, whereas only dynamics to be controlled is the mismatch between the vehicle and the simulator, which could be handled by a simpler compensator. We remark that, even if the possibility of running a full-fledged vehicle simulator on-board may sound far in the future, it rather fits well into the present context of an ever-increasing on-board and cloud-based computing power, driven by the necessity of processing signals coming from visual sensors, such as LiDARs. Indeed, this novel approach calls for a tuning procedure for the additional compensator, which will be also treated in this paper.

In order to show the potential of such an architecture as compared to traditional vehicle control, we consider a well-known and challenging safety-critical problem, namely active braking control [6]. More specifically, we consider a high-performance sport car as the vehicle to be controlled, the latter being an interesting proving ground for paradigm shifting technologies, due to the extremely challenging and highly dynamic driving conditions. For fair benchmarking, since model predictive control (MPC) has been recently applied to such a problem obtaining unparalleled performance [7], we build up our proof-of-concept starting from that research. Finally, we wish to remark here that recent work has already shown the potential of such a technology for accurate vehicle state estimation [5]. Indeed, to the best of our knowledge, this is the first time that the SiL paradigm is applied to control systems.

The remainder of the paper is as follows. Section II formally states the problem, while Section III discusses the active braking case study. Section IV applies SiL control to the proposed case study, and Section V-B shows how the algorithm parameters can be calibrated starting from a set of closed-loop experiments. Finally, Section V compares the SiL solution to the benchmark. The paper is ended by some concluding remarks.

II. PROBLEM STATEMENT

The traditional framework for the development of vehicle dynamics control systems usually follows the three-step procedure depicted in Fig. 1 in which we have considered the case of reference following without loss of generality.
Specifically, the full design procedure is as follows.

1) A controller $C(\theta)$, fully described by a set of parameters $\theta$, is first designed based upon a control-oriented model, capturing the main dynamics of interest - e.g. a quarter-car model in case of active braking control [7]. From now on, we denote the set of parameters selected using the simple model as $\theta_0$.

2) The controller is then tested on a high-fidelity multibody simulator, usually accounting for unmodeled behaviours and nonlinearities. The initial values $\theta_0$ are then adjusted until acceptable performance is reached, leading to a new optimal selection $\theta_s$.

3) Finally, the controller is implemented on an Electronic Control Unit (ECU) and tested on the real vehicle along a few driving hours. This step usually requires a final refinement of the parameters, yielding the new vector $\theta_v$, due, e.g., to neglected phenomena or measurement noises.

The key observation behind the idea of this work is that, once $C(\theta_s)$ has been tuned and tested on the simulator, an ideal control action $\hat{u}$ and the corresponding output $\hat{y}$ become available. The latter opens the possibility to a quantum leap in the field of vehicle controls: if the simulator can be run in real-time on the ECU, the ideal $\hat{u}$ can be used as a nominal control action with no additional computations. If the simulator and the vehicle dynamics coincide, such an input could be directly applied to the real system. Indeed, if the simulator and the vehicle differ in some way, a second control loop needs to be designed accounting for the mismatch between $\hat{y}$ and the measured $y$. Let us denote the additional compensator as $C_\delta$. Doing so, the control action on the real vehicle becomes equal to $u_\delta + \hat{u}$. We denote the above described control architecture as Simulator-in-the-Loop Control (SiL-C), and depict its schematics in Fig. 2a.

The proposed architecture shows a set of interesting features as compared to the standard practice:

- if the simulator is a faithful replica of the vehicle, most of the system nonlinearity and complexity is managed by $C$, and the nominal control action does most of the job. The compensator $C_\delta$ would thus play the role of the controller of a linearized system in the neighborhood of an operating point;
- since $C$ is operated on the simulator, it has access to all its states. This opens up the possibility of designing state-feedback approaches (for the nominal control action) even when the state is not (fully) available without the need of designing suitable observers;
- the EoL tuning procedure of Fig. 1 cannot be avoided, as $C_\delta$ needs to be tuned based on the mismatch between the vehicle and the simulator. However, the design might be largely simplified, as $C_\delta$ can be selected as a simple controller with few parameters even when $C$ is a complex (possibly optimization-based) controller.
- the SiL-C approach can in principle be generalized to any vehicle dynamics problem, classical examples being longitudinal, lateral and vertical dynamics control.

### III. The case study: Active Braking Control

As a case study to illustrate the potential of the proposed approach, let us consider the problem of longitudinal dynamics control during braking, i.e., the design of an anti-blocking system (ABS) for a high-performance car. In particular, we focus on the control of the wheel slip $\lambda$ aimed to track a desired behaviour $\lambda$ so that a certain braking force can be guaranteed, and the vehicle can stop without wheel locking. In this seminal study, we will consider two instances of the same simulator (as detailed next) to play the role of the digital twin and the real car. The problem of selecting $\lambda$ is instead out of the scopes of this work and will not be discussed here.
We wish to stress that the main goal of this research is not to design a full braking control algorithm per se, but to simplify the EoL tuning problem by relying on the availability of the simulator on-board. As the nominal controller $C$ in Fig. 2a, we thus start from the existing, state-of-the-art, Model Predictive Control (MPC) approach in [7] (to be computed independently for each wheel, thus $ij = fl, fr, rl, rr$, as showed in Fig. 3).

The wheel slip at corner $ij$ is defined as

$$
\lambda_{ij} = \frac{v_{x,ij} - \omega_{ij} R_{ij}}{\max\{v_{x,ij}, \omega_{ij} R_{ij}\}},
$$

(1)

where $v_{x,ij}$ is the wheel ground contact point velocity, $\omega_{ij}$ is the wheel angular rate and $R_{ij}$ is the wheel radius [6]. We consider straight braking, i.e., we assume almost zero camber and sideslip wheel angles. Such an assumption does not affect the scope and outcomes of the research; it is in fact well known that such variations affect the friction analogously to the vertical wheel loads, and can be neglected by suitably scheduling the slip references [6].

A. The vehicle and the simulator

Both the vehicle - a sport car - and its digital twins are here modeled in the VI-Grade CarRealTime (CRT) simulation environment [8]. The CRT simulator include a 6 degree-of-freedom (DOF) object for the chassis, a 1-DOF model for the suspensions and a 1-DOF model for the wheel dynamics. A model of the electro-hydraulic brake (EHB) is also added, so as to take into account realistic actuation limits. The most critical vehicle parameters influencing the dynamics during braking [6] are given in Table I.

| $M_{tot}$ [kg] | $R_{f}^{nom}$ [m] | $R_{r}^{nom}$ [m] | $J_{f}^{nom}$ | $J_{r}^{nom}$ |
|----------------|-------------------|-------------------|---------------|---------------|
| 1612           | 0.33              | 0.35              | 1.49          |               |

$M_{tot}$ represents the total vehicle mass, encompassing sprung and unsprung masses, whereas $R_{f}^{nom}$, $R_{r}^{nom}$ represent the nominal wheel radius, for front and rear wheels respectively; note that the high-fidelity model also accounts for radius variations due to, e.g., increased wheel loads. $J_{f}^{nom}$ and $J_{r}^{nom}$ are the nominal spin inertia of front and rear wheels. $l_{f}$ and $l_{r}$ are the distances between the projection of the vehicle center-of-gravity (COG) on the ground and the front and the front and rear axles, respectively. $h$ is the COG height from the ground.

Formally, the control inputs $u$ and the set of driver commands $\xi$ are

$$
u = \begin{bmatrix} T_{f}^{cmd} & T_{r}^{cmd} & T_{rl}^{cmd} & T_{rr}^{cmd} \end{bmatrix}^t,
\xi = \begin{bmatrix} \phi_{throttle} & \phi_{brake} & \phi_{steer} & \phi_{gear} \end{bmatrix}^t,
$$

(2)

where $T_{ij}^{cmd}$ represents a torque command at wheel $ij$. Indeed, when the braking controller is active, the driver brake request is bypassed. Due to the straight braking assumption, throttle and steer commands are negligible. The variables $\xi$ simultaneously act on the simulator and on the vehicle, so the driver should be considered as an exogenous disturbance in the framework of Fig. 3.

In order to simulate a realistic difference between the the vehicle and its digital twin, we will consider, in two separate case studies, the effect of realistic measurement noises and that of unmodeled masses in the real vehicle. Specifically, since high performance cars are usually characterized by two front seats and a front trunk, we add a second concentrated mass on the passenger seat, and two unbalanced masses on the front trunk. A representation of the unmodeled loads is provided in Fig. 4 where it becomes clear how the presence of unmodelled masses changes the ratio between the distance of COG and front wheels $l_{f}$ and that of COG and rear wheels $l_{r}$, thus varying the front/rear load tranfer. The values of the additional masses are as in Table II.

| $m_{iq}$ [kg] | $m_{ip}$ [kg] | $m_{i}^{l}$ [kg] | $m_{i}^{r}$ [kg] |
|---------------|---------------|-----------------|-----------------|
| 75            | 80            | 90              | 90              |

B. Model predictive controller

The MPC in [7] is here considered as a nominal controller and is assumed to be designed before-hand, in the pre-development stage. Such a controller employs a wheel-specific predictive model of the slip dynamics

$$
\dot{\lambda}_{ij} = \frac{1 - \lambda_{ij}^\text{mpc}}{v_{x,ij}} a_{x,ij} - \frac{R_{ij}^{nom}}{J_{ij}^{nom} v_{x,ij}} T_{ij}^{act},
$$

(3)
where \( \lambda_{ij}^{\text{mpc}} \) represents the prediction of the slip at each wheel and the parameters are taken from Table I. The actuator dynamics is a second order system with the same model and parameters of (7). The model (3) is used in the MPC under the assumption of constant wheel speed and acceleration during the prediction horizon \( T_{p} \), namely

\[
\begin{align*}
    a_{x,ij} (t_0 + t) &= a_{x,ij} (t_0), \quad \forall t \in [t_0, t_0 + T_{p}] \\
    v_{x,ij} (t_0 + t) &= v_{x,ij} (t_0), \quad \forall t \in [t_0, t_0 + T_{p}].
\end{align*}
\]

(4)

Under the assumptions above - motivated by the different time scales between slip and chassis longitudinal dynamics \( \text{[6]} \) - the model in Eq. (3) becomes linear and time-invariant (LTI). Said LTI model can be then discretized and written in velocity form \( \text{[9]} \) - i.e., transforming states and inputs into their instantaneous variations; the slip tracking error is then further introduced as a state. At this point, an integral action can be easily implemented in the MPC: such formulation has been shown to be robust to constant disturbances, guaranteeing zero steady-state error. We refer the reader to \( \text{[7]} \) for more details regarding this predictive controller design.

C. Sensor model

Any controller acting on the system described in Section III-A is based upon sensor measurements. For a realistic case study, we cannot neglect the effect of the noise model on the performance. In particular, we introduce the noise affecting speed, acceleration and slip as detailed next. No noise is added to the braking torque, assuming it is fully controllable and known.

Acceleration. Longitudinal acceleration measurements at each wheel \( (a_{x,ij}) \) are seldom available on production vehicles. Chassis acceleration is then employed when considering straight braking \( \text{[7]} \). Such signal is obtained through a Inertial Measurement Unit (IMU), usually affected by high-frequency noise \( \text{[10]} \). We thus consider random gaussian noise in the acceleration measurement

\[
a_{x}^{n} = a_{x} + n_{a}, \quad n_{a} \sim WN(0, \sigma_{a}^2),
\]

(5)

where \( n_{a} = WN(\mu, \sigma^2) \) denotes a gaussian noise with expected value \( \mu \) and variance \( \sigma^2 \).

Chassis speed. The COG-referenced longitudinal speed \( v_{x} \) cannot be directly measured without high accuracy Global Positioning System (GPS) sensors, then it is usually estimated through state observers \( \text{[11]} \), and reported at each wheel via kinematic relations. Hence, we consider low-pass filtered version \( n_{lp} \) of a white noise \( n_{v_{x}} \) on the speed signal, so as to mimic the state observer dynamics

\[
v_{x}^{n} = v_{x} + n_{lp}(n_{v_{x}}), \quad n_{v_{x}} \sim WN(0, \sigma_{v_{x}}^2).
\]

(6)

Wheel speed. The angular rates are measured through incremental encoders: such sensors are well known to be affected by periodic noise \( \text{[12]} \), mostly due to unavoidable geometrical or misalignment errors in the sensor structure. The amplitude of such a noise increases depending on the rotational speed itself. We thus include a speed-scheduled sinusoidal error term

\[
\omega_{ij}^{n} = \omega_{ij} + A_{\omega} \sin (\omega_{ij} t), \quad A_{\omega} = \omega_{ij}^{0} + k_{\omega} \omega_{ij}.
\]

(7)

Wheel slip. Given the noise models defined in (6) and (7), the slip in (1) is also affected by a noise term, depending on both noisy speed and wheel rate measurements

\[
\lambda_{ij}^{n} = \lambda_{ij} + n_{ij}^{\lambda} (v_{x,ij}^{n}, \omega_{ij}^{n}).
\]

(8)

The terms \( \sigma_{v_{x}}, \ k_{\omega}, \ \omega_{ij}^{0} \) appearing in equations above are tuned so as to achieve signal-to-noise ratios (snrs) on the slip measurements which are compatible with those observed in real setups \( \text{[13]} \), namely \( \text{snr} \approx 4 \). A comparison between noiseless and noisy slip signals is given in Fig. 5.

![Fig. 5: Noisy and noiseless slip signals, during a closed-loop braking maneuver.](image)

IV. SiL-C DESIGN

Given the vehicle model and the benchmark controller described in the previous section, we now discuss the SiL-C approach to the active braking problem, see Fig. 6.

![Fig. 6: SiL-C braking control. The nominal MPC runs on the simulator in the above control loop, while a second loop is closed on the real vehicle via \( C_{s} \) to compensate for unmodelled and stochastic dynamics.](image)
A. The SiL-C block

The compensator scheme is shown in Fig. 7.

![Fig. 7: The detailed $C_\delta$ architecture.](image)

A linear regulator $R(s) = \frac{N_r(s)}{D_r(s)}$ processes the measurement error $e_{ij}$ so as to obtain the control action $T_{ij}^{cmd}$. Such a regulator is implemented in an anti-windup fashion [14], where a suitable de-saturation function $\Gamma(s)$ is employed in the scheme. The regulator is selected as a Proportional-Integral (PI) one

$$R(s) = \frac{N_r(s)}{D_r(s)} = k_p \frac{1 + sT_i}{sT_i}, \quad \Gamma(s) = 1 + sT_i.$$  

Then, it is discretized at sampling frequency $f_s = 200$ Hz via the Tustin approach. Notice that the parameterization of $R(s)$ might be different for front and rear wheels, e.g., due to the different spin inertia or radii.

Since the wheel slip dynamics highly depend on the vehicle speed (see again (3)), we include a speed-based scheduling law, defined a-priori, for the controller gain [15]:

$$k_p = \begin{cases} k_{p, f}^{nom}, & v_x \geq v_{lb} \\ k_{p, r}^{nom}, & v_{lb} \leq v_x < v_{ub} \\ k_p^{nom}, & v_x < v_{lb} \end{cases}$$

Considering different parameters for front and rear wheels, one obtains the following parameter vector, to be tuned

$$\theta_{SiL-C} = [k_{p, f}^{nom} \quad T_{i, f} \quad k_{p, r}^{nom} \quad T_{i, r}].$$  

Notice also that, in case the simulated car completes the braking maneuver before the real vehicle, e.g., due to a lower simulated mass, the reference $\lambda_{ij}$ goes to zero, and so does the nominal control $T_{ij}^{cmd}$.

This calls for the smooth switching structure in the lower loop of Fig. 7 in which we introduce the signal $\phi$, driving the de-activation of the braking control for the simulated vehicle.

As soon as the simulator ends the braking maneuver earlier than the vehicle, the reference signal $\lambda_{ij}$ is switched to a constant value $\bar{\lambda}$. Then, a second controller tracks the total torque commanded to the system $T_{ij}^{cmd} + \bar{T}_{ij}$. In this way, a smooth switching is guaranteed, from the full SiL control action to $C_\delta$ only. Such scheme de-facto implements a soft-insertion of the control action, which is a typical solution in PI control design [14], where $T_{ij}^{cmd} + \bar{T}_{ij}$ serves as a manual-mode control action. Since the accurate tuning of such a de-activation logic is not among the scopes of this research, we assume that the SiL-C is de-activated as soon as the speed hits 10 km/h.

B. Controller tuning

As $C_\delta$ is aimed to control the dynamics of the residual between the system output and the output of the best available model, no model can be used to properly tune such $C_\delta$ in model-based fashion. The design of this block must therefore rely only on measurements collected on the plant.

Many data-driven methods for tuning PI controller parameters exist, see, e.g., [16], [17], [18], [19]. However, such approaches are mainly defined for LTI systems and are not suited for the specific schemes of Fig. 7, where an additional control action is provided, coming from the nominal closed-loop. For the above reasons, we will instead employ a Bayesian Optimization (BO) rationale [20], [21]. BO deals with black-box optimization problems, where the cost function and the constraints are unknown but the values corresponding to some instances of the decision variables can be properly “measured”.

Specifically, BO relies on the assumption that the cost function, here describing the closed-loop performance with certain control parameters, can be modeled as a Gaussian process (GP). Closed-loop data can then be used to estimate a GP model of such a cost function, while a suitable acquisition function is chosen to select the next set of parameters to evaluate, by looking for a balanced exploration/exploitation trade-off. For a more accurate description of the BO procedure, the interested reader can refer to, e.g., [22].

In the following, we consider BO as a decision maker, in order to solve an optimization problem of the type

$$\min_{\theta \subseteq \Theta} J(\theta)$$

subject to $\theta \subseteq \Theta,$

whereas $\theta$ is a set of controller parameters, to be searched for within an a-priori defined set $\Theta$. Since the SiL-C goal is the control of residual dynamics between ideal and real loops, $J$ is selected as the root mean square of the average slip tracking error among the four wheels

$$J(\theta_{SiL-C}) = \left( \frac{\sum_{k=1}^{N_s} \sum_{ij = fl, fr, rl, rr} e_{ij}^2(k)}{4N_s} \right)^{0.5}$$

where the dependence of $e_{ij}$, $ij = fl, fr, rl, rr$ upon $\theta_{SiL-C}$ is dropped for the sake of brevity, and $N_s$ is the number of samples in the experiment. From a set of closed-loop experiments, one could re-calibrate the compensator parameters.

Specifically, the experiment used to train $C_\delta$ is illustrated in Fig. 8. A coasting-down phase is followed by a strong braking, yielding the braking control activation. The slip reference signal is superimposed with a pulse wave varying signal, in order to better excite the system dynamics to control.
In this section, we will show the performance of the SiL-C scheme on a test braking maneuver. As a baseline for a fair assessment of the results, we will also consider a standard EoL procedure, in which the MPC is fine tuned using the same data available for the design of $C_\delta$.

More specifically, since the aim of the EoL calibration is to obtain on the real vehicle the same performance attained on the simulator, the predictive model parameters are adjusted so as to minimize the distance between the ideal and the measured output in closed-loop. A suitable cost function to this scope is the root mean square of the wheel-averaged MPC prediction error

$$J(\theta_{MPC}) = \left( \frac{1}{4N_s} \sum_{k=1}^{N_s} \sum_{ij=fl,fr,rl,rr} (\lambda_{ij}(k) - \lambda_{ij}^{mpc}(k))^2 \right)^{0.5}$$

where $\lambda_{ij}^{mpc}$ is computed according to Eq. (3), and projected forward in time - for each time step - depending on the prediction horizon.

Three indices are employed to quantitatively assess the performance of the controllers; the first - and most important one - is the cost in Eq. (15). The second index instead represents the control effort, evaluated through the time derivative of the actuated braking torques

$$J_u = \left( \frac{1}{4N_s} \sum_{k=1}^{N_s} \sum_{ij=fl,fr,rl,rr} \left( T_{ij}^{act}(k) \right)^2 \right)^{0.5}$$

Finally, the total braking time is displayed, defined as the time passing from the braking control activation until vehicle speed hitting 10 km/h.

In what follows, we will consider two case studies. In the first one, we will assume that the only difference between the simulator and the real vehicle is the presence of measurement noise (as expressed in Section III-C). In the second scenario, we will neglect the effect of noise but we will investigate the case where the real vehicle has a different mass distribution, as illustrated in Section II-A.

A. The case of noisy measurements

The braking experiment with noisy data is displayed in Fig. 9. Only the front-left and the rear-left wheels are illustrated, as no significant differences between left-right corners exist during straight braking. Figure 9a shows the reference tracking performance, whereas Figure 9b depicts the actuated torques. As one can note, SiL-C is able to maintain a smoother tracking of the reference, in spite of the significant measurement noise. This is due to the fact that MPC-computed torques show undesirable oscillations induced by the presence of noise (neglected in the prediction model). Table III confirms what is noted in the figures: SiL-C significantly reduces both the tracking error and the control action aggressiveness, while also improving the braking time.

**TABLE III: Performance indices in the noisy vehicle validation experiment.**

| Performance indices | $J_\lambda$ [%] | $J_u$ [Nm/s] | $J_{time}$ [s] |
|---------------------|----------------|--------------|----------------|
| SiL-C               | 1.06           | 4.17         | 3.88           |
| MPC                 | 2.85           | 10.82        | 4.00           |
B. The case of additional unmodelled loads

Let us consider the case where the real vehicle - with noiseless data - is equipped with additional (unmodelled) masses. Figure 10 shows the actuated torques (upper plot) and the wheel slips (mid plot).

As one can note, the MPC controlled vehicle exhibits significant slip tracking error and oscillations during the transient. This is due to the increased mass on the front trunk, almost leading to instability when coupled with the actuator nonlinearities. Instead, the SiL-C scheme shows good performance as the reference is well tracked with a proper torque actuation. In Figure 10c, we also show the split between the contributions given by the nominal control and the SiL-C compensator. From this additional plot, it can be noted that the most significant contribution of the control action is in fact the nominal one, while $C_5$ only produces a small compensation term (thus confirming the suitability of a linear compensator for small-signal regulation).

One can also appreciate the importance of the switching architecture described in Fig. 7. In fact, after approximately 5 s, the nominal torque goes to zero, as the virtual vehicle is stopping due to the reduced mass. However, the overall control action is kept at the same level due to the second loop running in parallel (Fig. 7).

As a final remark, one might argue that feeding the MPC scheme in Fig. 3 with the ideal outputs $\tilde{\lambda}_{ij}$ instead of the piecewise references might increase performance, as the former signals are smoother. However, Fig. 11 shows that no visible improvements are achieved. This is due to the fact that the traditional controller is characterized by a single block that needs to be both suitable for the nominal case and robust to parameter variations. Instead, the two blocks building the SiL-C scheme play different roles and, while the MPC is aimed only to push the nominal loop at its limits, the residual dynamics is taken care of by the additional compensator.

The above observations are confirmed by the performance indices reported in Table IV.

As for completeness, we report in Table V also a comparison between the two strategies when both measurement noises and unmodeled masses characterize the real vehicle. As expected, the qualitative conclusions previously discussed are confirmed also in this case.
TABLE IV: Performance indices for the case where the real vehicle has a different mass configuration.

| Performance indices | $J_\lambda$ [%] | $J_u$ [Nm/s] | $J_{time}$ [s] |
|---------------------|----------------|----------|---------------|
| SI-L-C              | 0.96           | 3.05     | 3.96          |
| MPC                 | 2.00           | 7.18     | 4.01          |

TABLE V: Performance indices for the case where the real vehicle has a different mass configuration and the measurements are noisy.

| Performance indices | $J_\lambda$ [%] | $J_u$ [Nm/s] | $J_{time}$ [s] |
|---------------------|----------------|----------|---------------|
| SI-L-C              | 1.84           | 3.19     | 4.01          |
| MPC                 | 3.06           | 13.85    | 4.19          |

VI. CONCLUSIONS

In this paper, we have proposed a new approach for vehicle dynamics control, based on the use of a full-fledged simulator, to be run in-the-loop directly on the vehicle ECU in order to compute the nominal control action. The main advantage of such a configuration is that, in the EoL calibration phase, there is no need anymore to fine tune the (possibly many parameters of the) controller designed and calibrated on the simulator. Instead, as far as the simulator is an accurate software replica of the dynamics of the vehicle, a simple linear regulator can be used to compensate for the mismatch between the expected and the measured outputs. The new architecture shows promising results particularly when dealing with measurement noise and unmodeled terms and outperforms the state-of-the-art solution within an active braking control case study.

This being a preliminary work on the topic, many research questions remain open, e.g., concerning stability analysis, safe controller tuning and the generalization to different vehicle dynamics problems.

REFERENCES

[1] G. Bhatti, H. Mohan, and R. Raja Singh, “Towards the future of smart electric vehicles: Digital twin technology,” Renewable and Sustainable Energy Reviews, vol. 141, p. 110801, 2021.
[2] E. Kutluay and H. Winner, “Validation of vehicle dynamics simulation models—a review,” Vehicle System Dynamics, vol. 52, no. 2, pp. 186–200, 2014.
[3] A. Lucchini, S. Formentin, M. Corno, D. Piga, and S. M. Savaresi, “Torque vectoring for high-performance electric vehicles: An efficient mpe calibration,” IEEE Control Systems Letters, vol. 4, no. 3, pp. 725–730, 2020.
[4] M. Tanelli, G. Panzani, S. M. Savaresi, and C. Pirola, “Transmission control for power-shift agricultural tractors: Design and end-of-line automatic tuning,” Mechatronics, vol. 21, no. 1, pp. 285–297, 2011.
[5] G. Riva, S. Formentin, M. Corno, and S. M. Savaresi, “Simulator-in-the-loop state estimation for vehicle dynamics control: theory and experiments,” arXiv:2204.0253, 2022.
[6] S. M. Savaresi and M. Tanelli, Active braking control systems design for vehicles. Springer Science & Business Media, 2010.
[7] G. Riva, S. Formentin, M. Corno, and S. M. Savaresi, “Model predictive control of high-performance braking systems: a force-based approach,” IEEE Control Systems Letters, pp. 1–1, 2022.
[8] VI-Grade, “Vi-carrealtime,” https://www.vi-grade.com/en/products/vi-carrealtime/ 2022, [Online; accessed 23-Mar-2022].
[9] M. Lalo and R. Scattolini, Advanced and multivariable control. Pitagora edirice Bologna, 2014.
[10] B. A. N. Campos and J. M. S. T. Motta, “Online measuring of robot positions using inertial measurement units, sensor fusion and artificial intelligence,” IEEE Access, vol. 9, pp. 5678–5689, 2021.
[11] E. Hashemi, A. Kasaeiezadeh, S. Khosravani, A. Khajepour, N. Moschuk, and S.-K. Chen, “Estimation of longitudinal speed robust to road conditions for ground vehicles,” Vehicle System Dynamics, vol. 54, no. 8, pp. 1120–1146, 2016.
[12] G. Panzani, M. Corno, and S. M. Savaresi, “On the periodic noise affecting wheel speed measurement*,” IFAC Proceedings Volumes, vol. 45, no. 16, pp. 1695–1700, 2012, 16th IFAC Symposium on System Identification.
[13] S. Formentin, P. De Filippi, M. Corno, M. Tanelli, and S. M. Savaresi, “Data-driven design of braking control systems,” IEEE Transactions on Control Systems Technology, vol. 21, no. 1, pp. 186–193, 2013.
[14] K. J. Åström and T. Hägglund, Advanced PID Control. International Society of Automation, 2006.
[15] T. Johansen, I. Petersen, J. Kalkkuhl, and J. Ludemann, “Gain-scheduled wheel slip control in automotive brake systems,” IEEE Transactions on Control Systems Technology, vol. 11, no. 6, pp. 799–811, 2003.
[16] T. Yamamoto, K. Takao, and T. Yamada, “Design of a data-driven pid controller,” IEEE Transactions on Control Systems Technology, vol. 17, no. 1, pp. 29–39, 2008.
[17] S. Formentin, S. M. Savaresi, and L. Del Re, “Non-iterative direct data-driven controller tuning for multivariable systems: theory and application,” IET control theory & applications, vol. 6, no. 9, pp. 1250–1257, 2012.
[18] S. Formentin, K. Van Heusden, and A. Karimi, “A comparison of model-based and data-driven controller tuning,” International Journal of Adaptive Control and Signal Processing, vol. 28, no. 10, pp. 882–897, 2014.
[19] S. Formentin, M. C. Campi, A. Caré, and S. M. Savaresi, “Deterministic continuous-time virtual reference feedback tuning (vrft) with application to pid design,” Systems & Control Letters, vol. 127, pp. 25–34, 2019.
[20] M. Khosravi, C. Koenig, M. Maier, R. S. Smith, J. Lygeros, and A. Rupenyan, “Safety-aware cascade controller tuning using constrained bayesian optimization,” IEEE Transactions on Industrial Electronics, pp. 1–1, 2022.
[21] M. Neumann-Brosig, A. Marco, D. Schwarzmann, and S. Trimpe, “Data-efficient autotuning with bayesian optimization: An industrial control study,” IEEE Transactions on Control Systems Technology, vol. 28, no. 3, pp. 730–740, 2020.
[22] P. I. Frazier, “A tutorial on bayesian optimization,” arXiv preprint arXiv:1807.02811, 2018.