A Model Predictive Approach for High Level Vehicle Stability Control

A. Balatsiuk (6356176), F. Borsani (6354432), H. Gong (6215246), L. van der Niet (4934067),
C. Nelissen (5413400), N. Visentin (6354815)
RO47017 Vehicle Dynamics and Control - TU Delft

***Abstract*—Vehicle Stability Control is an active electronic system whose goal is to maintain the stability of the vehicle during critical situations at the limit of handling, such as sharp turns or sudden steering maneuvers. It's a complex system that involves many components and different levels of control, and it acts by controlling the yaw rate of the car, typically though the generation of yaw moment exploiting differential braking. In this report we focus on the design of the high level controller of this system, which takes a reference yaw rate and computes the corresponding yaw moment contribution. An MPC regulator is implemented for this purpose, and its performance is compared with other common controllers, namely a PD and an LQR. The testing is performed using Simulink and Matlab, simulating a Sine with Dwell maneuver and evaluating the tracking capabilities and the control effort of the controllers.**

I. Introduction

Vehicle Stability Control (VSC), also known as Electronic Stability Control (ESC), is a fundamental component in modern vehicles for ensuring safe handling, in particular during extreme maneuvers. Its basic functioning principle consists in taking the steering input δ from the driver, computing the corresponding desired behaviour of the vehicle (e.g. in terms of yaw rate r) and trying to track it as much as possible. This is done by an high level controller that calculates the yaw moment ΔM_z necessary to "correct" the error with respect to the reference. This moment is then provided to the car usually through differential braking, as shown in Figure 1. ESC is extremely important, as it improves the stability of the vehicle by a very big margin, allowing to avoid loss of control specially in dangerous and emergency situations, as schematized in Figure 2. This motivates the necessity of implementing advanced control strategies, such as MPC, that can make this system more performing and accurate. The general architecture of VSC is presented in Figure 3, and the focus from this moment will be on the high level controller.

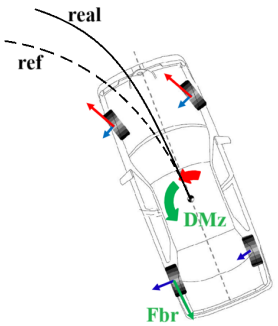

Fig. 1: Here, additional yaw moment is provided by braking with the rear left wheel

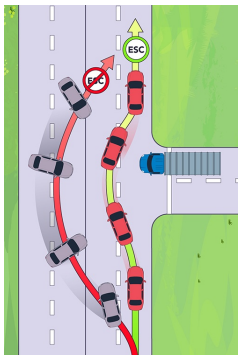

Fig. 2: An example that shows the importance of vehicle stability control in emergency situations

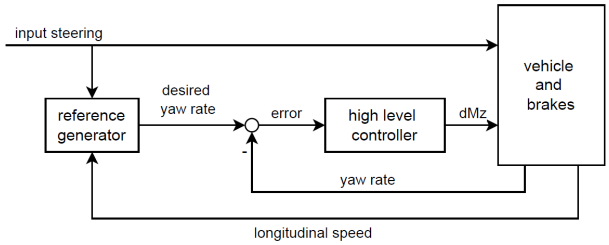

Fig. 3: VDC system simplified general architecture

Over the years, various control strategies have been developed for this component. PID controllers have been widely used in early ESC systems due to their simplicity and ease of implementation. Proportional-Integrative-Derivative (PID) controllers respond fast to yaw rate errors during rapid steering changes and control gains are tunable to balance stability and responsiveness. However, Van Zanten [1] showed that while PID improves basic stability, it fails in limit-handling scenarios. According to the studies by Rajamani [2], PID-based ESC also struggles with high-speed sine-dwell tests due to unmodeled nonlinearities.

Linear Quadratic Regulator (LQR) provides an optimal control framework by minimizing a quadratic cost function, and it usually performs better than PID in handling coupled yaw-roll dynamics. There are also some limitations on LQR controllers. As it assumes linear vehicle dynamics, LQR will lead to inaccuracies in aggressive maneuvers. Anderson and Hrynevych [3] found that LQR improves yaw stability but lags in roll mitigation during sine-dwell. Nam et al. [4] proposed gain-scheduled LQR to adapt to different steering frequencies, yet performance was still limited near tire saturation.

Model Predictive Control (MPC) has gained attention for its ability to handle vehicle nonlinearities and constraints during aggressive maneuvers. Unlike PID and LQR, MPC optimizes future control actions over a prediction horizon and can explicitly enforce safety limits on yaw rate, sideslip, and roll. Falcone et al. [5] showed that MPC significantly improves yaw and lateral control in sine-dwell tests by anticipating tire

saturation and dynamics. Di Cairano et al. [6] demonstrated that constrained MPC outperforms LQR in roll mitigation, while Lee and Park [7] introduced adaptive MPC to enhance robustness under varying road conditions. A recent comparative study by Tatar et al. [8] confirmed that MPC achieves more accurate yaw-rate reference tracking than both PID and LQR, making it more suitable for stability control under fast steering inputs.

This work involves the design, implementation and testing of a VSC system in Simulink, in particular focusing on the design of the high level controller component. The report is organized as follows. In section II the derivation of the reference generator is briefly explained. Then the design of the MPC high level controller is discussed in section III, along with the derivation of PD and LQR regulators. In section IV simulations results are presented and the performances of the various control strategies are evaluated and compared. Finally, conclusions and final considerations are drawn in section V.

II. Reference generator

As we already introduced, the role of the high level controller is to track a certain desired reference yaw rate r_{ref} that corresponds to the imposed steering angle from the driver. The goal of the reference generator is computing this reference: to do so, a simplified approach is used. First, a linear bicycle model (Appendix A) is used to calculate the steady-state response r_{ss} (steady-state cornering):

$$\dot{\mathbf{x}} = A\mathbf{x} + \mathbf{B}\delta \quad \rightarrow \quad \mathbf{x_{ss}} = -A^{-1}\mathbf{B}\delta, \quad \mathbf{x_{ss}} = [v_{y,ss} \quad r_{ss}]^T$$

Then, the maximum yaw rate is saturated basing on the "available friction", obtaining r_{sat}: if the lateral acceleration a_y is too high, there won't be enough friction force to sustain it. This limit is found as:

$$|r_{ss}| < 0.85 \ \frac{\mu g}{v_x}$$

where we only consider the first term in $a_y = ur + \dot{y}$, that contributes for $\sim 85\%$ of a_y[1]. Finally, since we used a linear, steady-state model for computing r_{ss}, we filter r_{sat} using a second order transfer function to "mimic" the real behaviour of the system:

$$\mathrm{R}_{ref}(s) = \frac{\omega_n^2(1+\tau s)}{s^2 + 2\xi\omega_n s + \omega_n^2} \ \mathrm{R}_{sat}(s)$$

Characteristic parameters[2] ω_n, ξ and τ are found experimentally through a sine-swept test and are reported in Appendix A. The general scheme of this component is shown in Figure 4.

III. High level controller

A linear MPC is designed to accomplish the functioning of the high level controller in Figure 3. For comparison, a PD and an LQR are implemented and tested as well. This section presents the design and derivation of all of them.

[1]In reality, friction coefficient is not know, so we also would need an estimator.

[2]Those parameters are taken as constant, but they should be depending on the speed.

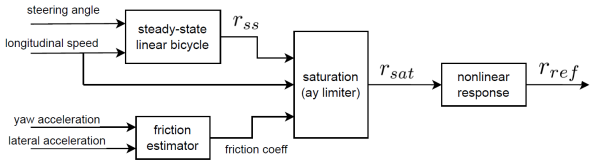

Fig. 4: Simplified scheme of the reference generator block

A. PD

A Proportional-Derivative controller is an output feedback controller that works on the yaw rate error. It does not require a model of the system and is very fast and simple. It is defined as:

$$\Delta M_z = k_p \cdot (r_{ref} - r) + k_d \cdot (\dot{r}_{ref} - \dot{r})$$

We assume we are able to directly measure the yaw rate r. Since the lateral behavior of the car also depends on the speed, gain scheduling is applied:

$$k_p = k_p(v_x) \quad \text{and} \quad k_d = k_d(v_x)$$

After some trial-and-error, the values of the gains were chosen trying to accomplish a fair trade-off between tracking performance and control effort.

B. LQR

A Linear Quadratic Regulator is a more refined model based, full-state feedback controller that relies on optimality. In particular, we will consider the infinite time LQR formulation, which is proven to be asymptotically stable at the origin under the assumptions of having a controllable LTI system[3], positive definite Q matrix and positive semidefinite R matrix. The formulation of the problem is the following. We have a linear system defined by the bicycle model (Appendix A). Since we are interested in tracking r, we define:

$$z = \mathbf{C}\mathbf{x}, \quad \text{with} \quad \mathbf{C} = [0 \quad 1]$$

and develop an output tracking problem as described in [9], by finding the reference state:

$$\mathbf{x_{ref}} = [v_{y,ref} \quad r_{ref}]^T \ \text{ such that: } \begin{cases} \mathbf{x_{ref}} = A\mathbf{x_{ref}} + \mathbf{B_M}\Delta M_{z,ref} \\ z_{ref} = \mathbf{C}\mathbf{x_{ref}} \end{cases}$$

and then shifting coordinates and building the LQR on $\tilde{\mathbf{x}} = \mathbf{x} - \mathbf{x_{ref}}$ (remember that an LQR brings the system at the origin). However, in our case we already have $\mathbf{x_{ref}}$ from the reference generator, as described in section II. At this point, the control law is:

$$\Delta M_z = -\mathbf{K}\tilde{\mathbf{x}}$$

where $\mathbf{K} = [K_1 \quad K_2]$ is the LQR gain computed as $\mathbf{K} = -R\mathbf{B_M}^T P$ being P the solution of the Algebraic Riccati Equation (ARE)[4]:

$$PA + A^T P - P\mathbf{B_M}R^{-1}\mathbf{B_M}^T P + Q = 0$$

[3]Notice that our system is not time-invariant, as the system's matrices depend on the longitudinal velocity, that changes in time.

[4]The theory behind LQR (cost function, optimality of the gains, convergence, convexity of the problem, ARE, etc) is not reported here, since it is out of the scope of this work.

The weight matrices Q and R were chosen as:

$$Q = \begin{bmatrix} 0 & 0 \\ 0 & \frac{700}{r_{max}^2} \end{bmatrix}, \qquad R = \frac{v_x}{\Delta M_{max}^2}$$

Notice that all quantities are normalized in order to take into account for the different order of magnitude of the different physical variables. In particular, r_{max} was chosen as 1 rad/s (which is approximately the order of magnitude observed in the PD simulations) and ΔM_{max} as 10000 Nm (maximum allowable yaw moment from the actuators). Finally, gain scheduling was performed as the weights depend on the longitudinal speed v_x. Once again, the final values of the weights was chosen trying to ensure a good balance between tracking performance and control effort. v_y has a null weight since we are only interested in tracking the yaw rate.

C. MPC

Model Predictive Control is an advanced approach that relies on optimality, similarly to LQR. The big difference is that MPC solves the optimization online, relying on the current measurement of the state and on a prediction of the future behaviour of the system. This makes this control strategy much more flexible and powerful, in particular enabling the inclusion of constraints, the ability of dealing with nonlinear systems and with dynamical environments. On the other side, this requires online solution of the minimization problem, increasing the computational effort by a lot, which may result critical in some real-time applications.
The general functioning principle of MPC is the following:

1) Get a measurement (or estimation) of the current state.
2) Compute a prediction of the evolution of the system for the next N time steps (prediction horizon) as a function of the control sequence.
3) Solve an optimization problem to find the optimal control sequence that brings the state towards zero according to a certain cost function.
4) Apply the first control action of the resulting optimal sequence.
5) Repeat from point 1) after one time step.

From this brief description it is clear that we need a model of the system and also a certain cost function and weights. The model we chose is once again the linear bicycle presented in Appendix A, while the cost function is the typical quadratic function that assures convexity of the problem:

$$J = \frac{1}{2}\tilde{\mathbf{x}}_N^T P \tilde{\mathbf{x}}_N + \frac{1}{2}\sum_{k=0}^{N-1}\left(\tilde{\mathbf{x}}_k^T Q \tilde{\mathbf{x}}_k + R\ \Delta M_z^2\right) \qquad (1)$$

Since MPC brings the state to zero, also in this case the problem is defined on $\tilde{\mathbf{x}} = \mathbf{x} - \mathbf{x_{ref}}$. For ease of notation, in this subsection $\tilde{\mathbf{x}} \equiv \mathbf{x}$, $\mathbf{B_M} \equiv \mathbf{B}$ and $\Delta M_z \equiv u$. With this convention, Equation 1 becomes:

$$J = \frac{1}{2}\mathbf{x}_N^T P \mathbf{x}_N + \frac{1}{2}\sum_{k=0}^{N-1}\left(\mathbf{x}_k^T Q \mathbf{x}_k + R u_k^2\right) \qquad (2)$$

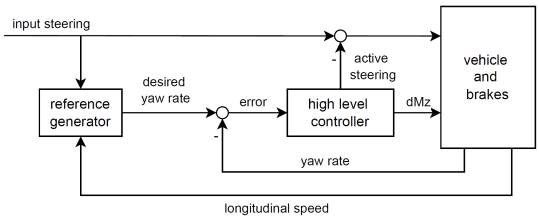

Fig. 5: VSC architecture including active steering

The optimization problem to be solved at each time step is thus:

$$\min_{\mathbf{x}_k, u_k} J(\mathbf{x}_k, u_k)$$

subj. to:

$$\mathbf{x}_{k+1} = A\mathbf{x}_k + \mathbf{B}u_k, \quad \forall k = 0, \dots, N-1$$
$$\mathbf{x}_0 = \mathbf{x}_{curr}$$
$$\mathbf{x}_k \in \mathbb{X}, \quad u_k \in \mathbb{U}, \quad \forall k = 0, \dots, N-1$$
$$\mathbf{x}_N \in \mathbb{X}_f$$

Where $\mathbb{X}$, $\mathbb{U}$ and $\mathbb{X}_f$ are the constraint sets and N the prediction horizon. Notice that with this formulation, the dynamics is an explicit constraints on the minimization, which is performed on $\mathbf{x}_k$ and u_k. This is not the way we implemented it: a detailed explanation of that can be found in Appendix B.
Similarly to LQR, the weights for the MPC were chosen as:

$$Q = \begin{bmatrix} 0 & 0 \\ 0 & \frac{700}{1^2} \end{bmatrix}, \qquad R = \frac{10}{10000^2}, \qquad P = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}$$

while the constraints:

$$|u| \leq 10000 \ \text{ Nm}$$
$$|r - r_{ref}| \leq 0.5 \ \text{ rad/s}$$

Finally, the prediction horizon is $N = 20$, to ensure feasibility of the optimization and a the same time a limited computational effort, and the controller frequency $f_c = 100$ Hz for an accurate but reasonable discretization. No terminal constraint was imposed.

D. MPC with active steering (MPCAS)

Drawing inspiration from [10], VSC system was modified by also including active steering contribution to improve tracking performance. The idea is to use an MPC with an extended model that also accounts for the steering angle as input, and then act on the behaviour of the vehicle not only through differential braking but also by providing a "correction" on the steering angle. The updated structure of the VSC is shown in Figure 5. From the implementation point of view, the only thing that changes with respect to the previous MPC is the model, that now accounts for an input $\mathbf{u} = [\Delta M_z \quad \Delta\delta]^T$. This model is presented in Appendix A and the cost function is now:

$$J = \frac{1}{2}\mathbf{x}_N^T P \mathbf{x}_N + \frac{1}{2}\sum_{k=0}^{N-1}\left(\mathbf{x}_k^T Q \mathbf{x}_k + \mathbf{u}_k^T R \mathbf{u}_k\right) \qquad (3)$$

Moreover, an additional constraint on the active steering angle contribution is included:

$$|\Delta\delta| \leq 10^o$$

IV. Simulations and Performance Evaluation

According to ISO 19365:2016, the performance of the controllers is tested using a Sine with Dwell (SwD) maneuver, for longitudinal initial speeds of 60 km/h and 100 km/h. Two different metrics are used to evaluate each controller:

- **Yaw rate metric:** This metric corresponds to the difference between the integrated yaw rate response and the integrated reference yaw rate, divided by the integrated reference yaw rate:

$$r_{metric} = \frac{\int_{t_1}^{\infty} |r(t)|dt - \int_{\bar{t}_1}^{\infty} |r_{\text{ref}}(t)|dt}{\int_{\bar{t}_1}^{\infty} |r_{\text{ref}}(t)|dt}$$

  The integration starts from the instant when the yaw rate crosses the x-axis after the first half of the maneuver. In practice, r_{metric} quantifies the tracking performance.

- **Pass/fail criteria:** One second after the completion of the SwD steering, vehicle's yaw rate (in absolute value) must not exceed 35% of the maximum yaw rate (in absolute value) recorded during the test. After 1.75 s, vehicle's yaw rate (in absolute value) must not exceed 20% of the maximum yaw rate (in absolute value) recorded during the test.

$$|r(t_{end}+1)| \leq 0.35\ |r_{max}|$$

$$|r(t_{end}+1.75)| \leq 0.35\ |r_{max}|$$

  Exceeding this threshold indicates insufficient stability and results in a test failure. Moreover, lateral displacement of the vehicle should be sufficiently high to validate the test.

The results of the first metric are compared in Figure 13, while the second metric is indicated directly in each yaw rate plot. Results of the test without any VSC active are reported in Figure 6, showing insufficient tracking performance.

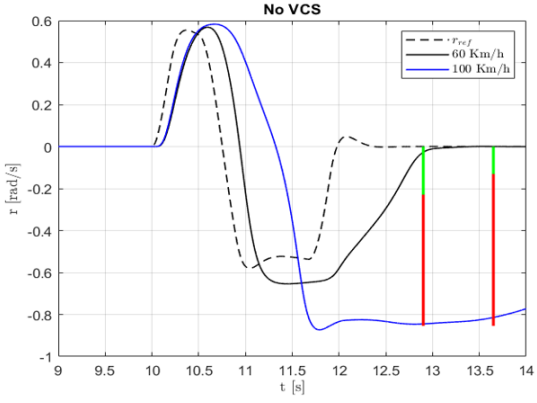

Fig. 6: Sine with Dwell without VSC

A. Sine with Dwell at 60 km/h

As seen in Figure 7, at a longitudinal speed of 60 km/h all controllers successfully track the reference yaw rate with minimal error. In terms of the yaw rate metric, PID, LQR, and MPC exhibit similar performance, with a slight improvement observed when active steering is added.

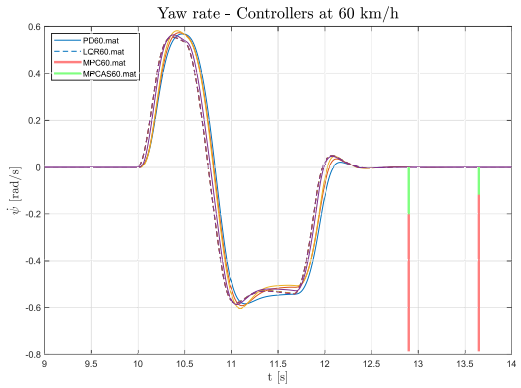

Fig. 7: Sine with Dwell maneuver at 60 km/h

The control effort is very limited, with slight variations between the controllers due to the tuning parameters.

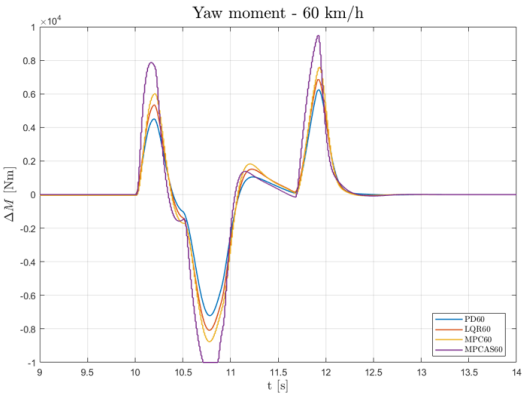

Fig. 8: Input torque at 60 km/h

B. Sine with Dwell at 100 km/h

At 100 km/h, the performance gap between the controllers increases compared to the 60 km/h test (Figure 9). In particular, a larger overshoot in the system response is observed following rapid changes in the yaw rate. The controller most affected by the speed increase is the PD, which exhibits the greatest performance drop. The LQR also shows a slight reduction in tracking accuracy. The best-performing controllers are the MPC and the MPCAS, whose performance degrades less significantly. Among all, MPCAS continues to achieve the best overall performance. The yaw input torque in Figure 10 shows the control inputs computed by each controller. As can be seen, all controllers produce inputs with a similar overall shape, with the main difference being the magnitude of the peaks. Unlike at lower speeds, at 100 km/h both MPC and

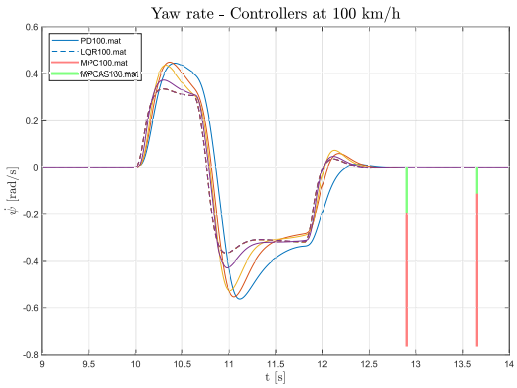

Fig. 9: Sine with Dwell maneuver at 100 km/h

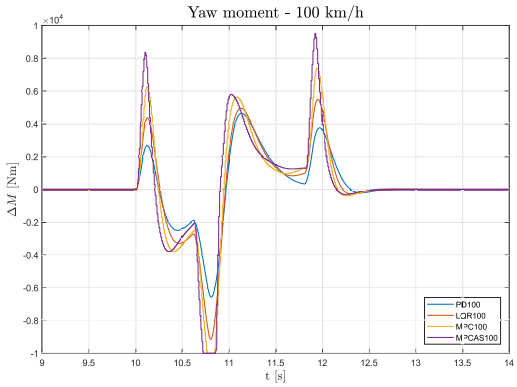

Fig. 10: Input torque at 100 km/h

MPCAS briefly reach saturation (for less than 1 s). In the case of MPCAS, this saturation could potentially be mitigated by relying more on the active steering input (Figure 11), which, in contrast, never reaches saturation.

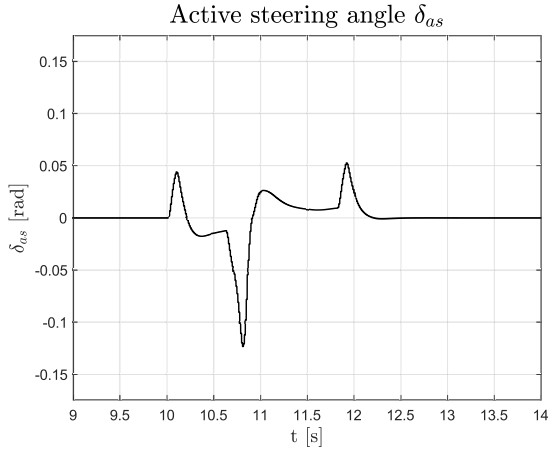

Fig. 11: Active steering input angle at 100 km/h

C. Sine with Dwell at 100 km/h with noise

The controllers are also tested under the introduction of noise on the yaw rate measurement. In particular, zero-mean uniformly distributed noise is added with amplitude of 1 deg, reflecting realistic sensor inaccuracies. As can be seen from

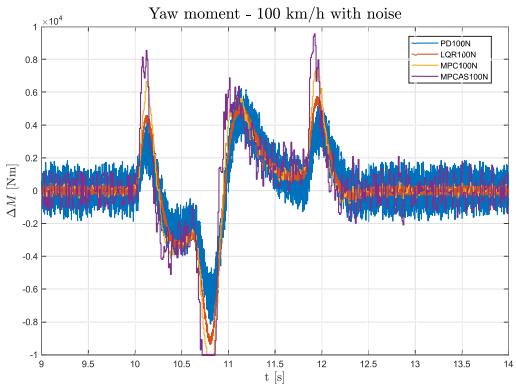

Fig. 12: Input torque at 100 km/h with noise.

Figure 13, the addition of noise does not lead to any substantial difference in tracking performance. The only controller showing a slight degradation is the MPCAS. However, this difference remains minor, as its performance still surpasses that of the other controllers. Major differences can instead be observed in the input plots shown in Figure 12. In particular, PD and MPCAS exhibit the noisiest control inputs, whereas the MPC and LQR maintain a lower variance in their input signals.

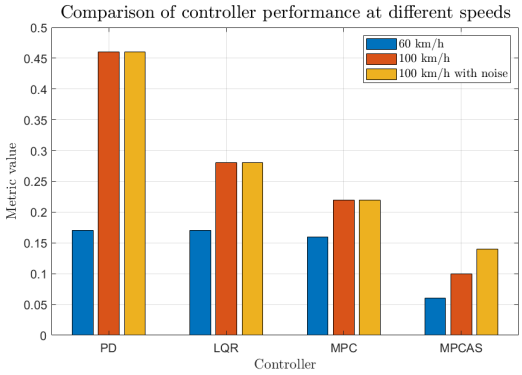

Fig. 13: Yaw rate metric results

V. Conclusions

In this paper, the design, implementation and comparative evaluation of high-level control strategies for vehicle stability control were presented, with a focus on the performance of yaw rate tracking under critical dynamic conditions. Three controllers, PD, LQR and MPC, were created and evaluated using a Sine with Dwell maneuver at different speeds, both with and without measurement noise. According to simulation results, all controllers perform accurately at lower speeds (60 km/h), but for more critical situations (100 km/h), their behaviour diverges considerably. In general, as visible from Figure 6, any type of control improves significantly car's response, remarking the importance of VSC system.
PD controller is the simplest to implement, as it does not rely on any model of the system. However, for the same reason it is the one showing the worst performance in high speed

scenarios. Also, due to the derivative contribution, this type of approach is very sensitive to noise.
LQR and MPC performed better than PD, showing similar performances both in terms of tracking and control effort. MPC though, also gave the possibility to include constraints on the states and inputs. Both of these regulators also allowed for a more intuitive tuning of the controller weights, as they reflected specific objectives (control effort and tracking accuracy).
The best overall performance was obtained by integrating active steering in MPC, of course at cost of a more complex formulation.
Finally, we need to remember that both MPCs come with an higher computational effort, as they need to solve an online optimization. However, the running times observed during the simulations were absolutely compatible with a real-time implementation, mainly thanks to the choice of the relatively small prediction horizon and the controller frequency.

The main challenges faced during this project involved MPC formulation and practical implementation in Simulink and gain/weights tuning for all the controllers.
Future improvement of this work might concentrate on implementing and comparing more refined models, for example also incorporating tire nonlinearities (nonlinear MPC), road friction estimation, limits on the sideslip angle or higher-DOF dynamics. Another promising avenue is to look into adaptive or learning-based MPC techniques, for example implementing data-driven learnt models for the MPC.

REFERENCES

[1] A. T. V. Zanten, "Bosch esp systems: 5 years of experience," SAE Technical Paper 2000-01-1633, Tech. Rep., 2000.

[2] R. Rajamani, *Vehicle Dynamics and Control*, 2nd ed. Springer, 2012.

[3] M. Anderson and M. Hrynevych, "Evaluation of linear quadratic regulation for electronic stability control," *IEEE Transactions on Control Systems Technology*, vol. 24, no. 4, pp. 1346–1354, 2016.

[4] E. Nam, D. Kim, and J. Lee, "Gain-scheduled lqr for vehicle yaw stability control in high-speed maneuvers," *International Journal of Automotive Technology*, vol. 19, no. 1, pp. 25–34, 2018.

[5] P. Falcone, F. Borrelli, J. Asgari, H. E. Tseng, and D. Hrovat, "Predictive active steering control for autonomous vehicle systems," *IEEE Transactions on Control Systems Technology*, vol. 15, no. 3, pp. 566–580, 2007.

[6] S. D. Cairano, H. E. Tseng, D. Bernardini, A. Bemporad, and I. Kolmanovsky, "Vehicle yaw stability control by coordinated active front steering and differential braking in the tire sideslip angles domain," *IEEE Transactions on Control Systems Technology*, vol. 21, no. 4, pp. 1236–1248, 2013.

[7] J. Lee and J. Park, "Adaptive model predictive control for vehicle stability with online model update under uncertain driving conditions," *IEEE Transactions on Intelligent Vehicles*, vol. 6, no. 2, pp. 309–319, 2021.

[8] A. Tatar and A. Sharma, "Comparative study of mpc and pid controllers in autonomous vehicle application," *Procedia Computer Science*, vol. 221, pp. 1018–1025, 2023.

[9] J. Rawlings and D. Mayne, *Model Predictive Control: Theory and Design*. Nob Hill Publishing, 2008.

[10] M. Choi and S. Choi, "Mpc for vehicle lateral stability via differential braking and active front steering considering practical aspects," *Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering*, vol. 230, 06 2015.

APPENDIX A
BICYCLE MODEL AND PARAMETERS

The linear bicycle model used for the reference generator (steering angle δ as input) is [2]:

$$\dot{\mathbf{x}} = A\mathbf{x} + \mathbf{B}\delta, \quad \mathbf{x} = [v_y \;\; r]^T$$

where:

$$A = \begin{bmatrix} -\dfrac{C_{\alpha f} + C_{\alpha r}}{m v_x} & \dfrac{l_r C_{\alpha r} - l_f C_{\alpha f}}{m v_x} - v_x \\ \dfrac{l_r C_{\alpha r} - l_f C_{\alpha f}}{I_z v_x} & -\dfrac{l_r^2 C_{\alpha r} + l_f^2 C_{\alpha f}}{I_z v_x} \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} \dfrac{C_{\alpha f}}{m} \\ \dfrac{l_f C_{\alpha f}}{I_z} \end{bmatrix}$$

The linear bicycle model used for the state-based controllers (yaw moment ΔM_z as input) is:

$$\dot{\mathbf{x}} = A\mathbf{x} + \mathbf{B_M}\, \Delta M_z, \quad \mathbf{x} = [v_y \;\; r]^T \tag{4}$$

where:

$$\mathbf{B_M} = \begin{bmatrix} 0 \\ 1/I_z \end{bmatrix}$$

The linear bicycle model used for active steering MPC is:

$$\dot{\mathbf{x}} = A\mathbf{x} + B_{\text{ex}} \begin{bmatrix} \Delta M_z \\ \delta \end{bmatrix}, \quad \mathbf{x} = [v_y \;\; r]^T \tag{5}$$

where:

$$B_{\text{ex}} = \begin{bmatrix} 0 & \dfrac{C_{\alpha f}}{m} \\ \dfrac{1}{I_z} & \dfrac{l_f C_{\alpha f}}{I_z} \end{bmatrix}$$

Vehicle, simulations and controller parameters are reported in the following tables.

Param.	Value	Unit	Description
m	1380	kg	vehicle mass
I_z	2634.5	kg·m^2	moment of inertia about z-axis
L	2.79	m	wheelbase
l_f	1.384	m	distance from front axle to CoG
l_r	1.406	m	distance from rear axle to CoG
i_{steer}	15.4	-	steering ratio
m_f	695.4	kg	front sprung mass
m_r	684.6	kg	rear sprung mass
$C_{\alpha f}$	120000	N/rad	front axle cornering stiffness
$C_{\alpha r}$	190000	N/rad	rear axle cornering stiffness
K_{us}	0.0022	N^{-1}	understeer gradient
ω_n	11	rad/s	yaw rate natural freq. (sine swept test)
ξ	0.7	-	yaw rate damping (sine swept test)
τ	0.09	s	yaw rate time constant (sine swept test)
g	9.81	m/s^2	gravitational acceleration
V_0	60 or 100	Km/h	constant speed before the maneuver
μ	1	-	friction coefficient

TABLE I: Vehicle parameters

Param.	Value	Unit	Description
t_{sim}	20	s	simulation duration
f_{sim}	1000	Hz	simulation frequency
Δt	0.01	s	MPC sampling time
N	20	-	MPC prediction horizon
t_0	10	s	maneuver start time

TABLE II: MPC and simulation parameters

Symbol	Unit	Description
v_x	m/s	longitudinal velocity
v_y	m/s	lateral velocity
a_y	m/s^2	lateral acceleration
y	m	lateral position of the CoG
r	rad/s	yaw rate
δ	rad	steering angle
ΔM_z	Nm	added yaw moment
$\mathbf{x}$		state vector
$\mathbf{u}$		input signal (output of the high level controller)

TABLE III: Symbols and notation

APPENDIX B
EXTENSIVE MPC FORMULATION

The general working principle of MPC is presented in subsection III-C. Here we show in detail how it was implemented. The notation in this appendix is the same of subsection III-C, with $\tilde{\mathbf{x}} \equiv \mathbf{x}$, $\mathbf{B_M}$ (or B_{ex}) $\equiv B$ and ΔM_z (or $[\Delta M_z \ \ \Delta\delta]^T$) $\equiv \mathbf{u}$. Dimension of $\mathbf{x}$ is 2 and dimension of $\mathbf{u}$ is n_u

First of all, MPC works in discrete time. The continuous system in Equation 4 (or Equation 5) is discretized with sampling time Δt:

$$\dot{\mathbf{x}} = A\mathbf{x} + B\mathbf{u} \quad \rightarrow \quad \mathbf{x}_{k+1} = A_\mathrm{d}\mathbf{x}_k + B_\mathrm{d}\mathbf{u}_k$$

with:

$$e^{\Delta t\begin{bmatrix} A & B \\ 0 & 0 \end{bmatrix}} = \begin{bmatrix} A_\mathrm{d} & B_\mathrm{d} \\ 0 & \mathrm{I} \end{bmatrix}$$

Then, a dense formulation was used to express the evolution of the dynamics in the prediction horizon (N time steps):

$$\mathbf{X} = T\mathbf{x}_0 + S\mathbf{U}$$

with:

$$\mathbf{X} = [\mathbf{x}_0^T \ \ \mathbf{x}_1^T \ \ ... \ \ \mathbf{x}_N^T]^T \in \mathbb{R}^{2(N+1)}$$
$$\mathbf{U} = [\mathbf{u}_0^T \ \ \mathbf{u}_1^T \ \ ... \ \ \mathbf{u}_{N-1}^T]^T \in \mathbb{R}^{n_u \cdot N}$$

Notice that we are expressing the state evolution as a only function of the initial condition $\mathbf{x}_0$ and the control sequence $\mathbf{U}$. Prediction matrices T and S are:

$$T = \begin{bmatrix} I \\ A_\mathrm{d} \\ A_\mathrm{d}^2 \\ \vdots \\ A_\mathrm{d}^N \end{bmatrix} \in \mathbb{R}^{2(N+1)\times 2}$$

$$S = \begin{bmatrix} 0 & 0 & \dots & 0 & 0 \\ B_\mathrm{d} & 0 & \dots & 0 & 0 \\ A_\mathrm{d}B_\mathrm{d} & B_\mathrm{d} & \dots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ A_\mathrm{d}^{N-1}B_\mathrm{d} & A_\mathrm{d}^{N-2}B_\mathrm{d} & \dots & A_\mathrm{d}B_\mathrm{d} & B_\mathrm{d} \end{bmatrix} \in \mathbb{R}^{2(N+1)\times n_u \cdot N}$$

As A_d and B_d depend on the velocity v_x, all these matrices need to be computed online, increasing the computational load with respect to a classical LTI MPC.

The cost function in Equation 2 (or Equation 3) can be formulated in dense form as:

$$J = \frac{1}{2}\mathbf{U}^T H\mathbf{U} + (\mathbf{h}\mathbf{x}_0)^T\mathbf{U} + \text{const}$$

where:

$$H = S^T\bar{Q}S + \bar{R}, \quad H \in \mathbb{R}^{n_u \cdot N \times n_u \cdot N}$$
$$\mathbf{h} = S^T\bar{Q}^T T, \quad \mathbf{h} \in \mathbb{R}^{n_u \cdot N \times 2}$$

and:

$$\bar{Q} = \begin{bmatrix} Q & & & \\ & \ddots & & \\ & & Q & \\ & & & P \end{bmatrix} \in \mathbb{R}^{2(N+1)\times 2(N+1)}$$

$$\bar{R} = \begin{bmatrix} R & & \\ & \ddots & \\ & & R \end{bmatrix} \in \mathbb{R}^{n_u \cdot N \times n_u \cdot N}$$

At this point the constraints are built. We want to impose:

input constraints: $\mathbf{u}_{min} \leq \mathbf{u}_k \leq \mathbf{u}_{max} \quad \forall k = 0, ..., N-1$

state constraints: $\mathbf{x}_{min} \leq \mathbf{x}_k \leq \mathbf{x}_{max} \quad \forall k = 1, ..., N$

Also in this case, everything can be formulated in matrix form as:

$$G\mathbf{U} \leq \mathbf{g}$$

where:

$$G = \bar{D}\tilde{S} + \bar{E}, \quad G \in \mathbb{R}^{2\cdot(2+n_u)N \times n_u \cdot N}$$
$$\mathbf{g} = \bar{\mathbf{b}} - \bar{D}\tilde{T}\mathbf{x}_0, \quad \mathbf{g} \in \mathbb{R}^{2\cdot(2+n_u)N}$$

and:

$$\bar{D} = \begin{bmatrix} D & & \\ & \ddots & \\ & & D \end{bmatrix} \in \mathbb{R}^{2\cdot(2+n_u)N \times 2N}$$

$$D = \begin{bmatrix} A_\mathrm{d} \\ -A_\mathrm{d} \\ 0 \\ 0 \end{bmatrix} \in \mathbb{R}^{2\cdot(2+n_u)\times 2}$$

$$\bar{E} = \begin{bmatrix} E & & \\ & \ddots & \\ & & E \end{bmatrix} \in \mathbb{R}^{2\cdot(2+n_u)N \times n_u \cdot N}$$

$$E = \begin{bmatrix} B_\mathrm{d} \\ -B_\mathrm{d} \\ \mathrm{I} \\ \mathrm{I} \end{bmatrix} \in \mathbb{R}^{2\cdot(2+n_u)\times n_u}$$

$$\bar{\mathbf{b}} = \begin{bmatrix} \mathbf{b} \\ \vdots \\ \mathbf{b} \end{bmatrix} \in \mathbb{R}^{2\cdot(2+n_u)N}$$

$$\mathbf{b} = \begin{bmatrix} \mathbf{x}_{max} \\ -\mathbf{x}_{min} \\ \mathbf{u}_{max} \\ -\mathbf{u}_{min} \end{bmatrix} \in \mathbb{R}^{2\cdot(2+n_u)}$$

$\tilde{S} = [S$ matrix without the last block row$] \in \mathbb{R}^{2N \times n_u \cdot N}$

$\tilde{T} = [T$ matrix without the last block row$] \in \mathbb{R}^{2N \times 2}$

Everything is ready, and a quadratic programming solver (we chose `quadprog` in Matlab/Simulink) can be used to

compute:

$$\mathbf{U}^* = \arg\min_{\mathbf{U}} J(\mathbf{U})$$

subj. to:

$$\mathbf{x}_0 = \mathbf{x}_{curr}$$

$$G\mathbf{U} \leq \mathbf{g}$$

The first element of $\mathbf{U}^*$ is used as constant control input until a new optimization is solved after Δt seconds.