Predictive handling limits monitoring and agility improvement with torque vectoring on a rear in-wheel drive electric vehicle

Predictive handling limits monitoring and agility improvement with torque vectoring on a rear in-wheel drive electric vehicle

Luis M. Castellanos Molina, Raffaele Manca, Shailesh Hegde, Nicola Amati and Andrea Tonoli

Center for Automotive Research and Sustainable Mobility, Mechatronics Laboratory, Department of Mechanical and Aerospace Engineering, Politecnico di Torino, Corso Duca degli Abruzzi, 24, Turin, Italy.

ARTICLE HISTORY
Compiled November 1, 2023

ABSTRACT
This work presents a predictive torque vectoring controller that optimally monitors the vehicle limits of handling using active torque distribution in the rear axle of a fully electric vehicle. It works in combination with a feedforward controller designed to improve the vehicle's agility. The overall torque vectoring strategy is described together with the vehicle lateral dynamics, sideslip angle estimator, and torque allocation method. Numerical simulations for various scenarios and road profiles show the benefits of predicting the vehicle's handling limits and the enhancement of vehicle stability in terms of reduced vehicle sideslip angle and driver effort. The proposed optimal control method for predicting vehicle handling limit violations does not require a dedicated solver, making it a promising candidate for real-time applications. The case study is a vehicle equipped with two rear in-wheel motors in the framework of HiPERFORM, an ECSEL Joint Undertaking (JU) European research project. Hardware-in-the-loop (HiL) tests were performed on a dedicated e-axle test bench to integrate the torque vectoring controller with the real e-motors and a dual inverter. The results of the HiL testing demonstrate that the torque-vectoring requirements are satisfied by the hardware configuration in use.

KEYWORDS
electric vehicle; in-wheel motor; torque vectoring; optimal control, feedforward control

1. Introduction

Electric mobility research is paving the way for the development of new powertrain layouts to maximize the potential of electric motors. Among all the configurations, vehicles equipped with in-wheel e-motors seem quite promising since intelligent traction control can be done to improve vehicle dynamics and active safety performance. The powertrain acts as a chassis actuator in these vehicles, allowing full control of wheel torque distribution, i.e., the so-called torque vectoring (TV) [1].

Different control strategies can be exploited for torque vectoring control. Typically, the steering angle information, which is a known exogenous disturbance applied by the driver, is used to generate either a yaw rate target or a direct yaw moment that

CONTACT Luis M. Castellanos Molina Email: luis.castellanos@polito.it

modifies the dynamics of the vehicle chassis [2]. A comparison of feedback control techniques for torque vectoring control of fully electric vehicles is presented in [3]. Proportional Integral controller for tracking a desired yaw rate [4] or adaptive linear quadratic regulators for both rear-wheel drive [5] and all-wheel drive layouts [6,7] are well-known candidates. Applications of sliding mode controllers for torque vectoring, are particularly effective to improve tracking performance in the presence of plant-model mismatch and disturbances [8]. A common drawback of sliding mode controllers is chattering which can be improved with advanced anti-chattering techniques [9,10]. More sophisticated feedforward TV control variants have been reported in the literature either by utilising the flatness properties of a vehicle model [11] or by employing input-output linearization [12]. In general, a feedforward-based TV can improve the vehicle's agility. However, because feedforward operates in an open loop, special attention should be paid to the plant-model mismatch when using nonlinear feedforward controllers. Adding a state feedback regulator to compensate for any undesirable behaviour is a common practice.

The use of model predictive controllers (MPC) for torque vectoring is gaining attention because it offers a unique framework to optimally handle the vehicle dynamics considering the bounds on the yaw rate and sideslip angle, and the limits of the yaw moment generator (e-motor torques in the case of fully electric vehicles) or tire slips [13–16]. Detailed comparison of linear and non-linear MPC-based TV controllers for a vehicle equipped with a rear e-axis is presented in [16]. Once again, questions have been raised regarding the real-time implementation of the MPC as well as the relative balance between performance and complexity of the problem. Using MPC requires solving a quadratic programming problem in real time which is hard to formulate when the vehicle model predictor contains tire nonlinearities. Its implementation becomes more complex when limits on the yaw rate and sideslip angle are added to the problem formulation. They should be properly handled by using softening constraints to avoid infeasibility issues.

In this work, we propose a novel torque vectoring method that generates two yaw moment contributions: one coming from a feedforward controller that improves the agility of the vehicle and a state feedback regulator that optimally monitors the vehicle's handling limits. The latter is based on a nonlinear control problem that predicts the evolution of the vehicle as well as the limits of handling violation in terms of maximum yaw rate and body sideslip angle. The way the control problem is formulated ensures that a correcting yaw moment is only applied when the vehicle starts to veer outside of its handling limits. The nonlinear control problem is cast into a quadratic programming problem without inequality constraints and is hence simple enough to be solved in real time without the need for a dedicated quadratic programming solver. The current study attempts, to some extent, to bridge the gap between purely theoretical simulations and practical concerns about torque vectoring implementation.

The target of the proposed optimal controller is to guarantee vehicle stability similar to standard Electronic Stability Control (ESC) strategies but acting on rear in-wheel motors and using the prediction of the vehicle limits of handling. Typically, ESC variants are activated when the vehicle reaches predefined limits in terms of maximum allowable yaw rate and sideslip angle. In other words, ECS reacts in a "triggered" mode only when an unsafe deviation is measured [2, § 5.3] and with no prediction at all. This work solves the triggering phenomenon and the lack of prediction because the control problem is formulated to effectively handle the activation and deactivation of a safe yaw moment generation using the prediction of yaw rate and vehicle sideslip limits.

Some studies have proposed predictive torque vectoring controllers based on MPC formulations that treat the state deviations as inequality constraints, and hence the limits of yaw rate and body sideslip are predicted and properly handled. However, to the author's knowledge, no previous work has investigated the benefits of using these limits to generate a correcting yaw moment without the need for a dedicated quadratic programming solver as proposed in this work. The case study is a vehicle equipped with two rear in-wheel motors in the framework of HiPERFORM, an ECSEL Joint Undertaking (JU) European research project. The scope of the project includes the performance evaluation of a rear e-axle, consisting of a dual inverter (Ideas & Motion) and two in-wheel motors (Elaphe).

The performance of the proposed controller is firstly assessed through numerical simulations of both open-loop manoeuvres (i.e., by setting a steering angle profile with no driver on board) and closed-loop manoeuvres (i.e., driver in the loop). It is also validated by hardware-in-the-loop testing on a dedicated e-axle test rig with real vehicle components. The dual inverter controls the torque in both in-wheel motors. It receives, via CAN communication, the wheel torque references from the torque vectoring controller. The latter is deployed into dSPACE SCALEXIO AutoBox hardware, which is a modular real-time system for in-vehicle use. The test campaign is intended to demonstrate that the hardware configuration in use meets the torque-vectoring requirements.

The work is organized as follows. The vehicle model formulation is described in Section 2, which includes details on the modeling of the vehicle tire and of the operation of the in-wheel motors. The torque vectoring control strategy is discussed in Section 3, including the two main ingredients: an optimal controller that monitors the limits of handling and a feedforward controller designed to improve vehicle agility. Furthermore, in the dedicated subsection, the methodology for the estimation of the vehicle state sideslip angle is presented. The control framework is completed with a description of the torque distribution and anti-slip control algorithms. The numerical simulations are presented in Section 4, with a focus on open-loop manoeuvres and closed-loop manoeuvres to assess the influence of the state target prediction. Section 5 presents the performance obtained through the hardware in the loop with the e-axle test campaign. Finally, Section 6 concludes the work.

2. Vehicle model

The vehicle under study is a full-electric car outfitted with two in-wheel motors in the rear axle. The goal is to properly handle the torque generated by each motor to guarantee the requested longitudinal force while, at the same time, generating a correcting yaw moment to improve the lateral dynamics. The yaw moment and wheel forces are depicted in a top view of the vehicle in Figure 1.

The lateral dynamics of the vehicle is modeled in terms of yaw rate (r) and sideslip angle (β) using a dual-track vehicle model as described in [17, § 8.4] and based on the assumptions of negligible force due to road bank angle, known and constant vehicle speed, small sideslip angle and steering angle δ, negligible aerodynamic forces and negligible tire self-alignment moment:

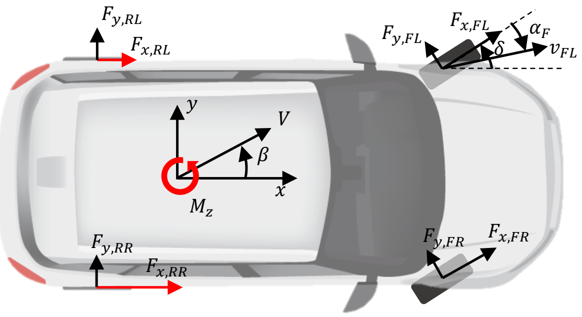


Figure 1. Tire forces and yaw moment on the vehicle.

$$mv_x\left(\dot{\beta}+r\right)=F_{yF}+F_{yR} \tag{1}$$

$$J_z\dot{r}=l_FF_{yF}-l_RF_{yR}+M_z, \tag{2}$$

where m is the vehicle mass, v_x is the vehicle longitudinal velocity, J_z is the yaw moment of inertia, l_F and l_R are the longitudinal distances from the centre of gravity. M_z is a yaw moment actively generated by the rear in-wheel motor when a differential torque is applied. The front and rear tire lateral forces (F_{yF}, F_{yR}) are given by

$$\begin{aligned}F_{y,F}&=F_{y,FL}(\alpha_F,\ F_{z,FL})+F_{y,FR}(\alpha_F,\ F_{z,FR}),\\F_{y,R}&=F_{y,RL}(\alpha_R,\ F_{z,RL})+F_{y,RR}(\alpha_R,\ F_{z,RR}),\end{aligned} \tag{3}$$

with front and rear slip angles given by

$$\begin{aligned}\alpha_F&=\beta+\frac{l_Fr}{v_x}-\delta,\\\alpha_R&=\beta-\frac{l_Rr}{v_x}.\end{aligned} \tag{4}$$

The vertical forces $F_{z,ij}$ are calculated using the weight distribution but assuming lateral and longitudinal acceleration of the carriage $(a_x,\ a_y)$ as measured in the vehicle [17]:

$$\begin{aligned}F_{z,FL}&=m\cdot\left(\frac{l_R}{l}g-\frac{h_{CoG}}{l}a_x\right)\left[\frac{1}{2}-\frac{h_{CoG}\cdot a_y}{b_F\cdot g}\right]\\F_{z,FR}&=m\cdot\left(\frac{l_R}{l}g-\frac{h_{CoG}}{l}a_x\right)\left[\frac{1}{2}+\frac{h_{CoG}\cdot a_y}{b_F\cdot g}\right]\\F_{z,RL}&=m\cdot\left(\frac{l_F}{l}g+\frac{h_{CoG}}{l}a_x\right)\left[\frac{1}{2}-\frac{h_{CoG}\cdot a_y}{b_R\cdot g}\right]\\F_{z,RR}&=m\cdot\left(\frac{l_F}{l}g+\frac{h_{CoG}}{l}a_x\right)\left[\frac{1}{2}+\frac{h_{CoG}\cdot a_x}{b_R\cdot g}\right],\end{aligned} \tag{5}$$

where l is the vehicle wheelbase, and b_F and b_R are the front and rear track width, respectively.

Equations (1)-(5) represent the nominal model of the vehicle and can be condensed to the nonlinear time-domain representation

$$\dot{\xi} = f(\xi, u, \delta, p), \tag{6}$$

with the state vector $\xi = [\beta,\ r]^\intercal$, the yaw moment $M_z \equiv u$ as manipulated input, the steering angle (δ) as known disturbance input, and vector $p = [v_x,\ a_x,\ a_y]^\intercal$ lumps the time-varying parameters: longitudinal speed, longitudinal acceleration and lateral acceleration of the chassis, respectively, all assumed as measured quantities available from the vehicle control unit (VCU). Hereinafter, the nonlinear system (6) is used as the nominal model of the vehicle. The analysis carried out in this work refers to a vehicle having the data reported in Table 1.

Table 1. Vehicle data

Vehicle parameter	value	Description	Units
m	1430	mass	kg
J_z	2059.2	Yaw moment of inertia	kgm^2
l_F	0.996	Distance of COG from front axle	m
l_R	1.494	Distance of COG from rear axle	m
h_{CoG}	0.65	Distance of COG from ground	m
$b_F = b_R$	1.565	Track width	m
R_w	0.308	Wheel effective rolling radius	m

2.1. Vehicle tires

The tire forces in Equation (3) are obtained considering the time-varying distribution of the vertical force. To do so, the heuristic approach

$$F_{y,ij} = k_{red,ij}\left(k_1 - \frac{F_{z,ij}}{k_2}\right) \cdot F_{z,ij} \cdot \arctan\left(k_3 \cdot \alpha_i\right), \tag{7}$$

with $i = F$(front), R(rear), $j = L$ (left), R(right) is adopted from [17, § 9.5.2]. Equation (7) is a nonlinear function of the vertical force and the sideslip angle of the wheel. It approximates the Pacejka-based formulation when time-varying vertical forces are considered but using only parameters: $k_1,\ k_2,\ k_3$. A reduction factor $k_{red,ij}$ is present to limit the tire forces with the tire-road friction and the combined lateral-longitudinal slip effect.

The lateral forces in Equation (7) are fitted to the type of tire under study: 205/55r16. The coefficients are fitted once and offline from a given Pacejka-based tire dataset as depicted in Fig. 2. The error of this approximation is minimized using nonlinear least squares in Matlab$^{\text{TM}}$. The parameters are $k_1 = 0.6819$, $k_2 = 1.385e{+}05$ and $k_3 = 40.85$ with 95 % confidence bounds. The fit (NRMSE index) value is 85.6 % which is considered sufficient.

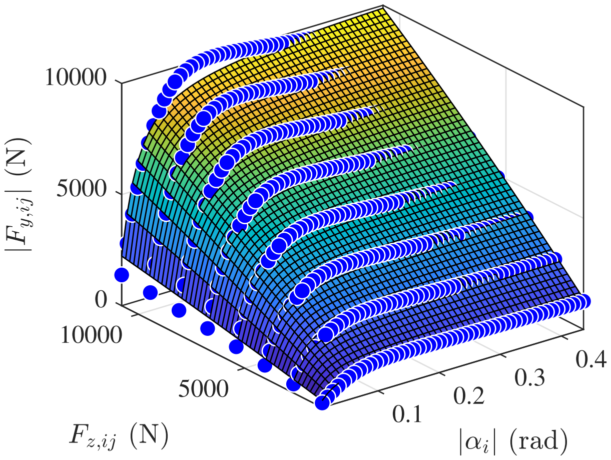


Figure 2. Comparison of Equation (7) (surface) to the data of the tire 205/55r16 (blue dotted lines). The latter is obtained from the Pacejka magic formula.

2.2. In-wheel motors operation

Two in-wheel motors Elaphe M700 (version VD2) are mounted on the rear axle of the vehicle. They are water-cooled outer rotor synchronous in-wheel motors with surface-mounted permanent magnets. Each motor is able to deliver a continuous torque of 400 Nm (maximum torque 700 Nm), 60 kW of maximum power. Both in-wheel motors are controlled in torque mode i.e., the torque vectoring controller receives the accelerator pedal demand coming from the driver and then translates it into two torque demands for both motors. Note that if the torque demand is not equally split, a yaw moment is generated.

3. Torque vectoring control strategy

Figure 3 depicts the overall control scheme. The goal is to generate a correcting yaw moment by applying a differential torque on the rear axle, which is equipped with two in-wheel e-motors. The torque vectoring produces two distinct yaw moments: $M_{z,FF}$, $M_{z,FB}$. The first one is produced by a feedforward controller, which forces the vehicle to behave in a desired manner. The second contribution comes from an optimal state feedback controller, which is used to stabilize the vehicle if its dynamics exceed the limits of handling. An extended Kalman filter estimator combined with a kinematic formula is also present to estimate the body sideslip angle. All of these elements are discussed in detail further below.

3.1. Optimal controller for handling limits monitoring

A state feedback controller is proposed to monitor vehicle behaviour, specifically the vehicle's handling limits in terms of maximum permissible sideslip angle (β_{max}) and yaw rate (r_{max}). It predicts when the vehicle tends to move out of the limits of handling and hence generates a correcting yaw moment $M_{z,FB}$ if needed. To do so, we propose

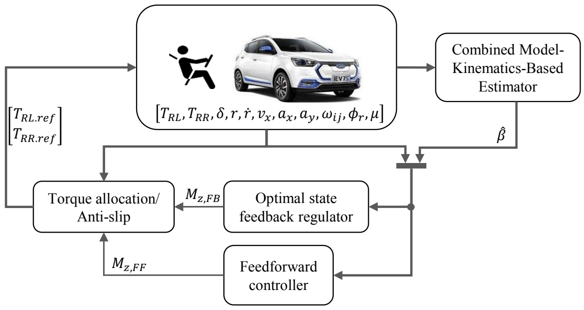

Combined Model-Kinematics-Based Estimator
$[T_{RL}, T_{RR}, \delta, r, \dot{r}, v_x, a_x, a_y, \omega_{ij}, \phi_r, \mu]$
$\begin{bmatrix} T_{RL.ref} \\ T_{RR.ref} \end{bmatrix}$
$\hat{\beta}$
Torque allocation/ Anti-slip
$M_{z,FB}$
Optimal state feedback regulator
$M_{z,FF}$
Feedforward controller
iEV7S

Figure 3. General torque vectoring control scheme. The vehicle, predictive driver, scenario and vehicle environment are set using Matlab$^{\text{TM}}$ environment, specifically the Vehicle Dynamics Blockset$^{\text{TM}}$. The main ingredients of the proposed TV controller are: i) a kinematic-based sideslip angle estimator, ii) a feedforward controller for agility improvement, iii) an optimal state feedback regulator for handling limits monitoring, and iv) a torque allocation method / anti-slip controller that sets the torque references to the rear in-wheel motors.

a finite-horizon optimal control problem defined as

$$\min_{\Delta u_0,\ldots,\Delta u_{N-1}} \frac{1}{2} \sum_{i=0}^{N-1} \left(\left\| e_{i|k} \right\|_Q^2 + \left\| u_{i|k} \right\|_{R_u}^2 + \left\| \Delta u_{i|k} \right\|_{R_{\Delta u}}^2 \right) \tag{8a}$$

$$\begin{aligned} &\text{subj. to} \\ \xi_{i+1|k} &= A_k \xi_{i|k} + B_k u_{i|k} + B_{dk} \delta_k, \quad i = 0, \ldots, N-1 && \text{(8b)} \\ u_{i|k} &= u_{i-1|k} + \Delta u_{i|k}, \quad i \geq 0 && \text{(8c)} \\ u_{-1|k} &= M_{z,sat|k-1} && \text{(8d)} \\ \xi_{0|k} &= \hat{\xi}_k && \text{(8e)} \\ e_{i|k} &= \xi_{i|k} - \xi_{ref,i|k}, \quad i = 0, \ldots, N-1, && \text{(8f)} \end{aligned}$$

to find the optimal yaw moment variation from $i = 0$ to N:

$$\Delta \mathbf{u}^\star = \left[\Delta u_0^\star, \ \ldots, \ \Delta u_{N-1}^\star \right]^\intercal. \tag{9}$$

The optimal yaw moment contribution $M_{z,FB}$ to be applied during the next time interval is generated using only the first element $\Delta u_0^\star$ of the optimal predicted sequence:

$$M_{z,FB} = u_{k-1} + \Delta u_0^\star. \tag{10}$$

The subscript $_{i|k}$ in problem (8) is used to denote a variable at time $k + i$ that is predicted at time k (i.e. predictions of their values i steps ahead). Matrices A, B, B_d are initialized by applying exact discretization to the continuous Jacobians of the nonlinear model (6). Equation (8c) is used to cast the problem into an incremental input–output (IIO) model and hence use $\Delta \mathbf{u}^\star$ as the optimization variable. Note that Equation (8d) is used to initialize the optimizer with the active yaw moment that was effectively applied during the previous time interval: $M_{z,sat|k-1}$.

The state deviation (8f) is calculated using state targets $\xi_{ref_k} = \left[r_{ref_k} \ \beta_{ref_k} \right]^\intercal$

given by

$$
\begin{aligned}
\beta_{\mathrm{ref}\,i|k} &= \beta_{max} \cdot \tanh\left(\frac{\beta_{i|k}}{\beta_{max}}\right), \\
r_{\mathrm{ref}\,i|k} &= r_{max} \tanh\left(\frac{r_{i|k}}{r_{max}}\right), \quad i = 0, \ldots, N-1,
\end{aligned}
\tag{11}
$$

where $\beta_{max} = \arctan(0.02\mu g)$, and $r_{max} = 0.85\mu g/V_{COG}$ define the limits of handling of the vehicle [18]. With these target definitions, the cost term of the state deviation is nonzero only when the vehicle is out of the limits of handling. This formulation is necessary for allowing the feedforward controller to act on the vehicle but only if the resulting manoeuvre is safe. If the vehicle tends to move out of the limits of handling $(r_{max},\ \beta_{max})$, the state feedback regulator enters into action.

Because the state targets presented in (11) are nonlinearly dependent on the state evolution, they introduce nonlinearity in the cost function (8a). To reduce the complexity of the problem, in this work, the evolution of the state targets is not predicted inside the optimizer but calculated a priori every time step. To do so, at time k, the predictor model (8b, 8c) is firstly simulated from $i = 0$ to $N-1$ but using the previous optimal input sequence $(\Delta\mathbf{u}^{\star}{}_{i=1:N-1|k-1})$ to obtain

$$
\left[\bar{\xi}_{0|k},\ ...,\ \bar{\xi}_{N|k}\right]^{\mathsf{T}} \equiv \bar{\mathcal{X}}\left(\hat{\xi}_{0|k},\ \Delta\mathbf{u}^{\star}{}_{i=1:N-1|k-1},\ \delta_k\right). \tag{12}
$$

which is an approximation of the state evolution from $i = 0$ to N. Hence, the predicted target vector

$$
\mathcal{X}_{ref} = \left[\xi_{ref\,0|k},\ ...,\ \xi_{ref\,N|k}\right]^{\mathsf{T}}, \tag{13}
$$

which is needed in (8f), is obtained from (11) as a function of $\bar{\mathcal{X}},\ \beta_{max},\ r_{max}$. This procedure is done at every time step before running the optimizer to populate $\xi_{ref,i|k}$ in (8f). In this way, the nonlinear complexity on $e_{i|k}$ is avoided because, inside the optimizer, the targets are no longer dependent on the current state evolution, and a numerical approximation is used instead.

Another simpler approach to solving the nonlinearity on the cost function is considering persistent targets i.e., $\xi_{ref\,i|k} = \xi_{ref\,0|k},\ i = 0, ..., N$. The drawback, in that case, is that the prediction of state deviation deteriorates. The importance of state target prediction is studied with numerical simulations in Section 4.

3.1.1. Solving the quadratic programming (QP) problem

By handling the nonlinearities on the cost function as previously explained, and properly selecting the weights $Q \succeq 0$, $R \succeq 0$ and $R_{\Delta u} \succ 0$, problem (8) turns into a quadratic programming problem.

The optimizer is initialized with estimates of current states in (8e) and previous input applied in (8d). The latter is the output coming from the torque allocation/anti-slip function.

The weights Q, R_u, and $R_{\Delta u}$ are used to normalize the cost function. They are defined in terms of maximum deviation of states, input u, and input variation Δu as

$$Q = \begin{bmatrix} \frac{1}{e^2_{\beta_{max}}} & 0 \\ 0 & \frac{1}{e^2_{r_{max}}} \end{bmatrix}, R_u = \frac{1}{M^2_{z,max}}, \;\; R_{\Delta u} = \frac{1}{\Delta M^2_{z,max}} \tag{14}$$

$$M_{z,max} = \left(|T_{RR,max}| + |T_{RL,max}|\right) \frac{b_R}{2R_w} \tag{15}$$

$$\Delta M_{z,max} = \left(|\Delta T_{RR,max}| + |\Delta T_{RL,max}|\right) \frac{b_R}{2R_w} \tag{16}$$

Problem (8) is a quadratic programming (QP) problem with only equality constraints. It is then reformulated, using the batch approach [19,20], in terms of the decision variable $\mathbf{\Delta u}$ by substituting the plant model (8b, 8c) into the objective function from $i = 0$ to $N - 1$ as

$$\mathcal{V}^{\star} \triangleq \min_{\mathbf{\Delta u}} \frac{1}{2} \mathbf{\Delta u}^{\mathsf{T}} \mathcal{M} \mathbf{\Delta u} + g\left(\hat{\xi}_k, \; \hat{u}_{k-1}, \; \delta_k, \; \mathcal{X}_{ref}\right)^{\mathsf{T}} \mathbf{\Delta u} + \bar{\mathcal{V}}, \tag{17}$$

where $\mathcal{M}$ is the Hessian matrix, $g(\cdot)$ define the linear term of the cost function, $\bar{\mathcal{V}}$ has no influence on the optimizer as it only affects the optimal value of (17). The expression (17) is a positive definite quadratic function of $\mathbf{\Delta u}$. Therefore, its minimum can be found by computing its gradient and setting it to zero. This yields the optimal vector of future input deviations:

$$\begin{aligned} \mathbf{\Delta u}^{\star} &= \mathcal{M}^{-1} g\left(\hat{\xi}_k, \; \hat{u}_{k-1}, \; \delta_k, \; \mathcal{X}_{ref}\right)^{\mathsf{T}} \\ &= \left[\Delta u_0^{\star}, \; \ldots, \; \Delta u_{N-1}^{\star}\right]^{\mathsf{T}} \end{aligned} \tag{18}$$

It should be noted that the linear term $g(\cdot)$ of the cost function is dependent on the current state estimates, the previously applied input, the actual steering angle, and the predicted evolution of the state targets. The state feedback controller acts differently than a pure feedback controller because it uses the measured steering disturbance and an approximation of the state target evolution to calculate the optimal yaw moment variation that penalizes the violation of the limits of handling. The vehicle dynamics is evolved inside the optimizer to obtain the optimal yaw moment variation Δu_0^*. Note that the optimal controller accounts for the overall yaw moment applied during the previous time interval, see $\hat{u}_{k-1}$ in equation (8d). The latter is needed because at each time step, the applied yaw moment is the combination of both the optimal controller and a feedforward controller actions.

3.2. Feedforward controller

The feedforward (FF) controller is designed to enable the vehicle to handle like a more compact sedan. To generate the feedforward control action, two vehicle models are used, one representing the desired vehicle dynamics and the other representing the original vehicle dynamics. Then the feedforward controller generates a yaw moment $M_{z,FF}$ that forces the vehicle to behave as the desired vehicle dynamics.

Figure 4 shows the models involved in obtaining the FF controller. It is projected in such a way that the yaw rate of the nominal model equals the yaw rate generated by the desired model when a steering disturbance is applied, i.e., $r_e = 0$.

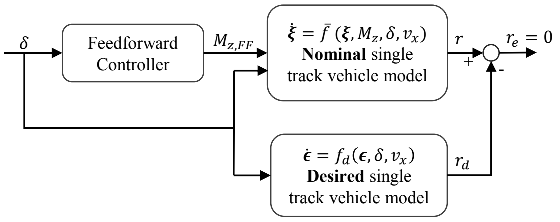

δ
Feedforward Controller
$M_{z,FF}$
$\dot{\xi} = \bar{f}\,(\xi, M_z, \delta, v_x)$
Nominal single track vehicle model
r
$r_e = 0$
+
-
$\dot{\epsilon} = f_d(\epsilon, \delta, v_x)$
Desired single track vehicle model
r_d

Figure 4. FF controller design. The feedforward controller generates a yaw moment $M_{z,FF}$ that steers the vehicle to behave as the desired vehicle dynamics.

The well known linear single-track vehicle model based on cornering stiffness coefficients [18, § 2.6].

$$\dot{\xi} \approx \tilde{A}(v_x)\xi + \tilde{B}M_{z,FF} + \tilde{B}_d\delta, \tag{19}$$

is preferred to design the FF controller. It is a simplified version of the nonlinear vehicle model (6), with matrices

$$\begin{aligned} \tilde{A} &= \begin{pmatrix} -\frac{C_{\alpha,F}+C_{\alpha,R}}{v_x m} & -\frac{\frac{C_{\alpha,F}l_F}{v_x} - \frac{C_{\alpha,R}l_R}{v_x}}{v_x m} - 1 \\ \frac{C_{\alpha,R}l_R}{J_z} - \frac{C_{\alpha,F}l_F}{J_z} & -\frac{C_{\alpha,F}l_F^2}{J_z v_x} - \frac{C_{\alpha,R}l_R^2}{J_z v_x} \end{pmatrix}, \\ \tilde{B} &= \begin{pmatrix} 0 \\ \frac{1}{J_z} \end{pmatrix} \tilde{B}_d = \begin{pmatrix} \frac{C_{\alpha,F}}{v_x m} \\ \frac{C_{\alpha,F}l_F}{J_z} \end{pmatrix}, \end{aligned} \tag{20}$$

with front and rear cornering stiffness values $C_{\alpha,F} = 117310$ N/rad and $C_{\alpha,R} = 58855$ N/rad, respectively. The remaining model parameters are detailed in Table 1.

The desired vehicle $f_d(\cdot)$ in Figure 4 is similarly formulated as

$$\dot{\epsilon} = f_d(\epsilon, \delta, v_x) = \bar{A}(v_x)\epsilon + \bar{B}(v_x)\delta, \tag{21}$$

where

$$\bar{A} = \begin{pmatrix} -\frac{\bar{C}_{\alpha,F}+\bar{C}_{\alpha,R}}{v_x \bar{m}} & -\frac{\frac{\bar{C}_{\alpha,F}\bar{l}_F}{v_x} - \frac{\bar{C}_{\alpha,R}\bar{l}_R}{v_x}}{v_x \bar{m}} - 1 \\ \frac{\bar{C}_{\alpha,R}\bar{l}_R}{\bar{J}_z} - \frac{\bar{C}_{\alpha,F}\bar{l}_F}{\bar{J}_z} & -\frac{\bar{C}_{\alpha,F}\bar{l}_F^2}{\bar{J}_z v_x} - \frac{\bar{C}_{\alpha,R}\bar{l}_R^2}{\bar{J}_z v_x} \end{pmatrix}, \; \bar{B} = \begin{pmatrix} \frac{\bar{C}_{\alpha,F}}{v_x \bar{m}} \\ \frac{\bar{C}_{\alpha,F}\bar{l}_F}{\bar{J}_z} \end{pmatrix},$$

with state vector $\epsilon = [\bar{\beta}\ \bar{r}]^\intercal$. The bar accent $(\bar{\cdot})$ over the parameters and states of the desired model means that they belong to a desired vehicle and hence they are not necessarily identical to those ones of the nominal vehicle model.

For the sake of clarity and without losing generality, at a specific time instant, the FF controller can be written using Laplace notation as

$$\begin{aligned} FF(s) = \frac{M_{z,FF}(s)}{\delta(s)} &= \frac{G_{des}(s) - G_{pd}(s)}{G_{pu}(s)} \\ &= C_{FF}(sI - A_{FF})^{-1}B_{FF} + D_{FF} \end{aligned} \tag{22}$$

where $G_{des}(s) = \frac{r_d(s)}{\delta(s)}$, $G_{pd}(s) = \frac{r(s)}{\delta(s)}$, $G_{pu}(s) = \frac{r(s)}{M_{z,FF}(s)}$.

Equation (22) is valid only in the vicinity of that operating point. Hence to achieve nominal zero error $r_e = 0$ in open loop, the FF controller actually results in a continuous linear-time-variant (LTV) state space model

$$\begin{aligned} \dot{\zeta}(t) &= A_{FF}(t)\zeta(t) + B_{FF}(t)\delta(t) \\ M_{z,FF} &= C_{FF}(t)\zeta(t) + D_{FF}(t)\delta(t). \end{aligned} \tag{23}$$

It is generated using Matlab$^{\text{TM}}$ symbolic tools as presented in Appendix A. At each time step k, the state space matrices are updated and exact discretization is then applied to obtain M_{z,FF_k}.

In this work, the desired vehicle model was designed with lower inertia ($\bar{J}_z = 0.75J_z$), which corresponds to a conventional compact sedan vehicle powered by ICE and without the high voltage battery and in-wheel motors. Figure 5 shows a Bode diagram of the FF controller at different vehicle speeds. The feedforward generates a yaw moment $M_{z,FF}$ that fulfils the control requirement $r_e = 0$ (see Figure). The feedforward reacts in open loop directly as a consequence of the applied steering angle.

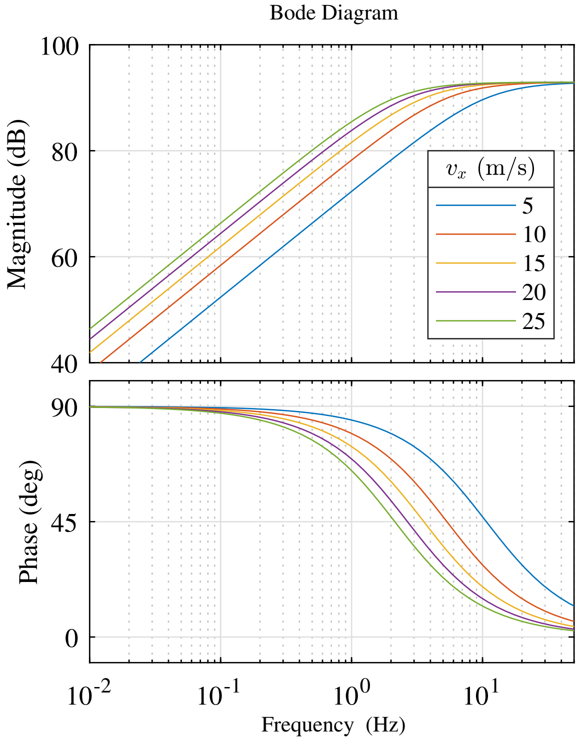


Figure 5. Bode diagram of feedforward controller: $FF(s) = \frac{M_{z,FF}(s)}{\delta(s)}$ obtained at different vehicle speeds. The desired vehicle model is formulated with lower yaw inertia than the nominal vehicle i.e., $\bar{J}_z = 0.75J_z$.

Only the variation in the inertia of the desired vehicle is investigated in this work; the remaining parameters of the desired model are set to those of the nominal model (19). A more in-depth investigation is reserved for future works.

Even if the feedforward controller works in open-loop mode depending merely on the steering angle, the yaw moment it generates ($M_{z,FF}$) is known by the state feedback controller (see Equation (8d) in problem (8)) and hence properly handled by the state feedback controller.

3.3. Combined model-kinematics-based estimation of sideslip angle

Estimating the vehicle sideslip angle β is crucial to improve vehicle stability. It is done by using a combined model-kinematics-based estimation approach i.e., the fusion of an extended Kalman filter (EKF) that estimates the sideslip ($\hat{\beta}$), with a kinematics-based formulation that estimates its time derivative $\hat{\dot{\beta}}_{kin}$. The EKF is model-based while the kinematics equation relies on direct measurements.

The mathematical reasoning behind the EKF algorithm is quite involved and beyond the scope of this work. The reader is directed to references [21], [22] for an extensive treatment. A simplified version of the EKF presented in [21] is proposed here as follows.

The nonlinear dynamics of the vehicle are known from (6) as a continuous-time model whereas the output vector:

$$z = [r,\ F_{yF},\ F_{yR}]^{\intercal}\,; \tag{24}$$

will be available at discrete instants of time. Literature refers that adding the estimated lateral forces ($F_{yF},\ F_{yR}$) as measured quantities improves the speed of the sideslip angle estimation. The calculation of the lateral forces is already presented in Equation (3). Equation (6) is then rewritten in the hybrid form [21]:

$$\begin{aligned}
\dot{\xi} &= f\left(\xi, u, \delta, p, \boldsymbol{w_n}, t\right), && \text{(25a)}\\
z_k &= g\left(\xi_k, u_k, \delta, p_k, \boldsymbol{v_{nk}}\right), && \text{(25b)}\\
\boldsymbol{w_n}(t) &\sim (0, \bar{Q}), && \text{(25c)}\\
\boldsymbol{v_{nk}} &\sim (0, \bar{R}) && \text{(25d)}
\end{aligned}$$

where the process noise $\boldsymbol{w_n}$ is continuous-time white noise with covariance $\bar{Q}$, and the measurement noise $\boldsymbol{v_n}$ is discrete-time white noise with covariance $\bar{R}$. They are used to tune the EKF estimates.

With model (25a)-(25b), the current state estimate is projected ahead in time, while the EKF measurement update step (used to adjust the projected estimate by an actual measurement at that time) is done using the measurement of the yaw rate (r_{meas}) and the estimation of lateral forces ($\hat{F}_{yF},\ \hat{F}_{yR}$) from the measured lateral acceleration a_y and yaw acceleration $\dot{r}_{meas}$, and from the vehicle parameters m, J_z, l_f, l_R:

$$\hat{F}_{yF} = \frac{m a_y l_R + J_z \dot{r}_{meas}}{l_F + l_R} \tag{26}$$

$$\hat{F}_{yR} = \frac{m a_y l_F - J_z \dot{r}_{meas}}{l_F + l_R} \tag{27}$$

The EKF steps to obtain the state estimates $\hat{x} = [\hat{\beta}_{EKF}\ \hat{r}_{EKF}]^{\intercal}$ are described in Appendix B.

The estimation of the sideslip using only the EKF can deteriorate due to plant-model mismatch. To avoid this, the kinematics formula:

$$\dot{\beta}_{kin} = \frac{a_y}{v_x} - r + g\sin\phi_r. \tag{28}$$

is quite often proposed to get a reliable time derivative of the sideslip angle i.e., $\dot{\beta}_{kin}$, from direct measurements of speed, lateral acceleration and roll angle ϕ_r. However, the direct integration of (28) to obtain $\hat{\beta}$ is prompt to errors and hence avoided in real implementations. In this work, Equation (28) is combined with the EKF as proposed in [23]:

$$\hat{\beta} = \frac{1}{\tau s + 1}\hat{\beta}_{EKF} + \frac{\tau}{\tau s + 1}\dot{\beta}_{kin}, \tag{29}$$

with $\tau = 10/(2\pi)$. By using the new estimation method, the need to do integration of the kinematics-based value is eliminated. Instead, a pseudo integral is used in (29) for the kinematics-based value. The reader is directed to reference [23] for an extensive treatment on this topic.

3.4. Torque distribution and anti-slip control

The yaw moment is applied by distributing the corresponding torque on each in-wheel motor of the rear axle. Firstly, the driver request is equally split to apply the same torque on each e-motor of the rear axle:

$$T_{bias} = \frac{T_{req}}{2} \tag{30}$$

The application of only T_{bias} on each rear wheel will guarantee the longitudinal force request. A torque quantity ΔT is added as follows:

$$T_{RR,ref} = T_{bias} + \Delta T, \tag{31}$$

$$T_{RL,ref} = T_{bias} - \Delta T, \tag{32}$$

and calculated as

$$\Delta T = \frac{M_z R_w}{2b_R}, \tag{33}$$

to generate the desired yaw moment M_z where

$$M_z = M_{z,FF} + M_{z,FB}, \tag{34}$$

which lumps both the feedforward yaw moment contribution $M_{z,FF}$ and the feedback control action $M_{z,FB}$. It is worth noticing that even if the two controllers are designed separately, problem (8) is updated every time step with the actual yaw moment that has been applied during the previous time interval.

3.4.1. Anti-slip control and yaw moment saturation

The differential torque needed to generate M_z is limited by the anti-slip control. The latter is based on a saturation function:

$$S_{sat} = S_{max} \tanh\left(\frac{S_{actual}}{S_{max}}\right), \tag{35}$$

that depends on the current longitudinal slip value S_{actual} of the wheel and a desired maximum wheel slip value S_{max} as shown in Fig. 6. A normalized reduction factor

$$\gamma = \frac{|S_{sat} - S_{actual}|}{|S_{sat}(S_{max})|};\ 0 \leq \gamma \leq 1 \tag{36}$$

is then calculated. The procedure is done in both the rear wheels to obtain $\gamma_{RL},\ \gamma_{RR}$, and the highest reduction factor is used to limit the differential torque as follows:

$$\Delta T_{sat} = (1 - \gamma_{max})\Delta T. \tag{37}$$

The saturated yaw moment:

$$u_{sat} = M_{z,sat} = \left(\frac{T_{bias} + \Delta T_{sat}}{R_w} - \frac{T_{bias} - \Delta T_{sat}}{R_w}\right) b_R, \tag{38}$$

is finally applied. The anti-slip control works as a time-varying yaw moment saturation stage that produces zero yaw moment when the condition $S_{actual} \geq S_{max}$ is met in any of the rear wheels. The benefit of this saturation approach is, first of all, its simplicity, since it depends merely on a unique tuning parameter: S_{max}. Note also that the anti-slip control acts only on the differential torque ΔT and hence does not affect the longitudinal tire force request.

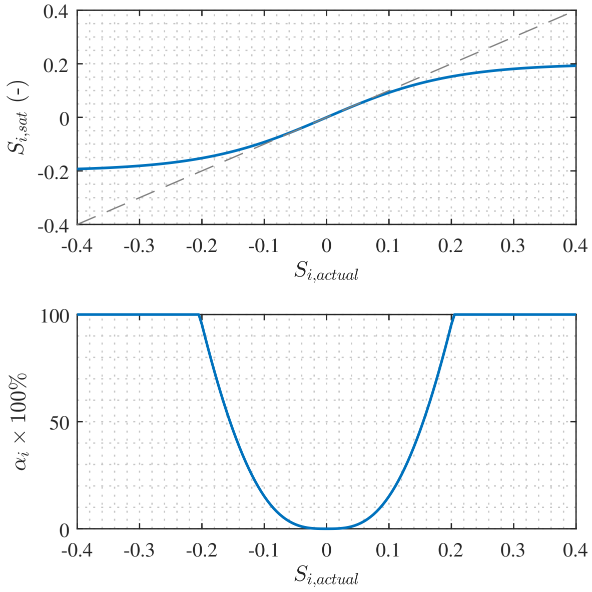


Figure 6. Slip saturation stage. The tire slip is limited to $S_{max} = 0.2$. Note that the limitation in torque is minimum when the tire slip is below 0.1. A normalized reduction factor α_i for each rear wheel ($i = RL,\ RR$) is calculated at each time step. They are used to limit the differential torque ΔT and hence the applied yaw moment as stated in Equation (38).

4. Numerical Results

The proposed torque vectoring controller is a combination of an optimal controller with a feedforward controller as explained in Section 3. Throughout all the simulations, both the controllers are executed at 50 Hz while the inner anti-slip control runs at 100 Hz. The optimal controller is set with a prediction horizon $N = 30$, maximum allowed deviations of the sideslip $e_{\beta_{max}} = \beta_{max}$ and yaw rate $e_{r_{max}} = r_{max}$. The input weights R_u and $R_{\Delta u}$ are set as detailed in (14) - (16) to properly normalize the objective function to give equal effort for same "badness". The maximum wheel torque values $T_{RL,max}$ and $T_{RR,max}$ are updated every time step with the torque boundary map of the in-wheel motors at the actual speed, while $\Delta M_{z,max}$ is set to 1000 Nm.

The torque vectoring controller is initially evaluated in simulation using open and closed-loop manoeuvres. Its performance is compared to that of an identical, passive vehicle, i.e., one with no TV controller intervention and equal power distribution on the rear axle.

4.1. Open loop manoeuvres

The steady-state performance evaluation is done with slow ramp steer manoeuvre (SRS) [24]. The steering wheel angle δ_{Sw} is increased from 0° to 100° with a slope of 1 deg/s at a constant speed of 100 km/h. The manoeuvre has been performed in low adherence conditions ($\mu = 0.5$) to better highlight the shaping of the understeering response in the proximity to the limits of handling.

The understeer curve is reported in Figure 7 (a). The proposed controller does not affect the steady-state vehicle response in the lateral acceleration range below 0.3 g, while at higher values of lateral acceleration, a more progressive response in the transition from the linear to the nonlinear range can be noticed. The peak value of the sideslip angle is reduced by 36 % for the controlled vehicle (Figure 7 (b)). The sideslip angle is progressively limited while approaching the lateral acceleration limits, improving rear-end stability. The state feedback control action is primarily responsible for the improvement of stability.

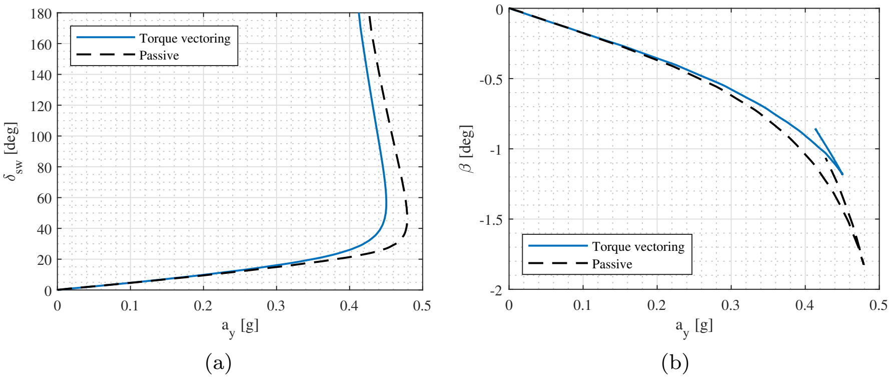


Figure 7. Numerical simulation of slow ramp steer (SRS) manoeuvre 100 km/h, low adherence conditions $\mu = 0.5$. Controlled vehicle = blue continuous line; passive vehicle = black dotted line. (a) Steering wheel angle δ_{Sw} vs. lateral acceleration a_y. (b) Vehicle sideslip angle β vs. lateral acceleration a_y.

The transient steering behaviour of the vehicle is evaluated through the sine sweep manoeuvre with increasing frequency (SSI) at the constant speed of 100 km/h [25]. The linear transfer function between the steering wheel angle δ_{Sw} input and the vehicle's yaw rate r output is obtained with the Welch's averaged periodogram method, then the frequency response function is evaluated, Figure 8. The aim of the test is to assess the performance of the vehicle and the influence of the controller in terms of lateral agility [1,12]. The torque vectoring controller leads to a different natural eigenfrequency $f(\frac{r_{max}}{\delta_{Sw}})$ with respect to the passive configuration. The resonance peak of the controlled vehicle is shifted towards a frequency 30 % higher. On the other hand, the torque vectoring controller configuration has a more underdamped response showing a higher magnitude of the resonance peak. Furthermore, by analyzing the phase response plot (Figure 8 b), the controlled vehicle shows a faster front-end response and a reduced phase delay for the frequency range below 1.1 Hz.

The agility enhancement of the vehicle related to the transient steering response evaluated through the SSI test is mainly provided by the feedforward control action.

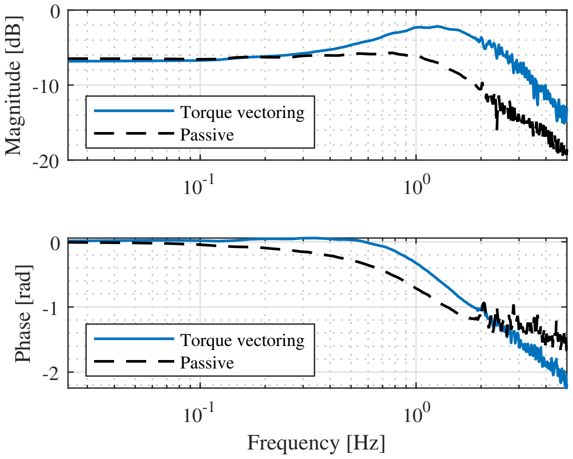


Figure 8. Numerical simulation of sine sweep manoeuvre (SSI) 100 km/h, dry asphalt $\mu = 1$. Frequency response function between δ_{Sw} input and r output. Controlled vehicle = blue continuous line; passive vehicle = black dotted line.

4.2. Closed loop manoeuvres

The performance of the controller is evaluated by following Standard [26]. An obstacle avoidance manoeuvre with a double lane change in a tight path is applied as shown in Figure 9. From the analysis of the Lissajous diagram between the steering wheel angle δ_{sw} and the yaw rate r reported in Figure 9 (a), the hysteresis of the yaw rate response to the steering input is reduced in the case of the controlled vehicle. The action provided by the feedforward control contribution leads to a more linear relationship between the steering angle and the yaw response. The controlled vehicle shows a yaw response closer to the reference one (3.2), as far as a reduced steering wheel effort is required by the driver to overcome the manoeuvre. The yaw moment contribution provided by the state feedback regulator is more evident when the vehicle approaches the stability limits, as it emerges from Figure 9 (b). The sideslip angle β of the controlled vehicle is reduced throughout the manoeuvre (Figure 9 (d)). The peak value of the sideslip angle is reduced by 29 % when torque vectoring is enabled. The simulated trajectory (Figure 9 (c)) shows that the controlled vehicle can better follow the reference trajectory (goodness of fit of 64 % for the controlled vehicle and 61 % for the passive).

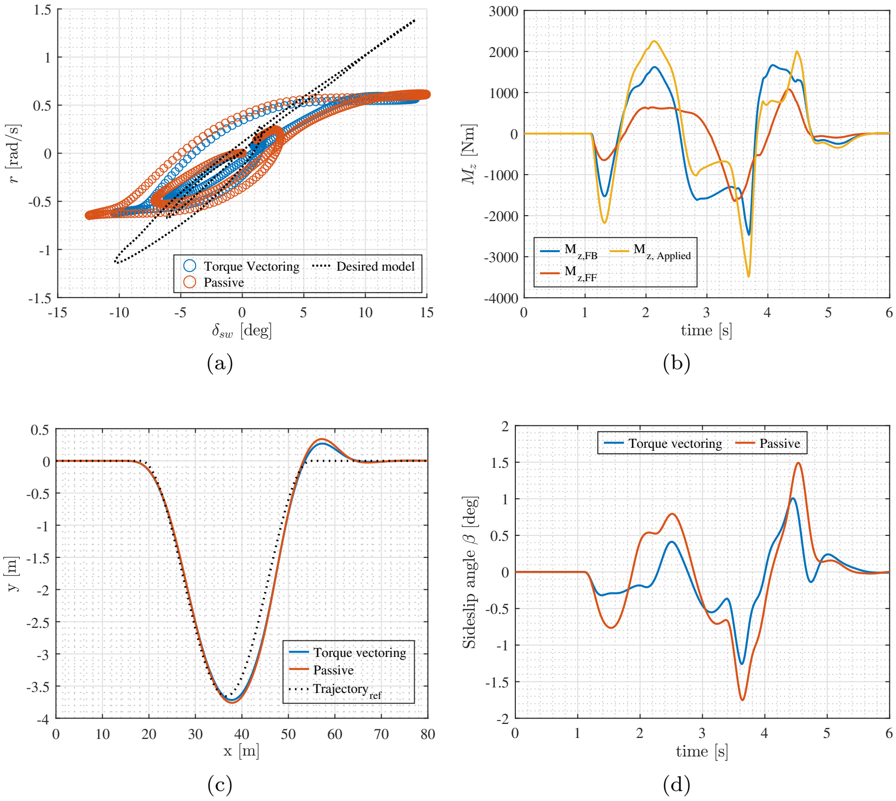


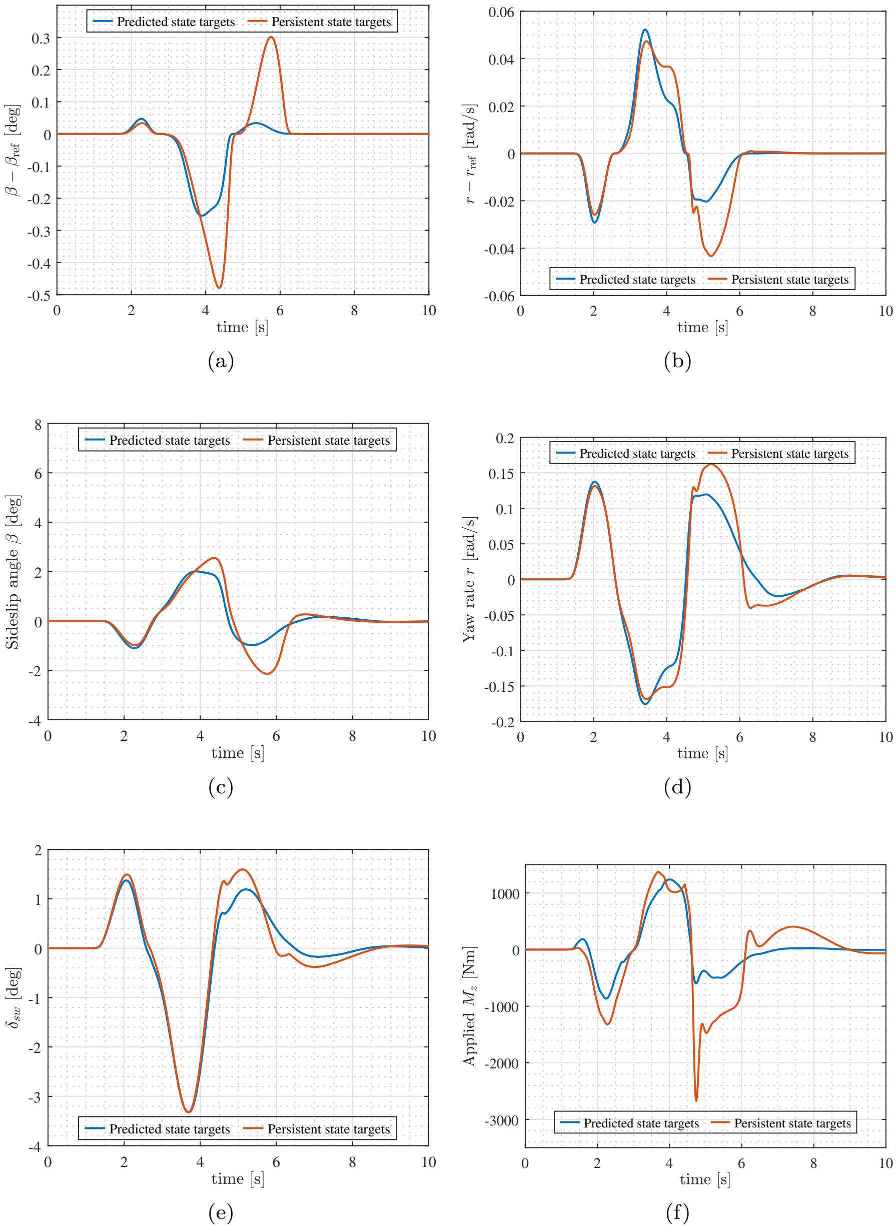
Figure 9. Numerical simulations of obstacle avoidance manoeuvre with a double lane change. Vehicle speed: 50 km/h, tight path, $\mu = 1$. Controlled vehicle = blue continuous line; passive vehicle = orange continuous line. a) Lissajous diagram of steering wheel angle and yaw rate. b) Yaw moment contributions vs time. c) Vehicle trajectory. d) Sideslip angle β vs. time.

4.3. Assessment of the state target prediction

The limit's handling monitoring is assessed in closed-loop manoeuvre through double lane change at 120 km/h. To highlight the stability improvement provided by the prediction of the state targets, the manoeuvre has been performed in low friction conditions ($\mu = 0.5$). In this condition the passive vehicle is not able to accomplish the manoeuvre, the vehicle spins out of the trajectory.

The simulation results are reported in Figure 10. Two different configurations of the optimal state feedback controller are reported: "predicted state target" where the evolution of the state targets is considered in a predicted target vector (13), and "persistent target" with a constant target vector for the entire time horizon.

The torque vectoring controller with the "predicted state target" configuration reduces the state deviation (8f): up to 51 % for the sideslip angle β, and up to 45 % for the yaw rate r when compared with the "persistent state target" configuration (figure 10 a,b). The peak values of the vehicle sideslip angle and the yaw rate are reduced (figure 10 c,d) and get closer to the stability limit reference state targets (11), thus the stability performance of the vehicle in the simulated manoeuvre is enhanced. Furthermore, as shown in Figure 10 (f), the applied corrective yaw moment (38) is much smoother and faster when the prediction of the state target evolution is used in comparison to the configuration with "persistent state target," resulting in a faster generation of the yaw rate (figure 10 d) and less steering wheel effort required by the driver (figure 10 e).



(a)

(b)

(c)

(d)

(e)

(f)

Figure 10. Numerical simulation of obstacle avoidance manoeuvre with a double lance change. Vehicle speed 120 km/h, normal path, low adherence road conditions ($\mu = 0.5$). Comparison between optimal state feedback controller with state targets prediction (blue continuous line) and with persistent state targets (orange continuous line). a) Sideslip angle β state deviation. b) Yaw rate r state deviation. c) Sideslip angle β vs. time. d) Yaw rate r vs. time. e) Steering wheel angle δ_{sw} vs. time. f) Applied yaw moment vs. time.

5. Hardware in the Loop with e-axle

The methodology is validated through the hardware-in-the-loop testing on a devoted rig realized within the HiPERFORM project at the CARS research center of the Politecnico di Torino, in collaboration with the project partners. HIL aims to fine-tune the motor control and validate the reliability of the torque vectoring control strategy in real time. In addition, HIL ensured the efficient execution of the designed control scheme in real time. The vehicle, driver model, and control strategy were deployed to a modular real-time system (dSpace SCALEXIO Autobox). The rig consists of two in-wheel motors, a dual inverter, and an e-axle test bench equipped with two dynamometers. The rig mimics the wheels and the layout is shown in Figure 11. The wheel speed ($\omega_{RL,ref}, \omega_{RR,ref}$) is imposed by the vehicle model onto the dynamometer and the actual speed ($\omega_{RL,act}, \omega_{RR,act}$) is fed back to the TV controller. The torque references ($T_{RL,ref},\ T_{RR,ref}$) are set by the torque vectoring controller (driver model) in case of torque vectoring equipped vehicle (Passive vehicle). The actual torque ($T_{RL,act}, T_{RR,act}$) measured in the rig is fed back to the vehicle dynamics model. CAN protocol is used to interface all signals between the rig and the Modular real-time system.

To this end, an exhaustive testing campaign is carried out, and two tests are reported: a) An obstacle avoidance manoeuvre involving a double lane change on a tight path at a speed of 50 km/h and a friction coefficient of 0.85. b) An obstacle avoidance manoeuvre with a double lane change at a speed of 120 km/h and a friction coefficient of 0.85. The key performance indices for the tests are still the body sideslip angle (β), yaw rate (r), and steering angle (δ) but this time in a more realistic scenario with real in-wheel motors, a dual inverter and a SCALEXIO AutoBox hardware. For the HiL test, a high $\mu = 0.85$ was used in order to reduce tire slide and hence avoid wheel torque saturation inside the anti-slip control.

The test campaign started with the integration of the dual inverter and e-motors, the calibration of the motor position sensors and the torque controller. The torque commands ($T_{RL,ref},\ T_{RR,ref}$) are sent every 10 ms via CAN. A slew rate limit set to 500 Nm/s inside the dual-inverter was enough to guarantee the torque references in all driving scenarios. A maximum communication latency of almost 20 ms was present on the test bench but it did not affect the vehicle's performance.

Figure 12a reports the performance with the TV controller enabled and disabled (passive vehicle) in an obstacle avoidance manoeuvre involving a double lane change on a tight path at a speed of 50 km/h and a friction coefficient of 0.85. It is evident from the plots that the TV vehicle outperforms the passive vehicle. The body sideslip angle in the TV vehicle is reduced by 54 % while the yaw rate is reduced by 4.8 %, and the steering effort is reduced by 14 %. A similar benefit can be observed in a double lane change manoeuvre at 120 km/h and a friction coefficient of 0.85 as shown in Figure 12b. The steering wheel angle is set in open loop, which means the driver model is absent from the loop and a previously recorded steering wheel angle is sent to the TV controller and vehicle dynamics model. The test yielded in reduction of body sideslip angle by 32.5 % and yaw rate by 11.4 %.

6. Conclusions

A novel torque vectoring control method to enhance vehicle manoeuvrability and predict handling limit violation has been investigated in this study. An adaptive predictive state feedback controller generates a correcting yaw moment when the vehicle tends to move beyond the handling limits. It acts in combination with a feedforward controller. The latter is designed to improve vehicle agility using the front steering angle information. The resulting yaw moment is converted into torque references for the two

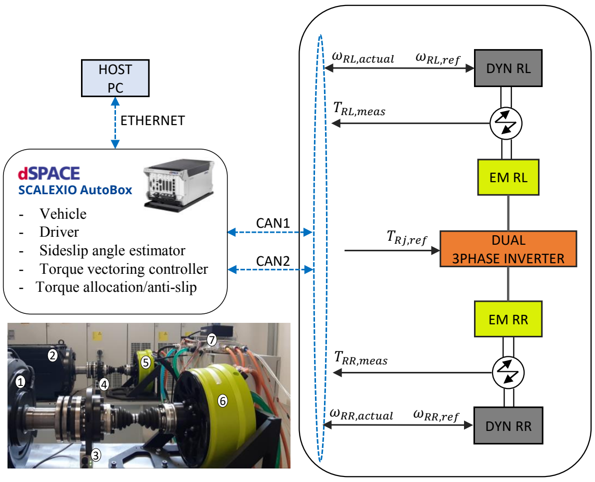


Figure 11. Hardware in the loop scheme with a picture of the test bench. EM: e-motors (5, 6), DYN: Dynamometers (1, 2), HBM torque sensors (3, 4). Dual inverter (7). CAN communication: CAN1 (10 ms): SCALEXIO ↔ e-axle testbench. CAN2 (10 ms): SCALEXIO ↔ Dual Inverter. Ethernet: SCALEXIO ↔ Host PC for real-time monitoring and data acquisition. RR: Rear Right, RL: Rear Left, ref: reference, meas: measured, ω: Speed, T: Torque.

rear in-wheel motors installed on the vehicle. The torque vectoring scheme is discussed in detail including also a sideslip angle estimator, a torque allocation method and the anti-slip control.

The contribution of the present study is twofold: first, the way the vehicle handling limits are dealt with. In fact, violations of vehicle handling limits are directly written into the objective function of an optimal control problem that predicts vehicle evolution. It resulted in a nonlinear formulation, which has been approximated to a quadratic programming problem with no inequality constraints. Its implementation does not require a dedicated solver, making it a promising candidate for real-time applications. Second: the improvement of vehicle agility using a dynamic feedforward controller. Typically, the literature proposes torque vectoring schemes with yaw rate targets that change proportionally with the steering commands i.e., yaw rate targets purely based on the evaluation of the well-known bicycle model at steady-state to guarantee desired steering characteristics: understeering, neutral steering or oversteering. Instead, we proposed a dynamic feedforward controller written as an adaptive state-space system that considers also the dynamics of both actual and desired vehicle models.

Several numerical simulations were carried out. Firstly, the steady-state performance evaluation is done with slow ramp steer manoeuvre showing a sideslip reduction of up to 36 %. More aggressive manoeuvres were conducted to assess the torque vectoring controller in terms of vehicle handling and stability performance showing a significant improvement when torque vectoring is applied. In critical conditions where a passive vehicle is unstable, the proposed torque vectoring scheme is able to stabilize the vehicle. In general, the active safety and the overall stability of the vehicle were improved in low adherence conditions.

Furthermore, the benefits of using predictive state target evolution are demonstrated. When the handling limits are monitored using predicted vehicle evolution, the deviation of the vehicle states beyond the identified limits of sideslip angle β and yaw rate r is significantly reduced. It is worth noting that this is accomplished by

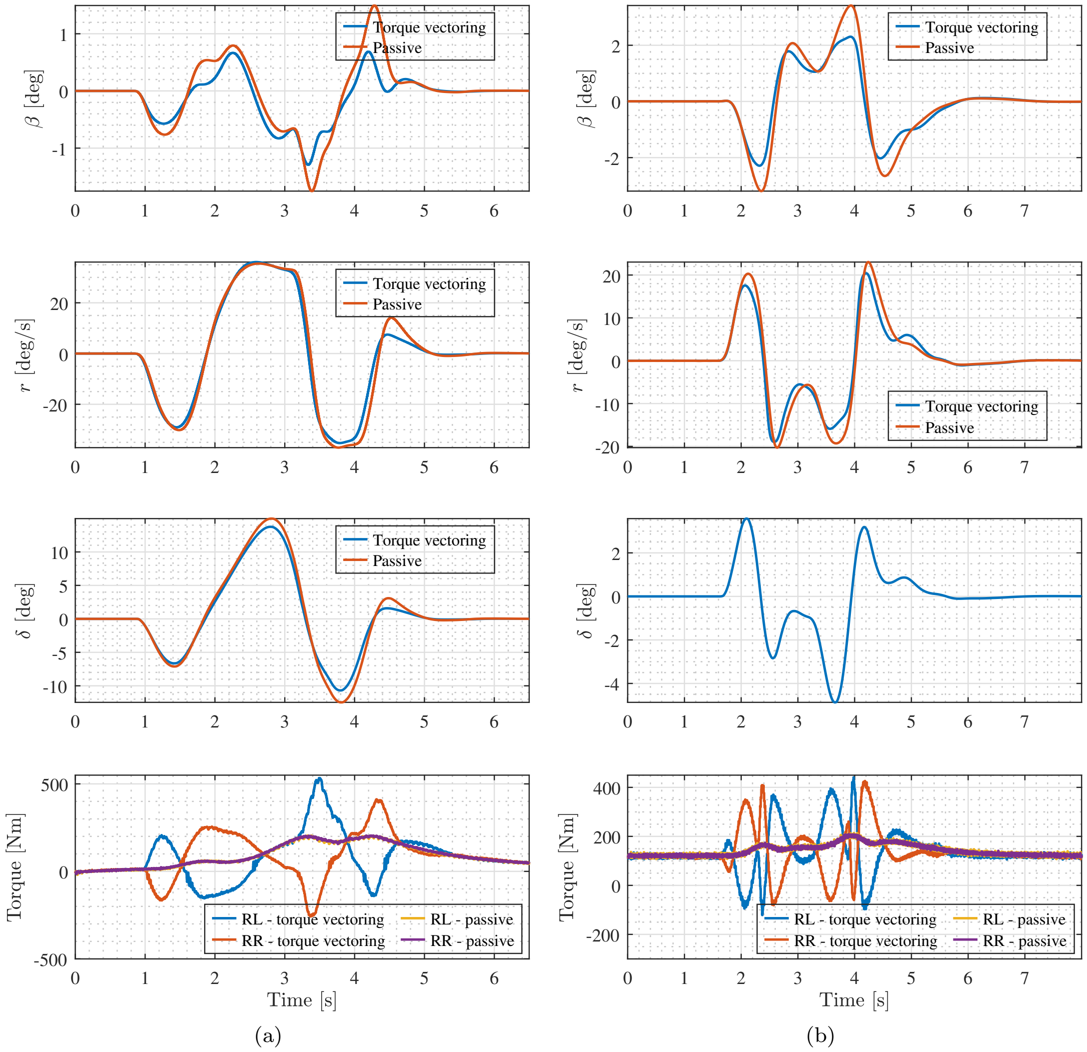


Figure 12. Hardware in the loop with and without torque vectoring (Passive), road friction coefficient $\mu = 0.85$. Type of test: Obstacle avoidance manoeuvre with a double lane change at different conditions: a) vehicle speed: 50 km/h in a tight path b) Vehicle speed: 120 km/h, normal path and no driver in the loop i.e., the same steering disturbance δ is applied in both cases.

generating a lower active yaw moment (i.e., smoother control action) when compared to the case of persistent target generation.

Finally, the hardware in the loop with the e-axle test campaign allowed for validating the deployability of the control algorithm, the feasibility of generating the required yaw moment in terms of e-motor torque references, and the control method reliability when integrated into the CAN communication network with the dual inverter.

Future work will focus on fine-tuning the vehicle model and observer using experimental data collected from a real vehicle, as well as assessing how a real driver feels when using the proposed torque vectoring controller.

Acknowledgement(s)

Authors sincerely thank the assistance they received from HiPERFORM project partners, particularly Ideas & Motion for their assistance with the e-motor control calibration and the setup of the dual inverter they provided; Elaphe for the two in-wheel motors they made available; and Encopim for the e-axle test bench they provided to conduct the experimental tests.

Disclosure statement

There are no relevant financial or non-financial competing interests to report.

Funding

This research has received funding from the ECSEL Joint Undertaking (JU) under grant agreement No 783174.

References

[1] Mazzilli V, De Pinto S, Pascali L, et al. Integrated chassis control: Classification, analysis and future trends. Annual Reviews in Control. 2021;51:172–205.

[2] Lugner P. Vehicle dynamics of modern passenger cars. Springer; 2019.

[3] De Novellis L, Sorniotti A, Gruber P, et al. Comparison of feedback control techniques for torque-vectoring control of fully electric vehicles. IEEE Transactions on Vehicular Technology. 2014;63(8):3612–3623.

[4] El Rifai K. Nonlinearly parameterized adaptive pid control for parallel and series realizations. In: 2009 American Control Conference; IEEE; 2009. p. 5150–5155.

[5] Siampis E, Massaro M, Velenis E. Electric rear axle torque vectoring for combined yaw stability and velocity control near the limit of handling. In: 52nd IEEE conference on Decision and Control; IEEE; 2013. p. 1552–1557.

[6] Vignati M, Sabbioni E, Tarsitano D. Torque vectoring control for iwm vehicles. International Journal of Vehicle Performance. 2016;2(3):302–324.

[7] Cheli F, Melzi S, Sabbioni E, et al. Torque vectoring control of a four independent wheel drive electric vehicle. In: International Design Engineering Technical Conferences and Computers and Information in Engineering Conference; Vol. 55843; American Society of Mechanical Engineers; 2013. p. V001T01A003.

[8] de Carvalho Pinheiro H, Carello M, Punta E. Torque vectoring control strategies comparison for hybrid vehicles with two rear electric motors. Applied Sciences. 2023;13(14):8109.

[9] Liang Z, Zhao J, Dong Z, et al. Torque vectoring and rear-wheel-steering control for vehicle's uncertain slips on soft and slope terrain using sliding mode algorithm. IEEE Transactions on Vehicular Technology. 2020;69(4):3805–3815.

[10] Goggia T, Sorniotti A, De Novellis L, et al. Integral sliding mode for the torque-vectoring control of fully electric vehicles: Theoretical design and experimental assessment. IEEE Transactions on Vehicular Technology. 2014;64(5):1701–1715.

[11] Kaiser G, Holzmann F, Chretien B, et al. Torque vectoring with a feedback and feed forward controller-applied to a through the road hybrid electric vehicle. In: 2011 IEEE Intelligent Vehicles Symposium (IV); IEEE; 2011. p. 448–453.

[12] Warth G, Frey M, Gauterin F. Design of a central feedforward control of torque vectoring and rear-wheel steering to beneficially use tyre information. Vehicle System Dynamics. 2020;58(12):1789–1822.

[13] Parra A, Tavernini D, Gruber P, et al. On nonlinear model predictive control for energy-efficient torque-vectoring. IEEE Transactions on Vehicular Technology. 2020;70(1):173–188.

[14] Jalali M, Khajepour A, Chen Sk, et al. Integrated stability and traction control for electric vehicles using model predictive control. Control Engineering Practice. 2016;54:256–266.

[15] Rubin D, Arogeti SA. Vehicle yaw stability control using active limited-slip differential via model predictive control methods. Vehicle System Dynamics. 2015;53(9):1315–1330.

[16] Siampis E, Velenis E, Gariuolo S, et al. A real-time nonlinear model predictive control strategy for stabilization of an electric vehicle at the limits of handling. IEEE Transactions on Control Systems Technology. 2017;26(6):1982–1994.

[17] Kiencke U, Nielsen L. Automotive control systems: for engine, driveline, and vehicle ; 2000.

[18] Rajamani R. Vehicle dynamics and control. Springer Science & Business Media; 2011.

[19] Goodwin G, Seron MM, De Doná JA. Constrained control and estimation: an optimisation approach. Springer Science & Business Media; 2006.

[20] Borrelli F, Bemporad A, Morari M. Predictive control for linear and hybrid systems. Cambridge University Press; 2017.

[21] Simon D. Optimal state estimation: Kalman, h infinity, and nonlinear approaches. John Wiley & Sons; 2006.

[22] Welch G, Bishop G. An introduction to kalman filter. In: International Conference on Computer Graphics and Interactive Techniques; 1995.

[23] Piyabongkarn D, Rajamani R, Grogg JA, et al. Development and experimental evaluation of a slip angle estimator for vehicle stability control. IEEE Transactions on control systems technology. 2008;17(1):78–88.

[24] ISO I. 4138: Passenger cars–steady-state circular driving behaviour–open-loop test methods. ISO: Geneva, Switzerland. 2012;.

[25] for Standardization IO. Road vehicles-lateral transient response test methods-open-loop test methods. International Organization for Standardization; 2011.

[26] ISO. 3888: International organization for standardization: Passenger cars test track for a severelane-change manoeuvre. part 1: Double lane-change. ISO: Geneva, Switzerland. 1999;.

Appendix A. Feedforward controller design

Matlab$^{\mathrm{TM}}$ symbolic tools are used to obtain the time-varying feedforward controller:

$$FF(s) = \frac{\frac{\bar{a}_{21}\,\bar{b}_{12}}{\bar{a}_{12}\,\bar{a}_{21}-\bar{a}_{11}\,\bar{a}_{22}+\bar{a}_{11}\,s+\bar{a}_{22}\,s-s^2} - \frac{\tilde{a}_{21}\,\tilde{b}_{12}}{\sigma_1} - \frac{\bar{b}_{22}\,(\bar{a}_{11}-s)}{\bar{a}_{12}\,\bar{a}_{21}-\bar{a}_{11}\,\bar{a}_{22}+\bar{a}_{11}\,s+\bar{a}_{22}\,s-s^2} + \frac{\tilde{b}_{22}\,(\tilde{a}_{11}-s)}{\sigma_1}}{\frac{\tilde{a}_{21}\,\tilde{b}_{11}}{\sigma_1} - \frac{\tilde{b}_{21}\,(\tilde{a}_{11}-s)}{\sigma_1}} \tag{A1}$$

with $\sigma_1 = \tilde{a}_{12}\,\tilde{a}_{21} - \tilde{a}_{11}\,\tilde{a}_{22} + \tilde{a}_{11}\,s + \tilde{a}_{22}\,s - s^2$. Terms $(\tilde{a}_{ij},\ \tilde{b}_{ij})$, $(\bar{a}_{ij},\ \bar{b}_{ij})$ are the matrix elements of the corresponding state space models (19) and (21), respectively.

Only the variation of yaw moment inertia is evaluated in this work however note that all the coefficients involved in the desired model are modifiable as desired.

Appendix B. EKF

At time instant k, the extended Kalman filter uses the following partial derivatives evaluated at the current operating point:

$$
\begin{aligned}
F &= \left.\frac{\partial f}{\partial \xi}\right|_{\substack{\xi=\hat{\xi}_k^- \\ u=M_{z\,k-1} \\ \delta=\delta_k \\ p=p_k}}, \; W = \frac{\partial f}{\partial \boldsymbol{w_n}} = I_{2\times 2} \\
H &= \left.\frac{\partial g}{\partial \xi}\right|_{\substack{\xi=\hat{\xi}_k^- \\ u=M_{z\,k-1} \\ \delta=\delta_k \\ p=p_k}}, \; V = \frac{\partial g}{\partial \boldsymbol{v_n}} = I_{3\times 3}
\end{aligned}
$$

with $f(\cdot)$ and $g(\cdot)$ given by Equations (25a)-(25b). The EKF algorithm is executed as follows:

(1) *EKF Initialization:* Set the initial estimates for the states and its covariance error:

$$
\begin{aligned}
\hat{\xi}_0^+ &= E(\xi_0) \\
P_0^+ &= E[(\xi_0 - \xi_0^+)(\xi_0 - \xi_0^+)^T]
\end{aligned}
$$

(2) *EKF time update Equations:* Integrate the state estimate and its covariance from time $(k-1)^+$ to time k^-:

$$
\begin{aligned}
\dot{\hat{\xi}} &= f\left(\hat{\xi}, u, \delta, p, 0\right) \\
\dot{P} &= -PH^\intercal \bar{R}^{-1} HP + FP + PF^\intercal + \bar{Q}
\end{aligned}
$$

The integration starts with $\hat{\xi} = \hat{\xi}_{k-1}^+$ and $P = P_{k-1}^+$. At the end of the integration we have $\hat{\xi} = \hat{\xi}_k^-$ and $P_k = P_k^-$. The predicted output is

$$
z_k^- = g\left(\hat{\xi}_k^-, u_{k-1}, \delta_k, p_k, 0\right)
$$

(3) *EKF measurement update Equations:*

$$
\begin{aligned}
K_k &= P_k^- H_k^\intercal \left(H_k P_k^- H_k^\intercal + V_k \bar{R}_k V_k^\intercal\right)^{-1} \\
\hat{\xi}_k^+ &= \hat{\xi}_k^- + K_k \left(z_k - z_k^-\right) \\
P_k^+ &= \left(I - K_k H_k\right) P_k^-
\end{aligned}
$$

From step 3 we have the state estimates: $[\hat{\beta}_{EKF},\ \hat{r}_{EKF}]^\intercal = \hat{\xi}_k^+$.

Step 1) is executed once during the initialization whereas 2) and 3) are repeated at each time step. All the terms with a *plus* superscript means that they have been already corrected by the measurement (i.e., step 3) completed) while the *minus* means only the update from the known equations (i.e., just after step 2) is completed). The term P_k is the error covariance matrix of the state estimate at time step k.