Lyapunov-based Fault Tolerant Control Allocation

Ricardo de Castro & Jonathan Brembeck

This work addresses the design of an active fault-tolerant motion controller for overactuated road vehicles. Our concept revolves around a control allocation framework, extended with Lyapunov-based stability constraints and costs. Besides the ability to cope with actuator constraints and optimal actuator reconfiguration, the proposed Lyapunov-based control allocation (LCA) allows the graceful degradation of control performance. Theoretical analysis and simulation results demonstrate that, in comparison with classical control allocation, the proposed LCA reduces tracking errors of the motion controller in the aftermath of actuator faults. Experimental tests carried out with the ROboMObil, a robotic electric vehicle prototype with wheel-based steering and traction actuation, demonstrate the effectiveness of the LCA

Ricardo de Castro & Jonathan Brembeck (2021) Lyapunov-based fault tolerant control allocation, Vehicle System Dynamics, DOI: 10.1080/00423114.2021.1971265

Lyapunov-based Fault Tolerant Control Allocation

ARTICLE HISTORY
Compiled June 10, 2021

ABSTRACT
This work addresses the design of an active fault-tolerant motion controller for over-actuated road vehicles. Our concept revolves around a control allocation framework, extended with Lyapunov-based stability constraints and costs. Besides the ability to cope with actuator constraints and optimal actuator reconfiguration, the proposed Lyapunov-based control allocation (LCA) allows the graceful degradation of control performance. Theoretical analysis and simulation results demonstrate that, in comparison with classical control allocation, the proposed LCA reduces tracking errors of the motion controller in the aftermath of actuator faults. Experimental tests carried out with the ROboMObil, a robotic electric vehicle prototype with wheel-based steering and traction actuation, demonstrate the effectiveness of the LCA.

KEYWORDS
Road Vehicles, Fault-tolerant Control, Control Allocation

1. Introduction

High levels of vehicle automation and electrification are introducing more sensing and drive-by-wire actuation in road vehicles [1]. This raises integration complexity and likelihood of malfunctions, which can compromise the ability of the vehicle sub-systems to safely perform their desired functions, such as steering, braking or accelerating. If these faults are not timely diagnosed and mitigated, hazardous events might arise. For example, [2] demonstrated that electrical and mechanical faults in the vehicle actuation can quickly lead to lane departures and collisions with other road participants. Full vehicle autonomy, e.g., SAE level 5 [3], is also expected to raise utilization rates of vehicles and availability demands, e.g., autonomous trucks or robotic taxis operating 24/7. To fulfill these high availability demands, it is crucial to incorporate fault-tolerant control (FTC) strategies in automated vehicles.

One key aspect for the deployment of FTC strategies is over-actuation, where redundant or back-up actuators are installed in the vehicle to tolerate faults. In road vehicles, several types of redundant propulsion, steering and braking actuation have been investigated. For example, redundant propulsion can be obtained through the inclusion of traction motors in the front and rear axles [4], multiple motors inside the wheel's hub [5] or close to the wheel [6]. Steering redundancy can be achieved through all-wheel steering with axle or single-wheel actuation [7], while braking redundancy can be implemented with a combination of regenerative and friction braking [8]. Thanks to this actuation redundancy, FTC strategies can be deployed in road vehicles. In the aftermath of fault(s), FTC re-distributes actuation effort among the remaining healthy actuators, enabling vehicle motion with normal or reduced performance. Additionally, redundant propulsion/steering actuation also presents opportunities to improve vehicle stability (e.g., active yaw-rate control [6]) and maneuverability (e.g., parallel vehicle motion enabled by four-wheel steering [9]). On the negative side, it is important to note that the inclusion of redundant actuators increases costs and integration complexity of road vehicles.

This work focuses on (active) FTC strategies, where the vehicle's motion controller is reconfigured based on a control allocation (CA) framework. As discussed in [10], CA aggregates the total control effect produced by the actuators into virtual control inputs, providing a dedicated (virtual) control channel for each degree of freedom of the vehicle, such as center-of-gravity accelerations or the yaw-moment. The inclusion of these virtual control inputs paves the way for a two-layer control architecture, composed of a high-level motion controller and a low-level CA. The former handles the controller's tracking goals (typically related to vehicle velocity [11] or trajectory [12]) through the manipulation of a small number of virtual inputs, while the latter maps the desired virtual inputs into a larger number of physical actuators.

This two-layer architecture offers several advantages. First, it breaks down the control problem into two sub-problems, which are simpler to solve than the original control problem, where tracking and allocation goals are simultaneously handled. Second, the high-level controller can be designed independently from the vehicle's actuation configuration, which furthers improves modularity. In particular, the low-level CA can be adapted to specific actuation configurations (e.g., 2-wheel steering, 4-wheel steering, torque vectoring, etc.), while keeping the high-level motion control algorithm unchanged [10]. Third, the CA can also effectively deal with the actuation saturations and promote allocation strategies that pursue secondary control objectives, such as minimization of power consumption [13], tire slip or tire workload [14]. Fourth, the CA can explore the actuation redundancy to reconfigure itself when failures occur, e.g., by mapping the virtual input to the remaining healthy actuators [15], without requiring modification of the high-level controller. As a result, CA has been used as a reconfiguration mechanism for fault-tolerant operation in automotive applications, especially to deal with brake [16], steer [17] and traction motor faults [18].

Physical infeasibility of virtual input in the aftermath of faults is one of the main challenges when designing CA-based FTC. In this case, the CA might be unable to fulfill the requested virtual inputs, which might degrade the performance of the high-level controller and lead to vehicle instability. To attenuate this issue, some CA approaches adapt the weights of the CA costs as a mean to discourage the allocator from using faulty actuators [19]. This adaption provides stability guarantees of the overall motion controller but under the restrictive assumption of unconstrained actuation. Another approach is to penalize the virtual input error, i.e., the difference between the requested virtual input and the physically attainable virtual input, in the CA cost function [10]. While intuitive at first glance, these error-based formulations do not explicitly consider stability or transient criteria in the allocation strategy.

Spurred by these shortcomings, we propose an alternative approach. Our main contribution is to augment the CA formulation with: i) an additional constraint, based on Lyapunov stability analysis of the high-level motion controller, and ii) an additional cost term that penalizes violation of the Lyapunov stability constraint. The resulting allocation strategy, called Lyapunov CA (LCA), brings several advantages in comparison with classical CA formulations. For example, the LCA's stability constraint promotes allocation strategies that do not destabilize the operation of the high-level motion controller. Additionally, the LCA's additional cost seeks to gracefully degrade the performance of the high-level controller when faults are present, and the virtual input is not physically attainable. This superior LCA performance is proved via the-

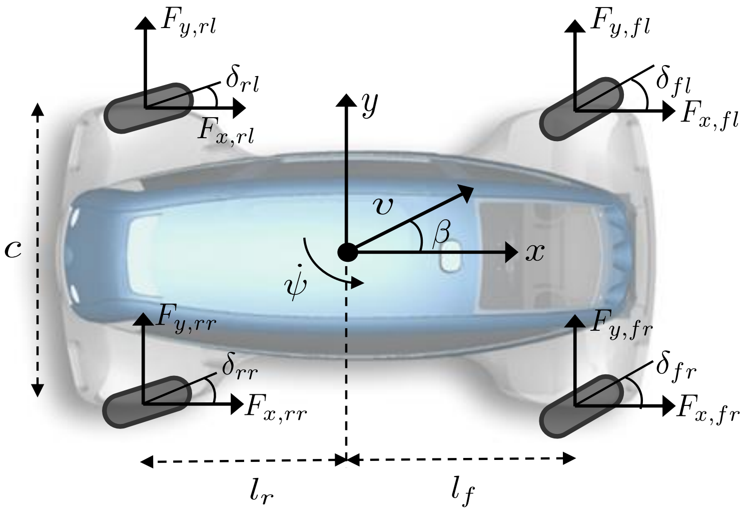


Figure 1. Planar vehicle model.

oretical and simulation analysis.

The second contribution of this work is the experimental validation of the LCA algorithm. Specifically, we implemented the LCA in the RoboMobil, an over-actuated electric vehicle with in-wheel steering and propulsion actuators [20,21], and validated its effectiveness when the vehicle is subject to faulty actuators. A preliminary version of this work was presented at [11,22]; it is extended here with additional theoretical analysis of the LCA algorithm and with experimental validation.

2. Model

This section presents the vehicle model employed in the design of the fault-tolerant motion controller. As a starting point, we consider a planar vehicle model, with longitudinal (x), lateral (y) and yaw (ψ) degrees of freedom [23] and wheel-level actuation:

$$\dot{v} = \frac{\cos(\beta)}{m}\sum_{i\in\mathcal{I}} F_{x,i}(T_i,\delta_i,v,\beta,\dot{\psi}) + \frac{\sin(\beta)}{m}\sum_{i\in\mathcal{I}} F_{y,i}(T_i,\delta_i,v,\beta,\dot{\psi}) \quad (1a)$$

$$\dot{\beta} = -\dot{\psi} + \frac{\cos(\beta)}{mv}\sum_{i\in\mathcal{I}} F_{y,i}(T_i,\delta_i,v,\beta,\dot{\psi}) - \frac{\sin(\beta)}{mv}\sum_{i\in\mathcal{I}} F_{x,i}(T_i,\delta_i,v,\beta,\dot{\psi}) \quad (1b)$$

$$\ddot{\psi} = \frac{1}{I_z}\sum_{i\in\mathcal{I}} b_{x,i}F_{x,i}(T_i,\delta_i,v,\beta,\dot{\psi}) + b_{y,i}F_{y,i}(T_i,\delta_i,v,\beta,\dot{\psi}) \quad (1c)$$

where v is the vehicle speed, β the body side-slip angle and $\dot{\psi}$ the yaw-rate (see Figure 1). The model inputs are the steering angle (δ_i) and torque (T_i) of the wheel $i \in \mathcal{I} = \{fl, fr, rl, rr\}$. These inputs are applied to the tires, which then develop longitudinal ($F_{x,i}$) and lateral ($F_{y,i}$) forces and dictate the vehicle motion. Semi-empirical models, such as magic tire formula [24], are often employed to represent the tire forces. The vehicle model also depends on inertial parameters, such as mass (m) and yaw inertia (I_z), and other parameters ($b_{x,i}, b_{y,i}$) related to the location of the center of gravity

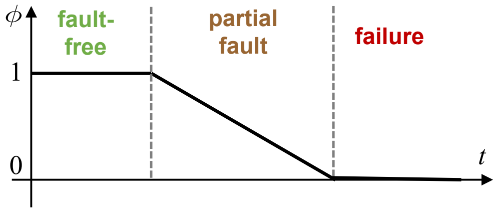


Figure 2. Effect of fault-free, partial fault and failure in the control effectiveness (ϕ).

(l_f, l_r) and trackwidth (c) – see Table 1.

Similarly to [15,25], we consider that the wheel actuators can be subject to faults that reduce their effectiveness. This is modeled as:

$$T_i = \phi_{T,i} T_i^*, \quad \delta_i = \phi_{\delta,i} \delta_i^* \tag{2}$$

where (T_i^*, δ_i^*) are the torque and steering angle references and $\phi_{j,i}$ the effectiveness of the actuator (j, i), with $j \in \{T, \delta\}$ and $i \in \mathcal{I}$. For example, $\phi_{j,i} = 1$ means fault-free operation, $\phi_{j,i} \in (0, 1)$ implies a partial loss of effectiveness and $\phi_{j,i} = 0$ represents a complete actuation failure (see Figure 2). In practice, there are several potential sources of failure. For example, in-wheel electric motors might experience reduced torque effectiveness due to motor overheating or open-phase faults; the steering actuator can be also affected by a broken actuator shaft or an electronics failure [2,26].

The focus of this work lies in the control of the vehicle's lateral dynamics. To derive a simplified control-oriented model for the lateral dynamics, we assume [23],[27]:

- the steering angles and body side slip are small, such as $\cos(\beta) \approx 1, \sin(\beta) \approx 0$
- the lateral force $F_{y,i}$ is proportional to the tire side-slip angle α_i, i.e., $F_{y,i} \approx C_i \alpha_i$, where C_i is the cornering stiffness of the tire and $\alpha_i = \delta_i - \beta - b_{y,i}\dot{\psi}/v$
- longitudinal forces are proportional to wheel torque (T_i) and the wheel inertial effects can be neglected [28], i.e., $F_{x,i} \approx T_i \frac{1}{r}$, where r is the wheel radius
- vehicle speed v can be treated as a known exogenous input

Inserting these assumptions into (1) allow us to represent the side-slip and yaw-rate dynamics as:

$$\dot{\beta} \approx -\dot{\psi} + \frac{1}{mv} \sum_{i \in \mathcal{I}} C_i \left(\delta_i - \beta - b_{y,i} \frac{\dot{\psi}}{v} \right) + d_\beta \tag{3a}$$

$$\ddot{\psi} \approx \frac{1}{I_z} \sum_{i \in \mathcal{I}} b_{x,i} \frac{T_i}{r} + b_{y,i} C_i \left(\delta_i - \beta - b_{y,i} \frac{\dot{\psi}}{v} \right) + d_\psi \tag{3b}$$

where d_x, d_y captures modeling approximation errors and external disturbances, such as yaw-moments or side-wind forces. The above model can compactly encapsulated

Table 1. Auxiliary parameters employed in the yaw-rate dynamics

$b_{x,fl}$	$b_{x,fr}$	$b_{x,rl}$	$b_{x,rr}$	$b_{y,fl}$	$b_{y,fr}$	$b_{y,rl}$	$b_{y,rr}$
$-c/2$	$c/2$	$-c/2$	$c/2$	l_f	l_f	$-l_r$	$-l_r$

into the following state-space representation:

$$\dot{x} = A(v)x + B_0(v)\Phi u + d \tag{4}$$

where $x = \begin{bmatrix}\beta & \dot{\psi}\end{bmatrix}^T \in R^n$ contains the lateral states and $d = \begin{bmatrix}d_x & d_\psi\end{bmatrix}^T$. The effectiveness matrix Φ and velocity-dependent matrices system matrices $A(v)$ and $B_0(v)$ are given by:

$$\begin{aligned}
A(v) &= \sum_{i\in\mathcal{I}} \begin{bmatrix} -\frac{1}{mv}C_i & -1-\frac{1}{mv^2}C_i b_{y,i} \\ -\frac{1}{I_z}C_i b_{y,i} & -\frac{1}{I_z v}C_i b_{y,i}^2 \end{bmatrix}, && (5)\\
B_0(v) &= \begin{bmatrix} 0 & 0 & 0 & 0 & \frac{C_{fl}}{mv} & \frac{C_{fr}}{mv} & \frac{C_{rl}}{mv} & \frac{C_{rr}}{mv} \\ -\frac{c}{2rI_z} & \frac{c}{2rI_z} & -\frac{c}{2rI_z} & \frac{c}{2rI_z} & \frac{l_f C_{fl}}{I_z} & \frac{l_f C_{fr}}{I_z} & -\frac{l_r C_{rl}}{I_z} & -\frac{l_r C_{rr}}{I_z} \end{bmatrix} && (6)\\
\Phi &= \mathrm{diag}(\phi_{T,fl},\ldots,\phi_{T,rr},\phi_{\delta,fl},\ldots,\phi_{\delta,rr}) && (7)
\end{aligned}$$

The control input $u \in R^m$ is composed of the reference wheel torques (T_i^*) and steering angles (δ_i^*);

$$u = \begin{bmatrix} T_{fl}^* & \ldots & T_{rr}^* & \delta_{fl}^* & \ldots & \delta_{rr}^* \end{bmatrix}^T \tag{8}$$

These references are physically constrained by the set $U = \{u \in R^m : \underline{u} \leq u \leq \overline{u}\}$ where $\underline{u}$, $\overline{u}$ characterize the minimum and maximum torques and steering angles.

One challenge in the control design is the vehicle over-actuation. Since the number of control inputs ($m = 8$) is higher than controllable degrees of freedom ($n = 2$), the matrix $B_0(v) \in R^{n\times m}$ does not possess full column rank. This means that there are an infinite number of control actions u that yield the same impact in the vector field of (4). To simplify the design task, a control allocation (CA) approach is adopted. The CA exploits the fact that B_0 is rank deficient to factorize this matrix as $B_0(v) = B(v)B_u(v)$, where $B \in R^{n\times k}$ and $B_u \in R^{k\times m}$ are matrices with rank k [29]. This factorization allows us to introduce an auxiliary variable, denoted virtual control input $\tau \in R^k$ and defined as

$$\tau = B_u(v)\Phi u \tag{9}$$

In this work, the factorization of $B_0(v)$ was performed such that: *i)* B_u is independent of the vehicle velocity, and *ii)* B is diagonal with $k = 2$. The first property is useful to make the CA less sensitive to velocity variations, while the second property paves the way for a decoupled action of the virtual input τ, where each channel in τ only directly affects one state (see (6)).

Based on these considerations, the vehicle model can be compactly represented through the following differential-algebraic model

$$\begin{aligned}
\dot{x} &= A(v)x + B(v)\tau + d, \qquad \tau = B_u\Phi u && (10a)\\
B(v) &= \begin{bmatrix} 1/v & 0 \\ 0 & 1 \end{bmatrix}, \quad B_u = \begin{bmatrix} 0 & 0 & 0 & 0 & \frac{C_{fl}}{m} & \frac{C_{fr}}{m} & \frac{C_{rl}}{m} & \frac{C_{rr}}{m} \\ -\frac{c}{2rI_z} & \frac{c}{2rI_z} & -\frac{c}{2rI_z} & \frac{c}{2rI_z} & \frac{l_f C_{fl}}{I_z} & \frac{l_f C_{fr}}{I_z} & -\frac{l_r C_{rl}}{I_z} & -\frac{l_r C_{rr}}{I_z} \end{bmatrix}
\end{aligned}$$

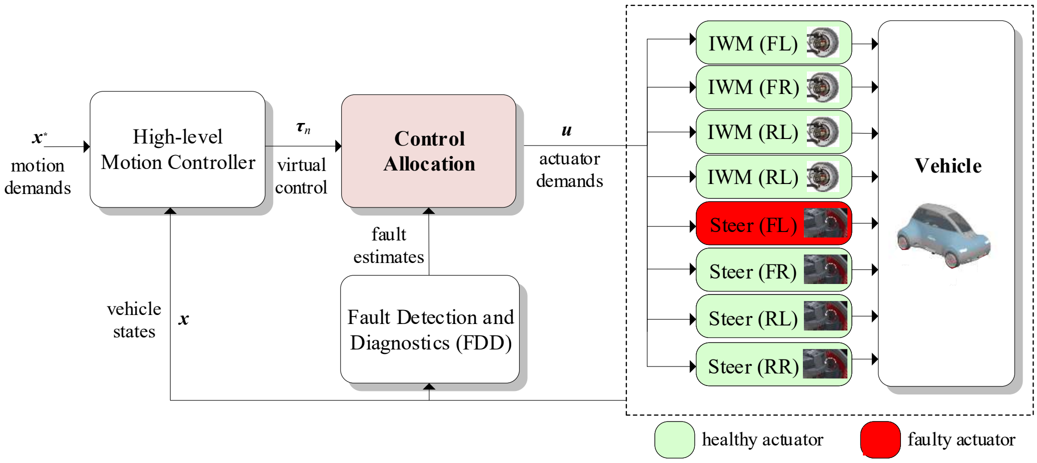


Figure 3. Block diagram of the FTC architecture.

3. Control Allocation

Figure 3 depicts the block diagram of the fault-tolerant controller adopted in this work, which is composed of three main components. The first, the high-level motion controller, computes a nominal value for the virtual input (τ_n), such that the state x follows the reference x^* as accurately as a possible, despite the disturbance d. To fulfill this goal, a disturbance-observer framework [30] is employed. The second, the fault detection and diagnosis (FDD), estimates actuation faults and the fault effectiveness matrix Φ. These faults can be self-diagnosed locally by the actuators or centrally by a high-level supervisor [26]. The third, the control allocation (CA), maps the virtual input into the actuator demands (u), while considering actuator constraints (U) and faults (Φ). The remainder of this section discusses the technical details of the control allocation block (the main contribution of this work) and presents a brief summary of the high-level controller. The FDD algorithm is beyond the scope of this work; the interested reader is referred to [26] for details.

3.1. *High-level Controller*

To facilitate the design of the controller, it is useful to consider the tracking error $e = x - x^*$ and its dynamics:

$$\dot{e} = A(v)e + B(v)\tau + \gamma(x^*, \dot{x}^*) + d \tag{11a}$$

$$\tau = B_u \Phi u \tag{11b}$$

where $\gamma(x^*, \dot{x}^*) = A(v)x^* - \dot{x}^*$. We assume that the reference x^* and its derivative $\dot{x}^*$ are sufficiently smooth and available. This information can be provided, e.g., by a trajectory planning or a trajectory controller [31].

3.1.1. Design of the Control Law

The virtual input generated by the high-level controller is obtained as:

$$\tau_n = B^{-1}(v)\left(-\gamma(x^*, \dot{x}^*) - \hat{d} - K(v)e\right) \tag{12}$$

and designed to

- eliminate the effects of input nonlinearities through the inversion of $B(v)$, which can be easily performed due to the diagonal structure of $B(v)$ (see (10))
- cancel the effect of the state reference x^* in the error dynamics through the feedforward term $-\gamma(x^*, \dot{x}^*)$
- annul the effect of additive disturbance $(-\hat{d})$ and model mismatches through disturbance observer (DoB) techniques [30]
- modify the closed-loop error dynamics through state-feedback control action $(-K(v)e)$

The DoB and state-feedback terms are the main design elements within the high-level control law. The DoB estimates the disturbance ($\hat{d} \in R^n$) using a nominal model of the vehicle and then generates a counter control action $(-B^{-1}(v)\hat{d})$ that attempts to cancel the effect of the real disturbance. The state-space representation of the DoB can be described as [30]

$$\hat{d} = z - Le \tag{13a}$$

$$\dot{z} = LB^{-1}(v)(-\gamma(x^*, \dot{x}^*) - \hat{d} - K(v)e) \tag{13b}$$

where L is a matrix selected by the designer and z and auxiliary dynamic variable. The state-feedback matrix is parameterized as $K(v) = A(v) + A_e$, where A_e is a Hurwitz matrix defined by the designer that is employed to place the poles of the error dynamics [32].

To gain further insight into the operation of the high-level controller, it is useful to introduce the disturbance estimation error $e_d = \hat{d} - d$, and its dynamics:

$$\dot{e}_d = \dot{z} - L\dot{e} = Le_d - \dot{d} \tag{14}$$

which allow us to represent the tracking error dynamic as

$$\dot{e} = (A(v) - K(v))e - e_d = A_e e - e_d \tag{15}$$

The joint tracking and disturbance estimation error can be described as:

$$\begin{bmatrix} \dot{e} \\ \dot{e}_d \end{bmatrix} = \begin{bmatrix} A_e & -I \\ 0 & L \end{bmatrix} \begin{bmatrix} e \\ e_d \end{bmatrix} + \begin{bmatrix} 0 \\ -I \end{bmatrix} \dot{d} \tag{16}$$

where I is the identity matrix. Simple tuning is one of the main advantages of this DoB-based design. Note that, due to the upper triangular structure of (16), the controller's parameters (L, A_e) can easily be selected to place the desired eigenvalues of the joint error dynamics. Robustness is another advantage of this DoB design framework. For example, if the disturbance varies slowly over time ($\dot{d} \approx 0$) and L and A_e are Hurwitz matrices, then the tracking and estimation errors converge asymptotically to the origin. More advanced tuning techniques for DoB-based controllers, which take into account time-varying disturbances and unstructured uncertainties, can also be used (see [33] for details).

3.1.2. Ultimate Boundedness

Since the disturbance d might be time varying non-vanishing, the high-level motion controller should be able to bound the tracking error e. In this work, we are particularly interested in fulfilling ultimate boundedness of the tracking error, where e converges to a (small) neighborhood around the origin after a transient period has elapsed. This property can be enforced if the high-level controller (τ_n) is accompanied by a Lyapunov function $V(e)$ that satisfies the following conditions [34, Th. 4.18]

$$i)\ c_1 \|e\|^2 \leq V(e) \leq c_2 \|e\|^2 \tag{17a}$$

$$ii)\ \dot{V} = \frac{\partial V(e)}{\partial e}\left(A(v)e + B(v)\tau_n + \gamma(x^*, \dot{x}^*) + d\right) \leq -c_3 V(e), \quad \forall \|e\| \geq \mu_e \tag{17b}$$

where c_1, c_2 and c_3 are positive constants, μ_e a small positive scalar and $\|\cdot\|$ the Euclidean norm. If these conditions are fulfilled, then the tracking error is ultimately bounded by [34, Th. 4.18]

$$\|e(t)\| \leq \mu_e \sqrt{c_2/c_1} = \bar{e}, \quad t \geq t_{min} \tag{18}$$

after some settling time period t_{min}. It can be shown that the high-level controller (12) admits a quadratic Lyapunov function $V(e) = e^T P e$, where P is a positive definite matrix, which fulfills the above conditions (see Appendix 7.1 for details).

Remark 1. Other types of control techniques could be employed to design the high-level controller. The only requirement is related to the knowledge of a Lyapunov function $V(e)$, which can be obtained in different ways. For example, some control methods provide constructive processes to build V, such as input-output linearization, linear state feedback or backstepping [34]. The region of attraction analysis, e.g., via polynomial optimization, is another common approach to build V [35].

3.2. Control Allocation

3.2.1. Classical Control Allocation

The CA goal is to compute the control actions $u \in U$ that produce the virtual input τ_n, requested by the high-level controller. Mathematically, this means solving a set of equality constraints $\tau_n = B_u \Phi u$ for the unknown $u \in U$. Since the set of equality constraints is usually underdetermined (yielding more unknowns than equations), multiple solutions exist. This multitude of solutions can be used by the allocator to pursue secondary goals, such as the minimization of the energy consumption, actuator wear or other criteria [10]. Such goals can be encoded into a cost function $J(u)$ and coupled with the equality and actuation constraints to formulate an optimization-based CA problem. Additionally, due to actuation saturations and faults, it might be impossible to generate the virtual input requested by the high-level controller without violating the constraint set U. In this case, the virtual input is said to be unfeasible, i.e., $\{u \in R^m : \tau_n = B_u \Phi u, u \in U\} = \emptyset$. To recover feasibility, a corrective term, $\Delta\tau$, is usually added to the requested virtual input. Based on these considerations, the classical CA (CCA) is formulated as:

$$
\begin{aligned}
(u^*, \Delta\tau^*) \quad &= \quad \underset{u,\Delta\tau}{\operatorname{argmin}}\, u^T W_u u + \Delta\tau^T W_\tau \Delta\tau && \text{(19a)}\\
& \quad\; s.t. \quad \tau_n + \Delta\tau = B_u \Phi u && \text{(19b)}\\
& \qquad\qquad \underline{u} \leq u \leq \overline{u} && \text{(19c)}
\end{aligned}
$$

where $J(u) = u^T W_u u$ is a quadratic penalization of the actuation effort and $\Delta\tau^T W_\tau \Delta$ penalizes allocation errors, which are weighted by the matrices W_u and W_τ, respectively. The CCA's optimal solution is denoted as $(u^*, \Delta\tau^*)$.

3.2.2. Lyapunov Control Allocation

The above CCA is commonly used to handle over-actuation in road vehicles, as previously reported in [10,11,14]. It provides a systematic framework to cope with actuation constraints, to minimize secondary costs and to reconfigure the controller in case of actuator faults. However, one issue that has received less attention so far is related with the effects of the corrective virtual input $\Delta\tau$ in the overall performance and stability of the control system. When unfeasible virtual inputs are requested, the CA needs to generate a non-zero correction term $\Delta\tau \neq 0$, deviating the virtual input from the nominal value τ_n. This introduces a perturbation in the control system, which might lead to a violation of the condition (17b), causing performance degradation or instability. Although the CCA attempts to minimize the magnitude of this correction term, there is no guarantee that the computed $\Delta\tau$ will preserve stability or honor the ultimate bound (18). Motivated by this observation, we augment the CCA with additional constraints that seek to reduce the risk of control instability and performance degradation due to $\Delta\tau$. To better understand this point, let us revisit the Lyapunov stability condition (17b) under the presence of the correction term $\Delta\tau$. By inserting $\Delta\tau$ in this condition, we obtain:

$$
\dot{V} = \frac{\partial V(e)}{\partial e}\left(A(v)e + B(v)\tau_n + \gamma(x^*, \dot{x}^*) + d\right) + \underbrace{\frac{\partial V(e)}{\partial e} B(v)\Delta\tau}_{\gamma(e,v,\Delta\tau)} \tag{20}
$$

where $\gamma(e, v, \Delta\tau)$ is an extra term due to $\Delta\tau$. Recall that our goal is to ensure that $\dot{V}$ is negative when the tracking error e is outside a small neighborhood around the origin, i.e., $\dot{V} \leq -c_3 V(e) < 0$ for $\|e\| \geq \mu_e$. Consequently, if the corrective virtual input $\Delta\tau$ is computed in such a way that the extra term $\gamma(e, v, \Delta\tau)$ is non-positive, i.e.,

$$
\gamma(e, v, \Delta\tau) = \frac{\partial V(e)}{\partial e} B(v)\Delta\tau \leq 0 \tag{21}
$$

then the Lyapunov condition (17b) can be preserved, and the nominal performance of the high-level controller recovered. The Lyapunov CA (LCA) builds on this insight and incorporates (21) as an additional constraint of the allocation problem. Since it might be impossible to fulfill (21) without violating the physical actuation constraints (U), we relax this inequality with a non-negative slack variable (s). The resulting LCA formulation is described as:

$$
\begin{aligned}
(u^*, \Delta\tau^*, s^*) \quad &= \quad \underset{u,\Delta\tau,s}{\operatorname{argmin}}\, u^T W_u u + \Delta\tau^T W_\tau \Delta\tau + W_s s^2 && \text{(22a)}\\
&\quad\ s.t. \quad \tau_n + \Delta\tau = B_u \Phi u, && \text{(22b)}\\
&\qquad\qquad \underline{u} \leq u \leq \overline{u} && \text{(22c)}\\
&\qquad\qquad \frac{\partial V(e)}{\partial e} B(v) \Delta\tau \leq s, \quad s \geq 0 && \text{(22d)}
\end{aligned}
$$

where W_s is a weight that penalizes high values of the slack variable s and violations of the constraint (21). The term $\frac{\partial V(e)}{\partial e}$ represents the gradient of the Lyapunov function. Since a quadratic Lyapunov function is employed here[1], the gradient is linearly dependent on the tracking error, i.e., $\frac{\partial V(e)}{\partial e} = 2e^T P$. The optimal values are denoted as $(u^*, \Delta\tau^*, s^*)$.

From a computational perspective, the LCA contains two additional inequalities (22d) and one additional decision variable (s) when compared to the CCA formulation. Thus, the LCA requires a modest increase in numerical complexity. Furthermore, the decision variables $(u, \Delta\tau, s)$ appear linearly in the constraints and quadratically in the cost function; as a result, LCA belongs to a class of quadratic-programming (QP) problems, which can be efficiently solved. We employed Hildredth's algorithm [36] to solve the QP problem (22). This algorithm relies on an active-set method that iteratively solves the dual optimization problem without requiring matrix inversions. Hildredth's algorithm has also shown good convergence properties when tackling small- to medium-dimension optimization problems [37,38]. Consequently, this algorithm is well suited for tackling automotive CA problems, where the number of decision variables is usually small.

Remark 2. Enforcement of state constraints is not explicitly considered in the LCA design. To address this issue, the LCA can be augmented with control barrier functions, which exploit the null-space of the control input domain to enforce safety constraints (see [11] for details).

3.3. Analysis

Let us now analyze the ultimate boundedness properties that are obtained when the high-level controller and the LCA are combined. This analysis is divided in two parts. The first part establishes conditions under which the nominal ultimate bound ($\bar{e}$) expected by the high-level controller can be honored by the LCA. The second defines the worst-case ultimate bound. Recall that, if (17) is fulfilled, then tracking error e is ultimately bounded by $\bar{e} = \mu_e \sqrt{c_2/c_1}$, i.e., e is attracted in finite time to a ball with radius $\bar{e}$ (see Figure 4a).

Proposition 3.1. *The LCA recovers the nominal ultimate bound ($\bar{e}_{LCA} = \bar{e}$) if the slack variable s^* satisfies*

$$
s^* \leq \frac{\mu_e^2 c_1 c_3}{2} \tag{23}
$$

[1] recall: $V(e) = e^T P e$

where c_1, c_3, μ_e are positive constants introduced in Section 3.1.2.

Proof. To demonstrate this result, let us revisit the Lyapunov condition (17b) in the presence of the corrective virtual input $\Delta\tau$,

$$\dot{V} \leq -c_3 V(e) + \gamma(e, v, \Delta\tau), \quad \forall \|e\| \geq \mu_e \tag{24}$$

The LCA formulation (22) guarantees that $\gamma(e, v, \Delta\tau)$ is upper-bounded by the slack variable $s^*(e, v)$, which depends on the tracking error (e) and vehicle velocity (v). To simplify the proof, we perform a local Lyapunov analysis in the domain $e \in \mathcal{E}$ (a compact neighborhood around the origin) and $v \in \mathcal{V}$ (velocity range of interest for the vehicle). We assume that $s^*(e, v)$ can be upper-bounded in this domain by:

$$\overline{s} = \max_{e \in \mathcal{E}, v \in \mathcal{V}} s^*(e, v) \tag{25}$$

Inserting this upper bound into (24) yields

$$\begin{aligned}
\dot{V} &\leq -c_3 V(e) + s^*(e, v) \leq -c_3 V(e) + \overline{s}, \quad \forall \|e\| \geq \mu_e && (26)\\
&= -\frac{c_3}{2} V(e) - \frac{c_3}{2} V(e) + \overline{s}, \quad \forall \|e\| \geq \mu_e && (27)\\
&\leq -\frac{c_3}{2} V(e) - \frac{c_3}{2} c_1 \|e\|^2 + \overline{s}, \quad \forall \|e\| \geq \mu_e && (28)
\end{aligned}$$

where the last inequality results from the lower bound (17a). If the tracking error e is sufficiently large, then the quadratic error term $(\|e\|^2)$ dominates the bound $\overline{s}$, leading to:

$$\dot{V} \leq -\tilde{c}_3 V(e), \quad \forall \|e\| \geq \tilde{\mu}_e \tag{29}$$

with $\tilde{c}_3 = \frac{c_3}{2}$, $\tilde{\mu}_e = \max\{\mu_e, \sqrt{2\overline{s}/(c_1 c_3)}\}$. Using this inequality together with [34, Th. 4.18] allows us to compute LCA's ultimate bound $\overline{e}_{LCA} = \tilde{\mu}_e \sqrt{c_2/c_1}$. Consequently, if

$$\mu_e > \sqrt{2\overline{s}/(c_1 c_3)} \tag{30}$$

then $\tilde{\mu}_e = \mu_e$ and the nominal ultimate bound can be recovered, $\overline{e}_{LCA} = \overline{e}$. Moving $\overline{s}$ to the left side of the (30) yields the inequality $\overline{s} \leq \mu_e^2 c_1 c_3/2$. Condition (23) is finally obtained by combining this inequality with the bound $s^* \leq \overline{s}$ induced from (25). □

Proposition 3.2. *If condition* (23) *cannot be honored, then the LCA will enlarge the ultimate bound according to*

$$\overline{e}_{LCA}(\overline{s}) = \sqrt{2\overline{s}\frac{c_2}{c_1^2 c_3}} \geq \overline{e} \tag{31}$$

where $\overline{s}$ is an upper-bound for s^*, computed over a region of interest (see (25)).

Proof. This result is a direct consequence of the LCA's ultimate bound $\overline{e}_{LCA} = \tilde{\mu}_e \sqrt{c_2/c_1}$, which was determined in proof of Proposition 3.1. □

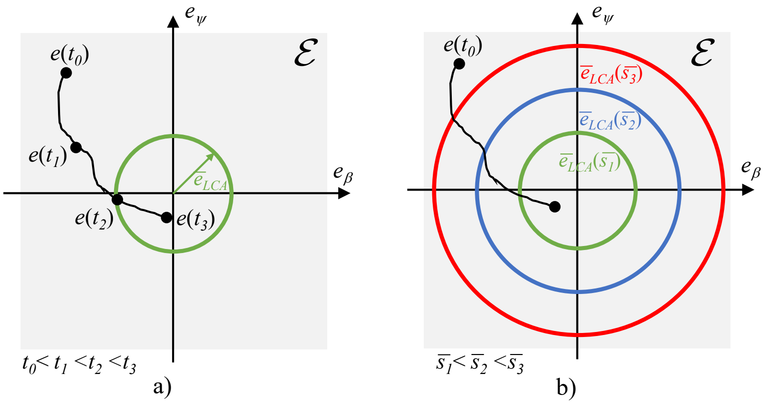

e_ψ
$e(t_0)$
$\mathcal{E}$
$e(t_1)$
e_β
$\overline{e}_{LCA}$
$e(t_2)$
$e(t_3)$
$t_0< t_1 <t_2 <t_3$
a)
e_ψ
$e(t_0)$
$\mathcal{E}$
$\overline{e}_{LCA}(\overline{s}_3)$
$\overline{e}_{LCA}(\overline{s}_2)$
e_β
$\overline{e}_{LCA}(\overline{s}_1)$
$\overline{s}_1< \overline{s}_2 <\overline{s}_3$
b)

Figure 4. a) Graphical representation of the ultimate bound of the tracking error ($\overline{e}_{LCA}$); b) effect of the slack variable's upper bound ($\overline{s}$) in the tracking error's ultimate error ($\overline{e}_{LCA}$). The set $\mathcal{E}$ represents a neighborhood around the origin and is highlighted in gray.

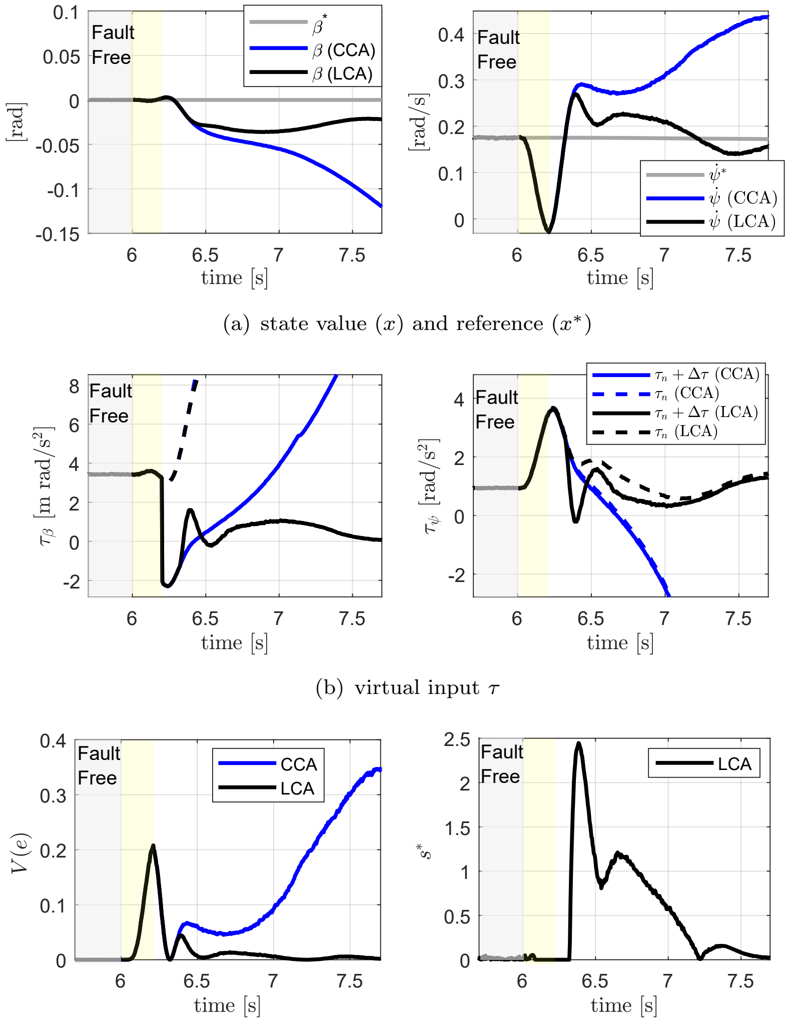
Table 2. Controller Parameters

Variable	Symbol	Value
High-level Controller		
Desired error dynamics	A_e	$-\operatorname{diag}(1,2)$
Disturbance-observer dynamics	L	$-\operatorname{diag}(5,8)$
Lyapunov function	P	$\operatorname{diag}(0.05, 0.1)$
Gradient of Lyapunov function	$\frac{\partial V}{\partial e}$	$2e^T P$
Control Allocation		
Weight matrix -actuator use	W_u	$\operatorname{diag}(w_T, w_T, w_T, w_T, w_\delta, w_\delta, w_\delta, w_\delta)$ $w_T = 5 \times 10^{-6}, w_\delta = 100$
Weight matrix - virtual input error	W_τ	$\operatorname{diag}(10, 100)$
Weight - slack variable	W_s	10^6
Maximum actuation limit	$\overline{u}$	$\begin{bmatrix}\overline{T} & \overline{T} & \overline{T} & \overline{T} & \overline{\delta} & \overline{T} & \overline{\delta} & \overline{\delta}\end{bmatrix}^T$ $\overline{T} = 160$ Nm, $\overline{\delta} = 0.3489$ rad
Minimum actuation limit	$\underline{u}$	$-\overline{u}$
Torque-to-acceleration gain	$B_{u,a}$	$\begin{bmatrix}k_e & k_e & k_e & k_e & 0 & 0 & 0 & 0\end{bmatrix}$ $k_e = 0.0036$ (m/s^2)/Nm
FDD		
nominal delay	t_d	0.2s
nominal uncertainty	η	0

Figure 4b provides a graphical interpretation of Proposition 3.2. This graph illustrates that higher values for the slack variable s (or its upper bound $\overline{s}$ to be technically correct) increase the ultimate error bound $\overline{e}_{LCA}$. Since the LCA seeks to minimize s, this will also decrease $\overline{e}_{LCA}$. Consequently, the s-cost included in the LCA can be seen as a surrogate function for minimizing the ultimate error bound ($\overline{e}_{LCA}$). In contrast, the CCA formulation (19) lacks a systematic mechanism to reduce the tracking error; it favors the minimization of the magnitude of $\Delta\tau$, which might not necessarily lead to lower ultimate bounds in the tracking error.

4. Numerical Simulation

This section evaluates the performance of the CCA and LCA through numerical simulation. The evaluation is carried out in a co-simulation environment between Matlab-Simulink and Dymola. The former tool is used for implementation of the control functions, while the latter provides a Modelica-based vehicle model [39]. The vehicle model



(a) state value (x) and reference (x^*)

(b) virtual input τ

(c) Lyapunov function (V) and slack variable (s)

Figure 5. Numerical simulation of the LCA and CCA, when subject to front-steering fault. There are three operation modes: i) fault-free (gray background); ii) fault injection and controller operation without knowledge of the fault (yellow background); iii) fault injection and controller operation with knowledge of the fault (white background). Note: the FDD was parameterized with $t_d = 0.2$s and $\eta_{\delta,fl} = \eta_{\delta,fr} = 0$.

employs a double-track model to represent the chassis motion, as well as vertical dynamics, wheel rotational dynamics and nonlinear tire-road friction forces (see [26,40] for details). The CA problem was also augmented with an additional equality constraint for the allocation of the longitudinal acceleration. More specifically, we included $a_x^* = B_{u,a}\Phi u$ in (19) and (22), where a_x^* is the acceleration reference requested by the longitudinal controller (fixed to $a_x^* = 0$ during the maneuvers), while $B_{u,a}$ is the static gain between actuators and longitudinal acceleration (see Table 2).

As a reference maneuver for the tests, we considered a steady-state cornering maneuver, with a radius 140m and a constant velocity 25 m/s. During this maneuver, a steering failure is injected, leading to a complete loss of front steering actuation. Such a fault can occur, for example, because of communication failures between the motion controller and the vehicle's front axle [26]. Additional simulation tests obtained with dynamic double lane change maneuvers are included in Appendix 7.2.

Given that fault diagnosis is not the focus of this work, a pragmatic model is employed to approximate the FDD behavior. The FDD estimation is assumed to be related with true actuation effectiveness signal ($\phi_{j,i}$), delayed by t_d seconds and subject to multiplicative uncertainty ($\eta_{j,i}$):

$$\hat{\phi}_{j,i} = (1 + \eta_{j,i})\phi_{j,i}(t - t_d) \tag{32}$$

where $\hat{\phi}_{j,i}$ is the fault estimation of actuator (j, i). As discussed in [41], the model (32) allows for a precise control over the FDD's fault detection time and estimation uncertainty. This feature is particularly useful to evaluate the effect of the FDD uncertainties in the CA performance, which is performed in Section 4.2. The control and FDD parameters are shown in Table 2.

4.1. Time-domain Results

Figure 5a shows the reference signals and the vehicle states obtained during the steady-state cornering maneuver. These results can be divided into three phases. The first phase, highlighted in gray, focuses on fault-free operation. During this phase, both CCA and LCA present identical control performance, with a negligible tracking error in the side-slip and yaw-rate channels. To produce the desired virtual inputs, the control allocators combine front and rear steering with a small amount of torque difference between the left and right wheels (Figure 6). In the second phase, which begins at $t_0 =$ 6.0s, a fault in the front steering actuators is triggered, reducing the lateral front force and the yaw rate. From t_0 to t_0+t_d (which corresponds to 6.0s to 6.2s in Figures 5 and 6 and is highlighted in yellow), the controller operates without knowledge of the fault; it relies mainly on the integral action of the high-level controller to compensate for the fault disturbance (Figure 5b). In the third phase, which corresponds to $t \geq t_0 + t_d =$ 6.2s, the CA algorithms receive the fault estimation from the FDD. This leads to the re-configuration of the allocation strategy. In particular, the torque difference between left and right motors is increased (and saturated) as a mean to quickly compensate for the loss of yaw-moment.

The fault and subsequent reconfiguration introduce a large disturbance in the operation of the high-level controller. The CCA-based controller struggles to mitigate this disturbance, leading to overshooting the yaw-rate and vehicle instability (Figure 5a). In contrast, the LCA-based controller can recover from the fault in less than 1.5s. The key reason for this recovery lies in the incorporation of the Lyapunov stability condition in the allocation problem. As shown in Figure 5c, after the reconfiguration, the LCA gracefully degrades the control performance by minimizing the value of the slack variable s. This degradation mechanism is crucial to correct the original virtual input generated by the high-level controller (see the significant difference between τ_n and $\tau_n + \Delta\tau$ illustrated in Figure 5b after 6.2s) and to decrease the tracking errors.

4.2. Sensitivity Analysis

This section analyses the impact of the fault estimation accuracy in the post-fault control performance. Of particular interest is the effect of the FDD delay (t_d), FDD uncertainty (η) and post-fault actuation effectiveness (ϕ) when tracking the side-slip angle and yaw-rate references. To assist in this analysis, we consider the average ab-

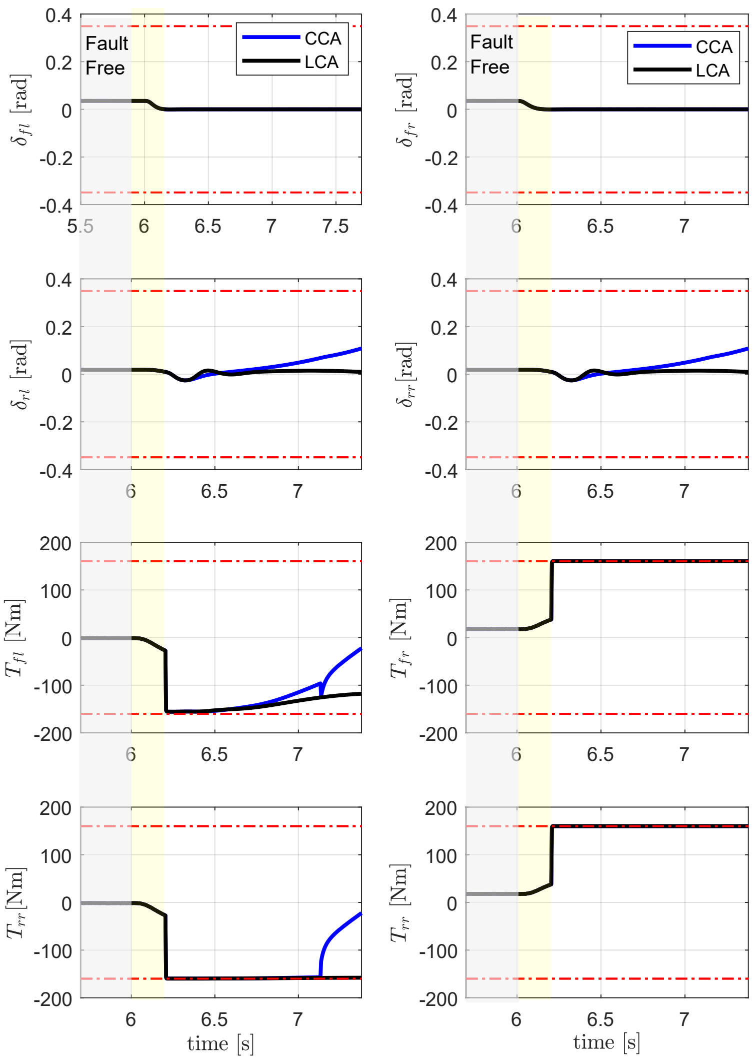


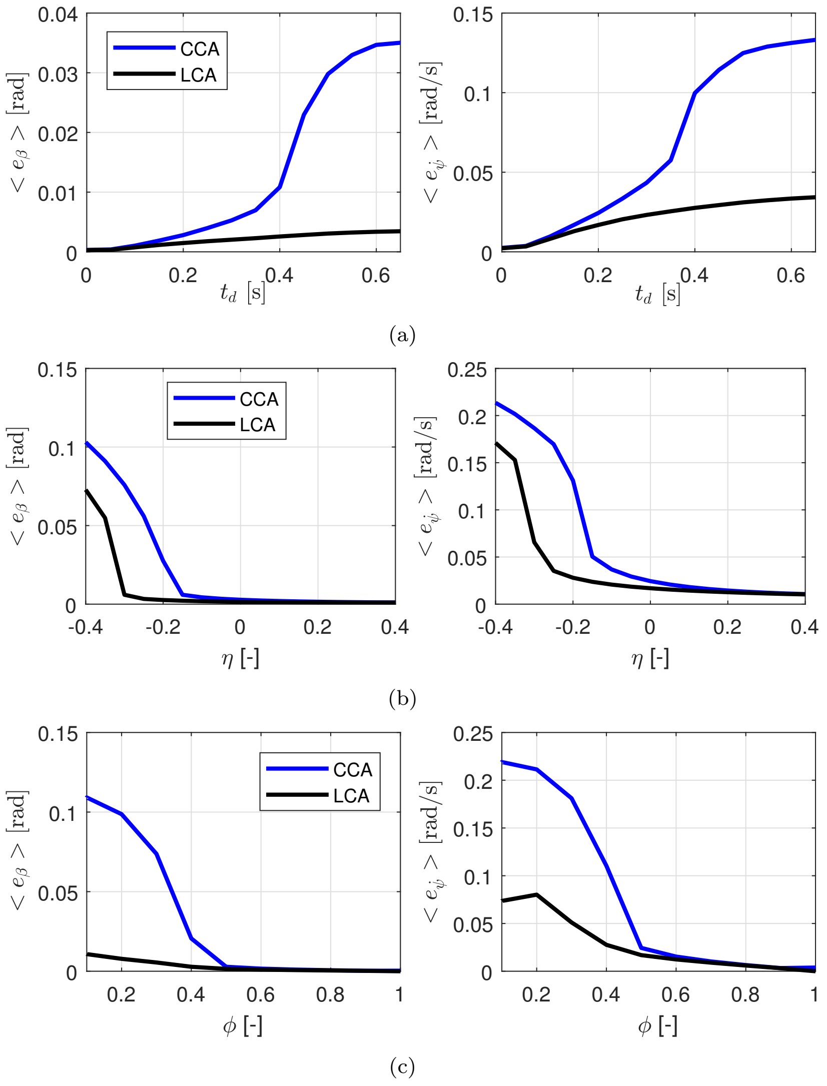
Figure 6. Control inputs (u^*) generated by the LCA and CCA during the numerical simulation. Actuation limits are represented in dashed red lines. There are three operation modes: i) fault-free (gray background); ii) fault injection and controller operation without knowledge of the fault (yellow background); iii) fault injection and controller operation with knowledge of the fault (white background).

solute tracking error as the main performance metric:

$$< e_j >= \frac{1}{t_1 - t_0} \int_{t_0}^{t_1} |x_j(t) - x_j^*(t)| dt \tag{33}$$

where $j \in \{\beta, \dot{\psi}\}$ is the control channel (side-slip or yaw-rate), t_0 the time instant when the fault is injected and t_1 the duration of the maneuver.

Figure 7a shows the effect of the FDD delay (t_d) when injecting a front steering fault with actuation effectiveness of 0.5 and zero estimation error (i.e., $\phi_{\delta,i}(t) = 0.5$, for



(a)

(b)

(c)

Figure 7. Average side-slip and yaw-rate tracking errors ($< e_\beta >, < e_{\dot\psi} >$) and their sensitivities to FDD time delays (t_d), estimation errors (η) and fault magnitude (ϕ).

$t \geq t_0$, and $\eta_{\delta,i} = 0$, $i \in \{fl, fr\}$). When the FDD delay is inferior to $0.1s$, both CCA and LCA can quickly recover from the fault disturbance with a low tracking error. On the other hand, as the estimation delay t_d increases, the controller operates longer periods without fault estimation, leading to more severe disturbances. The results show that the LCA copes better with these longer delays than the CCA. For example, for $t_d = 0.4$s, the CCA's average yaw-rate error in post-fault operation is 0.1 rad/s, while the LCA presents an error of 0.025 rads/s (-76%). Similar results are observed in the side-slip channel.

Figure 7b shows the effect of the estimation error (η) when injecting a front steering fault with a post-fault actuation effectiveness of 0.5 and $t_d = 0.2$s. For $\eta > 0$, the FDD over-estimates the fault, i.e., the estimated effectiveness value $\hat{\phi}$ is higher than the real effectiveness ϕ. The results show that these types of estimation errors have a small impact in the control performance. On the other hand, when the FDD under-estimates the fault ($\eta < 0$), higher tracking errors emerge. Independently of the type

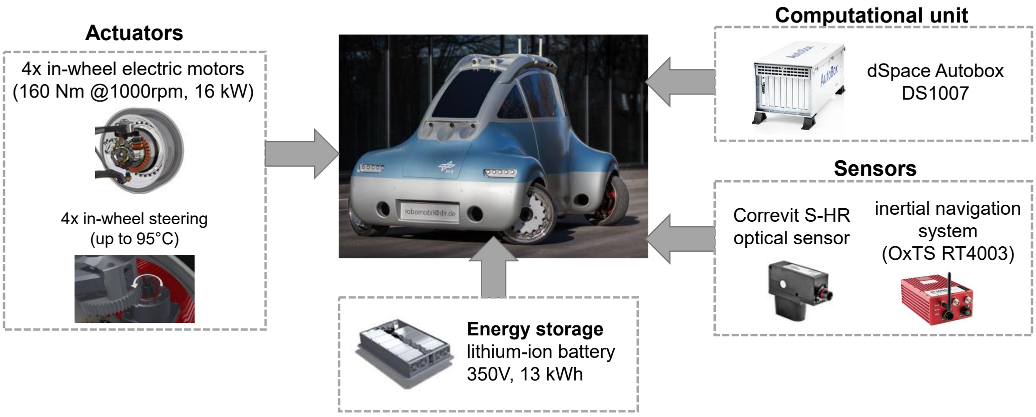


Figure 8. Overview of the prototype vehicle employed in the experimental validation of this work.

of estimation error, the LCA reduces the tracking error when compared to CCA.

Figure 7c depicts the impact of the post-fault actuator effectiveness in the tracking performance. As expected, a lower (post-fault) actuator effectiveness leads to more severe disturbances and higher tracking errors in both side-slip and yaw-rate control channels. When compared to CCA, the LCA significantly reduces these errors, especially for ϕ lower than 0.5. Overall, these results demonstrate the inherent superiority of the LCA in handling estimation uncertainties in the FDD stage and in coping with large fault magnitudes.

5. Experimental Validation

This section reports the experimental validation of the LCA. The experiments were performed with the ROboMObil prototype [9,21,26], an advanced electric vehicle with distributed wheel-based propulsion. As depicted in Figure 8, each wheel has a permanent magnet motor, embedded in the wheel hub, with a peak torque and power of 160Nm and 16kW, respectively. It also contains wheel-independent steer actuators, which allows us to command steering angles in the range $[-25°, 95°]$. The inertial and geometric parameters of the ROboMObil are summarized in Table 3. The yaw-rate and accelerations sensing information was generated by an inertial measurement unit (OxTS RT4003), while the vehicle velocity and side-slip angle was measured by an optic sensor (Correvit S-HR). The LCA was implemented in a dSPACE AutoBox DS1007–a rapid control prototyping system– with a sample time of 4ms. The previous

Table 3. Parameters of the ROboMObil prototype

Variable	Symbol	Value
vehicle mass	m	1000 kg
yaw inertia	I_z	1130 kg.m^2
cornering stiffness (fl)	C_{fl}	30 kN/rad
cornering stiffness (fr)	C_{fr}	30 kN/rad
cornering stiffness (rl)	C_{rl}	35 kN/rad
cornering stiffness (rr)	C_{rl}	35 kN/rad
center of gravity location	l_f	1.22 m
center of gravity location	l_r	1.18 m
trackwidth	c	1.45 m
wheel radius	r	0.274 m

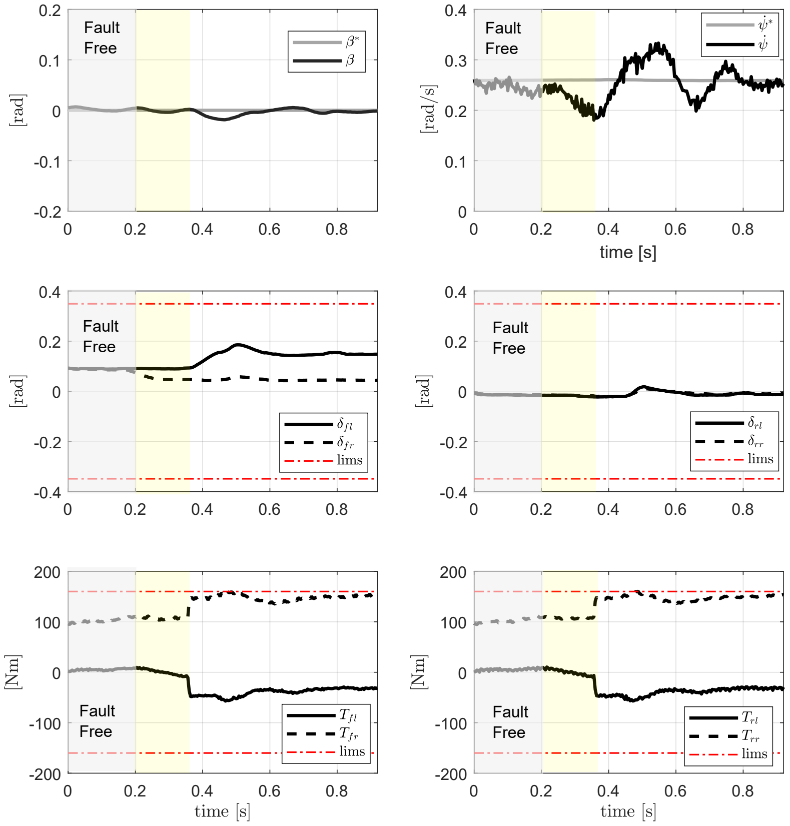
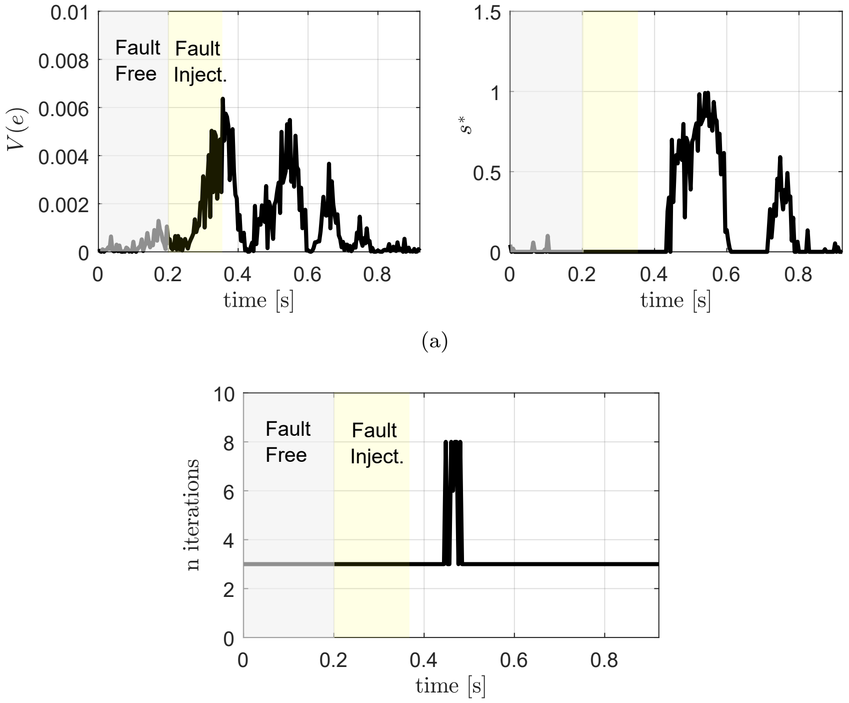


Figure 9. Experimental validation of the LCA: i) fault-free period highlighted in gray; ii) fault injection period highlighted in yellow (during this period, the controller operates without knowledge of the fault estimate); iii) controller reconfiguration period highlighted with white background. Test carried out with velocity = 6.5 m/s.

section showed that the LCA-based controller offers superior performance and safety than the CCA; because of this, only the LCA is evaluated experimentally.

As a test maneuver, we selected a quasi-steady-state cornering with a radius of 26 m and velocity of 6.4 m/s. A fault in the front right actuator (δ_{fr}) is artificially triggered at 0.2s, leading to a 50% loss of effectiveness. The results, depicted in Figures 9 and 10, can be divided into three phases: *i*) fault-free operation, *ii*) fault injection and *iii*) control reconfiguration.

- During the first phase, fault-free operation (0s to 0.2s), the side-slip and yaw-rate references are followed with reduced tracking errors. The LCA employs a combination of front steering and torque vectoring (in both front and rear axle) for yaw-moment generation.
- During the second phase (0.2s to 0.38s), the effectiveness of the front right (fr) steering drops 50%, reducing the lateral front force and the overall yaw-moment applied to the vehicle. This induces vehicle understeering, i.e., measured yaw-rate is decreased with respect to the reference, and higher tracking errors.
- During the third phase ($t \geq$ 0.38s), the LCA receives fault estimate of the steering fault. The LCA reconfigures the actuation effort to attenuate the fault's



(a)

(b)

Figure 10. a) Evolution of the Lyapunov function $V(e)$ and slack variable s^* during the experimental tests; b) number of iterations necessary for the LCA's numerical solver to converge.

effect and to recover tracking performance. This is achieved by increasing the steering angle of the front left (healthy) actuator and raising the torque difference between the right and left motors; the rear steering is also slightly increased at around 0.5s to attenuate the transient yaw-rate.

After 0.8s the controller recovers control performance, enabling us to continue vehicle operation despite the faulty actuator. It is interesting to note that, thanks to the quick control reconfiguration, the tracking performance of the side-slip angle is almost unaffected.

To gain further insights into the LCA operation, Figure 10a shows the evolution of the Lyapunov function (V) and slack variable (s^*). As expected, during the fault injection period ($0.2 - 0.38$s), the Lyapunov function V increases, indicating loss of control performance. After the control reconfiguration ($t \geq 0.38$s), the value of V is decreased. However, this decrease is noisier and more oscillatory than the response obtained during the simulation results, which can be explained by higher measurement noise and model mismatches present in the experiments. Additionally, the actuator fault forces the controller to relax the Lyapunov stability condition and to decrease performance; the LCA attempts to attenuate the loss of control performance by minimizing the value of the slack variable (s^*) during the post-fault transient, reaching a maximum peak value of 1 (Figure 10a). After 0.8s, the LCA completely eliminates the fault transient and recovers an almost zero tracking error (and zero V and s^*).

Figure 10b shows the number of iterations that the LCA's numerical solver required to find optimal solutions. The solver requires less than 8 iterations to converge, with a maximum computational time of 0.5 ms, which is significantly inferior to the controller's sample time (4ms). These results demonstrate the good numerical efficiency and real-time capability of the LCA.

6. Conclusion and Outlook

An active fault-tolerant controller for over-actuated electric vehicles was developed in this work. It relied on a control allocation framework to distribute the actuation effort and to re-configure the control policy in the aftermath of faults. To improve stability and tracking performance, additional constraints and costs, based on Lyapunov analysis, were incorporated into the control allocation framework. Numerical simulations and theoretical analysis showed that, in comparison to the classical allocation strategies, the proposed Lyapunov-based control allocation (LCA) reduces tracking error during post-fault operation, while requiring a modest increase in computational effort. Experimental tests carried out with an over-actuated electric vehicle prototype verified the effectiveness of the LCA.

Future work will integrate LCA in trajectory tracking controllers and perform comparisons against other model-based and learning-based controllers; we also plan to investigate in more detail the potential of this allocation concept to improve safety in vehicles with high levels of autonomy.

Acknowledgement

The authors would like to thank the German Aerospace Center (DLR) for funding this research as well as the ROboMObil team, in particular Clemens Satzger and Lok Man Ho, for the support in the experimental tests.

7. Appendix

7.1. High-Level Controller: Lyapunov Function and Ultimate Boundedness

This Appendix presents a Lyapunov function for the high-level controller (12) that fulfills assumptions (17). First, let us consider a Lyapunov function for the disturbance estimation error, $V_d = e_d^T P_d e_d$. The matrix P_d is positive definite and can be obtained by solving[2] $P_d L + L^T P_d = -I$, where I is the identity matrix. Assuming $\dot{d} \approx 0$, the estimation error is bounded by [34, Theorem 4.10]:

$$\|e_d(t)\| \leq k \|e_d(0)\| = \bar{e}_d, \quad \text{for all } t \geq 0, \tag{34}$$

where $k = \sqrt{\lambda_{max}(P_d)/\lambda_{min}(P_d)}$. The variables $\lambda_{max}(P_d)$, $\lambda_{min}(P_d)$ represent the maximum and minimum eigenvalue of P_d, respectively.

Second, let us consider the tracking error e, and consider the quadratic Lyapunov function $V(e) = e^T P e$, where P is a positive define matrix obtained by solving $PA_e + A_e^T P = -I$. Condition (17a) is satisfied with $c_1 = \lambda_{min}(P)$ and $c_2 = \lambda_{max}(P)$.

[2] The matrix equation can be easily obtained through standard numerical tools, such as *lyap*(.) function from Matlab's Control System Toolbox [42] or analytical algebraic techniques [43].

To verify condition (17b), we compute the following upper bound for $\dot{V}$:

$$\begin{aligned} \dot{V} &= -e^T e + 2e^T P e_d \leq -\|e\|^2 + 2|e^T P e_d| = -\|e\|^2 + 2\|e\|\|Pe_d\| \\ &\leq -\|e\|^2 + 2\|e\|\|P\|\|e_d\| \end{aligned} \tag{35}$$

where the last inequality makes use of the relation $\|Pe_d\| \leq \|P\|\|e_d\|$ with $\|P\| = \lambda_{max}(P^T P)^{1/2}$ [34]. Next, we use the upper bound ($\overline{e}_d$) from the disturbance estimation error (34) to obtain:

$$\dot{V} \leq -\|e\|^2 + 2\|e\|\|P\|\overline{e}_d = -(1-\theta)\|e\|^2 - \theta\|e\|^2 + 2\|P\|\overline{e}_d \tag{36}$$

$$\leq -(1-\theta)\|e\|^2, \quad \forall \|e\| \geq 2\|P\|\overline{e}_d/\theta \tag{37}$$

where θ is an arbitrary constant in the range $(0,1)$. This allows us to fulfill condition (17b) with $c_3 = (1-\theta)$ and $\mu_e = 2\|P\|\overline{e}_d/\theta$.

7.2. Additional Simulation Results: Double lane change

Figure 11 depicts the obtained simulation results when the vehicle is subject to faults during a double lane change maneuver, carried out with an initial speed of 25m/s. The maneuver can be divided in three phases. The first phase (highlight in gray) corresponds to the fault-free operation, where both LCA and CCA provide identical results. In the second phase (highlighted in yellow), a fault in the steering actuators is injected, leading to a sudden loss in the front steering capability (Figure 12). During this phase, both LCA and CCA operate without knowledge of the fault, degrading tracking performance. In the third phase ($t > 2.5$s), the fault is detected and both controllers perform aggressive corrective actions to eliminate the tracking errors. The LCA is more effective in the generation of these corrective actions, particularly in the yaw-rate channel. Figure 11a shows that, in comparison with the CCA, the LCA offers a less oscillatory yaw-rate recovery and requires less time to eliminate yaw-rate tracking error (3.3s with LCA vs 4s with CCA). Overall, the CCA's average yaw-rate error during post-fault conditions is 0.067 rad/s, while the LCA offers 0.033 rad/s (-50%). Regarding the side-slip tracking, the LCA offers slight reduction in the maximum tracking error: 0.041 rad with CCA *vs* 0.031 rad with LCA. The superior post-fault recovery offered by the LCA can be explained by the additional Lyapunov-based constraints and costs included in the optimization problem, which minimize violation of Lyapunov stability constraint (via s) and lead to lower values of $V(e)$ (Figure 12c).

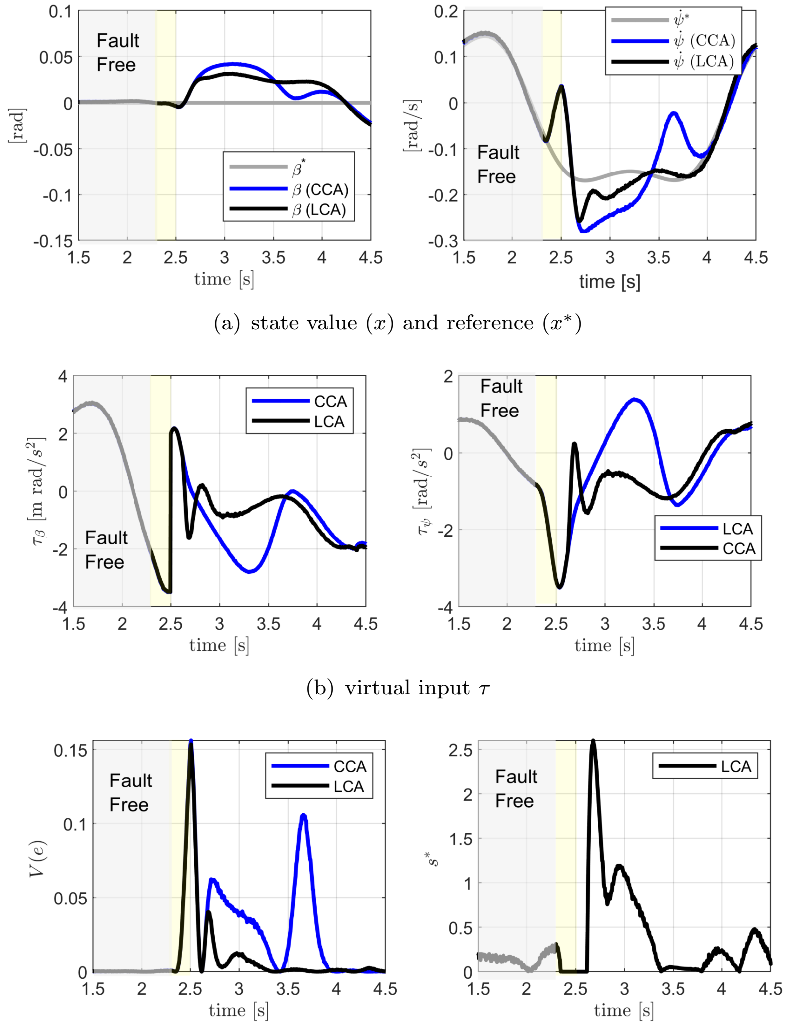
References

[1] Klomp M, Jonasson M, Laine L, et al. Trends in vehicle motion control for automated driving on public roads. Veh Syst Dyn. 2019 Jul;57(7):1028–1061.

[2] Wanner D, Drugge L, Trigell AS. Fault classification method for the driving safety of electrified vehicles. Veh Syst Dyn. 2014 May;52(5):704–732.

[3] SAE. Taxonomy and definitions for terms related to driving automation systems for On-Road motor vehicles. SAE International; 2018. J3016.

[4] Mutoh N, Nakano Y. Dynamics of Front-and-Rear-Wheel-Independent-Drive-Type electric vehicles at the time of failure. IEEE Trans Ind Electron. 2012 Mar;59(3):1488–1499.



(a) state value (x) and reference (x^*)

(b) virtual input τ

(c) Lyapunov function (V) and slack variable (s)

Figure 11. Numerical simulation of the LCA and CCA, when subject to front-steering fault during a double lane change (initial speed = 25m/s). There are three operation modes: i) fault-free (gray background); ii) fault injection and controller operation without knowledge of the fault (yellow background); iii) fault injection and controller operation with knowledge of the fault (white background)

[5] Savitski D, Ivanov V, Augsburg K, et al. Wheel slip control for the electric vehicle with In-Wheel motors: Variable structure and sliding mode methods. IEEE Trans Ind Electron. 2020 Oct;67(10):8535–8544.

[6] Wang W, Chen X, Wang J. Motor/Generator applications in electrified vehicle Chassis—A survey. IEEE Transactions on Transportation Electrification. 2019 Sep;5(3):584–601.

[7] Harrer M, Pfeffer P, editors. Steering handbook. Springer; 2017.

[8] de Castro R, Araújo RE, Tanelli M, et al. Torque blending and wheel slip control in EVs with in-wheel motors. Veh Syst Dyn. 2012 Jan;50(sup1):71–94.

[9] Bünte T, Ho LM, Satzger C, et al. Central vehicle dynamics control of the robotic research platform RoboMobil. ATZelektronik worldwide. 2014 Jun;9(3):58–64.

[10] Johansen TA, Fossen TI. Control allocation—a survey. Automatica. 2013 May;49(5):1087–1103.

[11] de Castro R, Brembeck J. Lyapunov-based control allocation for over-actuated nonlinear systems. In: 2019 American Control Conference (ACC); Jul.; 2019. p. 5033–5038.

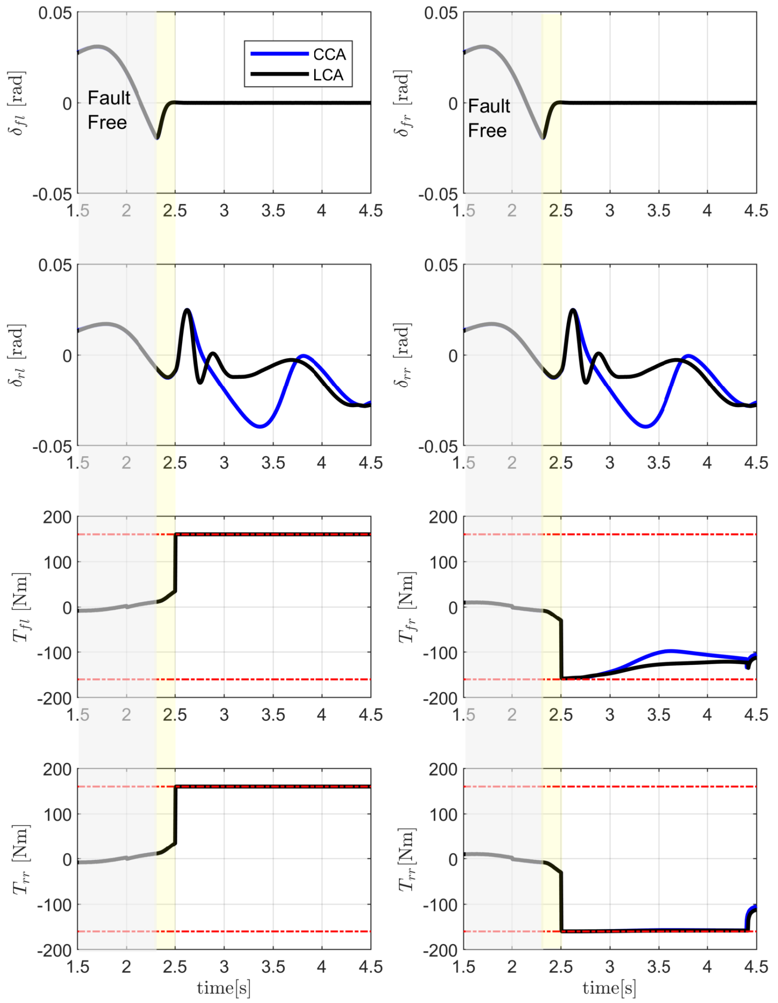


Figure 12. Control inputs (u^*) generated by the LCA and CCA during the double lane change maneuver. Actuation limits are represented in dashed red lines. There are three operation modes: i) fault-free (gray background); ii) fault injection and controller operation without knowledge of the fault (yellow background); iii) fault injection and controller operation with knowledge of the fault (white background).

[12] Guo J, Luo Y, Li K. An adaptive hierarchical trajectory following control approach of autonomous Four-Wheel independent drive electric vehicles. IEEE Trans Intell Transp Syst. 2018 Aug;19(8):2482–2492.

[13] Dizqah AM, Ballard BL, Blundell MV, et al. A Non-Convex control allocation strategy as Energy-Efficient torque distributors for On-Road and Off-Road vehicles. Control Eng Pract. 2020 Feb;95:104256.

[14] De Novellis L, Sorniotti A, Gruber P. Wheel torque distribution criteria for electric vehicles with Torque-Vectoring differentials. IEEE Trans Veh Technol. 2014 May;63(4):1593–1602.

[15] Casavola A, Garone E. Fault-tolerant adaptive control allocation schemes for overactuated systems. Int J Robust Nonlinear Control. 2010 Nov;20(17):1958–1980.

[16] Gong X, Ge W, Yan J, et al. Review on the development, control method and application prospect of Brake-by-Wire actuator. Actuators. 2020 Mar;9(1):15.

[17] Huang C, Naghdy F, Du H, et al. Fault tolerant steer-by-wire systems: An overview. Annu Rev Control. 2019 Jan;47:98–111.

[18] Wanner D, Wallmark O, Jonasson M, et al. Control allocation strategies for an electric vehicle with a wheel hub motor failure ; 2015.

[19] Alwi H, Edwards C. Fault tolerant control using sliding modes with on-line control allocation. Automatica. 2008 Jul;44(7):1859–1866.

[20] Brembeck J, Ho LM, Schaub A, et al. ROMO - the robotic electric vehicle. 1. In: 22nd IAVSD International Symposium on Dynamics of Vehicle on Roads and Tracks; 2011.

[21] Brembeck J, Bals J, Baumgartner D, et al. Recent research on automotive control at DLR institute of system dynamics and control - an overview. In: 2020 28th Mediterranean Conference on Control and Automation (MED); Sep.; 2020. p. 188–193.

[22] de Castro R, Ho LM, Brembeck J. An active Fault-Tolerant controller for overactuated electric vehicles. In: Proceedings of the 25th International Symposium on Dynamics of Vehicles on Roads and Tracks (IAVSD 2017); Sep.; 2017.

[23] Kiencke U, Nielsen L. Automotive control systems: For engine, driveline, and vehicle. Springer; 2005.

[24] Pacejka HB. Tyre and vehicle dynamics. Butterworth-Heinemann; 2006. Automotive engineering.

[25] Cristofaro A, Johansen TA. Fault tolerant control allocation using unknown input observers. Automatica. 2014 Jul;50(7):1891–1897.

[26] Ho LM. Model-Based fault detection and isolation for a novel X-by-Wire road vehicle architecture [dissertation]. University Newcastle; 2017.

[27] Jazar RN. Vehicle dynamics: Theory and application. Springer,; 2017.

[28] Lenzo B, Bucchi F, Sorniotti A, et al. On the handling performance of a vehicle with different front-to-rear wheel torque distributions. Veh Syst Dyn. 2019 Nov;57(11):1685–1704.

[29] Harkegard O, Glad ST. Resolving actuator redundancy—optimal control vs. control allocation. Automatica. 2005 Jan;41(1):137–144.

[30] Sariyildiz E, Oboe R, Ohnishi K. Disturbance observer-based robust control and its applications: 35th anniversary overview. IEEE Trans Ind Electron. 2020;67(3):2042–2053.

[31] de Castro R, Brembeck J. Cascaded trajectory tracking control for automated vehicles. In: 2019 IEEE 90th Vehicular Technology Conference (VTC2019-Fall); Sep.; 2019. p. 1–5.

[32] Biannic JM. Linear parameter-varying control strategies for aerospace applications. In: Sename O, Gaspar P, Bokor J, editors. Robust control and linear parameter varying approaches; 2013.

[33] de Castro R, Buente T, Brembeck J. Design and validation of the second generation of the robomobil's vehicle dynamics controller. In: 24th Symposium of the International Association for Vehicle System Dynamics,; 2015.

[34] Khalil HK. Nonlinear systems. Upper Saddle River, NJ: Prentice-Hall; 2002. 3rd ed.

[35] Seiler P, Balas GJ. Quasiconvex sum-of-squares programming. In: 49th IEEE Conference on Decision and Control (CDC); Dec.; 2010. p. 3337–3342.

[36] Wang L. Model predictive control system design and implementation using MATLAB®. Springer London; 2009. Advances in Industrial Control.

[37] Huyck B, Ferreau HJ, Diehl M, et al. Towards online model predictive control on a programmable logic controller: Practical considerations. Math Probl Eng. 2012 Nov;2012.

[38] Satzger C, de Castro R. Predictive brake control for electric vehicles. IEEE Trans Veh Technol. 2018 Feb;67(2):977–990.

[39] Tobolar J, Otter M, Buente T. Modelling of vehicle powertrains with the modelica PowerTrain library. In: Dynamisches Gesamtsystemverhalten von Fahrzeugantrieben; 2007.

[40] Orend R. Integrierte fahrdynamikregelung mit einzelradaktorik- ein konzept zur darstellung des fahrdynamischen optimums [dissertation]. Universitaet Erlangen; 2006.

[41] Jiang J, Yu X. Fault-tolerant control systems: A comparative study between active and

passive approaches. Annu Rev Control. 2012 Apr;36(1):60–72.

[42] Mathworks. Matlab's control system toolbox [https://www.mathworks.com/products/control.html]; 2021. Accessed: 2021-6-8.

[43] Wilson R. Linear systems theory. Prentice Hall; 1995.