

Disturbance-aware minimum-time planning strategies for motorsport vehicles with probabilistic safety certificates

Martino Gulisano^a, Matteo Masoni^a, and Marco Gabiccini,^{a,*}

^a Dipartimento di Ingegneria Civile e Industriale, Università di Pisa, Pisa, Italy.

ARTICLE HISTORY

Compiled April 3, 2025

ABSTRACT

This paper analyzes ...

KEYWORDS

Minimum lap-time trajectory planning; stochastic vehicle dynamics; probabilistic safety certificates.

1. Introduction

As the automotive industry advances towards higher levels of automation, ...

1.1. *Related work*

In recent years, extensive research has been conducted on ...

1.2. *Paper's contributions*

In this study, we investigate ...

2. Stochastic vehicle dynamics

Some text of a previous paper left for reference on equation and pseudo-code formatting.

2.1. *Kinematic Tree Structure of the Vehicle*

The vehicle is modeled as an articulated rigid-body system with chassis, knuckles, and wheels as rigid bodies. The chassis represents all sprung masses rigidly connected, while the knuckles (zero-mass bodies) decouple steering and vertical suspension motion from the wheel spindle. In the second model, knuckles are split into two zero-mass bodies, introducing an additional degree of freedom per wheel to simulate dynamic camber

* Contact: Marco Gabiccini. Email: marco.gabiccini@unipi.it

control (DCC). Wheels include all rotating unsprung masses, with non-rotating parts (e.g., brake calipers) incorporated into the wheel-tire mass.

Each body has a right-handed barycentric frame: $\{B\}$ (chassis), $\{H\}$ and $\{K\}$ (knuckles), and $\{W\}$ (wheels). Additional frames $\{G\}$ (ground) and $\{S\}$ (centerline) are defined, with x -axis of $\{S\}$ tangent to the centerline, pointing forward. Frame relationships are described by homogeneous transformation matrices $g(d, \Phi) \in SE(3)$, with d as position and Φ (Euler angles) parameterizing the rotation.

As shown in Fig. 6, bodies are connected by joints in a kinematic tree. The second model generalizes the first, where $\{H\}$ and $\{K\}$ coincide, so only the second model's kinematic tree is shown for brevity.

The joints modeled are: hub bearings, connecting the rims $\{W\}$ to the knuckles $\{H\}$ (or $\{K\}$ for the second model), and the suspension linkages, connecting the knuckles $\{H\}$ to the chassis $\{B\}$. For the second model a 1 DoF revolute joint connecting the two parts of the knuckles is also present. The chassis is a floating-base and its position and orientation with respect to the ground are unconstrained. This situation is conveniently encoded by a virtual six-DoF joint connecting $\{B\}$ to $\{G\}$. The resulting systems have $4+4+6=14$ and $4+4+4+6=18$ degrees of freedom, respectively.

2.2. Suspension Analysis

The suspensions connect the chassis to the unsprung masses. This connection cannot be described by an elementary joint, such as prismatic or revolute. Therefore, a set of six variables q_{bh} , that parameterize a homogeneous transformation matrix, is necessary. The transformation can be realized by composing three translations and three rotations (the ZXY Euler sequence was selected) using the global Product of Exponentials (PoE) formula [?], which reads:

$$g_{bh} = e^{\hat{Y}_1 q_{bh,1}} \dots e^{\hat{Y}_6 q_{bh,6}} g_{bh}(0), \quad (1)$$

where the offset $g_{bh}(0) = I$ and $Y_i = e_i$, with $e_i \in \mathbb{R}^6$ are the six normalized screw vectors (canonical basis elements in \mathbb{R}^6) associated with the Cartesian coordinates and the Euler ZXY angles in q_{bh} . Clearly, the six components of q_{bh} are not independent; the modeled suspension is, in fact, a double wishbone that has: 2 DoFs for wheels that

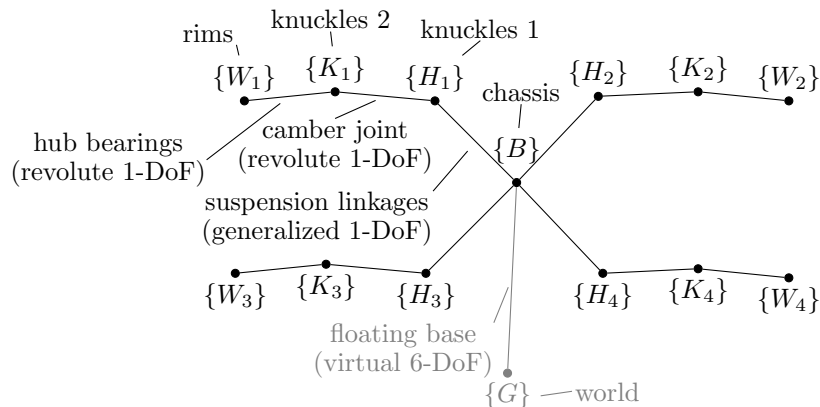


Fig. 1. Kinematic tree representing the interconnections between bodies. Bodies are referred to by their attached frame.

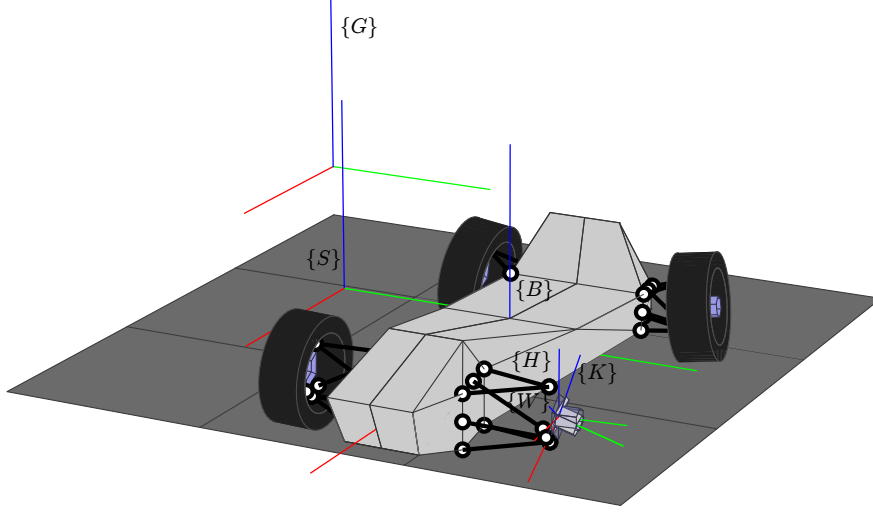


Fig. 2. Reference frames: $\{G\}$ is a fixed inertial reference frame with the z -axis antiparallel to the gravity vector; $\{S\}$ is a reference frame following the vehicle on track; $\{B\}$ and $\{K\}$ are FLU (Forward Left Up) reference frames aligned with the principal axes of inertia of the chassis and wheel, respectively; $\{H\}$ is an auxiliary reference frame necessary to add the camber DoF; and $\{W\}$ rotates about the y -axis with respect to $\{K\}$ by the wheel rotation angle.

can steer, and 1 DoF otherwise. In this case, both axles have wheels that can steer; hence, a pair of independent variables $(z; \delta)$ is associated with each wheel. A detailed analysis is omitted here for the sake of brevity. The interested reader is referred to [?] for a similar treatment using the PoE formalism.

3. Tire Model

The tangential forces exchanged between the tire and road are computed using Pacejka's *Magic Formula*:

$$[F_x, F_y] = F_{MF}(\kappa, \alpha, F_z, \gamma), \quad (2)$$

where F_x, F_y are the longitudinal and lateral forces, dependent on tire slips (κ, α) , vertical load (F_z) , and camber angle (γ) . As shown in Fig. 8, an auxiliary frame $\{N\}$ is introduced to calculate F_z . This depends upon the interpenetration d of the tire into the ground, following a penalty-based compliant tire model. This model treats the wheel as a radial spring-damper system with constant stiffness k_t and damping c_t . The vertical force is expressed as:

$$F_z = F_0 \log_2 \left(1 + 2^{\frac{k_t d + c_t \dot{d}}{F_0}} \right) \quad (3)$$

which is differentiable for $d = 0$ and $\dot{d} = 0$, improving numerical stability. The model ensures F_z approaches zero when the tire loses contact with the ground, while maintaining a non-zero gradient to guide optimization algorithms in scenarios where the

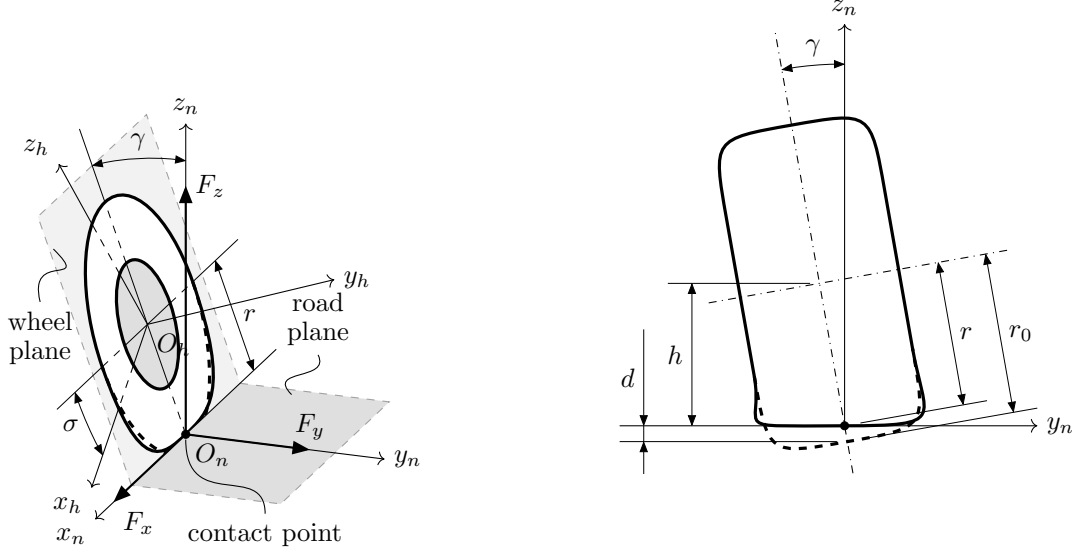


Fig. 3. Decomposition of the tyre forces along the axes of the auxiliary frame $\{N\}$. The frame $\{N\}$ is such that: x_n lies along the intersection between the wheel plane ($x_h z_h$ -plane) and the road plane ($x_s y_s$ -plane), pointing forward; y_n points left (w.r.t. the driver) along the intersection line between the road plane ($x_s y_s$ -plane) and the transverse plane ($y_h z_s$ -plane) through the wheel center O_h ; z_n is normal to the road surface. The contact point is estimated in correspondence to the origin O_n . On the right, we report a detail of the tyre vertical deformation.

tyre detaches from the ground or encounters sudden changes.

3.1. Camber System Modeling

The dynamic camber control (DCC) system is modeled as a simplified actuated revolute joint connecting the two virtual halves of the knuckle, though a real implementation might require a more complex 1-DoF joint. While this simplification may influence joint values from the optimizer, it results in nearly the same optimal wheel posture. The choice was driven by the limited literature on such systems for FSAE vehicles, which is the vehicle type employed in our analysis. A realistic model, if available, could be characterized using the same process applied to the suspension. During the design phase of the actual joint, some parameters could be treated as time-invariant optimization variables, constrained within dimensional bounds, to achieve the best performance configuration.

In Table 2, we report the range of variation of function $q_{hk}(t)$ and its first and second derivatives. Considering the scarcity of open literature available, reasonable values have been selected. In Figure 9 the wheel is shown in a general posture with zero camber (shaded black) and with maximum positive camber (solid black). Due to the convention adopted for rotations, this posture corresponds to the minimum value of q_{hk} .

q_{hk} (rad)	\dot{q}_{hk} (rad/s)	\ddot{q}_{hk} (rad/s ²)
$\pm\pi/20$	$\pm\pi/20$	$\pm\pi/10$

Table 1. Upper and lower bounds for camber angle value $q_{hk}(t)$ and its derivatives.

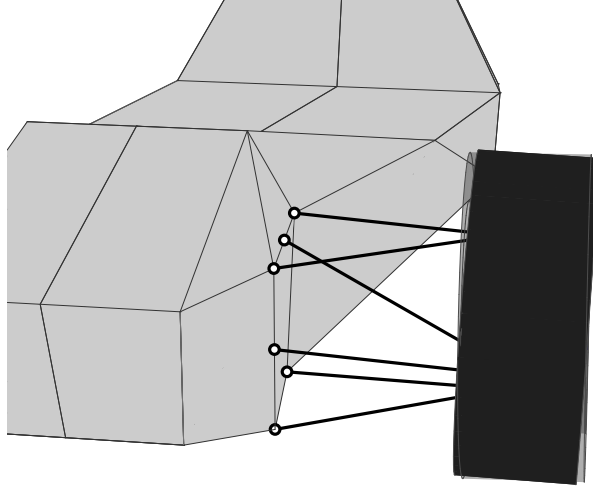


Fig. 4. Real scale representation of the wheel in two configurations: $q_{hk} = 0$ (shaded surface) and $q_{hk} = -\pi/20$ (solid surface).

3.2. External Wrenches

The external wrenches acting on each body, from wheels to chassis, are expressed in body-fixed components. Only gravity, aerodynamic forces, and road-tyre interactions are considered. Following [?], the aerodynamic wrench in frame $\{B\}$ is $W_a = -\frac{1}{2}\rho S v_{gb,1}^2 [C_x, 0, C_z, 0, -h_0 C_x - a_f C_{zf} + a_r C_{zr}, 0]^T$, where ρ is air density, S is the frontal area, $v_{gb,1}$ the forward velocity (component of *distal rigid-body twist* V_{gb} , see [? , p. 4]), and C_x, C_z are drag/lift coefficients. C_z is split into $C_{zf} = k_a C_z$ and $C_{zr} = (1 - k_a) C_z$ by the aerodynamic balance coefficient k_a .

The total external wrench on a wheel is $W_{we} = \text{Ad}_{g_{hn}}^* W_t + W_{w_h}$, where $W_t = [F_x, F_y, F_z, 0, 0, 0]^T$ is the ground wrench transferred from $\{N\}$ to $\{H\}$ by the *starred* Adjoint operator, and $W_{w_h} = [F_{w_h}^T R_{hg}^T, 0, 0, 0]^T$ is the weight contribution with $F_{w_h} = [0, 0, -m_h g]^T$ in frame $\{G\}$, where m_h is the wheel mass.

Knuckles experience no external forces, therefore:

$$W_{he} = [0, 0, 0, 0, 0, 0]^T \quad \text{and} \quad W_{ke} = [0, 0, 0, 0, 0, 0]^T. \quad (4)$$

The chassis's external wrench is $W_{be} = W_a + W_{w_b}$, where W_a and W_{w_b} are aerodynamic and weight contributions, respectively. For $W_{w_b} = [F_{w_b}^T R_{bg}^T, 0, 0, 0]^T$, the weight force is $F_{w_b} = [0, 0, -m_b g]^T$ in frame $\{G\}$, with m_b as the chassis mass.

3.3. Recursive computation of the dynamics equations with ABA

Thanks to the ABA algorithm, assuming known joint positions and velocities and joint forces and torques, it is possible to compute recursively and efficiently the accelerations

of the joint variables. Following [?], twists are streamlined as follows: $V_{gb}^b = V_{gb} = V_b$, $V_{bh}^h = V_{bh}$, $V_{hw}^h = V_{hw}$. For the second model (DCC) the last expression is replaced with $V_{kw}^k = V_{kw}$ and $V_{hk}^k = V_{hk}$ is added. Given that the first model (without camber control) is a particular case of the DCC one, to avoid duplications only the algorithm for DCC is reported here.

The traditional ABA requires knowledge of the active forces and torques, denoted as τ_{ij} , exerted by the parent body (i) on the child body (j). Its purpose is to solve the *Forward Dynamics* problem of an articulated systems of rigid bodies. In our floating-base system, given the twist V_b of the parent body $\{B\}$, the values and time derivatives of all joint variables (q and \dot{q}), and the torque vector (τ), the objective is to compute the joint accelerations (\ddot{V}_b and \ddot{q}). The overall computation is performed in three sweeps defined recursively. One of the greatest computational benefits when it is employed to assemble symbolic dynamic equations, as in our case, is that *no matrix inversion is required*, thus contributing to streamline the algebraic expressions.

The first step of the algorithm, as described in **Step 1**, performs the forward propagation of rigid-body velocities: from the root to the leaves of the tree. In Algorithm 4 the terms “Knuckle 1” and “Knuckle 2” are used to identify the two halves of the knuckle, where, using kinematic trees terminology, the first is the parent of the second. The Adjoint operator is defined as follows:

$$\text{Ad}_g = \begin{bmatrix} R & \hat{d}R \\ 0 & R \end{bmatrix}. \quad (5)$$

In the case $g_{bh} \in SE(3)$ is considered, $R_{bh} \in SO(3)$ and $d_{bh} \in \mathbb{R}^3$ are functions of (z, δ) . In the case of g_{hk} , the expressions are simply $d_{hk} = [0, 0, 0]^T$ and $R = \text{rot}_X(q_{hk})$, the latter being an elementary rotation about the common x -axis of the angle q_{hk} .

Step 1 Forward Propagation of Velocity

- 1: **for** $h = h_1, h_2, h_3, h_4$ **do**
 - 2: $V_{bh} = J_{bh,z}\dot{z} + J_{bh,\delta}\dot{\delta}$
 - 3: $V_h = \text{Ad}_{g_{bh}}^{-1} V_b + V_{bh}$ ▷ Knuckle 1 Rigid-Body Velocity
 - 4: $V_{hk} = J_{hk}\dot{q}_{hk}$
 - 5: $V_k = \text{Ad}_{g_{hk}}^{-1} V_h + V_{hk}$ ▷ Knuckle 2 Rigid-Body Velocity
 - 6: $V_{kw} = J_{kw}\omega$
 - 7: $V_w = V_k + V_{kw}$ ▷ Rim Rigid-Body Velocity
 - 8: **end for**
-

The second step of the algorithm, as detailed in **Step 2**, performs the backward propagation of inertia and bias terms.

In the case the DCC architecture is considered, the camber angle acceleration \ddot{q}_{hk} is known, since $q_{hk(t)}$ is the assumed input, while the torque τ_{hk} has to be computed together with the rest of unknown accelerations, which are the outcome of the third sweep. This leads to a *Hybrid Dynamics* problem that requires to carefully revisit the ABA’s second and third steps. The approach used is inspired by [? , Section 2.3] where the notation and conventions here employed were adapted according to [?].

Starting from the leaf nodes, which in our case are represented by the wheels ($\{W\}$), after providing information such as external wrenches, twists, joint geometry, body

Step 2 Backward Propagation of Articulated Inertia and Bias

- 1: **for** $h = h_1, h_2, h_3, h_4$ **do**
 - 2: $\hat{M}_w = M_w$ ▷ Rim Articulated Inertia
 - 3: $\hat{B}_w = -W_{we} + \text{ad}_{V_w}^* M_w V_w$ ▷ Rim Articulated Bias
 - 4: $\bar{M}_w = \hat{M}_w - \frac{\hat{M}_w J_{hw} J_{hw}^T \hat{M}_w}{J_{hw}^T \hat{M}_w J_{hw}}$
 - 5: $\bar{B}_w = \hat{B}_w - \frac{\hat{M}_w J_{hw} J_{hw}^T \hat{B}_w}{J_{hw}^T \hat{M}_w J_{hw}} - \bar{M}_w \text{ad}_{V_{hw}} V_w$
 - 6: $\hat{M}_k = \bar{M}_w$ ▷ Knuckle 2 Articulated Inertia
 - 7: $\hat{B}_k = \bar{B}_w$ ▷ Knuckle 2 Articulated Bias
 - 8: $\hat{M}_h = \text{Ad}_{g_{hk}}^* \hat{M}_k \text{Ad}_{g_{hk}}^{-1}$ ▷ Knuckle 1 Articulated Inertia
 - 9: $\hat{B}_h = \text{Ad}_{g_{hk}}^* (\hat{M}_k (J_{hk} \ddot{q}_{hk} - \text{ad}_{V_{hk}} \text{Ad}_{g_{hk}} V_k) + \hat{B}_k)$ ▷ Knuckle 1 Articulated Bias
 - 10: $\bar{M}_h = \hat{M}_h - \frac{\hat{M}_h J_{bh,z} J_{bh,z}^T \hat{M}_h}{J_{bh,z}^T \hat{M}_h J_{bh,z}}$
 - 11: $\bar{B}_h = \hat{B}_h - \frac{\hat{M}_h J_{bh,z} (J_{bh,z}^T \hat{B}_h - \tau)}{J_{bh,z}^T \hat{M}_h J_{bh,z}} - \bar{M}_h (\text{ad}_{V_{bh}} V_h - \dot{J}_{bh,z} \dot{z} - \dot{J}_{bh,\delta} \dot{\delta} - J_{bh,\delta} \ddot{\delta})$
 - 12: **end for**
 - 13: $\hat{M}_b = M_b + \sum_h \text{Ad}_{g_{bh}}^* \bar{M}_h \text{Ad}_{g_{bh}}^{-1}$ ▷ Chassis Articulated Inertia
 - 14: $\hat{B}_b = -W_{be} + \text{ad}_{V_b}^* M_b V_b + \sum_h \text{Ad}_{g_{bh}}^* \bar{B}_h$ ▷ Chassis Articulated Bias
-

masses, joint forces / torques for traditional connections, and time evolution of variables defining kinematically imposed joints, the backward propagation of inertia and bias can be performed.

Finally the last step, denoted as **Step 3**, starting from the root node, computes joint variable accelerations and consequently body twists derivatives. For the kinematically controlled camber joints, the torques τ_{hk} needed to maintain the desired state of motion (DCC architecture) are computed as a by-product of the local inverse dynamics. This information is crucial to prevent actuators from saturating or operating outside their bandwidth in the planned trajectories.

The root twist acceleration is determined by solving equation $W = \hat{M}\dot{V} + \hat{B}$, considering that the chassis is free-floating. The other unknown quantities, such as variable accelerations and torques, are computed projecting the Newton-Euler equations of each body along the respective Jacobian and then solving for the variable. See also [?] for more details in the derivation of the equations.

3.4. Dynamic Model

The dynamics equations, assembled via the procedure described in Sec. 6.3, are implemented in the software code by using a state-space representation $\dot{x} = f(x, u)$. The state vector is $x = (q_{gb}; V_{gb}; z; \dot{z}; \omega; q_{hk}; \dot{q}_{hk}) \in \mathbb{R}^{32}$, with components defined in

Step 3 Forward Propagation of Acceleration

```

1:  $\dot{V}_b = -\hat{M}_b^{-1} \hat{B}_b$  ▷ Chassis Rigid-Body Acceleration
2: for  $h = h_1, h_2, h_3, h_4$  do
3:    $\ddot{z} = -\frac{J_{bh,z}^T (\hat{B}_h + \hat{M}_h (\text{Ad}_{g_{bh}}^{-1} \dot{V}_b - \text{ad}_{V_{bh}} V_h + \dot{J}_{bh,z} \dot{z} + \dot{J}_{bh,\delta} \dot{\delta} + J_{bh,\delta} \ddot{\delta})) - \tau}{J_{bh,z}^T \hat{M}_h J_{bh,z}}$ 
4:    $\dot{V}_{bh} = \dot{J}_{bh,z} \dot{z} + \dot{J}_{bh,\delta} \dot{\delta} + J_{bh,z} \ddot{z} + J_{bh,\delta} \ddot{\delta}$ 
5:    $\dot{V}_h = \text{Ad}_{g_{bh}}^{-1} \dot{V}_b - \text{ad}_{V_{bh}} V_h + \dot{V}_{bh}$  ▷ Knuckle 1 Rigid-Body Acceleration
6:    $\tau_{hk} = J_{hk}^T (\hat{B}_k + \hat{M}_k (\text{Ad}_{g_{kh}} \dot{V}_h - \text{ad}_{V_{hk}} V_k + J_{hk} \ddot{q}_{hk}))$ 
7:    $\dot{V}_{bh} = J_{hk} \ddot{q}_{hk}$ 
8:    $\dot{V}_k = \text{Ad}_{g_{hk}}^{-1} \dot{V}_h - \text{ad}_{V_{hk}} V_k + \dot{V}_{bh}$  ▷ Knuckle 2 Rigid-Body Acceleration
9:    $\dot{\omega} = -\frac{J_{kw}^T (\hat{B}_w + \hat{M}_w (\dot{V}_k - \text{ad}_{V_{kw}} V_w))}{J_{kw}^T \hat{M}_w J_{kw}}$ 
10: end for

```

previous sections. As already mentioned, in the model the camber joint is kinematically controlled as per the DCC architecture. Hence, the associated control vector is $u = (T_a; T_b; \delta; \ddot{q}_{hk}) \in \mathbb{R}^{16}$, where each element is in \mathbb{R}^4 and collects the controls of the branch to each wheel. The torque is split into acceleration and braking components, allowing for minor modifications to model on-board motors coupled with traditional brakes and/or in-wheel motors as well. These components have their reaction torques in two different bodies in case of on-board motors and traditional brakes: the chassis and the knuckles, respectively. The δ vector collects the four translations of the steering rod attachment points, as described in Section 5.2, while for the vector \ddot{q}_{hk} no further explanation should be necessary. Clearly, this representation is valid for the model with active camber. The simpler model (no DCC) shares with the former the first 24 components of the state vector and the first 12 of the control vector.

The time derivative of the state vector has also been expressed previously except for the first element, q_{gb} . Its derivative is a function of the state and can be recovered using a standard procedure [?] as follows:

$$\dot{q}_{gb} = J_{gb}^{-1}(q_{gb}) V_{gb}, \quad (6)$$

where $J_{gb}(q_{gb}) \in \mathbb{R}^{6 \times 6}$ is the Jacobian of the virtual joint connecting the ground to the chassis, and it is computed via the PoE parameterization of the transformation g_{gb} .

4. Open-loop planning via n-steps ahead predictions

The model described in Sec. ?? is embedded into a minimum lap time problem which is briefly outlined below.

4.1. Direct Collocation

The objective is to recast the MLTP into a Non Linear Program (NLP) using a direct orthogonal collocation approach, specifically based on Gauss-Legendre collocation points. The problem is formulated within the CasADi framework [?] and solved using the interior point method implemented in IPOPT [?]. The resulting NLP can be expressed in the form:

$$\underset{x,v,u,z}{\text{minimize}} \quad \sum_{k=1}^N L_k(x_k, v_k, u_k, z_k) \quad (7)$$

$$\text{subject to} \quad g_k(x_{k-1}, x_k, v_k, u_k, z_k) = 0, \quad k = 1, \dots, N, \quad (8)$$

$$h_k(x_k, u_k, z_k) \leq 0, \quad k = 1, \dots, N. \quad (9)$$

In Eqs. (7), (8) and (9), in addition to the state and control vectors, v_k and z_k are included, representing the state vector at the collocation points and the algebraic quantities, respectively. The controls and the algebraic variables are approximated as piecewise constant, while the states are approximated as quadratic polynomials within each finite element. Specifically, the algebraic variables in the problem are the time steps ν_k and the lateral and vertical displacements of the Center of Mass relative to the origin of the k -th frame $\{S\}$, attached to the track's centerline (e_y and e_z). Their function is to serve as handle variables in the optimization process to promote problem sparsity.

The track centerline is represented by a curvilinear abscissa $\alpha \in [0, 1]$, sampled at $N + 1$ points $\alpha_0, \dots, \alpha_N$. Each grid point is associated with a reference frame $\{S_k\}$, whose origin lies at the corresponding point on the track centerline, as illustrated in Figure 5. A more detailed description of the 3D track modeling can be found in [?]. Specifically, this representation allows us to capture track slope and banking, but it does not support non-developable track surfaces.

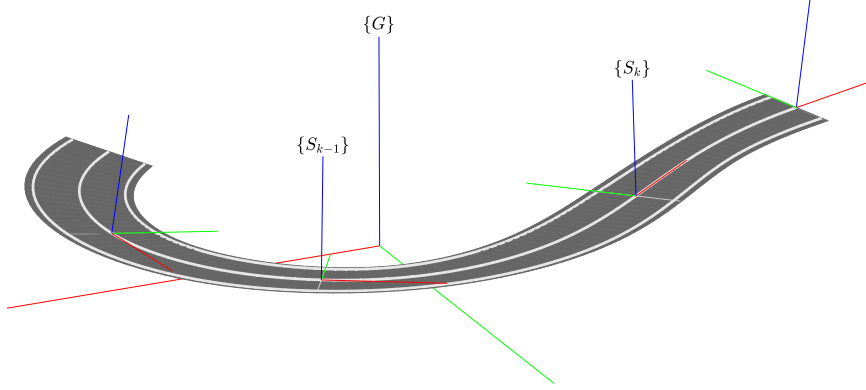


Fig. 5. Track frames corresponding to adjacent nodes of the grid over α . Note that the grid points need not be equispaced. If desired, they can be chosen irregularly, so as to capture more features of the track where it is more critical (e.g. in proximity to corners) and save space where it is more regular (e.g. on the straights).

In addition to the classical collocation and continuity constraints [?], the inequalities in Eq. (9) for our MLTP ensure the vehicle remains within track bounds. These include the requirement for the CoM to lie on the yz plane of the frame $\{S_k\}$, as well as power

and adherence limits.

In our optimizations, the cost function at each step is expressed as follows:

$$L_k = \nu_k^2 + k_\delta(\delta_k - \delta_{k-1})^2 + k_T(T_k - T_{k-1})^2 + k_v x_{k,2}^2 + l_k(\ddot{q}_{hk,k}, \delta_k, x_k), \quad (10)$$

where the first term penalizes travel time, the second and third prevent abrupt variations of the inputs and avoid singular arcs. The quadratic penalization ensures a well-defined dependence of the Hamiltonian on the control variables, preventing indeterminate solutions. The last two serve to shape the driving behaviors according to the designer preferences.

Their primary goal is to improve the optimization landscape, ensuring that the solver is less likely trapped by undesirable local minima. As an example, the fourth term, which includes the lateral velocity $x_{k,2}$, has been included to prevent undesirable behaviors which could emerge as a result of the increased system's DoFs in combination with the simplified aerodynamic modeling adopted in this work. Some local minima could automatically be avoided by incorporating more realistic aerodynamic maps and tyre wear models. These enhancements will be explored in future works. In this context, an interesting direction is presented in [?], where the authors introduce an aerodynamic model that explicitly accounts for the influence of suspension and vehicle configuration. This approach offers a more comprehensive representation of the aero forces and could be effectively used to refine the current aerodynamic map. By incorporating such a model, we expect not only a more physically consistent representation of the aerodynamics but also a smoother and more meaningful optimization landscape, reducing the occurrence of non-physical local minima and improving overall solution robustness.

In the current context, we imposed coefficient k_v to be a function of the (absolute value of) track curvature c_k according to Eq. (11). Here, the two constant \bar{k}_v and k_c are selected heuristically for each track; the first increases/decreases the weight of the cost term, while the second varies the hardness of the intervention in the range $[0, 1]$.

$$k_v = \bar{k}_v(1 - \tanh(k_c c_k)) \quad (11)$$

To ensure reproducibility, we report the values of the main penalty weights for the cost function terms used in the Siena track simulations: $k_\delta = 0.05$, $k_T = 0.005$, $\bar{k}_v = 0.005$, and $k_c = 100$.

5. Closed-loop uncertainty-aware planning over the full horizon

Building upon our previous paper that presented a 3D multibody model suitable for embedding within an MLTP problem [?] - treated here as the *baseline* model - two models with increasing complexity are presented. The first represents a vehicle capable of independently steering all four wheels (AS) and applying four independent torques to each wheel (AIWD, aka, torque vectoring), whereas the second model additionally incorporates an active camber variation system for each wheel (DCC).

5.1. Kinematic Tree Structure of the Vehicle

The vehicle is modeled as an articulated rigid-body system with chassis, knuckles, and wheels as rigid bodies. The chassis represents all sprung masses rigidly connected, while the knuckles (zero-mass bodies) decouple steering and vertical suspension motion from the wheel spindle. In the second model, knuckles are split into two zero-mass bodies, introducing an additional degree of freedom per wheel to simulate dynamic camber control (DCC). Wheels include all rotating unsprung masses, with non-rotating parts (e.g., brake calipers) incorporated into the wheel-tire mass.

Each body has a right-handed barycentric frame: $\{B\}$ (chassis), $\{H\}$ and $\{K\}$ (knuckles), and $\{W\}$ (wheels). Additional frames $\{G\}$ (ground) and $\{S\}$ (centerline) are defined, with x -axis of $\{S\}$ tangent to the centerline, pointing forward. Frame relationships are described by homogeneous transformation matrices $g(d, \Phi) \in SE(3)$, with d as position and Φ (Euler angles) parameterizing the rotation.

As shown in Fig. 6, bodies are connected by joints in a kinematic tree. The second model generalizes the first, where $\{H\}$ and $\{K\}$ coincide, so only the second model's kinematic tree is shown for brevity.

The joints modeled are: hub bearings, connecting the rims $\{W\}$ to the knuckles $\{H\}$ (or $\{K\}$ for the second model), and the suspension linkages, connecting the knuckles $\{H\}$ to the chassis $\{B\}$. For the second model a 1 DoF revolute joint connecting the two parts of the knuckles is also present. The chassis is a floating-base and its position and orientation with respect to the ground are unconstrained. This situation is conveniently encoded by a virtual six-DoF joint connecting $\{B\}$ to $\{G\}$. The resulting systems have $4+4+6=14$ and $4+4+4+6=18$ degrees of freedom, respectively.

5.2. Suspension Analysis

The suspensions connect the chassis to the unsprung masses. This connection cannot be described by an elementary joint, such as prismatic or revolute. Therefore, a set of six variables q_{bh} , that parameterize a homogeneous transformation matrix, is necessary. The transformation can be realized by composing three translations and three rotations (the ZXY Euler sequence was selected) using the global Product of Exponentials (PoE)

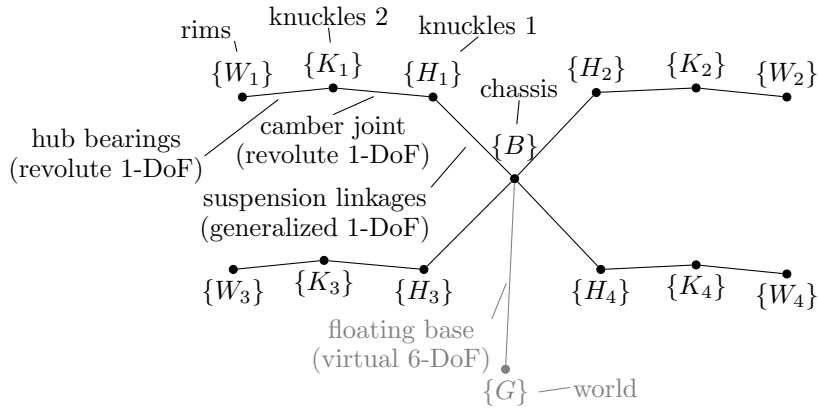


Fig. 6. Kinematic tree representing the interconnections between bodies. Bodies are referred to by their attached frame.

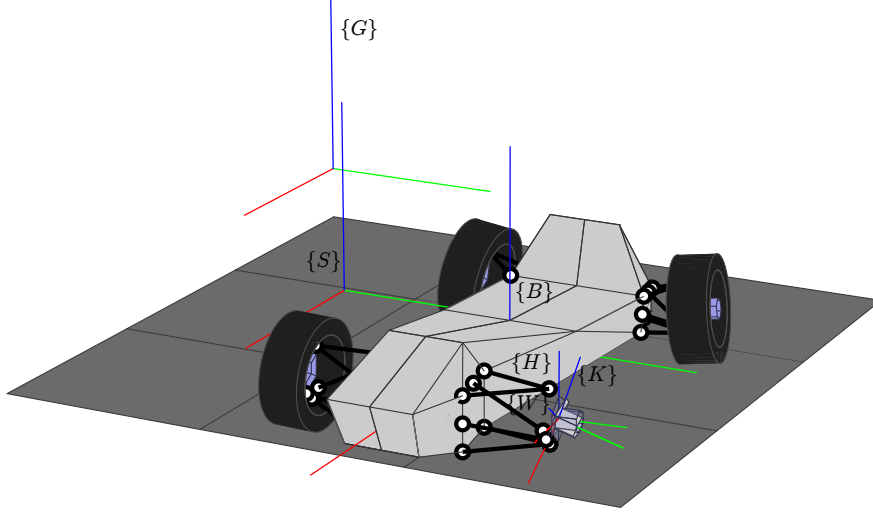


Fig. 7. Reference frames: $\{G\}$ is a fixed inertial reference frame with the z -axis antiparallel to the gravity vector; $\{S\}$ is a reference frame following the vehicle on track; $\{B\}$ and $\{K\}$ are FLU (Forward Left Up) reference frames aligned with the principal axes of inertia of the chassis and wheel, respectively; $\{H\}$ is an auxiliary reference frame necessary to add the camber DoF; and $\{W\}$ rotates about the y -axis with respect to $\{K\}$ by the wheel rotation angle.

formula [?], which reads:

$$g_{bh} = e^{\hat{Y}_1 q_{bh,1}} \dots e^{\hat{Y}_6 q_{bh,6}} g_{bh}(0), \quad (12)$$

where the offset $g_{bh}(0) = I$ and $Y_i = e_i$, with $e_i \in \mathbb{R}^6$ are the six normalized screw vectors (canonical basis elements in \mathbb{R}^6) associated with the Cartesian coordinates and the Euler ZXY angles in q_{bh} . Clearly, the six components of q_{bh} are not independent; the modeled suspension is, in fact, a double wishbone that has: 2 DoFs for wheels that can steer, and 1 DoF otherwise. In this case, both axles have wheels that can steer; hence, a pair of independent variables $(z; \delta)$ is associated with each wheel. A detailed analysis is omitted here for the sake of brevity. The interested reader is referred to [?] for a similar treatment using the PoE formalism.

6. Tire Model

The tangential forces exchanged between the tire and road are computed using Pacejka's *Magic Formula*:

$$[F_x, F_y] = F_{MF}(\kappa, \alpha, F_z, \gamma), \quad (13)$$

where F_x, F_y are the longitudinal and lateral forces, dependent on tire slips (κ, α) , vertical load (F_z) , and camber angle (γ) . As shown in Fig. 8, an auxiliary frame $\{N\}$ is introduced to calculate F_z . This depends upon the interpenetration d of the tire into the ground, following a penalty-based compliant tire model. This model treats the wheel as a radial spring-damper system with constant stiffness k_t and damping c_t .

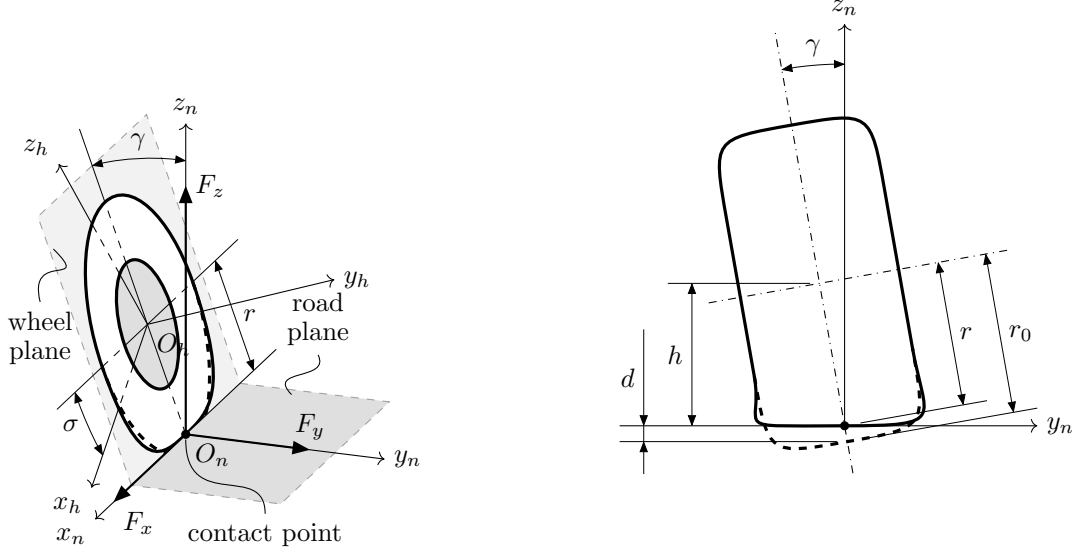


Fig. 8. Decomposition of the tyre forces along the axes of the auxiliary frame $\{N\}$. The frame $\{N\}$ is such that: x_n lies along the intersection between the wheel plane ($x_h z_h$ -plane) and the road plane ($x_s y_s$ -plane), pointing forward; y_n points left (w.r.t. the driver) along the intersection line between the road plane ($x_s y_s$ -plane) and the transverse plane ($y_h z_s$ -plane) through the wheel center O_h ; z_n is normal to the road surface. The contact point is estimated in correspondence to the origin O_n . On the right, we report a detail of the tyre vertical deformation.

The vertical force is expressed as:

$$F_z = F_0 \log_2 \left(1 + 2^{\frac{k_t d + c_t \dot{d}}{F_0}} \right) \quad (14)$$

which is differentiable for $d = 0$ and $\dot{d} = 0$, improving numerical stability. The model ensures F_z approaches zero when the tire loses contact with the ground, while maintaining a non-zero gradient to guide optimization algorithms in scenarios where the tire detaches from the ground or encounters sudden changes.

6.1. Camber System Modeling

The dynamic camber control (DCC) system is modeled as a simplified actuated revolute joint connecting the two virtual halves of the knuckle, though a real implementation might require a more complex 1-DoF joint. While this simplification may influence joint values from the optimizer, it results in nearly the same optimal wheel posture. The choice was driven by the limited literature on such systems for FSAE vehicles, which is the vehicle type employed in our analysis. A realistic model, if available, could be characterized using the same process applied to the suspension. During the design phase of the actual joint, some parameters could be treated as time-invariant optimization variables, constrained within dimensional bounds, to achieve the best performance configuration.

In Table 2, we report the range of variation of function $q_{hk}(t)$ and its first and

second derivatives. Considering the scarcity of open literature available, reasonable values have been selected. In Figure 9 the wheel is shown in a general posture with zero camber (shaded black) and with maximum positive camber (solid black). Due to the convention adopted for rotations, this posture corresponds to the minimum value of q_{hk} .

q_{hk} (rad)	\dot{q}_{hk} (rad/s)	\ddot{q}_{hk} (rad/s ²)
$\pm\pi/20$	$\pm\pi/20$	$\pm\pi/10$

Table 2. Upper and lower bounds for camber angle value $q_{hk}(t)$ and its derivatives.

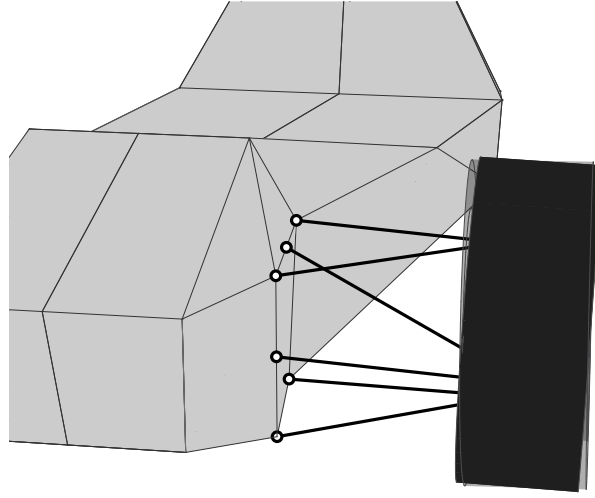


Fig. 9. Real scale representation of the wheel in two configurations: $q_{hk} = 0$ (shaded surface) and $q_{hk} = -\pi/20$ (solid surface).

6.2. External Wrenches

The external wrenches acting on each body, from wheels to chassis, are expressed in body-fixed components. Only gravity, aerodynamic forces, and road-tyre interactions are considered. Following [?], the aerodynamic wrench in frame $\{B\}$ is $W_a = -\frac{1}{2}\rho S v_{gb,1}^2 [C_x, 0, C_z, 0, -h_0 C_x - a_f C_{zf} + a_r C_{zr}, 0]^T$, where ρ is air density, S is the frontal area, $v_{gb,1}$ the forward velocity (component of *distal rigid-body twist* V_{gb} , see [? , p. 4]), and C_x, C_z are drag/lift coefficients. C_z is split into $C_{zf} = k_a C_z$ and $C_{zr} = (1 - k_a) C_z$ by the aerodynamic balance coefficient k_a .

The total external wrench on a wheel is $W_{we} = \text{Ad}_{g_{hn}}^* W_t + W_{w_h}$, where $W_t = [F_x, F_y, F_z, 0, 0, 0]^T$ is the ground wrench transferred from $\{N\}$ to $\{H\}$ by the *starred* Adjoint operator, and $W_{w_h} = [F_{w_h}^T R_{hg}^T, 0, 0, 0]^T$ is the weight contribution with $F_{w_h} = [0, 0, -m_h g]^T$ in frame $\{G\}$, where m_h is the wheel mass.

Knuckles experience no external forces, therefore:

$$W_{he} = [0, 0, 0, 0, 0, 0]^T \quad \text{and} \quad W_{ke} = [0, 0, 0, 0, 0, 0]^T. \quad (15)$$

The chassis's external wrench is $W_{be} = W_a + W_{w_b}$, where W_a and W_{w_b} are aerodynamic

and weight contributions, respectively. For $W_{w_b} = [F_{w_b}^T R_{bg}^T, 0, 0, 0]^T$, the weight force is $F_{w_b} = [0, 0, -m_b g]^T$ in frame $\{G\}$, with m_b as the chassis mass.

6.3. Recursive computation of the dynamics equations with ABA

Thanks to the ABA algorithm, assuming known joint positions and velocities and joint forces and torques, it is possible to compute recursively and efficiently the accelerations of the joint variables. Following [?], twists are streamlined as follows: $V_{gb}^b = V_{gb} = V_b$, $V_{bh}^h = V_{bh}$, $V_{hw}^h = V_{hw}$. For the second model (DCC) the last expression is replaced with $V_{kw}^k = V_{kw}$ and $V_{hk}^k = V_{hk}$ is added. Given that the first model (without camber control) is a particular case of the DCC one, to avoid duplications only the algorithm for DCC is reported here.

The traditional ABA requires knowledge of the active forces and torques, denoted as τ_{ij} , exerted by the parent body (i) on the child body (j). Its purpose is to solve the *Forward Dynamics* problem of an articulated systems of rigid bodies. In our floating-base system, given the twist V_b of the parent body $\{B\}$, the values and time derivatives of all joint variables (q and \dot{q}), and the torque vector (τ), the objective is to compute the joint accelerations (\dot{V}_b and \ddot{q}). The overall computation is performed in three sweeps defined recursively. One of the greatest computational benefits when it is employed to assemble symbolic dynamic equations, as in our case, is that *no matrix inversion is required*, thus contributing to streamline the algebraic expressions.

The first step of the algorithm, as described in **Step 1**, performs the forward propagation of rigid-body velocities: from the root to the leaves of the tree. In Algorithm 4 the terms “Knuckle 1” and “Knuckle 2” are used to identify the two halves of the knuckle, where, using kinematic trees terminology, the first is the parent of the second. The Adjoint operator is defined as follows:

$$\text{Ad}_g = \begin{bmatrix} R & \hat{d}R \\ 0 & R \end{bmatrix}. \quad (16)$$

In the case $g_{bh} \in SE(3)$ is considered, $R_{bh} \in SO(3)$ and $d_{bh} \in \mathbb{R}^3$ are functions of (z, δ) . In the case of g_{hk} , the expressions are simply $d_{hk} = [0, 0, 0]^T$ and $R = \text{rot}_x(q_{hk})$, the latter being an elementary rotation about the common x -axis of the angle q_{hk} .

Step 4 Forward Propagation of Velocity

- 1: **for** $h = h_1, h_2, h_3, h_4$ **do**
 - 2: $V_{bh} = J_{bh,z} \dot{z} + J_{bh,\delta} \dot{\delta}$
 - 3: $V_h = \text{Ad}_{g_{bh}}^{-1} V_b + V_{bh}$ ▷ Knuckle 1 Rigid-Body Velocity
 - 4: $V_{hk} = J_{hk} \dot{q}_{hk}$
 - 5: $V_k = \text{Ad}_{g_{hk}}^{-1} V_h + V_{hk}$ ▷ Knuckle 2 Rigid-Body Velocity
 - 6: $V_{kw} = J_{kw} \omega$
 - 7: $V_w = V_k + V_{kw}$ ▷ Rim Rigid-Body Velocity
 - 8: **end for**
-

The second step of the algorithm, as detailed in **Step 2**, performs the backward propagation of inertia and bias terms.

In the case the DCC architecture is considered, the camber angle acceleration \ddot{q}_{hk}

Step 5 Backward Propagation of Articulated Inertia and Bias

- 1: **for** $h = h_1, h_2, h_3, h_4$ **do**
 - 2: $\hat{M}_w = M_w$ ▷ Rim Articulated Inertia
 - 3: $\hat{B}_w = -W_{we} + \text{ad}_{V_w}^* M_w V_w$ ▷ Rim Articulated Bias
 - 4: $\bar{M}_w = \hat{M}_w - \frac{\hat{M}_w J_{hw} J_{hw}^T \hat{M}_w}{J_{hw}^T \hat{M}_w J_{hw}}$
 - 5: $\bar{B}_w = \hat{B}_w - \frac{\hat{M}_w J_{hw} J_{hw}^T \hat{B}_w}{J_{hw}^T \hat{M}_w J_{hw}} - \bar{M}_w \text{ad}_{V_{hw}} V_w$
 - 6: $\hat{M}_k = \bar{M}_w$ ▷ Knuckle 2 Articulated Inertia
 - 7: $\hat{B}_k = \bar{B}_w$ ▷ Knuckle 2 Articulated Bias
 - 8: $\hat{M}_h = \text{Ad}_{g_{hk}}^* \hat{M}_k \text{Ad}_{g_{hk}}^{-1}$ ▷ Knuckle 1 Articulated Inertia
 - 9: $\hat{B}_h = \text{Ad}_{g_{hk}}^* (\hat{M}_k (J_{hk} \ddot{q}_{hk} - \text{ad}_{V_{hk}} \text{Ad}_{g_{hk}} V_k) + \hat{B}_k)$ ▷ Knuckle 1 Articulated Bias
 - 10: $\bar{M}_h = \hat{M}_h - \frac{\hat{M}_h J_{bh,z} J_{bh,z}^T \hat{M}_h}{J_{bh,z}^T \hat{M}_h J_{bh,z}}$
 - 11: $\bar{B}_h = \hat{B}_h - \frac{\hat{M}_h J_{bh,z} (J_{bh,z}^T \hat{B}_h - \tau)}{J_{bh,z}^T \hat{M}_h J_{bh,z}} - \bar{M}_h (\text{ad}_{V_{bh}} V_h - \dot{J}_{bh,z} \dot{z} - \dot{J}_{bh,\delta} \dot{\delta} - J_{bh,\delta} \ddot{\delta})$
 - 12: **end for**
 - 13: $\hat{M}_b = M_b + \sum_h \text{Ad}_{g_{bh}}^* \bar{M}_h \text{Ad}_{g_{bh}}^{-1}$ ▷ Chassis Articulated Inertia
 - 14: $\hat{B}_b = -W_{be} + \text{ad}_{V_b}^* M_b V_b + \sum_h \text{Ad}_{g_{bh}}^* \bar{B}_h$ ▷ Chassis Articulated Bias
-

is known, since $q_{hk}(t)$ is the assumed input, while the torque τ_{hk} has to be computed together with the rest of unknown accelerations, which are the outcome of the third sweep. This leads to a *Hybrid Dynamics* problem that requires to carefully revisit the ABA's second and third steps. The approach used is inspired by [?, Section 2.3] where the notation and conventions here employed were adapted according to [?].

Starting from the leaf nodes, which in our case are represented by the wheels ($\{W\}$), after providing information such as external wrenches, twists, joint geometry, body masses, joint forces / torques for traditional connections, and time evolution of variables defining kinematically imposed joints, the backward propagation of inertia and bias can be performed.

Finally the last step, denoted as **Step 3**, starting from the root node, computes joint variable accelerations and consequently body twists derivatives. For the kinematically controlled camber joints, the torques τ_{hk} needed to maintain the desired state of motion (DCC architecture) are computed as a by-product of the local inverse dynamics. This information is crucial to prevent actuators from saturating or operating outside their bandwidth in the planned trajectories.

The root twist acceleration is determined by solving equation $W = \hat{M}\dot{V} + \hat{B}$, considering that the chassis is free-floating. The other unknown quantities, such as variable accelerations and torques, are computed projecting the Newton-Euler equations of each body along the respective Jacobian and then solving for the variable. See also [?] for more details in the derivation of the equations.

Step 6 Forward Propagation of Acceleration

- 1: $\dot{V}_b = -\hat{M}_b^{-1} \hat{B}_b$ ▷ Chassis Rigid-Body Acceleration
 - 2: **for** $h = h_1, h_2, h_3, h_4$ **do**
 - 3: $\ddot{z} = -\frac{J_{bh,z}^T (\hat{B}_h + \hat{M}_h (\text{Ad}_{g_{bh}}^{-1} \dot{V}_b - \text{ad}_{V_{bh}} V_h + \dot{J}_{bh,z} \dot{z} + \dot{J}_{bh,\delta} \dot{\delta} + J_{bh,\delta} \ddot{\delta})) - \tau}{J_{bh,z}^T \hat{M}_h J_{bh,z}}$
 - 4: $\dot{V}_{bh} = \dot{J}_{bh,z} \dot{z} + \dot{J}_{bh,\delta} \dot{\delta} + J_{bh,z} \ddot{z} + J_{bh,\delta} \ddot{\delta}$
 - 5: $\dot{V}_h = \text{Ad}_{g_{bh}}^{-1} \dot{V}_b - \text{ad}_{V_{bh}} V_h + \dot{V}_{bh}$ ▷ Knuckle 1 Rigid-Body Acceleration
 - 6: $\tau_{hk} = J_{hk}^T (\hat{B}_k + \hat{M}_k (\text{Ad}_{g_{hk}}^{-1} \dot{V}_h - \text{ad}_{V_{hk}} V_k + J_{hk} \ddot{q}_{hk}))$
 - 7: $\dot{V}_{bh} = J_{hk} \ddot{q}_{hk}$
 - 8: $\dot{V}_k = \text{Ad}_{g_{hk}}^{-1} \dot{V}_h - \text{ad}_{V_{hk}} V_k + \dot{V}_{bh}$ ▷ Knuckle 2 Rigid-Body Acceleration
 - 9: $\dot{\omega} = -\frac{J_{kw}^T (\hat{B}_w + \hat{M}_w (\dot{V}_k - \text{ad}_{V_{kw}} V_w))}{J_{kw}^T \hat{M}_w J_{kw}}$
 - 10: **end for**
-

6.4. Dynamic Model

The dynamics equations, assembled via the procedure described in Sec. 6.3, are implemented in the software code by using a state-space representation $\dot{x} = f(x, u)$. The state vector is $x = (q_{gb}; V_{gb}; z; \dot{z}; \omega; q_{hk}; \dot{q}_{hk}) \in \mathbb{R}^{32}$, with components defined in previous sections. As already mentioned, in the model the camber joint is kinematically controlled as per the DCC architecture. Hence, the associated control vector is $u = (T_a; T_b; \delta; \ddot{q}_{hk}) \in \mathbb{R}^{16}$, where each element is in \mathbb{R}^4 and collects the controls of the branch to each wheel. The torque is split into acceleration and braking components, allowing for minor modifications to model on-board motors coupled with traditional brakes and/or in-wheel motors as well. These components have their reaction torques in two different bodies in case of on-board motors and traditional brakes: the chassis and the knuckles, respectively. The δ vector collects the four translations of the steering rod attachment points, as described in Section 5.2, while for the vector \ddot{q}_{hk} no further explanation should be necessary. Clearly, this representation is valid for the model with active camber. The simpler model (no DCC) shares with the former the first 24 components of the state vector and the first 12 of the control vector.

The time derivative of the state vector has also been expressed previously except for the first element, q_{gb} . Its derivative is a function of the state and can be recovered using a standard procedure [?] as follows:

$$\dot{q}_{gb} = J_{gb}^{-1}(q_{gb}) V_{gb}, \quad (17)$$

where $J_{gb}(q_{gb}) \in \mathbb{R}^{6 \times 6}$ is the Jacobian of the virtual joint connecting the ground to the chassis, and it is computed via the PoE parameterization of the transformation g_{gb} .

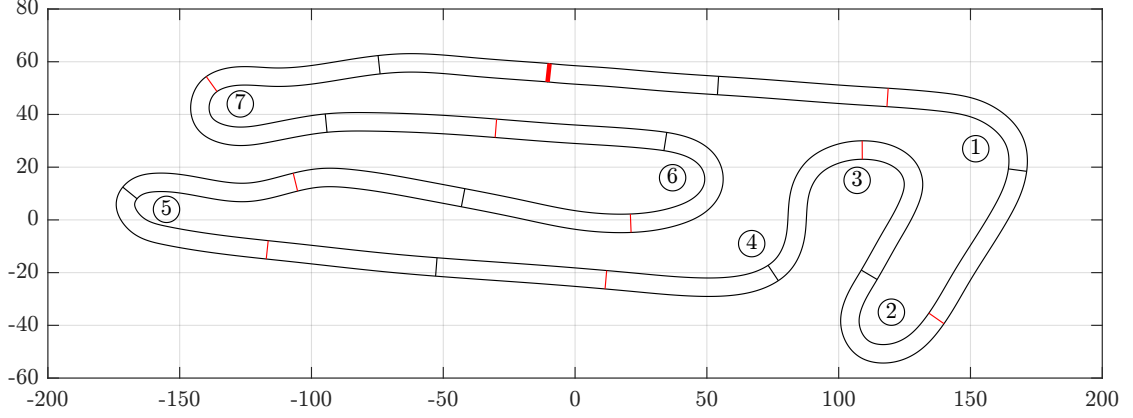


Fig. 10. Siena Circuit. The bold red line indicates the starting point, while the other red lines are equally spaced by a $\Delta\alpha = 0.1$. The black lines are also spaced like the red ones but are shifted by $\alpha_s = 0.05$. The track is run in a clockwise direction. The curves, consisting of 2 to the left and 5 to the right, are sequentially numbered based on the order of encounter.

7. Performance comparison

In this section, some of the new vehicle architectures are compared to the traditional vehicle to enlighten the different driving styles that lead to a faster or a slower lap. The track selected for the comparisons is Siena, described previously and shown in Figure 10. Siena track presents low-speed curves in which dynamic camber control (DCC) architecture or four independent steering wheels (AS) impact substantially. The track is run clockwise from the start point, represented by the bold red line in Figure 10. The thin red lines are spaced by a $\Delta\alpha = 0.1$, and the black lines, also spaced by $\Delta\alpha = 0.1$, are shifted from the red ones by $\alpha_s = 0.05$. Additionally, the plot shows the enumeration of the turns, from 1 to 7, which will be useful for further observations.

First of all, a traditional vehicle (RWD FS) is compared with an AIWD FS vehicle, capable of applying four independent torques to the wheels. As already mentioned, this feature provides the highest improvement in performance. In the analyzed case, the vehicle is able to apply not only four driving torques, each one completely independent of the others, but also four independent braking torques. This is possible if we assume the vehicle has four separate oil lines or a different braking system. For this type of vehicle, usually driven by a human, this design is never applied due to the driver's inability to manage four control inputs, such as pedal forces. The decision to maintain the four independent braking torques is made here considering that the vehicles under study are not (yet) existing models and can be thought of as new architectures envisioned for future autonomous vehicle competitions. In this case, having more inputs would not be a problem unless some limits in the real-time computing power of the ECU are encountered — a situation not discussed here. In this work, in fact, we focused on identifying and analyzing these solutions, which serve as a benchmark for maximizing vehicle performance, while intentionally postponing their implementation in a real vehicle to future designs.

7.1. All Independent Wheel Drive, Forward Steering Vehicle - AIWD FS

To perform a valid comparison between vehicles with different powertrain, they are assumed to have the same maximum (and maximum negative, i.e. braking) power. While the RWD vehicle, modeled with an open differential, will have identical torques on the driving wheels to generate power, the AIWD vehicle will optimally distribute the power among four wheels, maintaining the same total value of the RWD vehicle.

For most turns, the two vehicles follow different trajectories, with the AIWD staying significantly closer to the inside boundary of the track during turns, as it can be observed in Figure 11, third panel. See also the accompanying video [?]. The third plot, in fact, depicts the evolution of the variable $e_y(m)$, which represents the lateral distance, described in $\{S\}$ components, of the CoM with respect to the center-line, at the corresponding α . In a left turn, positive values of e_y correspond to points that are closer to the inner boundary. Vice versa, in a right turn, negative values correspond to points that are closer to the inner boundary. Each of the three plots shown in Figure 11 includes green dash-dot lines marking the turns, while the forward speed plot (first panel) additionally displays the turn numbers, with labels placed over double-ended arrows. The enumeration follows the one described in Figure 10, and each label indicates a letter that identifies the direction of the curve (L for left turns, R for right turns). Again from Figure 11, third plot, we can observe that the AIWD vehicle begins to move toward the inner boundary of the track before the RWD vehicle. This behavior is evident in turn 2, which starts at $\alpha = 0.2$, where the AIWD vehicle (blue line) reaches the inner boundary before the RWD (red line), with a difference of $\delta\alpha = 0.01$. On this track, this difference in curvilinear abscissa corresponds to 12.9 m. Consequently, as the vehicles approach turn 2, the AIWD vehicle reaches the inner boundary approximately 13 m ahead of the RWD vehicle, indicating a significantly different driving style.

Still referring to Figure 11, in particular to the first plot, which shows the forward speed u over the curvilinear abscissa α , we notice that, during cornering, the AIWD vehicle (blue line) approaches the point of minimum speed earlier than the RWD vehicle (red line). The combination of this trend, and the previously analyzed tendency to remain closer to the inner boundary while cornering, allows the AIWD vehicle to exit with higher forward speed than the RWD vehicle. This affects positively the lap time.

In the second plot of Figure 11, which reports the lateral acceleration a_y over the curvilinear abscissa α , we can notice that during cornering the AIWD vehicle (blue line) maintains a higher mean absolute value of the lateral acceleration.

In particular, traveling the chicane composed by turn 3 and 4, starting at $\alpha = 0.27$, the lateral acceleration a_y of the AIWD FS vehicle (blue line) is greater (in magnitude) than the lateral acceleration of the RWD FS vehicle (red line), except for the small range, in which $\alpha \in [0.27, 0.28]$.

Still referring to the second plot of Figure 11, and focusing on turn 4, starting approximately at $\alpha = 0.33$, the lateral acceleration a_y of the AIWD vehicle has smaller variation than the lateral acceleration of the RWD vehicle, maintaining almost a constant value during cornering, denoting a smoother behavior.

7.2. Rear Wheel Drive, All Steering Vehicle - RWD AS

We can observe in Figure 12, first plot, the road wheel angles $\delta_{i,j}$ of the four wheels with respect to the curvilinear abscissa α for the RWD AS vehicle. It is worth recalling

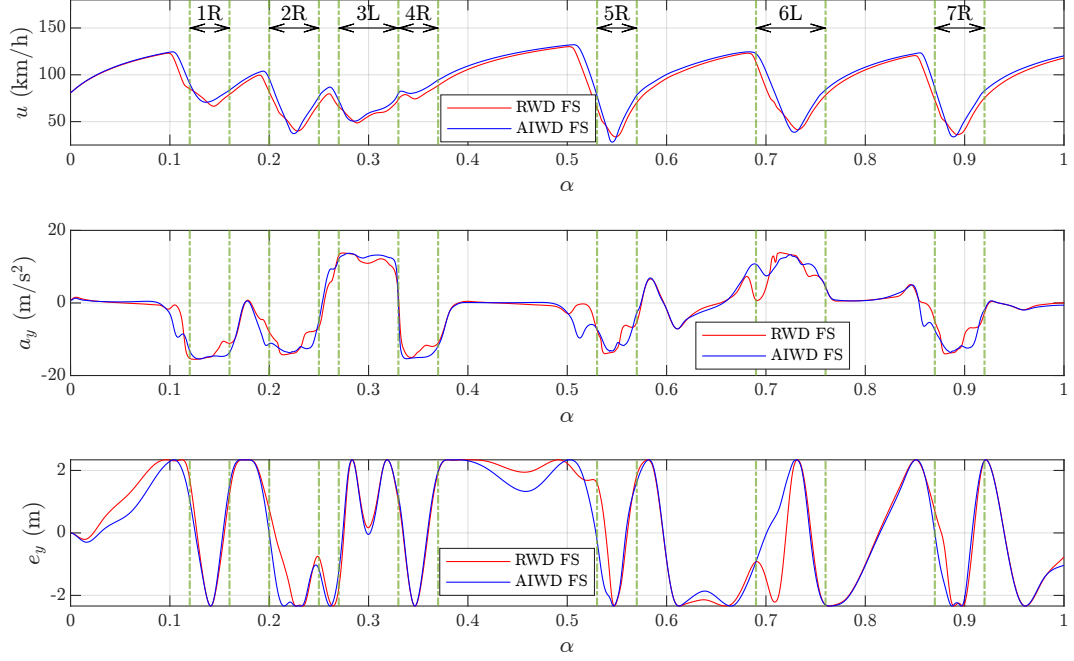


Fig. 11. Forward speed u , lateral acceleration in body components a_y and e_y evolution for RWD FS and AIWD FS vehicle, in red and blue respectively, over curvilinear abscissa α . The quantity e_y represents the lateral distance, in $\{S\}$ reference, of the CoM with respect to the center-line at the corresponding α . The turns are indicated in the forward speed plot (first panel) with the corresponding number (see Figure 10) and a letter that identifies the direction (L for left turns and R for right turns). The more positive the values, the closer the trajectory to the left edge of the track, while the more negative the values, the closer the trajectory to the right edge.

that, during cornering, this vehicle can exhibit different road wheel angle values for the wheels of the same axle.

In the first plot, the color code is such that wheels of the same axle share the color (red for the front axle, blue for the rear axle), while wheels of the same side share the same line style (solid line for the left side, dashed line for the right side). In the second plot, the forward speed u with respect to the curvilinear abscissa α is depicted. In both plots, the green vertical lines mark the start and end point of the turns. In the lower plot, the turns are labeled with a progressive number, according to Figure 10 and a letter indicating its direction (L for left-hand turns and R for right-hand turns).

Although a slight penalization is applied to the cost function to avoid discordant wheel angles on the same axle during turns, the independent behavior of the wheels allows for better control of ground forces and their direction, enhancing turning performance. This is especially noticeable in turn 5 (right turn), where the left wheels steer more than the right, and in turn 6 (left turn), where the right wheels steer more than the left. See also the accompanying video [?]. Generally, the outer wheel, which bears a greater vertical load due to lateral transfer, has a larger road wheel angle thus generating more lateral force, while the inner wheel steers less since it contributes less to the total lateral force.

Another aspect worth noting is the behavior of the rear wheels in relation to the front ones. In fact, the rear wheels steer in phase with the front wheels for most of the corners

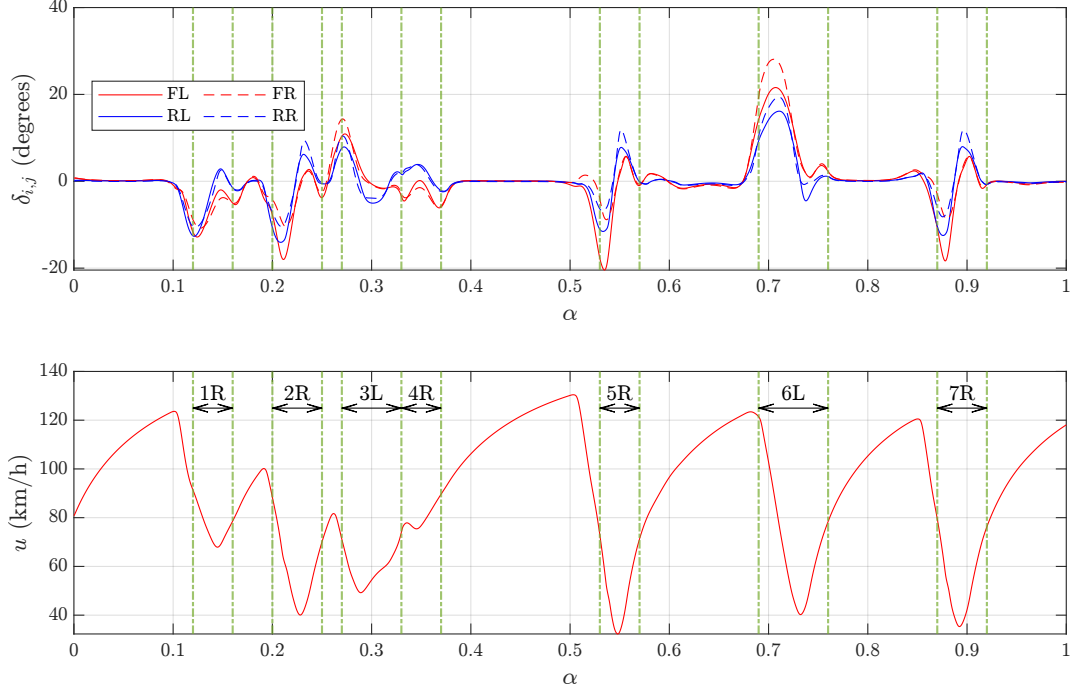


Fig. 12. Road wheel angle δ of all wheels (above), and forward speed u (below), evolution over curvilinear abscissa α . As reported in the legend, wheels of the same axle share the same color, while wheels of the same side (left/right) share the same line style. The green dash-dot lines represent the starting and ending point of turns. The turns are indicated in the forward speed plot with the corresponding number (see Figure 10) and a letter that identifies the direction (L for left turns and R for right turns).

and for most of the corner entrances, except for some cases where, in the slower part of the turn, the rear wheels steer in counterphase with the front wheels. In particular, referring again to Figure 12, it is possible to observe that in turn 2-5-7 the front and rear road wheel angles are concordant, while in the remaining turns (1-3-4-6), the road wheel angles are discordant for most part of the turn. This behavior is generally consistent with Guiggiani’s analysis of MAP’s (Map of Achievable Performance) for a Single-Track model of vehicle. In fact, in [?, Section 6.8.1] the author affirms that a desired behavior is to have a vehicle where the β angle spans a small range. This can be achieved by having in-phase rear steering at high forward speeds and counter-phase rear steering at low forward speeds.

Although a general rule like this may not hold true for every corner on the track, it nonetheless represents the desired trend to optimize performance, which is here confirmed by quantitative lapsim outputs. It also may be worthwhile to point out that a human driver can drive differently with respect to the optimizer, due to the fact that it lacks the perfect view of the track beforehand which, on the other hand, is profitably exploited by the optimizer.

7.3. Rear Wheel Drive, Forward Steering, Dynamic Camber Control - RWD FS DCC

The ability to adjust the camber of each wheel during cornering allows the vehicle to maximize the ground forces even more. In fact, in a traditional vehicle, the position and orientation of the wheels during a turn are determined by the passive motion of the suspension, in addition to the steering system for the front wheels. This means that, even if the suspension is well designed, the position of the wheels will not be always the best for performance and could be improved in each turn.

The first plot of Figure 13 shows the camber angles γ of the front right wheel (FR) over the curvilinear abscissa α for a RWD FS vehicle (red line) and for a RWD FS DCC vehicle (blue line), respectively. The second plot of Figure 13 shows the lateral force F_y of the front right wheel (FR) for the RWD FS vehicle (red solid line) and for the RWD FS DCC vehicle (blue solid line). Furthermore the maximum and minimum achievable lateral forces are shown in dashed lines and following the colors of the solid lines (RWD FS in red, RWD FS DCC in blue). Both plots show green dash-dot lines to delimit turns, while the first plot, in addition, reports labels with turn numbers (that follow the enumeration of Figure 10), and an indication of the turn direction (L for left turns, R for right turns).

It is possible to note that, during cornering, e.g. turn 3, the camber tends to settle around a value of -7.5 degrees which contributes to increasing the lateral force by 128 N. This corresponds to a 7.6% increase w.r.t. the baseline, which can be considered a significant achievement (see the zoomed area in the second plot at turn 3, where $\alpha \in [0.26, 0.33]$). See also the accompanying video [?].

Similarly, in turn 6, starting at $\alpha = 0.69$, it can be observed that the maximum lateral force F_y achievable is higher for the RWD FS DCC vehicle, implying a better wheel positioning relative to the road. We remind that, according to the convention of positive lateral forces F_y pointing to the left of the vehicle, “the higher (more positive) the better” is valid if we analyze a left turn, such as turn 3 or 6. Otherwise, in a right turn, an improvement on force limits would correspond to a lower minimum for achievable lateral force. For this reason, considering here the front right (FR) wheel, we focus only on left turns.

For brevity, we only mention that the RWD AS DCC vehicle combines the advantages of the previous two, thereby guaranteeing the highest performance. Similarly to the RWD AS vehicle, it steers more with the outer wheels thus exploiting the lateral load transfers; at the same time, as registered for the RWD FS DCC vehicle, it adjusts the camber during cornering to settle around the value of maximum ground force available.

8. AIWD 0S DCC: in depth analysis

This section aims to showcase that on a track with characteristics like the Nürburgring section used or Siena, whose results we present here, a vehicle able of applying four independent acceleration torques (AIWD) and dynamically controlling the camber (DCC) has a very significant advantage in terms of lap time and cornering performance.

To stress the power of these features, a vehicle with such characteristics but *without steering abilities* (0S, zero steering) has been considered. See also the accompanying video [?]. Cornering is still possible if the vehicles applies different torques to the wheels on the same axle, as we can observe in Figure 14. Here, the evolution of the

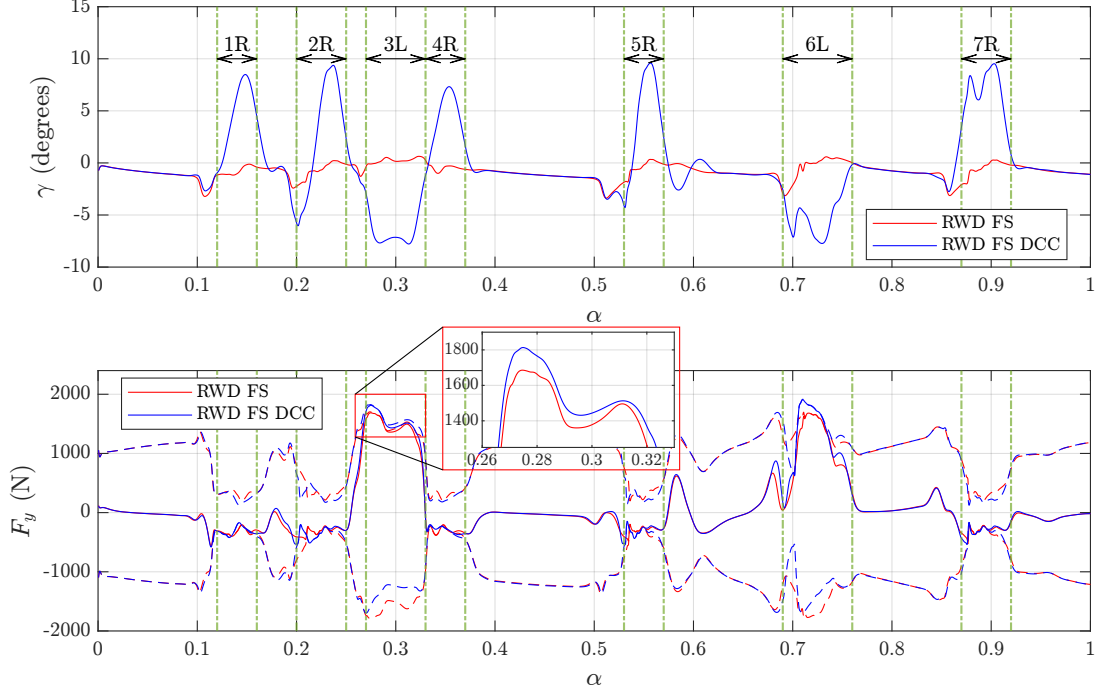


Fig. 13. Camber angle γ (first plot), and lateral force F_y (second plot) over curvilinear abscissa α , for the front right wheel (FR) of a RWD FS (red line) and a RWD FS DCC (blue line) vehicle. The dashed lines represent the upper and lower limit of the forces for each vehicle, coded by the same color. The green dash-dot lines represent the starting and ending point of turns. The turns are indicated in the forward speed plot with the number and the letter that identifies the direction (L for left turns and R for right turns). The zoomed area reports the section with $\alpha \in [0.26, 0.33]$ to highlight the improvement of $\simeq 130$ N in the lateral force achievable by Dynamic Camber Control (DCC).

torques applied to the wheels $T_{i,j}$ over the curvilinear abscissa α for the front axle (upper plot) and the rear axle (lower plot) is illustrated. The left wheels (FL and RL) are plotted with red lines, while the right wheels (FR and RR) are plotted with blue lines. Similarly to the previous figures, the green dash-dot lines in both plots delimit the turns, while the labels in the first plot denote the turn number, according to Figure 10, and a letter which discriminates the direction (L for left-hand turns, R for right-hand turns).

During turns 2-5-6-7, the torque of the inner wheel settles to zero, as is possible to observe in the first panel of Figure 14. Here, the inner wheel starts a free rolling motion, with no applied torque by the motor.

Otherwise a different behavior emerges in turn 4, starting around $\alpha = 0.33$. Here, the two wheels experience torques that are approximately mirror-symmetric with respect to the horizontal line $T = 100$ Nm, with the inner torque (FR, blue line) that assumes negative values closely to $\alpha = 0.33$, without losing contact to the ground.

While the AIWD FS DCC vehicle completes a lap in 51.40 seconds, the AIWD OS DCC vehicle does so in 51.64 seconds – only 0.24 seconds slower but 3.66 seconds faster than a traditional RWD FS vehicle. This is achieved by properly adjusting camber and applying four independent torques, thus compensating for the lack of steering.

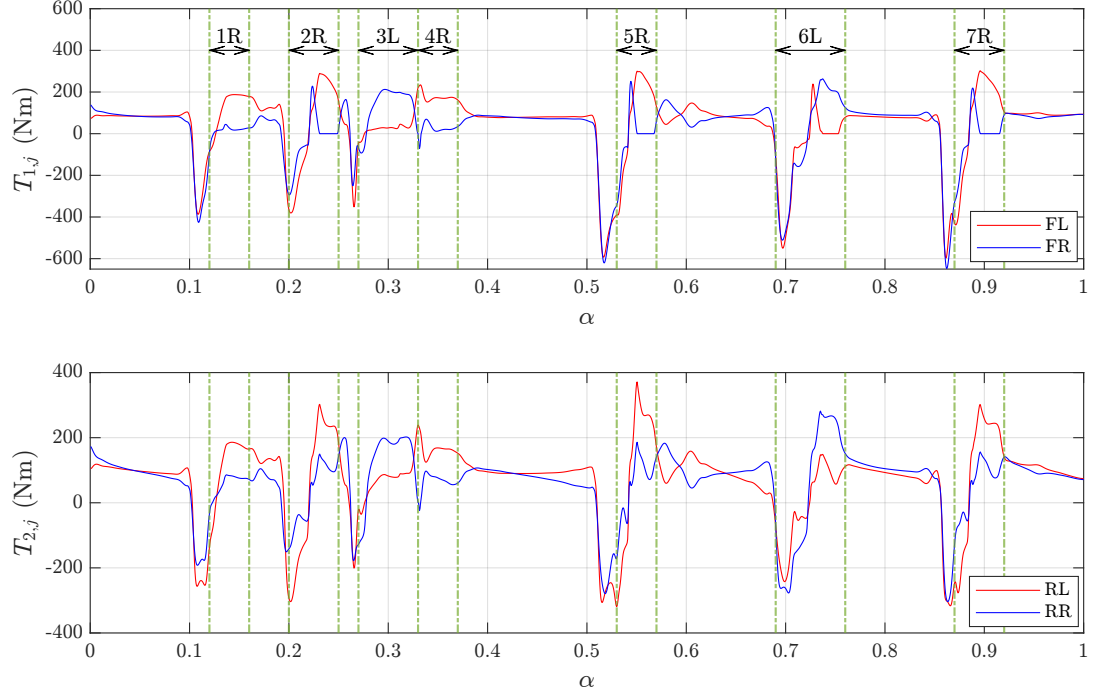


Fig. 14. Wheel torques $T_{i,j}$ evolution over curvilinear abscissa α , for the front wheels (above) and for the rear wheels (below) of a AIWD 0S DCC vehicle (0S indicates incapability of steering). The red lines refer to the left side while the blue lines refer to the right side. As for the previous plots, the green dash-dot lines indicates the turns, labeled in the first plot by the number and a letter L, for left turns, or R, for right turns.

The torque vectoring logic, used in both AIWD FS DCC and AIWD 0S DCC, plays a key role in optimizing turning maneuvers. By simply adjusting the intervention to avoid steering, the lap is completed with a 0.47% increase in lap time compared to the AIWD FS DCC. It is worth mentioning in passing that the model implements power limits as total values, neglecting potential motor saturation, and may lead to an overestimation of performance. Therefore, this analysis should not be taken as a direct guideline for designing a AIWD 0S DCC vehicle but an informative case study highlighting factors relevant to cornering and steering.

9. Conclusions

In this work we investigated the potential of some new technologies in empowering next generation autonomous race vehicles. In particular, the upgrades considered here were: four independent steering wheels (All Steering - AS), four independent wheel torques (All Independent Wheel Drive - AIWD), and independent control of the four camber angles (Dynamic Camber Control - DCC).

Firstly, we discussed the impact of enabling these control inputs on the vehicle performance, in relation to the mass increase associated to the new components needed for the systems. The results confirmed that the vehicle endowed with all the features (All Independent Wheel Drive, All Steering, Dynamic Camber Control - AIWD AS

DCC) is the most performant and forgiving to added mass. Compared to a traditional vehicle (Rear Wheel Drive, Forward Steering - RWD FS), its advantage in lap time persists until 123.5 kg of added mass.

Then, lap times are presented for the baseline (Rear Wheel Drive, Forward Steering - RWD FS), and for vehicles with single upgrades added, keeping constant the total mass. In particular, the configuration analyzed are: Rear Wheel Drive, All Steering - RWD AS, All Independent Wheel Drive, Forward Steering - AIWD FS, and Rear Wheel Drive, Forward Steering, Dynamic Camber Control - RWD FS DCC. From these results it is evident that the single feature that improves the performance the most is the AIWD FS in Siena circuit, and the RWD FS DCC in Nürburgring circuit.

Focusing on trajectories, we can highlight the following features. The All Independent Wheel Drive, Forward Steering (AIWD FS) vehicle, being able to perform torque vectoring, travels turns closer to the inner boundary of the track, compared to a traditional vehicle, and exits with an higher forward speed.

A Rear Wheel Drive, All Steering (RWD AS) vehicle experiences in-phase rear steering for high speed turns, and counter-phase rear steering for low speed turns, hence improving cornering in each condition.

Being able to control the camber dynamically, a Rear Wheel Drive Vehicle, Forward Steering, Dynamic Camber Control (RWD FS DCC) vehicle, positions each wheel w.r.t. the road to raise the force limits, thus maximizing the ground forces.

To highlight the impact of the torque vectoring and camber control, a vehicle with these characteristics but *without steering abilities* (All Independent Wheel Drive, Zero Steering, Dynamic Camber Control - AIWD 0S DCC) has been analyzed. This vehicle completes a lap in Siena 3.66 seconds faster than a traditional vehicle (RWD FS), and corners effectively by applying higher torques on outer wheels than the inner wheels.

Introducing a temperature-dependent tyre model, we observed that the least affected vehicle is the traditional RWD FS, which stresses the wheels the least and thus loses only 11 ms due to the drop in grip factor. In contrast, All Independent Wheel Drive (AIWD) vehicles tend to stress all four wheels much more. As a result, when the tire thermal model in the MLTP takes into account the reduction in grip with increasing temperature, AIWD vehicles are penalized more than RWD FS vehicles, losing approximately 100 ms each in the considered sector.

10. Future Work

While this study provides a broad overview of the possibilities and implications of introducing certain features to a new generation of autonomous vehicles, some aspects require further investigation.

A more complete tire model, which could account for tread wear and the consequent inevitable performance drop due to tire usage, should be implemented. To achieve this, either a testing campaign on the actual modeled tires to fine-tune the parameters of the wear-dependent phenomena, or a new dataset of fully characterized tires, is essential.

The introduction of realistic aerodynamic maps could indeed lead to a refinement of the model, enhancing the accuracy of the vehicle dynamics representation.

Modeling a non-developable 3D track that incorporates bumps and holes would represent a significant advancement in fully understanding the potential and limitations of these features. This approach would enhance the physical realism of the road-vehicle interaction, providing deeper insights into its dynamics.

Additionally, future developments could include an evaluation of design solutions for

the various subsystems that control the additional degrees of freedom. These solutions must, however, take into account their practical feasibility for real-world systems, considering current technological limitations in sensors and actuators, as well as the challenges posed by their integration.

Acknowledgement(s)

The authors would like to thank Dr. Lorenzo Bartali for his significant contributions to the initial version of the optimization framework developed and utilized in this paper.

Disclosure statement

No potential conflict of interest was reported by the author(s).

Funding

This work is supported by project PRIN 2022 PNRR “Global Stability of road vehicle motion - STAVE” CUP I53D23005670001.