Fast zonotope-tube-based LPV-MPC for autonomous vehicles

Eugenio Alcalá1, Vicenç Puig2, Joseba Quevedo1, Olivier Sename2

1Department of Automatic Control, Polytechnic University of Catalonia, Barcelona, Spain
2Control System Department at GIPSA-Lab, Grenoble, France

Abstract: In this study, the authors present an effective online tube-based model predictive control (T-MPC) solution for autonomous driving that aims at improving the computational load while ensuring robust stability and performance in fast and disturbed scenarios. They focus on reformulating the non-linear original problem into a pseudo-linear problem by transforming the non-linear vehicle equations to be expressed in a linear parameter varying (LPV) form. An scheme composed by a nominal controller and a corrective local controller is proposed. First, the local controller is designed as a polytopic LPV-H$_\infty$ controller able to reject external disturbances. Moreover, a finite number of accurate reachable sets, also called tube, are computed online using zonotopes taking into account the system dynamics, the local controller and the disturbance-uncertainty bounds considered. Second, the nominal controller is designed as an MPC where the LPV vehicle model is used to speed up the computational time while keeping accurate vehicle representation. They test the presented scheme and compared the local controller performance against the LQR design as state-of-the-art approach. They demonstrate its effectiveness in a disturbed fast driving scenario being able to reject strong exogenous disturbances and fulfilling imposed constraints at a very reduced computational cost.

1 Introduction

In last recent years, the number of vehicles on the roads has grown significantly and subsequently the risk of car accidents. In a near future, when autonomous vehicles are finally in the streets, we will expect them to handle the most challenging situations that humans handle nowadays. They will have to deal with the complete net of transportation, i.e. vehicles, traffic rules, pedestrians etc., but also with extreme weather situations. Most of these cases can be either forecasted or approximated by rules since they follow known physical behaviours in the weather case or traffic rules in the case of vehicles and pedestrians. However, sometimes this may not happen as expected and the vehicle is suddenly running into extreme situations such as for example very windy situation on highways. Addressing these situations is what control engineering refers to as robustness: the ability of the controller to handle unexpected situations such as internal variations in the system or in the external environment affecting the system.

A large variety of control strategies have been studied to address the robustness in control of systems. All these methods pursue the same objective: ensure asymptotic stability, robustness and performance [1, 2].

Model predictive control (MPC) is an effective control strategy that allows to deal with constrained problems and multiple-input multiple-output systems. However, dealing with uncertainty or disturbances is something that conventional MPC algorithms do not handle and then, robust MPC (RMPC) formulations have to be considered where the design is done by means of robustifying the constraints [3]. In [4], the author presents a review on current MPC formulations with their limitations and future development directions.

During the last years, two differentiated and consolidated approaches for robust MPC have been addressed: Min-max MPC and Tube-based MPC (T-MPC). On the one hand, the min-max or worst-case problem aims to find the optimal solution based on minimising the maximum value of the cost function. In [5], the authors present a robust self-triggered min-max MPC approach for constrained non-linear systems with both parameter uncertainties and disturbances. On the other hand, T-MPC is based on computing a region around the nominal prediction that ensures the state of the system to remain inside under any possible uncertainty and disturbance [6].

Our work is mostly inspired by the T-MPC technique. This strategy has been widely employed in the mobile robotics field [7–10]. However, from a self-driving car perspective we do not find many references in the literature. In Fig. 1, we show a diagram made for classifying the T-MPC technique applied to autonomous driving as function of some involved design characteristics. Furthermore, we present in Table 1 a review of those works dealing with T-MPC in autonomous driving of cars considering the properties presented in Fig. 1.

In this paper, we present a robust T-MPC approach faster than the state-of-the-art strategies being able to reject large exogenous disturbances. This optimal algorithm uses a linear parameter varying (LPV) vehicle model for simulating future behaviour. The principal concept behind the LPV modelling approach is that the non-linear model representation can be expressed as a combination of linear models that depend on some scheduling variables without using linearisation [16]. Furthermore, the introduction of zonotope-based operations to compute reachable sets allow to make the algorithm faster and more accurate [17].

We summarise the innovative points with respect to the state-of-the-art as follows:

• Using zonotopy theory, we are able to reduce the computational cost of basic operations, i.e. Minkowsky sum and difference, in comparison with standard polytopes-based operations.
• The use of zonotopy-based calculations allows to bound more tightly the tube, hence obtaining a less conservative result and more accurate result.
• Using $H_\infty$ control design to obtain a gain scheduling polytopic LPV local controller allows to reject large exogenous disturbances acting over the vehicle. Current T-MPC techniques in the state-of-the-art are using LQR technique.
• Currently, most of the works based on a robust MPC design use a local controller than runs at the same frequency than the nominal
controller (MPC). In this work, we propose a faster loop to achieve a faster and better performance of the control scheme.

The paper is structured as follows: Section 2 presents the problem statement. Section 3 addresses the core of this work: the online T-LPV-MPC using zonotopic algorithm. In Section 4, we present the main results and a proper discussion. Finally, last section presents the conclusions of the work.

2 General problem statement

This paper addresses the problem of designing an online T-MPC for controlling an autonomous vehicle (see Fig. 2) formulated as the following non-linear system:

$$ x^+ = f(x,u) + w, $$

(1)

where $x \in \mathbb{R}^n$ is the state vector, $u \in \mathbb{R}^m$ the input vector and $f(x,u)$ represents the non-linear map obtained after modelling the physics of the real system. Vector $w \in \mathbb{R}^n$ contains all the unmodelled physics of the real plant and exogenous disturbances acting over it.

Note that in this paper the notation $x^+$ is used for the successor of vector $x$, i.e. $x = x(k)$ and $x^+ = x(k+1)$.

At this point, the following uncertain, LPV, discrete-time system is formulated as:

$$ x^+ = A_\xi x + B_\eta u + w, $$

(2)

where $A_\xi \in \mathbb{R}^{n \times n}$ and $B_\eta \in \mathbb{R}^{n \times m}$ are the LPV state matrices which depend on the varying scheduling vector $\zeta \in \mathbb{R}^{n_\zeta}$, being $n_\zeta$ the number of scheduling variables.

**Remark 1:** The system $x^+ = f(x,u)$ in (1) can be represented in the form $x^+ = A(x) + B(u)$ in (2) without linearisation by embedding the non-linearities in the varying parameters $\zeta$ using the non-linear embedding approach [18].

The state, control and disturbance vectors are bounded as

$$ x \in X, \quad u \in U, \quad w \in W, $$

(3)

where $X \subseteq \mathbb{R}^n$, $U \subseteq \mathbb{R}^m$ and $W \subseteq \mathbb{R}^n$. The set $W$ has been generated in simulation taking into account the effect of worst-case disturbances (friction and wind) considered.

To achieve the tracking and robust control purposes, a two-layer control scheme is considered (see Fig. 2):

- **Reference tracking control problem.** The LPV-MPC strategy deals with the following disturbance-free system for tracking the dynamic references while handling system constraints

$$ \dot{x}^+ = A_\xi \dot{x} + B_\eta \bar{u}, $$

(4)

where the state ($\dot{x} \in \mathbb{R}^n$) and optimal control ($\bar{u} \in \mathbb{R}^m$) vectors are bounded as

$$ \dot{x} \in \bar{X}, \quad \bar{u} \in \bar{U}, $$

(5)

where $\bar{X} \subseteq \mathbb{R}^n$ and $\bar{U} \subseteq \mathbb{R}^m$. System (4) will be referred as nominal model throughout the work.

- **Robust control problem.** The main idea is to compensate the mismatch between the states of (1) and the nominal state vectors (4). This difference is computed as

$$ e = x - \dot{x}, $$

(6)

where $e$ is the error state. In order to minimise such a mismatch, the following control law is considered:

---

Table 1: Classification of T-MPC technique in autonomous driving field. Some used acronyms: SF := state feedback, LQR := linear quadratic regulator, LTI := linear time invariant, RPI := robust positively invariant, GS := gain scheduling

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Control problem</td>
<td>cruise control</td>
<td>steering control</td>
<td>steering control</td>
</tr>
<tr>
<td>type of model</td>
<td>long. dynamics</td>
<td>lateral dynamics</td>
<td>lateral. dynamics</td>
</tr>
<tr>
<td>type of system</td>
<td>LTI</td>
<td>LPV</td>
<td>LTI</td>
</tr>
<tr>
<td>local control / design</td>
<td>LQR</td>
<td>polytopic</td>
<td>LQR</td>
</tr>
<tr>
<td>local control / implement.</td>
<td>static SF gain</td>
<td>static SF gain</td>
<td>static SF gain</td>
</tr>
<tr>
<td>tube computation</td>
<td>polytopic</td>
<td>minimal RPI</td>
<td>online terminal RPI</td>
</tr>
<tr>
<td>computational time / horizon</td>
<td>no info</td>
<td>25 ms / 7 steps</td>
<td>100 ms / 6 steps</td>
</tr>
</tbody>
</table>

---

Fig. 1: Diagram of different characteristics involved in the design of T-MPC technique for self-driving vehicles.

Fig. 2: Robust control scheme composed of the proposed controller (ZT-LPV-MPC) and a local corrective controller (LPV-$K^\infty_\rho$)

---

IET Control Theory Appl., 2020, Vol. 14 Iss. 20, pp. 3676-3685
© The Institution of Engineering and Technology 2020
\[ u = \dot{x} + u_m \]
\[ u_m = K_\varepsilon^o v, \]
\[ \dot{v}_m \in \mathbb{R}^n \]

where \( u_m \in \mathbb{R}^n \) is the corrective action and \( K_\varepsilon^o \in \mathbb{R}^{n \times n} \) is the state feedback gain computed online as a gain scheduling controller using the \( H_\infty \)-LMI-based problem for the design.

Finally, the closed-loop error dynamics are defined as
\[ e^r = x^r - \dot{x}^r = (A_\varepsilon + B_\varepsilon K_\varepsilon^o) e + w. \]

### 3 Vehicle modelling

An LPV system is a dynamical system of finite dimension whose state-space matrices are fixed functions of a vector of measurable scheduling variables. Obtaining the LPV formulation of a non-linear system may be sometimes a non-trivial task. Particularly, trying to obtain the LPV representation presented in (2) may result on many different options and not all of them with the same quality representation.

The non-linear equations considered in this work defining the behaviour of the vehicle are
\[ v_f = a_f + \frac{F_f f \sin \delta - F_{df} f}{m} \]
\[ v_r = \frac{F_r f \cos \delta + F_{dr} f}{m} - a v_i \]
\[ \dot{\omega} = \frac{F_f f \cos \delta - F_{dr} f}{I} \]
\[ x_p = v_i \]
\[ \theta = \omega \]
\[ a_f = \delta - \frac{F_f f \sin \delta}{v_i} + \left( \frac{1}{2} \rho C_d f v_i^2 \right) \]
\[ a_r = \frac{v_i}{v_f} \frac{F_{df} f}{v_s} \]
\[ F_{f f} = C_f(\alpha_f) a_f \]
\[ F_{r r} = C_r(\alpha_r) a_r \]
\[ F_{df} = \frac{\mu mg + \frac{1}{2} \rho C_d f v_i^2}{m} \]

where the dynamic vehicle variables \( v, v_i, \) and \( \omega \) represent the body frame velocities, i.e. linear in \( x \), linear in \( y \) and angular velocities, respectively. Variables \( x_p \) and \( \theta \) are the integral with respect of time of \( v_i \) and \( \omega \), respectively. The control variables \( \delta \) and \( a \) are the steering angle at the front wheels and the longitudinal acceleration vector on the rear wheels, respectively, \( F_f \) and \( F_r \) are the lateral forces produced in front and rear tires, respectively. Both \( C_f(\alpha_f) \) and \( C_r(\alpha_r) \) represent the linear and non-linear function of the front and rear stiffness coefficient, respectively. Front and rear slip angles are represented as \( \alpha_f \) and \( \alpha_r \), respectively. \( m \) and \( I \) represent the vehicle mass and inertia and \( l_f \) and \( l_r \) are the distances from the vehicle centre of mass to the front and rear wheel axes, respectively. \( \mu, \rho \) and \( g \) are the friction coefficient, the air density and the gravity values, respectively. \( C_{dA} \) is the product of drag coefficient and vehicle frontal cross-sectional area. All the dynamic vehicle parameters are properly defined in Table 2 and shown in Fig. 3.

Then, denoting the state and control vectors, respectively, as
\[ x = \begin{bmatrix} v_f \\ v_i \\ v_s \\ x_p \\ \theta \end{bmatrix}, \]
\[ u = \begin{bmatrix} \delta \\ a \end{bmatrix}, \]
\[ A_\varepsilon = \begin{bmatrix} A_{11} & A_{12} & A_{13} & 0 & 0 \\ 0 & A_{22} & A_{23} & 0 & 0 \\ 0 & A_{32} & A_{33} & 0 & 0 \\ -T_s & 0 & 0 & 1 & 0 \\ 0 & 0 & -T_s & 0 & 1 \end{bmatrix}, \]
\[ B_\varepsilon = \begin{bmatrix} s \frac{1}{m} \sin \delta C_f T_s & T_s \\ \frac{1}{m} \cos \delta C_f T_s & 0 \\ \frac{1}{T_s} \cos \delta C_r T_r & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \]

being

---

**Table 2 Dynamic model parameters of the driverless UPC Car**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>( l_f )</td>
<td>0.902 m</td>
<td>( l_r )</td>
<td>0.638 m</td>
</tr>
<tr>
<td>( m )</td>
<td>196 kg</td>
<td>( I )</td>
<td>93 kg m^\text{2}</td>
</tr>
<tr>
<td>( d_f )</td>
<td>8.255</td>
<td>( c_f )</td>
<td>1.6</td>
</tr>
<tr>
<td>( b_f )</td>
<td>6.1</td>
<td>( \mu )</td>
<td>1.4</td>
</tr>
<tr>
<td>( d_r )</td>
<td>8.255</td>
<td>( c_r )</td>
<td>1.6</td>
</tr>
<tr>
<td>( b_r )</td>
<td>6.1</td>
<td>( \rho )</td>
<td>1.225 kg m^{-3}</td>
</tr>
<tr>
<td>( C_{dA} )</td>
<td>1.64 g</td>
<td>( g )</td>
<td>9.81 m/s^{-2}</td>
</tr>
<tr>
<td>( C_{dA} )</td>
<td>1.82 w</td>
<td></td>
<td>1.45 m</td>
</tr>
<tr>
<td>( d )</td>
<td>2.3 m</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
\[
A_1 = 1 + (\frac{\mu}{v_s} - \frac{\rho C_d v r}{2 m}) T_s \\
A_2 = \frac{C_s \sin \delta}{m v_s} T_s \\
A_3 = \left(\frac{C_f \sin \delta}{m v_s^3} + v_s\right) T_s \\
A_4 = 1 + \left(\frac{C_t + C_f \cos \delta}{m v_s}\right) T_s \\
A_5 = \left(-\frac{C_f \sin \delta - C_i}{m v_s} - v_s\right) T_s \\
A_6 = \left(-\frac{C_f \sin \delta - l_i C_t}{v_s} \right) T_s \\
A_7 = 1 + \left(-\frac{C_r \sin \delta + \bar{C}_r}{v_s}\right) T_s \\
\]

where \(T_s\) is the sampling time for discretising the continuous model. The discrete-time LPV model (4) is obtained using the Euler discretisation approach. Note that, for a easier comprehension \(C_s(x)\) is denoted by \(C_s\) being \(i = f, r\).

### 4 Online T-LPV-MPC using zonotopes

In this section, we present the zonotope-tube-based LPV-MPC (ZT-LPV-MPC) scheme to significantly reduce the computational effort in RMPC techniques for autonomous driving (see Fig. 2). The main purpose of this strategy is to achieve robust stability in the presence of modeling errors and bounded exogenous disturbances.

In the most recent literature, the linear quadratic regulator (LQR) control strategy is explained step-by-step for a correct comprehension. First of all, a controller to reduce the mismatch between the states of system (1) and the nominal state vectors (4) even under the presence of disturbances, the \(C_\rho\) ranges now over a fixed polytope \(\Theta\) and the number of vertexes of polytope \(\Theta\) is equal to \(2^n\); \(\mu(\zeta)\) is the linear membership function defined by

\[
\mu(\zeta) = \prod_{j=1}^{n} \xi_j(\eta_j^i, \eta_j^r), \quad \forall i = 1, \ldots, N, \quad (13)
\]

with

\[
\eta_j^i = \frac{\bar{c}_j - \zeta_j(k)}{\xi_j - \bar{c}_j}, \quad \eta_j^r = 1 - \eta_j^i, \quad (14)
\]

where each scheduling variable \(\zeta_j\) is known and varies in a defined interval \(\zeta_j \in [\bar{c}_j, \xi_j] \subseteq \Theta\) and \(\xi_j(\cdot)\) corresponds with the function that performs the \(N = 2^n\) possible combinations. In addition, next conditions must be satisfied

\[
\sum_{i=1}^{N} \mu(\zeta) = 1, \quad \mu(\zeta) \geq 0, \quad \forall \zeta \in \Theta. \quad (15)
\]

Matrix \(B\) in (11) is an instantiation of \(B_c\) in (10d) at \(\delta = 0\) and \(C_f\) at a particular constant value. \(E\) is the disturbance input matrix, \(d\) represents the exogenous disturbance vector and its product \(Ed\) is always contained in \(W_z\). \(z\) represents the controlled variables vector and \(C, D, \mu, \delta\) and \(D\) are tuning matrices of appropriate dimensions.

From the polytopic LPV system (11) and considering the state feedback control law \(u = K_{\pi}^\pi x\), we can formulate the transfer function from \(d\) to \(z\) as

\[
G_{zd} = (C + D_i K_{\pi})(J - (A_{\pi} + B K_{\pi}^\pi))^T E + D_z. \quad (16)
\]

Hence the proposed problem consists on finding a polytopic state feedback gain \(K_{\pi}\) such that

\[
\|G_{zd}\|_\infty \leq \gamma. \quad (17)
\]

holds for the attenuation scalar \(\gamma \in \mathbb{R}\). To find the solution, we solve the \(H_{\infty}\) problem in discrete time via LMIs using the polytopic approach as suggested in [20] given by

\[
\begin{aligned}
\min_{x, w_i} \\
\text{s.t.} \\
x &= Xs E 0 \\
* X 0 X C^T - W_i D_i^T \\
* * \gamma I D_i^T > 0 \\
* * * \gamma I X > 0 \\
S &= A_X - BW_i \\
\forall j = 1, \ldots, N
\end{aligned} \quad (18)
\]

being the solutions \(x = P^{-1} \in \mathbb{R}^{n_x \times n_x}\) and \(w_i = K P^{-1} \in \mathbb{R}^{m_x \times n_x}\), where \(P \in \mathbb{R}^{n_x \times n_x} > 0\) represents the common Lyapunov matrix for the
polytopic LPV system. Then, the resulting vertices of the new polytopic controller are obtained by \( K_i = W_iX^{-1} \in \mathbb{R}^{m \times n} \) and the \( H_\infty \) norm of \( G_{cd} \) is \( \gamma \).

Remark 2: In \( H_\infty \) control, we have even more degrees of freedom to include additional performance weights and better attenuate unknown inputs (disturbance and noise).

### 4.1.2 Online computation

At each control iteration \( k \), the state feedback LPV control gain \( K_\infty^{(k)} \) is updated based on the current value of the scheduling vector \( \zeta \). To do so, a linear combination of the vertices of the polytopic controller, i.e. the set of \( K_\infty \) is computed as

\[
K_\infty^{(k)} = \sum_{i=1}^{N} \nu(\zeta)K_i.
\]  

(19)

### 4.2 Terminal robust invariant set and cost

A commonly used approach to guarantee asymptotic stability of deterministic robust MPC consists in incorporating both a terminal cost, \( P \), and a terminal constraint set, \( \chi_T \). In this section, we propose an offline method to compute both \( P \) and \( \chi_T \). Thus, the closed-loop system convergence to the origin is ensured if

- \( Q = Q^T \geq 0 \), \( R = R^T > 0 \) and \( P > 0 \).
- The sets \( X, \chi_T \) and \( U \) are zonotopes containing the origin.
- The terminal cost is a Lyapunov function in \( \chi_T \).
- \( \chi_T \) is the minimal robust positively invariant (mRPI) set, \( \chi_T \subseteq X \).

On the one hand, the computation of \( P \) is carried out by solving the LMI-based \( H_\infty \) problem (18). Furthermore, a polytopic robust controller is found. The optimal problem solutions, i.e. \( X \) and \( W_p \), are used to calculate the controllers at the vertices of the polytope as \( K_i = W_iX^{-1} \). Note that the Lyapunov function in the optimisation problem is found to be equal to \( X^{-1} \) and will be used later in (33) as \( P \).

On the other hand, the terminal set \( \chi_T \) will be the mRPI set if and only if it is contained in any closed-RPI set and is convex and unique. Then, the mRPI set for the stable and disturbed system (8) is computed by the following recursive procedure:

1. Initialisation: \( \Omega_0 = \Omega_0 \).
2. Loop: \( \Omega_{k+1} = \mathcal{A}(\Omega_k) \oplus W \). \( \Omega_{k+1} = \mathcal{A}(\Omega_k) \oplus W \). \( k \) = 0 \( \Omega_{k+1} = \mathcal{A}(\Omega_k) \oplus W \).
3. Termination condition: \( \text{stop when } \Omega_{k+1} = \Omega_k \). \( \chi_T = \Omega_{k+1} \).

where \( \mathcal{A}(\cdot) \) is the set mapping defined as

\[
\mathcal{A}(\Omega_k) = \text{Conv}\left\{ \sum_{i=1}^N (A_i + BK_i^p)\Omega_k \right\}.
\]  

(21)

Note that, \( \text{Conv}(\cdot) \) represents the convex hull and is used to compute the one-step reachable set for the polytopic system case. This allows to preserve the convexity of the resulting set within the recursive iterations. However, this recursive approximation to compute the mRPI set is intractable and not realistic since we may need infinite iterations to reach the termination condition. For that reason, in [21], the authors propose an outer approximation method for computing the mRPI set with a given precision. This approach consists on replacing the termination condition in (20) by the condition of terminating when there exist a \( k^* \) iteration such that

\[
\mathcal{A}^k(\Omega_k) \subseteq \mathcal{A}^*_{\infty}(\cdot),
\]  

(22)

where \( \mathcal{A}^k_{\infty}(\cdot) = \{ x \in \mathbb{R}^n : \| x \|_p \leq \epsilon \} \) defines a ball of arbitrary small size. Therefore, in such reference, it is concluded that the set \( \Omega_k \) is an outer approximation of the mRPI set \( \Omega_k \) with the given precision \( \mathcal{A}^k_{\infty}(\cdot) \) as well as an RPI set too.

In addition, the initialisation condition in (20) is still not defined. To find \( E_{\infty} \), which is an RPI set for the system (8), it is necessary to solve the following iterative algorithm where there exist a finite \( k^* \) such that the termination condition is reached

1. Loop:
   \[
   \mathcal{A}(E_k) = \text{Conv}\left\{ \sum_{i=1}^N (A_i + BK_i^p)E_k \right\}.
   \]  

(23)

2. Termination condition:
   - \( \text{stop when } \mathcal{A}(E_{k+1}) = \mathcal{A}(E_k) \).

Furthermore, given the stabilised system (8), the initial convex set \( E_0 \supseteq \Omega_0 \) can be computed as

\[
E_0 = \bigoplus_{i=0}^{k^*-1} \mathcal{A}(B(r)) = \bigoplus_{i=0}^{k^*-1} \bigoplus_{i=0}^{k^*-1} \frac{\rho^i}{\xi_i} B(r).
\]  

(24)

where \( \xi \in (0, 1) \), \( \rho^i \in \mathbb{N} \) and \( B(r) = \{ x \in \mathbb{R}^n : \| x \|_\infty \leq r \} \) is a box containing \( W \). Note that, we should find a proper \( E_0 \) such that \( \mathcal{A}(B(r)) \subseteq B(r) \) holds for \( k \geq \rho^i \).

### 4.3 Online reachable sets

This section addresses the reachable sets calculation also known as the one-step forward-reachable set computation using zonotopic-based representation.

A zonotope, represented as \( \langle c_w, R_w \rangle \) with the centre \( c_w \in \mathbb{R}^n \) and the generator matrix \( R_w \in \mathbb{R}^{m \times p} \), is a particular form of a polytope defined as the linear image of the unit cube [22]

\[
W = \langle c_w, R_w \rangle = \{ c_w + R_w x : \| x \|_\infty \leq 1 \}.
\]  

(25)

Note that, the linear image of a zonotope \( W = \langle c_w, R_w \rangle \) by a compatible matrix \( M \) is defined as

\[
M \cdot W = \langle Mc_w, MR_w \rangle.
\]  

(26)

Along this work, zonotopes are treated as centred zonotopes denoted by \( \langle 0, R_w \rangle \). Then, the linear image is defined as

\[
M \cdot W = \langle 0, MR_w \rangle
\]  

(27)

and the Minkowski sum of two centred zonotopes \( W = \langle c_w, R_w \rangle \) and \( G = \langle c_g, R_g \rangle \) is defined as

\[
W \oplus G = \langle 0, [R_w, R_g] \rangle.
\]  

(28)

In this work, zonotopes are used to compute reachable sets and therefore, the tube to implement the proposed robust MPC approach. The main reason for the use of zonotopes lies in their simplicity to operate with sets. Therefore, a set operation such as the Minkowski sum is reduced to a simple matrix addition. Note that, the use of Minkowski sum or difference of two polytopes is costly. However, using zonotopes the computational cost is reduced allowing a fast computation of basic sets operations [17]. These sets define the problem of finding the set of states that can be reached from a given set of states in a set of finite steps [23]. In
this approach, the main idea of using reachability theory is to bound the maximum achievable values for the mismatch error (8) between the prediction model and the real measurements at every sampling time. To this aim, the one-step robust reachable set from the set $\Phi$ is denoted as

$$\text{Reach}(\Phi, W) = \{ \exists x \in \Phi, \exists w \in W \text{ s.t. } y = (A_x + B_x K_x^m) x + w \}. \tag{29}$$

Note that, by using zonotic notation, the robust reachable set $\text{Reach}(\Phi, W)$ can be compactly written as

$$\text{Reach}(\Phi, W) = \{(A_x + B_x K_x^m) \Phi \oplus W \}. \tag{30}$$

Denoting the first initial reachable set as a null zonotope ($\Phi_0 = (0_{n_x}, 0_{n_x})$) and the disturbance set as a constant predefined zonotope ($W = (w_{c_u}, R_w)$), at every sampling time $k$ a group of reachable sets is computed by

$$\Phi_{k+i} = (A_{k+i} + B_{k+i} K_{k+i}^m) \Phi_{k+i} \oplus W, \quad \forall i = 0, \ldots, H_p, \tag{31}$$

where $H_p$ is the prediction horizon of the MPC strategy.

Note that, at time $k$, a number of $H_p + 1$ reachable sets are computed. Since the scheduling variables can be measured/estimated and computed, as the case of $\delta$, $\Phi_{k+i}$ is considered as $W$.

Then, the computation of each reachable set $\Phi_{k+i}$ will depend on its past realisation $\Phi_{k+i}$, the scheduling vector $\delta_k$, for computing system matrices ($A_{k+i}, B_{k+i}$), the controller $K_{k+i}$, and the uncertainty/disturbance set $W$. Finally, these reachable sets are used for computing the concatenation of consecutive resulting state/input sets along the prediction horizon at each time $k$, known as tube.

### 4.4 MPC design

Considering the previous discussions about the terminal conditions, the local controller and the reachable sets, in this section, we focus our attention on the T-MPC implementation. Fig. 2 shows the complete scheme used in this work. Note that, the model predictive strategy is in charge of controlling the nominal system while the differences between the real system and the nominal one are compensated by the local controller. Such a difference may be produced by external sources as an exogenous disturbances, unmodelled dynamics or by uncertain parameters in the nominal model. Then, in order to guarantee robustness against all these sources, the reachable sets are used to compute the input/state space where the feasibility is ensured under the presence of the maximum disturbances considered in the design.

**Remark 3:** Considering large disturbances acting over the vehicle implies bounding the differences between the real and the nominal system in a large set $W$ which will lead to a more conservative result and also to the reduction of the maximum prediction horizon in the MPC design.

The inputs and states sets are updated at every control iteration and introduced as the new input/state constraints throughout the prediction window (see e.g. of a two-inputs-two-states system in Fig. 4), as follows:

$$\begin{align*}
X_{k+i} &= X \oplus \Phi_{k+i}, \quad \forall i = 0, \ldots, H_p, \\
U_{k+i} &= U \oplus (K_{k+i}^m, \Phi_{k+i}), \quad \forall i = 0, \ldots, H_p - 1.
\end{align*} \tag{32}$$

Note that, as the prediction horizon increases the possibility of reaching empty sets becomes higher resulting then in an optimal problem without solution.

Finally, the grouping of all the previous steps allows us to formulate the optimal problem as a quadratic optimisation problem (see Fig. 5) to determine the next sequence of control actions considering that the values of $x_k$ and $u_{k-i}$ are known.

$$\begin{align*}
\min_{\Delta u_{k+i}} & \quad \bar{x}_{k+i}^T H_x \bar{x}_{k+i} + \sum_{i=0}^{H_p-1} \sum_{j=0}^{H_p-1} (x_{k+i} - \bar{x}_{k+i})^T Q (x_{k+i} - \bar{x}_{k+i}) \\
& \quad + \Delta \bar{u}_{k+i}^T R \Delta \bar{u}_{k+i} \\
\text{s.t.} & \quad \bar{x}_{k+i+1} = A_{k+i} \bar{x}_{k+i} + B_{k+i} \bar{u}_{k+i} \\
& \quad \bar{u}_{k+i} = \bar{u}_{k+i} + \bar{u}_{i+k} \\
& \quad \bar{u}_{k+i} \in U \ominus (K_{k+i}^m, \Phi_{k+i}) \\
& \quad \bar{x}_{k+i} \in X \ominus \Phi_{k+i} \\
& \quad \bar{x}_{k+i} + R_x \bar{x}_{k+i} \\
& \quad \bar{x}_{k+i} \in X \ominus \Phi_{k+i} \\
& \quad \bar{x}_{k+i} \in X \ominus \Phi_{k+i} \\
& \quad \bar{x}_{k+i} \in X \ominus \Phi_{k+i}
\end{align*} \tag{33}$$

where $r$ is the reference, $\bar{x}$ is the state vector of the prediction model (4), $\bar{u}$ is the optimal control action, $x$ is the feedback state vector from the real system, $P \in R^{n_x \times n_x} > 0$ represents the weighting matrix associated to the terminal cost computed in (18), $Q = Q^T \in R^{n_x \times n_x} \geq 0$ and $R = R^T \in R^{n_u \times n_u} \geq 0$ are the tuning matrices for the states and the variation of the control inputs, respectively.

**Remark 4:** The use of the weighting matrix $P$ associated to the terminal cost to enforce stability is widely used in the MPC literature. In the case of using a static $H_\infty$ controller, the proof that using the matrix $P$ obtained from (18) in the terminal cost enforces stability is presented in [24].

### 4.5 Summary of the method

In the following, the proposed method is summarised by means of two algorithms that summarises the off-line and on-line phases are as follows:

---

**Fig. 4** Example of reachable sets ($\Phi$) growth and new MPC constraints ($\bar{X}$ and $\bar{U}$) evolution for a prediction horizon of four steps using a two states two inputs system. $X$ and $U$ represent the original constraints.

**Fig. 5** Example of the prediction stage in the ZT-LPV-MPC technique at time $k = 0$. Reachable sets ($\Phi$) growth and new constraints ($\bar{X}$) are adapted throughout this stage to guarantee robust feasibility and stability.
4.5.1 Off-line phase: The off-line phase can be summarised in the following steps:

Step 1: Obtain the LPV model (2) of the vehicle from the non-linear model (1) using the non-linear embedding approach [18].
Step 2: Obtain the polytopic LPV model of the vehicle (11) using the bounding box approach [25].
Step 3: Design the LPV $H_\infty$ local controller solving the LMI problem (18).
Step 4: Calculate the invariant set with the algorithm presented in (23).

4.5.2 On-line phase: The on-line phase can be summarised in the following steps at each sampling time $k$:

Step 1: Measure the vehicle state $x_k$.
Step 2: Calculate the local control gain $K_\infty^r$ using (19).
Step 3: Calculate the reachable sets (31).
Step 4: Update the state and input bounds according to (32).
Step 5: Solve the MPC optimisation problem (33).
Step 6: Apply the control action $u = \bar{u} + u_\infty$.

5 Results
In this section, we validate the performance of the proposed ZT-LPV-MPC control scheme in a racing scenario through simulation in MATLAB. The principal objective of the presented scheme is to follow the proposed racing-based references ensuring asymptotic stability and the highest possible level of robust performance while dealing with exogenous disturbances.

The racing references are provided by a trajectory planner [26] and make the vehicle to perform close to its dynamic limits. The reference vector ($r$ in Fig. 2) is composed by two variables, the linear longitudinal speed and the angular velocity. Both are depicted as dashed lines in Fig. 6. Note that, the linear speed reference belongs to a low velocity interval, i.e. between 10 and 25 km/h. However, we understand a driving behaviour is closer to the limits of handling as the product between linear and angular velocities increases. The non-linear model used for simulation is a high-fidelity bicycle-based representation of the Driverless UPC vehicle [27] used in the Formula Student challenge [28] and is presented in Appendix 1. An identified tire model using the simplified Magic Formula [29] is used for generating accurate lateral forces from the front and rear slip angles. To verify the real-time feasibility of the presented approach, we perform the simulations on a DELL inspiron 15 (Intel core i7-8550U CPU @ 1.80GHzx8).

To show the effectiveness of the $H_\infty$-based approach presented in this work for computing the tube (Section 4.1), we perform a comparison against the LQR-based technique presented in [11] but redesigned for our presented vehicle model in (10). Hence, the comparison scenario is the same for both cases using the scheme presented in Fig. 2 where only the local controller changes for comparison purposes. The proposed scenario consists on two disturbance sources affecting the non-linear vehicle while driving in simulation. Such disturbance variables are chosen to be the road slope acting over the longitudinal vehicle dynamics ($\phi$) and lateral wind affecting the lateral and angular vehicle dynamics ($F_w$) (see Fig. 7). These external disturbances contained in $w$ belong to the set $W = \{ w \in \mathbb{R}^5 : H_w w \leq b_w \}$, where

$$H_w = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & -1 \end{bmatrix}, \quad b_w = \begin{bmatrix} 0.074 \\ 0.074 \\ 0.192 \\ 0.192 \\ 0.105 \\ 0.105 \\ 0.0 \\ 0.0 \end{bmatrix}. \quad (34)$$

The ZT-LPV-MPC uses the predicted data in the past realisation to instantiate the state-space matrices at every time step within the MPC prediction stage. Then, the optimal control problem (33) is solved at a frequency of 30 Hz using the solver GUROBI [30] through YALMIP [31] framework and the local controller is run at a higher frequency of 200 Hz. The tuning parameters for the robust LPV-MPC and LPV-$H_\infty$ problems are listed in Table 3 and (35), respectively.

$$P = 10^6 \begin{bmatrix} 0.0200 & -0.0000 & 0.0000 & -0.0268 & -0.0003 \\ -0.0000 & 0.0252 & -0.0128 & 0.0000 & 0.0146 \\ 0.0000 & -0.0128 & 0.0110 & -0.0000 & -0.0009 \\ -0.0268 & 0.0000 & -0.0000 & 0.0376 & 0.0004 \\ -0.0003 & 0.0146 & -0.0089 & 0.0004 & 1.8130 \end{bmatrix} \quad (35a)$$

$$E = 0.3 \begin{bmatrix} 0 & 0.5 & 0 & 0 & 0 & 0 \\ 2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.0001 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}. \quad (35b)$$

The off-line phase can be summarised in the following steps:

1. Measure the vehicle state $x_k$.
2. Calculate the local control gain $K_\infty^r$ using (19).
3. Calculate the reachable sets (31).
4. Update the state and input bounds according to (32).
5. Solve the MPC optimisation problem (33).
6. Apply the control action $u = \bar{u} + u_\infty$.

The on-line phase can be summarised in the following steps at each sampling time $k$:

1. Measure the vehicle state $x_k$.
2. Calculate the local control gain $K_\infty^r$ using (19).
3. Calculate the reachable sets (31).
4. Update the state and input bounds according to (32).
5. Solve the MPC optimisation problem (33).
6. Apply the control action $u = \bar{u} + u_\infty$.
\[
C = 10^{-4} \begin{bmatrix}
0.2 & 0 & 0 & 0 & 0 \\
0 & 0.2 & 0 & 0 & 0 \\
0 & 0 & 0.2 & 0 & 0 \\
0 & 0 & 0 & 0.2 & 0 \\
0 & 0 & 0 & 0 & 0.1 \\
0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 \\
\end{bmatrix}, \quad (35c)
\]

\[
D_1 = 10^{-4} \begin{bmatrix}
0 & 0 & -0.2 & 0 & 0 \\
0 & 0 & 0 & -0.2 & 0 \\
0 & 0 & 0 & 0 & -0.2 \\
0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 \\
\end{bmatrix}, \quad (35d)
\]

\[
D_2 = 10^{-3} \begin{bmatrix}
1 & 0 & 0 & 0 & 0 \\
-1 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 \\
0 & -1 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 & -1 \\
\end{bmatrix}, \quad (35e)
\]

\[
H_\zeta = \begin{bmatrix}
10 & -0.5 & 0.6 & 1.0 & 1000.0 & 3.14 & 3.14 \\
0 & 0 & 0 & 0 & 0 & 0.0 & 0.0 \\
0 & 0 & 0 & 0 & 0 & 3.14 & 3.14 \\
\end{bmatrix}, \quad b_\zeta = \begin{bmatrix}
0.15 & 0.15 \\
0 & 0 \\
0 & 0 \\
\end{bmatrix}, \quad (35f)
\]

The reference tracking results are depicted in Fig. 6. It can be seen the significant improvement of the presented scheme with respect to the ZT-LPV-MPC using the LQR controller as the corrective error approach ('LQR local control' in figures). Furthermore, the disturbance rejection has enhanced using a local controller whose design has been based on minimising the infinity norm instead of the 2-norm as the case of LQR approach. However, note that using a \( H_\infty \) design may produce troubles in the closed-loop response because of the large gains that are obtained and hence, a meticulous tuning is needed.

In Fig. 8, the errors (or mismatch) between the predicted state and the measured state are presented showing that the errors inside the considered bounds \( W \). Note that, such a vector of errors corresponds to the vector entering the state feedback local controller (\( e \) in Fig. 2). It can be appreciated the better performance of the strategy presented in this work being able to reject most of the error produced by the uncertainty and the applied exogenous disturbances.

Fig. 9 shows the control actions applied during the simulation test. Fig. 10 shows the elapsed time per iteration of the ZT-LPV-MPC strategy where the mean elapsed time per iteration is 16.4 ms using a prediction horizon of five steps.

In addition, in Table 4, we perform an elapsed time comparison between polytope-based and zonotope-based operations for computing the tube in a particular time instant. In this table, we show a computational improvement when using zonotopes of around 285 times faster than using a standard polytope formulation.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>( Q )</td>
<td>( 0.8^{\text{diag}}(\frac{1}{\xi_1^2} 0 0 0 0) )</td>
<td>( R )</td>
<td>( 0.2^{\text{diag}}(\frac{1}{\xi_2^2} 0 0 0) )</td>
</tr>
<tr>
<td>( \xi )</td>
<td>( [15 1.4 \infty \infty] )</td>
<td>( \xi )</td>
<td>( [1 -1.4 -\infty -\infty] )</td>
</tr>
<tr>
<td>( u )</td>
<td>( [0.267 13] )</td>
<td>( \Delta u )</td>
<td>( [-0.267 -2] )</td>
</tr>
<tr>
<td>( \Delta u )</td>
<td>( [0.05 0.5] )</td>
<td>( \Delta u )</td>
<td>( [-0.05 -0.5] )</td>
</tr>
<tr>
<td>( T_s )</td>
<td>33 ms</td>
<td>( H_p )</td>
<td>5</td>
</tr>
</tbody>
</table>

Fig. 8 Mismatch between real and nominal states. \( e_v \) represents the error in the longitudinal behaviour, \( e_v \) the error in the lateral behaviour, and \( e_\omega \) represents the error for the angular behaviour. Dotted red lines represent the maximal bounds for each one of the errors defining then the set \( W \).

Fig. 9 Control actions applied to the simulated vehicle (\( u \) in Fig. 2)

The comparison of the proposed approach, improving up to thirty times the angular velocity tracking error with respect to the compared strategy in the proposed disturbed scenario.
7 Acknowledgments

This work was funded by the Spanish Ministry of Economy and Competitiveness (MINECO) and FEDER through the projects SCAV (ref. DPI2017-88403-R) and HARCRICS (ref. DPI2014-58104-R). The author was supported by a FI AGAU grant (ref 2017 FI B00433).

8 References


IET Control Theory Appl., 2020, Vol. 14 Iss. 20, pp. 3676-3685
© The Institution of Engineering and Technology 2020
9 Appendix

9.1 Appendix 1: Vehicle model for simulation

For simulation purposes, we use a higher fidelity vehicle model. Unlike the model used for control design, this considers a more precise tire model, i.e. the Pacejka ‘Magic Formula’ tire model where the parameters $b$, $c$, and $d$ define the shape of the semi-empirical curve. Also, a more accurate computation of the tire slip angles is given.

Notice that, variables $\varphi$ and $v_r$ are exogenous disturbances and represent the longitudinal road slope and the lateral wind velocity, respectively. Furthermore, $C_{dAI}$ is the product of drag coefficient and vehicle lateral cross-sectional area. Parameters $d_f, d_r, c_f, c_r, b_f$ and $b_r$ are the simplified Pacejka model constants.

All the vehicle parameters are properly defined in Table 2.

\[
\begin{align*}
\dot{v}_x &= a_x + \frac{-F_{sf}\sin\delta - F_{df}}{m} + \alpha v_x - g\sin\varphi \\
\dot{v}_y &= \frac{F_{sf}\cos\delta + F_{sy} - F_w}{m} - \alpha v_y \\
\dot{\alpha}_f &= \delta - \tan^{-1}\left(\frac{v_y}{v_x}\right) \\
\dot{\alpha}_r &= -\tan^{-1}\left(\frac{v_y}{v_x}\right) \\
F_{sf} &= d_f\sin(c_f\tan^{-1}(b_f\alpha_f)) \\
F_{sy} &= d_r\sin(c_r\tan^{-1}(b_r\alpha_r)) \\
F_{df} &= \mu mg + \frac{1}{2}\rho C_{dAI}v_x^2 \\
F_w &= \frac{1}{2}\rho C_{dAI}v_y^2.
\end{align*}
\]

(36)

9.2 Tire stiffness LPV model

The Pacejka tire equations in (36) for front and rear wheels are reformulated in a LPV representation for a proper introduction in the final LPV vehicle model (10). Hence, starting from previous data representing the dynamics of the tires obtained by means of experimental tests, a least-squares algorithm is used to find two polynomials fitting the experimental tire data as

\[
F_i(\alpha) = p_1\alpha^n + p_2\alpha^{n-1} + \cdots + p_n\alpha + p_n+1,
\]

(37)

where $p$ constants are the estimated coefficients that define the particular model structure and $n$ represents the order of the corresponding polynomial.

Once the polynomial is adjusted, the embedding approach of the non-linearities inside a varying parameter has to be used in order to obtain its LPV representation. Then, the following formulation is proposed

\[
F_i = C(\alpha)\alpha,
\]

(38)

where

\[
C(\alpha) = p_1\alpha^{n-1} + p_2\alpha^{n-2} + \cdots + p_n + p_{n+1}\alpha + \epsilon
\]

(39)

is known as the tire stiffness coefficient and $\epsilon$ is a very small constant. Note that, as $\alpha$ becomes close to zero in (39), $C(\alpha)$ grows exponentially. To avoid this behaviour, a saturation is added in the small interval $\alpha \in [0, 0.0075]$ such that $C(\alpha)$ value stay at $4 \times 10^4$. Table 6 shows the coefficients used in (39).

\begin{table}[h]
\centering
\caption{Polynomial parameters of (39) for the front and rear tires (upper indexes $f$ and $r$)}
\begin{tabular}{lcc}
\hline
Parameter & Value & Parameter & Value \\
\hline
$n$ & 4 & $\epsilon$ & $10^{-4}$ \\
$p_1^f$ & $-2.167 \times 10^6$ & $p_1^r$ & $1.284 \times 10^6$ \\
p_2^f & $-0.288 \times 10^6$ & $p_2^r$ & $0.029 \times 10^6$ \\
p_3^f & 15.038 & $p_3^r$ & $1.198 \times 10^6$ \\
p_4^f & $-2.130 \times 10^6$ & $p_4^r$ & $0.024 \times 10^6$ \\
p_5^f & $-0.252 \times 10^6$ & $p_5^r$ & \\
p_6^f & 14.551 & & \\
\hline
\end{tabular}
\end{table}