



HAL
open science

NMPC via qLPV models and Taylor-based Scheduling Parameter Extrapolation: A Cartesian Robot Case Study

Marcelo Menezes Morato, Amir Naspolini, Julio Elias Normey-Rico, Olivier
Sename

► **To cite this version:**

Marcelo Menezes Morato, Amir Naspolini, Julio Elias Normey-Rico, Olivier Sename. NMPC via qLPV models and Taylor-based Scheduling Parameter Extrapolation: A Cartesian Robot Case Study. MED 2022 - 30th Mediterranean Conference on Control and Automation, Jun 2022, Athènes, Greece. 10.1109/MED54222.2022.9837209 . hal-03654135

HAL Id: hal-03654135

<https://hal.science/hal-03654135v1>

Submitted on 28 Apr 2022

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

NMPC via qLPV models and Taylor-based Scheduling Parameter Extrapolation: A Cartesian Robot Case Study

Marcelo M. Morato^{1,2}, Amir Napolini¹, Julio E. Normey-Rico¹ and Olivier Sename²

Abstract—In this brief paper, we present an overview of recent advances on Model Predictive Control (MPC) synthesis for nonlinear systems using quasi-Linear Parameter Varying (qLPV) embeddings. For such, we consider a highly nonlinear Cartesian robot benchmark as a case study. Specifically, we advocate on the use of recursive Taylor-based extrapolation maps to generate accurate estimates for the future trajectories of the qLPV scheduling parameters, as shown in recent findings. We show how these estimates can be used to enhance and fasten the corresponding MPC algorithms, offering comparable performances to state-of-the-art techniques, while maintaining relieved numerical burden during the implementation. Through realistic simulations of the Cartesian robot, we demonstrate the effectiveness and the real-time capabilities of the discussed method, which is tested against widely acknowledged techniques (the SQP qLPV MPC framework, and the CasADi NMPC solver).

I. INTRODUCTION

Model Predictive Control (MPC) is a very wide-spread method for the regulation of constrained processes [1]. Over the last decades, it has had considerable research interest, with extensions developed for a wide variety of systems and settings. The theoretical establishment of MPC corresponds to the formal guarantees of recursive feasibility of the optimisation and closed-loop stability [2].

Many works have studied the application of MPC for nonlinear systems (NMPC), e.g. [3], [4]. Nevertheless, such algorithms are usually not trivial and their corresponding online implementation comes at the cost of increases numerical burden due to the inherent nonlinear predictions, which complicates real-time applications. Until the late 10s, even the most efficient NMPC algorithms displayed exponential complexity growth w.r.t. system size. Recent tools have shown how these algorithms can be fastened [5], [6], yet through approximations of the nonlinear optimisation.

Anyhow, recent advances have shown how exact NMPC solutions with real-time capabilities can be generated through quasi-Linear Parameter Varying (qLPV) embeddings, see [7], [8]; compare also to the survey [9] and references therein. LPV models are able to describe nonlinear and time-varying behaviours under linear dynamics structures [10], which depend on known, bounded scheduling parameters ρ .

*This work has been supported by Campus France (Eiffel Scholarship), by CNRS (“Investissements d’Avenir”, ANR-15-IDEX-02), and by CNPq (304032/2019 – 0 researcher grant and PIBIC scientific initiation scholarship). The Authors thank J.P. Jordanou for his helpful insights on CasADi.

¹Dept. de Automação e Sis., Univ. Fed. de Santa Catarina, Brazil. marcelomnm@gmail.com

²Univ. Grenoble Alpes, CNRS, Grenoble INP[†], GIPSA-lab, 38000 Grenoble, France. [†]Institute of Engineering Univ. Grenoble Alpes.

The elegance of qLPV-embedding MPC design resides on the fact that the “full-blown” nonlinear predictions are replaced by linear ones [9], thus enabling fast operation of the corresponding optimisation. The majority of these algorithms (first proposed in [11] being established and generalised there since) are based on an iterative mechanism to estimate the future scheduling trajectories, applying the qLPV scheduling function over the previewed state trajectory and re-evaluating the optimisation until a certain convergence criteria is reached. Thereby, the resulting control algorithm has the numerical burden of a Sequential Quadratic Program (SQP).

Recent works have developed an alternative approach to the prior, where the future scheduling trajectories are estimated through linear recursive Taylor-based extrapolation laws [12], [13]. The novelty of this alternative is that the resulting MPC algorithm does not require to evaluate many QPs per sample, but rather just a single one. Moreover, convergence of the estimates to the true scheduling trajectories are obtained within a few samples, enabled by easy-to-verify sufficient conditions [14].

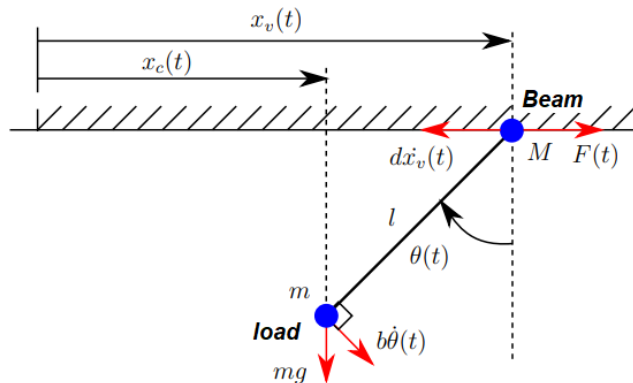


Fig. 1. Case study: Cartesian robot benchmark.

The concept of NMPC using the qLPV-embedding approach is recent, and the advances on linear Taylor-based scheduling trajectory estimates are even more recent. Therefore, in this brief paper, we recall this concept, detailing all necessary implementation steps. Moreover, we use a (highly nonlinear) Cartesian robot system, as illustrated in Fig. 1, as a benchmark to demonstrate the effectiveness of these algorithms. Specifically, we compare the Taylor-based technique, as proposed by the Authors in [14], to the state-of-the-art SQP qLPV MPC approach from [7], as well as to a benchmark NMPC solution generated through CasADi

[5]. Our intent is to advocate the use of qLPV-embedding with Taylor-based parameter extrapolation as a benchmark approach to NMPC.

Paper organisation. In Sec. II, we present the Cartesian robot benchmark system and the required setup for the corresponding qLPV embedding. In Sec. III, we provide the general qLPV NMPC problem setup, as well as the underlying assumptions required for correct implementation; we also give simple solutions to generate terminal ingredients that ensure stability and recursive feasibility properties. In Sec. IV, we detail the Taylor-based scheduling parameter trajectory extrapolation mechanism, also providing a brief convergence proof. In Sec. V, simulation results are presented, comparing the qLPV MPC approach to state-of-the-art NMPC. Concluding remarks are drawn in Sec. VI.

Notation. The index set $\mathbb{N}_{[a,b]}$ represents $\{i \in \mathbb{N} \mid a \leq i \leq b\}$, with $0 \leq a \leq b$. The identity matrix of size j is denoted as I_j ; $I_{j,\{i\}}$ denotes the i -th row of I_j . $\text{col}\{\cdot\}$ denotes the vectorisation of the entries and $\text{diag}\{v\}$ denotes the diagonal matrix generated with the line vector v . The predicted value of a given variable $v(k)$ at time instant $k+i$, computed based on the information available at instant k , is denoted as $v(k+i|k)$. \mathcal{K} refers to the class of positive and strictly increasing scalar functions that pass through the origin. $\|\cdot\|$ denotes the 2-norm. In matrix inequalities, (\star) denotes the corresponding symmetrical transpose.

II. PRELIMINARIES: CASE STUDY AND QLPV EMBEDDING

In this Section, we present the basic requirements for qLPV embeddings and then apply them to considered case study. For such, consider the following generic discrete-time nonlinear system:

$$\mathcal{G} := \begin{cases} x(k+1) &= f(x(k), u(k)), \\ y(k) &= f_y(x(k), u(k)), \end{cases} \quad (1)$$

being $k \in \mathbb{N}$ the sampling instant, $x \in \mathbb{R}^{n_x}$ the vector of states, $u \in \mathbb{R}^{n_u}$ the vector of control inputs, and $y \in \mathbb{R}^{n_y}$ the vector of measured outputs. Without loss of generality, we assume that the origin is an admissible equilibrium point of \mathcal{G} , i.e. there exists some \bar{u} s.t. $f(0, \bar{u}) = 0$. We assume that the states x are measured, thus we use the state-feedback $u(k) := \kappa(k)x(k)$ to ensure that the dynamics of \mathcal{G} adhere to desired specifications. We consider two possible objectives: (a) Regulation (steering x to the origin), and (b) Tracking (steering y to a steady-state target y_r , which conversely means steering (x, u) to a steady-state target pair (x_r, u_r)).

The system in Eq. (1) is subject to state and input constraints, which define an admissible operation: $x \in \mathcal{X} := \{x \in \mathbb{R}^{n_x} : \|x_j\| \leq \bar{x}_j, \forall j \in \mathbb{N}_{[1, n_x]}\}$ and $u \in \mathcal{U} := \{u \in \mathbb{R}^{n_u} : \|u_i\| \leq \bar{u}_i, \forall i \in \mathbb{N}_{[1, n_u]}\}$. The nonlinear function $f(x, u)$ is assumed of class \mathcal{C}^1 over $\mathcal{Z} := \mathcal{X} \times \mathcal{U}$, i.e. differentiable with first continuous derivatives.

A. qLPV Embedding

Assume that the nonlinear system \mathcal{G} satisfies an exact differential inclusion property [9], which means that it can

be embedded to a qLPV model. Therefore, suppose that there exists a map $H(x, u) \subseteq \mathbb{R}^{(n_x+n_y) \times (n_x+n_u)}$ such that $[f(x, u) \ f_y(x, u)] := H(x, u) \begin{bmatrix} x & u \end{bmatrix}^T$, $\forall (x, u) \in \mathcal{Z}$. In this case, it is implied that Eq. (1) is equivalent to:

$$\mathcal{G} := \begin{cases} x(k+1) &= A(\rho(k))x(k) + B(\rho(k))u(k), \\ y(k) &= C(\rho(k))x(k) + D(\rho(k))u(k), \\ \rho(k) &= f_\rho(x(k)) \in \mathcal{P}. \end{cases} \quad (2)$$

The model in Eq. (2) is qLPV, with an endogenous nonlinear function $f_\rho(\cdot)$ that results in the scheduling parameters $\rho(k)$, which are bounded and known at each instant k . These variables are unknown for any future instant $k+j$, $\forall j \in \mathbb{N}_{[1, \infty]}$. Through the sequel, we use $\mathcal{P} := \{\rho_j \in \mathbb{R} \mid \underline{\rho}_j \leq \rho_j \leq \bar{\rho}_j, \forall j \in \mathbb{Z}_{[1, n_\rho]}\}$.

In order to relieve conservativeness of the considered MPC algorithm, we assume that the scheduling variables exhibit bounded rates of variation:

Assumption 1: The scheduling variables $\rho(k)$ evolve along the horizon with bounded rates of variation. This is: $\delta\rho(k+1) = (\rho(k+1) - \rho(k)) \in \delta\mathcal{P} := \{\delta\rho_j \in \mathbb{R} : \underline{\delta\rho}_j \leq \delta\rho_j \leq \bar{\delta\rho}_j, \forall j \in \mathbb{N}_{[1, n_\rho]}\}$.

Remark 1: For simplicity, we present qLPV embedding in Eq. (2) for a realisation of the nonlinear dynamics with a scheduling proxy that is only dependent on the state variables, i.e. $f_\rho(\cdot)$ is only a function of $x(k)$. We stress some cases this function is state- and input-dependent, i.e. $\rho = f_\rho(x, u)$. The detailed procedures are applicable to such cases without loss of generality.

B. A Nonlinear Benchmark: Cartesian Robot

As previously stated, a highly nonlinear case study is used as a benchmark to demonstrate the effectiveness of the qLPV-embedding approach to MPC. Specifically, we consider the Cartesian robot system, which basically consists of a winch (vertical movement, z -axis), and is capable of performing movements along rails (longitudinal movement, x -axis. This robot is a benchmark nonlinear system in the likes of [15]; a graphical scheme is given in Fig. 1.

In order to model the dynamics of this Cartesian robot, the following baseline hypotheses are taken into account:

- 1) The beam translation force along the tracks $F(t)$, with corresponding mass M , can be imposed by an operator;
- 2) The winch cables are rigid, of negligible mass, and of constant length l ;
- 3) The load mass m is concentrated at one point, that is, the geometry of the load is negligible compared to the dimensions of the bridge;
- 4) The pendulum motion of the load is subjected to a viscous frictional torque given by $b\dot{\theta}(t)$, with $b > 0$;
- 5) The translational motion of the beam along the rail undergoes a viscous friction $d\dot{x}_v(t)$, with $d > 0$;
- 6) The acceleration of gravity is denoted g .

Thence, using Lagrangian mechanics and the Euler-

Lagrange equation, the following set of ODEs is obtained:

$$\begin{aligned}\ddot{x}_v(t) &= \frac{F(t)}{M + m \sin^2(\theta(t))} - \frac{b\dot{\theta}(t) \cos(\theta(t))}{l(M + m \sin^2(\theta(t)))} \\ &\quad - \frac{ml\dot{\theta}(t)^2 \sin(\theta(t))}{M + m \sin^2(\theta(t))} - \frac{mg \sin(\theta(t)) \cos(\theta(t))}{M + m \sin^2(\theta(t))} \\ &\quad - \frac{d\dot{x}_v(t)}{M + m \sin^2(\theta(t))}, \\ \ddot{\theta}(t) &= \frac{F(t) \cos(\theta(t))}{l(M + m \sin^2(\theta(t)))} - \frac{d\dot{x}_v(t) \cos(\theta(t))}{l(M + m \sin^2(\theta(t)))} \\ &\quad - \frac{g \sin(\theta(t))(M + m)}{l(M + m \sin^2(\theta(t)))} - \frac{b\dot{\theta}(t)(M + m)}{ml^2(M + m \sin^2(\theta(t)))} \\ &\quad - \frac{m\dot{\theta}(t)^2 \sin(\theta(t)) \cos(\theta(t))}{M + m \sin^2(\theta(t))}.\end{aligned}$$

We consider that the output of this system is the load position (which is to be controlled), given by:

$$y(t) = x_c(t) = x_v(t) - l \sin(\theta(t)),$$

where the control input is $u(t) = F(t)$ and the vector of state variables is given by $x(t) = [\theta(t) \ \dot{\theta}(t) \ x_v(t) \ \dot{x}_v(t)]^T$. The following parameters are used: the cable length l is of 7 m. The mass of the beam M and the mass of the load m are, respectively, 5000 Kg and 20000 Kg. The viscous friction constant d in the beam is of 2000 Kg/s and the viscous friction torque constant b in the load is of 0.00172 Kg m²/s.

This system must be regulated such that the following constraints are respected: $\theta \in [-\pi, \pi]$ rad (angle between the load and the beam); $\dot{\theta} \in [-\frac{1}{7}, \frac{1}{7}]$ rad/s (load angular speed); $x_v \in [-50, 50]$ m (beam position on the rail); $\dot{x}_v \in [-1, 1]$ m/s (beam speed); $u \in [-100, 100]$ N (force on the beam); $y \in [-50, 50]$ m (load position).

This system operates under a sampling period of $T_s = 1$ s, which gives a corresponding qLPV-embedding realisation as in Eq. (2), with system matrices given in Eqs. (5)-(6). This qLPV realisation is generated with:

$$\rho(k) = f_\rho(x(k), u(k)) = \begin{bmatrix} \cos(x_1(k)) \\ \frac{1}{M+m \sin^2(x_1(k))} \\ \text{sinc}(x_1(k)) \\ x_1(k+1)^2 \\ x_1(k)^2 \end{bmatrix}. \quad (3)$$

We note that both $x(k)$ and $u(k)$ are known and measured variables at instant k , and thus $\rho(k)$ can be generated explicitly through the nonlinear proxy $f_\rho(\cdot)$. As previously explained, the future behaviour of the scheduling parameters, i.e. $\rho(k+j), \forall j \geq 0$ is unknown, but estimated.

III. THE MPC SETUP

Taking into account the presented case study and the corresponding qLPV embedding realisation, we now detail the closed-loop paradigm with MPC.

Consider the robot system \mathcal{G} is regulated through an MPC algorithm, which generates a state-feedback control law

$u(k) = \kappa x(k)$, where κ is the corresponding feedback gain. This policy must be such that the state trajectories are steered to a given steady-state target, in an admissible manner (in the regulation case, this steady-state is the origin). Thus, in order to generate the MPC law, the qLPV model from Eq. (2) is spanned along a prediction horizon of N_p steps. For such the future scheduling trajectory $P_k = \text{col}\{\rho(k+j|k)\}, \forall j \in \mathbb{N}_{[0, N_p-1]}$, from which only $\rho(k)$ is *a priori* known. P_k is generated through the Taylor-based extrapolation procedure, detailed in Sec. IV.

A. MPC Design

Consider the following cost function:

$$\begin{aligned}J(x(k), P_k) &= \sum_{i=1}^{N_p} \ell(x(k+i|k), u(k+i-1|k)) \\ &\quad + V(x(k+N_p|k)),\end{aligned}$$

where the stage cost, which weights the performance along the horizon, is quadratic, as follows: $\ell(x, u) = \|x\|_Q + \|u\|_R$. We name $V(x)$ as the terminal cost, used to penalise the state at the end of the prediction horizon. We consider Q and R as positive definite weighting matrices, used to imply the envisioned trade-off between control effort and state regulation. In the sense of the case study, this trade-off is set between driving the load to a certain position (regulation) while minimising the beam translation force (control effort).

Definition 1: Positive Invariant Set

Assume that there exists a terminal set \mathbf{X}_f . \mathbf{X}_f is a positively invariant set for \mathcal{G} iff, for any $x(k) \in \mathbf{X}_f$ and $\rho(k) \in \mathcal{P}$, it follows that $x(k+1) \in \mathbf{X}_f$, where $x(k+1) := (A(\rho(k)) + B(\rho(k))\kappa)x(k)$.

Consider that there exists a terminal set \mathbf{X}_f which is positive invariant for \mathcal{G} . Then, the MPC is implemented as follows: at each sampling instant k , the states $x(k)$ are measured, the scheduling trajectory P_k is computed, and the following optimisation problem is solved, embedding performance objectives and constraints:

$$\begin{aligned}\min_{U_k} & J(x(k), P_k), \\ \text{s.t.} & \quad X_k = \mathbf{A}(P_k)x(k) + \mathbf{B}(P_k)U_k, \\ & \quad Y_k = \mathbf{C}(P_k)x(k) + \mathbf{D}(P_k)U_k, \\ & \quad x(k+j|k) \in \mathcal{X}, \forall j \in \mathbb{N}_{[1, N_p]}, \\ & \quad u(k+j-1|k) \in \mathcal{U}, \forall j \in \mathbb{N}_{[1, N_p]}, \\ & \quad x(k+N_p|k) \in \mathbf{X}_f,\end{aligned} \quad (4)$$

where $X_k = \text{col}\{x(k+j|k)\}, \forall j \in \mathbb{N}_{[1, N_p]}$ denotes the predicted sequence of states, $Y_k = \text{col}\{y(k+j|k)\}, \forall j \in \mathbb{N}_{[0, N_p-1]}$ gives the predicted sequence of outputs, and $U_k = \text{col}\{u(k+j|k)\}, \forall j \in \mathbb{N}_{[0, N_p-1]}$ gives the sequence of control actions along the prediction horizon. The prediction matrices $\mathbf{A}(\cdot), \dots, \mathbf{D}(\cdot)$ are given in Eqs. (7)-(9), maintaining the same form for all discrete-time samples $k \geq 0$.

Let $J^*(x(k), P_k)$ be the optimal solution of the optimisation in Eq. (4) at instant k , from which U_k^* is the optimal sequence of control inputs. Then, the MPC law

$$A(\rho) = \begin{bmatrix} 0 & 1 & 0 & 0 \\ \frac{-T_s g(M+m)\rho_3 \rho_2}{l} - m\rho_1 \rho_2 \rho_3 (\rho_4 + \rho_5) + \frac{b(M+m)\rho_2}{l^2 m} & 1 + 2m\rho_1 \rho_2 \rho_3 \rho_5 - \frac{b(M+m)\rho_2}{l^2 m} & \frac{d\rho_1 \rho_2}{l} & \frac{-d\rho_1 \rho_2}{l} \\ 0 & 0 & 0 & 1 \\ -T_s m g \rho_1 \rho_2 \rho_3 + \frac{b\rho_1 \rho_2}{l} - m l \rho_2 \rho_3 (\rho_4 + \rho_5) & -\frac{b\rho_1 \rho_2}{l} + 2m l \rho_2 \rho_3 \rho_5 & d\rho_2 & 1 - d\rho_2 \end{bmatrix}. \quad (5)$$

$$B(\rho) = \begin{bmatrix} 0 & \frac{T_s \rho_1 \rho_2}{l} & 0 & T_s \rho_2 \end{bmatrix}^T, \quad C(\rho) = \begin{bmatrix} -l\rho_2 & 0 & 1 & 0 \end{bmatrix}. \quad (6)$$

$$\mathbf{A}(P_k) = \begin{bmatrix} A(\rho(k)) & & & \\ A(\rho(k+1))A(\rho(k)) & & & \\ \vdots & & & \\ A(\rho(k+N_p-1))A(\rho(k+N_p-2)) \dots A(\rho(k)) & & & \end{bmatrix}, \quad \mathbf{C}(P_k) = \begin{bmatrix} C(\rho(k)) & & & \\ C(\rho(k+1))A(\rho(k)) & & & \\ \vdots & & & \\ C(\rho(k+N_p-1))A(\rho(k+N_p-2)) \dots A(\rho(k)) & & & \end{bmatrix}. \quad (7)$$

$$\mathbf{B}(P_k) = \begin{bmatrix} B(\rho(k)) & & & \\ A(\rho(k+1))B(\rho(k)) & & & \\ \vdots & & & \\ A(\rho(k+N_p-1)) \dots A(\rho(k+1))B(\rho(k)) & A(\rho(k+N_p-1)) \dots A(\rho(k+2))B(\rho(k+1)) & \dots & \end{bmatrix}. \quad (8)$$

$$\mathbf{D}(P_k) = \begin{bmatrix} D(\rho(k)) & & & \\ C(\rho(k+1))B(\rho(k)) & & & \\ \vdots & & & \\ C(\rho(k+N_p-1))A(\rho(k+N_p-2)) \dots A(\rho(k+1))B(\rho(k)) & C(\rho(k+N_p-1)) \dots C(\rho(k+2))B(\rho(k+1)) & \dots & \end{bmatrix}. \quad (9)$$

at instant k is given by the application the first entry of $U_k^* = \text{col}\{u^*(k+j|k)\}, \forall j \in \mathbb{N}_{[0, N_p-1]}$, i.e. $u^*(k|k)$, to the process. In order for the MPC to render closed-loop stability and a recursively feasible optimisation, the terminal ingredients (cost $V(\cdot)$ and set \mathbf{X}_f) should verify some conditions, further detailed in Sec. III-C.

B. Tracking

Consider that a tracking objective is envisioned, as in the considered case study application, for which the controlled output is the load position, which must steered to a given set-point. Then, the stage cost $\ell(x, u)$ must be set in order to minimise the variations of (x, u) from the desired set-point target (x_r, u_r) , which implies in an output target goal y_r . Following the lines of [16], the target goal can be generated using an offline reference optimisation selector, as follows.

Proposition 1: Assume there exists a linear (parameter varying) combination of the states x and inputs u that ensures $y(k) \rightarrow y_r$, considering the qLPV dynamics in Eq. (2). Then, the target goal $z_r = [x_r \quad u_r]^T$ can be determined through the solution of:

$$\begin{aligned} \min_{z_r} \quad & \| [C(f_\rho(z_r)) \quad D(f_\rho(z_r))] z_r - y_r \|^2, \quad (10) \\ \text{s.t.} \quad & [(A(f_\rho(z_r)) - \mathbb{I}_{n_x}) \quad B(f_\rho(z_r))] z_r = 0_{n_x}, \\ & f_\rho(z_r) \in \mathcal{P}, z_r^T \in (\mathbf{X}_f \times \mathcal{U}). \end{aligned}$$

Proof: This optimisation finds an admissible steady-state target z_r that imposes the output tracking objective. Full details of the proof are given in [16]. Note that this target selection problem can be solved online, at each sampling instant, if the output reference goal y_r is time-varying, which adds computational complexity. ■

C. Terminal Ingredients

We proceed by further detailing the how the terminal ingredients should be generated in order to render an asymptotically stable closed-loop, as well as a recursively feasible optimisation.

Assumption 2: The scheduling parameters assume a constant value in steady-state, i.e. $\rho = \rho_r$.

Assumption 3: The stage cost function is positive definite and uniformly continuous such that $\ell(e, \delta) \geq \alpha_\ell(\|e\|)$ and $|\ell(e_1, \delta_1) - \ell(e_2, \delta_2)| \leq \lambda_e(\|e_1 - e_2\|) + \lambda_\delta(\|\delta_1 - \delta_2\|)$, where α_ℓ , λ_e and λ_δ are \mathcal{K} -functions. It is implied that $\ell(0, 0) = 0$.

- Assumption 4:*
- 1) There exists an admissible terminal feedback law $\kappa_t(x) \in \mathcal{U}, \forall x \in \mathcal{X}$.
 - 2) The terminal set \mathbf{X}_f is closed, contains the origin, and represents an admissible positive invariant set.
 - 3) The terminal cost $V(\cdot)$ is continuous and positive for all $x \in \mathcal{X}$. Moreover $V(\cdot)$ represents a control Lyapunov function for the unconstrained qLPV system in Eq. (2), meaning that there exist constants $b > 0$ and $\sigma > 1$ such that $V(x) \leq b|x|^\sigma$. It is implied, thus, that $V(A(f_\rho(x))x + B(\rho)\kappa_t(x)) - V(x) \leq \ell(x, \kappa_t(x))$, for $V(x_1) - V(x_2) \leq \alpha_r(|x_1 - x_2|)$ (i.e. $V(\cdot)$ is a \mathcal{K} function).

The previous assumptions are standard. Accordingly, the following theorems ensure an admissible MPC operation.

Theorem 1: Stability and Recursive Feasibility [2]

Suppose there exists a terminal control law $u = \kappa_t(x)$. Consider that the qLPV system in Eq. (2) is controlled by the MPC, as rendered through Eq. (4). Then, input-to-state stability is ensured if the following conditions hold:

- (C1) The origin $x = 0$ lies in the interior of \mathbf{X}_f ;
- (C2) \mathbf{X}_f is positively invariant under the terminal feedback controller $\kappa_t(\cdot)$.
- (C3) The discrete Lyapunov equation is verified $\forall x \in \mathbf{X}_f$ and $\forall \rho \in \mathcal{P}: V(x^+) - V(x) \leq -\ell(x, \kappa_t(x))$.
- (C4) The image of the terminal control is admissible, i.e. $\kappa_t(\cdot) \in \mathcal{U}, \forall \rho \in \mathcal{P}$.
- (C5) The terminal set \mathbf{X}_f is a subset of \mathcal{X} .

Assuming that the initial solution of the MPC problem U_k^* is feasible, then, the MPC is recursively feasible.

Proof: This proof is standard; refer to [2]. ■

In order to satisfy the conditions required by Theorem 1, we choose the following quadratic terminal cost function $V(x) = x^T T x$, where $T = T^T$ is a positive definite weight. Accordingly, the terminal set \mathbf{X}_f is taken as a sub-level set of the terminal cost, i.e. $\mathbf{X}_f := \{x \in \mathbb{R}^{n_x} \mid x^T T x \leq 1\}$. By construction, \mathbf{X}_f is an ellipsoidal set constraint, which should be positively invariant for the terminal feedback $\kappa_t(\cdot)$. Thus, the following theorem gives a numerically solvable sufficient solution that generates these terminal ingredients.

Theorem 2: Terminal Ingredients [8]

Conditions (C1)-(C5) from Theorem 1 are satisfied if there exist a symmetric positive definite matrix $T \in \mathbb{R}^{n_x \times n_x}$ and a rectangular matrix $W \in \mathbb{R}^{n_u \times n_x}$ such that $Y = T^{-1} > 0$, $W = KY$ and that LMIs (11)-(13) hold under the minimisation of $\log \det\{Y\}$ for all $\rho \in \mathcal{P}$. The terminal feedback is then given by $\kappa_t(\cdot) = Kx$.

$$\begin{bmatrix} Y & \star & \star & \star \\ (A(\rho)Y + B(\rho)W) & Y & \star & \star \\ Y & 0 & Q^{-1} & \star \\ W & 0 & 0 & R^{-1} \end{bmatrix} \geq 0, \quad (11)$$

$$\left[\begin{array}{c|c} \bar{x}_j^2 & I_{\{j\}}Y \\ \hline I_{\{j\}}Y^T & Y \end{array} \right] \geq 0, \quad j \in \mathbb{N}_{[1, n_x]}, \quad (12)$$

$$\left[\begin{array}{c|c} \bar{u}_i^2 & I_{\{i\}}W \\ \hline \star & Y \end{array} \right] \geq 0, \quad i \in \mathbb{N}_{[1, n_u]}. \quad (13)$$

Proof: Consider P_k is known (via the extrapolation procedure detailed in Sec. IV). Since $\rho(k) = f_\rho(x(k))$, it follows that $\rho(k) = \rho_r$ in steady-state, iff $\lim_{k \rightarrow +\infty} x(k) = x_r$. Then, by applying a Schur complement to LMI (11), we obtain condition (C3), which suffices for (C2). By definition, an ellipsoid ensures (C1). (C4) and (C5) are respectively satisfied by applying Schur complements to LMI (12) and (13). In turn, the terminal feedback $\kappa_t(x) = Kx$ ensures that the qLPV system is asymptotically stable. This concludes the proof. \blacksquare

Remark 2: The terminal ingredients provided through Theorem 2 ensure recursive feasibility and asymptotic stability of the state trajectories. We note that LMI (11) is infinite-dimensional; in practice, its solution can be found by enforcing the inequalities over a sufficiently dense grid of points $\rho \in \mathcal{P}$. Then, the solution can be verified over a denser grid. If quadratic terminal ingredients are excessively conservative, a parameter-dependent alternative can be sought, if necessary, refer to [8].

IV. SCHEDULING PARAMETER EXTRAPOLATION

The concept of MPC is based on spanning a prediction of the process variables along a future horizon window. For such, we use the full state and output predictions $X_k = \mathbf{A}(P_k)x(k) + \mathbf{B}(P_k)U_k$, $Y_k = \mathbf{C}(P_k)x(k) + \mathbf{D}(P_k)U_k$, with prediction matrices given in Eqs. (7)-(9). For such, the future scheduling sequence P_k is required.

Accordingly, we detail, in this Section, the recent framework from [13], [14], which uses a first-order Taylor expansion of the scheduling proxy $f_\rho(x(k))$ to construct the future scheduling sequence. First and foremost, we stress that the main advantage of this extrapolation is that predicted

sequence P_k is generated through a single recursive linear law, numerically much cheaper than the state-of-the-art procedure from [7], which requires the iterative re-evaluation of $P_k^l = f_\rho(X_k^{l-1})$, where the super-index l indicates the l -th iteration of the MPC optimisation.

Denote $\delta x(k+j) = x(k+j+1) - x(k+j)$ as the incremental state deviation. By definition, δx is bounded¹ to a compact and convex box-type set $\Delta\mathcal{X} := \{\delta x \in \mathbb{R}^{n_x} : |\delta x_j| \leq \bar{\delta}x_j, \forall j \in \mathbb{N}_{[1, n_x]}\}$.

The method is as follows: re-write the static scheduling map $f_\rho(x)$ through the following Taylor expansion: approximated by:

$$f_\rho(x) = f_\rho(x)|_{\check{x}} + \left. \frac{\partial f_\rho(x)}{\partial x} \right|_{\check{x}} (x - \check{x}) + \xi_\rho, \quad (14)$$

being \check{x} the expansion point and ξ_ρ a residual which inherits the discrepancy between the real static map and its approximate. Since $f_\rho(x)$ is assumed class \mathcal{C}^1 , it is direct that the partial derivatives $\left. \frac{\partial f_\rho(x)}{\partial x} \right|_{\check{x}}$ are ultimately bounded for all $\check{x} \in \mathcal{X}$.

From this development, we obtain:

$$\begin{aligned} \rho(k+1) &= \rho(k) + f_\rho^\partial(k)\delta x(k) + \xi_\rho(k), \\ &\vdots \\ \rho(k+N_p-1) &= \rho(k+N_p-2) \\ &\quad + f_\rho^\partial(k+N_p-2)\delta x(k+N_p-2) \\ &\quad + \xi_\rho(k+N_p-2), \end{aligned}$$

where the derivative terms are given by $f_\rho^\partial(k+j) = \left. \frac{\partial f_\rho(x)}{\partial x} \right|_{x(k+j)}$. Note that $\rho(k)$ and $\delta x(k)$ are known variables at each instant k , whereas $f_\rho^\partial(k)$ can be numerically evaluated. In practice, $f_\rho^\partial(k+j)$ is unknown for $j \in \mathbb{N}_{[1, N-2]}$ but it can be replaced by $f_\rho^\partial(k+j) = f_\rho^\partial(k), \forall j \in \mathbb{N}_{[1, N-2]}$, where f_ρ^∂ denotes the partial derivative evaluated at instant k , see the discussion in [14]. By doing so, it is implied that $\rho(k+j) \approx \rho(k+j-1) + f_\rho^\partial \delta x(k+j-1|k)$.

Therefore, the estimate for the future scheduling variables can be written as the sum of the estimate from the previous sample corrected with an adjustment term $f_\rho^\partial \delta x(k+j-1)$. Through the sequel, we neglect the bias term ξ_ρ , since it is bounded and small, as argued in [14]. In any case, robustified MPC schemes with respect to such estimation error are discussed in [8], [13]. Accordingly, we can write the vector-wise extrapolation in a recursive fashion:

$$P_k = P_{k-1} + f_\rho^\partial \delta X_k, \quad (15)$$

where the sequence of state increments is given by $\delta X_k = \text{col}\{\delta x(k+j)\}^T, \forall j \in \mathbb{N}_{[0, N_p-2]}$.

Lemma 1: Assume that $f_\rho(\cdot)$ is class \mathcal{C}^1 and that f_ρ^∂ is ultimately bounded. Assume that the system (2) is stable in closed-loop with $u(k) = \kappa x(k)$. Then, the recursive extrapolation algorithm in Eq. (15) converges.

¹Bounds can be generated directly from Eq. (2), using $x \in \mathcal{X}$.

Proof: This proof is reduced for brevity, full details are given in [14], which presents five (simple-to-verify) sufficient conditions for convergence. Consider that the residual term $\xi_\rho(k+j)$ should turn null. Thus, use $\lim_{k \rightarrow \infty} x(k) = x_r$ holds (stability) and take $\xi_\rho(k) = f_\rho(x(k+1)) - f_\rho(x(k)) - f_\rho^\partial \delta x(k)$. Due to the stabilisation, it directly follows that $\lim_{k \rightarrow \infty} f_\rho(x(k)) = \lim_{k \rightarrow \infty} f_\rho(x_r)$ and $\lim_{k \rightarrow \infty} \delta x(k) = 0$, which implies in $\lim_{k \rightarrow \infty} \xi_\rho(k) = -\lim_{k \rightarrow \infty} f_\rho^\partial \delta x(k) \rightarrow 0$. This concludes the proof. ■

We recall the major advantages of this Taylor-based extrapolation method: (i) the resulting online operation is recursive, only residing in linear operations (i.e. Eq. (15)), thus being numerically-cheap; (ii) the estimated scheduling parameter trajectories P_k converge to the true values within a finite number of discrete-time samples (as gives Lemma 1), which means that precise information on the future behaviour can be passed to the MPC; (iii) the estimation error is small and bounded (refer to [14, Lemma 3]).

V. NONLINEAR SIMULATION RESULTS

In this Section, we show the simulation results of the nonlinear case study. Specifically, in order to illustrate the effectiveness of the qLPV-embedding approach to NMPC design, three distinct strategies are used to control the Cartesian robot:

- The benchmark approximation-based NMPC solution CasADi, from [5].
- The state-of-the-art SQP-based qLPV MPC from [7], previously shown as a comparable method to approximation-based NMPC solutions (such as ACADO, GRAMPC, and CasADi), e.g. [17]. In this approach, the QP in Eq. (4) is solved l times per sample, with the scheduling parameter trajectory estimate being given by $P_k^l = f_\rho(X_k^l)$.
- The highlighted Taylor-based extrapolation single QP qLPV MPC from [13], [14], where the scheduling trajectory is estimated by the means of Eq. (15).

For simplicity, we denote these methods henceforth as “CasADi NMPC”, “SQP qLPV MPC”, and “Taylor qLPV MPC”, respectively. The following results were obtained in a 2.1 GHz, 8 RAM PC, using Matlab, Yalmip, and Gurobi (qLPV MPCs), and Python (CasADi).

As previously stated, the control objective is to steer the load position $y(k)$ according to a piece-wise constant reference signal $r(k)$, which must be tracked. Thereof, we use the corresponding tracking stage cost $\ell(x, u) = \|y - y_r\|_Q + \|u\|_R$, which can be rewritten as $\|x - x_r\|_{Q_x} + \|u - u_r\|_{R_x}$ using suitable coordinate changes. The tracking objective given through (x_r, u_r) is obtained via Proposition 1. For the synthesis of the MPCs, we use a prediction horizon of $N_p = 10$ samples, together with the following weighting matrices $Q = 1$, and $R = 0.02$ (thus prioritising output tracking over control effort minimisation). The initial conditions are: $x(0) = [0 \quad -\frac{1}{4000} \quad 3 \quad \frac{1}{100}]^T$ and $u(0) = 0$.

Remark 3: The CasADi solution is implemented with minor changes to the control input constraints (a stricter slew rate is used) and to the stage cost $\ell(\cdot)$ (used as

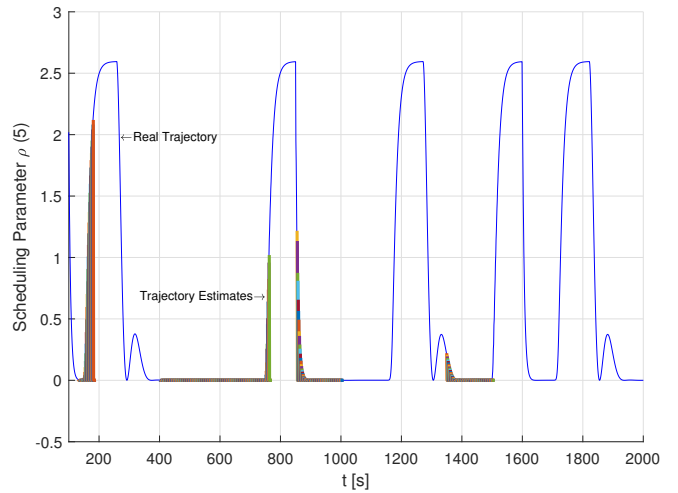


Fig. 2. Scheduling trajectory and Taylor-based extrapolation estimates (at different samples). Coloured lines represent the trajectory estimates of N_p -samples, while the blue curve stands for the real scheduling parameter trajectory.

$\|x - x_r\|_Q + \|u\|_R$). These adjustments are done in order to bypass the strong nonlinearities of the MPC optimisation at some samples, which are softened and thus infeasibility is avoided.

First of all, we show the accurateness of the scheduling parameter trajectory estimated generated by the Taylor-based recursive argument, detailed in Sec. IV. This is illustrated in Fig. 2, which gives the full behaviour of ρ_5 plotted against sampled trajectories of the fifth entry of P_k . Evidently, the recursive extrapolation law generates very precise estimations, with negligible residual (also true for the remaining scheduling variables, not shown due to lack of space). This further emphasises the applicability of the Taylor-based extrapolation scheme, which is reliable and passes precise information to the MPC regarding the future behaviour of ρ . A significant advantage is that each P_k is generated linearly at sample k , which is numerically very cheap (in contrast to the SQP method, which requires a nonlinear vector map $P_k^l = f_\rho(X_k^l)$ and multiple QPs).

Next, Figs. 3 and 4 show the output and input trajectories, respectively, obtained with each control method. The results from CasADi are slightly deviated due to the issues mentioned in Remark 3. We stress that both qLPV MPC methods yield very similar results in terms of reference tracking, which is obtained with offset-free error in steady state. For the considered system, the SQP MPC approach took, in average, two iterations of its internal loop to achieve a solution, meaning that it held a computational complexity of roughly two QPs per sample.

Finally, Table I presents corresponding performance indexes², given w.r.t. those from the CasADi NMPC solution (benchmark). We highlight the key findings:

- 1) In terms of real-time capabilities, the qLPV MPC approaches are shown to be very fast solutions to

²We use IAE as $\sum_k \|r(k) - y(k)\|$, being $r(k)$ the reference signal.

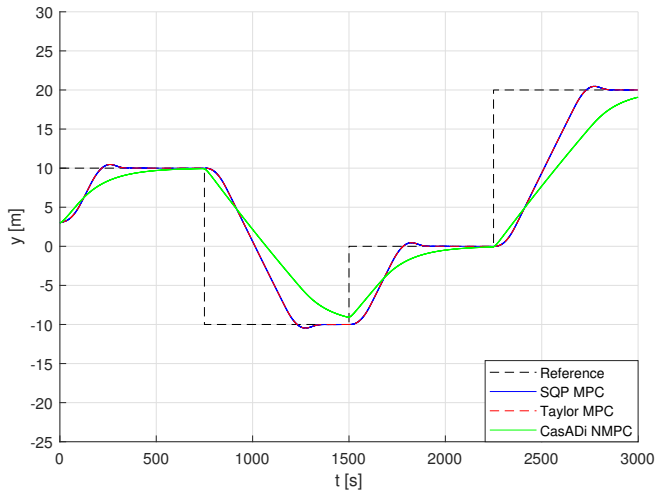


Fig. 3. Cartesian robot: Reference tracking results.

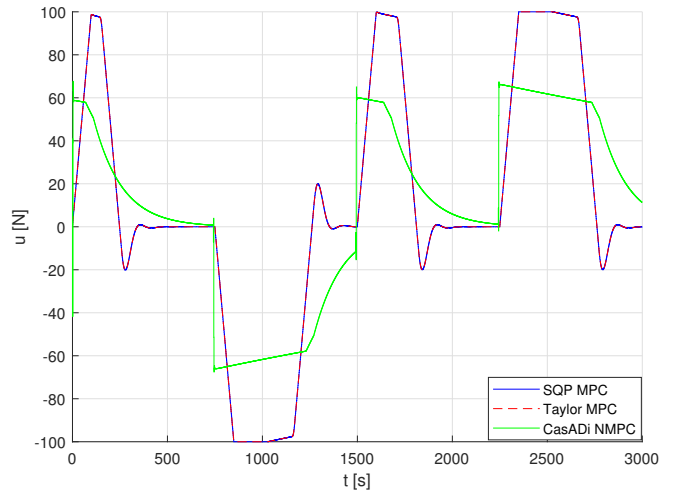


Fig. 4. Cartesian robot: Control inputs.

NMPC. The average computational burden with these approaches (t_c) is even smaller than what is required by CasADi (more than 20 times smaller), which is quite impressive.

- 2) Despite the SQP qLPV MPC solution being widely applied in recent works, e.g. [7], [17], we advocate for the use of the Taylor-based approach from [14]. By using a linear recursive estimate to estimate P_k , the total numerical burden of the controller is reduced in over 30 %, since only one QP is required to be solved per sample (and not multiple ones as in [7]). As shown in [14, Lemma 3]), we note that the estimation residual can be easily verified to be bounded, small, and null in steady-state.
- 3) The use of qLPV-embeddings is an elegant and efficient way to encapsulate nonlinearities of system that satisfy differential inclusion. The corresponding MPCs do not have to resort to any nonlinear optimisation, but rather quadratic ones. Furthermore, less conservative control is allowed, since recursive feasibility is ensured and the artifices to circumvent feasibility losses due to local minima (i.e. Remark 3) do not have to be used.

TABLE I
PERFORMANCE COMPARISON.

Method	$\text{rms}\{J(\cdot)\}$	IAE	t_c
CasADi NMPC	100 %	100 %	353.9 ms
SQP qLPV MPC	0.33 %	77.83 %	17.4 ms
Taylor qLPV MPC	0.33 %	77.83 %	12.3 ms

VI. CONCLUDING REMARKS

In this paper, we advocate for the qLPV-embedding framework as a standard approach for the development of NMPC algorithms with exact solution. Specifically, we highlight and exemplify recent findings on use of recursive Taylor-based extrapolation maps to generate accurate estimates for the future trajectories of the process qLPV scheduling parameters.

Using a highly nonlinear Cartesian robot system, we demonstrate the effectiveness of the qLPV approach using these Taylor-based extrapolation estimates. For such, the method is compared in terms of numerical burden and achieved performances to a state-of-the-art qLPV MPC technique and a benchmark NMPC solution (CasADi). We show how the analysed framework is able to achieve comparable results to state-of-the-art NMPC algorithms with much reduced numerical burden; the method enables real-time NMPC applications using only one quadratic program and one linear recursive law per sample during the implementation, which is a significant advance.

REFERENCES

- [1] E. F. Camacho and C. Bordons, *Model predictive control*. Springer Science & Business Media, 2013.
- [2] D. Q. Mayne, J. B. Rawlings, C. V. Rao, and P. O. Scokaert, "Constrained model predictive control: Stability and optimality," *Automatica*, vol. 36, no. 6, pp. 789–814, 2000.
- [3] J. Köehler, M. A. Müller, and F. Allgöwer, "A nonlinear model predictive control framework using reference generic terminal ingredients," *IEEE Transactions on Automatic Control*, 2019.
- [4] V. M. Cunha and T. L. Santos, "Robust nonlinear model predictive control with bounded disturbances based on zonotopic constraint tightening," *Journal of Control, Automation and Electrical Systems*, vol. 33, no. 1, pp. 78–91, 2022.
- [5] J. A. Andersson, J. Gillis, G. Horn, J. B. Rawlings, and M. Diehl, "CasADi: a software framework for nonlinear optimization and optimal control," *Mathematical Programming Computation*, vol. 11, no. 1, pp. 1–36, 2019.
- [6] S. Gros, M. Zanon, R. Quirynen, A. Bemporad, and M. Diehl, "From Linear to Nonlinear MPC: Bridging the Gap via the Real-time Iteration," *International Journal of Control*, vol. 93, no. 1, pp. 62–80, 2020.
- [7] P. S. G. Cisneros and H. Werner, "Nonlinear model predictive control for models in quasi-linear parameter varying form," *International Journal of Robust and Nonlinear Control*, vol. 30, no. 10, pp. 3945–3959, 2020.
- [8] M. M. Morato, V. Cunha, T. Santos, J. Normey-Rico, and O. Sename, "Robust nonlinear predictive control through qLPV embedding and zonotope uncertainty propagation," *IFAC-PapersOnLine*, vol. 54, no. 8, pp. 33–38, 2021, 4th IFAC Workshop on Linear Parameter Varying Systems LPVS 2021.
- [9] M. M. Morato, J. E. Normey-Rico, and O. Sename, "Model predictive control design for linear parameter varying systems: A survey," *Annual Reviews in Control*, vol. 49, pp. 64–80, 2020.

- [10] J. Mohammadpour and C. W. Scherer, *Control of linear parameter varying systems with applications*. Springer Science & Business Media, 2012.
- [11] P. S. Cisneros, S. Voss, and H. Werner, "Efficient nonlinear model predictive control via quasi-LPV representation," in *2016 IEEE 55th Conference on Decision and Control (CDC)*. IEEE, 2016, pp. 3216–3221.
- [12] M. M. Morato, G. Q. Bao Tran, G. N. dos Reis, J. E. Normey-Rico, and O. Sename, "NMPC through qLPV embedding: A tutorial review of different approaches," *IFAC-PapersOnLine*, vol. 54, no. 6, pp. 302–307, 2021, 7th IFAC Conference on Nonlinear Model Predictive Control NMPC 2021.
- [13] M. M. Morato, J. E. Normey-Rico, and O. Sename, "A fast dissipative robust nonlinear model predictive control procedure via quasi-linear parameter varying embedding and parameter extrapolation," *International Journal of Robust and Nonlinear Control*, vol. 31, no. 18, pp. 9619–9651, 2021.
- [14] —, "Sufficient conditions for convergent recursive extrapolation of qLPV scheduling parameters along a prediction horizon," 2022, pre-print. [Online]. Available: <https://hal.archives-ouvertes.fr/hal-03536927>
- [15] C.-C. Cheng and C.-Y. Chen, "Controller design for an overhead crane system with uncertainty," *Control Engineering Practice*, vol. 4, no. 5, pp. 645–653, 1996.
- [16] D. Limon, A. Ferramosca, I. Alvarado, and T. Alamo, "Nonlinear MPC for tracking piece-wise constant reference signals," *IEEE Transactions on Automatic Control*, vol. 63, no. 11, pp. 3735–3750, 2018.
- [17] P. G. Cisneros, S. Voss, A. Datar, H. Martinez, P. Gottsch, and H. Werner, "Plenary at the 3rd IFAC workshop on linear parameter varying systems: Fast Nonlinear MPC via quasi-LPV Models," IFAC, 2019. [Online]. Available: <https://lpvs2019.tue.nl/plenary-speakers/>