



**HAL**  
open science

## NMPC Through qLPV Embedding: A Tutorial Review of Different Approaches

Marcelo Menezes Morato, Gia Quoc Bao Tran, Guilherme N G dos Reis, Julio E Normey-Rico, Olivier Sename

► **To cite this version:**

Marcelo Menezes Morato, Gia Quoc Bao Tran, Guilherme N G dos Reis, Julio E Normey-Rico, Olivier Sename. NMPC Through qLPV Embedding: A Tutorial Review of Different Approaches. 7th IFAC Conference on Nonlinear Model Predictive Control, Jul 2021, Bratislava, Slovakia. hal-03210253v1

**HAL Id: hal-03210253**

**<https://hal.science/hal-03210253v1>**

Submitted on 27 Apr 2021 (v1), last revised 30 Apr 2021 (v2)

**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 Through qLPV Embedding: A Tutorial Review of Different Approaches <sup>★</sup>

Marcelo M. Morato<sup>\*,\*\*</sup> Gia Quoc Bao Tran<sup>\*\*</sup>  
Guilherme N. G. dos Reis<sup>\*</sup> Julio E. Normey-Rico<sup>\*</sup>  
Olivier Sename<sup>\*\*</sup>

<sup>\*</sup> *Dept. de Automação e Sistemas (DAS), Univ. Fed. de Santa Catarina, Florianópolis-SC, Brazil (marcelomnzm@gmail.com).*

<sup>\*\*</sup> *Univ. Grenoble Alpes, CNRS, Grenoble INP<sup>†</sup>, GIPSA-lab, 38000 Grenoble, France. <sup>†</sup>Institute of Engineering.*

---

**Abstract:** Nonlinear Model Predictive Control (NMPC) formulations through quasi-Linear Parameter Varying (qLPV) embeddings have been brought to focus in recent literature. The qLPV realisation of the nonlinear dynamics yields linear predictions at each sampling instant. Thereby, these control strategies offer online implementation with numerical toughness of Sequential Quadratic Programs (SQPs), which can be solved much faster than the Nonlinear Programs (NPs) generated with “regular” NMPC design. The general lines of such methods are as follows: (i) The qLPV embedding is formulated with state-dependent scheduling parameters; (ii) recursive extrapolation procedures are used to estimate the values of these parameters along the prediction horizon; (iii) these estimates are used to compute linear predictions, which are used to compute the control law through a constrained optimisation procedure. This paper details the overall concept of these novel NMPC techniques and reviews two different (efficient) implementation options. Realistic academic examples are also provided, considering a suspension system and a cascaded tank process.

*Keywords:* Nonlinear Model Predictive Control, Quasi-Linear Parameter Varying Systems, Tutorial, Sub-Optimal, Efficient Algorithms.

---

## 1. INTRODUCTION

Model Predictive Control (MPC) is very well recognised both in industry and academia as a standard tool for the optimal regulation of processes subject to constraints. Nonetheless, nonlinear MPC (NMPC) design is not trivial and comes with an increased numerical burden, which often becomes an impediment for real-time applications. This was a very substantial issue until the late 10’s, since even the most efficient NMPC algorithms available by then displayed exponential complexity increase with the number of system states and prediction horizon size, as emphatically stated by all systematic reviews on the topic (Allgöwer and Zheng, 2012).

Anyhow, over the last decade, there has been remarkable progress on expanding the real-time capabilities of NMPC (Gros et al., 2020), with tools that enable sufficiently fast solutions of the Nonlinear Programs (NPs), such as real-time iteration methods (Quirynen et al., 2015), gradient-based approaches (Englert et al., 2019), and input-parametrisation techniques (Rathai, 2020).

In parallel to the progress on real-time NMPC, the Linear Parameter Varying (LPV) toolkit has been progressively popularised (Sename et al., 2013). For many nonlinear systems, quasi-LPV (qLPV) embedding offers an exact representation of the process dynamics with elegance and

simplicity since there is no need to compute nonlinear state transitions but only linear maps with time-varying parameters  $\rho$ . These parameters are bounded and known between sampling instants.

With regard to this context, recent advances have conceived the design of NMPC schemes with qLPV embedded models, as surveyed in (Morato et al., 2020a). These qLPV MPC algorithms also provide fast online implementation since the “full-blown” nonlinear predictions are replaced by a sequence of linear predictions, which derive Sequential Quadratic Programs (SQPs), e.g., (Hanema et al., 2017; Morato et al., 2019; Cisneros and Werner, 2020).

The notion of deploying NMPC through qLPV embedded models is very recent. Therefore, in this brief tutorial paper, we review and discuss the overall concept of these methods, as well as detail the implementation steps and illustrate their application to nonlinear processes.

The rest of this paper is organised as follows. In Sec. 2, we provide the NMPC problem setup, together with the procedure to obtain qLPV realisations and the underlying assumptions required for correct implementation. In Sec. 3, we present different mechanisms that can be used to estimate the future values of the qLPV scheduling parameters. Sec. 4 gives standard LMIs for the computation of terminal ingredients, used to ensure the stability of the closed-loop system and recursive feasibility of the MPC optimisation. In Sec. 5, we illustrate two different applications, consider-

---

<sup>\*</sup> This work has been supported by *CNPq* (304032/2019 – 0), *CAPES* (001) and *ITEA3* European project (15016) *EMPHYSIS*.

ing the regulation goal of a semi-active suspension system and a level tracking problem in a quadruple-tank process. Concluding remarks are drawn in Sec. 6.

## 2. PROBLEM SETUP

Consider the following discrete-time nonlinear system:

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

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, consider that the origin is an equilibrium point.

In order to apply an MPC algorithm to control this system, a suitable operation is defined through set constraints:  $x \in \mathcal{X}$  and  $u \in \mathcal{U}$ , with:

$$\begin{aligned} \mathcal{X} &:= \{x_j \in \mathbb{R} : \|x_j\| \leq \bar{x}_j, \forall j \in \mathbb{N}_{[1, n_x]}\}, \\ \mathcal{U} &:= \{u_i \in \mathbb{R} : \|u_i\| \leq \bar{u}_i, \forall i \in \mathbb{N}_{[1, n_u]}\}. \end{aligned}$$

For simplicity, we consider that the states  $x$  are measured for all sampling instants. Therefore, MPC can be formulated through state-feedback  $u(k) := \kappa(k)x(k)$  such that the closed-loop dynamics  $x(k+1) = f_\pi(x(k))$ ,  $y(k) = f_{y_\pi}(x(k))$  abide to the desired specifications. W.r.t. this matter, there are two possible control objectives: (a) regulation, meaning that the states  $x$  must be steered to the origin, and (b) tracking, meaning that the outputs  $y$  must be steered to some admissible point  $y_r$ , which conversely means to steer  $(x, u)$  to  $(x_r, u_r)$ .

### 2.1 qLPV Embedding

A great deal of nonlinear systems in the form of Eq. (1) can be described with an exact qLPV realisation, if the Linear Differential Inclusion (LDI) property is verified (Abbas et al., 2014). Suppose that  $\exists H(x, u) \in \mathbb{R}^{(n_x+n_y) \times (n_x+n_u)}$ ,  $\forall (x, u) \in (\mathcal{X} \times \mathcal{U})$  s.t.

$$[(f(x, u))^T (f_y(x, u))^T]^T := H(x, u) [x^T u^T]^T.$$

Then, Eq. (1) within  $(\mathcal{X} \times \mathcal{U})$  is equivalent to:

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

The qLPV model in Eq. (2) is scheduled by an endogenous nonlinear map  $f_\rho(\cdot)$ , which imposes bounded and known scheduling variables  $\rho$  for all sampling instants. These variables are unknown for any future instant  $k+j$ ,  $\forall j \in \mathbb{N}_{[1, \infty]}$ . The scheduling set is given by:

$$\mathcal{P} := \left\{ \rho_j \in \mathbb{R} \mid \underline{\rho}_j \leq \rho_j \leq \bar{\rho}_j, \forall j \in \mathbb{Z}_{[1, n_\rho]} \right\}.$$

*Remark 1.* If the LDI property is not verified, one can still find qLPV models that represent the nonlinear dynamics through convex or convex-concave inclusions (Sala et al., 2020), at the expense of conservativeness.

*Remark 2.* Through the sequel, we will drop the dependency of  $f_\rho$  on  $u$ , for notation simplicity. We note that qLPV embedding with state-dependent scheduling parameter holds for the vast majority of qLPV applications

(Morato et al., 2020a). Nevertheless, input-dependent scheduling variables can be treated with the same methodologies herein described.

*Assumption 1.* The scheduling variables  $\rho(k)$  evolve along the horizon with bounded rates of variation. This is:  $\partial\rho(k+1) = (\rho(k+1) - \rho(k)) \in \partial\mathcal{P}$ , with:

$$\partial\mathcal{P} := \left\{ \partial\rho_j \in \mathbb{R} : \underline{\partial\rho}_j \leq \partial\rho_j \leq \bar{\partial\rho}_j, \forall j \in \mathbb{N}_{[1, n_\rho]} \right\}. \quad (3)$$

### 2.2 The MPC formulation

In order to suitably control the nonlinear process in Eq. (2), we consider an MPC approach. For such, at each sampling instant  $k$ , the states  $x(k)$  are measured, the scheduling parameters  $\rho(k)$  are computed, and the following optimisation problem, which embeds the performance objectives (regulation/tracking), as well as the operational constraints, is solved:

$$\begin{aligned} \min_{U_k} & V(x(k+N_p|k)) \\ & + \sum_{j=1}^{N_p} \ell(x(k+j|k), u(k+j-1|k)) \\ \text{s.t.} & x(k+j+1|k) = A(\rho(k+j|k))x(k+j|k) \\ & + B(\rho(k+j|k))u(k+j|k), j \in \mathbb{N}_{[0, N_p-1]}, \\ & y(k+j) = C(\rho(k+j|k))x(k+j|k) \\ & + D(\rho(k+j|k))u(k+j|k), j \in \mathbb{N}_{[0, N_p-1]}, \\ & (x^\top(k+j|k), u^\top(k+j|k))^\top \in (\mathcal{X} \times \mathcal{U}), j \in \mathbb{N}_{[0, N_p-1]}, \\ & x(k+N_p|k) \in \mathcal{X}_f. \end{aligned} \quad (4)$$

The quadratic stage cost  $\ell(x, u) = \|x - x_r\|_Q^2 + \|u - u_r\|_R^2$  can be defined in terms of regulation or tracking purposes (Limon et al., 2018). For regulation, it is implied that  $x_r$  and  $u_r$  are null. Furthermore,  $V(\cdot)$  is a terminal cost and  $\mathcal{X}_f$  is a terminal set (denoted terminal ingredients, see Sec. 4). The solution  $U_k^* = \text{col}\{u^*(k+j|k)\}$ ,  $\forall j \in \mathbb{N}_{[0, N_p-1]}$  provides the sequence of future control inputs inside the prediction horizon of  $N_p$  steps. The first entry  $u^*(k|k)$  is applied to the process.

*Remark 3.* The scheduling variables  $\rho$  are known at each sampling period  $k$ , although unknown for any future sample  $k+j$ ,  $j \in \mathbb{N}_{[1, N_p]}$ . This means that appropriate mechanisms must be included to the MPC optimisation for it to be solvable, as those detailed in Sec. 3.

### 2.3 Tracking

A comment must be made regarding reference tracking. The considered stage cost  $\ell(x, u)$  must be set in order to minimise the variations of  $(x, u)$  from the desired point target  $(x_r, u_r)$ , which implies in the output target  $y_r$ . Assume there exists a linear (parameter varying) combination of the states  $x$  and inputs  $u$  that ensures  $y(k) \rightarrow y_r$ . Following the lines of (Limon et al., 2018), one can use an offline reference optimisation selector, which determines the state tracking target  $z_r = [x_r \ u_r]^\top$ :

$$\begin{aligned} \min_{z_r} & \| (C(f_\rho(z_r)) \ D(f_\rho(z_r))) z_r - y_r \|^2, \\ \text{s.t.} & \begin{bmatrix} (\mathbb{I} - A(f_\rho(z_r))) & -B(f_\rho(z_r)) \\ C(f_\rho(z_r)) & D(f_\rho(z_r)) \end{bmatrix} z_r = \begin{bmatrix} 0_{n_x} \\ y_r \end{bmatrix}, \\ & f_\rho(z_r) \in \mathcal{P}, z_r^\top \in (\mathbf{X}_f \times \mathcal{U}), \end{aligned} \quad (5)$$

which ensures an admissible steady-state target  $z_r$  that imposes the output tracking objective.

This state reference selection problem can be solved online, at each sampling instant, if the output reference goal  $y_r$  is time-varying. Nonetheless, this adds computational complexity, as discusses (Köhler et al., 2020).

### 3. QLPV MPC ALGORITHMS

The MPC formulation in Eq. (4) requires the knowledge of the future values of the scheduling parameters  $\rho(k+j)$ , along the prediction horizon. A direct solution is to plug the nonlinear proxy  $\rho(k+j) = f_\rho(x(k+j))$  as a constraints of the optimisation procedure. Nonetheless, this converts the optimisation into an NP, which has the computational complexity of regular “full-blown” NMPCs.

Recent literature has shown the development of algorithms that overlap this issue by replacing the true values of  $\rho(k+j)$  by estimates  $\hat{\rho}(k+j)$ , as reviewed in (Morato et al., 2020a). By doing so, the NPs are converted into more efficient programs, with the complexity of SQPs (or even QPs, in some cases).

With regard to regular NMPC formulations, these novel qLPV embedding frameworks are attractive because the nonlinear state predictions and constraints are handled in an LPV fashion, which is cost-efficient (being linear on states and inputs). The numerical effectiveness of these methods is comparable to the fast real-time iterations NMPC algorithms, as demonstrated in (Cisneros and Werner, 2019).

Accordingly, we detail the basic concepts of two of these methods ((Cisneros and Werner, 2020) and (Morato et al., 2019)), discussing the implementation steps and the convergence property of these methods. Their advantages and drawbacks are further illustrated through realistic simulation essays (Sec. 5). The methods are:

- (1) The Sequential guessing technique (Sec. 3.1):  
This first method was derived in the original paper by (Cisneros et al., 2016). In that work, the evolution of the scheduling parameters is iteratively guessed based on the state prediction provided by the previous optimisation solution. The underlying mechanism is able to refine the predictions for the scheduling parameters based on the nonlinear proxy  $\rho := f_\rho(x)$ . Such method was extended and further formalised in (Cisneros and Werner, 2017), where the reference tracking problem was embedded in the formulation, with convergence proofs provided in (Cisneros and Werner, 2020). Tube-based MPC extensions, for the case of bounded additive uncertainties, have also been established (Hanema et al., 2017).
- (2) The Recursive extrapolation approach (Sec. 3.2):  
This second method was developed in (Morato et al., 2019, 2020b), replacing the previous iterative guessing

mechanism by recursive extrapolation procedures, which also provide  $\hat{\rho}(k+j), \forall j \in \mathbb{N}_{[0, N_p-1]}$ .

#### 3.1 Sequential qLPV MPC

##### Method Description

The core idea of the SQP method (Cisneros and Werner, 2020) is the following: the MPC is solved through iterative operations of the optimisation problem in Eq. (4), for which the qLPV process model is replaced by a “frozen” LTI model (different at each sampling instant). This prediction model is found by plugging the estimate for the sequence of scheduling parameters into Eq. (2). The future “scheduling sequence” guess is denoted  $\hat{P}_k := \text{col}\{\hat{\rho}(k+j)\}^\top, \forall j \in \mathbb{N}_{[0, N_p-1]}$ .

Note that if the actual scheduling sequence  $P_k$  was, in fact, known, the MPC would be able to ensure regulation through a single QP (solved at each sampling instant). Nevertheless, the method starts by choosing an initial guess for  $\hat{P}_k$ . Then, the internal state predictions of the MPC optimisation, denoted  $X_k := \text{col}\{x(k+j|k)\}^\top, \forall j \in \mathbb{N}_{[1, N_p]}$  are used to formulate the next guess for  $P_k$ , using the nonlinear scheduling proxy function over each entry of  $X_k$ , as follows:

$$\hat{P}_k = f_\rho \left( \overbrace{[x^\top(k), X_k^\top]}^{X_k^*} \right)^\top. \quad (6)$$

Then,  $\hat{P}_k$  is plugged into the MPC QP and the procedure is re-iterated, until  $\hat{P}_k \rightarrow P_k$

##### Advantages

The method guarantees this convergence within a relatively small number of iterations. The major advantage is that, at each iteration, the problem is formulated as a constrained QP, which can be tackled for many time-critical applications with modern solvers. The underlying QP of each iteration is based on the following prediction problem:

$$X_k = \mathcal{A}(\hat{P}_k)x(k) + \mathcal{B}(\hat{P}_k)U_k, \quad (7)$$

being  $\mathcal{A}(\hat{P}_k)$  and  $\mathcal{B}(\hat{P}_k)$  nonlinear matrices on the future scheduling parameters  $\rho(k+j|k), \forall j \in \mathbb{N}_{[0, N_p-1]}$ . These nonlinear matrices maintain the same form at each iteration and, thus, can be efficiently computed.

##### Limitations

The main limitation of this approach is that the internal loop may take several iterations (QPs) to converge. This is not desirable because the number of iterations needed for convergence may require more time than the available sampling period. In practice, this approach is not set to freely iterate until the convergence of  $\hat{P}_k$  to  $P_k$ . Therefore, a stop criterion is added to the mechanism so that iterations stop at a given threshold. A warm-start can also be included by shifting  $X_{k-1}$  and  $\hat{P}_{k-1}$  as the initial guesses for the optimisation at sampling  $k$ , which ensures that the proposed algorithm reaches convergence within few iterations.

### 3.2 Recursive qLPV MPC

#### Method Description

An alternative formulation to the previous concept has been proposed in papers (Morato et al., 2019, 2020b). While the method from (Cisneros et al., 2016) requires an SQP solution, since the MPC QP is iterated until  $\hat{P}_k$  converges, the proposition in these two papers is based on a single QP operation coupled to a recursive extrapolation method for  $\hat{P}_k$ . In this approach, the convergence of  $\hat{P}_k$  to  $P_k$  takes some samples to be achieved and is ensured as long as the MPC is recursively feasible.

The extrapolation mechanism is implemented through:

$$\begin{aligned}\hat{P}_k &= \Phi(\hat{P}_{k-1}, X_{k-1}^*) \\ &= \lambda \hat{P}_{k-1} + \sigma_k X_{k-1}^*,\end{aligned}\quad (8)$$

for which  $\lambda$  is a forgetting factor and  $\sigma_k$  is a time-varying gain. This factor can be derived by the solution of a least-square argument, imposing a time-varying auto-regressive model for each entry of  $\hat{P}_k$ , or via Taylor expansion, taken as  $\frac{df_\rho(X)}{dX}|_{X_k^*}$ , details are given in (Morato et al., 2020b).

#### Advantages

This approach does not require us to evaluate online  $f_\rho(X_k^*)$ , which can be numerically expensive. Therefore, the online computational burden is that of a QP, which can be solved very fast by standard solvers.

#### Limitations

Nevertheless, the convergence of  $\hat{P}_k \rightarrow P_k$  is not achieved within a single sampling period. Therefore, there appears an inherent discrepancy between  $\hat{P}_k$  and  $P_k$  during the first samples, which vanishes over time. Accordingly, during these initial steps, the solution of the MPC problem is sub-optimal, which may deteriorate performances.

### 3.3 Implementation

Both these previous methods can be implemented through Algorithm 1. The application departs from an initial state sequence  $X_0$  and an initial scheduling sequence  $P_0$ . These vectors can be simply taken as  $N_p$  repeated instances of  $x(0)$  and  $\rho(0)$ . The implementation also depends on a known terminal set condition  $\mathbf{X}_f$ , which will be detailed in Sec. 4, and a target reference goal  $z_r$ . For regulation purposes,  $z_r$  is null, while for tracking purposes, it comes from the offline solution of (5).

### 3.4 Convergence Property

In order to demonstrate the convergence of these methods, under the form of Algorithm 1, we invoke the well-known result for the convergence of Newton SQPs from (Izmailov and Solodov, 2011), which implies that a quadratic sub-problem program of SQP algorithms can be derived by a second-order approximation of the SQP optimisation cost and linearisation of its constraints.

Therefore, under the assumptions previous exhibited, the solution of Algorithm 1 is equivalent to that of a quadratic sub-problem in standard Newton SQP form, from which local convergence property can be readily found. Consider the following generic NP:

---

### Algorithm 1 qLPV MPC from Secs. 3.1 and 3.2

---

**Initialise:**  $x(0) = x_0, \rho(0) = \rho_0, k = 0$ .

**Require:**  $Q, R, N_p, z_r$ .

**Require:**  $P_0, X_0, U_0$ .

**Loop:**

- Step (1):
  - (A) (Cisneros and Werner, 2020): **Loop until convergence:**
    - (i) Shift and update  $X_k \implies X_k^*$ ;
    - (ii) Based on  $\hat{P}_{k-1}$ , compute the LTI predictions with Eq. (7);
    - (iii) Solve the optimisation in Eq. (4);
    - (iv) Compute  $\hat{P}_k = f_\rho(X_k^*)$ ;
  - (B) (Morato et al., 2019): Solve  $\hat{P}_k = \Phi(\hat{P}_{k-1}, X_{k-1}^*)$  and compute the LTI predictions with Eq. (7);
- Step (2): Solve the optimisation in Eq. (4) with predictions from Eq. (7);
- Step (3): Take  $u(k) = u^*(k|k)$  and apply this local control to the process;
- Step (4):  $k \leftarrow k + 1$ .

**end**

---

$$\begin{cases} \min_{x_c} & J_c(x_c) \\ \text{s.t.} & h_j(x_c) = 0 \\ & g_i(x_c) \leq 0 \end{cases},$$

This optimisation problem has an equivalent quadratic sub-problem in the form of:

$$\begin{aligned}\min_{\check{x}_c} & \left( \check{x}_c^\top H_{J_c}(\bar{x}_c) \check{x}_c + (\nabla J_c(x_c)|_{x_c=\bar{x}_c})^\top \check{x}_c \right) \\ \text{s.t.} & (\nabla h_j(x_c)|_{x_c=\bar{x}_c}) \check{x}_c + (\nabla h_j(x_c)|_{x_c=\bar{x}_c}) = 0 \\ & (\nabla g_i(x_c)|_{x_c=\bar{x}_c}) \check{x}_c + (\nabla g_i(x_c)|_{x_c=\bar{x}_c}) \leq 0\end{aligned}\quad (9)$$

where  $H_{J_c}(x_c)$  denotes the Hessian of the optimisation cost  $J_c(x_c)$  and  $\nabla h_j(x_c)$  and  $\nabla g_i(x_c)$  denote divergent operators. This sub-problem is evaluated at a given solution estimate  $\bar{x}_c$ , for which  $\check{x}_c = x_c - \bar{x}_c$ .

If LDI is used to provide the qLPV model in Eq. (2), then the iterations of Algorithm 1 are in equivalence to a Newton SQP sub-problem. The sub-problem in Eq. (9) is identical to either the optimisation given through the consecutive iterations Eq. (4) (Cisneros and Werner, 2020) and the recursive operator of Eq. (8) with the solution of Eq. (4) (Morato et al., 2019).

It follows that if local convergence of the equivalent Newton SQP can be established, the convergence of Algorithm 1 is also ensured. The sufficient conditions for this property are equivalent to those detailed in (Houska et al., 2011), which refer to the form of the quadratic cost  $\ell(\cdot)$  and the terminal ingredients ( $V(\cdot)$  and  $\mathbf{X}_f$ ).

## 4. STABILITY AND OFFLINE PREPARATIONS

Next, we briefly detail how to construct the terminal ingredients for Algorithm 1, in order to ensure the stability of the closed-loop dynamics and recursive feasibility of the optimisation.

The usual approach with terminal ingredients resides in satisfying some conditions with the terminal set  $\mathbf{X}_f$  and

the terminal cost  $V(x(k+N_p|k))$ , w.r.t. a nominal feedback  $u = K(x - x_r)$ . For notation lightness, we proceed with  $x_r$  null<sup>1</sup>. Consider that there exists a terminal state-feedback gain  $K(\rho), \forall \rho \in \mathcal{P}$ , and a terminal set  $\mathbf{X}_f := \{x | x^\top P(\rho)x \leq \alpha_P\}$ . This is a centered ellipsoidal set with a radius of  $\alpha_P$ . The terminal cost is a corresponding sub-level set:  $V(x, \rho) := x^\top P(\rho)x$ . Under regular  $\mathcal{K}$ -class properties on  $\ell(\cdot)$  (lower bounded) and  $V(\cdot)$  (upper bounded), the following Theorem gives the sufficient conditions for closed-loop stability and recursive feasibility:

*Theorem 1.* Consider that the MPC is given by Eq. (4) under a feedback  $u = K(\rho)x$ , with a terminal state set given by  $\mathbf{X}_f(\rho)$  and a terminal cost  $V(x, \rho)$ . Assume that the initial solution  $U_k^*$  is feasible. Then, input-to-state stability is ensured and the optimisation remains recursively feasible if the following conditions hold  $\forall \rho \in \mathcal{P}$ :

- (C1) The origin lies in the interior of  $\mathbf{X}_f$ ;
- (C2)  $x^+ := (A(\rho) + B(\rho)K(\rho))x$  lies within  $\mathbf{X}_f, \forall x \in \mathbf{X}_f$ ;
- (C3) The following Lyapunov equation is verified  $\forall x \in \mathbf{X}_f, \forall \rho \in \mathcal{P}$  and  $\forall \partial\rho \in \partial\mathcal{P}$ :  $V(x^+, \rho + \partial\rho) - V(x, \rho) \leq -x^\top Qx - x^\top K^\top(\rho)RK(\rho)x$ .
- (C4) The image of the nominal feedback lies within the admissible control domain:  $K(\rho)x \in \mathcal{U}, \forall \rho \in \mathcal{P}$ .
- (C5) The terminal set  $\mathbf{X}_f(\rho)$  is a subset of  $\mathcal{X}$ .

The proof of Theorem 1 is standard (Mayne et al., 2000). Accordingly, the following Theorem provides parameter-dependent terminal ingredients which verify Theorem 1.

*Theorem 2.* The conditions (C1)-(C5) of Theorem 1 are satisfied if there exists a symmetric parameter-dependent positive definite matrix  $P(\rho) : \mathbb{R}^{n_p} \rightarrow \mathbb{R}^{n_x \times n_x}$ , a parameter-dependent rectangular matrix  $W(\rho) : \mathbb{R}^{n_p} \rightarrow \mathbb{R}^{n_u \times n_x}$ , and a scalar  $0 < \alpha \in \mathbb{R}$  such that  $Y(\rho) = (P(\rho))^{-1} > 0$ ,  $W(\rho) = K(\rho)Y(\rho)$  and that LMIs (10)-(12) hold for all  $\rho \in \mathcal{P}$  and  $\partial\rho \in \partial\mathcal{P}$ , under the minimisation of  $\alpha$ .

$$\left[ \begin{array}{cc|cc} Y(\rho) & & \star & \star \\ (A(\rho)Y(\rho) + B(\rho)W(\rho))Y(\rho + \partial\rho) & & \star & \star \\ \hline Y(\rho) & 0 & Q^{-1} & \star \\ W(\rho) & 0 & 0 & R^{-1} \end{array} \right] \geq 0 \quad (10)$$

$$\left[ \begin{array}{c|c} \alpha \bar{u}_i^2 & I_{\{i\}}W(\rho) \\ \hline \star & Y(\rho) \end{array} \right] \geq 0, i \in \mathbb{N}_{[1, n_u]}, \quad (11)$$

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

The proof of Theorem 2 follows from (Cisneros and Werner, 2020), using  $\alpha = \frac{1}{\alpha_P}$ . This Theorem ensures a positive definite parameter-dependent matrix  $Y(\rho) = P^{-1}(\rho)$ , which gives the terminal ingredients  $V(\cdot)$  and  $\mathbf{X}_f$ . Theorem 2 provides infinite-dimensional inequalities, which must hold  $\forall \rho \in \mathcal{P}$  and  $\forall \partial\rho \in \partial\mathcal{P}$ . In practice, the solution can be found by enforcing the inequalities over a sufficiently dense grid of points  $(\rho, \partial\rho)$  along the  $\mathcal{P} \times \partial\mathcal{P}$  plane. Then, the solution can be verified over a denser grid. The parameter-dependency of  $P$  may be dropped if the system is quadratically stabilizable, but this may result in further conservativeness.

<sup>1</sup> The tracking equivalency is easily done with  $\frac{dx_r}{dt} = 0$  and by computing the qLPV model with shifted states dynamics  $(x - x_r)$ .

*Remark 4.* The solution of Theorem 2 is a parameter-dependent map  $Y(\rho) = \sum_{j=1}^{n_p} \rho_j Y_j$ . The online procedure, nonetheless, depends on an inversion of  $Y(\rho)$  in order to compute the terminal ingredients  $V$  and  $\mathcal{X}_f$ , at each sampling instant. We note that  $K(\rho)$  is a fictive feedback gain used only to prove the stability conditions; the actual feedback is that determined by the online optimisation procedure.

## 5. APPLICATION RESULTS

In this Sec., we provide two different application results of the considered methods from (Cisneros and Werner, 2020) and (Morato et al., 2019), which are henceforth denoted Sequential qLPV MPC and Recursive qLPV MPC, respectively. The following results are obtained using MATLAB, Yalmip, and SDPT3 and Gurobi solvers, using high-fidelity nonlinear models of the processes.

### 5.1 Regulation: Semi-active Suspension System

In this first case study, we consider the regulation problem of a semi-active suspension system. Controllable suspensions are used to mitigate the vertical oscillations in vehicles when road bumps are encountered, thus improving driving comfort. Among these, semi-active suspensions are most interesting since they provide better performances than passive suspensions while being less energy-consuming than active ones (Poussot-Vassal et al., 2012).

Consider the corner dynamics of a vehicle equipped with an electro-rheological (ER) semi-active suspension system, as detailed in (Morato et al., 2020b). A suspension system connects the vehicle body ( $m_s$ ) and the wheel link ( $m_{us}$ ) with a spring and a controllable damper. Let  $z_s$  and  $z_{us}$  denote the vertical displacement of these two vehicle parts, respectively. The vertical road disturbance is denoted  $z_r$ , while the ER damper force is denoted  $F_{ER}$ . We use  $z_d = z_s - z_{us}$  and  $z_w = z_{us} - z_r$ . Then, using Newton's second law of motion, and a hyperbolic tangent damping force model, we find the following nonlinear model:

$$\ddot{z}_s(t) = -k_s z_d(t) - (c_0 + c_u u(t)) \dot{z}_d(t) - F_{ER}(t),$$

$$\ddot{z}_{us}(t) = +k z_d(t) + (c_0 + c_u u(t)) \dot{z}_d(t) + F_{ER}(t) - k_t z_w(t),$$

$$\tau \dot{F}_{ER}(t) = -F_{ER}(t) + f_c \tanh(k_1 z_d(t) + c_1 \dot{z}_d(t)) u(t).$$

Take the following discrete-time states:

$$x(k) := [z_d(k), \dot{z}_s(k), z_w(k), \dot{z}_{us}(k), F_{ER}(k)]^\top$$

. The system operates with a sampling period of  $T_s = 5$  ms. The qLPV model matrices are obtained through an Euler approximation:

$$A(\rho) = \mathbb{I}_{n_x} + T_s \begin{bmatrix} 0 & 1 & 0 & -1 & 0 \\ -\frac{k_s}{m_s} & -\frac{(c_0 + c_u \rho_1)}{m_s} & 0 & \frac{(c_0 + c_u \rho_1)}{m_s} & -\frac{1}{m_s} \\ 0 & 0 & 0 & 1 & 0 \\ \frac{k_s}{m_{us}} & \frac{(c_0 + c_u \rho_1)}{m_{us}} & -\frac{k_t}{m_{us}} & -\frac{(c_0 + c_u \rho_1)}{m_{us}} & \frac{1}{m_{us}} \\ 0 & 0 & 0 & 0 & \frac{-1}{\tau} \end{bmatrix},$$

$$B(\rho) = T_s [0 \ 0 \ 0 \ 0 \ \frac{f_c}{\tau} \rho_2]^\top, \quad B_2 = T_s [0 \ 0 \ -1 \ 0 \ 0]^\top,$$

which corresponds to the qLPV embedded model, in the form of Eq. (2) with an additional term  $+B_2 \dot{z}_r(k)$  on the state dynamics. The scheduling parameters are  $\rho :=$

Table 1. Suspension System Characteristics.

| Parameter | Description                                | Value  | Unit  |
|-----------|--|--------|-------|
| $m_s$     | Chassis body mass                          | 2.27   | kg    |
| $m_{us}$  | Wheel link mass                            | 0.25   | kg    |
| $k_s$     | Spring stiffness                           | 1396   | N/m   |
| $k_t$     | Tire stiffness                             | 12270  | N/m   |
| $k_0$     | Passive damper stiffness coefficient       | 170.4  | N/m   |
| $k_1$     | Hysteresis coefficient due to displacement | 218.16 | N.s/m |
| $c_0$     | Viscous damping coefficient                | 63.98  | N.s/m |
| $c_u$     | Viscous damping coefficient                | 11.84  | N.s/m |
| $c_1$     | Hysteresis coefficient due to velocity     | 21     | N.s/m |
| $f_c$     | Dynamic yield force of ER fluid            | 28.07  | N     |
| $\tau$    | Time constant                              | 42     | ms    |

| Variable | Description                | Lower bound | Upper bound | Unit |
|----------|----------------------------|-------------|-------------|------|
| $x_1$    | Suspension deflection      | -0.15       | 0.15        | m    |
| $x_2$    | Chassis body velocity      | -3          | 3           | m/s  |
| $x_3$    | Wheel deflection           | -0.05       | 0.05        | m    |
| $x_4$    | Wheel link velocity        | -3          | 3           | m/s  |
| $x_5$    | Controlled ER damper force | -30         | 30          | N    |
| $u$      | Control input (PWM)        | 0           | 1           | -    |

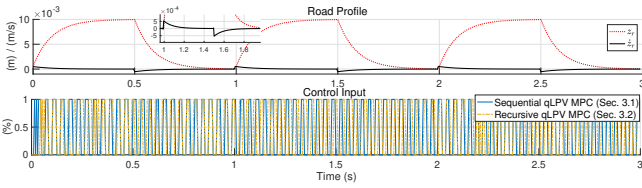


Fig. 1. Road scenario and control input.

$[\rho_1 \ \rho_2]^\top$ ,  $\rho_1 := u \in [0, 1]$ , and  $\rho_2 := \tanh(\Gamma x) \in [-1, 1]$ , with  $\Gamma = [k_1 \ c_1 \ 0 \ -c_1 \ 0]$ . The parameter values and constraints are those from a 1/5-scale vehicle testbed<sup>2</sup>, as gives Table 1.

As previously discussed, the main challenge of deploying NMPC through qLPV models lies in the fact that the values of the scheduling parameters  $\rho$  are unknown throughout the prediction horizon. Therefore, we apply and compare the previously surveyed techniques. The Sequential qLPV MPC method operates with a threshold of  $N_{iter} = 5$  iterations of the internal loop (see Algorithm 1, Step (1)), while the Recursive qLPV MPC operates with  $\lambda = 1$  and  $\sigma_k = 0$  in Eq. (8), for simplicity. This implies a higher degree of conservatism with this latter method since we basically consider that the scheduling parameters remain constant throughout the horizon for prediction purposes.

Consider a simulation road scenario of a sequence of three 10 mm vertical bumps. The NMPCs are tuned with a prediction horizon of 20 steps, using unitary weights ( $Q = \mathbb{I}_{n_x}$ ,  $R = \mathbb{I}_{n_u}$ ). Considering an initial condition of  $x_0 = [0.001, -0.001, 0.001, 0.001, 0]^\top$ , results are obtained in a 2.4 GHz, 8 GB RAM Macintosh computer, using a high-fidelity noisy nonlinear suspension system model (Morato et al., 2019).

Firstly, Fig. 1 provides the road scenario (unmeasured load disturbances) and the corresponding control law, a PWM signal that regulates the damper force. Accordingly, Fig. 2 presents some state trajectories, for brevity, considering both surveyed methods. Clearly, regulation is obtained with both techniques, and the road bumps are rejected in steady-state.

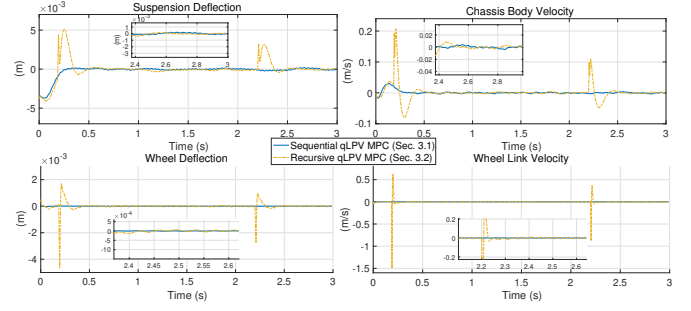


Fig. 2. Suspension Sys.: State trajectories.

To further evaluate these NMPC methods, we assess the obtained closed-loop trajectories in terms of performance indexes. In Tab. 2, we provide the RMS values for the MPC stage cost  $\ell(x, u)$ , along the whole simulation. As a result, we can conclude that both strategies are efficient for the regulation purpose, while the performances obtained with the Sequential method are better than those with the Recursive tool (16.5% average performance enhancement). This is mainly because the recursive extrapolation leads to model-process mismatches during the first samples after each bump, which means that the NMPC is sub-optimal while  $\hat{P}_k \neq P_k$ . We stress that the Sequential method ensures the convergence of the scheduling sequence even with the  $N_{iter}$  threshold. Nonetheless, we note that the average online computational stress ( $t_c$  index) with the recursive solution is smaller than the one with the SQPs (less than 10 times smaller) since the nominal predictions are linear at each sampling instant and only one QP is solved per sample, while at least  $N_{iter}$  QPs are solved with the Sequential scheme.

## 5.2 Tracking: Quadruple-Tank Process

The second case study for which we apply these novel NMPC techniques is an academic example of a quadruple-Tank process, from (Johansson, 2000). This system consists of four interconnected tanks, as illustrated in Figure 3, for which two pumps, along with two valves, regulate the flow of water that circulates the system. The nonlinear level dynamics are:

$$\begin{aligned} \dot{h}_1(t) &= -\frac{a_1}{A_1} \sqrt{2gh_1(t)} + \frac{a_3}{A_1} \sqrt{2gh_3(t)} + \frac{\gamma_1(t)k_1}{A_1} u_1(t), \\ \dot{h}_2(t) &= -\frac{a_2}{A_2} \sqrt{2gh_2(t)} + \frac{a_4}{A_2} \sqrt{2gh_4(t)} + \frac{\gamma_2(t)k_2}{A_2} u_2(t), \\ \dot{h}_3(t) &= -\frac{a_3}{A_3} \sqrt{2gh_3(t)} + \frac{(1 - \gamma_2(t))k_2}{A_3} u_2(t), \\ \dot{h}_4(t) &= -\frac{a_4}{A_4} \sqrt{2gh_4(t)} + \frac{(1 - \gamma_1(t))k_1}{A_4} u_1(t). \end{aligned}$$

Each  $h_i(t)$  represents the water level at the  $i$ -th tank;  $u_j$  represents the power of the  $j$ -th pump, for which the corresponding flow is  $k_j u_j(t)$ , and each  $\gamma_j$  gives opening percentage of the  $j$ -th valve, directing more/less flow to the upper/lower tanks. We assume that all levels are measured. The tank cross sections  $A_i$  are of  $1 \text{ cm}^2$ , while the outlet hole cross sections  $a_i$  are of  $0.05 \text{ cm}^2$ . The pump parameters  $k_j$  are of  $1.4 \text{ cm}^3/\text{Vs}$ . The process constraints are:  $h_j \in [0.1, 10], \forall j \in \mathbb{N}_{[1,2]}$  (level, process variable);

<sup>2</sup> Refer to <http://www.gipsa-lab.fr/projet/inove/>.

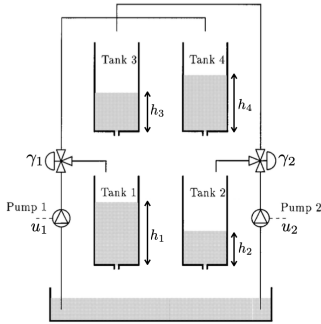


Fig. 3. 4-Tank Process. Fig. from (Johansson, 2000).

$u_j \in [0.05, 5] \text{ V}, \forall j \in \mathbb{N}_{[1,2]}$  (pump flow, control input);  
 $\gamma_j \in [1, 100] \%$ ,  $\forall j \in \mathbb{N}_{[1,2]}$  (valve opening).

Considering a sampling period of  $T_s = 250 \text{ ms}$ , the exact realisation of the nonlinear dynamics with the qLPV embedding model in Eq. (2) is obtained with:

$$A(\rho) = \mathbb{I}_{n_x} + T_s \begin{bmatrix} \frac{-a_1 \sqrt{2g\rho_1}}{A_1} & 0 & \frac{a_3 \sqrt{2g\rho_3}}{A_1} & 0 \\ 0 & \frac{-a_2 \sqrt{2g\rho_2}}{A_2} & 0 & \frac{a_4 \sqrt{2g\rho_4}}{A_2} \\ 0 & 0 & \frac{-a_3 \sqrt{2g\rho_3}}{A_3} & 0 \\ 0 & 0 & 0 & \frac{-a_4 \sqrt{2g\rho_4}}{A_4} \end{bmatrix},$$

$$B = T_s \begin{bmatrix} \frac{\gamma_1 k_1}{A_1} & 0 & 0 & \frac{(1-\gamma_1)k_1}{A_4} \\ 0 & \frac{\gamma_2 k_2}{A_2} & \frac{(1-\gamma_2)k_2}{A_3} & 0 \end{bmatrix}^T.$$

The scheduling parameters  $\rho = \text{col}\{\rho_j\}$ , with  $\rho_j(k) := (h_j(k))^{-\frac{1}{2}} \in [0.31, 3.16]$ ,  $\forall j \in \mathbb{N}_{[1,4]}$ . We note that the input matrix  $B$  is scheduled by the valve opening signals  $\gamma_j$ , which are known from the control viewpoint.

The application of the qLPV MPC methods is done using the same values for  $N_{iter}$ ,  $\lambda$  and  $\sigma$  from the previous essay. In this study, we add a stop criterion to the SQP method, as suggests (Cisneros and Werner, 2020): the internal loop stops if the infinity-norm of difference w.r.t.  $P_k$  between internal iterations is smaller than  $10^{-3}$ .

In this second essay, a tracking objective is used:  $\ell(x, u) := \|(x - x_r)\|_Q^2 + \|(u - u_r)\|_R^2$ , being  $(x_r, u_r)$  admissible time-varying reference signals (that satisfy Eq. (5)). This tracking objective concerns the first and second level signals. We use a prediction horizon of  $N_p = 10$  samples and weighting matrices  $Q = \text{diag}([10 \ 10 \ 0 \ 0])$ , and  $\mathbb{I}_{n_u}$ .

The following simulation scenario is considered: the initial condition is  $x(0) = [5 \ 5 \ 5 \ 5]^T$ ,  $u(0) = [5\sqrt{0.5} \ 5\sqrt{0.5}]$ , and  $\gamma(0) = [1 \ 1]^T$ . Ten-second-long piecewise-constant reference signals are used. Simulations are carried out with normalised variables.

Fig. 4 presents the tracking performances for the two level signals. Evidently, both qLPV MPC methods are able to steer the states to the admissible targets while respecting constraints on  $x$  and  $u$ . Fig. 5 shows the corresponding control inputs. All state trajectories converge in finite time with small overshoots and no oscillations. Both performances are numerically equivalent. This similarity resides in the Sequential method requiring no more than one or two iterations of the internal loop in over 70%

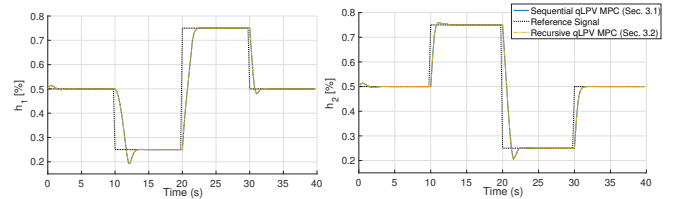


Fig. 4. 4-Tank Sys.: Reference Tracking.

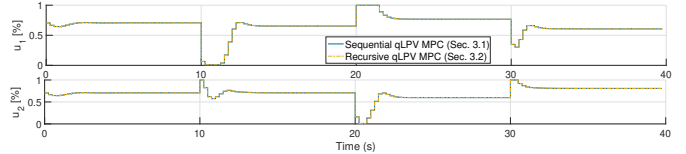


Fig. 5. 4-Tank Sys.: Control Signal.

of the sampling instants. Nevertheless, we must stress that the Recursive approach is over 2.5 times faster<sup>3</sup> (on average) than the Sequential technique to compute the control action, as gives Table 2.

Table 2. Performance Results

| Regulation            | $r\{\ell(\cdot)\}$ | $t_c$           |
|-----------------------|--------------------|-----------------|
| Sequential (Sec. 3.1) | <b>4.548</b>       | 3.795 ms        |
| Recursive (Sec. 3.2)  | 5.445              | <b>0.377 ms</b> |
| Tracking              | $r\{\ell(\cdot)\}$ | $t_c$           |
| Sequential (Sec. 3.1) | <b>0.102</b>       | 49.40 ms        |
| Recursive (Sec. 3.2)  | <b>0.102</b>       | <b>18.90 ms</b> |

### 5.3 Discussion

Based on these two realistic essays, we can conclude that both qLPV MPC methods are interesting options for the predictive control of nonlinear processes without the need for NP-hard solutions.

Both methods are conceived with estimates for the future behaviour of the scheduling variable. The SQP technique yields overall better estimates for all sampling instants, while the recursive scheme takes some samples for these estimates to converge, which may lead to some performance deterioration.

Nevertheless, we must debate that this performance deterioration may be tolerable since the method evaluates the control law significantly faster. The average online computational stress ( $t_c$ ) with the recursive solution is, in both cases, much smaller than the one with the SQPs. The approach from (Cisneros and Werner, 2020) requires the solution of (at most)  $N_{iter}$  QPs coupled with the nonlinear vector-wise operation of Eq. (6), which may impede real-time applications with very ultra-fast sampling rates. Accordingly, the Recursive method is more flexible for such fast systems. Robustness w.r.t. the (bounded) model-process mismatches that occur while  $\hat{P}_k \neq P_k$  in the recursive scheme can be ensured through the terminal set  $\mathbf{X}_f$ , as in standard bounded disturbance rejection problems.

## 6. CONCLUSION

In this paper, we revisited some novel NMPC formulations based on qLPV embedding. These methods provide

<sup>3</sup> In a 2 GHz, 4 GB RAM PC.



real-time NMPC solutions since the qLPV realisation offers linear predictions at each sampling period. To this matter, two different methods are detailed; both solve the NMPC problem by estimating the future behaviour of the scheduling variables. The first iterates the MPC optimisation multiple times, using the state predictions to compute the scheduling sequence; the second uses a recursive extrapolation mechanism to guess the evolution of these variables along the horizon. We thoroughly detail the implementation of these techniques, considering both regulation and tracking control objectives. We also discuss tracking target selection, qLPV terminal ingredients, and convergence properties. For illustration purposes, we provide simulation results of a semi-active suspension system (regulation) and a quadruple-tank process (tracking). As evidenced, good performances are obtained with relatively small numerical stress. Evidently, the qLPV MPC framework offers comparable qualities to modern solver-based NMPC solutions, such as ACADO and GRAMPC. The main advantage is that no real-time iteration/Lagrangian mappings must be applied since only the solution of QPs is required, which is tackled by most standard solvers. The future promise of NMPC through qLPV embedding is vast since only a handful of papers have investigated this topic. Formal comparisons to ACADO and GRAMPC, for instance, are still lacking, as well as experimental validation of some of the topics herein discussed. We stress that the tracking of time-varying reference is still not thoroughly established for this framework since the requirement of an online steady-state selector may drastically diminish numerical performances.

## REFERENCES

- Abbas, H.S., Toth, R., Petreczky, M., Meskin, N., and Mohammadpour, J. (2014). Embedding of Nonlinear Systems in a Linear Parameter-varying Representation. *IFAC Proceedings Volumes*, 47(3), 6907–6913.
- Allgöwer, F. and Zheng, A. (2012). *Nonlinear Model Predictive Control*, volume 26. Birkhäuser.
- Cisneros, P.G. and Werner, H. (2017). Fast Nonlinear MPC for Reference Tracking Subject to Nonlinear Constraints via quasi-LPV Representations. *IFAC-PapersOnLine*, 50(1), 11601–11606.
- Cisneros, P.S.G. and Werner, H. (2020). Nonlinear Model Predictive Control for Models in Quasi-Linear Parameter Varying Form. *International Journal of Robust and Nonlinear Control*, 30(10), 3945–3959.
- Cisneros, P.S., Voss, S., and Werner, H. (2016). Efficient Nonlinear Model Predictive Control via quasi-LPV Representation. In *IEEE Conference on Decision and Control*, 3216–3221. IEEE.
- Cisneros, P.S. and Werner, H. (2019). Wide Range Stabilization of a Pendubot using quasi-LPV Predictive Control. *IFAC-PapersOnLine*, 52(28), 164–169.
- Englert, T., Völz, A., Mesmer, F., Rhein, S., and Graichen, K. (2019). A Software Framework for Embedded Nonlinear Model Predictive Control using a Gradient-based Augmented Lagrangian Approach (GRAMPC). *Optimization and Engineering*, 20(3), 769–809.
- Gros, S., Zanon, M., Quirynen, R., Bemporad, A., and Diehl, M. (2020). From Linear to Nonlinear MPC: Bridging the Gap via the Real-time Iteration. *International Journal of Control*, 93(1), 62–80.
- Hanema, J., Tóth, R., and Lazar, M. (2017). Stabilizing Non-linear MPC using Linear Parameter-varying Representations. In *56<sup>th</sup> Annual Conference on Decision and Control*, 3582–3587. IEEE.
- Houska, B., Ferreau, H.J., and Diehl, M. (2011). An Auto-generated Real-time Iteration Algorithm for Nonlinear MPC in the Microsecond Range. *Automatica*, 47(10), 2279–2285.
- Izmailov, A.F. and Solodov, M.V. (2011). On Attraction of Linearly Constrained Lagrangian Methods and of Stabilized and quasi-Newton SQP Methods to Critical Multipliers. *Mathematical Programming*, 126(2), 231–257.
- Johansson, K.H. (2000). The Quadruple-tank Process: A Multivariable Laboratory Process with an Adjustable Zero. *IEEE Transactions on Control Systems Technology*, 8(3), 456–465.
- Köhler, J., Müller, M.A., and Allgöwer, F. (2020). A Nonlinear Tracking Model Predictive Control Scheme for Dynamic Target Signals. *Automatica*, 118, 109030.
- Limon, D., Ferramosca, A., Alvarado, I., and Alamo, T. (2018). Nonlinear MPC for Tracking Piece-wise Constant Reference Signals. *IEEE Transactions on Automatic Control*, 63(11), 3735–3750.
- Mayne, D.Q., Rawlings, J.B., Rao, C.V., and Sokaert, P.O. (2000). Constrained Model Predictive Control: Stability and Optimality. *Automatica*, 36(6), 789–814.
- Morato, M.M., Normey-Rico, J.E., and Sename, O. (2020a). Model Predictive Control Design for Linear Parameter Varying Systems: A Survey. *Annual Reviews in Control*, 49, 64 – 80.
- Morato, M.M., Normey-Rico, J.E., and Sename, O. (2019). Novel qLPV MPC Design with Least-Squares Scheduling Prediction. *IFAC-PapersOnLine*, 52(28), 158–163.
- Morato, M.M., Normey-Rico, J.E., and Sename, O. (2020b). Sub-optimal Recursively Feasible Linear Parameter-Varying Predictive Algorithm for Semi-active Suspension Control. *IET Control Theory & Applications*, 14(18), 2764–2775.
- Poussot-Vassal, C., Spelta, C., Sename, O., Savaresi, S.M., and Dugard, L. (2012). Survey and Performance Evaluation on some Automotive Semi-active Suspension Control Methods: A Comparative Study on a Single-corner Model. *Annual Reviews in Control*, 36(1), 148–160.
- Quirynen, R., Vukov, M., Zanon, M., and Diehl, M. (2015). Autogenerating Microsecond Solvers for Nonlinear MPC: A Tutorial using ACADO Integrators. *Optimal Control Applications and Methods*, 36(5), 685–704.
- Rathai, K.M.M. (2020). *Synthesis and Real-time Implementation of Parameterized NMPC Schemes for Automotive Semi-active Suspension Systems*. Ph.D. thesis, Grenoble INP, Communauté Université Grenoble Alpes, Grenoble, France.
- Sala, A., Ariño, C., and Robles, R. (2020). Set-based Gain-scheduled Control via Quasi-convex Difference Inclusions. *International Journal of Robust and Nonlinear Control*, 1–23. doi:https://doi.org/10.1002/rnc.5180.
- Sename, O., Gaspar, P., and Bokor, J. (2013). *Robust Control and Linear Parameter Varying Approaches: Application to Vehicle Dynamics*, volume 437. Springer.