



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. NMPC 2021 - 7th IFAC Conference on Nonlinear Model Predictive Control, Jul 2021, Bratislava, Slovakia. 10.1016/j.ifacol.2021.08.561 . hal-03210253v2

HAL Id: hal-03210253

<https://hal.science/hal-03210253v2>

Submitted on 30 Apr 2021

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 [★]

Marcelo M. Morato^{*,**} Gia Quoc Bao Tran^{**}
Guilherme N. G. dos Reis^{*} Julio E. Normey-Rico^{*}
Olivier Sename^{**}

^{*} *Dept. de Automação e Sistemas (DAS), Univ. Fed. de Santa Catarina, Florianópolis-SC, Brazil (marcelomnzm@gmail.com)*

^{**} *Univ. Grenoble Alpes, CNRS, Grenoble INP[†], GIPSA-lab, Grenoble, France [†]Institute of Engineering Univ. Grenoble Alpes*

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 strategies generate online programs with reduced numerical burden, much faster to solve than the Nonlinear Programs generated with “regular” NMPC. The general lines of these methods: (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 incorporated by the 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 provided to illustrate their performances.

Keywords: Nonlinear Model Predictive Control, Linear Parameter Varying Systems, Tutorial.

1. INTRODUCTION

Model Predictive Control (MPC) is very well recognised for the purposes of regulation and reference tracking in constrained processes. Anyhow, nonlinear MPC (NMPC) is not trivial and comes with an increased numerical burden, complicating real-time implementation. This was a very substantial issue until the late 10’s, since even the most efficient NMPC algorithms displayed exponential complexity grow w.r.t. system size.

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): real-time iterations (Quirynen et al., 2015), gradient use (Englert et al., 2019), and input parameterisation (Rathai, 2020).

In parallel to this progress, the Linear Parameter Varying (LPV) toolkit has been widely popularised (Sename et al., 2013). For many nonlinear systems, quasi-LPV (qLPV) embeddings offer exact realisations with elegance and simplicity since there are no nonlinear state transitions but rather linear maps scheduled by (known, bounded) parameters ρ . Accordingly, recent advances have provided NMPC schemes via qLPV embedding, see (Morato et al., 2020a) and references therein. These algorithms enable real-time operation because the “full-blown” nonlinear predictions are replaced by a sequence of linear ones, solved through Sequential Quadratic Programs (SQPs).

The notion of deploying qLPV-embedding NMPC is very recent. Therefore, in this brief tutorial paper, we discuss this concept, detailing the implementation steps and illustrating their application to different processes. We stress that this paper significantly differs from (Morato et al., 2020a), which surveys all kinds of LPV MPC algorithms (robust, sub-optimal, etc.). The focus herein is only set upon the most recent algorithms, which use embeddings to fasten the resulting NPs.

The rest of this paper is organised as follows. In Sec. 2, we provide the qLPV NMPC problem setup, as well as the underlying assumptions required for correct implementation. In Sec. 3, we detail two different mechanisms for the estimation of the future scheduling parameters. Section 4 provides terminal ingredients that ensure stability and recursive feasibility. In Sec. 5, we provide two different applications: the regulation goal of a semi-active suspension system and the level tracking problem in a quadruple-tank process. Finally, 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. Consider a controlled equilibrium point, i.e., 0 is s.t. there exists a $\bar{u} \in \mathcal{U}$ s.t. $f(0, \bar{u}) = 0$. The suitable operation is defined by: $x \in \mathcal{X}$

[★] This work has been supported by *CNPq* (304032/2019 – 0), *CAPES* (001) and *ITEA3* European project (15016) *EMPHYSIS*.

and $u \in \mathcal{U}$, with $\mathcal{X} := \{x_j \in \mathbb{R} : \|x_j\|_2 \leq \bar{x}_j, \forall j \in \mathbb{N}_{[1, n_x]}\}$, and $\mathcal{U} := \{u_i \in \mathbb{R} : \|u_i\|_2 \leq \bar{u}_i, \forall i \in \mathbb{N}_{[1, n_u]}\}$. The states x are measured, thus we use the state-feedback $u(k) := \kappa(k)x(k)$ to ensure that the dynamics adheres 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 (x_r, u_r)).

2.1 qLPV Embedding

Nonlinear systems can be described with an exact qLPV realisation if the Linear Differential Inclusion (LDI) property is verified: Suppose that $\exists H(x, u) \in \mathbb{R}^{(n_x+n_y) \times (n_x+n_u)}$ s.t. $[(f(x, u)) (f_y(x, u))] := H(x, u)[x u]$. Then, Eq. (1) is equivalent to:

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

This qLPV model is scheduled by an endogenous nonlinear map¹ $f_\rho(\cdot)$, which imposes bounded and known scheduling variables ρ for all sampling instants. The scheduling set is given by: $\mathcal{P} := \{\rho_j \in \mathbb{R} \mid \underline{\rho}_j \leq \rho_j \leq \bar{\rho}_j, \forall j \in \mathbb{Z}_{[1, n_\rho]}\}$.

Assumption 1. The scheduling variables exhibit bounded rates of variation. This is: $\partial\rho(k) = (\rho(k) - \rho(k-1)) \in \partial\mathcal{P} := \{\partial\rho_j \in \mathbb{R} : \underline{\partial\rho}_j \leq \partial\rho_j \leq \bar{\partial\rho}_j, \forall j \in \mathbb{N}_{[1, n_p]}\}$.

2.2 The MPC Formulation

The MPC is applied at each sampling instant as follows: The states are measured, the scheduling parameters are computed, and the optimisation problem is solved:

$$\begin{aligned} \min_{U_k} V(x(k+N_p)) + \sum_{j=1}^{N_p} \ell(x(k+j), u(k+j-1)), \quad (3) \\ \text{s.t. } x(k+j+1) = A(\rho(k+j))x(k+j) \\ + B(\rho(k+j))u(k+j), j \in \mathbb{N}_{[0, N_p-1]}, \\ y(k+j) = C(\rho(k+j))x(k+j) \\ + D(\rho(k+j))u(k+j), j \in \mathbb{N}_{[0, N_p-1]}, \\ (x^T(k+j), u^T(k+j))^T \in (\mathcal{X} \times \mathcal{U}), j \in \mathbb{N}_{[0, N_p-1]}, \\ x(k+N_p) \in \mathbf{X}_f. \end{aligned}$$

$V(\cdot)$ is an offset and \mathbf{X}_f is a compact set (denoted terminal ingredients, see Sec. 4). The first entry of solution $U_k^* = \text{col}\{u^*(k+j)\}, \forall j \in \mathbb{N}_{[0, N_p-1]}$ is applied to the process.

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. For regulation, it is implied that x_r and u_r are nil. For tracking, a set-point target $z_r := (x_r^T, u_r^T)^T$ should imply the desired output target y_r . Assume there exists a linear (parameter varying) combination of the states x and inputs u that ensures $y(k) \rightarrow y_r$. As gives (Limon

¹ Through the sequel, we drop the dependency of f_ρ on u , for notation simplicity, but with no loss of generality.

et al., 2018), the following reference selector can be used to determine z_r :

$$\begin{aligned} \min_{z_r} \|(C(f_\rho(z_r)) \ D(f_\rho(z_r))) z_r - y_r\|^2, \quad (4) \\ \text{s.t. } \begin{bmatrix} (\mathbb{I} - A(f_\rho(z_r))) & -B(f_\rho(z_r)) \\ C(f_\rho(z_r)) & D(f_\rho(z_r)) \end{bmatrix} z_r = \begin{bmatrix} 0_{n_x} \\ y_r \end{bmatrix}, \\ (z_r^T, f_\rho(z_r)^T)^T \in ((\mathcal{X} \times \mathcal{U}) \times \mathcal{P}), \end{aligned}$$

which ensures an admissible steady-state target z_r that imposes the output tracking objective. If the output reference goal y_r is time-varying, this target selection problem can be solved online, increasing computational complexity.

3. QLPV NMPC

The MPC formulation in Eq. (3) 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 constraint 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 $\hat{\rho}(k+j)$. By doing so, the NPs are converted into more efficient programs, with the complexity of SQPs.

With regard to regular NMPC formulations, these novel qLPV embedding frameworks are attractive because the nonlinear state predictions and constraints are handled linearly at each sample. Thus, the numerical effectiveness becomes comparable to the fast real-time NMPCs, as demonstrated in (Cisneros and Werner, 2019).

Accordingly, we detail the basic concepts of two of these methods: Sequential guessing technique from (Cisneros and Werner, 2020) and the Recursive extrapolation approach from (Morato et al., 2019). We discuss the implementation steps and their convergence properties, pointing out advantages and drawbacks and providing realistic simulation essays (Sec. 5).

3.1 Sequential qLPV MPC

Method Description: The core idea of the SQP method (Cisneros and Werner, 2020) is the following: The MPC optimisation problem is solved multiple times per sampling period. At each iteration, the optimisation from Eq. (3) is solved based on “frozen” predictions. These are generated by plugging a “scheduling sequence” guess $\hat{P}_k := \text{col}\{\hat{\rho}(k+j)\}^T, \forall j \in \mathbb{N}_{[0, N_p-1]}$ into Eq. (2). At each iteration, \hat{P}_k is updated, until this guess converges to $\hat{P}_k = P_k$. The solution to the QP is the same as from a “full-blown” NP.

The method starts with an initial guess \hat{P}_k . Then, the internal state predictions of the MPC optimisation, denoted $X_k := \text{col}\{x(k+j)\}^T, \forall j \in \mathbb{N}_{[1, N_p]}$, are used to formulate the next guess, using the nonlinear scheduling

proxy function as follows: $\hat{P}_k = f_\rho \left(\underbrace{\begin{bmatrix} x^T(k), X_k^T \end{bmatrix}}_{X_k^*} \right)^T$.

Advantages: The method guarantees convergence within a relatively small number (around 3-5) of inner iterations. Moreover, at each inner iteration, the problem is formulated as a QP, based on the following prediction:

$$X_k = \mathcal{A}(\hat{P}_k)x(k) + \mathcal{B}(\hat{P}_k)U_k \quad , \quad (5)$$

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

Limitations: The main restraint of this approach is that the internal loop may take several iterations (QPs) to converge. This is not desirable because the number of inner iterations needed for convergence may require more time than the available sampling period. In practice, 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 instant k .

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 first method results in an SQP solution (since the QP is iterated until \hat{P}_k converges), this second approach is based on a single QP coupled to a recursive extrapolation method for \hat{P}_k . Accordingly, 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:

$$\hat{P}_k = \Phi(\hat{P}_{k-1}, X_{k-1}^*) = \lambda \hat{P}_{k-1} + \sigma_k X_{k-1}^* \quad , \quad (6)$$

for which λ is a forgetting factor and σ_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: 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 MPC solution 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 , an initial scheduling sequence P_0 ², known terminal ingredients, and a target reference goal z_r ³.

² These vectors can be simply taken as N_p repeated instances of $x(0)$ and $\rho(0)$.

³ For regulation purposes, z_r is nil, while for tracking purposes, it comes from the offline solution of (4).

Any embedded hardware is suitable to implement this Algorithm, as long as being able to solve QPs and perform linear matrix operations and nonlinear vectorisations.

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. (5);
 - (iii) Solve the optimisation in Eq. (3);
 - (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. (5);
- Step (2): Solve the optimisation in Eq. (3) with predictions from Eq. (5);
- Step (3): Take $u(k) = u^*(k)$ and apply this local control to the process;
- Step (4): $k \leftarrow k + 1$.

End

3.4 Convergence of $\hat{P}_k \rightarrow P_k$

In order to verify the convergence of scheduling trajectory estimates obtained with these methods, we can invoke the result from Newton SQPs, as demonstrated in (Cisneros and Werner, 2020). Basically, 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 from Sec. 2, 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. Note that if LDI is used to provide the qLPV model, then the iterations of Algorithm 1 are equivalent to those in the Newton SQP sub-problem, which is identical to both the optimisation given through the consecutive iterations Eq. (3) and the recursive operator of Eq. (6) with the solution of Eq. (3). The sufficient conditions, then, require local stability, which is ensured through adequate terminal ingredients.

4. STABILITY AND OFFLINE PREPARATIONS

Next, we briefly detail how to construct the terminal ingredients for Algorithm 1, in order to ensure stability and recursive feasibility properties.

The usual approach resides in \mathbf{X}_f and $V(\cdot)$ satisfying some conditions⁴ w.r.t. a nominal feedback $u = K(x - x_r)$. Consider that there exists a terminal state-feedback gain $K(\rho)$ and an ellipsoidal terminal set $\mathbf{X}_f :=$

⁴ For notation lightness, we proceed with x_r nil. The tracking equivalency is easily done with $\frac{dx_r}{dt} = 0$ and by computing the qLPV model with shifted states dynamics $(x - x_r)$.

5. APPLICATION RESULTS

$\{x | x^T P(\rho)x \leq \alpha_P\}$ with a sub-level cost $V(x, \rho) := x^T P(\rho)x$. 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. (3) 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 is feasible. Then, input-to-state stability and recursive feasibility are ensured 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 $\forall \partial\rho \in \partial\mathcal{P}$: $V(x^+, \rho + \partial\rho) - V(x, \rho) \leq -x^T Qx - x^T K^T(\rho)RK(\rho)x$;
- (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 Theo. 1 is standard (Mayne et al., 2000). Accordingly, the following Theorem provides parameter-dependent terminal ingredients which verify Theo. 1.

Theorem 2. The conditions (C1)-(C5) of Theo. 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 the following LMIs hold for all $\rho \in \mathcal{P}$ and $\partial\rho \in \partial\mathcal{P}$, under $\min \alpha$.

$$\begin{aligned} & \left[\begin{array}{c|cc} Y(\rho) & \star & \star \\ (A(\rho)Y(\rho) + B(\rho)W(\rho))Y(\rho + \partial\rho) & \star & \star \\ \hline Y(\rho) & 0 & Q^{-1} \star \\ W(\rho) & 0 & R^{-1} \end{array} \right] \geq 0, \\ & \left[\begin{array}{c|c} \alpha \bar{u}_i^2 & I_{\{i\}} W(\rho) \\ \star & Y(\rho) \end{array} \right] \geq 0, i \in \mathbb{N}_{[1, n_u]}, \\ & \left[\begin{array}{c|c} \alpha \bar{x}_j^2 & I_{\{j\}} Y(\rho) \\ \hline I_{\{j\}}^T Y^T(\rho) & Y(\rho) \end{array} \right] \geq 0, j \in \mathbb{N}_{[1, n_x]}. \end{aligned}$$

The proof of Theo. 2 follows from (Cisneros and Werner, 2020), using $\alpha = \alpha_P^{-1}$. 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 can be dropped if the system is quadratically stabilisable, but this may result in increased conservativeness.

Remark 1. The solution of Theo. 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 \mathbf{X}_f at each sampling instant. $K(\rho)$ is a fictive feedback gain used only to prove the stability conditions; the actual feedback is that determined by the online optimisation procedure.

In this Sec., we provide two different application results of the considered methods, which are henceforth denoted Sequential qLPV MPC and Recursive qLPV MPC. The following results are obtained using MATLAB simulation with the YALMIP and Gurobi solvers on a 2.4 GHz, 8 GB RAM Macintosh computer, using high-fidelity nonlinear models of the processes. SDPT3 was used to solve the LMIs in Theo. 2.

5.1 Regulation: Semi-active Suspension System

Firstly, we consider the regulation problem of a semi-active suspension system. This system is used to mitigate the vertical oscillations of a car when road bumps are encountered, thus improving comfort. Consider the corner dynamics of a vehicle equipped with ER suspensions, 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 parts, respectively. The road disturbance and the damper force are denoted z_r, F_{ER} , respectively. Use $z_d = z_s - z_{us}$ and $z_w = z_{us} - z_r$. From Newton's law of motion, we obtain:

$$\begin{aligned} \ddot{z}_s(t) &= -k_s z_d(t) - (c_0 + c_u u(t)) \dot{z}_d(t) - F_{ER}(t), \\ \ddot{z}_{us}(t) &= +k z_d(t) + (c_0 + c_u u(t)) \dot{z}_d(t) + F_{ER}(t) - k_t z_w(t), \\ \tau \dot{F}_{ER}(t) &= -F_{ER}(t) + f_c \tanh(k_1 z_d(t) + c_1 \dot{z}_d(t)) u(t). \end{aligned}$$

This system operates with a sampling period of $T_s = 5$ ms. Considering states $x = [z_d, \dot{z}_s, z_w, \dot{z}_{us}, F_{ER}]^T$, we obtain the following qLPV matrices:

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

$B(\rho) = T_s [0 \ 0 \ 0 \ 0 \ \frac{f_c}{\tau} \rho_2]^T$, $B_2 = T_s [0 \ 0 \ -1 \ 0 \ 0]^T$, which relates to Eq. (2) with an additional term $+B_2 \dot{z}_r(k)$ on the state dynamics. The scheduling parameters are $\rho := [\rho_1 \ \rho_2]^T$, where $\rho_1 := u \in [0, 1]$ 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 provided in (Morato et al., 2020b).

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 ($Q = \mathbb{I}_{n_x}$, $R = \mathbb{I}_{n_u}$). An $N_{iter} = 5$ iteration threshold for the Sequential qLPV MPC, and $\lambda = 1$ and $\sigma_k = 0$ in Eq. (6) for the Recursive method⁵. Consider an initial condition of $x_0 = [0.001, -0.001, 0.001, 0.001, 0]^T$. Figure 1 provides the road scenario and the corresponding control law (PWM signal that regulates the damper force). Accordingly, Fig. 2 presents some of the state trajectories (due to lack of space). Clearly, regulation is obtained with both NMPC techniques, and the road bumps are rejected in steady state.

⁵ These weights imply a higher degree of conservatism with the latter method, since we basically consider that the scheduling trajectory is constant.

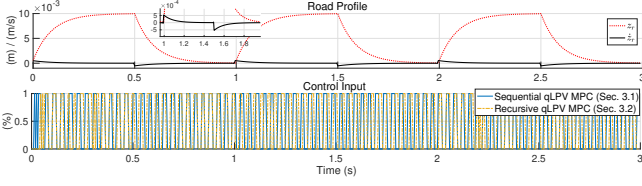


Fig. 1. Suspension system: Road scenario & control input.

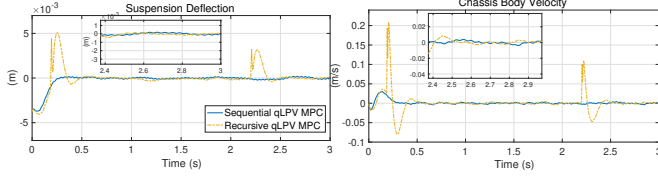


Fig. 2. Suspension system: State trajectories.

Both strategies are efficient for the regulation. Anyhow, we further assess the obtained results with the RMS index of the stage cost $\ell(x, u)$ along the whole simulation (Tab. 1). We can thus conclude that 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 $\hat{P}_k \neq P_k$ and thereby generates larger overshoots. 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 in Tab. 1) with the recursive solution is more than 10 times smaller than the one with the SQPs 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. The selection of the method will certainly depend on the hardware capabilities and the available sampling period. If the system had more states, $t_c < T_s$ could have been violated with the first method.

5.2 Tracking: Quadruple-tank Process

Now, consider the quadruple tank process from (Johansson, 2000): Four interconnected tanks (two on top of two) are regulated by two pumps and two valves, which determine the flow of circulating water:

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

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 γ_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 cm^2 , while

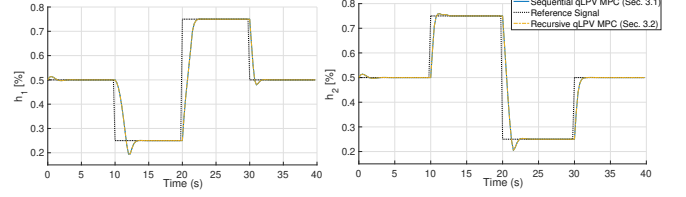


Fig. 3. Four-tank system: Reference tracking results.

the outlet hole cross sections a_i are of 0.05 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); $u_j \in [0.05, 5] \text{ V}$, $\forall j \in \mathbb{N}_{[1,2]}$ (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 qLPV realisation from 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 valve opening signals $\gamma_j = 1$ are assumed constant and known.

The application of the qLPV MPC methods is made using the same values for N_{iter} , λ , and σ from the previous essay. In this study, we add a complementary stop criterion to the SQP method: the internal loop stops if $\|\hat{P}_k - P_k\|_\infty \leq 10^{-3}$, which is a negligible bound for this system.

A tracking objective is considered: $\ell(x, u) := \|(x - x_r)\|_Q^2 + \|(u - u_r)\|_R^2$, being (x_r, u_r) admissible piece-wise constant reference signals (that satisfy Eq. (4)). This 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 $x(0) = [5 \ 5 \ 5 \ 5]^T$ and $u(0) = [5\sqrt{0.5} \ 5\sqrt{0.5}]$.

Figure 3 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 . Figure 4 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% of the sampling instants. Nevertheless, we must stress that the Recursive approach is over 2.5 times faster (on average) than the Sequential technique in computing the control action, as gives Tab. 1.

5.3 Discussion

Based on these two realistic essays, we can conclude that both qLPV MPC methods are interesting options for the

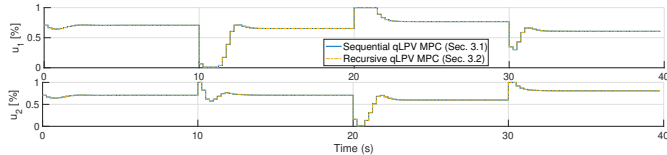


Fig. 4. Four-tank system: Control signals.

Table 1. Performance results.

Regulation: Suspension system	$r\{\ell(\cdot)\}$	t_c
Sequential qLPV MPC (Sec. 3.1)	4.548	3.795 ms
Recursive qLPV MPC (Sec. 3.2)	5.445	0.377 ms
Tracking: Four-tank system	$r\{\ell(\cdot)\}$	t_c
Sequential qLPV MPC (Sec. 3.1)	0.102	49.40 ms
Recursive qLPV MPC (Sec. 3.2)	0.102	18.90 ms

predictive control of nonlinear processes without the need for NP-hard solutions. Both methods generate good estimates for the scheduling trajectories. The SQP technique yields overall faster convergence, while the convergence of the recursive scheme requires some samples, which may lead to instantaneous performance degradation.

We stress that this performance deterioration may be tolerable since the recursive 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 SQP method requires the solution of (at most) N_{iter} QPs coupled with the nonlinear vector-wise operation of $f_\rho(\cdot)$, which may impede real-time applications for 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.

We also note that as long as if the closed-loop system is stable (implied through the terminal ingredients), there arise no control loss issues with these qLPV methods, since the model in Eq. (2) is an exact realisation of the nonlinear dynamics from Eq. (1). This is an advantage that must be emphasised, given that NMPCs with real-time iteration solutions, which are widely used, may have control losses due to the successive linearisations.

6. CONCLUSION

In this paper, we revisited some novel NMPC formulations based on qLPV embedding. These methods provide real-time NMPC solutions since the qLPV realisation offers linear predictions at each sampling period. 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. 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, CasADi, and GRAMPC. The main advantage is that no linearisations are performed, and only QP solutions are 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 the solver-based solutions, for instance, are still lacking, as well as experimental validation of the topics herein discussed.

REFERENCES

- 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. 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.
- 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.
- 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.
- 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.
- Sename, O., Gaspar, P., and Bokor, J. (2013). *Robust Control and Linear Parameter Varying Approaches: Application to Vehicle Dynamics*, volume 437. Springer.