Novel qLPV MPC Design with Least-Squares Scheduling Prediction

Marcelo Menezes Morato, Julio E Normey-Rico, Olivier Sename

Novel *qLPV MPC* Design with Least-Squares Scheduling Prediction ⋆

Marcelo Menezes Morato * **Julio E. Normey-Rico** *
Olivier Sename **

* *Dept. de Automação e Sistemas (*DAS*), Univ. Fed. de Santa Catarina, Florianópolis-SC, Brazil ({marcelomnzm} at gmail.com , {julio.normey} at ufsc.br).*
** *Univ. Grenoble Alpes, CNRS, Grenoble INP, GIPSA-lab, 38000 Grenoble, France ({olivier.sename} at gipsa-lab.fr).*

Abstract: The design of a Model Predictive Control (*MPC*) algorithm for *quasi* Linear Parameter Varying (*qLPV*) systems is developed herein. An online Least-Squares procedure that computes the future evolution of the *qLPV* scheduling parameters is at the core of the proposed method, which enables the replacement of a complex nonlinear optimization by a (much simpler) Quadratic Programming Problem (*QP*) one. The method also uses contractive terminal set constraints and a Lyapunov-associated terminal cost to the *MPC QP*, so that the domain of attraction of this controller is enlarged and feasibility is guaranteed. This paper ends with a successful simulation of this technique applied to the control of vehicular suspensions.

Keywords: Model Predictive Control, *quasi* Linear Parameter Varying Systems, Least Square Prediction, Constrained Control, Semi-Active Suspensions

1. INTRODUCTION

Over the last decade, Model Predictive Control (*MPC*) (Camacho and Bordons, 2013) has become a very well established technique, with more than 5800 successful applications (Alamir, 2013). It is a natural method towards optimal control of processes subject to constraints (Normey-Rico and Camacho, 2007). In *MPC* loops, a model is used to predict future outputs, based on past and current values and on the (optimal) future control actions; these actions are calculated by some solver that takes into account a cost function (performance goals) and process constraints (adequate operation).

Yet very powerful, standard *MPC* design is mainly attached to the idea of controlling plants with **linear time-invariant** (*LTI*) models, which is no true for nonlinear systems controlled over larger operating conditions or for when the process responses depend on external parameters, that are not directly managed by the control loop.

The concept of *MPC* itself is not restricted to linear models, as it can be extended to **nonlinear** ones, although the inclusion of nonlinear model predictions (*NMPC*) is not trivial and much increases the algorithm's complexity (Allgöwer and Zheng, 2012). *NMPC* algorithms suffer from issues related to their high complexity, especially when sought to run in real-time (for fast processes).

In paralel to the growth of predictive control applications, literature became very rich on design methods for Linear Parameter Varying (*LPV*) systems (Mohammadpour and Scherer, 2012; Sename et al., 2013). Such systems are nonlinear ones that dependent on **known**, **bounded** scheduling parameters ρ. Thanks to linear differential inclusion, nonlinear systems can be represented within a *quasi LPV* (*qLPV*) setting, with simple (*LTI* alike) mathematical frameworks. Although there exist generalized *NMPC* tools, the study of this control method for nonlinear systems with *qLPV* models is yet to be properly researched, and, therefore, the main motivation of this paper.

In the recent literature, interesting results have been presented to simplify the *LPV-MPC* problem into feasible (simpler) algorithms. Some of these are mentioned: (Ayala-Bravo and Normey-Rico, 2009) propose interpolation-based predictive controllers for a nonlinear system based on local *LTI* models, which have been successfully applied for the control of desalination plants (Ayala-Bravo et al., 2011). The downside of these methods is that they do not take into account the variability of the scheduling parameter, but plan the *LPV* system into many local *LTI* ones; therefore, they are not strictly optimal. Considering bounded rates of the scheduling parameters, robust *LPV-MPC* algorithms were developed in (Jungers et al., 2011) and (Casavola et al., 2012; Bumroongsri and Kheawhom, 2012), where the evolution of ρ is treated by offline procedures (Linear Matrix Inequality (*LMI*) and ellipsoidal constraints, respectively). The problem with such works is that they demand heavy offline computational procedures, which are not necessarily simple to perform. (Abbas et al., 2015, 2018) present very efficient robust *LPV-MPC* algorithms with *LMI* constraints, but only applicable to *LPV* system in the input-output (I/O) representation form. Explicit *MPC*s for *LPV* systems were investigated in (Besselmann et al., 2012) with stability and optimality

⋆ This work has been supported by *CNPq* (305785/2015 − 0) and *ITEA3* European project (15016) *EMPHYSIS*.

guarantees. The downside in these works is that, since the future values of ρ are unknown, the algorithm ensures the constraints are satisfied for all possible system trajectories, which leads to quite conservative results and highly-demanding *QP*s (numerical-wise). Cisneros et al. (2018) present an iterative *MPC* algorithm for *qLPV* systems that basically uses an initially frozen trajectory guess for ρ (that iterates according to measurements) and transforms the nonlinear problem into a linear one. The issue that resides with such method is that the results may be sub-optimal and that the system trajectory might not be inside the region of attraction of the *MPC*, resulting in infeasibility. Tube-based *MPC* design must also be rapidly mentioned, since interesting performance results can be achieved, as in (Hanema et al., 2016, 2017). Anyhow, if the tubes are badly planned, the algorithms may become excessively conservative, which is not the focus of this paper. Concerning the authors' works, an *LPV-MPC* algorithm for the control of automotive suspension dampers was developed in (Morato et al., 2018b), with a fixed ρ prediction, which is rather over-simplified, but effective; in (Morato et al., 2019), a filtered-*MPC* strategy is proposed for an *LPV* energy system, where a feedback filter adapts the (*LTI*) *MPC* law according to the evolution of ρ.

In Section 2, the standard *LPV-MPC* problem is defined, where it becomes evident that the future evolution of ρ becomes a computational issue, since: *i)* it is (*a priori*) unknown; and *ii)* it transforms the usual *QP MPC* algorithm into a complex nonlinear optimization procedure.

There is a vague space for feasible and implementable *LPV-MPC* laws, specially those that use neither heavy offline procedures nor excessive online conservativeness (as when computing all possible scheduling trajectories). Such novel tool would be an welcome extended to the *MPC* paradigm. Thereby, this paper's contributions to this question of prime importance are:

- Considering *qLPV* models, an online Least-Squares algorithm is proposed for the prediction of ρ inside a future prediction horizon (Section 3);
- With these predictions, a *qLPV-MPC* algorithm is proposed via a standard *QP*. It is developed with contractive terminal set constraints and a terminal stage cost, used to enlarge the domain of attraction and guarantee feasibility (Section 4);
- Numerical simulations of the proposed algorithm applied to a Vehicle Semi-Active Suspension system are presented to demonstrate its effectiveness (Section 5). Conclusions are drawn in Section 6.

2. PREDICTIVE CONTROL WITH *LPV* MODELS

The complete standard *MPC* algorithm is now described. This well-established technique is capable of obtaining an optimal control law that takes into account constraints on the states, outputs and control actions. With some bland assumptions, it is possible to guarantee closed-loop asymptotic stability. *MPC* is widely used for reference tracking and disturbance rejection in processes control (Camacho and Bordons, 2013) and it resides in solving[1]:

[1] Notation $(k+i|k)$ is used to represent a predicted value for instant $k+i$, computed at instant k. For now, the presence of disturbances is suppressed, for simplicity.

Problem 1.

$$\min_{U} \; J = \sum_{i=1}^{N_p} \overbrace{\ell\left(x(k+i|k), y(k+i|k), u(k+i-1|k)\right)}^{\text{MPC Cost}} \quad (1)$$

$$\text{s.t.} \quad \underbrace{x(k+i) = f(x(k), u(k))}_{\text{System Model}}\,, \quad (2)$$

$$u(k+i-1|k), x(k+i|k), y(k+i|k) \,\in \mathcal{U}, \mathcal{X}, \mathcal{Y}\,. \quad (3)$$

where U is the sequence of actions inside the prediction horizon N_p, i.e. $\mathrm{col}\{u(k|k)\,, \ldots\,, u(k+N_p-1|k)\}$. Sometimes, a terminal stage cost is also minimized, as well as the use of terminal constraints and slew rates. Throughout this work, take $x \in \mathbb{R}^n$, $u \in \mathbb{R}^m$ and $y \in \mathbb{R}^p$, with $\mathcal{U}$, $\mathcal{X}$ and $\mathcal{Y}$ as the set contraints that define feasibility.

If the system model equality (2) is linear (*LTI* case with $x(k+i) = Ax(k+i) + Bu(k)$), this optimization procedure is in fact a *QP*, which is easily tackled with standard solvers. Nonetheless, if the system model is *LPV*, the prediction problem is, in fact, nonlinear. Consider the following generic discrete-time *LPV* model:

$$\begin{aligned} x(k+1) &= A(\rho(k))x(k) + B_1(\rho(k))u(k)\,, \\ y(k) &= C(\rho(k))x(k) + D_1(\rho(k))u(k)\,, \end{aligned} \quad (4)$$

with a vector of scheduling parameters ρ that evolves as:

$$\Gamma_k = \mathrm{col}\{\rho(k+1)\,,\, \rho(k+2)\,\ldots\,,\, \rho(k+N_p)\}\,. \quad (5)$$

The model-based prediction that the optimization has to internally solve, with initial condition $x(k) = x_k$, is:

$$\begin{aligned} 1)\ x(k+1|k) &= A(\rho(k))x_k + B_1(\rho(k))u(k|k)\,, \quad (6)\\ y(k|k) &= C(\rho(k))x_k + D_1(\rho(k))u(k|k)\,;\\ 2)\ x(k+2|k) &= A(\rho(k+1))A(\rho(k))x_k \quad (7)\\ &+A(\rho(k+1))\,B_1(\rho(k))\,u(k|k) + B_1(\rho(k+1))u(k+1|k)\,,\\ y(k+1|k) &= C(\rho(k+1))A(\rho(k))x_k\\ &+C(\rho(k+1))\,B_1(\rho(k))\,u(k|k) + D_1(\rho(k+1))u(k+1|k)\,; \end{aligned}$$

and so forth, up to the N_p-th prediction. Notice that nonlinear terms are already present on the second prediction, as gave Eq. (7), which results in a non-*QP* version of Problem 1. Moreover, to say one has knowledge of Γ_k is obviously false, since only $\rho(k)$ is known. For these two reasons, this work investigates how to feasibly translate Problem 1 into a *QP* version.

3. PREDICTION OF ρ USING LEAST-SQUARES IDENTIFICATION

As discussed in Sec. 2, for *MPC* design, it is imperious to describe the response of the system in the future (up to N_p steps ahead). With *qLPV* models, the future outputs $y(k+t|k)$ depend not solely on the future inputs $u(k+t)$, but also on the future (endogenous) scheduling parameters $\rho(k+t|k)$. Although the scheduling signals are known (measured/observed) at instant k, their future behaviours are unknown. Herein, the description of the scheduling parameters (for the future N_p) steps is performed according to an online Least Squares (*LS*) procedure, as follows:

Assumption 1. The behaviour of the endogenous scheduling parameters of *qLPV* systems can be approximately described [2] by linear autoregressive (*ARX*) models, function of past values of the control signal and measured outputs. This *ARX* model for the vector of endogenous scheduling parameters is:

$$\begin{aligned}\rho(k+N_p) &= a_1\rho(k) + \cdots + a_{N_p}\rho(k-N_p) \quad (8)\\ &+ b_1 u(k-1) + \ldots + b_{N_p}u(k-N_p-1)\\ &+ \underbrace{\left[\, c_1^1 \,\ldots\, c_1^p \,\right]}_{c_1} y(k) + \ldots + \underbrace{\left[\, c_{N_p}^1 \,\ldots\, c_{N_p}^p \,\right]}_{c_{N_p}} y(k-N_p)\,,\end{aligned}$$

which is in fact a discrete-time model with N_p delays.

Model (8) can be extended to write Γ_k based solely on known values, which is welcome for the *MPC* design procedure, as discussed in the sequel. For such, it remains to find parameters a_1 to $c^p_{N_p}$. These parameters are coupled together as $\Theta = \text{col}\{a_1 \ldots c_{N_p}\}$, giving:

$$\rho(k) = \Theta\Psi^T\,, \tag{9}$$

for $\Psi = [\rho(k-N_p) \ldots \rho(k-2N_p)\; u(k-N_p-1) \ldots u(k-2N_p-1)\; y^T(k-N_p) \ldots y^T(k-2N_p)]$. The direct solution used to find Θ is an online recursive *LS* algorithm:

$$\begin{aligned}\Theta(k) &= \Theta(k-1) + \lambda Q_\theta\left(\Psi(k), \rho(k), y(k), u(k)\right)\,, \quad (10)\\ \Psi(k) &= \Psi(k-1) + \mu Q_\psi\left(\Psi(k-1), \rho(k), y(k), u(k)\right)\,,\end{aligned}$$

where λ and μ are update percentage parameters (forgetting factors) and Q_θ and Q_ψ are update functions [3].

After the *ARX* model parameters are obtained at instant k, based on historical data, an approximate prediction for Γ_k from Eq. (5) (namely $\hat{\Gamma}_k$) is found directly, by computing Eq. (8).

4. NOVEL *QUASI LPV MPC* ALGORITHM

Using the *LS*-derived prediction guess $\hat{\Gamma}_k$, Problem 1 can be converted into a *QP* version, since the nonlinearities from the predictions (Eqs. (6)-(7)) are no longer [4]:

$$\begin{aligned}x(k+j|k) &= A^j(\hat{\Gamma}_k)x_k + B_1^j(\hat{\Gamma}_k)U\,, \quad (11)\\ y(k+j-1|k) &= C^{j-1}(\hat{\Gamma}_k)x_k + D_1^{j-1}(\hat{\Gamma}_k)U\,.\end{aligned}$$

The proposed *qLPV-MPC* algorithm must use some other tools in order to guarantee that the domain of attraction is enlarged and that the controller operation is indeed feasible. Notice that, since an approximate solution for the evolution of the scheduling parameters is used, the control policy computed via Problem 1 at instant k may be infeasible due to model-plant mismatches (caused by the differences between Γ_k and $\hat{\Gamma}_k$) or even drive the system out of an stability region, which can never be allowed.

Firstly introduced by Mayne et al. (2000), the use of *i)* contractive terminal sets and *ii)* terminal stage costs has become extremely important for the control of uncertain systems (or nonlinear ones with model-plant mismatches, as it is the case herein). These tools allow the controlled system to meet the performance objectives (such as reference tracking, disturbance rejection *etc*), whilst stability and feasibility are maintained. Both these techniques have been combined with *iii)* artificial reference tracking, which leads to an enlargement of the domain of attraction of *MPC* algorithms, finding more options of stable closed-loop equilibrium points (Limon et al., 2005).

Remark 1. In (Limón et al., 2008), these three tools are generalized for the case of reference tracking. The framework proposed therein was extended in (Ferramosca et al., 2009), becoming able to guarantee performance and feasibility of such *MPC* policies when applied to nonlinear systems. In the sequel, they are individually explained:

4.1 Reference Tracking

Instead of the usual *MPC* reference tracking procedure (i.e. weighting the quadratic difference between output and reference in the cost function), consider: 1) $Q \in \mathbb{R}^{n\times n}$ and $R \in \mathbb{R}^{m\times m}$ as positive definite matrices, and 2) $K \in \mathbb{R}^{m\times n}$ as an arbitrary stabilizing control gain s.t. $(A(\rho(k+N_p))+B(\rho(k+N_p))K)$ is Hurwitz and $P \in \mathbb{R}^{n\times n}$ as a positive definite matriz s.t. $(A(\rho(k+N_p))+B(\rho(k+N_p))K)^T P(A(\rho(k+N_p))+B(\rho(k+N_p))K) - P = -(Q+K^TRK)$.

With these two hypothesis verified and any feasible initial state x_0, it can be guaranteed that an *MPC* controller can asymptotically steer the controlled system to the steady-state reference $\hat{x}_s$ in an admissible manner, by minimizing the following adjusted cost function (Limón et al., 2008):

$$\begin{aligned}J_{RT} = J_o &+ ||x(k+N_p|k) - x_s||_P^2 \quad (12)\\ &+ \sum_{i=1}^{N_p}\left(||x(k+i|k) - x_s||_Q^2\right)\\ &+ \sum_{i=1}^{N_p}\left(||u(k+i-1|k) - u_s||_R^2\right)\,,\end{aligned}$$

with $x_s \in \mathcal{X}$, $u_s \in \mathcal{U}$ and J_o as a quadratic offset function that penalizes the deviation between the artificial reference x_s and the target operation point $\hat{x}_s$.

Note that, with this tool, the pseudo-reference x_s is created s.t. the system is set to track it, while this signal must stay as close as possible to the actual reference $\hat{x}_s$. Moreover, if an output reference $\hat{y}$ is the preferred option, it must be true that the (possibly time-varying) target steady-state equilibrium $p_t = (\hat{x}_s, u_s)$ leads to the desired output $\hat{y}$, i.e. $\hat{y} = C(\rho(k+N_p))\hat{x}_s + D_1(\rho(k+N_p))u_s$. Remark, once again, that this procedure is approximated due to the use of $\rho(k+N_p)$, which is taken from the *LS* estimative $\hat{\Gamma}_k$.

It is reasonable to assume that the target operation point $p_t = (\hat{x}_s, u_s)$ is an admissible steady-state, which

[2] Note that this assumption is quite reasonable for *qLPV* models, since the scheduling parameters, at some instant k', are imperiously function of the states and inputs, i.e. $\rho(k') = f_\rho(u(k'), y(k'))$. Although ρ is varying over its whole spectrum, the *ARX* model can give an approximate guess for its future behaviour (N_p steps) from the viewpoint of instant k'.

[3] This paper will not prolong itself on this procedure; the complete deduction is found in (Ljung, 1987), see Chapter 11.2.

[4] Note that the nonlinear terms become constant matrices, dependent on (the fixed) $\hat{\Gamma}_k$.

derives from system (4) being LPV-stabilizable, w.r.t. the definition presented by Shamma (2012).

4.2 Terminal Cost J_o

The inclusion of a suitable penalization of the terminal state combined with a terminal constraint can lead to asymptotic stability with satisfaction of performance contraints can be proved, as done in (Ferramosca et al., 2009). For such, the offset J_o should be convex s.t.:

$$\beta_1 ||x_s - \hat{x}_s||_1 \leq J_o(p_s, p_t) \leq \beta_2 ||x_s - \hat{x}_s||_1 \,, \tag{13}$$

where β_1, β_2 are positive real constants and $p_s = (x_s, u_s)$.

These two tools guarantee that if the system evolves as predicted (i.e. $\Gamma_k = \hat{\Gamma}_k$) and if p_t is an admissible point contained inside the tracking set, then it is an asymptotically stable point in closed-loop. Elsewise, the final closed-loop equilibrium is $p_s^\star = (x_s^\star, u_s^\star) = \arg\min_{p_s} J_o(p_s, p_t)$.

4.3 Contractive Terminal Set

As (very importantly) introduced by Blanchini (1999), the notion of reachable sets is recalled: 1) a set $\Upsilon \subset \mathbb{R}^n$ is a *control invariant set* for system (4), subject to its operational constraints (3), if, for all $x \in \Upsilon$ there exists an admissible input $u = u(x) \in \mathbb{R}^m$ such that $f(x, u) \in \Upsilon$ [5]; 2) the *one-step set* of Υ, $\mathcal{Q}\{\Upsilon\}$, stands for the set of states which can be steered in one step k to the target set Υ by an admissible control action; 3) a given set Υ is, thence, a *control invariant set* iff $\Upsilon \subseteq \mathcal{Q}\{\Upsilon\}$; 4) a *sequence of reachable sets* $\{\Upsilon_i\}$ is the sequence of sets by which the x can be driven through, passing from one set Υ_i to the following Υ_{i-1}, in an admissible way, finally reaching the (target invariant set) Υ, see (Bertsekas and Rhodes, 1971).

Then, to guarantee that in N_r steps the controlled system (4) reaches a control invariant set Υ that contains the target performance steady-state equilibrium point p_t, the following contractive terminal set constraint is derived:

$$x(N_P) \in \Upsilon_j \,, \; j = \max\{N_r - k \,, 0\} \,, \tag{14}$$

under the assumption that a sequence of N_r reachable sets $\{\Upsilon_i\}$ is available. This terminal set Υ_j is equal to the larger Υ_{N_r} at the initial instant k_0 being shrinked subsequently until, at $k_0 + N_r$, it becomes the smallest set Υ.

When the above constraint is coupled to the MPC design, there is indeed an enlargement of its domain of attraction, giving further holds on stability and feasibility, which are needed due to model-plant differences, i.e. $\Gamma_k \neq \hat{\Gamma}_k$. Note that the sequence of reachable sets are computed with $\hat{\Gamma}_k$. If further robustness is sought, one could assume bounded rates on ρ (i.e. $\frac{d\rho}{dt}(t) \in \dot{\mathcal{P}}$) and compute all possible trajectories and reachable set sequences from x_k and, finally, take their intersection as $\{\Upsilon_i\}$.

Since all the necessary tools have now been presented, the novel *qLPV-MPC* algorithm is obtained as follows:

Algorithm 1. (1) Iterate Eq. (10), obtaining an approximate guess for the N_p-steps evolution of the scheduling parameters $\hat{\Gamma}_k$;

(2) Find the linear system evolution/prediction laws that approximate the behaviour of the controlled $qLPV$ system for the next N_p steps, given by Eq. (11) with $j = 1, \ldots, N_p$.

(3) Then, solve the following QP:

$$\begin{aligned} \min_U \; & J = J_{RT} \\ & + \sum_{i=1}^{N_p} \ell\left(x(k+i|k), y(k+i|k), u(k+i-1|k)\right) \\ \text{s.t.} \quad & \text{System Evolution: Eq. (11)}\,, \\ & y(k+i|k) \in \mathcal{Y}\,, \\ & u(k+i-1|k) \in \mathcal{U}\,, \\ & x(k+i|k) \in \mathcal{X}\,, \\ & x(k+N_P|k) \in \Upsilon_j \,,\; j = \max\{N_r - k\,, 0\}\,, \end{aligned} \tag{15}$$

(4) From the solution U, take the first entry $u(k|k)$ and apply it to the controlled plant.

Note that the policy derived from this algorithm is imperiously time-varying for the first N_r samples, due to the contractive terminal constraint. Usually, $N_r \geq N_p$.

5. NUMERICAL EXAMPLE

Simulation results are now presented to assess the performance of the proposed *qLPV-MPC* policy achieved with Algorithm 1. The considered study-case is the control of the vertical dynamics of a 1/5-scaled vehicle equipped with 4 semi-active dampers [6]. This system is described by a Quarter-of-Vehicle $qLPV$ model that comprises the vertical displacement of each chassis corner $z_s(t)$ and of each wheel $z_{us}(t)$, due to the road disturbances $z_r(t)$. The control input for this system is the damping coefficient variation $u(t)$; the complete damping force is given by $F_d(t) = (c + u(t))\,(\dot{z}_s(t) - \dot{z}_{us}(t))$. This damping force is naturally bounded, which leads to the dissipativity contraints $u(t) \in \mathcal{D} = [\underline{u}\,, \overline{u}]$. The endogenous scheduling parameter is the suspension deflection velocity $\rho = \dot{z}_s(t) - \dot{z}_{us}(t)$. The vertical acceleration variables are the sole measurable outputs, with the use of on-board vehicle sensors (i.e. inertial units), this is:

$$\begin{aligned} \dot{x}(t) &= Ax(t) + B_1(\rho)u(t) + B_2 z_r(t) \\ y(t) &= Cx(t) + D_1(\rho)u(t) \end{aligned} \,, \tag{16}$$

with $x(t) = \left[\, z_s(t)\ \dot{z}_s(t)\ z_{us}(t)\ \dot{z}_{us}(t) \,\right]^T$ and $y(t) = \left[\, \ddot{z}_s(t)\ \ddot{z}_{us}(t) \,\right]^T$. Model matrices are:

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 \\ \frac{-k_s}{m_s} & \frac{-c}{m_s} & \frac{k}{m_s} & \frac{c}{m_s} \\ 0 & 0 & 0 & 1 \\ \frac{k_s}{m_{us}} & \frac{c}{m_{us}} & -\frac{(k_t + k_s)}{m_{us}} & \frac{-c}{m_{us}} \end{bmatrix}, B_2 = \begin{bmatrix} 0 \\ 0 \\ 0 \\ \frac{k_t}{m_{us}} \end{bmatrix},$$

$$B_1(\rho) = \begin{bmatrix} \overbrace{\left[0 \; -\frac{\rho}{m_s}\right]}^{=-D_1^T(\rho)} & D_1^T(\rho) \end{bmatrix}^T, C = \begin{bmatrix} 1 & 0 & -1 & 0 \\ \frac{-k_s}{m_s} & 0 & \frac{k_s}{m_s} & 0 \end{bmatrix}.$$

[5] The vectorial map f represents the LPV system model application, i.e. $f(x, u) = A(\rho(k))x(k) + B_1(\rho(k))u(k)$.

[6] Refer to *http://www.gipsa-lab.fr/projet/inove/*. Results are shown for the front-left corner of the vehicle; similar results were obtained for the other corners

The control goal $\ell(\cdot)$ is set to minimize both chassis and wheel accelerations and, by doing so, to achieve a smoother and more comfortable drive, while respecting the semi-active (min./max.) dissipativity constraints (Morato et al., 2018a).

$$\min_{u(t)} \int_0^{\tau} \overbrace{\left(a_1 \ddot{z}_s^2(t) + a_2 \ddot{z}_{us}^2(t)\right)}^{\ell(\cdot)} dt \,, \qquad (17)$$
$$\text{s.t. } u(t) \in \mathcal{D} \,.$$

a_1 and a_2 are taken, respectively, as 0.95 and 0.05 so that passengers are isolated from the road bumping. To compute the *MPC* control action, the above model is discretized with a sampling period of $T_s = 5\,\text{ms}$.

The following results are obtained with the aid of softwares packages *Matlab*, *Yalmip* and *SDPT*3 (*QP*) solver. Model parameters are: $m_s = 2.27\,\text{kg}$; $m_{us} = 0.32\,\text{kg}$; $k_t = 12270\,\text{N/m}$; $k = 1396\,\text{N/m}$; $c = 70\,\text{N.s/m}$. The chosen road disturbance, $z_r(t)$ in Fig. 1, represents a car running in a straight line on a dry road, when it encounters ($t' = 0.5\,\text{s}$) a sequence of 5 mm bumps on all its wheels, exciting a bouncing motion.

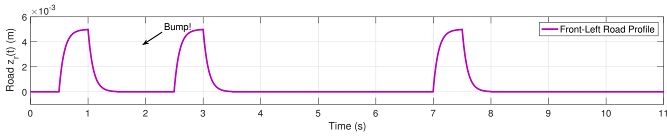

Fig. 1. (Front-Left) Simulation Scenario

As suggests Morato et al. (2018a), the prediction horizon N_p is taken as 10 samples, while the contractive horizon N_r is taken as 50. This means that *control set* shrinks with the pace five times slower than the receding horizon (note that N_r does not slide). To be in accordance with the control goal, the traget equilibrium is taken as $p_t = (\hat{x}_s\,, u_s) = ([\,\cdot\; 0\; \cdot\; 0\,]\,, 0)$. Weighting matrices are $Q = a_1 \mathbb{I}_n$ and $R = a_1 \mathbb{I}_m$.

To elucidate the effectiveness of Algorithm 1, the proposed controller is compared to a simpler one (namely *SMPC*), that makes no use of the terminal cost and set tools described in Sec. 4, simply solving the original *MPC* Problem (1) with a constant prediction guess for the scheduling parameter, i.e. $\hat{\rho}(k+t|k) \approx \rho(k|k)$.

In terms of numerical complexity, the proposed *qLPV-MPC* method takes, in average, **only** $3.09\,\%$ **longer** elapsed (computational) time to compute the control policy $u(k)$. This is tolerable amount, given that it does not violate operational contraints, but achieves significantly better performances, as demonstrated in the sequel:

Fig. 2 shows the controlled outputs (accelerations of the chassis axle $\ddot{z}_s(t)$ and wheel link $\ddot{z}_{us}(t)$) with both methods. It is evident that the proposed *qLPV-MPC* technique can further minimize the control objective $\ell(\cdot)$, while abiding to the dissipativity constraints $\mathcal{D}$ (shown in Fig. 3[7]). Numerically speaking, the proposed approach (compared to the *SMPC*) presents a **significant** $9.35\,\%$ **of reduction** of the root-mean-square value of the performance objective $\ell(\cdot)$[8], which would certainly be felt in terms of **passenger comfort**.

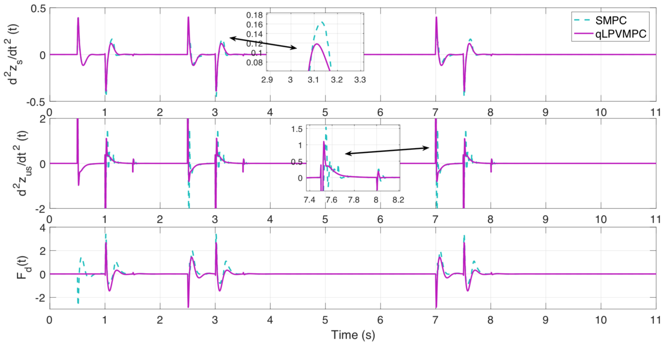

Fig. 2. (Front-Left) Sprung/Unsprung Accelerations and Damper Force

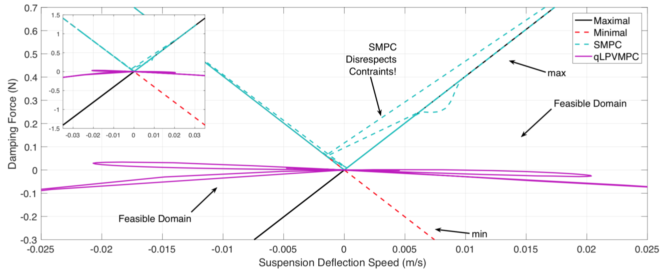

Fig. 3. (Front-Left) Damper Dissipativity Constraints

Fig. 4 shows some snippets of the evolution of the (front-left corner) scheduling parameters $\dot{z}_s(t) - \dot{z}_{us}(t)$ compared with the *LS* predictions at some points. It is clear that the *ARX* model given by Eq. (8) cannot catch the complete behaviour of ρ, but it provides a sufficient guess that is adequately incorporated to the *MPC* loop. In comparison with a constant/fixed prediction, the use of the *LS* tool provides much more trustworthiness.

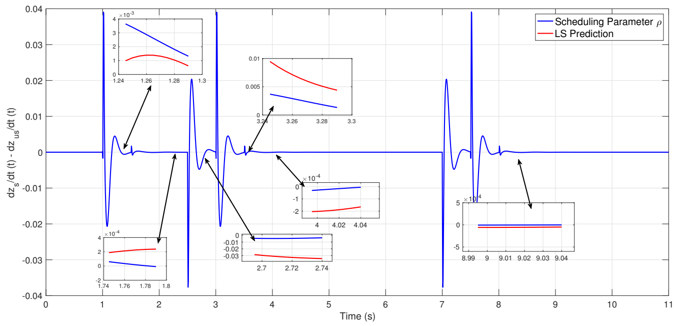

Fig. 4. (Front-Left) Scheduling Parameter Evolution

Finally, Fig. 5 shows the evolution of the velocity states $\dot{z}_s(t)$ and $\dot{z}_{us}(t)$ and illustrates the *sequence of reachable control sets*[9]. It is clear that the contractive terminal set constraint (14) makes these velocities converge to a final set Υ (from $t = 1$ to $5\,\text{s}$), guaranteeing that the system is not driven into instability.

[7] Note that the *SMPC* approach disrespects these contraints at some moments!

[8] The obtained rms$\{\ell(\cdot)\}$ values were 0.21217 (*SMPC*) and 0.19233 (*qLPV-MPC*).

[9] The boxes are not the actual sets, but just illustration tools to show how they shrink.

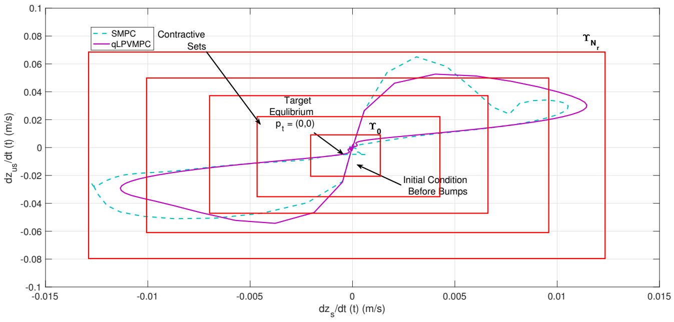

Fig. 5. Contractive Terminal Set Contraint

6. CONCLUSIONS

This paper elaborated on a novel *MPC* algorithm for nonlinear systems with *qLPV* models. The method takes a *LS* guess for the future scheduling parameters behaviours, which transforms the nonlinear prediction problem into a linear *QP*. This is a welcome application, since many nonlinear systems can be embedded in the *qLPV* representation form. The algorithm is applied to the control of a Semi-Active suspension system, achieving good results. For further works, stronger stability, optimality and feasibility holds of the proposed algorithm will be presented.

REFERENCES

Abbas, H.S., Hanema, J., Tóth, R., Mohammadpour, J., and Meskin, N. (2018). A new approach to robust MPC design for LPV systems in input-output form. *IFAC-PapersOnLine*, 51(26), 112–117.

Abbas, H.S., Tóth, R., Meskin, N., Mohammadpour, J., and Hanema, J. (2015). An MPC approach for LPV systems in input-output form. In *Decision and Control (CDC), 2015 IEEE 54th Annual Conference on*, 91–96. IEEE.

Alamir, M. (2013). *A pragmatic story of model predictive control: self-contained algorithms and case-studies.* CreateSpace Independent Publishing Platform.

Allgöwer, F. and Zheng, A. (2012). *Nonlinear model predictive control*, volume 26. Birkhäuser.

Ayala-Bravo, C.O. and Normey-Rico, J.E. (2009). Approach for non-linear predictive control based on the local model ideas. *Sba: Controle & Automação Sociedade Brasileira de Automatica*, 20(4), 465–481.

Ayala-Bravo, C.O., Roca, L., Guzman, J.L., Normey-Rico, J.E., Berenguel, M., and Yebra, L. (2011). Local model predictive controller in a solar desalination plant collector field. *Renewable energy*, 36(11), 3001–3012.

Bertsekas, D.P. and Rhodes, I.B. (1971). On the minimax reachability of target sets and target tubes. *Automatica*, 7(2), 233–247.

Besselmann, T., Lofberg, J., and Morari, M. (2012). Explicit MPC for LPV systems: Stability and optimality. *IEEE Transactions on Automatic Control*, 57(9), 2322–2332.

Blanchini, F. (1999). Set invariance in control. *Automatica*, 35(11), 1747–1767.

Bumroongsri, P. and Kheawhom, S. (2012). An ellipsoidal off-line model predictive control strategy for linear parameter varying systems with applications in chemical processes. *Systems & Control Letters*, 61(3), 435–442.

Camacho, E.F. and Bordons, C. (2013). *Model predictive control.* Springer Science & Business Media.

Casavola, A., Famularo, D., Franzè, G., and Garone, E. (2012). A fast ellipsoidal MPC scheme for discrete-time polytopic linear parameter varying systems. *Automatica*, 48(10), 2620–2626.

Cisneros, P.S., Sridharan, A., and Werner, H. (2018). Constrained predictive control of a robotic manipulator using quasi-LPV representations. *IFAC-PapersOnLine*, 51(26), 118–123.

Ferramosca, A., Limon, D., Alvarado, I., Alamo, T., and Camacho, E.F. (2009). Mpc for tracking of constrained nonlinear systems. In *Proceedings of the 48th IEEE Conference on Decision and Control held jointly with 28th Chinese Control Conference*, 7978–7983.

Hanema, J., Lazar, M., and Tóth, R. (2017). Stabilizing tube-based model predictive control: Terminal set and cost construction for LPV systems. *Automatica*, 85, 137–144.

Hanema, J., Tóth, R., and Lazar, M. (2016). Tube-based anticipative model predictive control for linear parameter-varying systems. In *55th Conference on Decision and Control (CDC)*, 1458–1463. IEEE.

Jungers, M., Oliveira, R.C., and Peres, P.L. (2011). MPC for LPV systems with bounded parameter variations. *International Journal of Control*, 84(1), 24–36.

Limon, D., Alamo, T., and Camacho, E.F. (2005). Enlarging the domain of attraction of MPC controllers. *Automatica*, 41(4), 629–635.

Limón, D., Alvarado, I., Alamo, T., and Camacho, E.F. (2008). MPC for tracking piecewise constant references for constrained linear systems. *Automatica*, 44(9), 2382–2387.

Ljung, L. (1987). *System identification: theory for the user.* Prentice-hall.

Mayne, D.Q., Rawlings, J.B., Rao, C.V., and Scokaert, P.O. (2000). Constrained model predictive control: Stability and optimality. *Automatica*, 36(6), 789–814.

Mohammadpour, J. and Scherer, C.W. (2012). *Control of linear parameter varying systems with applications.* Springer Science & Business Media.

Morato, M.M., Mendes, P.R.C., Normey-Rico, J.E., and Bordons, C. (2019). LPV-filtered predictive control design for fault-tolerant energy managements. *IFAC-PapersOnLine*, 51(1), 166–171.

Morato, M.M., Nguyen, M.Q., Sename, O., and Dugard, L. (2018a). Design of a fast real-time LPV model predictive control system for semi-active suspension control of a full vehicle. *Journal of the Franklin Institute.*

Morato, M.M., Sename, O., and Dugard, L. (2018b). LPV-MPC fault tolerant control of automotive suspension dampers. *IFAC-PapersOnLine*, 51(26), 31–36.

Normey-Rico, J. and Camacho, E. (2007). *Control of dead-time processes.* Springer Science & Business Media.

Sename, O., Gaspar, P., and Bokor, J. (2013). *Robust control and linear parameter varying approaches: application to vehicle dynamics*, volume 437. Springer.

Shamma, J.S. (2012). An overview of lpv systems. In *Control of linear parameter varying systems with applications*, 3–26. Springer.