



HAL
open science

An Input-to-State Stable Model Predictive Control Framework for Lipschitz Nonlinear Parameter Varying Systems

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

► **To cite this version:**

Marcelo Menezes Morato, Julio E Normey-Rico, Olivier Sename. An Input-to-State Stable Model Predictive Control Framework for Lipschitz Nonlinear Parameter Varying Systems. *International Journal of Robust and Nonlinear Control*, 2021, 31 (7), pp.8239-8272. 10.1002/rnc.5243. hal-02927449

HAL Id: hal-02927449

<https://hal.science/hal-02927449>

Submitted on 1 Sep 2020

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.

SPECIAL ISSUE ARTICLE

An Input-to-State Stable Model Predictive Control Framework for Lipschitz Nonlinear Parameter Varying Systems

Marcelo Menezes Morato^{*1,2} | Julio Elias Normey-Rico¹ | Olivier Sename²

¹Departamento de Automação e Sistemas, Universidade Federal de Santa Catarina, Florianópolis, Brazil

²GIPSA-lab, Grenoble INP, Université Grenoble Alpes, 38000 Grenoble, France

Correspondence

*Marcelo Menezes Morato,
DAS/CTC/UFSC, Trindade, Caixa Postal
476, Florianópolis-SC, 88040-900, Brazil
Email: marcelomnzm@gmail.com

Summary

This paper develops a state-feedback Model Predictive Control (MPC) framework for Nonlinear Parameter Varying (NLPV) systems with explicit Lipschitz nonlinearities along the input trajectory. Considering a guess for the evolution of the scheduling parameters along the prediction horizon, the proposed optimization procedure for the MPC design includes a terminal stage cost, a contracting terminal region constraint and nominal predictions scheduled with respect to this guess. The terminal region is taken so that it is monotonically decreasing to a pre-determined set, which guarantees recursive feasibility of the algorithm with respect to a bound on the admissible uncertainties (introduced from the scheduling parameter guess). The terminal set is a scheduled robust control invariant set for Lipschitz nonlinear systems, computed through some proposed LMIs. The inclusion of the terminal ingredient also serves to demonstrate Input-to-State Quadratic Stability. This paper ends with a successful numerical simulation example of the technique applied to the control of a Semi-Active automotive suspension system equipped with Electro-Rheological dampers.

KEYWORDS:

Model Predictive Control, Nonlinear Parameter Varying Systems, Lipschitz Nonlinearity

1 | INTRODUCTION

Model Predictive Control (MPC) is a very well-established and widespread technique, with a large gamma of industrial applications, more than 5800 successful results are reported^{1,2}, including application for processes in chemical, mechanical and thermal areas³.

The basic MPC algorithm provides a natural framework for optimal control, computing some action u at each discrete instant k by solving an optimization problem, which is written in terms of a process prediction model¹. The optimization cost function is used to include performance goals, such as reference tracking and disturbance rejection. Furthermore, this framework allows to explicitly consider the effect of input, output and state constraints in the control design procedure, which is rather convenient.

The first MPC schemes were originally intended for linear time-invariant (LTI) processes, using impulse-response or state-space formulations^{4,5}. Nowadays, LTI MPC design has many variations and adaptations, including design with feedforward

⁰**Abbreviations:** LPV, Linear Parameter Varying; NLPV, Nonlinear Parameter Varying; LTI, Linear Time Invariant; MPC, Model Predictive Control; PC, Proper C; ISS, Input-to-state Stability.

¹Of course, predictive control has also been generalized for the case of model-free paradigms, which are often referred to as “data-driven” or “learning” algorithms. These are not within the scope of this paper.

capabilities, for economic purposes, using mixed-integer constraints and so on. The future promise of predictive control and some of these developments have been unfolded by a recent survey⁶.

Soon enough, these LTI predictive control algorithms were adapted for nonlinear paradigms⁷, for those when systems are controlled over larger operating conditions or for when the process responses heavily depend on external parameters. To extend MPC to include nonlinearities (which is often referred to as NMPC) in the process model is not a trivial task, increasing the algorithm's complexity⁸. It is worth noting that stability can be guaranteed for nonlinear plants controlled under MPC policies with quite systematic verification axioms⁹. Literature has blossomed with quite efficient NMPC frameworks, as recently reviewed¹⁰. Although the majority of NMPC design requires excessive computational capacities, some papers show real-time, embedded applications of these methods^{11,12,13}.

To overlap the numerical complexity of full-blown NMPC design, the neatness of these recent methods include: *a*) some approaches that overcome the nonlinearities via model linearization along the trajectory¹⁴; *b*) manipulating the nature of the nonlinearities to represent them in some specific manner (as bilinearities, for instance), which may reduce complexity¹⁵; *c*) a few approaches that parametrize the control inputs in finite amount of possible discrete values, solving a search algorithm to find the smallest cost instead of an actual optimization procedure^{16,17}. Essentially, all these works let go of optimality and use sub-optimal solutions of the complex nonlinear prediction problem¹⁸. We understand that real embedded NMPC applications for fast processes (with small sampling rates, at the magnitude of a few milliseconds) without abdicating optimality concerns are still not available, but should soon enough be established, in the view of the recent progress in this field.

1.1 | Linear Parameter Varying Systems

Literature points out that to tackle and facilitate the control of nonlinear processes, the use of Linear Parameter Varying (LPV) systems^{19,20} is an interesting option. Today, the LPV framework has become one of the most popular kinds of modelling toolkits for process with complex dynamics^{21,22}.

LPV systems are nonlinear ones that depend on known, bounded scheduling parameters ρ (their current values $\rho(k)$ can be measured online or estimated, while their future behaviour is generally not known). Due to the inclusion of these scheduling parameters, LPV systems are linear in the state space, but nonlinear in the parameter space². Due to this property (linear differential inclusion²⁵), LPV systems are represented with a much simpler framework than full nonlinear models, being very similar to LTI models (and, thus, possessing many of the LTI advantages) - one can say that LPV embedding is somewhere in between the nonlinear and the LTI formalisms.

Due to the advantages of LPV embedding, it is indeed an interesting option for NMPC purposes²⁶. NMPC design based on LPV models has been formally investigated since the beginning of the 00's; some relevant papers are recalled:

- Explicit methods with stability and optimality guarantees have been addressed^{27,28}; the main downside in such methods is that, since the future values of the scheduling parameters are unknown, the algorithm ensures the constraints are satisfied for all possible system trajectories, which leads to conservative performances and (numerical-wise) high-demanding optimization procedures.
- Dynamic output feedback algorithms have also been developed^{29,30,31,32}. Some of these papers use an input/output LPV formulation. Anyhow, they are all robust with respect to ρ , solving worst-case (usually referred to as "min./max.") optimization procedures and resulting in conservative results.
- Other papers^{26,33,34} consider bounded limits upon the rates of variation of the scheduling parameters, which a major advance. This simple constraint simplifies the optimization procedure, treating the evolution of the scheduling parameters offline, via Linear Matrix Inequality (LMI) or ellipsoidal constraints.
- Frozen guesses for the scheduling parameter trajectory, that iterates according to measurements, have also been used³⁵, transforming the nonlinear optimization problem into a linear one. The issue that resides with such methods 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.

²Theoretical analysis of LPV system properties, such as stability, observability, controllability, often falls into the framework of linear time-varying systems or of nonlinear systems, which usually presents more difficulty compared to the classical LTI case^{23,24}.

- In previous works, the Authors have also developed a Quadratic Programming (QP) version of the LPV MPC algorithm using a frozen parameter trajectory guess^{36,37} and a scheduling trajectory estimation algorithm³⁸. In the latter, Lyapunov-decreasing terminal costs and control invariant sets are used to guarantee that feasibility with an enlarged domain of attraction (addressing the issue of possible infeasibility).

1.2 | Nonlinear Parameter Varying Systems

As evidenced, the LPV toolkit has allowed the extent of MPC techniques to the control of nonlinear processes without necessarily resorting to full-blown nonlinear programming optimization. The advances were made by using the linearity of the system and the known scheduling parameter to achieve simplified programs for the online part.

However, as also discussed in the parallel papers of this special issue, retaining (at least a portion of) the nonlinearity instead of reducing it to linearity by appropriately “hiding” it embedded to the scheduling parameters, can be more appropriate in some cases, leading to less conservativeness and over-approximations. Recent literature shows that there has been a growing concern in extending available LPV methods to the class of systems that have been progressively named Nonlinear Parameter Varying (NLPV),^{39,40,41,42}. NLPV have been brought to focus because in many practical applications, where the time-varying model part appears as an explicit nonlinear term^{43,44,45}.

For the above-mentioned reasons, this paper deals with MPC applied to the class of NLPV systems, focusing on those that contain an explicit Lipschitz nonlinearity. We must mention that MPC applied to NLPV systems has not yet been seen in the literature.

1.3 | Lipschitz Nonlinearities

This paper is specially interested in NLPV systems with Lipschitz nonlinearities because this class of nonlinearities appears in many real applications⁴⁶. As an example, when controlling vehicle suspension dampers, a hyperbolic tangent function appears due to the hysteresis behaviour of the damping fluid. This function, indeed, agrees with a Lipschitz condition⁴⁵.

The control⁴⁷ and observation of Lipschitz nonlinear systems^{48,49} has been analysed in several works. Tube-based MPC has been applied to nonlinear Lipschitz systems⁵⁰, but the NLPV design paradigm is not used, which essentially means that more complex nonlinear optimization procedure had to be implemented. The input-to-state stability of MPC with bounded uncertainties derived from a Lipschitz nonlinearity has also been assessed⁵¹. LMI techniques for stabilization of Lipschitzian nonlinear systems have been seen in^{52,53,45}.

1.4 | Contributions

With respect to the previous discussion and the identified literature gaps, this paper proposes a novel MPC design framework for NLPV systems with explicit Lipschitz nonlinear terms. More specifically, the contributions we put forward are:

1. The notions of Robust Positive Invariant Sets and Scheduled Robust Positive Invariant Sets are recalled (Section 2). Two Lemmas are presented to compute extended Lipschitz conditions for the NLPV system. Moreover, a Theorem is provided for computing Scheduled Robust Positive Invariant Sets for NLPV Lipschitz systems using LMIs.
2. Then, an MPC optimization procedure is formally proposed for the case of NLPV systems (Section 3), which uses contracting terminal ingredients bases on the previous Scheduled Robust Positive Invariant Sets. Moreover, the MPC stage cost function is provided for input-to-state stabilization.
3. Input-to-state stability and recursive feasibility analysis of the proposed MPC design method are analysed (Section 4).
4. Some practical issues on implementation details are discussed (Section 5).
5. A realistic simulation example considering the case of a vehicle suspension system is presented to analyse and discuss the effectiveness of the proposed method (Section 6).

Concluding remarks are drawn in Section 7.

1.5 | Notation

In this work, the set of nonnegative real number is denoted by \mathbb{R}_+ , whilst the set of nonnegative integers including zero is denoted by \mathbb{N} . The index set $\mathbb{N}_{[a,b]}$ represents $\{i \in \mathbb{N} \mid a \leq i \leq b\}$, with $0 \leq a \leq b$. \mathbb{I}_j denotes an identify matrix of size j , $\text{col}\{a, b, c\}$ denotes the vectorization (collection) of the entries and $\text{diag}\{v\}$ denotes the diagonal matrix generated with the line vector v . The predicted value of a given variable $v(k)$ at time instant $k+i$, made based on the information available at instant k , is denoted as $v(k+i|k)$.

For a given matrix P , $\alpha_{\min}\{P\}$ and $\alpha_{\max}\{P\}$ denote, respectively, its minimal and maximal eigenvalues.

\mathcal{K} refers to the class of positive and strictly increasing scalar functions that pass through the origin. A given function $f : \mathbb{R} \rightarrow \mathbb{R}$ is of class \mathcal{K} if $f(0) = 0$ and $\lim_{\xi \rightarrow +\infty} f(\xi) \rightarrow +\infty$.

C^n denotes the set of all compact convex subsets of \mathbb{R}^n . A convex and compact set $X \in C^n$ with non-empty interior, which contains the origin, is named a PC-set. A subset of \mathbb{R}^n is denoted a polyhedron if it is an intersection of a finite number of half spaces. A polytope is defined as a compact polyhedron. A polytope can be analogously represented as the convex hull of a finite number of points in \mathbb{R}^n . A hyperbox is a convex polytope where all the ruling hyperplanes are parallel with respect to their axes. Finally, consider two sets $A \subset \mathbb{R}^n$ and $B \subset \mathbb{R}^n$. A scalar set contraction ($0 < \lambda < 1$) or expansion ($\lambda > 1$) of A is given with $\lambda \in \mathbb{R}$ and $\lambda A = \{\lambda a \mid a \in A\}$. The Minkowski set addition is defined by $A \oplus B := \{a + b \mid a \in A, b \in B\}$, while the Pontryagin set difference is defined by $A \ominus B := \{a \mid a \oplus B \subseteq A\}$.

2 | PRELIMINARY CONSIDERATIONS

2.1 | Lipschitz NLPV Systems

Throughout this paper, we consider the following class of discrete-time affine NLPV system:

$$x(k+1) = A(\rho(k))x(k) + B(\rho(k))\Phi(x(k))u(k), \quad (1)$$

where $k \in \mathbb{N}$ represents the sampling instant, $x : \mathbb{N} \rightarrow \mathcal{X} \subset \mathbb{R}^{n_x}$ are the system states, $\rho : \mathbb{N} \rightarrow \mathcal{P} \subset \mathbb{R}^{n_p}$ is a scheduling parameter and $u : \mathbb{N} \rightarrow \mathcal{U} \subset \mathbb{R}^{n_u}$ is the control input. Matrices $A(\cdot)$ and $B(\cdot)$ are parameter-dependent and have appropriate dimensions. The sets \mathcal{X} and \mathcal{U} are named the feasibility sets for the states and for the control law, respectively; these sets define the operational constraints for this process, under closed-loop. \mathcal{X} contains the origin. The scheduling set \mathcal{P} defines the following constraints for the scheduling parameters: $\underline{\rho} \leq \rho(k) \leq \bar{\rho} \forall k \in \mathbb{N}$. The map $\Phi(x(k))$ defines an explicit nonlinearity along the input.

Assumption 1. This nominal NLPV system in Eq. (1) is such that the origin is a steady-state.

Assumption 2. The NLPV system in Eq. (1) has a structurally known explicit nonlinear map $\Phi(x(k))$ which obeys a local Lipschitz condition around $x(k)$. Assuming that the nonlinearity $\Phi(x) : \mathbb{R}^{n_x} \rightarrow \mathbb{R}^{n_x}$ is continuously differentiable with respect to x , the Lipschitz condition is given by:

$$\|\Phi(x) - \Phi(\hat{x})\| \leq \Gamma \|x - \hat{x}\|, \forall x \in \mathcal{X}, \forall \hat{x} \in \mathcal{X}, \quad (2)$$

where the smallest constant Γ that satisfies Eq. (2) is known as the Lipschitz constant for the nonlinearity $\Phi(\cdot)$.

Remark 1. In the majority of LPV models for nonlinear systems, the nonlinearities $\Phi(\cdot)$ are embedded into the scheduling parameters ρ . Much on the contrary, in an NLPV setting, they are willingly made explicit (outside of ρ), implying that the designer knows how they are physically derived. To assume that the designer knows the explicit structure of the nonlinearity $\Phi(\cdot)$ is not at all absurd, since they arise from phenomenological characteristics of the processes. In many cases, for example, when two physical systems are cascaded (actuator and actual process, for instance), and one of them has a phenomenologically known nonlinearity, this nonlinearity re-appears in the model of the complete cascaded block. Take a vehicle suspension system: the dynamics of the suspended vehicle chassis depend on the suspension control system (damper and spring), which inherently have known nonlinear behaviours - for the case of electro-rheological suspensions, this behaviour is a hyperbolic tangent⁴⁵; therefore $\Phi(x)$ is a hyperbolic tangent map.

Assumption 3. The considered class of NLPV systems given by Eq. (1) agrees with the following hypothesis:

- The states are measurable at each and every discrete-time instant $k \in \mathbb{N}$;

- The scheduling parameter ρ is also measurable or perfectly estimated at each and every discrete-time instant $k \in \mathbb{N}$;
- The feasibility set \mathcal{X} is a PC-set, which is defined as:

$$\mathcal{X} := \{x(k) \in \mathbb{R}^{n_x} \mid \underline{x} \leq x(k) \leq \bar{x} \forall k \in \mathbb{N}\}; \quad (3)$$

- The admissible control inputs belong to the PC-set \mathcal{U} , which is defined as:

$$\mathcal{U} := \{u(k) \in \mathbb{R}^{n_u} \mid |u| \leq u_{max} \forall k \in \mathbb{N}\}; \quad (4)$$

- The system is controllable, which means that the rank of matrix C_{trb} is full (i.e. equal to n_x) for all $x \in \mathcal{X}$ and all $\rho \in \mathcal{P}$, with:

$$C_{trb} = [B(\rho)\Phi(x) \ A(\rho)B(\rho)\Phi(x) \ A(\rho)^2B(\rho)\Phi(x) \ \dots \ A(\rho)^{n_x-1}B(\rho)\Phi(x)]; \quad (5)$$

- The system is quadratic stabilizable within the feasibility region defined by \mathcal{X} ;
- The system matrices depend affinely on the scheduling parameters, this is:

$$A(\rho(k)) = A_0 + \sum_{j=1}^{n_p} \rho_j(k)A_j, \quad B(\rho(k)) = B_0 + \sum_{j=1}^{n_p} \rho_j(k)B_j, \quad (6)$$

where ρ_j stands for the j^{th} element of the scheduling parameter vector $\rho(k) = \text{col}\{\rho_1(k), \dots, \rho_{n_p}(k)\}$ and A_j and B_j are constant matrices of appropriate sizes;

- The rate of variation of the scheduling variable is denoted by $\partial\rho(k) = \rho(k) - \rho(k-1)$. This rate is bounded by a slew constraint defined as $\partial\rho(k) \in \partial\mathcal{P} \forall k \in \mathbb{N}$, being $\partial\mathcal{P}$ a hyperbox defined as:

$$\partial\mathcal{P} := \{\partial\rho(k) \in \mathbb{R}^{n_p} \mid \overline{\partial\rho_j} \leq \partial\rho_j(k) \leq \underline{\partial\rho_j}, j \in \mathbb{N}_{[1, n_p]}\}. \quad (7)$$

In this work, we aim to develop a new framework for the design of an MPC algorithm for NLPV systems with an explicit Lipschitz nonlinearity along the input. As previously discussed, the predictive control paradigm is structured upon model-based predictions of the future output/state response of the system over a (sliding) prediction horizon of N_p steps ahead of the sampling instant k . For this reason, it is reasonably direct to understand that for (both LPV and) NLPV models, the values for the future scheduling parameters ahead, i.e. $\rho(k+j)$ with $j \in \mathbb{N}_{[1, N_p-1]}$, are needed to construct the state predictions $x(k+j|k)$. In practice, the only available value is of $\rho(k)$. Therefore, the following Assumptions are needed:

Assumption 4. The NLPV system in Eq. (1) is, in fact, quasi-NLPV. This means that the scheduling parameters ρ depend, at each and every sampling instant k , on a map of endogenous variables of the system, i.e.:

$$\rho(k) = f_\rho(x(k), u(k)). \quad (8)$$

Remark 2. Assumption 4 excludes from Eq. (1) those systems for which the scheduling parameters ρ are exogenous, such as activation signals, for instance.

Assumption 5. The collection of the scheduling parameters necessary to make state predictions for the next N_p steps are named scheduling sequence, denoted as $P_k = \text{col}\{\rho(k), \dots, \rho(k+N_p-1)\}$. This scheduling sequence, at each sampling instant k , is estimated accurately enough by some estimation scheme. Moreover, considering \hat{P}_k as this estimation/guess for the scheduling sequence, it is assumed that $\|P_k - \hat{P}_k\|_\infty < \mu_p$, with a small real bound μ_p . Since each vector $\rho(k)$ is bounded to $\mathcal{P} \forall k \in \mathbb{N}$, by definition, it is implied that $P_k \in \mathcal{P} \times \dots \times \mathcal{P} \subseteq \mathbb{R}^{(n_p)N_p}$, i.e.:

$$\underline{P} \leq P_k \leq \bar{P} \forall k \in \mathbb{N}. \quad (9)$$

Remark 3. The estimation of the scheduling sequence P_k (computed at each discrete-time instant) is, in fact, realizable for real-time applications with different kinds of estimation algorithms, as verified in the recent literature^{35,38,54}. This estimation, given by \hat{P}_k , is possible through recursive least-square algorithm³⁸, or even with iterative guessing methods³⁵. It is important to notice that good-quality estimation is usually yielded for the first few samples of P_k , while the quality gets corrupted for bigger horizons N_p . Nonetheless, \hat{P}_k will be used for the MPC framework at instant k and a new estimation \hat{P}_{k+1} is used at $k+1$. Since MPC uses a sliding horizon concept, this means that the portion of the scheduling sequence with less precision is not so important after all, given that the horizon slides and new information gets plugged to the control policy at each instant. For simplicity, in

the sequel, we will consider that $\hat{P}_k = P_k$. The admissible bound on the scheduling prediction μ_p will be used to compute the terminal ingredients of the MPC.

2.2 | Some Useful Definitions

There are a couple of useful definitions that will be needed to develop the NLPV MPC paradigm proposed in this paper. Firstly, we will recall the regular MPC online algorithm and the notions of stabilizability, input-to-state stability and recursive feasibility of processes regulated under MPC; these will serve later on for verification purposes of the proposed framework (in Section 4). Secondly, we will put forth the definitions of Scheduled Robust Positively Invariant Control Sets (and sequence of sets), that will be used as terminal ingredients of the proposed MPC tool for these Lipschitzian NLPV systems.

2.2.1 | The Regular State-Feedback MPC Algorithm

MPC policies are essentially obtained by solving an online optimization procedure, which takes into account a process model and states, outputs and input constraints. The (states/outputs) performance goals (such as reference tracking, disturbance rejection, sensitivity to noise and so forth) are written in the cost function of this optimization procedure. Below, the regular state-feedback MPC optimization problem¹ is given:

Problem 1. Standard MPC Optimization Procedure

$$\min_{U_k} J_{N_p} = \min_{U_k} \left(\sum_{i=1}^{N_p} (\ell(x, u, k)) + V(x(k + N_p|k)) \right) \quad (10)$$

$$\text{s.t. System Model: } x(k + i) = \mathcal{F}_x(x(k + i - 1), u(k + i - 1), \rho(k + i - 1)) \quad \forall i \in \mathbb{N}_{[1, N_p]}, \quad (11)$$

$$u(k + i - 1|k) \in \mathcal{U} \quad \forall i \in \mathbb{N}_{[1, N_p]}, \quad (12)$$

$$x(k + i|k) \in \mathcal{X} \quad \forall i \in \mathbb{N}_{[1, N_p]}, \quad (13)$$

$$x(k + N_p|k) \in \mathbb{X}^{N_r}, \quad (14)$$

where $U_k = \text{col}\{u(k|k), \dots, u(k + N_p - 1|k)\}$ is the vector which collects the sequence of control actions inside the prediction horizon N_p . The algorithm resides in solving the optimization Problem 1 at each discrete-time instant, applying the first entry of the control vector, i.e. $u(k|k)$, to the controlled plant, and then “rolling the horizon forward”, incrementing k .

The MPC stage cost J_{N_p} includes a terminal ingredient (stage value for x in the last step of the horizon, $k + N_p$) and a terminal set constraint, which are used to ensure feasibility, which will be discussed in the sequel.

When the system model is LTI, the process constraint function $\mathcal{F}_x(\cdot)$ is inherently linear and the optimization in Problem (1) becomes a regular constrained QP, which is easily tackled by standard solvers. For the studied case, since the system is NLPV, the optimization procedure will become nonlinear, even though the scheduling sequences P_k is assumed to be known/estimated along the horizon, since the Lipschitz term $\Phi(\cdot)$ is maintained in the NLPV formulation. In the LPV case, when P_k is available, Problem 1 is also translated as a QP, as discussed by the Authors in previous papers^{36,38}.

Remark 4. Load disturbances can be included into the optimization problem if they are known along the future horizon of N_p steps. Including them directly in the process model constraint results in a feedforward compensation of their behaviour and of their effects upon the state variable.

2.2.2 | Stability, Feasibility

There are three standard axioms that must be accounted for in order to guarantee recursive feasibility of MPC algorithms. Recursive feasibility means that if the optimization is feasible for a starting condition $x(0) = x_0$, it will also be so for following iterations. These axioms are recalled:

Definition 1. Recursive Feasibility of MPC Algorithms⁵⁵

Consider that the terminal set constraint on x exists such that $\mathbb{X}^{N_r} \subset \mathcal{X}$, with \mathcal{X} closed, convex and compact and that the origin

lies within the interior of Ω , for Ω being the largest admissible set³ such that $\Omega \subseteq \mathcal{X}$. Then, essentially, the following axioms verify if the MPC terminal cost function is Lyapunov-decreasing along the control horizon:

- A1) $\ell(\cdot) \geq \beta_1(\|x\|), \forall x \in \Omega, \forall u \in \mathcal{U}, \forall \rho \in \mathcal{P}$, for $\beta_1(\cdot)$ of class \mathcal{K}^4 .
- A2) $V(\cdot) \leq \beta_2(\|x\|), \forall x \in \mathbb{X}^{N_r}, \forall \rho \in \mathcal{P}$, for $\beta_2(\cdot)$ of class \mathcal{K}^5 .
- A3) $V(x(k+1)) - V(x(k) + \ell(x(k), u(k))) \leq 0, \forall x \in \mathbb{X}^{N_r}, \forall u \in \mathcal{U}, \forall \rho \in \mathcal{P}, \forall k^6$.

If these three axioms hold, the MPC will be recursively feasible for any starting condition $x_0 \in \mathcal{X}$.

The terminal set \mathbb{X}^{N_r} is given by $\{x \in \mathbb{R}^{n_x} \mid V(x) \leq \alpha_s\}$ such that $\mathbb{X}^{N_r} \subset \Omega$. Moreover, α_s is some scalar such that for all $x \in \Omega, \mathcal{F}_x(x, \kappa x, P_k) \in \mathbb{X}^{N_r}$.

Definition 2. Quadratic Stabilizability inside the Feasibility Region

The considered NLPV system given in Eq. (1) is said to be stabilizable if there exists a positive definite map $V : x(k) \rightarrow x(k)^T P x(k)$, where $P = P^T > 0$ and $P \in \mathbb{R}^{n_x \times n_x}$ and a state-feedback control policy of fashion $u(k) = \kappa x(k)$, with $\kappa \in \mathbb{R}^{n_u \times n_x}$, such that the following inequality:

$$V(A(\rho(k))x(k) + B(\rho(k))\Phi(x(k))\kappa x(k)) - V(x(k)) \leq -x(k)^T (Q + \kappa^T R \kappa) x(k) \quad (15)$$

holds for all $x \in \mathcal{X}$ and $\rho \in \mathcal{P}$, with $Q = Q^T > 0$ and $R = R^T > 0$. Then, the origin is globally exponentially stable within the feasibility region \mathcal{X} is globally exponentially stable for $x(k+1) \forall \rho \in \mathcal{P}$ and any initial condition $x_0 \in \mathcal{X}$.

Remark 5. The above notion of quadratic stabilizability “inside the feasibility region” is slightly smoother than the notion of pure quadratic stabilizability, which would require the verification of the inequality for all $x \in \mathbb{R}^{n_x}$. The notion of the feasibility regionalization implies that only \mathcal{X} must be considered, which may be *a priori* a smaller PC-set than \mathbb{R}^{n_x} (inequality must hold for all $x \in \mathcal{X}$ instead of \mathbb{R}^{n_x}).

The concept of input-to-state stability (ISS) is used to verify the stability and also allows control synthesis for nonlinear systems⁵¹. In this paper, we will use the concept of ISS generalized for discrete-time nonlinear process⁵⁶. Note that we will be concerned with input-to-state stability because the NLPV system in Eq. (1) will be controlled by a state-feedback MPC of form $u(k) = \kappa x(k)$, since the states are measurable (Assumption 3), which means that the states must be stabilized by the action implied by the predictive control policy.

Definition 3. Input-to-State Stability⁵⁶

Consider a generalized discrete-time nonlinear plant whose dynamics are given by:

$$x(k+1) = \mathcal{F}_x(x(k), u(k), w(k)), \quad (16)$$

where $x(k)$ is the state of the system and $u(k)$ is a bounded control input of the system, while $w(k)$ is a bounded model-process mismatch variable/disturbance such that $|w(k)| \leq w_{max} \forall k \in \mathcal{N}$. Then, this nonlinear system is said to be input-to-state stable if there exists a pair of \mathcal{K} -function $\beta_i(\cdot, \cdot)$ and $\sigma_i(\cdot)$ such that the following inequality holds:

$$\|x(k)\| \leq \beta_i(x(0), k) + \sigma_i(w_{max}) \quad (17)$$

Remark 6. We must stress that an ISS system is asymptotically stable in the absence of inputs u and w or if the inputs are time-decaying. Note that if the inputs are merely bounded, thus the evolution of the system states will be ultimately bounded in a set for which the size depends on the bounds of the inputs, which is quite logical.

Remark 7. The system described by Eq. (16) indeed encompasses the class of Lipschitzian NLPV processes as those in Eq. (1).

Definition 4. ISS-Lyapunov Functions⁵⁶

A continuous map $V(\cdot) : \mathbb{R}^{n_x} \rightarrow \mathbb{R}$ is denoted an ISS Lyapunov function for the nonlinear system described by Eq. (16) if there exists four class \mathcal{K} functions $\alpha_1(\cdot), \alpha_2(\cdot), \alpha_3(\cdot)$ and $\gamma_w(\cdot)$ such that the following inequalities hold:

$$\alpha_1(\|x\|) \leq V(x) \leq \alpha_2(\|x\|), \quad (18)$$

$$V(\mathcal{F}_x(x, u, w)) - V(x) \leq -\alpha_3(\|x\|) + \gamma_w(\|w\|). \quad (19)$$

³In fact, this set must also be positively control invariant; this definition is given in the sequel.

⁴This axiom implies that ℓ is function-wise lower bounded.

⁵This axiom implies that V is function-wise upper bounded.

⁶This axiom implies that V decreasing along the horizon.

Remark 8. We must raise attention to the question of equivalence between input-to-state stability and recursive feasibility. As evidenced, the inequality described for an ISS-Lyapunov function and those for the axioms of recursive feasibility are much alike. Anyhow, recursive feasibility does not necessarily imply in stabilization to the origin - because sub-optimally may guarantee the stabilization to some other steady-state equilibria near the origin - this is due to the fact that function $\alpha_3(\cdot)$ is usually equivalent to the MPC cost $\ell(\cdot)$ in the recursive feasibility inequalities, but the term $\gamma_w(\|w\|)$ is not present. For these reasons, in the sequel, we will investigate and verify recursive feasibility of the proposed algorithm *a priori* and, thence, the ISS property of the closed-loop system regulated with the MPC policy.

2.2.3 | Sets, Sequence of Sets

Definition 5. The 1-Step Robust Positive Invariant Set⁵⁷

The 1-Step set of \mathbb{X} is named $\mathcal{Q}_1\{\mathbb{X}\}$. The set $\mathcal{Q}_1\{\mathbb{X}\}$ stands for the set of states $x(k)$, considering the system in Eq. (1), that can be steered in a single sampling with an admissible control action $u(k) \in \mathcal{U}$ into the target set \mathbb{X} .

Definition 6. Robust Positive Invariant Set⁵⁷

A set $\mathbb{X} \subset \mathcal{X}$ is robust positively invariant (RPI) for the considered system in Eq. (1) with $u(k) = \kappa x(k) \in \mathcal{U}$ if, for all $x(k) \in \mathbb{X}$ and $\rho(k) \in \mathcal{P}$, $x(k+1) \in \mathbb{X}$. The definition of a RPI set holds if and only if $\mathbb{X} \subseteq \mathcal{Q}_1\{\mathbb{X}\}$.

Definition 7. Scheduled Robust Positive Invariant Set

A scheduled robust positively invariant (SRPI) set $\mathbb{X} \subset \mathcal{X}$ for the considered system in Eq. (1) with $u(k) = \kappa x(k) \in \mathcal{U}$ is a RPI computed based on the scheduling sequence (guess) from instant k , i.e. it is RPI $\forall p_j \in P_k$.

Definition 8. Finite-step Set Contraction Sequence⁵⁸

Let $N_r \geq 1$ be an integer and let $\lambda \in [0, 1)$. Take $\mathcal{S}^{N_r} = \{\mathbb{X}^1, \dots, \mathbb{X}^{N_r}\}$ as a sequence of PC-sets. This sequence is said a Finite-Step Set Contraction Sequence if and only if:

- \mathbb{X}^1 is a RPI set,
- $\mathbb{X}^j = \mathcal{Q}_1\{\mathbb{X}^{j+1}\} \forall j \in \mathbb{N}_{[1, N_r-1]}$,
- $\mathbb{X}^{j+1} \subset \mathbb{X}^j \forall j \in \mathbb{N}_{[0, N_r-1]}$ and
- $\mathbb{X}^{N_r} = \lambda \mathbb{X}^1$.

Remark 9. With respect to the previous notation, the set Ω is equivalent to \mathbb{X}^{N_r-1} , i.e. the one-step set from the terminal set \mathbb{X}^{N_r} .

Definition 9. Scheduled Robust Positive Invariant Sequence

A scheduled RPI sequence is a Finite-step sequence of N_r steps computed at instant k with respect to the available scheduling sequence P_k . This scheduled robust positive invariant sequence is denoted \mathcal{S}^{N_r} and congregates the set sequence through which x can be steered through, leaping from one set \mathbb{X}^j to the following \mathbb{X}^{j+1} , with feasible control actions $u(k) = \kappa x(k) \in \mathcal{U}$, until finally reaching the target invariant set \mathbb{X}^{N_r} . This scheduled RPI sequence agrees with all the necessities of the definition of a Finite-Step Set Contraction Sequence. Moreover, it is defined that \mathbb{X}^{N_r} is a SRPI set for the system in Eq. (1) for all $P_k \in \mathcal{P} \times \dots \times \mathcal{P} \subseteq \mathbb{R}^{(n_p)^{N_p}}$. By design, it is imposed that $N_r \leq N_p$.

2.3 | Preliminary Results

Now, we present some preliminary results (Lemmas and Theorems) that will be used to analyse and design the discussed MPC strategy for NLPV systems.

2.3.1 | RPI Sets

As shown and discussed in the prequel, and mentioned throughout the cited literature, RPI sets play a vital role in MPC and optimization-based control paradigms for feasibility concerns, since the terminal ingredient of the optimization must always be a RPI set. Now, we present the first (preliminary) results of this paper, which concern how to compute these scheduled RPI sets for the case of Lipschitzian NLPV systems. The following Lemma sets a result in terms of nominal boundaries and the Theorem presents an LMI-traceable solution for these set computations. With this Theorem, we provide conservative sufficient conditions

for the computation of a quadratic Lyapunov map $S(e(k)) = e^T(k)Pe(k)$ that can be used to compute RPI sets for the considered class of systems.

Lemma 1. Set Bounds, adapted from⁵⁹

Let $S : \mathbb{R}^{n_x} \rightarrow [0, \infty)$ be a continuously differentiable function and $s_1(\|e\|) \leq S(e) \leq s_2(\|e\|)$, with s_1 and s_2 as class \mathcal{K} functions and $e \in \mathcal{X}$. Suppose $u : \mathbb{N} \rightarrow \mathcal{R}^{n_u}$ is *a priori* chosen and there exist two constants $\gamma > 0$ and $\nu > 0$ such that:

$$S(e(k+1)) - \gamma S(e(k)) - \nu w^T w \leq 0, \quad \forall w \in \mathcal{W}, \quad (20)$$

where w represents some bounded uncertainty with maximal normed value w_{max} . Then, the system trajectories departing from $e(k_0) \in \Omega \subseteq \mathcal{X}$ will remain contained within this set Ω , where:

$$\Omega := \left\{ e \in \mathbb{R}^{n_x} \mid S(e) \leq \frac{\nu w_{max}^2}{1-\gamma} \right\}. \quad (21)$$

Proof. This proof is very straightforward: denote $S(k+1) = S(e(k+1))$ and $S(k) = S(e(k))$, for simplicity; apply the Final Value Theorem to Eq. (20) with $k \rightarrow \infty$, since the ‘‘input’’ variable w in this difference equation is ultimately bounded. This leads to $(1-\gamma)S(\infty) = \nu w^T w = \nu w_{max}^2$. Therefore, for whichever trajectory e follows along k , it holds that the maximal value S takes is bounded within Ω . This concludes this proof. \square

Remark 10. Consider that $x(k)$ are the actual system states and $\hat{x}(k|k)$ are state predictions; both these variables are limited to the feasibility set \mathcal{X} . Then, a state prediction error variable $e(k) = x(k) - \hat{x}(k|k)$ is also bounded to a constrained feasibility set \mathcal{E} , given by:

$$\mathcal{E} := \{e(k) \in \mathbb{R}^{n_x} \mid \underline{e} \leq e(k) \leq \bar{e} \forall k \in \mathbb{N}\}. \quad (22)$$

Theorem 1. Scheduled RPI Set for Lipschitzian NLPV Systems

Consider that Eq. (1) is used to make predictions for the behaviour for NLPV System with an explicit Lipschitz nonlinearity. Moreover, consider that the actual trajectories of this NLPV system behave according to Eq. (1), but also subject to some bounded additive uncertainty $w(k)$, with maximal normed value w_{max} , entering the system dynamics through some disturbance input matrix B_w , i.e. $x(k+1) = A(\rho(k))x(k) + B(\rho(k))\Phi(x(k))u(k) + B_w w(k)$. This uncertainty is used to represent the model-process mismatches due to the fact that a scheduling prediction guess P_k is used for the controller design⁷. Consider that an admissible control law $u(k) \in \mathcal{U}$ is applied to this system. Consider that Assumptions 1, 2, 3 and 4 hold albeit this additive uncertainty. Then, $e(k)$ represents the dynamics of the prediction error, given by the difference between the actual process and the prediction model ($\hat{x}(k+j|k) = \hat{x}(k+j)$), which is denoted as:

$$e(k+1) = A(\rho(k))e(k) + B(\Phi(x(k)) - \Phi(\hat{x}(k)))u(k) + B_w w(k). \quad (23)$$

Then, suppose that there exists a positive definite matrix affine on the scheduling parameters $X \in \mathbb{R}^{n_x \times n_x} = \sum_{j=1}^{n_p} \rho_j X_j$ and three scalars $\gamma > \gamma_0 > 0$ and $\nu > 0$ such that the two following inequalities hold for a fixed ρ_j :

$$\begin{bmatrix} (A(\rho_j)^T X A(\rho_j) - \gamma_0 X) & A(\rho_j)^T B_w \\ \star & -\nu I \end{bmatrix} \leq 0, \quad (24)$$

and

$$\Gamma \leq \frac{(\gamma - \gamma_0)\alpha_{max}\{P\}}{\|B\|u_{max}^2 (\alpha_{max}\{P\} + 2\|P\|\|A(\rho_j)\| + 2\|P\|\|B_w\|w_{max}^2)}. \quad (25)$$

If this is true, then the error system described by Eq. (23) is robust positively invariant within a set $\Omega \subseteq \mathcal{E}$ and the system trajectories departing from $e(k_0) = (x(k_0) - \hat{x}(k_0)) \in \Omega_e$ will remain contained inside this set, where:

$$\Omega_e := \left\{ e(k) \in \mathbb{R}^{n_x} \mid e^T(k)Pe(k) \leq \frac{\nu w_{max}^2}{1-\gamma} \right\}, \quad (26)$$

being $S(e(k)) := e^T(k)Pe(k)$, for $P = X^{-1}$. Moreover, the predicted system state trajectories $\hat{x}(k)$, described by Eq. (1), will be positively invariant within a set $\Omega \subseteq \mathcal{X}$ such that $\Omega = \mathcal{X} \ominus \Omega_e$, meaning that all nominal state trajectories departing from

⁷Note that w_{max} is linked to the bound on the scheduling prediction guess μ_p , as put forth in Lemma 2, in the sequel.

$\hat{x}(k_0)$ will be contained inside Ω . Finally, the actual state trajectories will be positively invariant within another set given by $\mathcal{X} \oplus \Omega_e$, which means that the state trajectories departing from $x(k_0)$ will be contained inside this robust set.

Proof. In this proof, we will denote A_j as $A(\rho_j)$, for notation simplicity. First of all, being $e(k+1)$ in Eq. (23) the prediction error system, we will denote $\epsilon(k+1)$ as a Lipschitzian-less error system, given by:

$$\epsilon(k+1) = A_j \epsilon(k) + B_w w(k). \quad (27)$$

Assume, then, that there exists an auxiliary Lyapunov function $\check{E}(k) = \check{E}(\epsilon(k)) := \epsilon^T(k) P \epsilon(k)$. It is implied that $\check{E}(k+1) = \epsilon^T(k+1) P \epsilon(k+1)$. Moreover, we assume that this auxiliary Lyapunov function is \mathcal{K} class envelope-bounded, i.e.:

$$\beta_{\epsilon_1}(\|\epsilon\|) \leq \check{E}(k) \leq \beta_{\epsilon_2}(\|\epsilon\|), \quad (28)$$

being $\beta_{\epsilon_1}(\cdot)$ and $\beta_{\epsilon_2}(\cdot)$ two class \mathcal{K} maps.

Then, we want to verify that $\check{E}(k)$ is indeed negative decrescent along the solution of $\epsilon(k)$, i.e. Eq. (27). For such, we define a map with respect to Lemma 1 so that we can bound the trajectories to a pre-defined set. This new map is obligatorily smaller or equal to zero, as gives:

$$H(k) = H(\check{E}(k)) := \check{E}(k+1) - \gamma_0 \check{E}(k) - v w^T(k) \epsilon(k) \leq 0. \quad (29)$$

Then, we develop:

$$\begin{aligned} H(k) &= \epsilon^T(k+1)^T P \epsilon(k+1) - \gamma_0 \epsilon^T(k) P \epsilon(k) - v w^T(k) w(k) \\ &= [A_j \epsilon(k) + B_w w(k)]^T P [A_j \epsilon(k) + B_w w(k)] \\ &\quad - \gamma_0 \epsilon^T(k) P \epsilon(k) - v w^T(k) w(k) \\ &= \epsilon^T(k) [A_j^T P A_j - \gamma_0 P] \epsilon(k) \\ &\quad + (w^T(k) B_w^T P A_j \epsilon(k)) + (\epsilon^T(k) A_j^T P B_w w(k)) - v w^T(k) w(k) \leq 0. \end{aligned} \quad (30)$$

Multiplying $H(k)$ from both sides with $\text{diag}\{P, \mathbb{1}\}$ and replacing P by X^{-1} , we obtain the following LMI:

$$\begin{bmatrix} (A(\rho_j)^T X A(\rho_j) - \gamma_0 X) & A(\rho_j)^T X B_w \\ \star & -v \end{bmatrix} \leq 0. \quad (31)$$

Then, multiplying this LMI from the left and right sides with $[\epsilon(k) \ w(k)]$ and $[\epsilon^T(k) \ w^T(k)]^T$, respectively, it becomes clear that ensuring Eq. (31) is sufficient to ensure that $H(\epsilon(k)) \leq 0$. Due to Lemma 1, there exists a positively invariant set Ω_0 such that the system in Eq. (27) is robustly invariant, where $\Omega_0 := \left\{ \epsilon(k) \in \mathbb{R}^{n_x} \mid \epsilon^T(k) P \epsilon(k) \leq \frac{v w_{\max}^2}{1-\gamma_0} \right\}$. From this, we continue. Denote $E(k) = E(\epsilon(k)) := \epsilon^T(k) P \epsilon(k)$ as the ‘‘real’’ Lyapunov map. Assume that it also agrees to a \mathcal{K} class envelope-bounded equivalent to Eq. (28). Then, to ensure an negative decrescent along its solution given by Eq. (23), we denote:

$$M(k) = M(E(k)) = E(k+1) - \gamma E(k) - v w^T(k) w(k) \leq 0. \quad (32)$$

Thus, we develop:

$$\begin{aligned} M(k) &= [A_j \epsilon(k) + B(\Phi(x(k)) - \Phi(\hat{x}(k))) u(k) + B_w w(k)]^T P [A_j \epsilon(k) + B(\Phi(x(k)) - \Phi(\hat{x}(k))) u(k) + B_w w(k)] \\ &\quad - \gamma \epsilon^T(k) P \epsilon(k) - v w^T(k) w(k) \\ &= \left(\epsilon^T(k) A_j^T P A_j \epsilon(k) + \epsilon^T(k) A_j^T P B_w w(k) + w^T(k) B_w^T P A_j \epsilon(k) + w^T(k) B_w^T P B_w w(k) \right) - \gamma \epsilon^T(k) P \epsilon(k) \\ &\quad + (u^T(k) (\Phi(x(k)) - \Phi(\hat{x}(k))))^T B^T P B (\Phi(x(k)) - \Phi(\hat{x}(k))) u(k) \\ &\quad + \left(\epsilon^T(k) A_j^T P B (\Phi(x(k)) - \Phi(\hat{x}(k))) u(k) + u^T(k) (\Phi(x(k)) - \Phi(\hat{x}(k)))^T B^T P A_j \epsilon(k) \right) \\ &\quad + (w^T(k) B_w^T P B (\Phi(x(k)) - \Phi(\hat{x}(k))) u(k) + u^T(k) (\Phi(x(k)) - \Phi(\hat{x}(k)))^T B^T P B_w w(k)). \end{aligned} \quad (33)$$

The **first** term of the above Equation is easily identified as $H(\epsilon(k)) + \gamma_0 \epsilon^T(k) P \epsilon(k)$, the **second** is maintained, while the **third** and **fourth** are symmetric and can be simplified, as follows:

$$\begin{aligned}
M(k) = & \left(H(e(k)) + (\gamma - \gamma_0)e^T(k)Pe(k) \right) \\
& + \left(u^T(k) (\Phi(x(k)) - \Phi(\hat{x}(k)))^T B^T PB (\Phi(x(k)) - \Phi(\hat{x}(k))) u(k) \right) \\
& + \left(2u^T(k) (\Phi(x(k)) - \Phi(\hat{x}(k)))^T B^T PA_j e(k) \right) \\
& + \left(2u^T(k) (\Phi(x(k)) - \Phi(\hat{x}(k)))^T B^T PB_w w(k) \right).
\end{aligned} \tag{34}$$

By definition, it is implied that the **second** term agrees with the following inequality:

$$\alpha_{\min}\{P\} \leq \frac{\left(u^T(k) (\Phi(x(k)) - \Phi(\hat{x}(k)))^T B^T PB (\Phi(x(k)) - \Phi(\hat{x}(k))) u(k) \right)}{\|B\| \|u(k)\| \|(\Phi(x(k)) - \Phi(\hat{x}(k)))\|} \leq \alpha_{\max}\{P\}. \tag{35}$$

Therefore, it follows that:

$$\left(u^T(k) (\Phi(x(k)) - \Phi(\hat{x}(k)))^T B^T PB (\Phi(x(k)) - \Phi(\hat{x}(k))) u(k) \right) \leq \alpha_{\max}\{P\} \|B\| \|u(k)\| \|(\Phi(x(k)) - \Phi(\hat{x}(k)))\|, \tag{36}$$

since the Lipschitz condition given by Eq. (2) holds and $\|u\| \leq u_{\max}^2$, it follows that:

$$\begin{aligned}
\left(u^T(k) (\Phi(x(k)) - \Phi(\hat{x}(k)))^T B^T PB (\Phi(x(k)) - \Phi(\hat{x}(k))) u(k) \right) & \leq \|B\| \alpha_{\max}\{P\} u_{\max}^2 \Gamma \|e(k)\| \\
& \leq \|B\| \alpha_{\max}\{P\} u_{\max}^2 \Gamma \|e(k)\|^2.
\end{aligned} \tag{37}$$

Thereof, and since $H(e(k)) \leq 0$, we can state that:

$$\begin{aligned}
M(k) \leq & \left((\gamma - \gamma_0)e^T(k)Pe(k) \right) \\
& + \|B\| \alpha_{\max}\{P\} \Gamma \|e(k)\|^2 u_{\max}^2 \\
& + \left(2u^T(k) (\Phi(x(k)) - \Phi(\hat{x}(k)))^T B^T PA_j e(k) \right) \\
& + \left(2u^T(k) (\Phi(x(k)) - \Phi(\hat{x}(k)))^T B^T PB_w w(k) \right) \leq 0.
\end{aligned} \tag{38}$$

Moving to the latter terms, we develop:

$$\begin{aligned}
\left(2u^T(k) (\Phi(x(k)) - \Phi(\hat{x}(k)))^T B^T PA_j e(k) \right) & \leq 2u^T(k) (\Phi(x(k)) - \Phi(\hat{x}(k)))^T B^T PA_j e(k) e^T(k) A_j^T PB (\Phi(x(k)) - \Phi(\hat{x}(k))) u(k) \\
& \leq 2\|B\| \|u(k)\| \|(\Phi(x(k)) - \Phi(\hat{x}(k)))\| \|P\| \|A_j\| \|e(k)\| \\
& \leq 2\|B\| \Gamma \|P\| \|A_j\| \|e(k)\|^2 u_{\max}^2.
\end{aligned}$$

Equivalently,

$$\begin{aligned}
\left(2u^T(k) (\Phi(x(k)) - \Phi(\hat{x}(k)))^T B^T PB_w w(k) \right) & \leq 2u^T(k) (\Phi(x(k)) - \Phi(\hat{x}(k)))^T B^T PB_w w(k) w^T(k) B_w^T PB (\Phi(x(k)) - \Phi(\hat{x}(k))) u(k) \\
& \leq 2\|B\| \Gamma \|P\| \|B_w\| \|e(k)\|^2 u_{\max}^2 w_{\max}^2.
\end{aligned}$$

From these developments, we obtain:

$$\begin{aligned}
M(k) \leq & \left((\gamma - \gamma_0) \alpha_{\max}\{P\} \|e\|^2 \right) \\
& + \|B\| \alpha_{\max}\{P\} \Gamma \|e(k)\|^2 u_{\max}^2 \\
& + \left(2\|B\| \Gamma \|P\| \|A_j\| \|e(k)\|^2 u_{\max}^2 \right) \\
& + \left(2\|B\| \Gamma \|P\| \|B_w\| \|e(k)\|^2 u_{\max}^2 w_{\max}^2 \right) \leq 0.
\end{aligned} \tag{39}$$

Finally, since $\|e(k)\|^2$ is always positive, it remains that:

$$\Gamma \leq \frac{(\gamma - \gamma_0) \alpha_{\max}\{P\}}{\|B\| u_{\max}^2 (\alpha_{\max}\{P\} + 2\|P\| \|A_j\| + 2\|P\| \|B_w\| w_{\max}^2)}. \tag{40}$$

Because of Lemma 1, it is sufficient to conclude that Ω_e , defined in terms of P , is a (scheduled) positively invariant set for the error system.

By the definition of the error system, with respect to Remark 10, it is direct to see that the (scheduled) positively invariant set for the nominal Lipschitzian NLPV system is given by the Pontryagin set difference between the state feasibility set and SRPI

set for $e(k)$, while for the real system by a Minkowski set addition, as follows:

$$\begin{aligned}\hat{x}(k) &= x(k) - e(k), \\ e(k) &\in \Omega_e, \\ \hat{x}(k), x(k) &\in \mathcal{X}, \\ \hat{x}(k) &\in (\mathcal{X} \ominus \Omega_e). \\ x(k) &\in (\mathcal{X} \oplus \Omega_e).\end{aligned}$$

This concludes the proof. \square

Remark 11. From the viewpoint of the MPC controller, the actual system trajectories are not known. Therefore, only $(\mathcal{X} \ominus \Omega_e)$ is used to bind the terminal value for the predicted system states, as gives constraint (14) in Problem 1.

Finally, we present a Lemma that relates the bounds on the estimation of the scheduling sequence, μ_p to the bounds on the disturbance variable $w(k)$.

Lemma 2. The bounds on the disturbance variable w_{max} can be given in terms of the bounds on the error of the scheduling sequence as follows:

$$w_{max} = A^{N_p}(\mu_p)\bar{e} + B^{N_p}(\mu_p)\Gamma\bar{e} \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix}_{(N_p+1)\times 1}^T u_{max} \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix}_{N_p \times 1}^T. \quad (41)$$

Proof. Firstly, we demonstrate how the actual state trajectories behave for the first two steps ahead of k and generalize them to j steps ahead:

$$x(k+1) = A(\rho(k))x_k + B(\rho(k))\Phi(x_k)u(k), \quad (42)$$

$$\begin{aligned}x(k+2) &= A(\rho(k+1))x(k+1) + B(\rho(k+1))\Phi(x(k+1))u(k+1), \\ &= A(\rho(k+1))A(\rho(k))x_k + A(\rho(k+1))B(\rho(k))\Phi(x_k)u(k) \\ &\quad + B(\rho(k+1))\Phi(x(k+1))u(k+1).\end{aligned} \quad (43)$$

We denote $\bar{X}_{k,j}$ as the collection of states from x_k until $x(k+j)$, i.e.: $\bar{X}_{k,j} = \text{col}\{x_k, x(k+1), \dots, x(k+j)\}$ and $U_{k,j}$ as the collection of control input $u(k+j)$ from $u(k)$ until $u(k+j-1)$, i.e. vector U_k truncated at the $(j-1)$ -th position. Note that $\bar{X}_{k,j} = \bar{A}_{k,j}x(k)$ and, ultimately:

$$\bar{X}_{k,j} \leq \bar{X}_{k,j} \leq \bar{X}_{k,j}, \quad (44)$$

$$\bar{A}_{k,j} \leq \bar{A}_{k,j} \leq \bar{A}_{k,j}. \quad (45)$$

We also extended the Lipschitz nonlinearity vector-wise, this is:

$$\Phi(\bar{X}_{k,j}) = \text{col}\{\Phi(x_k), \Phi(x(k+1)), \dots, \Phi(x(k+j))\}, \quad (46)$$

for which the Lipschitz condition also holds:

$$\|\Phi(\bar{X}_{k,j}) - \Phi(\bar{X}_{k,j})\| \leq \Gamma \|\bar{X}_{k,j} - \bar{X}_{k,j}\|. \quad (47)$$

Therefore, it follows that:

$$x(k+j) = A^j(P_k)x_k + B^j(P_k)\Phi(\bar{X}_{k,j})U_{k,j}, \quad (48)$$

being⁸:

$$A^j(P_k) = \prod_{i=0}^{j-1} A(\rho(k+i)) \quad \text{and} \quad (49)$$

$$B^j(P_k)\Phi(\bar{X}_{k,j})U_{k,j} = \sum_{m=1-k}^{j-k} \left(\prod_{n=k+i}^{m+1} [A(\rho(n))] B(\rho(j-m))\Phi(x(j-m))u(j-m) \right). \quad (50)$$

⁸ \prod stands for the left-side matrix product.

Now, take the state deviance variable and its trajectories:

$$\begin{aligned} e(k+j) &= x(k+j) - \hat{x}(k+j|k) \\ &= (A^j(P_k)x_k - A^j(\hat{P}_k)\hat{x}_k) \\ &\quad + \left(B^j(P_k)\Phi(\bar{X}_{k,j}) - B^j(\hat{P}_k)\bar{X}_{k,j} \right) U_{k,j}, \end{aligned} \quad (51)$$

since $A(\cdot)$ and $B(\cdot)$ are defined as affine on ρ , we can develop:

$$e(k+j) = A^j(P_k - \hat{P}_k)e_k + \left(B^j(P_k)\Phi(\bar{X}_{k,j}) - B^j(\hat{P}_k)\bar{X}_{k,j} \right) U_{k,j}, \quad (52)$$

which, due to the Lipschitz condition, holds as:

$$e(k+j) \leq A^j(\mu_p)e_k + B^j(\mu_p)\Gamma\bar{E}_{k,j}U_{k,j}. \quad (53)$$

Alternatively, with respect to Remark 10, we can write:

$$\underline{e} \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix}_{(j+1) \times 1}^T \leq \bar{E}_{k,j} \leq \bar{e} \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix}_{(j+1) \times 1}^T \quad \forall k \in \mathbb{N}, \quad (54)$$

$$-u_{max} \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix}_{j \times 1}^T \leq U_{k,j} \leq u_{max} \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix}_{j \times 1}^T \quad \forall k \in \mathbb{N}. \quad (55)$$

Since, with respect to Lemma 1 and Theorem 1, we need to express the upper bound on a disturbance variable $w(k)$, it holds, by definition, that $e(k+j) = B_w w(k+j-1)$, for which, the ultimate upper bound is:

$$e(k+N_p) \leq A^{N_p}(\mu_p)\bar{e} + B^{N_p}(\mu_p)\Gamma\bar{e} \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix}_{(N_p+1) \times 1}^T u_{max} \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix}_{N_p \times 1}^T. \quad (56)$$

Finally, it follows that:

$$B_w = \mathbb{1}_{n_x}, \quad (57)$$

$$w_{max} = A^{N_p}(\mu_p)\bar{e} + B^{N_p}(\mu_p)\Gamma\bar{e} \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix}_{(N_p+1) \times 1}^T u_{max} \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix}_{N_p \times 1}^T, \quad (58)$$

which concludes this proof. \square

2.3.2 | Extended Lipschitz Constraints

Now, finally, two Lemmas are presented to extended the Lipschitz condition for the complete state transition map $\mathcal{F}_x(\cdot)$ and to bind the state estimation error $e(k+j)$ with respect to w_{max} . These Lemmas will be, later on, used to verify the ISS property of the closed-loop system.

Lemma 3. Extended Lipschitz Condition

Consider the NLPV system in Eq. (1) such that Assumptions 1 and 2 are satisfied. Since $\Phi(x(k))$ in Eq. 1 is locally Lipschitz in $x(k)$, it holds that the complete nonlinear state transition map $\mathcal{F}_x(x(k), u(k), \rho(k)) = A(\rho(k))x(k) + B(\rho(k))\Phi(x(k))u(k)$ is locally Lipschitz in $x(k)$ in the domain $\mathcal{X} \times \mathcal{U} \times \mathcal{P}$, i.e. there exists a constant $0 < \Gamma_e < +\infty$ such that for all $x, \hat{x} \in \mathcal{X}$, $u \in \mathcal{U}$ and $\rho \in \mathcal{P}$,

$$\|\mathcal{F}_x(x, u, \rho) - \mathcal{F}_x(\hat{x}, u, \rho)\| \leq \Gamma_e \|x - \hat{x}\|. \quad (59)$$

Proof. This Lemma is demonstrated by a trivial extension of the local Lipschitz condition of $\Phi(\cdot)$ to $\mathcal{F}_x(\cdot)$:

$$\begin{aligned} \|\mathcal{F}_x(x, u, \rho) - \mathcal{F}_x(\hat{x}, u, \rho)\| &= \|(A(\rho)x + B(\rho)\Phi(x)u) - (A(\rho)\hat{x} + B(\rho)\Phi(\hat{x})u)\| \\ &= \|A(\rho)(x - \hat{x}) + B(\rho)(\Phi(x) - \Phi(\hat{x})u)\| \\ \|\mathcal{F}_x(x, u, \rho) - \mathcal{F}_x(\hat{x}, u, \rho)\| &\leq \|A(\rho)(x - \hat{x})\| + \|B(\rho)(\Phi(x) - \Phi(\hat{x})u)\| \\ &\leq A(\bar{\rho})\|x - \hat{x}\| + B(\bar{\rho})\|\Phi(x) - \Phi(\hat{x})\|\bar{u} \\ &\leq A(\bar{\rho})\|x - \hat{x}\| + B(\bar{\rho})\Gamma\|x - \hat{x}\|\bar{u} \\ &\leq \Gamma_e\|x - \hat{x}\|, \end{aligned}$$

for $\Gamma_e = (A(\bar{\rho}) + B(\bar{\rho})\Gamma\bar{u})$, which concludes this proof. \square

Lemma 4. Adapted from⁵¹

Consider the NLPV system in Eq. (1) such that Assumption 2 and Lemma 3 are satisfied. Assume, once again, that the actual trajectories of this NLPV system behave according to Eq. (1), but also subject to some bounded additive uncertainty $w(k)$, with maximal normed value w_{max} , which enters the system dynamics through some disturbance input matrix B_w , i.e. $x(k+1) = A(\rho(k))x(k) + B(\rho(k))\Phi(x(k))u(k) + B_w w(k)$. This uncertainty is used to represent the model-process mismatches due to the fact that a scheduling prediction guess P_k is used for the controller design. We will denote $w(k) \in \mathcal{W}$, $|w(k)| \leq w_{max} \forall k \in \mathbb{N}$. Then, for a given sequence of inputs, the difference between the nominal prediction of the states $\hat{x}(k+j|k)$ (that disregards $w(k)$) and the real states of the system $x(k+j)$ is bounded by:

$$\|x(k+j) - \hat{x}(k+j|k)\| = \|e(k+j)\| \leq \frac{\Gamma_e^j - 1}{\Gamma_e - 1} w_{max}, \quad (60)$$

where Γ_e is the extended Lipschitz constant of the system, with respect to Lemma 3.

Proof. This proof is lead using the triangular inequality, as follows:

$$\begin{aligned} \|x(k+1) - \hat{x}(k+1|k)\| &= \|B_w w(k)\| \leq B_w w_{max}, \\ \|x(k+2) - \hat{x}(k+2|k)\| &\leq \Gamma_e \|x(k+1) - \hat{x}(k+1|k)\| + \|w(k+1)\| \leq (\Gamma_e + 1)B_w w_{max}, \\ &\vdots \\ \|x(k+j) - \hat{x}(k+j|k)\| &\leq \Gamma_e \|x(k+j-1) - \hat{x}(k+j-1|k)\| + \|w(k+j-1)\| \leq \sum_{i=0}^{j-1} \Gamma_e^i B_w w_{max} \\ &= \frac{\Gamma_e^j - 1}{\Gamma_e - 1} B_w w_{max}. \end{aligned}$$

Due to Lemma 2, we take $B_w = \mathbb{1}_{n_x}$, which concludes this proof. \square

3 | MPC FOR LIPSCHITZIAN NLPV SYSTEMS

Bearing in mind the preliminary results, this Section discusses the proposed MPC method for Lipschitzian NLPV systems on the form of Eq. (1).

The proposed control algorithm resides in solving Problem 1, where $\mathcal{F}_x(\cdot)$ represents the nominal prediction model $\hat{x}(k+i)$ on the basis of Eq. (1). In the sequel, we show the used functions for the horizon cost $\ell(\cdot)$ (i.e. main MPC cost) and terminal ingredient $V(\cdot)$, used to construct the complete MPC stage cost J_{N_p} .

3.1 | Horizon Cost

The horizon cost is used to include performance goals to the MPC control policy. In this paper, we follow the structure of the majority of costs in regular MPC designs⁶⁰, where we weight the quadratic difference between the states and the target state together with the quadratic difference between the control signals along the horizon and the steady-state control input. This is:

$$\begin{aligned} \ell(x(k+i|k), u(k+i-1|k)) &= \|x(k+i|k) - x_s\|_Q^2 + \|u(k+i-1|k) - u_s\|_R^2 \\ &= (x(k+i|k) - x_s)^T Q (x(k+i|k) - x_s) + (u(k+i-1|k) - u_s)^T R (u(k+i-1|k) - u_s), \end{aligned} \quad (61)$$

where $p_s = (x_s, u_s)$ represents a steady-state target reference; Q and R are constant matrices of appropriate dimensions. Dimension and form-wise, Q is strictly defined positive with $\text{rank}\{Q\} = n_x$.

This kind of horizon cost $\ell(\cdot)$ explicitly includes a tracking objective, since the optimization procedure will force values of x (along the horizon) which are longest away from x_s to be further forced towards x_s . Moreover, disturbance rejection is implicitly included for this same reason: albeit some load disturbance forcing x to move away from x_s , u will be chosen such that this effect is corrected. The values for matrices Q and R are designer degrees-of-freedom, which can be tuned to achieve the aimed closed-loop responses: increasing Q forces tighter offset-free tracking performances, whilst decreasing R lets the control action move more loosely.

3.2 | Terminal Cost

The terminal cost is also taken as suggests⁶⁰, penalizing the difference between the target state reference and the last state prediction along the horizon, this is:

$$V(\cdot) = \|x(k + N_p|k) - x_s\|_P^2 = (x(k + N_p|k) - x_s)^T P (x(k + N_p|k) - x_s). \quad (62)$$

In this terminal cost, P is a constant matrix of adequate sizes taken according to LMI (24) and ensuring ISS property and recursive feasibility (explained in the sequel). P is also strictly positive definite with $\text{rank}\{P\} = n_x$.

3.3 | Closed-Loop Behaviour

Proposition 1. Steady-State Target Reference

The target operation point p_s is indeed an admissible steady-state for the system in Eq. (1).

Proof. Since this system is assumed to be quadratic stabilizable within the feasibility region defined by \mathcal{X} (property which is verified in Proposition 2), if $x_s \in \mathcal{X}$ (is admissible), then u_s is also feasible and given by:

$$B(\rho_s)\Phi(x_s)u_s = (\mathbb{I}_{n_x} - A(\rho_s))x_s, \quad (63)$$

for any fixed $\rho_s \in \mathcal{P}$. □

Lemma 5. Closed-loop Equilibrium

If the system is indeed quadratic stabilizable, recursively feasible and input-to-state stable⁹, and if p_s is an admissible steady-state target contained within the tracking set $\mathcal{T} = \mathcal{X} \oplus \mathcal{U}$, then it is an asymptotically stable point in closed-loop. Elsewise, the achieved closed-loop equilibrium is given by: $p_s^* = (x_s^*, u_s^*) = \arg \min_{x_s^*} V(x_s^*)$.

Proof. The tracking pairs are given by $p(k) = (x(k+i|k), u(k+i-1|k))$ and, therefore, the tracking set is given by the Minkowski set addition between the feasibility sets on the states and on the control inputs, i.e. $\mathcal{T} = \mathcal{X} \oplus \mathcal{U}$. Assume that p_s belongs to \mathcal{T} (i.e. is admissible). Then, from any initial condition x_0 contained within \mathcal{X} , recursive feasibility is guaranteed and, since the system is quadratic stabilizable and ISS property is verified, the proposed controlled with $u(k) = \kappa x(k)$ asymptotically steers the system to x_s in an admissible way. Elsewise, if p_s is not admissible and does not belong to \mathcal{T} , the terminal stage cost $V(\cdot)$ penalizes the final step for x inside the horizon, at $x(k + N_p)$, the closest x_s^* to x_s will be tracked and, thus, since recursive feasibility is ensured, the minimal cost $\min_{U_{N_p}} J_{N_p}$ found by the optimization procedure is $\min_{x_s^*} \|x_s^* - x_s\|_P^2$, being x_s^* the achieved steady-state $x(k + N_p) = x_s^*$, which ends this proof. □

3.4 | Terminal Constraint

With the previously detailed horizon cost $\ell(\cdot)$ and terminal cost $V(\cdot)$, the proposed MPC algorithm forces the convergence of $x(k+i)$ towards a reference steady-state target x_s . Anyhow, the model used in the optimization procedure of Problem 1 has a mismatch to the real process, since it is based on the scheduling sequence estimate \hat{P}_k , which differs from the actual scheduling sequence P_k (unknown). Because of the model uncertainty $|w(k)| \leq w_{max} \forall k \in \mathbb{N}$, a contractive terminal set constraint is used to ensure the states converge to a final target set containing x_s .

⁹These three conditions verified in the sequel.

MPC design couples to the use of positively invariant set sequences is widely used throughout the literature to ensure the algorithm guarantees asymptotic convergence despite these model-process uncertainties $u(k)$. To compute these set sequences, the bounds on rate of variation of the scheduling parameters $\partial\rho(k)$ are taken into account.

At each sampling instant k , a finite-step set contraction sequence, according to Definition 8, is computed (with $N_r = N_p$). This sequence S^{N_p} has the first element $S^{N_p}\{1\} = \mathbb{X}^1$ as a SRPI set for the NLPV system at instant k , containing the target goal x_s , and a final set element $S^{N_p}\{N_p\} = \mathbb{X}^{N_p}$ that still contains the target goal and is a contracted version of the first element, i.e. $\mathbb{X}^{N_p} = \lambda\mathbb{X}^1$. It follows that:

- At each sampling instant k , \mathbb{X}^1 is computed as a SRPI set for the Lipschitzian NLPV system in Eq. (1), computed according to Theorem 1, such that $\mathbb{X}^1 = \mathcal{X} \ominus \mathbb{X}_e^1$:

$$\mathbb{X}_e^1 := \left\{ e(k) \in \mathbb{R}^{n_x} \mid e^T(k) P e(k) \leq \frac{v w_{max}^2}{1 - \gamma} \right\}. \quad (64)$$

- It must hold that $x_s \in \mathbb{X}^1$;
- The final SRPI set \mathbb{X}^{N_p} is a contracted version of \mathbb{X}^1 , computed as $\lambda\mathbb{X}^1$ with a contraction variable $\lambda \in [0, 1)$ so that there exist $(N_p - 1)$ “in between” sets such that each set is the one-step RPI set from each other, i.e. $\mathbb{X}^j = \mathcal{Q}_1\{\mathbb{X}^{j+1}\}$ and $\mathbb{X}^{j+1} \subset \mathbb{X}^j, \forall j \in \mathbb{N}_{0, N_p-2}$.

To compute the final SRPI and the contraction variable λ , the following Corollary is used:

Corollary 1. The N_p -steps-ahead SRPI set is computed on the basis of Theorem 1, but for $x(k + N_p)$. Since \hat{P}_k gives an estimation for $\rho(k + N_p - 1)$ (that takes into account the rates of variation of scheduling parameters, i.e. $\rho(k) + (N_p - 1)\partial\rho \leq \rho(k + N_p - 1) \leq \rho(k) + (N_p - 1)\overline{\partial\rho}$), it suffices to solve the LMIs in this Theorem using $\rho_j = \rho(k + N_p - 1)$ to compute \mathbb{X}^{N_p} . Then, since \mathbb{X}^1 is computed for $\rho(k)$, the contraction variable is found directly.

With these previous developments in mind, to guarantee that within N_p steps from the initial instant k_0 the controlled system reaches a SRPI set \mathbb{X}^{N_p} , the following contractive terminal set constraint is included to the MPC design (instead of constraint (14)):

$$x(k + N_p | k) \in S^{N_p}\{j\}, \quad j = \max\{N_p - k, 0\}, \quad (65)$$

assuming that the finite-step set sequence S^{N_p} is available. Note that this terminal set $S^{N_p}\{j\}$ is equal to the larger \mathbb{X}^0 at the initial instant k being shrunk subsequently until, at $k + N_p$, it becomes the smallest set \mathbb{X}^{N_p} . This constraint makes the proposed MPC algorithm intrinsically time-varying, since, at least for the first N_p samples, the sets are contracting.

4 | STABILITY AND RECURSIVE FEASIBILITY ANALYSIS

In this Section, we verify Quadratic Stabilizability, Recursive Feasibility and Input-to-State Stability properties of the proposed NLPV MPC algorithm.

Assumption 6. (i) There exists a \mathcal{K} function $\beta_1(\|x\|)$ that lower bounds the horizon cost $\ell(x)$; and (ii) there exists another \mathcal{K} function $\beta_2(\|x\|)$ that upper bounds the terminal cost $V(x(k + N_p))$.

Proposition 2. Quadratic Stabilizability inside the Feasibility Region

The NLPV system given in Eq. (1) and regulated by the MPC policy with $u(k) = \kappa x(k)$ is quadratic stabilizable.

Remark 12. The MPC policy yields a quadratic stabilizability property if the third recursive feasibility axiom verifies, with $V = x(k)^T P x(k)$, as demonstrated in the sequel.

Proposition 3. Recursive Feasibility

The proposed algorithm is recursively feasible inside the feasibility set for any starting condition $x(k_0) = x_0 \in \mathcal{X}$.

Proof. Since only the measured (state-feedback) variable $x(k)$ and scheduling sequence P_k are used to solve Problem 1, at the following discrete-time instants $k > k_0 = 0$, the online optimization is not related to any disturbance variables and, thus, the recursive feasibility property can be analysed albeit disturbances (feedforward compensation is neglected).

Assume that Problem 1 is feasible for an initial condition x_0 , based on a P_{k_0} scheduling sequence, resulting in $U_{k_0}^*$ as the optimal sequence of control action which solves Eq. (10) at instant k_0 ; this optimal control policy leads to a minimal state sequence \bar{X}_{k_0, N_p}^* with respect to the cost function $J_{N_p}(\cdot)$. It holds that $x^*(k_0 + i) \in \mathcal{S}^{N_p}\{i\}$ and $u(k_0 + i - 1|k_0) \in \mathcal{U} \forall i \in \mathbb{N}_{[1, N_p]}$. Moreover, it is implied that $x^*(k_0 + N_p) \in \mathcal{S}^{N_p}\{N_p\} = \mathbb{X}^{N_p}$, which is a scheduled RPI set for Lipschitzian NLPV system, computed according to Theorem 1, with $\mathbb{X}^{N_p} = \mathcal{X} \ominus \mathbb{X}_e^{N_p}$, being $\mathbb{X}_e^{N_p} := \{e \in \mathbb{R}^{n_x} \mid e(k)Pe \leq \frac{\nu w_{max}^2}{1-\gamma}\}$.

The MPC control policy $u(k_0) = u^*(k_0|k_0)$ is applied to the process and steers the system from the initial state x_0 to a successor state $x(k_0 + 1) = \hat{x}^*(k_0 + 1|k_0) = x^*(k_0 + 1|k_0) = x^*$. Next, we need to demonstrate that, at instant $k_1 = k_0 + 1$, for initial condition $x_1 = x(k_1)$ and scheduling sequence P_{k_1} , there exists a feasible solution to Problem 1. We will use the feasibility of the solution at instant k_0 to construct a feasible solution at this following sample k_1 , which is shown for both the real and the nominal predictions of the system (x and \hat{x} , respectively).

For the real system, we take the bounds of the variation of the scheduling parameters into account, as follows:

$$x(k_1 + 1) = A(\rho(k_1))x_1 + B(\rho(k_1))\Phi(x_1)u(k_1), \quad (66)$$

since $u(k_1) = \kappa x_1 = \kappa x^*$ and $\rho(k_1) = \rho(k_0) + \sum_{l=0}^{k_1-k_0} \partial\rho(l)$, we arrive at:

$$\begin{aligned} x(k_1 + 1) &= A(\rho(k_0) + \partial\rho(k_0))x^* + B(\rho(k_0) + \partial\rho(k_0))\Phi(x^*)\kappa x^* \\ &= [A(\rho(k_0) + \partial\rho(k_0)) + B(\rho(k_0) + \partial\rho(k_0))\Phi(x^*)\kappa] x^* \\ &= \underbrace{(A(\rho(k_0) + \partial\rho(k_0))\Phi(x^*)\kappa)}_{A_{cl}(\rho(k_0), x^*)} x^* + \underbrace{(A(\partial\rho(k_0)) + B(\partial\rho(k_0))\Phi(x^*)\kappa)}_{w(k_1)} x^*. \end{aligned} \quad (67)$$

Since $w(k_1)$ is ultimately bounded because of Lemma 2 and due to the bounds on the scheduling parameters' variation rates, $\partial\rho \leq \partial\rho(k) \leq \bar{\partial\rho}$, and the bounds on x (i.e. $x \in \mathcal{X}$), it holds that $x(k_1 + 1)$ is admissible, being $A_{cl}(\rho(k_0), x^*)x^* + w(k_1) \in \mathcal{X}$.

For the nominal prediction model, it follows quite directly:

$$\hat{x}(k_1 + 1|k_1) = A_{cl}(\rho(k_1), x^*)x^*, \quad (68)$$

which also lies within \mathcal{X} , since x^* is admissible. Thus, it is direct to see that:

$$\begin{aligned} e(k_1 + 1) &= x(k_1 + 1) - \hat{x}(k_1 + 1|k_1) \\ &= A_{cl}(\rho(k_0) - \rho(k_1), x^*)e^* + w(k_1) \\ &\leq A_{cl}(\mu_P, x^*)e^* + w_{max}, \end{aligned} \quad (69)$$

which falls within the scheduled RPI set defined as:

$$\Omega^* = \left\{ e \in \mathbb{R}^{n_x} \mid e^T P e \leq \frac{\nu w_{max}^2}{1-\gamma} \right\}. \quad (70)$$

Finally, apart from this induction development, we also verify the three recursive feasibility axioms:

A1) Indeed $\ell(x)$ is \mathcal{K} -class lower bounded, i.e.¹⁰:

$$\begin{aligned} \ell(x) &= (x(k+i)^T Q x(k+1) + u(k)^T R u(k)) \\ &= (A(\rho)x + B(\rho)\Phi(x)u)^T Q (A(\rho)x + B(\rho)\Phi(x)u) + x^T \kappa^T R \kappa x \\ &= x^T (A(\rho)^T Q A(\rho) + 2A^T(\rho)Q B(\rho)\Phi(x)\kappa + \kappa^T \Phi^T(x)B^T(\rho)Q B(\rho)\Phi(x)\kappa + \kappa^T R \kappa) x \\ &= x^T (A_\ell(\rho, x)) x \geq x^T \beta_1 x = \beta_1(\|x\|), \end{aligned} \quad (71)$$

for any real constant scalar $\beta_1 = A_\ell(\rho, \underline{x}) \forall x \in \Omega = \mathcal{S}\{N_p - 1\}$.

¹⁰Notation is simplified, the $(k+i)$ is dropped.

A2) Indeed $V(\cdot)$ is \mathcal{K} -class upper bounded; departing from $x(0) = x$, it follows:

$$\begin{aligned}
V(x(k + N_p)) &= x^T(k + N_p)Px(k + N_p) \\
&= \left(A^{N_p}(P_k)x + B^{N_p}(P_k)\Phi(\bar{X}_{k,N_p})\kappa\bar{X}_{k,N_p} \right)^T P \left(A^{N_p}(P_k)x + B^{N_p}(P_k)\Phi(\bar{X}_{k,N_p})\kappa\bar{X}_{k,N_p} \right) \\
&= x^T \underbrace{\left(A^{N_p}(P_k)PA^{N_p}(P_k) \right)}_{A_V(P_k)} x + 2x^T \underbrace{\left((A^{N_p}(P_k))^T PB^{N_p}(P_k)\Phi(\bar{X}_{k,N_p})\kappa \right)}_{h(P_k, \bar{X}_{k,N_p})} \bar{X}_{k,N_p} \\
&\quad + \underbrace{\bar{X}_{k,N_p}^T \left(\kappa^T \Phi(\bar{X}_{k,N_p})(B^{N_p}(P_k))^T PB^{N_p}(P_k)\Phi(\bar{X}_{k,N_p})\kappa \right)}_{B_V(P_k, \bar{X}_{k,N_p})} \bar{X}_{k,N_p} \leq x^T \beta_2 x = \beta_2(\|x\|),
\end{aligned} \tag{72}$$

for any real constant scalar $\beta_2 = \left(A_V(P_k) + 2h(\bar{P}, \bar{X}_{k,N_p}), \bar{A}_{k,N_p} + \bar{A}_{k,N_p}^T B_V(\bar{P}, \bar{X}_{k,N_p}) \bar{A}_{k,N_p} \right)$, $\forall x \in \mathbb{X}^{N_p}$.

A3) Finally, the last axiom can be verified via an LMI. Consider $x(k + j + 1) = x(k + 1)$ and $x(k + j) = x$, for notation simplicity, which is valid for all $j \in \mathbb{N}_{[0, N_p - 1]}$. We must verify that the terminal ingredient is decrescent along the solution of x . Of course, the use of the contracting terminal constraints $x(k + N_p|k) \in \mathcal{S}^{N_p}\{i\}$, for $i = \max\{N_p - k, 0\}$, implied that the state trajectories are steered further towards the target set goal as k increases, this, by itself, should guarantee that V is decrescent^{61,60}. We demonstrate the decay of $V(\cdot)$:

$$\begin{aligned}
V(x(k + 1)) - V(x) + \ell(x) &\leq 0 \\
x(k + 1)^T Px(k + 1) - x^T Px + x^T (A_\ell(P_k, x))x &\leq 0 \\
x^T \left(A(\rho(k))^T PA(\rho(k)) + 2A(\rho(k))PB(\rho(k))\Phi(x)\kappa + \kappa^T \Phi^T(x)B^T(\rho(k))PB(\rho(k))\Phi(x)\kappa \right) x \\
&\quad - x^T Px + x^T (A_\ell(P_k, x))x \leq 0,
\end{aligned} \tag{73}$$

which is equivalent to:

$$\begin{aligned}
\left(A^T(\rho(k))(P + Q)A(\rho(k)) - P \right) + \left(\kappa^T \Phi^T(x)B^T(\rho(k))(P + Q)A(\rho(k)) + A^T(\rho(k))(P + Q)B(\rho(k))\Phi(x)\kappa \right) \\
+ \left(\kappa^T \Phi^T(x)B^T(\rho(k))(P + Q)B(\rho(k))\Phi(x)\kappa \right) \leq \kappa^T R \kappa,
\end{aligned} \tag{74}$$

multiplying it by both sides by $\text{diag}\{(P + Q)\}$, it yields the following LMI:

$$\begin{bmatrix}
(A^T(\rho(k))(P + Q)A(\rho(k)) - P) & A^T(\rho(k))(P + Q)B(\rho(k)) \\
\star & B^T(\rho(k))(P + Q)B(\rho(k))
\end{bmatrix} \leq \kappa^T R \kappa, \tag{75}$$

which, if multiplied by the left and, then, by the right with $[\Phi(x)\kappa]$ and $[\Phi^T(x)\kappa^T]$, respectively, is equivalent to the prior inequality in Eq. (74). It is direct to verify that if LMI holds, since $\kappa^T R \kappa$ is a scalar and LMI (24) implies the regional decrease of V , Axiom A3 is valid. Therefore, P is computed such that (24) and (75) hold, being Q also full-rank and positive definite by definition. This concludes proof. \square

Since recursive feasibility is verified, we will now verify the ISS property of the proposed algorithm.

Assumption 7. (i) There exists two \mathcal{K} functions $\beta_i(x(0), k)$ and $\sigma_i(w_{max})$ that, added up, upper bound the norm of the system states, for all $k \in \mathbb{N}$; and (ii) there exists a set of states \mathcal{D}_x for which the MPC optimization problem in Eq. (10) is feasible.

Proposition 4. The MPC cost function $\ell(\cdot)$ and the terminal stage cost $V(\cdot)$ also agree with a singular local Lipschitz condition in x for each of them. Let $\ell(x, u, k)$ be such that $\ell(0, 0, k) = 0 \forall k \in \mathbb{N}$. Assuming that $\ell(\cdot)$ is \mathcal{K} -class lower bounded, it is Lipschitz continuous in $\mathcal{X} \times \mathcal{U}$, with a Lipschitz constant Γ_ℓ in the 2-norm for all $x, \hat{x} \in \mathcal{X}$ and $u \in \mathcal{U}$. Moreover, let $V(x)$ be such that $V(0) = 0$. Assuming that $V(x)$ is \mathcal{K} -class lower bounded, it is Lipschitz continuous in \mathcal{X} , with a Lipschitz constant Γ_V in the 2-norm for all $x, \hat{x} \in \mathcal{X}$. This is:

$$\|\ell(x, u, k) - \ell(\hat{x}, u, k)\| \leq \Gamma_\ell \|x - \hat{x}\|, \tag{76}$$

$$\|V(x) - V(\hat{x})\| \leq \Gamma_V \|x - \hat{x}\|. \tag{77}$$

Proof. For simplicity, we take x_s and u_s as null. Then, we can directly find a locally Lipschitz condition in x , since $\ell(\cdot)$ and $V(\cdot)$ are sums of weighted norms on x . This is:

$$\begin{aligned} \|\ell(x, u, k) - \ell(\hat{x}, u, k)\| &= \|(x^T Q x + u^T R u) - (\hat{x}^T Q \hat{x} + u^T R u)\| \\ &= \|(x^T Q x) - (\hat{x}^T Q \hat{x})\| \\ &\leq \left\| \|x - \hat{x}\|_Q^2 \right\| \leq \Gamma_\ell \|x - \hat{x}\|. \end{aligned}$$

Equivalently,

$$\begin{aligned} \|V(x) - V(\hat{x})\| &= \|(x^T P x) - (\hat{x}^T P \hat{x})\| \\ &\leq \left\| \|x - \hat{x}\|_P^2 \right\| \leq \Gamma_V \|x - \hat{x}\|. \end{aligned}$$

This concludes the proof. \square

Proposition 5. Input-to-State Stability

The closed-loop is input-to-state stable for all initial conditions x_0 departing from within some region D_x .

Proof. Firstly, since Propositions 2 and 3, the NLPV system is stabilizable in the feasibility set \mathcal{X} and the control policy is recursively feasible for this same set. Therefore, we assume that the set region $D_x = \mathcal{X}$.

In the sequel, we denote $x^f(k+i|k)$ and $u^f(k+i-1|k)$ as the feasible sequences of states and control inputs, respectively; $\hat{x}(k+i|k-1)$ and $u^*(k+i-1|k-1)$ denote the state sequence and optimal control action sequence computed at $k-1$, respectively. Analogously, we denote the optimal cost at $k-1$ as $J_{N_p}^*(k-1)$ and the cost of the feasible sequence computed at k by $J_{N_p}^f(k)$. The difference between these costs is given by:

$$\begin{aligned} \partial J_{N_p}(k) &= J_{N_p}^f(k) - J_{N_p}^*(k-1) \\ &= \sum_{i=1}^{N_p} [\ell(x^f(k+i|k), u^f(k+i-1|k)) - \ell(\hat{x}(k+i|k-1), u^*(k+i-1|k-1))] \\ &\quad + V(x^f(k+N_p|k)) - V(\hat{x}(k+N_p|k-1)) \\ &= \ell(x^f(k+N_p|k), \kappa x(k+N_p-1|k)) - \ell(x_0, u_0^*) \\ &\quad + \sum_{i=1}^{N_p-1} [\ell(x^f(k+i|k), u^f(k+i-1|k)) - \ell(\hat{x}(k+i+1|k-1), u^*(k+i|k-1))] \\ &\quad + V(x^f(k+N_p|k)) - V(\hat{x}(k+N_p|k-1)), \end{aligned} \tag{78}$$

being x_0 and u_0^* the state of the system and the control input applied at instant $k-1$.

Since $u^f(k+i-1|k)$ is an admissibly feasible control sequence, it holds that $u^f(k+i-1|k) = u^*(k+i|k-1)$ for $i \in \mathbb{N}_{[1, N_p]}$ (i.e. it is optimal). Hence, we can write:

$$\ell(x^f(k+i|k), u^f(k+i-1|k)) - \ell(\hat{x}(k+i|k-1), u^*(k+i-1|k-1)) \leq \Gamma_\ell \Gamma_e w_{max}, \tag{79}$$

and, analogously:

$$V(x^f(k+N_p|k)) - V(\hat{x}(k+N_p|k-1)) \leq \Gamma_V \Gamma_e w_{max}. \tag{80}$$

Appending these expansion into Eq. (78) and considering that both $x^f(k+N_p|k)$ and $\hat{x}(k+N_p|k)$ belong to the RPI set \mathbb{X}^{N_p} , we arrive at the following inequality:

$$\begin{aligned} \partial J_{N_p}(k) &\leq [\ell(x^f(k+N_p|k), \kappa x(k+N_p-1|k)) + V(x^f(k+N_p|k)) - V(\hat{x}(k+N_p|k-1))] \\ &\quad + \overbrace{\left(\Gamma_V \Gamma_e^{N_p-2} + \Gamma_\ell \frac{\Gamma_e^{N_p-2} - 1}{\Gamma_e - 1} \right)}^{\Gamma_J} w_{max} \\ &\quad - \ell(x_0, u_0^*) \\ &\leq \Gamma_J w_{max} - \ell(\hat{x}(k|k-1), u^*(k-1|k-1)). \end{aligned} \tag{81}$$

Considering that $\ell(x, u, k) \geq \beta_1(\|x\|)$ (that it is \mathcal{K} -class lower bounded), as verified under Assumption 6 and Proposition 3, and that the optimality of the solution holds, thence:

$$J_{N_p}^*(k) - J_{N_p}^*(k-1) \leq \partial J_{N_p}^*(k) \leq \Gamma_J w_{max} - \beta_1(\|x\|). \quad (82)$$

Therefore, the optimal cost $J_{N_p}^*(\cdot)$ is indeed an ISS-Lyapunov map of the closed-loop system, with respect to Definition 4, being $\gamma_w(\|w\|) = \Gamma_J w_{max}$. Due to this fact, the closed-loop system is input-to-state stable under the referred predictive control policy. This concludes the proof. \square

Corollary 2. Suboptimality

Under the Assumptions that prelude Proposition 3, the optimality of the solution is a not necessary (but a sufficient) condition to guarantee the convergence of the closed-loop system. Thus, it suffices to compute the solution of the optimization procedure with a lower cost $J_{N_p}^{sub}(\cdot)$ than the one of the feasible solution $J_{N_p}^*(\cdot)$.

5 | IMPLEMENTATION DETAILS

Since the proposed MPC framework has been thoroughly unveiled (and analysed in terms of stabilizability, recursive feasibility and ISS properties), we will now proceed with discussing some implementation details of this algorithm.

For the proposed control policy to work properly, a necessary condition is to have the horizon cost as a \mathcal{K} -class lower bounded function that agrees with Proposition 4, it is a sufficient conditions that Q is positive defined with $\text{rank}\{Q\} = n_x$. Moreover, for Proposition 4 to be valid, P must also be positive defined with $\text{rank}\{P\} = n_x$ (as states in Theorem 1).

Being Q and R fixed matrices with constant values, the implementation of the proposed MPC paradigm follows the lines of the following Algorithm:

Proposed MPC Algorithm

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

Require: Q, R, N_p, p_s, Γ .

Loop:

- Step (1): Measure $\rho(k)$;
- Step (2): Compute the scheduling sequence guess \hat{P}_k ;
- Step (3): Solve LMIs (24) and (75), verifying that inequality (25) holds for the Lipschitz constant Γ . Compute the terminal set \mathbb{X}^{N_p} according to P and Theorem 1;
- Step (4) Compute \mathbb{X}^1 according to Definition 7, with $x(k+1|k) \in \mathbb{X}^1$. Compute the set sequence \mathcal{S}^{N_p} , with a contraction variable λ .
- Step (5): Solve the nonlinear optimization program of Eq. (10), finding κ .
- Step (6): Apply the local control policy $u(k) = \kappa x(k)$;
- Step (7): Increment k , i.e. $k \leftarrow k + 1$.

end

Remark 13. AQUI FALAR DA DIMENSAO DO ALGORITHMO

6 | SOME RESULTS

In this Section, simulation results are presented to demonstrate the performances of the proposed MPC method for NLPV systems with Lipschitz nonlinearities.

6.1 | Automotive Suspension System

We will consider the case of a 1/5-scaled vehicle equipped with 4 semi-active Electro-rheological (ER) dampers¹¹. Each corner of the vehicle represents a singular NLPV system, for which the dampers act as actuators to diminish the effect of the road upon the safety and comfort of the passengers onboard. Figure 1 shows an schematic diagram of the corner of a vehicle with a Semi-Active suspension unit: the chassis body is represent by a sprung mass m_s , which is connected to the wheel link, represented by the unsprung mass m_{us} , by a spring (with stiffness parameter k_s) and a damper (with a variable damping coefficient $c(\cdot)$); the wheel is represented by a spring with stiffness k_t . The vertical dynamics are given by the displacement of the sprung and unsprung masses ($z_s(t)$ and $z_{us}(t)$, respectively), while $z_r(t)$ represents the road profile (disturbance).

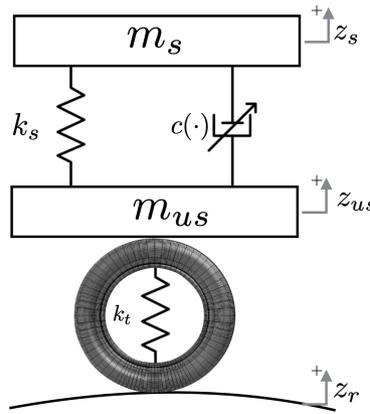


FIGURE 1 Vehicle Corner with Semi-Active Suspension System

We will consider a static nonlinear map⁶² to describe the force given by the ER damper, as follows:

$$\begin{aligned} F_d(t) &= k_0 z_{def}(t) + c(\cdot) \dot{z}_{def}(t) \\ &= k_0 z_{def}(t) + c_0 \dot{z}_{def}(t) + f_c \tanh(k_1 z_{def}(t) + c_1 \dot{z}_{def}(t)) u(t), \end{aligned} \quad (83)$$

where $z_{def}(t) = z_s(t) - z_{us}(t)$ represents the suspension deflection and $u(t)$ is duty cycle of a PWM signal that regulates the voltage input that provides the electrical field upon the damper. This electric field varies the viscosity of the ER fluid. In practice, it is this PWM signal $0 \leq u(t) \leq 1$ that acts as the control input to the application.

The spring and tire forces are given by:

$$F_s(t) = k_s z_{def}(t), \quad (84)$$

$$F_t(t) = k_t (z_{us}(t) - z_r(t)). \quad (85)$$

The dynamics of the sprung mass and unsprung mass displacements are obtained using regular laws of motion around an origin equilibrium:

$$m_s \ddot{z}_s(t) = -F_s(t) - F_d(t), \quad (86)$$

$$m_{us} \ddot{z}_{us}(t) = F_s(t) + F_d(t) - F_t(t). \quad (87)$$

Table 1 shows the values and description for each of the models parameters.

¹¹Refer to <http://www.gipsa-lab.fr/projet/inovel/>.

TABLE 1 Vehicle Model Parameters

Parameter	Description	Value	Unit
m_s	Sprung mass	2.27	kg
m_{us}	Unsprung mass	0.32	kg
k_s	Spring stiffness	1396	N/m
k_t	Tire stiffness	12270	N/m
k_0	Passive damper stiffness	170.4	N/m
k_1	Hysteresis displacement coefficient	218.16	N/m
c_0	Viscous damping coefficient	68.83	N.s/m
c_1	Hysteresis velocity coefficient	21	N.s/m
f_c	Dynamic yield force of the ER fluid	28.07	N

Finally, by selecting the systems states as:

$$x(t) = [z_s(t) \dot{z}_s(t) z_{us}(t) \dot{z}_{us}(t)]^T, \quad (88)$$

and $\omega(t) = z_r(t)$ as the load disturbance, we arrive at the following state-space model:

$$\dot{x}(t) = A_c x(t) + B_c(\rho)\Phi(x(t)) + B_{c_2} \omega(t), \quad (89)$$

being A_c , B_c and B_{c_2} matrices given by:

$$A_c = \begin{bmatrix} 0 & 1 & 0 & 0 \\ \frac{-(k_s+k_0)}{m_s} & \frac{-c_0}{m_s} & \frac{(k_s+k_0)}{m_s} & \frac{c_0}{m_s} \\ 0 & 0 & 0 & 1 \\ \frac{(k_s+k_0)}{m_{us}} & \frac{c_0}{m_{us}} & \frac{-(k_s+k_0+k_t)}{m_{us}} & \frac{-c_0}{m_{us}} \end{bmatrix}, \quad (90)$$

$$B_c(\rho) = \begin{bmatrix} 0 & \frac{-f_c \rho}{m_s} & 0 & \frac{f_c \rho}{m_{us}} \end{bmatrix}^T, \quad (91)$$

$$B_{c_2} = \begin{bmatrix} 0 & 0 & 0 & \frac{k_t}{m_{us}} \end{bmatrix}^T. \quad (92)$$

This model is Euler-discretized with a sampling period of $T_s = 0.5$ s, which yields a Lipschitzian NLPV system of the following form:

$$x(k+1) = Ax(k) + B(\rho(k))\Phi(x(k)) + B_2 \omega(k), \quad (93)$$

where the control signal u is, in fact, also the scheduling parameter of the system, $\rho(k) = u(k)$.

The explicit nonlinearity is:

$$\begin{aligned} \Phi(x(k)) &= \tanh(k_1(x_1(k) - x_3(k)) + c_1(x_2(k) - x_4(k))) \\ &= \tanh(\Gamma_{in} x(k)), \end{aligned} \quad (94)$$

with $\Gamma_{in} = [k_1, c_1, -k_1, -c_1]$. Therefore, $\Phi(x(k))$ undergoes a Lipschitz condition agreement with respect to $x(k)$:

$$\begin{aligned} \|\Phi(x(k)) - \Phi(\hat{x}(k))\| &\leq \|\Gamma_{in}(x(k) - \hat{x}(k))\| \\ &\leq \Gamma \|x(k) - \hat{x}(k)\|, \end{aligned} \quad (95)$$

with $\Gamma = \|\Gamma_{in}\|$ as the Lipschitz constant.

Remark 14. Note that this model in Eq. (93) is slightly different from the one pursued in the control development part of this paper, from Eq. (1): the control input $u(k)$ does not appear explicitly along with the nonlinearity map $\Phi(x(k))$, because it is the scheduling parameter of the plant. Nonetheless, all the prequel development holds, with $u(k) = 1$. Moreover, we have a load disturbance variable $\omega(t)$ that appears in Eq. (93) due to the road profile disturbances. To treat them so that the prediction model $\hat{x}(k+1)$ is similar to the one in Eq. (1), these disturbances which will be compensated out of the prediction, assuming that they

are known along the future N_p steps. This is reasonable for the case of vehicle suspension systems, where the future road can be very well estimated using cameras or adaptive algorithms^{63,64}.

6.2 | Control Goal

The primary control objective is to minimize chassis and wheel accelerations, which provides a smoother and more comfortable ride for the passengers of the vehicle⁶⁵. We will denote a control index given by:

$$J_{Comfort} = \int (a_1 \ddot{z}_s^2(t) + a_2 \ddot{z}_{us}^2(t)) , \quad (96)$$

where parameters a_1 and a_2 are taken as suggests⁶⁶, this is 0.95 and 0.05, respectively. By minimizing $J_{Comfort}$, it is ensured that passengers are isolated from the road bumping motion.

Since we can express these accelerations with respect to displacement and velocity variables (states) and control inputs, we arrive at:

$$J_{Comfort} \approx \int (x(t)^T Q x(t) + u(t)^T R u(t)) , \quad (97)$$

taking a constant value for $\Phi(x(t))$. The horizon cost matrices Q and R are, then, directly found, being full rank and as follows¹²:

$$Q = \text{diag}\{a_1(A_c\{2, : \}) + a_2(A_c\{4, : \})\} , \quad (98)$$

$$R = f_c \left(\frac{a_2}{m_{us}} - \frac{a_1}{m_s} \right) . \quad (99)$$

Moreover, this control objective is complemented with a parallel goal of tracking an origin reference target, i.e. $p_t = (x_s, u_s)$ with $x_s = [0 \ 0 \ 0 \ 0]^T$ and $u_s = 0$.

The control input for the suspension system is the controlled damping coefficient $u(t)$. The total damper force is $F_d(t) = (c + u(t)) \rho$, which is naturally bounded; this leads to the dissipativity constraints $u(t) \in \mathcal{D} = [\underline{u}, \bar{u}]$.

The physical constraints are the damper dissipativity rules on u and the saturation limits on states. Parameters a_1 and a_2 are taken as 0.95 and 0.05 to guarantee that Therefore, weighting matrices are $Q = \text{diag}\{[0 \ a_1 \ 0 \ a_2]\}$ and $R = a_1 \mathbb{1}_m$.

6.3 | Simulation Scenario

To evaluate comfort performances of the vehicle under a controlled semi-active suspension system, we consider the simulation scenario of a vehicle running in a straight line on a dry road, that suddenly (at $t^* = 2.5$ s) undergoes a series of 5 mm bumps along its four wheels. This scenario comprises 15 s with a sequence of three bumps; Figure 2 presents the road disturbance $z_r(k)$ which represents these bumps.

6.4 | Full Nonlinear Model

With respect to the control goal and the considered scenario, the simulation consists of a full-vehicle nonlinear model, which has been validated on a real car⁶⁷. As stated, this full nonlinear model is Euler-discretized considering a sampling period of $T_s = 0.5$ s. The control horizon is synthesized with a prediction horizon of $N_p = 10$ steps, as suggested in previous papers^{66,36}.

6.5 | Numerical Toughness

Firstly, with respect to the proposed control method, we evaluate the numerical toughness it yields. We measure the computational effort in terms of mean average time elapsed to compute the control law, at each sampling instant. The numerical results were obtained with the aid of *Matlab*, *Yalmip* and *SDPT3* and *fmincon* solvers, performed on a 2.4 GHz, 8 GB RAM Macintosh computer.

In Table 2, we present the average computational stress (in seconds) needed to perform the two major tasks for the proposed control: the computation of the RPI sets and the contracted sequence of sets (Steps (3) and (4) of Algorithm 1) and the actual

¹²Notation $A_c\{l, : \}$ denotes the full l^{th} line of matrix A_c .

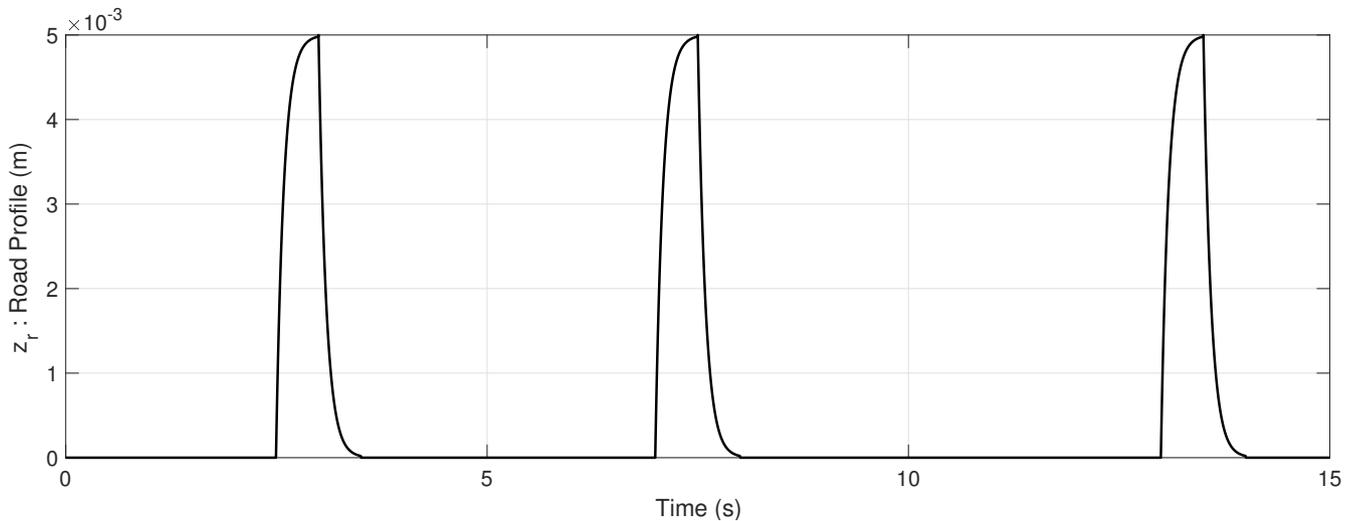


FIGURE 2 Road Profile Scenario

nonlinear optimization procedure to solve the control problem (Step (5) of Algorithm 1). As shown, the total average time is smaller than the sampling period, which means the strategy is feasible under 0.3 s.

Of course, it becomes evident that the online part of the proposed method should be further enhanced (made faster) for system with smaller sampling periods. The method is numerically heavy because the the predictions that arise from the NLPV model are computed online, which is incontrovertibly complex.

TABLE 2 Proposed MPC: Average Computational Stress

Operation	Average Computational Stress
RPI sets - Steps (3) and (4)	0.13 ms
Nonlinear optimization - Step (5)	0.35 ms
Total	0.48 ms

6.6 | Tracking Performances

In terms of performances, we compare the results obtained with the proposed MPC policy with those of a uncontrolled (passive) damper, which corresponds to an open loop policy. In the sequel, the controlled performances are depicted in **full red** lines, while the open loop results are depicted in **dashed blue** lines.

The chosen set-point x_s is the origin. Figure 3 shows the evolution of the four system states x . As shown, the states are brought to the origin as soon as the road bumping dissipates. Furthermore, it must be remarked that all saturation limits (feasibility set \mathcal{X}) are respected.

The chosen set-point x_s is the origin. Figure 3 shows the evolution of the four system states x , while the evolution of scheduling parameter of this system is presented in Figure 4. As shown through these Figures, the states are brought to the origin as soon as the road bumping dissipates. Furthermore, it must be remarked that all saturation limits (feasibility set \mathcal{X}) are respected by the MPC procedure. The LQR violates the lower constraints on x_2 ; evidently, this method provides the worst performances, noisy and abrupt, even deteriorated when compared the open-loop responses. This fact means that the robust LQR solution cannot handle well the variability of the scheduling parameter and the Lipschitz nonlinearity, only ensuring stabilization but no performance enhancements.

More importantly, Figure 5 shows the control policy and the behaviour of the chassis and wheel link accelerations ($\ddot{z}_s(k)$ and $\ddot{z}_{us}(k)$, respectively). These variables are those that the state-feedback policy is set to minimize, with respect to the chosen weighting matrices and performance index $J_{Comfort}$. Evidently, the MPC yields much smaller accelerations, which greatly impacts the comfort of the onboard passengers of the vehicle. Moreover, in terms of the comfort index $J_{Comfort}$, the MPC achieved 30.71 % enhanced performances with respect to the passive damper case, which is already quite significant.

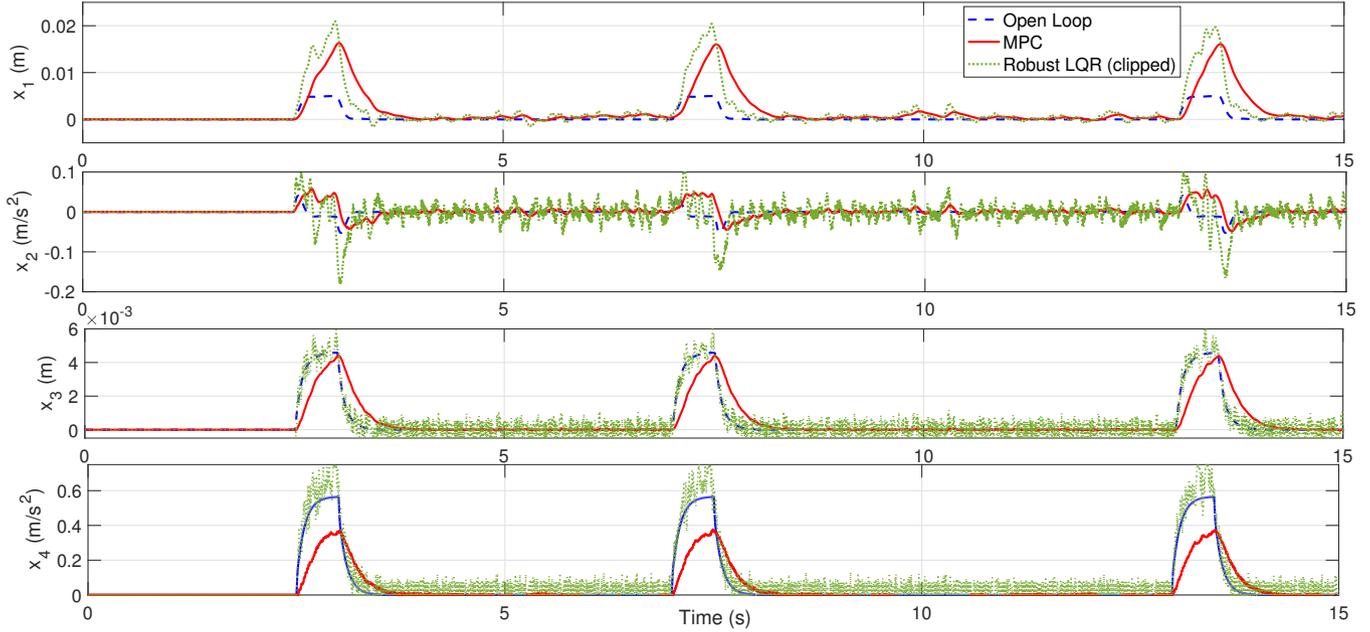


FIGURE 3 Performances: States

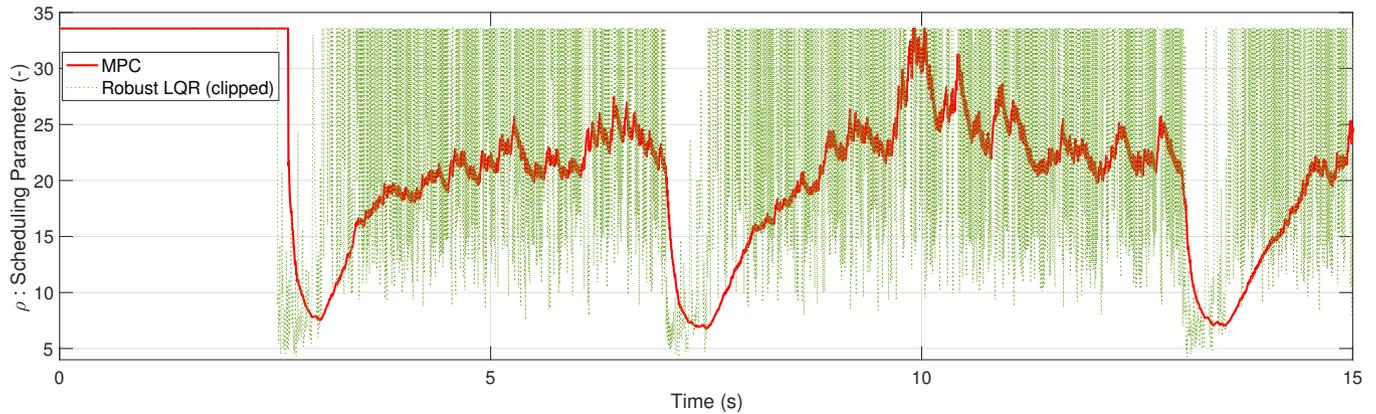


FIGURE 4 Performances: Scheduling Parameter

6.7 | Contracting Sets

To corroborate these performance results, we demonstrate how the sequence of scheduled RPI sets $S_{N_p}\{\cdot\}$ are used to guarantee that the last states $x(k + N_p)$, from the viewpoint of the MPC at instant k , belongs to an RPI set.

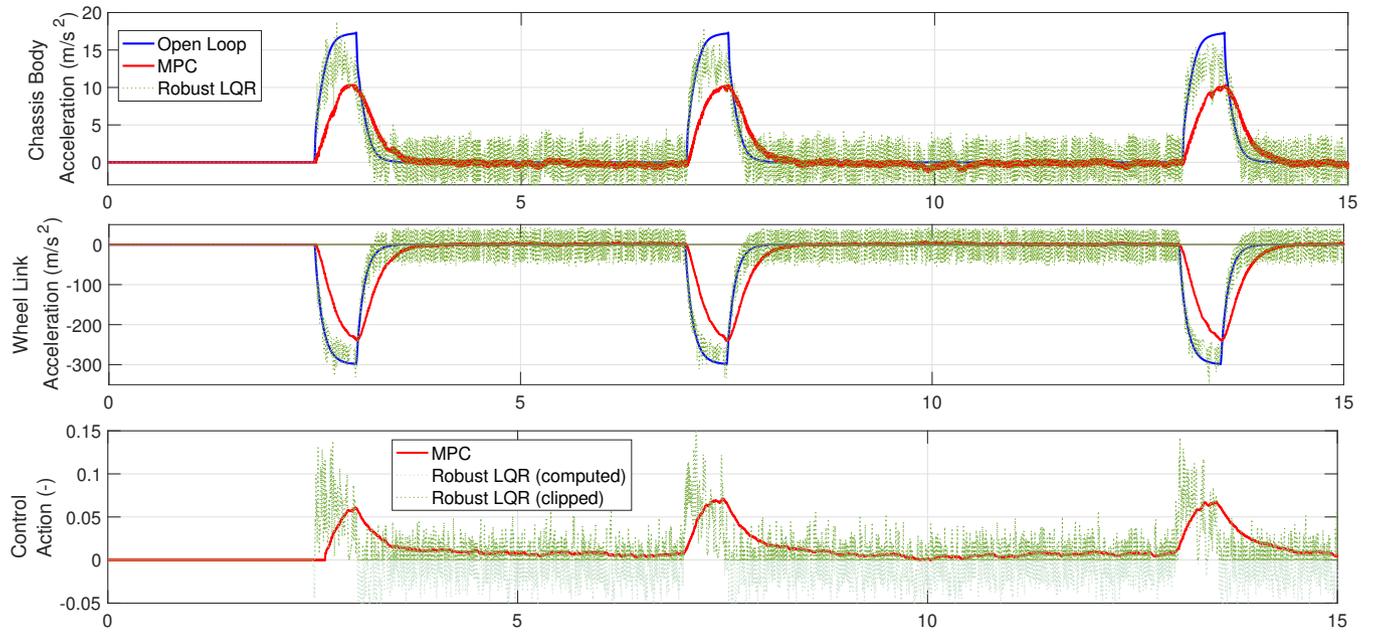


FIGURE 5 Performances: Acceleration and Control Policy

The RPI sets Ω_e given by Theorem 1 are of ellipsoidal form, while the state feasibility sets \mathcal{X} is a polyhedron. Since the actual RPI sets $\Omega = \mathcal{X} \ominus \Omega_e$, and Ω_e is always verified as a subset of \mathcal{X} , we see that the actual state RPI sets $\Omega = \Omega_e$ are also ellipsoidal. Since the state variables are four-dimensional, we can only show \mathbb{R}^2 and \mathbb{R}^3 projections of these ellipsoids.

In Figure 6, we show the RPI sets and their contraction along k , for the first moment when there occurs a bump ($t^* = 2.5$ s) until this bump has dissipated. This Figure shows the evolution for the first three states and how their trajectories are forced, by the MPC policy, to a final contract set which contains the origin (set-point goal). This becomes clearer in Figure 7, which is the same ellipsoids cut in the $x_1 \times x_3$ axis.

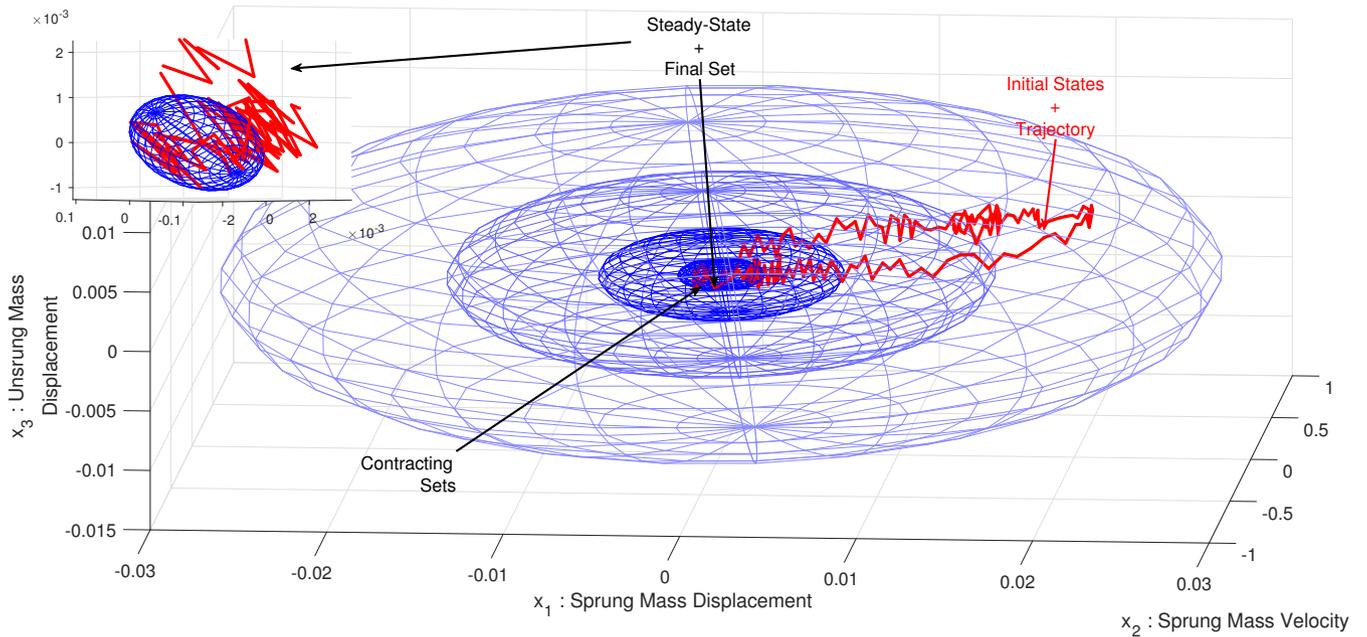


FIGURE 6 RPI Sets, Contraction, in \mathbb{R}^3

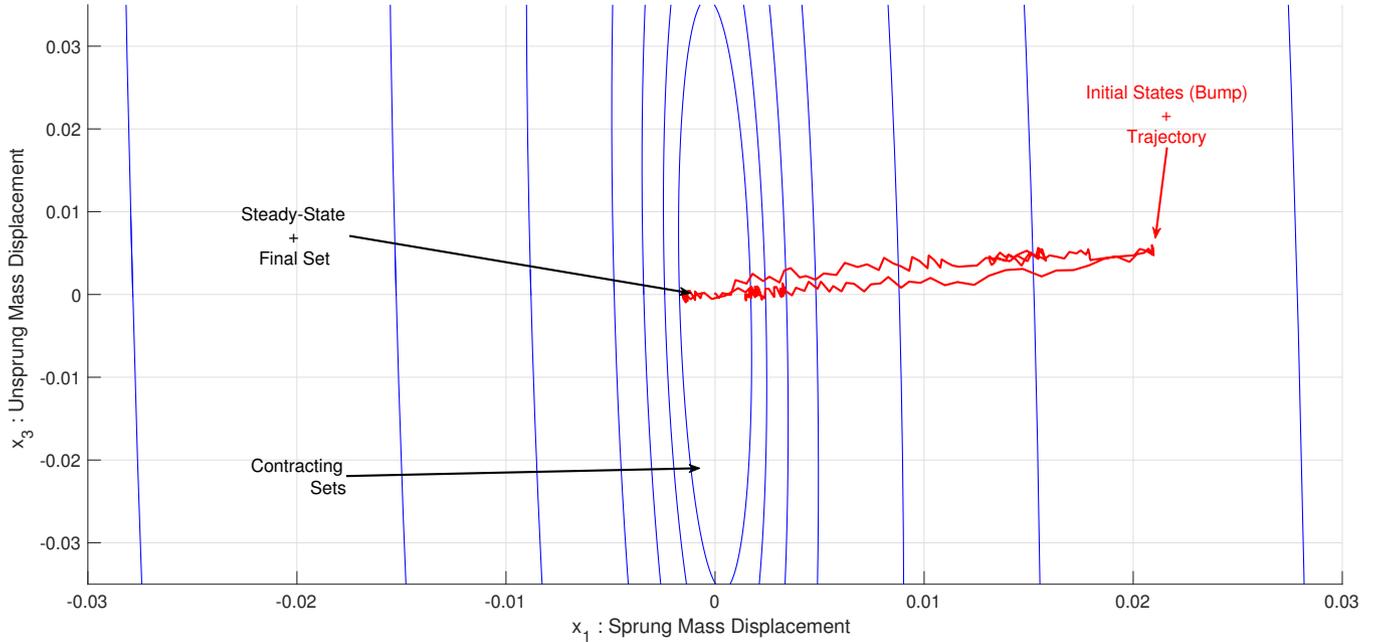


FIGURE 7 RPI Sets, Contraction, in \mathbb{R}^2

With respect to Theorem 1, we must also state that $\Gamma = 309.95$, while the average value found for the right-side of inequality 25 is 415.15.

6.8 | Recursive Feasibility and Input-to-State Stability

Now, we demonstrate experimentally that the Assumptions and Propositions discussed in Section 4 were indeed valid. These propositions were already demonstrated analytically; the following experimental results are simple complementary corollaries.

Firstly, we show that Assumption 6 is valid, showing $\ell(x)$, $V(x(k+N_p))$ and its lower and upper bounds $\beta_1(\|x\|)$ and $\beta_2(\|x\|)$, respectively. This Assumption refers to the validity of axioms A1 and A2 for the recursive feasibility property (Proposition 3), which is corroborated by A3, the decrescent decay of $V(\cdot)$. Figure 8 concatenates the validity of these axioms, showing, on its top subfigure, the lower \mathcal{K} -class bound on $\ell(k)$, on its middle subfigure, the upper \mathcal{K} -class bound on $V(k)$ and, finally, on its bottom subfigure, the decrescent decay of $V(k)$ (after each bump).

Furthermore, with respect to Proposition 5, we show in Figure 9 the experimental values for $\partial J_{N_p}(k)$, and its upper bound given by $\Gamma_J w_{max} - \beta_1(\|x\|)$. This means that the optimal cost map $J_{N_p}^*(\cdot)$ is indeed an ISS-Lyapunov map and the system is input-to-state stable.

7 | CONCLUSIONS

This paper presented a novel state-feedback predictive control method for Nonlinear Parameter Varying systems with explicit Lipschitzian nonlinear terms. As shown from the numerical simulation results, the method is of great effectiveness, guaranteeing input-to-state stabilization and recursive feasibility of the optimization procedure. In terms of implementation of the proposed method, we must stress that:

- the Scheduled Robust Invariant Set Sequence is computed online, at each sampling instant, which may be computationally costly for larger systems;
- at each sampling instant an estimation guess for the following scheduling sequence is necessary - which can be retrieved using simple (and fast) linear algorithms.

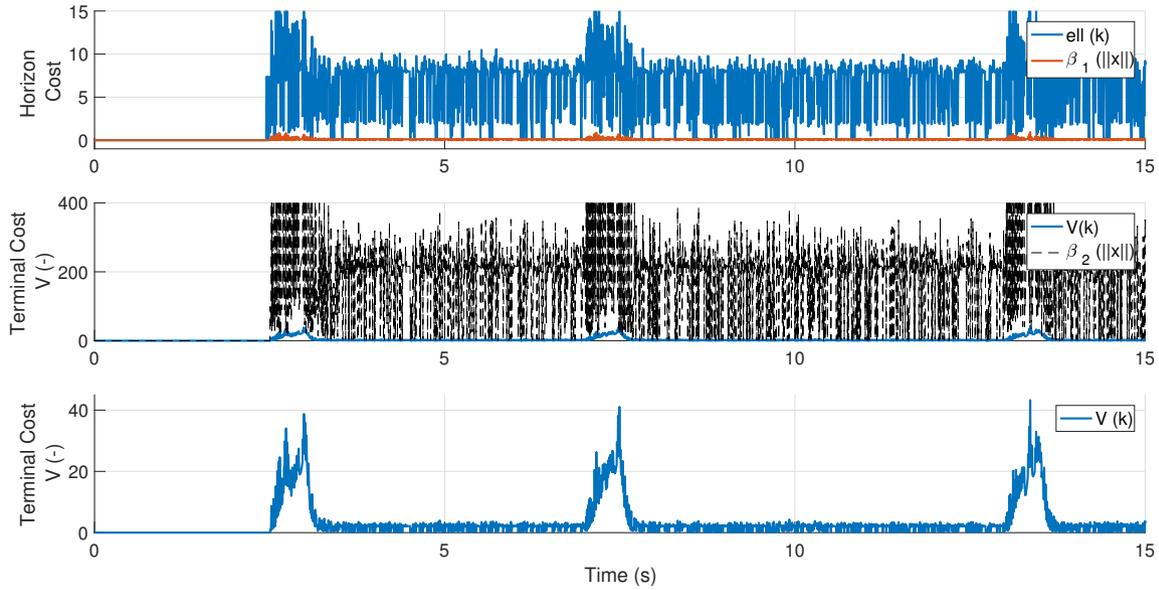


FIGURE 8 MPC Costs

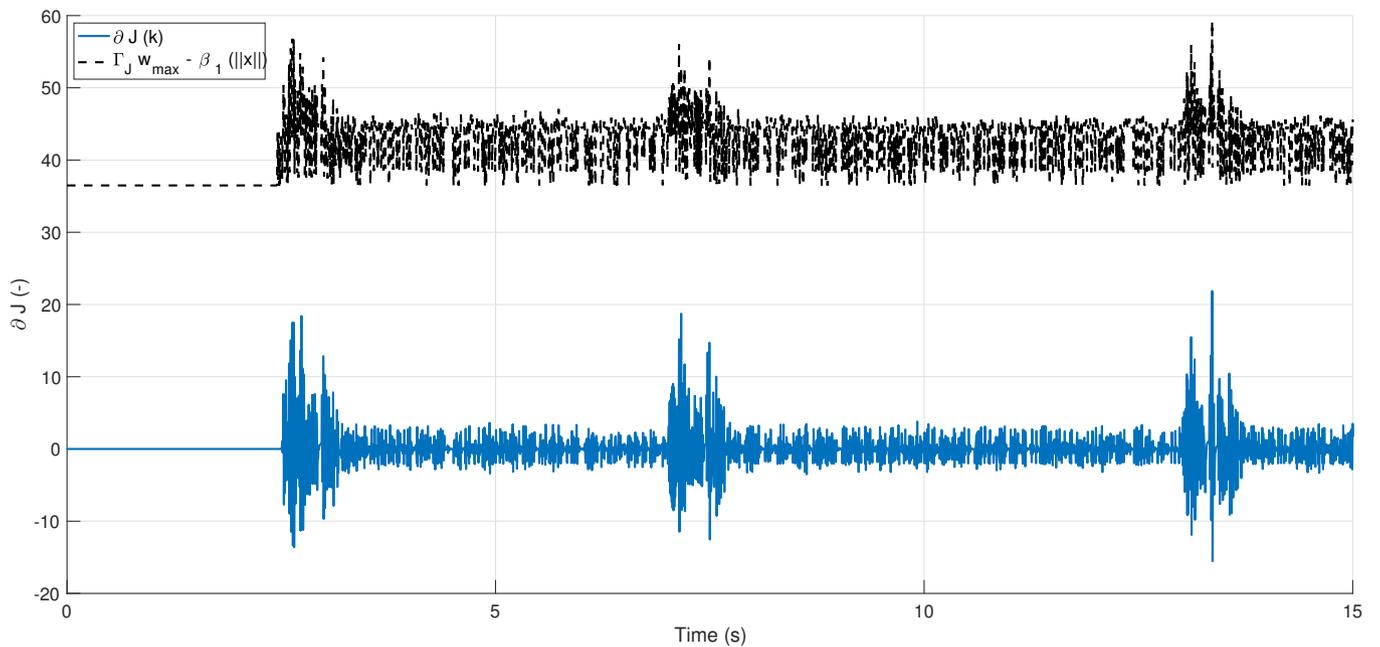


FIGURE 9 ISS-Lyapunov Function

Also note that a method for the computation of Robust Positively Invariant Sets for Lipschitzian NLPV system via LMIs has been proposed in this paper.

For further works, we plan on investigating the robustness of the method towards the elasticity of the μ_P parameter, which bounds the difference between the scheduling sequence estimation and real value.

ACKNOWLEDGMENTS

This work has been supported by *CNPq* project 304032/2019 – 0 and *ITEA3* European project 15016 *EMPHYSIS* (Embedded Systems With Physical Models in the Production Code Software).

Author contributions

All authors have contributed equally for this paper.

Financial disclosure

None reported.

Conflict of interest

The authors declare no potential conflict of interests.

References

1. Camacho EF, Bordons C. *Model predictive control*. Springer Science & Business Media . 2013.
2. Alamir M. *A pragmatic story of model predictive control: self-contained algorithms and case-studies*. CreateSpace Independent Publishing Platform . 2013.
3. Feng L, Wang JL, Poh EK. Improved robust model predictive control with structured uncertainty. *Journal of Process Control* 2007; 17(8): 683–688.
4. Cutler CR, Ramaker BL. Dynamic matrix control: A computer control algorithm. In: No. 17. ; 1980: 72.
5. Clarke DW, Mohtadi C, Tuffs P. Generalized predictive control:Part I. The basic algorithm. *Automatica* 1987; 23(2): 137–148.
6. Mayne DQ. Model predictive control: Recent developments and future promise. *Automatica* 2014; 50(12): 2967–2986.
7. Camacho EF, Bordons C. Nonlinear model predictive control: An introductory review. In: Springer. 2007 (pp. 1–16).
8. Allgöwer F, Zheng A. *Nonlinear model predictive control*. 26. Birkhäuser . 2012.
9. Magni L, Scattolini R. Stabilizing model predictive control of nonlinear continuous time systems. *Annual Reviews in Control* 2004; 28(1): 1–11.
10. Cannon M. Efficient nonlinear model predictive control algorithms. *Annual Reviews in Control* 2004; 28(2): 229–237.
11. Kirches C. *Fast numerical methods for mixed-integer nonlinear model-predictive control*. Springer . 2011.
12. Xu F, Chen H, Gong X, Mei Q. Fast nonlinear model predictive control on FPGA using particle swarm optimization. *IEEE Transactions on Industrial Electronics* 2015; 63(1): 310–321.
13. Graichen K, Käpernick B. A real-time gradient method for nonlinear model predictive control. In: IntechOpen. 2012.
14. Ayala CO, Roca L, Guzman JL, Normey-Rico JE, Berenguel M, Yebra L. Local model predictive controller in a solar desalination plant collector field. *Renewable energy* 2011; 36(11): 3001–3012.
15. Nguyen MQ, Canale M, Sename O, Dugard L. A Model Predictive Control approach for semi-active suspension control problem of a full car. In: IEEE. ; 2016: 721–726.

16. Rathai KMM, Alamir M, Sename O, Tang R. A Parameterized NMPC scheme for embedded control of semi-active suspension system. *IFAC-PapersOnLine* 2018; 51(20): 301–306.
17. Alamir M. A framework for real-time implementation of low-dimensional parameterized NMPC. *Automatica* 2012; 48(1): 198–204.
18. Zhang Y, Li S, Liao L. Near-optimal control of nonlinear dynamical systems: A brief survey. *Annual Reviews in Control* 2019.
19. Sename O, Gaspar P, Bokor J. *Robust control and linear parameter varying approaches: application to vehicle dynamics*. 437. Springer . 2013.
20. Mohammadpour J, Scherer CW. *Control of linear parameter varying systems with applications*. Springer Science & Business Media . 2012.
21. Tóth R. *Modeling and identification of linear parameter-varying systems*. 403. Springer . 2010.
22. Tóth R, Willems JC, Heuberger PS, Hof V. dPM. The behavioral approach to linear parameter-varying systems. *IEEE Transactions on Automatic Control* 2011; 56(11): 2499–2514.
23. Blanchini F, Miani S. Stabilization of LPV systems: state feedback, state estimation, and duality. *SIAM journal on control and optimization* 2003; 42(1): 76–97.
24. Szabó Z, Bokor J. Transformations for linear parameter varying systems. *IFAC-PapersOnLine* 2018; 51(26): 87–93.
25. Abbas HS, Toth R, Petreczky M, Meskin N, Mohammadpour J. Embedding of nonlinear systems in a linear parameter-varying representation. *IFAC Proceedings Volumes* 2014; 47(3): 6907–6913.
26. Jungers M, Oliveira RC, Peres PL. MPC for LPV systems with bounded parameter variations. *International Journal of Control* 2011; 84(1): 24–36.
27. Pistikopoulos E. Perspectives in multiparametric programming and explicit model predictive control. *AIChE journal* 2009; 55(8): 1918–1925.
28. Besselmann T, Lofberg J, Morari M. Explicit MPC for LPV systems: Stability and optimality. *IEEE Transactions on Automatic Control* 2012; 57(9): 2322–2332.
29. Abbas HS, Tóth R, Meskin N, Mohammadpour J, Hanema J. An MPC approach for LPV systems in input-output form. In: *IEEE*. ; 2015: 91–96.
30. Abbas HS, Hanema J, Tóth R, Mohammadpour J, Meskin N. A new approach to robust MPC design for LPV systems in input-output form. *IFAC-PapersOnLine* 2018; 51(26): 112–117.
31. Park JH, Kim TH, Sugie T. Output feedback model predictive control for LPV systems based on quasi-min–max algorithm. *Automatica* 2011; 47(9): 2052–2058.
32. Ding B, Ping X, Pan H. On dynamic output feedback robust MPC for constrained quasi-LPV systems. *International Journal of Control* 2013; 86(12): 2215–2227.
33. Casavola A, Famularo D, Franzè G, Garone E. A fast ellipsoidal MPC scheme for discrete-time polytopic linear parameter varying systems. *Automatica* 2012; 48(10): 2620–2626.
34. Bumroongsri P, Kheawhom S. An ellipsoidal off-line model predictive control strategy for linear parameter varying systems with applications in chemical processes. *Systems & Control Letters* 2012; 61(3): 435–442.
35. Cisneros PS, Sridharan A, Werner H. Constrained Predictive Control of a Robotic Manipulator using quasi-LPV Representations. *IFAC-PapersOnLine* 2018; 51(26): 118–123.
36. Morato MM, Sename O, Dugard L. LPV-MPC Fault Tolerant Control of Automotive Suspension Dampers. *IFAC-PapersOnLine* 2018; 51(26): 31–36.

37. Morato MM, Mendes PRC, Normey-Rico JE, Bordons C. LPV-MPC fault-tolerant energy management strategy for renewable microgrids. *International Journal of Electrical Power & Energy Systems* 2020; 117: 105644. doi: <https://doi.org/10.1016/j.ijepes.2019.105644>
38. Morato MM, Normey-Rico JE, Sename O. Novel qLPV MPC Design with Least-Squares Scheduling Prediction. In: ; 2019.
39. Larimore WE. Identification of nonlinear parameter-varying systems via canonical variate analysis. In: IEEE. ; 2013: 2247–2262.
40. Blesa J, Jiménez P, Rotondo D, Nejjari F, Puig V. An interval NLPV parity equations approach for fault detection and isolation of a wind farm. *IEEE Transactions on Industrial Electronics* 2014; 62(6): 3794–3805.
41. Cai X, Liu Y, Zhang W. Control design for a class of nonlinear parameter varying systems. *International Journal of Systems Science* 2015; 46(9): 1638–1647.
42. Sala A, Ariño C, Robles R. Gain-Scheduled Control via Convex Nonlinear Parameter Varying Models. *IFAC-PapersOnLine* 2019; 52(28): 70–75.
43. Rotondo D, Johansen TA. Analysis and design of quadratic parameter varying (QPV) control systems with polytopic attractive region. *Journal of the Franklin Institute* 2018; 355(8): 3488–3507.
44. Yang R, Rotondo D, Puig V. D-stable Controller Design for Lipschitz NLPV System. *IFAC-PapersOnLine* 2019; 52(28): 88–93.
45. Pham TP, Sename O, Dugard L. Real-time Damper Force Estimation of Vehicle Electrorheological Suspension: A Non-Linear Parameter Varying Approach. In: Proceedings of the 3th IFAC Workshop on Linear Parameter Varying Systems, Eindhoven, The Netherlands, Nov. 4-6, 2019. IFAC. ; 2019.
46. Zemouche A, Rajamani R, Kheloufi H, Bedouhene F. Robust observer-based stabilization of Lipschitz nonlinear uncertain systems via LMIs-discussions and new design procedure. *International journal of robust and nonlinear control* 2017; 27(11): 1915–1939.
47. Veselý V, Körösi L. Robust PID controller design for nonlinear systems. *Journal of Electrical Engineering* 2018; 69(1): 65–71.
48. Rajamani R. Observers for Lipschitz nonlinear systems. *IEEE transactions on Automatic Control* 1998; 43(3): 397–401.
49. Zhu F, Han Z. A note on observers for Lipschitz nonlinear systems. *IEEE Transactions on automatic control* 2002; 47(10): 1751–1754.
50. Yu S, Maier C, Chen H, Allgöwer F. Tube MPC scheme based on robust control invariant set with application to Lipschitz nonlinear systems. *Systems & Control Letters* 2013; 62(2): 194–200.
51. Limón D, Alamo T, Camacho EF. Input-to-state stable MPC for constrained discrete-time nonlinear systems with bounded additive uncertainties. In: . 4. IEEE. ; 2002: 4619–4624.
52. Gritli H, Belghith S. LMI-Based Design of State Feedback Controller for Lipschitzian Nonlinear Systems. In: IEEE. ; 2018: 353–361.
53. Zemouche A, Boutayeb M. On LMI conditions to design observers for Lipschitz nonlinear systems. *Automatica* 2013; 49(2): 585–591.
54. Cisneros PS, Voss S, Werner H. Efficient nonlinear model predictive control via quasi-LPV representation. In: IEEE. ; 2016: 3216–3221.
55. Mayne DQ, Rawlings JB, Rao CV, Scokaert PO. Constrained model predictive control: Stability and optimality. *Automatica* 2000; 36(6): 789–814.
56. Jiang ZP, Wang Y. Input-to-state stability for discrete-time nonlinear systems. *Automatica* 2001; 37(6): 857–869.

57. Blanchini F, Miani S. *Set-theoretic methods in control*. Springer . 2008.
58. Hanema J, Lazar M, Tóth R. Stabilizing tube-based model predictive control: Terminal set and cost construction for LPV systems. *Automatica* 2017; 85: 137–144.
59. Yu S, Böhm C, Chen H, Allgöwer F. Robust model predictive control with disturbance invariant sets. In: IEEE. ; 2010: 6262–6267.
60. Limón D, Alvarado I, Alamo T, Camacho EF. MPC for tracking piecewise constant references for constrained linear systems. *Automatica* 2008; 44(9): 2382–2387.
61. Limon D, Alamo T, Camacho EF. Enlarging the domain of attraction of MPC controllers. *Automatica* 2005; 41(4): 629–635.
62. Guo S, Yang S, Pan C. Dynamic modeling of magnetorheological damper behaviors. *Journal of Intelligent material systems and structures* 2006; 17(1): 3–14.
63. Tudón-Martínez JC, Fergani S, Sename O, Martínez JJ, Morales-Menendez R, Dugard L. Adaptive road profile estimation in semiactive car suspensions. *IEEE Transactions on Control Systems Technology* 2015; 23(6): 2293–2305.
64. Unger A, Schimmack F, Lohmann B, Schwarz R. Application of LQ-based semi-active suspension control in a vehicle. *Control Engineering Practice* 2013; 21(12): 1841–1850.
65. Kiencke U, Nielsen L. *Automotive control systems: for engine, driveline, and vehicle*. Springer Science & Business Media . 2005.
66. Morato MM, Nguyen MQ, Sename O, Dugard L. 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* 2018.
67. Poussot-Vassal C, Sename O, Dugard L, Savaresi S. Vehicle dynamic stability improvements through gain-scheduled steering and braking control. *Vehicle System Dynamics* 2011; 49(10): 1597–1621.

AUTHOR BIOGRAPHIES



Marcelo Menezes Morato is a PhD Candidate in Automation and Systems Engineering at UFSC/Brazil and Automation and Production and UGA/France. Currently, he holds a substitute professor position at the Automation and Systems Department at UFSC. He is ad-hoc referee to multiple journals, including International Journal of Electrical Power & Energy Systems, Control Engineering Practice, and Renewable Energy. He received the 2019 Green Talents award from the German government for young high-potential scientists due to his relevant research in sustainability.



Julio Elias Normey-Rico received his Ph.D. degree from University of Seville, Spain, in 1999. He is currently a Full Professor of the Dept. of Automation and Systems Engineering in Federal University of Santa Catarina (UFSC), in Brazil and head researcher of Renewable Energies Research Team (GPER/UFSC), lead research group in this topic in Latin America. He is the director of several research partnerships with energy industries and international cooperation agreements (Argentina, Uruguay, Spain, Chile and Italy). He is the author of over 260 conference and journal papers, and published four book chapters and a full textbook on Control of Dead-Time Processes (Springer). He has supervised over 60 PhD/MSc. candidates. He was associated Editor of Control Engineering Practice from 2007 to 2018 and Editor of the International Renewable Energy Congress since 2014. Since 2000 he integrates the NOC of several national conferences related with automatic control, and recently, he was the General Chair of the IFAC Symposium DYCOPS 2019.



Olivier Sename received a Ph.D. degree from Ecole Centrale Nantes in 1994. He is now Professor at the Grenoble Institute of Technology within GIPSA-lab. His main research interests include Linear Parameter Varying systems and automotive applications. He is the (co-)author of 1 book, more than 50 international journal papers, and more than 200 international conference papers. He has led several industrial (Delphi, Renault, Volvo Trucks, JTEKT) and international (Mexico, Italy, Hungary) collaboration projects. He has supervised 30 Ph.D. students.

How to cite this article: M. M. Morato, J.E. Normey-Rico, and O. Sename (2021), An Input-to-State Stable Model Predictive Control Framework for Lipschitz Nonlinear Parameter Varying Systems, *International Journal of Robust and Nonlinear Control*.