2016 IEEE 55th Conference on Decision and Control (CDC)
ARIA Resort & Casino
December 12-14, 2016, Las Vegas, USA

Efficient Nonlinear Model Predictive Control via quasi-LPV Representation

Pablo S. G. Cisneros[1], Sophia Voss[1] and Herbert Werner[1]

Abstract— **Nonlinear Model Predictive Control often suffers from excessive computational complexity, which becomes critical when fast plants are to be controlled. This papers presents an approach to NMPC that exploits the quasi-LPV framework. For quasi-LPV systems, the scheduling variables are determined by the state variables and/or inputs. By calculating an estimate of the state variables during prediction, the prediction model can be adapted to the estimated state evolution in each step. Stability of the proposed algorithm is enforced by the offline solution of an optimization problem with Linear Matrix Inequality (LMI) constraints. Furthermore, an iterative approach is presented with which the NMPC optimization problem can be handled by solving a series of Quadratic Programs (QP) or Second Order Cone Programs (SOCP) in each time step, which leads to computational efficiency. The algorithm is tested in simulation to highlight convergence of the prediction and stability of the closed-loop under contraints.**

I. INTRODUCTION

The development of advanced control strategies for a wide range of plant dynamics demands efficient modeling techniques. Over the last two decades research activities in the field of Linear Parameter Varying (LPV) systems have increased significantly. LPV model are used to represent linear plant models that depend on time-varying parameters (the *scheduling variables*), which can be exogenous. The scheduling variables can however also be used to model nonlinear dynamics, in this case they will be functions of state or input variables and the model is referred to as quasi-LPV (qLPV) model. The formulation of nonlinear dynamics in qLPV form is the basis for nonlinear control using efficient controller synthesis strategies originally developed for Linear Time Invariant (LTI) systems, like H_∞ control. In this paper we will focus on Model Predictive Control (MPC) for qLPV systems.

In the 1980s with Generalized Predictive Control the fundamentals of MPC based control strategies were formulated, see [1], [2]. Since then, the MPC framework has been extended to a variety of system classes and used in practical applications, e.g., [3], [4]. The MPC approach is based on online prediction of future system behavior given a system model. Therefore, handling difficult dynamics like time delays is comparatively easy. The most crucial benefit of MPC is the ability of constraining controlled and manipulated variables explicitly.

*This work was not supported by any organization

[1]Pablo S.G. Cisneros, Sophia Voss and Herbert Werner are with Institute of Control Engineering, Hamburg University of Technology, Germany `pablo.gonzalez@tuhh.de` `sophia.voss@tuhh.de` `h.werner@tuhh.de`

In this context, research effort has been directed recently toward MPC for LPV systems with exogenous scheduling variables. Since the evolution of the scheduling variables is unknown during prediction, the prediction model becomes uncertain. Therefore, using MPC control strategies that allow for parametric uncertainty is one way of designing MPC controllers for LPV systems. With Robust Model Predictive Control (RMPC) the MPC approach has been extended to different types of uncertainty, see [5], [6]. Another way of considering uncertainty in the MPC framework is given by Stochastic Model Predictive Control (SMPC) [7]. In SMPC a stochastic description of the uncertainty is assumed. Moreover, much research effort has been spent on explicit controller design for LPV systems [8]. Most of these approaches are based on a worst-case objective function that minimizes maximum values of an objective function for different values of the scheduling variables [9][10], which results in conservatism. There are relaxation techniques to reduce that conservatism, e.g., [11], [12].

In this paper the focus is on quasi-LPV systems, where the scheduling variables at each time step are determined by the state vector at that time. So far, research on MPC for quasi-LPV systems has mainly focused on state estimation errors in case of dynamic output feedback, see [13], [14]. The contribution of this paper is the formulation of an optimization problem that exploits the functional dependency of scheduling variables and state vector to develop a prediction strategy, thereby avoiding worst-case optimization. A second contribution is a numerically attractive solution that is iteratively driven to the exact solution. In comparison to other approaches mentioned above, the proposed method is less conservative and computationally highly efficient because the problem decomposes into solving a series of Second Order Cone Programs (SOCPs) or Quadratic Programs (QP) at each time step.

A fast approximate non-linear MPC using a moving linearization, resulting in a Linear Time-Varying (LTV) model is presented in [15]. The linearization is made about the previous predicted input trajectory, needing an initialization for time t_0 which comes from a flatness based analysis. This method bears similarity with the one presented in this paper; however, the use of the LPV framework allows for exact non-linear system representation and no time-consuming linearization takes place. Moreover, in the presented approach no flatness based trajectory is needed for initialization.

The paper is organized as follows: In Section II the problem of MPC controller design for quasi-LPV systems is described in a state space setting and a stability proof

978-1-5090-1837-6/16/$31.00 ©2016 IEEE

is given in terms of offline solutions to LMI problems. In Section III an iterative method is proposed which greatly reduces computational complexity. Section IV illustrates the proposed algorithm by a simulation example. In Section V conclusions and an outlook are given.

A. Notation

We denote a block diagonal matrix with matrices M repeated N times along the diagonal as $\mathrm{diag}_N(M)$. The *one* vector is denoted as $\mathbf{1}$ with dimension omitted when obvious from context. The notation $M \succ 0$ ($M \succeq 0$) means that M is positive (semi-)definite, similarly, $M \prec 0$ ($M \preceq 0$) means that $-M$ is positive (semi-)definite. The Kronecker product of two matrices A and B is $A \otimes B$.

II. PROBLEM SETUP

We consider the quasi-LPV model

$$x_{k+1} = A(\rho_k)\,x_k + B(\rho_k)\,u_k \tag{1}$$

where $x \in \mathbb{R}^n$, $u \in \mathbb{R}^m$, $\rho_k = f(x_k, u_k) \in \mathbb{R}^{n_\rho}$, and $A : \mathbb{R}^{n_\rho} \to \mathbb{R}^{n\times n}$, $B : \mathbb{R}^{n_\rho} \to \mathbb{R}^{n\times m}$, $f : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^{n_\rho}$ are continuous maps. We assume that $\rho_k \in \mathcal{P}\ \forall k \geq 0$ where $\mathcal{P}$ is a given compact set. This implies that A and B are bounded on $\mathcal{P}$. Throughout this paper we will also assume that $(A(\rho), B(\rho))$ is stabilizable $\forall \rho \in \mathcal{P}$.

The focus is on a predictive control scheme based on minimizing the finite horizon cost

$$J_k = \sum_{l=1}^{N} (x_{k+l}^T Q x_{k+l} + u_{k+l-1}^T R u_{k+l-1}) + \Psi(x_{k+N}) \tag{2}$$

where $Q = Q^T \succeq 0$, $R = R^T \succ 0$ and $\Psi(x) > 0\, \forall\, x \neq 0$, $\Psi(0) = 0$. At each sampling instant k, the values of x_k and u_{k-1} are known, and the optimization problem

$$\min_{U_k} J_k(U_k) \tag{3}$$

subject to $\quad U_k \in \mathcal{U}$,

$\qquad\qquad x_{k+N} \in \mathcal{X}$

is solved online for

$$U_k = \begin{bmatrix} u_k \\ u_{k+1} \\ \vdots \\ u_{k+N-1} \end{bmatrix} \in \mathbb{R}^{Nm}, \tag{4}$$

where $\mathcal{U}$ is a constraint set for the input and $\mathcal{X}$ a constraint set for the terminal state $x(k+N)$. We assume that $f(\mathcal{X} \times \mathcal{U}) \subset \mathcal{P}$. The control law is implemented in a receding horizon fashion, *i.e.* at time k control input u_k is applied, whereas at time $k+1$ the problem min J_{k+1} is solved for U_{k+1} and the newly calculated control input u_{k+1} is applied etc. We also define

$$X_k = \begin{bmatrix} x_{k+1} \\ x_{k+2} \\ \vdots \\ x_{k+N} \end{bmatrix} \in \mathbb{R}^{Nn}, \qquad \mathsf{P}_k = \begin{bmatrix} \rho_k \\ \rho_{k+1} \\ \vdots \\ \rho_{k+N-1} \end{bmatrix} \in \mathbb{R}^{Nn_\rho}, \tag{5}$$

and with a slight abuse of notation use f to denote $\mathsf{P}_k = f(\begin{bmatrix} x_k^\top & X_k^\top \end{bmatrix}^\top, U_k)$. Note that the map f is not directly applied to X_k but to $\begin{bmatrix} x_k^\top & X_k^\top \end{bmatrix}^\top$, as the vector P_k contains the parameters from time k to $k+N-1$ whereas the prediction is done for time $k+1$ to $k+N$.

Note that for a general LPV system it is not possible to solve the optimization problem (3), because the future state sequence cannot be predicted: X_k depends not only on the future control inputs U_k (the decision variables), but also on the future scheduling parameters P_k, which for a general LPV system are not assumed to be known *a priori* but only to be measurable online. In contrast, for a *quasi-LPV system*, where the scheduling parameters ρ_k are determined by x_k and u_k, the state trajectory can be predicted.

Lemma 1: For the quasi-LPV system (1), the predicted state vector $x(k+l)$ in (2) can be calculated for each $l > 0$ from $x(k)$ and $u(k+i)$, $i = 0, 1, \ldots, l-1$.

Proof: Let $\mathcal{A} : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^{n\times n}$ and $\mathcal{B} : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^{n\times m}$ denote the concatenation of maps $A \circ f$ and $B \circ f$, respectively, such that $A(\rho_k) = \mathcal{A}(x_k, u_k)$ and $B(\rho_k) = \mathcal{B}(x_k, u_k)$. We show that we can define a map

$$\Gamma^l : \mathbb{R}^n \times \mathbb{R}^m \times \ldots \times \mathbb{R}^m \;\to\; \mathbb{R}^n$$

such that given $x(k)$ and $u(k+i)$, $i = 0, 1, \ldots, l-1$, the evolution of states governed by (1) can be expressed as

$$x_{k+l} = \Gamma^l(x_k, u_k, \ldots, u_{k+l-1}).$$

Using the shorthand notation $\Gamma_k^l = \Gamma^l(x_k, u_k, \ldots, u_{k+l-1})$, this map is defined recursively by

$$\Gamma_k^0 = x_k,$$

$$\Gamma_k^1 = \mathcal{A}(\Gamma_k^0, u_k)\Gamma_k^0 + \mathcal{B}(\Gamma_k^0, u_k)u_k,$$

and

$$\Gamma_k^{l+1} = \mathcal{A}(\Gamma_k^l, u_{k+l})\Gamma_k^l + \mathcal{B}(\Gamma_k^l, u_{k+l})u_{k+l}.$$

It is straightforward to check that

$$x_{k+l} \;=\; \Gamma_k^l \;=\; \Phi_{k.0}^{l-1} x_k \;+\; \sum_{j=1}^{l} \Phi_{k.j}^{l-1} \mathcal{B}(\Gamma_k^{j-1}, u_{k+j-1})\, u_{k+j-1}, \tag{6}$$

where

$$\Phi_{k.q}^p = \begin{cases} \prod_{i=q}^{p} \mathcal{A}(\Gamma_k^i, u_{k+i}), & p \geq q \\ I_n, & p < q \end{cases}$$

and the right hand side of (6) depends only on $x(k)$ and control inputs up to $k+l-1$. ∎

To illustrate this, we examine a very simple example.

Assume that a single input, first order system with a single parameter $\rho_k = x_k$ is given by

$$x_{k+1} = \underbrace{(a_0 + a_1\rho_k)}_{a(\rho_k)} x_k + bu_k,$$

substituting $\rho_k = x_k$ and $\rho_{k+1} = x_{k+1}$, an expression for $a(\rho_{k+1}) = a(x_{k+1})$ can be very easily obtained as

$$a(x_{k+1}) = a_0 + a_1 x_{k+1} = a_0 + a_1 a_0 x_k + a_1^2 x_k^2 + a_1 bu_k.$$

The value of x_{k+2} is then

$$x_{k+2} = (a_0 + a_1 a_0 x_k + a_1^2 x_k^2 + a_1 bu_k)(a_0 x_k + a_1 x_k^2 + bu_k) + bu_{k+1}$$

It is easy to see that a similar expression can be obtained for all x_{k+i}, $i = 1, 2, ..., N$. Note that the dependence of the predicted states on the control inputs is in this case polynomial, and in the general case nonlinear.

A. Stability Analysis

Following [17] and [16], with the corresponding extensions to quasi-LPV systems, we have the following stability result.

Theorem 1 (Stabilizing MPC for quasi-LPV systems): Assume a nominal controller $F(x) = Fx$, a terminal state domain $\mathcal{X}$ and a terminal state weight $\Psi(x)$ exist such that

1) $0 \in \mathcal{X}$
2) $(A(\rho(x, Fx)) + B(\rho(x, Fx))F)x \in \mathcal{X}$, $\forall x \in \mathcal{X}$
3) $\Psi((A(\rho(x, Fx)) + B(\rho(x, Fx))F)x) - \Psi(x) \leq -x^T Qx - x^T F^T RFx$, $\forall x \in \mathcal{X}$
4) $Fx \in \mathcal{U}$, $\forall x \in \mathcal{X}$

hold. Then, assuming feasibility of the initial state, an MPC controller solving the optimization problem (2) guarantees asymptotic stability.

Proof: Assume that the optimal control sequence at time k is $U_k^* = [u_k^{*T}, u_{k+1}^{*T}, ..., u_{k+N-1}^{*T}]^T$. A feasible (possibly suboptimal) solution at time $k+1$ is given by $U_{k+1} = [u_{k+1}^{*T}, u_{k+2}^{*T}, ..., u_{k+N-1}^{*T}, (Fx_{k+N})^T]^T$. The value of the cost function at time step $k+1$ is then

$$\begin{aligned} J_{k+1} &= J_k - x_{k+1}^T Qx_{k+1} - u_k^{*T} Ru_k^* \\ &- \Psi(x_{k+N}) + x_{k+N+1}^T (Q + F^T RF) x_{k+N+1} + \Psi(x_{k+N+1}), \end{aligned}$$

where we subtracted the first term and final cost of J_k and added the last term and final cost of J_{k+1} for the equality to hold. According to assumption 3 of Theorem 1, the second line is negative; from this, and from the fact that $J_{k+1}^* \leq J_{k+1}$ it follows that

$$J_{k+1}^* \leq J_{k+1} \leq J_k - x_{k+1}^T Qx_{k+1} - u_k^{*T} Ru_k^*$$

which proves that J^* is a Lyapunov function and x converges to the origin. ∎

The assumptions in Theorem 1 are enforced by the offline solution of a set of LMI conditions. These LMIs can be solved on a grid over the admissible parameter set. Note that we assume the unknown matrix variables to be parameter independent. The control constraint set is mapped to $\mathbb{R}^n$ via the nominal control law $u = Fx$; from assumption 4 of Theorem 1 it follows that this is a polyhedron. We search for the largest ellipsoid $\mathcal{X} = \{x | x^T W x \leq 1\}$ contained in this polyhedron and use that as the terminal set.

In order to construct a suitable F, $\mathcal{X}$ and $\Psi(x)$, we consider the following problems:

Feasibility Problem
Define $P = Y^{-1}$ and $F = XY^{-1}$ such that

$$\begin{bmatrix} Y & (A(\rho)Y + B(\rho)X)^T & Y & X^T \\ A(\rho)Y + B(\rho)X & Y & 0 & 0 \\ Y & 0 & Q^{-1} & 0 \\ X & 0 & 0 & R^{-1} \end{bmatrix} \succeq 0, \quad \forall \rho \in \mathcal{P}. \tag{7}$$

Terminal Region
Solve for $W = Z^{-1}$ such that

$$\begin{aligned} &\max_{Z} \quad \det(Z) \\ \text{s. t.} \quad &\begin{bmatrix} -Z & Z(A(\rho) + B(\rho)F)^T \\ (A(\rho) + B(\rho)F)Z & -Z \end{bmatrix} \prec 0, \\ &F_i Z F_i^T \leq \bar{u}_i^2, \\ &\forall \rho \in \mathcal{P}. \end{aligned} \tag{8}$$

In the above LMIs, F_i denotes the ith row of F, accordingly $\bar{u}_i$ denotes de ith row of the vector $\bar{u}$ of admissible values of u. In contrast to [16], where (7) depends on the matrix variables Y and F, a linearizing change of variables with $X = FY$ is proposed here to ensure linear dependency on the unknown matrix variables X and Y.

We have the following result.

Theorem 2: [16] Assume P and F satisfy (7) and W satisfies (8). Then $\Psi(x) = x^T Px$, $\mathcal{X} = \{x | x^T W x \leq 1\}$ and F satisfy the assumptions of Theorem 1.

Remark 1: The LMIs of Theorem 2 are solved on a grid $\forall \rho \in \mathcal{P}$. This choice is conservative since the conditions need only to hold within the terminal region $\mathcal{X}$. However, the terminal region is calculated only after the solutions of said LMIs. Further conservatism comes from considering constant unknown matrices Y, Z, F. The conditions can be relaxed by including parameter dependency with relevant basis functions.

III. Iterative predictions using QP/SOCP

Even though (6) allows the online solution of the optimization problem (3), this is computationally unattractive because of the general nonlinear dependence of the predicted states on the control inputs. In this section we show how the solution to (3) can be approximated by a sequence of solutions to a modified optimization problem where the predicted states are linear in the control inputs.

Using the definitions (4) and (5), X_k can be written in vector form as

$$X_k = H(\mathsf{P}_k)x_k + S(\mathsf{P}_k)U_k, \qquad (9)$$

where

$$H(\mathsf{P}_k) = \begin{bmatrix} A(\rho_k) \\ A(\rho_{k+1})A(\rho_k) \\ \vdots \\ A(\rho_{k+N-1})A(\rho_{k+N-1})\dots A(\rho_k) \end{bmatrix}$$

and

$$S(\mathsf{P}_k) = \begin{bmatrix} B(\rho_k) & 0 & \dots \\ A(\rho_{k+1})B(\rho_k) & B(\rho_{k+1}) & \dots \\ \vdots & \vdots & \dots \\ A(\rho_{k+N-1})\dots A(\rho_{k+1})B(\rho_k) & A(\rho_{k+N-1})\dots A(\rho_{k+2})B(\rho_{k+1}) & \dots \end{bmatrix}$$

Using (9) and augmented block diagonal weighting matrices $\tilde{Q} = \mathrm{diag}_N(Q)$ and $\tilde{R} = \mathrm{diag}_N(R)$, the cost function (2) can be rewritten in vector form as

$$J_k = \big(H(\mathsf{P}_k)x_k + S(\mathsf{P}_k)U_k\big)^T \tilde{Q}\big(H(\mathsf{P}_k)x_k + S(\mathsf{P}_k)U_k\big) + U_k^T \tilde{R} U_k + \Psi(x_{k+N}). \qquad (10)$$

Note that for fixed scheduling trajectories P_k, the predicted states X_k in (9) are linear in the control inputs U_k, and one can - just as in the case of LTI systems - find a solution to problem (3) by solving an SOCP problem, which is considerably more efficient than solving a nonlinear optimization problem. If no rigorous stability guarantee is required, the terminal state constraint can be relinquished, thus making the online optimization problem a QP, which can be solved even faster. This motivates the following iterative approach:

At sampling time k

- Initially problem (3) is solved with the quasi-LPV model (1) replaced by the LTI model that is obtained when the state-dependent scheduling sequence P_k is replaced by $\mathsf{P}_k^0 = 1 \otimes f(x_k, u_{k-1})$.
- A scheduling sequence P_k^l is then iteratively driven towards its optimal value $\mathsf{P}_k^* = f(X_k^*, U_k^*)$, where X_k^* and U_k^* denote the state and input sequences corresponding to the optimal solution to (3).
- This is achieved by solving in iteration step l the optimization problem (3) with P_k replaced by P_k^l, and by generating a new scheduling sequence from the resulting optimal state sequence X_k^l as $\mathsf{P}_k^{l+1} = f(X_k^l, U_k^l)$.
- After the the last iteration, X_k^l and U_k^l are used in the next time step to get P_{k+1}^0, i.e. $\mathsf{P}_{k+1}^0 = f(X_k^l, U_k^l)$ (without time-shifting X_k^l). This is a "warm-start" as suggested in [15] and helps to achieve faster convergence.

Thus the idea is to solve a sequence of optimization problems where the quasi-LPV model (1) is replaced by an LTV model that is generated from (1) by imposing a fixed scheduling sequence, which is then updated in each iteration step using the optimized state sequence. The initial scheduling sequence P_k^0 yields an LTI system; we use the notation Σ^0 to refer to this LTI system. The optimization problem yields U^0 as an estimate of the control input, where the superscript is used to indicate that the sequence corresponds to the system Σ^0. The calculation of the state sequence X^0 is done using (9). This is then used to calculate a parameter trajectory for a subsequent iteration, i.e., $\mathsf{P}_k^1 = f(X_k^0, U_k^0)$, we call this system Σ^1. A new input trajectory U_k^1 can then be found by solving (3) again. Note that input and state trajectories are calculated iteratively.

When $X_k^l \approx X_k^{l-1}$, the input sequence U_k^l is taken as an approximation of the optimal solution U_k^* to (3). The first element of the sequence is then applied to the plant. The procedure is repeated at each sampling instant.

The proposed approach is summarized in Algorithm 1. The stop criterion for the iterations can be chosen as, e.g. the norm of the change of predicted trajectory $||X_k^l - X_k^{l-1}||$ being less than a predefined tolerance ε.

Algorithm 1 MPC for Quasi-LPV Systems

Initialization: reference trajectory, plant model, constants $\tilde{Q}$, $\tilde{R}$, N

$k \leftarrow 0$
Define $\mathsf{P}^0 = \mathbf{1} \otimes f(x_k, u_{k-1})$
repeat
    $l \leftarrow 0$
    repeat
        Solve (3) using P_k^l (system Σ^l) for U_k^l
        Use (9) to calculate X_k^l using P_k^l and U_k^l
        Define $\mathsf{P}_k^{l+1} = f(X_k^l, U_k^l)$
        $l \leftarrow l+1$
    until stop criterion
    Apply u_k to the system
    Define $\mathsf{P}_{k+1}^0 = f(X_k^l, U_k^l)$
    $k \leftarrow k+1$
until end

Remark 2: Experience shows that convergence is reached typically after 2 or 3 iterations (see Figure 3 below). To prove that Algorithm 1 converges to the optimal solution to (3), let $\bar{\Sigma} : \mathbb{R}^{n_\rho} \to \mathbb{R}^n$ denote the map that (given x_k) takes a scheduling sequence P_k^l into the optimal state sequence X_k^l, i.e., $X^l = \bar{\Sigma}(\mathsf{P}^l)$ is the response to the optimal input sequence for the LTV system Σ^l. Then convergence can be shown by showing that the map $\bar{\Sigma} \circ f$ is a contraction.

IV. SIMULATION RESULT

In this section we illustrate the scheme on Algorithm 1 with a simulation example. The system considered is the forced Van der Pol oscillator. This system is known to have a limit cycle and with actuator constraints it is difficult to stabilize the origin. The continous-time model of the system is given by the nonlinear expression

$$\ddot{x} = \mu(1 - x^2)\dot{x} - x + u,$$

with μ a possitive constant. By setting the parameter $\rho = x$, the model can be written as a qLPV state space model

$$\dot{\mathsf{x}} = \underbrace{\begin{bmatrix} 0 & 1 \\ -1 & \mu(1-\rho^2) \end{bmatrix}}_{A(\rho)} \mathsf{x} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u$$

In order to apply the MPC algorithm, this model must be discretized. Since the stabilty conditions are stated for a discrete time systems, the error introduced by discretization should be small. A Runge-Kutta 4 discretization is given by

$$\mathsf{x}_{k+1} = \underbrace{\left(\frac{1}{24}A_\rho^4 T_s^4 + \frac{1}{6}A_\rho^3 T_s^3 + \frac{1}{2}A_\rho^2 T_s^2 + A_\rho Ts + I\right)}_{A_d(\rho)} \mathsf{x}_k + \underbrace{\left(\frac{1}{24}A_\rho^3 B T_s^4 + \frac{1}{6}A_\rho^2 B T_s^3 + \frac{1}{2}A_\rho B T_s^2 + BT_s\right)}_{B_d(\rho)} u$$

where we denote A^2 as the product $A \cdot A$, and T_s is the sampling time, in this case set to $T_s = 0.1$. Note that the discretization makes B_d parameter dependent even though it was originally constant. This would not be the case if a simpler discretization is used (e.g. Euler). For this example we set the constant $\mu = 2$ and given knowledge of the system, the parameter range is $\rho \in [-2, 2]$. The weigthing matrices in (2) are chosen as $Q = \mathrm{diag}(1, 0.5)$ and $R = 0.1$ Given these parameters, solving the optimization problems (7) and (8) results in

$$P = \begin{bmatrix} 25.127 & 6.996 \\ 6.996 & 9.025 \end{bmatrix} \quad W = \begin{bmatrix} 29.095 & 17.144 \\ 17.144 & 23.712 \end{bmatrix}$$

$$F = \begin{bmatrix} -3.915 & -5.832 \end{bmatrix}$$

Figure 1 shows a comparison of the closed-loop response using a controller based on Algorithm 1 and a fixed state feedback controller F resulting from the solution of the first LMI in Theorem 2. It is clear that the predictive controller is able to cope with the saturation and still stabilize the origin, whereas the non-predictive controller fails to do so.

To assess the convergence properties of the iterative algorithm, in this example 9 iterations were carried out. Convergence is evaluated by comparing the iterative predictions X_k^l to the prediction of the exact quasi-LPV system X_k given input sequence U_k^9 (i.e., our "best guess" for U^*). If both trajectories are identical then that means that the LTV system corresponding to the LPV system with predefined trajectory P_k^9 has converged to the exact LPV system. The predictions at time step $k = 0$ for the exact LPV system, the LTI with frozen trajectory and the first 2 iterations can be seen in Figure 2 (the predictions for later iterations are not plotted for clarity, as they mostly coincide with X_k). In Figure 3, the norm of the mismatch between the exact quasi-LPV prediction, X_k, and the prediction at each iteration, X_k^l is shown to be convergent to 0. Moreover, it is easy to see that the mismatch at the second iteration is already small enough for practical purposes, this is even more pronounced after timestep $k = 1$ when the warm start makes the prediction convergence faster.

In Figure 4, a phase portrait of the predicted trajectory X_k^2 at time $k = 0$ is shown together with the ellipsoid obtained from the solution of (8). It is evident that the terminal state reaches the ellipsoid, thereby guarateeing stability. The

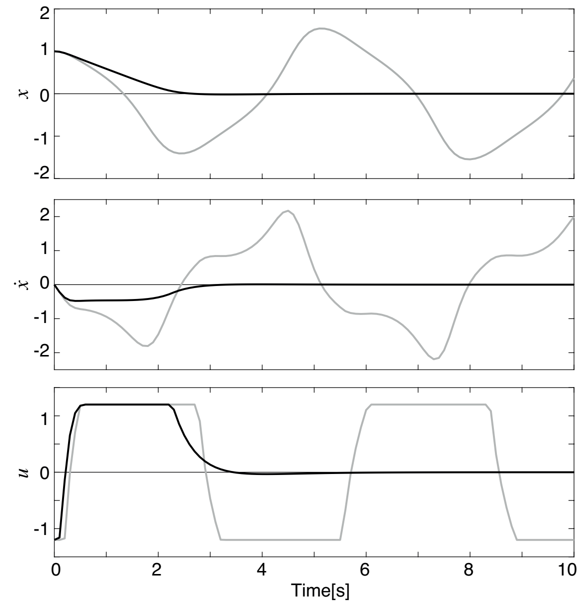

Fig. 1. Comparison of Algorithm 1 (———) and fixed state feedback (———).

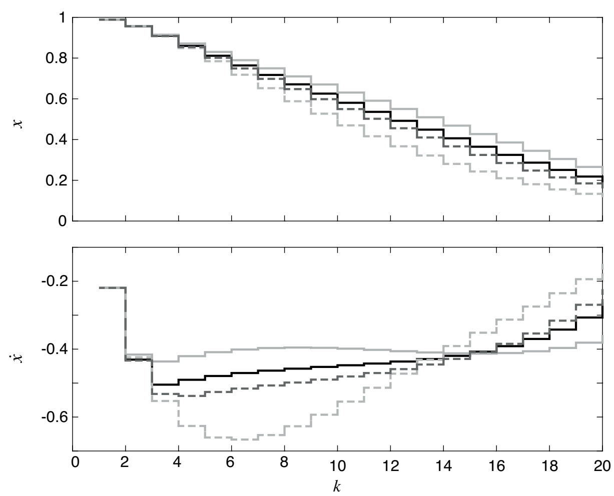

Fig. 2. Convergence of predictions at $k = 0$. X_k(———); X_k^0 (- - - - -); X_k^1 (———); X_k^2 (- - - - -). Later iterations are not shown for clarity.

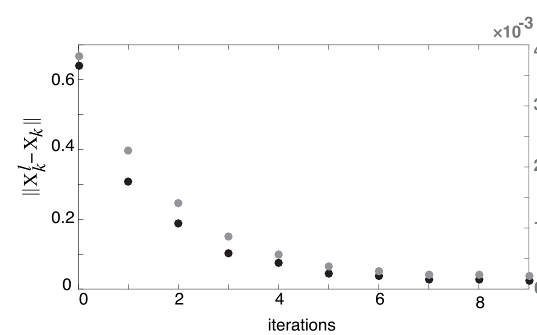

Fig. 3. Convergence of predictions at $k = 0$ (———) and $k = 1$ (———). Note that that y-axes are different for each case.

prediction horizon $N = 20$ was determined such that this condition is met. Note that the horizon is relatively long, this is due to the conservatism (see Remark 1) of the current stability conditions.

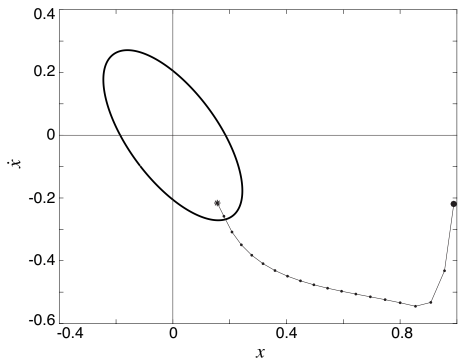

Fig. 4. Terminal constraint set $\mathcal{X}$ and predicted trajectory at $k = 0$

V. CONCLUSIONS

In this paper a novel MPC strategy for quasi-LPV systems is presented. The solution to the constrained optimization problem is found iteratively solving a series of SOCPs whose solution converges to the optimum of the exact problem after typically 2 or 3 iterations. By truncating, the solution is found approximately with a very small error, which, assuming a degree of robustness of the underlying algorithm will in general not render the plant unstable. This method is attractive because of its ease of implementation (basically the same algorithm as for LTI systems) and relatively fast computations, about 3 times slower than for an LTI plant. If no strict stability guarantees are required, the predictions can be done without the terminal constraint, turning the online optimization problem into a QP which is even faster, enabling the algorithm to be applied for fast plants. The LMIs in Theorem 2 are solved for fixed (parameter independent) matrices X, Y, Z and feedback gain F. A relaxation of these conditions can be achieved by considering parameter dependent matrices. This will result in a larger terminal set $\mathcal{X}$, thereby improving feasibility. This extension entails the necessity to define the parameter rate of change $\Delta\rho$ and change said LMIs accordingly. This issue is subject of current research.

REFERENCES

[1] D. W. Clarke, C. Mohtadi, and P. S. Tuffs, "Generalized predictive control-part i. the basic algorithm," *Automatica*, vol. 23, no. 2, pp. 137–148, 1987.

[2] D. W. Clarke, C. Mohtadi, and P. S. Tuffs, "Generalized predictive control-part ii. extensions and interpretations," *Automatica*, vol. 23, no. 2, pp. 149–160, 1987.

[3] J. A. Rossiter, *Model-Based Predictive Control: A Practical Approach*. CRC Press, 2003.

[4] J. M. Maciejowski, *Predictive Control with Constraints*. Harlow Prentice Hall, 2002.

[5] A. Bemporad and M. Morari, "Robust model predictive control: A survey," *Lecture Notes in Control and Information Sciences*, vol. 245, pp. 207–226, 1999.

[6] P. Falugi and D. Q. Mayne, "Getting robustness against unstructured uncertainty: A tube-based mpc approach," *IEEE Transactions on Automa*, vol. 59, pp. 1290–1295, May 2014.

[7] G. Schildbach, L. Fagiano, C. Frei, and M. Morari, "The scenario approach for stochastic model predictive control with bounds on closed-loop constraints violations," *Automatica*, vol. 50, pp. 3009–3018, December 2014.

[8] S. Yu, C. Böhm, H. Chen, and F. Allgöwer, "Stabilizing model predictive control for lpv systems subject to constraints with parameter-dependent control law," *American Control Conference*, pp. 3118–3123, June 2009.

[9] T. H. Kim, J. H. Park, and T. Sugie, "Output-feedback model predictive control for lpv systems with input saturation based on quasi-min-max algorithm," *IEEE Conference on Decision and Control*, vol. 45, pp. 1454–1459, December 2006.

[10] H. S. Abbas, R. Toth, N. Meskin, J. Mahammadpour, and J. Hanema, "A model predictive control design for linear parameter-varying systems in input-output form," *IEEE Conference on Decision and Control*, vol. 54, pp. 91–96, December 2015.

[11] S. M. Lee, J. H. Park, D. H. Ji, and S. C. Won, "Robust model predictive control for lpv systems using relaxation matrices," *IEEE Transactions on Control Theory and Applications*, vol. 1, pp. 1567–1573, November 2007.

[12] Y. Lu and Y. Arkun, "A quasi-min-max mpc algorithm for linear parameter-varying systems with bounded rate of change of parameters," *American Control Conference*, pp. 3234–3238, June 2000.

[13] X. Ping and N. Sun, "Dynamic output feedback robust model predictive control via zonotopic set-membership estimation for constrained quasi-lpv systems," *Journal of Applied Mathematics*, 2014.

[14] B. Ding, "Dynamic output feedback robust mpc for constrained quasi-lpv systems with both polyhedral and ellipsoidal bounds on estimation error," *Chinese Control Conference*, pp. 4143–4148, July 2013.

[15] T. H. S. Konstantine Kunz, Stephan M. Huck, "Fast model predictive control of miniature helicopters," *European Control Conference*, pp. 17–19, July 2013.

[16] J. Löfberg, *Linear Model Predictive Control: Stability and Robustness*. PhD thesis, Linköping Studies in Science and Technology, Linköping University, 2001.

[17] H. Michalska and D. Q. Mayne, "Robust receding horizon control of constrained nonlinear systems.," *IEEE Transactions on Automatic Control*, pp. 1623–1632, 1993.