Verification of Model Predictive Control Laws Using Weispfenning's Quantifier Elimination by Virtual Substitution Algorithm

K. Siaulys and J. M. Maciejowski

***Abstract*— A method based on a quantifier elimination algorithm is suggested for obtaining explicit model predictive control (MPC) laws for linear time invariant systems with quadratic objective and polytopic constraints. The structure of the control problem considered allows Weispfenning's 'quantifier elimination by virtual substitution' algorithm to be used. This is applicable to first order formulas in which quantified variables appear at most quadratically. It has much better practical computational complexity than general quantifier elimination algorithms, such as cylindrical algebraic decomposition. We show how this explicit MPC solution, together with Weispfenning's algorithm, can be used to check recursive feasibility of the system, for both nominal and disturbed systems. Extension to cases beyond linear MPC using Weispfenning's algorithm is part of future work.**

I. Introduction

MPC has found widespread success in industrial applications mainly because of its ability to naturally treat constrained optimal multivariate control problems [14]. Standard implementation of MPC relies on online optimisation to obtain the optimal sequence of inputs at each time step by solving an optimal control problem over a finite horizon. In the usual receding horizon approach, only the first input of the optimal input sequence obtained is applied to the controlled system. At the following time step, the measurement of the new current state is obtained and the whole procedure is repeated.

On the other hand, for conventional linear MPC, an analytic expression for the (piecewise-affine) optimal control law can be computed offline via multiparametric programming [13]. This approach, called explicit MPC, is useful for cases when implementation of MPC via online optimisation algorithm is not feasible; most of the computation is moved offline, while online computation gets reduced to a relatively simple calculation of the optimal input from a lookup table.

The idea of extending explicit MPC solutions to nonlinear polynomial systems with a polynomial objective and polynomial constraints has been extended in [15] (via Gröbner basis computation and an eigenvalue method for solving systems of polynomial equations) and [16] (via cylindrical algebraic decomposition). While both of these methods are extremely general, they are limited to problems with a small number of variables because of the complexity of the offline algebraic operations that are required.

*The research leading to these results has received funding from EPSRC and the European Union Seventh Framework Programme FP7/2007-2013 under grant agreement number 314 544, project "RECONFIGURE".

The authors are with University of Cambridge, Department of Engineering, Trumpington Street, Cambridge. CB2 1PZ. United Kingdom. {ks555,jmm}@eng.cam.ac.uk.

In this paper, we limit our attention to linear MPC problems. In general, Weispfenning's 'quantifier elimination by virtual substitution' algorithm [4] (which is much more efficient than cylindrical algebraic decomposition) is applicable as long as MPC optimisation problem can be expressed as an equivalent quantifier elimination problem where quantified variables appear at most quadratically. Hence, this method has potential to be applied to nonlinear MPC problems and this paper is a first step towards that direction by examining the linear case first. Additionally, once the explicit MPC solution is obtained, we show how the same efficient quantifier elimination algorithm can be used to calculate sets of recursively feasible states which are highly non-convex. Hence, this paper contributes towards verification of MPC laws using quantifier elimination methods.

The remainder of the paper is structured as follows: Section II discusses quantifier elimination in general and Weispfenning's quantifier elimination algorithm by virtual substitution in particular. Section III presents the suggested method for obtaining explicit MPC solution by using Weispfenning's algorithm, together with details of how it can be used to check recursive feasibility of the system. Section IV presents several illustrative examples demonstrating the applicability of the proposed methods; Section V provides conclusions.

II. Quantifier Elimination Algorithms

Many analysis and synthesis problems in control theory can be represented by the first order formula

$$\Phi(p_1,\ldots,p_m) \equiv Q_1 x_1 \ldots Q_n x_n \varphi(p_1,\ldots,p_m,x_1,\ldots,x_n), \tag{1}$$

where $Q_i \in \{\forall,\exists\}$ are either universal or existential quantifiers and φ is a quantifier-free formula constructed by conjunction ($\wedge$), disjunction ($\vee$) and negation ($\neg$) of atomic formulas of the form $f\,\rho\,0$ (where $f \in \mathbb{R}[p_1,\ldots,p_m,x_1,\ldots,x_n]$ is a polynomial and $\rho \in \{=,\neq,<,\leq\}$ is a relational operator). One of the ways to solve (1) is to feed Φ as an input formula to a quantifier elimination algorithm that outputs a quantifier-free formula φ' such that $\Phi(p_1,\ldots,p_m) = \varphi'(p_1,\ldots,p_m)\ \forall\, p_1,\ldots,p_m \in \mathbb{R}$.

Tarski discovered the first quantifier elimination algorithm [1] for the first order theory of real numbers. It exploited Sturm's theorem which allows one to count the number of real zeros of a polynomial in a single variable in some particular interval. Because of its high computational complexity, this algorithm is mainly of theoretical interest.

In 1975, Collins proposed a much more efficient quantifier elimination algorithm [2] based on cylindrical algebraic

decomposition which divides $\mathbb{R}^{n+m}$ into a disjoint set of regions, in each of which all polynomials from a given set are either positive, negative or zero. Again, application of this algorithm is limited to simple problems because of its complexity which is doubly exponential in the number of dimensions $n+m$.

On the other hand, for particular types of input formulas Φ, there exist quantifier elimination algorithms with much lower practical complexity. Suppose that quantified variables $x_1, \ldots, x_n$ appear at most quadratically in φ. In this case, Weispfenning's virtual substitution algorithm [4] (a generalisation of the linear quantifier elimination algorithm represented in [3]), whose complexity is independent of the number of free variables $p_1, \ldots, p_m$, is applicable.

This algorithm eliminates quantifiers as follows. Consider a first order formula $\exists x \varphi$, where φ is quantifier-free and constructed by conjunction and disjunction of atomic formulas of the form $f_i \ \rho_i \ 0,\ f_i = a_i x^2 + b_i x + c_i,\ i \in I$ which are at most quadratic in the quantified variable x, with a_i, b_i, c_i being polynomials that are independent of x. Denote:

$$\begin{aligned} D_i &= b_i^2 - 4a_i c_i, \\ \alpha_{i-} &= (-b_i - \sqrt{D_i})/(2a_i), \\ \alpha_{i+} &= (-b_i + \sqrt{D_i})/(2a_i). \end{aligned}$$

Let $\varphi[e/x]$ denote a formula that is obtained from φ by substituting e for x. For example, direct substitution of square-root expressions $e_i = \alpha_{i-}$ or $e_i = \alpha_{i+}$ for x in f_i will give expressions of the form:

$$f_i(e_i) = \frac{x_i + y_i\sqrt{D_i}}{z_i}. \tag{2}$$

To eliminate the presence of the radical, (2) is replaced by an equivalent expression depending on the type of relational operator ρ_i in the atomic formula $f_i \ \rho_i \ 0$ as follows:

$$\begin{aligned} f_i[e_i/x] = 0 \equiv& (x_i y_i \le 0) \wedge (x_i^2 - y_i^2 D_i = 0), \\ f_i[e_i/x] \ne 0 \equiv& (x_i y_i > 0) \vee (x_i^2 - y_i^2 D_i \ne 0), \\ f_i[e_i/x] \le 0 \equiv& (x_i z_i \le 0) \wedge (x_i^2 - y_i^2 D_i \ge 0) \vee \\ & (y_i z_i \le 0) \wedge (x_i^2 - y_i^2 D_i \le 0), \\ f_i[e_i/x] < 0 \equiv& (x_i z_i < 0) \wedge (x_i^2 - y_i^2 D_i > 0) \vee \\ & (y_i z_i \le 0) \wedge (x_i z_i < 0 \vee x_i^2 - y_i^2 D_i < 0). \end{aligned}$$

Finally, let I_1, I_2, I_3, I_4 be the set of indices $i \in I$ such that ρ_i is $=, \le, <, \ne$, respectively. Then $\exists x \varphi$ is equivalent to the following quantifier-free formula

$$\begin{aligned} &\bigvee_{i \in I_1 \cup I_2} ((a_i = 0 \wedge b_i \ne 0 \wedge \varphi[-c_i b_i^{-1}/x]) \vee \\ &\qquad (a_i \ne 0 \wedge D_i \ge 0 \wedge (\varphi[\alpha_{i-}/x]) \vee \varphi[\alpha_{i+}/x])) \vee \\ &\bigvee_{i \in I_3 \cup I_4} ((a_i = 0 \wedge b_i \ne 0 \wedge \varphi[(-c_i b_i^{-1} + \epsilon)/x]) \vee \\ &\qquad (a_i \ne 0 \wedge D_i \ge 0 \wedge (\varphi[(\alpha_{i-} + \epsilon)/x]) \vee \\ &\qquad \varphi[(\alpha_{i+} + \epsilon)/x])) \vee \varphi[-\infty/x], \end{aligned} \tag{3}$$

where ϵ denotes a positive infinitesimal. Expressions involving infinitesimals ϵ in (3) are replaced by equivalent ones not containing them according to the following set of rules:

$$\begin{aligned} f_i[(e+\epsilon)/x] = 0 \equiv& (a_i = 0) \wedge (b_i = 0) \wedge (c_i = 0), \\ f_i[(e+\epsilon)/x] \ne 0 \equiv& (a_i \ne 0) \vee (b_i \ne 0) \vee (c_i \ne 0), \\ f_i[(e+\epsilon)/x] \le 0 \equiv& f_i[(e+\epsilon)/x] = 0 \vee f_i[(e+\epsilon)/x] < 0, \\ f_i[(e+\epsilon)/x] < 0 \equiv& f_i[e/x] < 0 \vee f_i[e/x] = 0 \wedge \\ & \left(\frac{df_i}{dx}[e/x] < 0 \vee \frac{df_i}{dx}[e/x] = 0 \wedge a_i < 0\right). \end{aligned}$$

Similarly, expressions containing infinity in (3) are replaced by the following ones:

$$\begin{aligned} f_i[-\infty/x] = 0 \equiv& (a_i = 0) \wedge (b_i = 0) \wedge (c_i = 0), \\ f_i[-\infty/x] \ne 0 \equiv& (a_i \ne 0) \vee (b_i \ne 0) \vee (c_i \ne 0), \\ f_i[-\infty/x] \le 0 \equiv& f_i[-\infty/x] = 0 \vee f_i[-\infty/x] < 0, \\ f_i[-\infty/x] < 0 \equiv& \bigvee_{n=0}^{2} \left((-1)^n F_n < 0 \wedge \bigwedge_{m=n+1}^{2} F_m = 0\right), \\ & \text{where } F_0 = c_i, F_1 = b_i, F_2 = a_i. \end{aligned}$$

Even if all quantified variables appear at most quadratically in (1), Weispfenning's virtual substitution algorithm will not necessarily be able to iteratively eliminate all quantifiers, since eliminating one quantified variable by the procedure above may increase the degree of the remaining ones. On the other hand, full quantifier elimination by this algorithm is guaranteed if all quantified variables appear linearly except one, which appears quadratically and whose quantifier (either $\exists$ or $\forall$) is outermost or second to outermost in the prefix $Q_1 x_1 \ldots Q_n x_n$ in (1). As will be shown in Section III, many MPC problems (starting with a linear one) can be reduced to this type of first order formula.

Moreover, the virtual term substitution algorithm has been extended to formulas in which a single quantified variable appears at most cubically [5]. In principle, this algorithm can be extended to formulas in which the quantified variable appears with an unbounded degree, by exploiting Thom's Lemma for representation of real roots — for the state of the art, see [6], [7].

Weispfenning's virtual substitution algorithm has been implemented in tools like SyNRAC [8], which is a toolbox in Maple for solving real algebraic constraints, Mathematica [9] and Reduce/Redlog [10]. We will use the implementation in Mathematica for the computational examples in Section IV.

III. Problem Description

A. General MPC problem formulation

Consider a discrete-time linear time invariant system

$$x(k+1) = Ax(k) + Bu(k) \tag{4}$$

where $x \in \mathbb{R}^n$ is the state and $u \in \mathbb{R}^m$ is the input. Suppose we want to regulate (4) to the origin in such a way that polytopic constraints

$$\Psi(k) \equiv E(k)x(k) + F(k)u(k) \le b(k) \tag{5}$$

are satisfied at all time steps $k = 0, 1, \ldots$. Assuming measurement of the current state $x(0)$ is available, MPC solves

this regulation problem by solving the following optimisation problem

$$\min_{u(0),\ldots,u(N-1)} \sum_{i=0}^{N-1} \left(x(i)^T Q x(i) + u(i)^T R u(i) \right) + x(N)^T P x(N) \tag{6}$$
$$\text{s.t. } \Psi(i),\ i = 0,\ldots,N-1,\ E(N)x(N) \leq b(N)$$

at each time step where $x(i) = A^i x(0) + \sum_{j=0}^{i-1} A^{i-1-j} B u_j$ is the predicted state at time step i obtained given the current state $x(0)$ and the input sequence $\{u(0), u(1), \ldots, u(N-1)\}$, N is the control horizon and Q, R and P are state, input and terminal costs, respectively, with $Q = Q^T \geq 0$, $R = R^T > 0$, $P \geq 0$. (Here, $>$ and $\geq$ denote that the matrix is positive-definite and positive-semidefinite, respectively.)

Online implementation of MPC obtains the optimal input sequence $\{u^*(0), u^*(1), \ldots, u^*(N-1)\}$ at each time step by solving (6), then applies $u^*(0)$ to the system (4), and then repeats the whole process with the new current state.

On the other hand, an analytic expression of the optimal input $u^*(0)$ can be obtained offline as a piecewise-affine function of the state:

$$u_i^*(0) = K_i x(0) + G_i \text{ if } H_i x(0) \leq k_i,\ i = 1,\ldots,M \tag{7}$$

via multiparametric programming, which treats the current state $x(0)$ as a parameter [13]. Then online computation of the control law is reduced to the relatively simple computation of (7) depending on the value of the current state $x(0)$. Such an approach is referred to as 'explicit MPC'.

In Section III-B, we suggest an alternative method for obtaining the optimal explicit MPC control law by using quantifier elimination, and, in particular, Weispfenning's virtual substitution algorithm.

B. Explicit MPC solution via quantifier elimination

Let $L(i) = x(i)^T Q x(i) + u(i)^T R u(i)$ denote the i'th stage cost in the objective function in (6). Then, in a dynamic programming fashion, the single optimisation problem (6) in Nm variables $\{u(0), \ldots, u(N-1)\}$ can be expressed as N optimisation problems in m variables:

$$\begin{aligned} J(x(N-1)) = &\min_{u(N-1)} h(u(N-1), x(N-1)) = \\ &\min_{u(N-1)} L(N-1) + x(N)^T P x(N) \\ \text{s.t. } C(N-1) \equiv &\{x(N) = Ax(N-1) + Bu(N-1) \wedge \\ &\Psi(N-1) \wedge E(N)x(N) \leq b(N)\} \end{aligned}$$

$$\begin{aligned} J(x(N-2)) = &\min_{u(N-2)} h(u(N-2), x(N-2)) = \\ &\min_{u(N-2)} L(N-2) + J(x(N-1)) \\ \text{s.t. } C(N-2) \equiv &\{x(N-1) = Ax(N-2) + Bu(N-2) \wedge \\ &\Psi(N-2)\} \end{aligned}$$

$$\ldots$$

$$\begin{aligned} J(x(0)) = &\min_{u(0)} h(u(0), x(0)) = \\ &\min_{u(0)} L(0) + J(x(1)) \\ \text{s.t. } C(0) \equiv &\{x(1) = Ax(0) + Bu(0) \wedge \Psi(0)\} \end{aligned}$$

Each of these N optimisation problems can be expressed as a quantifier elimination problem

$$\exists u(i) \left(z \geq h(u(i), x(i)) \wedge C(i) \right),\ i = 0,\ldots,N-1 \tag{8}$$

where z represents a slack variable which will be used to obtain the optimal value function $J(x(i))$. Applying Weispfenning's algorithm to (8) produces an equivalent quantifier-free formula with no $u(i)$ dependence

$$\bigcup_{j=1,\ldots,n_i} \left(x(i) \in O_j(i) \wedge z \geq p_j(x(i)) \right) \tag{9}$$

with $p_j(x(i))$ a quadratic function of $x(i)$ and n_i overlapping regions $O_j(i)$ whose union is the set of feasible states:

$$O_1(i) \cup \ldots \cup O_{n_i}(i) \equiv \{x(i) : \exists u(i) \text{ such that } C(i)\}. \tag{10}$$

Since regions $O_j(i), j \in 1,\ldots,n_i$ overlap with each other, to obtain $J(x(i))$, we have to find the smallest $p_j(x(i))$ in each intersection of those regions. Therefore, region $O_1(i) \cup \ldots \cup O_{n_i}(i)$ is split into disjoint regions $D_1(i), \ldots, D_{m_i}(i)$ such that $D_l(i) \cap D_m(i) = \varnothing\ \forall l, m \in 1,\ldots,m_i$ according to the way overlapping regions intersect each other. (For illustration of this step, see the first example in Section IV together with Fig. 1 (a) and Fig. 1 (b).) Then for all $D_l(i), l \in 1,\ldots,m_i$, pick a point $x \in D_l(i)$ and evaluate all quadratic functions $p_j(x(i))$ corresponding to the regions $O_m(i), m \in 1,\ldots,n_i$ satisfying $O_m(i) \cap D_l(i) \neq \varnothing$ at $x(i) = x$. The quadratic function $p_j^*(x(i))$ with the smallest value $p_j(x)$ represents the actual optimal value function in the region $D_l(i)$, i.e. $p_j^*(x(i)) \equiv J(x(i))\ \forall x(i) \in D_l(i)$. Finally, after connecting neighbouring regions with the same optimal value function, $J(x(i))$ is obtained (see Fig. 1 (c) for an illustration).

Hence, the procedure to obtain the optimal explicit MPC control law $u_0^*(x)$ via quantifier elimination goes as follows. First, apply the quantifier elimination algorithm to the first order formula (8) with $i = N-1$, then obtain $J(x(N-1))$ via steps described in the previous paragraph. Next apply the quantifier elimination algorithm to (8) with $i = N-2$, with the $J(x(N-1))$ expression just obtained substituted into the formula. Repeat until the expression for $J(x_0)$ is obtained. Then the optimal input $u^*(0)$ is the solution of the system:

$$\begin{aligned} h(x(0), u(0)) &= 0, \\ J(x(0)) &= 0. \end{aligned} \tag{11}$$

Although only $u^*(0)$ is applied to the system (4), notice that we could have obtained optimal $u^*(1), \ldots, u^*(N-1)$, by solving systems analogous to (11).

Suppose that the obtained optimal explicit control law u_0^* is composed of M affine control laws in disjoint polytopic regions of the state space, and denote it as:

$$u_0^* = \bigcup_{i=1,\ldots,M} \left(u = K_i x + b_i \wedge H_i x \leq k_i \right). \tag{12}$$

In the following section, we will show how quantifier elimination, together with (12), can be used to analyse feasibility of the optimisation problem (6).

C. Checking feasibility via quantifier elimination

Let F_0 be the set of initial states $x(0)$ for which the optimisation problem (6) is feasible (i.e., has a solution). Hence, F_0 can be expressed as

$$F_0 \equiv \exists u(0)\exists u(1)\ldots\exists u(N-1)\bigwedge_{i=0}^{N-1}\Psi(i)\wedge(E(N)x(N)\leq b(N)) \tag{13}$$

and therefore can be calculated by substituting expressions for predicted states $x(i) = A^i x(0) + \sum_{j=0}^{i-1} A^{i-1-j}Bu_j,\ i = 1,\ldots,N-1$ and consequently eliminating quantifiers with Weispfenning's algorithm. Additionally, let the set $I_0 \equiv \mathbb{R}^n \setminus F_0$ denote initial states $x(0)$ for which optimisation problem (6) is *not* feasible.

Similarly, let F_1 be the set of states x for which (6) remains feasible after a single control update. Then

$$F_1 \equiv \exists u\,(u_0^* \wedge x \in F_0 \wedge Ax + Bu \in F_0) \tag{14}$$

which, again, can be calculated via quantifier elimination. Analogously, let I_1 be the set of states x for which (6) loses feasibility after a single time step:

$$I_1 \equiv \exists u\,(u_0^* \wedge x \in F_0 \wedge Ax + Bu \notin F_0)\,. \tag{15}$$

If $I_1 = \varnothing$, the control scheme is said to be recursively feasible (i.e., for all initially feasible states $x(0)$ and for all optimal input sequences, (6) remains feasible for all time steps k).

In general, let $F_k, k = 1,2,\ldots$ denote sets of states for which the optimisation problem (6) remains feasible for at least $k+1$ time steps, and $I_k, k = 1,2,\ldots$ sets of states for which (6) loses feasibility after k time steps. Then the sequence of sets F_k can be computed by applying Weispfenning's quantifier elimination algorithm to

$$F_k \equiv \exists u\,(u_0^* \wedge x \in F_{k-1} \wedge Ax + Bu \in F_{k-1})\,,\ k = 1,2,\ldots \tag{16}$$

while $I_k = F_0 \cap (F_0 \setminus F_k)$.

Now suppose that an additive disturbance $d(k) \in D$ acts on the system (4), i.e.

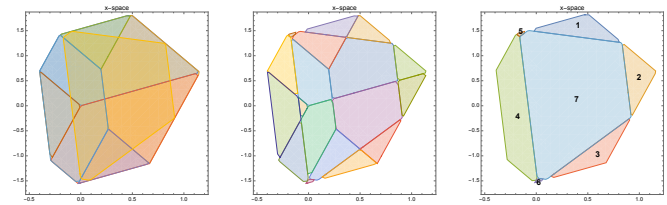
$$x(k+1) = Ax(k) + Bu(k) + d(k). \tag{17}$$

Then the set of states I_1 that lose feasibility after a single time step can be found by eliminating quantifiers from the formula:

$$\exists u\exists d\,(u_0^* \wedge d \in D \wedge x \in F_0 \wedge Ax + Bu + d \notin F_0)\,. \tag{18}$$

IV. Computational Examples

In this section, we apply the quantifier elimination approach to solve several MPC problems taken from [11]. All quantifier elimination computations were performed using Mathematica [9].



(a) Overlapping regions $O_1(0),\ldots,O_8(0)$. (b) Disjoint regions $D_1(0),\ldots,D_{18}(0)$. (c) Explicit MPC state-space partition.

Fig. 1: Steps in obtaining explicit MPC solution.

A. Explicit MPC solution

Consider a linear time-invariant discrete-time system with $A = \begin{bmatrix} \frac{3}{4} & -\frac{1299}{1000} \\ \frac{1299}{1000} & \frac{3}{4} \end{bmatrix}$, $B = \begin{bmatrix} -1 \\ 1 \end{bmatrix}$ and MPC controller designed using $N = 1$, $Q = I$, $R = 1$, together with the constraints $-1 \leq x_1 \leq 1$, $-1 \leq u_0 \leq 1$, $u_o \leq \frac{1}{5} + [1,1]x_1$ (as taken from the Example 2.3 in [11]).

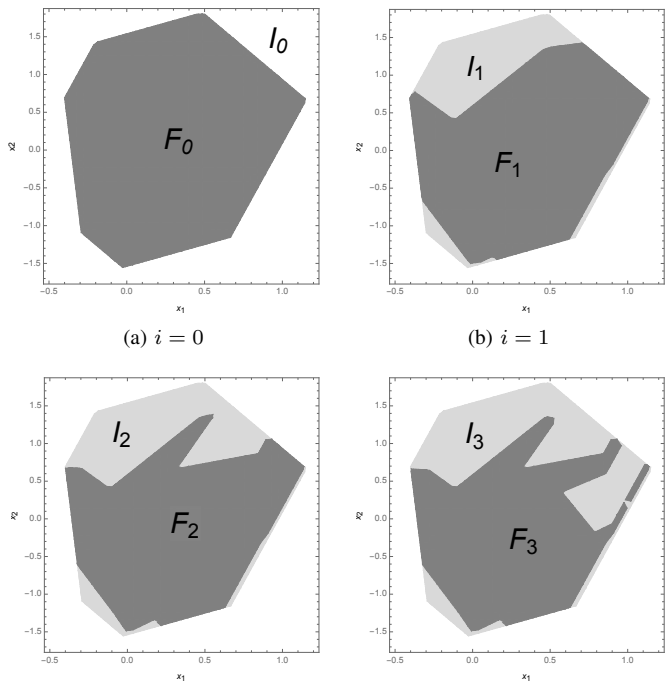
After performing quantifier elimination to find the optimal cost function z, as described in Section III, we obtain eight overlapping regions $O_1(0),\ldots,O_8(0)$, as shown in Fig. 1 (a). Then we split those overlapping regions into 18 disjoint regions $D_1(0),\ldots,D_{18}(0)$, as depicted in Fig. 1 (b). Finally, after picking the smallest cost function z in each of those regions, solving for the optimal input u_0^* and merging the regions $D_i(0), i = 1,\ldots,18$ with the same solution, we obtain the explicit MPC control law

$$u_0^*(x) = \begin{cases} -1 & \text{if } \begin{bmatrix} 0.8660 & 0.5000 \\ -0.5000 & 0.8660 \\ -0.2588 & -0.9659 \end{bmatrix} x \leq \begin{bmatrix} 1.3334 \\ 1.3334 \\ -1.4142 \end{bmatrix}, \\ & \text{(region \#1)} \\ [-1.299-0.750]x+1 & \text{if } \begin{bmatrix} 0.8660 & 0.5000 \\ 0.9659 & -0.2588 \\ -0.9982 & -0.0599 \end{bmatrix} x \leq \begin{bmatrix} 1.3334 \\ 0.9428 \\ -0.8944 \end{bmatrix}, \\ & \text{(region \#2)} \\ [0.750-1.299]x-1 & \text{if } \begin{bmatrix} -0.8345 & 0.5510 \\ 0.9659 & -0.2588 \\ 0.5000 & -0.8660 \end{bmatrix} x \leq \begin{bmatrix} -0.8944 \\ 0.9428 \\ 1.3334 \end{bmatrix}, \\ & \text{(region \#3)} \\ [2.049-0.549]x+0.2 & \text{if } \begin{bmatrix} 0.8660 & 0.5000 \\ -0.9982 & -0.0599 \\ 0.9659 & -0.2588 \\ -0.9659 & 0.2588 \\ -0.8660 & -0.5000 \\ 0.9982 & 0.0599 \end{bmatrix} x \leq \begin{bmatrix} 0.5333 \\ 0.3578 \\ 0.3771 \\ 0.5657 \\ 0.8000 \\ -0.0894 \end{bmatrix}, \\ & \text{(region \#4)} \\ [0.750-1.299]x+1 & \text{if } \begin{bmatrix} -0.5000 & 0.8660 \\ -0.8660 & -0.5000 \\ 0.8345 & -0.5510 \end{bmatrix} x \leq \begin{bmatrix} 1.3334 \\ -0.5333 \\ -0.8944 \end{bmatrix}, \\ & \text{(region \#5)} \\ 1 & \text{if } \begin{bmatrix} -0.9659 & 0.2588 \\ 0.5000 & -0.8660 \\ 0.2588 & 0.9659 \end{bmatrix} x \leq \begin{bmatrix} -0.3771 \\ 1.3334 \\ -1.4142 \end{bmatrix}, \\ & \text{(region \#6)} \\ [-0.183-0.683]x & \text{if } \begin{bmatrix} 0.9982 & 0.0599 \\ -0.8345 & 0.5510 \\ -0.2588 & -0.9659 \\ 0.2588 & 0.9659 \\ -0.9982 & -0.0599 \\ 0.8345 & -0.5510 \end{bmatrix} x \leq \begin{bmatrix} 0.8944 \\ 0.8944 \\ 1.4142 \\ 1.4142 \\ 0.0894 \\ 0.8944 \end{bmatrix} \\ & \text{(region \#7)} \end{cases} \tag{19}$$

whose state-space partition is illustrated in Fig. 1 (c). As expected, this result matches the one obtained using the Multi-Parametric Toolbox [12].

B. Recursive feasibility — nominal case

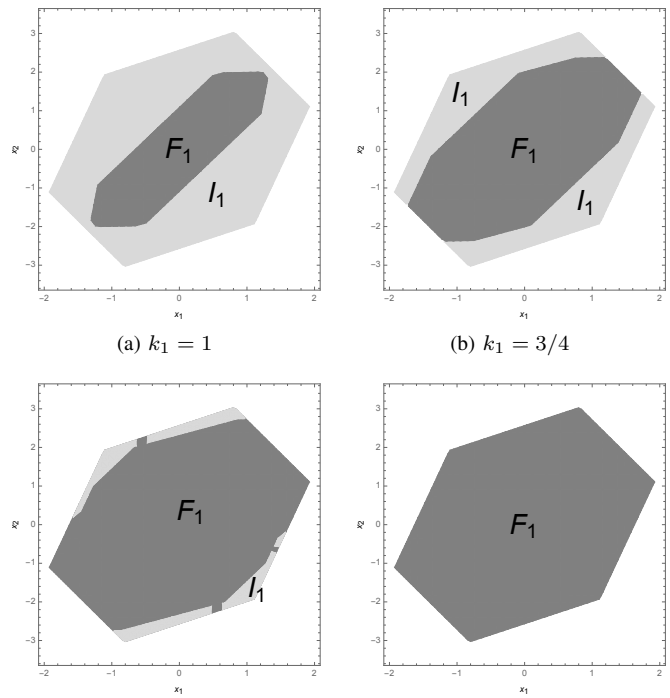
Is controller (19) recursively feasible? By eliminating quantifiers from the first order formula (15), we find that $I_1 \neq \varnothing$, as can be seen in Fig. 2 (b). Hence, recursive feasibility is violated. Moreover, by repeated application of



(a) $i = 0$ (b) $i = 1$

(c) $i = 2$ (d) $i = 3$

Fig. 2: Sets of states I_i which lose feasibility after i steps.



(a) $k_1 = 1$ (b) $k_1 = 3/4$

(c) $k_1 = 1/2$ (d) $k_1 = 1/4$

Fig. 3: Sets of states I_1 losing feasibility after a single time step.

quantifier elimination, we are able to calculate sets of states I_i which lose feasibility after i steps (see Fig. 2).

C. Recursive feasibility — disturbed case

Consider a nominal system $x(k+1) = Ax(k) + Bu(k)$ with $A = \begin{bmatrix} \frac{9}{20} & -\frac{3897}{5000} \\ \frac{3897}{5000} & \frac{9}{20} \end{bmatrix}$, $B = \begin{bmatrix} -1 \\ 1 \end{bmatrix}$, MPC controller designed for this system using prediction horizon $N = 3$, weights $Q = I$, $R = 1$, together with constraints $-1 \leq x_{i+1} \leq 1$, $-1 \leq u_i \leq 1$, $i = 0, 1, 2$ (taken from Example 5.2 in [11]).

Optimal explicit MPC solution is obtained for this system via the procedure described in Section III. Similarly, by quantifier elimination, we find that this controller is recursively feasible.

Now suppose that a disturbance $d(k) \in D$ acts on the system:

$$x(k+1) = Ax(k) + Bu(k) + d(k). \tag{20}$$

Let:

$$D = \left\{ \begin{bmatrix} d_1 \\ d_2 \end{bmatrix} \in \mathbb{R}^2 : -k_1 \leq d_1 \leq k_1, -k_1 \leq d_2 \leq k_1, k_1 \geq 0 \right\}. \tag{21}$$

For a given value of k_1, which states lose feasibility after a single time step? By posing this question as a quantifier elimination problem in the form (18), and then solving it with an elimination algorithm, we are able to find sets of states I_1 of interest, as depicted in Fig. 3. Hence, we see that with $k_1 = 1/4$, $I_1 = \varnothing$, and the system is recursively feasible for all allowable disturbances. Moreover, by keeping k_1 as a variable in (18) (rather than assigning a particular numerical value beforehand), and then applying Weispfenning's algorithm, we obtain the dependence of the set I_1 on the magnitude of the disturbance k_1, as depicted in Fig. 4 (a). As the magnitude of the disturbance k_1 gets larger, more and more states lose feasibility after a single time step, which is to be expected.

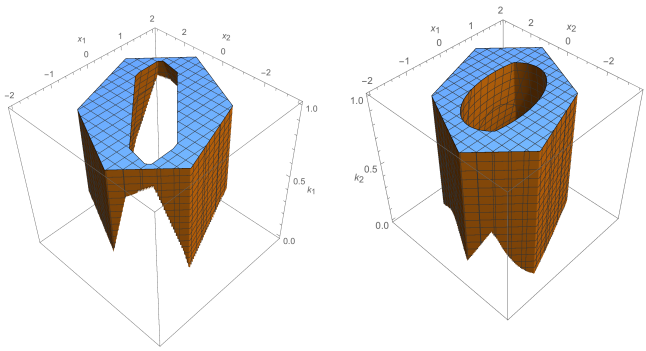
Now consider a state-dependant disturbance:

$$D = \left\{ \begin{bmatrix} d_1 \\ d_2 \end{bmatrix} \in \mathbb{R}^2 : d_1^2 + d_2^2 \leq k_2(x_1^2 + x_2^2) \right\}. \tag{22}$$

Despite the fact that two quantified variables (d_1 and d_2) appear quadratically in (18), Weispfenning's algorithm is capable of eliminating both of them and hence producing the dependence of the set I_1 on the parameter k_2, as illustrated in Fig. 4 (b).

V. Conclusions

A method based on quantifier elimination, as an alternative to multiparametric programming, has been proposed for obtaining explicit MPC solutions. Moreover, a way to analyse recursive feasibility via quantifier elimination, by taking advantage of the obtained explicit control law, has been suggested. In particular, repeated computation of sets F_k and I_k, in both nominal and disturbed cases, illustrates the capability of quantifier elimination methods to produce analytic expressions of highly non-convex semialgebraic sets. In this paper, we limited our attention to a particular (but common) type of linear MPC problem in order to test the the feasibility of applying Weispfenning's quantifier elimination algorithm which, in practice, has much better computational



(a) $0 \leq k_1 \leq 1$ (b) $0 \leq k_2 \leq 1$

Fig. 4: Dependence of the set I_1 on the uncertainty parameter $k_i, i = 1, 2$.

complexity than general quantifier elimination algorithms. The examples in section IV were computed in fractions of a second (examples A and B) or of the order of 1 second (example C), whereas attempts to use cylindrical algebraic decomposition (using Mathematica) were abandoned after about a day, with relentless use of memory.

In general, Weispfenning's algorithm is applicable as long as the MPC problem can be expressed as an equivalent quantifier elimination problem with a particular quantification structure (i.e., all quantified variables appear linearly except one, which appears quadratically and whose quantifier is outermost or second to outermost in the prefix). Hence, extension to cases beyond linear MPC using Weispfenning's algorithm is part of further work.

References

[1] A. Tarski, "A Decision Method for Elementary Algebra and Geometry", 1951. [Online]. Available: http://www.rand.org/pubs/reports/R109.html

[2] Collins, George E. "Quantifier elimination for real closed fields by cylindrical algebraic decompostion", *Automata Theory and Formal Languages 2nd GI Conference Kaiserslautern*, May 20–23, 1975, pp. 134–183.

[3] R. Loos, and V. Weispfenning, "Applying Linear Quantifier Elimination", *Comput. J.* 36, no. 5, 1993, pp. 450–461.

[4] V. Weispfenning, "Quantifier Elimination for Real Algebra — The Quadratic Case and Beyond", *Appl. Algebra Eng. Commun. Comput. 8*, no. 2, 1997, pp. 85–101.

[5] V. Weispfenning, "Quantifier Elimination for Real Algebra — The Cubic Case", in *Issac 1994: Proceedings of the International Symposium on Symbolic and Algebraic Computation*, 1994, pp. 258–263.

[6] K. Liiva, G. O. Passmore, and P. B. Jackson, "A note on real quantifier elimination by virtual term substitution of unbounded degree", in *Proceedings of the PAS workshop*, Vienna Summer of Logic, 2014.

[7] K. Marek, and T. Sturm, "A Generalized Framework for Virtual Substitution", arXiv preprint arXiv:1501.05826, 2015.

[8] H. Anai, and H. Yanami, "SyNRAC: A maple-package for solving real algebraic constraints", *Computational Science — ICCS 2003*, Springer Berlin Heidelberg, 2003, pp. 828-837.

[9] Wolfram Research, Mathematica 10.0, 2016.

[10] A. Dolzmann, and T. Sturm, "Redlog: Computer algebra meets computer logic", *Acm Sigsam Bulletin* 31, no. 2 (1997), pp. 2-9.

[11] J. Löfberg, "Oops! I cannot do it again: Testing for recursive feasibility in MPC", *Automatica*, vol. 48, no. 3, March 2012, pp. 550–555.

[12] M. Herceg, M. Kvasnica, C.N. Jones, and M. Morari, "Multi-Parametric Toolbox 3.0", In *Proc. of the European Control Conference*, pp. 502510, Zurich, Switzerland, July 1719 2013. http://control.ee.ethz.ch/mpt .

[13] A. Bemporad, M. Morari, V. Dua, E. N. Pistikopoulos, "The explicit linear quadratic regulator for constrained systems", *Automatica*, vol. 38, no. 1, January 2002, pp. 3-20.

[14] J. M. Maciejowski, *"Predictive control: with constraints"*, Pearson education, 2002.

[15] I. A. Fotiou, P. Rostalski, B. Sturmfels, and M. Morari, "An algebraic geometry approach to nonlinear parametric optimization in control", in *American Control Conference 2006*, pp. 6-pp. IEEE, 2006.

[16] I. A. Fotiou, P. A. Parrilo, and M. Morari, "Nonlinear parametric optimization using cylindrical algebraic decomposition", in *Decision and Control, 2005 and 2005 European Control Conference. CDC-ECC'05. 44th IEEE Conference on*, pp. 3735-3740. IEEE, 2005.