Automatica 48 (2012) 550–555

Contents lists available at SciVerse ScienceDirect

Automatica

journal homepage: www.elsevier.com/locate/automatica

Brief paper

Oops! I cannot do it again: Testing for recursive feasibility in MPC[☆]

Johan Löfberg

Division of Automatic Control, Department of Electrical Engineering, Linköping University, SE-581 83 Linköping, Sweden

ARTICLE INFO

Article history:
Received 8 March 2011
Received in revised form
23 June 2011
Accepted 9 August 2011
Available online 20 January 2012

Keywords:
Predictive control
Constrained control

ABSTRACT

One of the most fundamental problems in model predictive control (MPC) is the lack of guaranteed stability and feasibility. It is shown how Farkas' Lemma in combination with bilevel programming and disjoint bilinear programming can be used to search for problematic initial states which lack recursive feasibility, thus invalidating a particular MPC controller. Alternatively, the method can be used to derive a certificate that the problem is recursively feasible. The results are initially derived for nominal linear MPC, and thereafter extended to the additive disturbance case.

1. Introduction

An MPC controller repeatedly solves optimization problems online in order to decide the current input to the system. Although MPC opens up for general and advanced control schemes, it comes with a serious flaw. In contrast to, for instance, linear quadratic control which is stabilizing by construction, an MPC controller is not guaranteed to be stabilizing (Mayne, Rawlings, Rao, & Scokaert, 2000). Instead, stability is something which has to be built-in by judiciously defining the optimization problem. A related and even more severe issue is that feasibility of the optimization problem can be lost. The computations of optimal inputs may work for some time, but then, all of a sudden, the MPC controller has driven the state to a region where the optimization problem has no solution.

Almost all work on MPC assumes that these issues have been taken care of a-priori. The typical approach is to append the optimization problem with additional, somewhat artificial constraints, that will guarantee that loss of feasibility cannot occur (Mayne et al., 2000). When this is done, the MPC controller is said to be recursively feasible.

Unfortunately, although it is absolutely recommended from a theoretical point of view, it is not always wanted nor easy to construct an MPC controller which has an a-priori guarantee of recursive feasibility, either due to theoretical complications (Cagienard, Grieder, Kerrigan, & Morari, 2007; Gondhalekar, Imura, & Kashima, 2009), or pragmatic decisions in practice. Instead, we might have a situation where we are given an MPC controller, and the goal is to analyze it with respect to recursive feasibility. In this paper, the goal is primarily *invalidation* of an MPC controller, i.e., detect that it has problematic states where recursive feasibility is lost. A secondary objective is to find certificates of guaranteed recursive feasibility, both in the nominal and the disturbed case. We accomplish our goals by posing a suitably defined bilevel optimization problem originating from an application of Farka's lemma.

2. Problem formulation

We consider linear time-invariant discrete-time systems

$$x_{k+1} = Ax_k + Bu_k \tag{1}$$

with state $x \in \mathbb{R}^n$, control $u \in \mathbb{R}^m$, and (A, B) matrices of compatible dimensions. Although most of the setup can be generalized, the scope will be limited to a fairly basic definition of the MPC controller. As long as the resulting optimization problem is a linear or quadratic program linearly parameterized in the initial state x_k, the analysis and algorithms presented here hold. Introduce decision variables and predicted states

$$U_k = \begin{bmatrix} u_{k|k} \\ u_{k+1|k} \\ \vdots \\ u_{k+N-1|k} \end{bmatrix}, \qquad X_k = \begin{bmatrix} x_{k|k} \\ x_{k+1|k} \\ \vdots \\ x_{k+N|k} \end{bmatrix}, \tag{2}$$

where $x_{k+i+1|k} = Ax_{k+i|k} + Bu_{k+i|k}$. In this notation $u_{k+i|k}$ means a decision variable representing a thought future input at time k, and $x_{k+i+1|k}$ represents the associated state prediction. The MPC controller minimizes, at each time instance k, a finite-horizon quadratic performance objective $\sum_{i=0}^{N-1} x_{k+1+i|k}^T Q x_{k+1+i|k} + u_{k+i|k}^T R u_{k+i|k}$ under (assumed convex polytopic) state and control constraints. By stacking constraints and variables in a suitable

[☆] The author gratefully acknowledges funding from CENIIT at Linköping University. The material in this paper was not presented at any conference. This paper was recommended for publication in revised form by Associate Editor Martin Guay under the direction of Editor Frank Allgöwer.

E-mail address: johanl@isy.liu.se.

0005-1098/$ – see front matter © 2011 Elsevier Ltd. All rights reserved.
doi:10.1016/j.automatica.2011.12.003

fashion, it is straightforward to derive a model of the following quadratic programming form (since the current state is fixed regardless of the input trajectory, x_k is used instead of $x_{k|k}$ to reduce the notational burden)

$$U_k^* = \arg\min_{U_k} \frac{1}{2}U_k^T H U_k + U_k^T G x_k \tag{3a}$$

$$\text{subject to } Ex_k + FU_k \leq b. \tag{3b}$$

The actual input u_k to the system will be the first element of the optimal solution, $u_{k|k}^*$. After a time update, $k := k + 1$, a similar problem is solved, now with a different initial state and a new set of decision variables

$$U_{k+1}^* = \arg\min_{U_{k+1}} \frac{1}{2}U_{k+1}^T H U_{k+1} + U_{k+1}^T G x_{k+1} \tag{4a}$$

$$\text{subject to } Ex_{k+1} + FU_{k+1} \leq b. \tag{4b}$$

The central question is whether (4) is feasible when (3) was feasible. By using the optimal input to generate the succeeding state $x_{k+1} = Ax_k + Bu_{k|k}^*$, the question boils down to feasibility of the following constraint

$$E(Ax_k + Bu_{k|k}^*) + FU_{k+1} \leq b. \tag{5}$$

Let $\mathcal{F}_0$ denote the set of initially feasible states, $\mathcal{F}_0 = \{x : (\exists U \text{ such that } Ex + FU \leq b)\}$. The complement of the initially feasible set is denoted $\mathcal{I}_0$. The set of states which are initially feasible, but will lose feasibility after one time update is $\mathcal{I}_1 = \{x \in \mathcal{F}_0 : Ax + Bu^* \in \mathcal{I}_0\}$. These states will be called problematic states later on. The remaining states will not move to the infeasible region, $\mathcal{F}_1 = \{x \in \mathcal{F}_0 : Ax + Bu^* \notin \mathcal{I}_0\} = \mathcal{F}_0 \setminus \mathcal{I}_1$ and will thus be feasible for at least two time steps. Two sets of interest can now be defined using a recursion, $\mathcal{I}_i = \{x \in \mathcal{F}_{i-1} : Ax + Bu^* \in \mathcal{I}_{i-1}\}$ and $\mathcal{F}_i = \mathcal{F}_{i-1} \setminus \mathcal{I}_i$. The set $\mathcal{I}_i$ represents states which will lose feasibility after i time updates, and $\mathcal{F}_i$ represents all states which are guaranteed to stay feasible for at least $i+1$ steps. If $\mathcal{F}_i$ converges in a finite number of iteration, i.e., $\mathcal{F}_i = \mathcal{F}_{i-1}$ for some finite i, a recursively feasible set $\mathcal{F}_\infty$ has been found. Note that this set is nothing but the maximal positively invariant set of the closed-loop system.

A precise definition of the term recursive feasibility in our context might be suitable at this point.

Definition 2.1 (*Recursively Feasible*)**.** The MPC controller is recursively feasible if and only if for all initially feasible x_0 and for all optimal sequences of control inputs the MPC optimization problem remains feasible for all time.

In the notation above, this requires $\mathcal{I}_1$ to be empty. If x_0 is constrained to $\mathcal{F}_\infty$, the optimization problems will remain feasible, but the controller is not necessarily recursively feasible. Note also the difference to strongly recursively feasible solutions, which often occur in the literature and is the cornerstone of most stability proofs (Kerrigan, 2000; Mayne et al., 2000).

Definition 2.2 (*Strongly Recursively Feasible*)**.** The MPC controller is strongly recursively feasible if and only if for all initially feasible x_0 and for all sequences of feasible control inputs the MPC optimization problem remains feasible forever.

The typical approach to achieve (strongly) recursively feasible MPC controllers is to append the original MPC problem with a terminal state constraint $x_{k+N|k} \in \mathcal{T}$, typically based on (maximally) controlled invariant sets (Blanchini, 1999). This paper is however devoted to analysis of the controller, and does not propose any synthesis tools. Neither is our goal to explicitly compute the set of recursively feasible states $\mathcal{F}_\infty$ or the problematic states $\mathcal{I}_1$. As the following example illustrates, these sets are often severely nonconvex and thus intractably hard to compute, represent and work with in the general case.

Example 2.3. Let $A = \frac{3}{2}\begin{bmatrix} \cos\left(\frac{\pi}{3}\right) & -\sin\left(\frac{\pi}{3}\right) \\ \sin\left(\frac{\pi}{3}\right) & \cos\left(\frac{\pi}{3}\right) \end{bmatrix}$, $B = \begin{bmatrix} -1 \\ 1 \end{bmatrix}$. An MPC controller is designed using $N = 1$, $Q = I$ and $R = 1$, and the constraints are $-1 \leq x_{k+1|k} \leq 1$, $-1 \leq u_{k|k} \leq 1$ together with a mixed state-control constraint $u_{k|k} \leq 0.2 + \mathbf{1}^T x_{k+1|k}$. The numerical data associated with this model is given by

$$H = 3, \qquad G = \begin{bmatrix} 0.549 & 2.049 \end{bmatrix}$$

$$E = \begin{bmatrix} 0.750 & 1.299 & -0.750 & -1.299 & 0 & 0 & -2.049 \\ -1.299 & 0.750 & 1.299 & -0.750 & 0 & 0 & 0.549 \end{bmatrix}^T$$

$$F = \begin{bmatrix} -1 & 1 & 1 & -1 & 1 & -1 & 1 \end{bmatrix}^T,$$

$$b = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 0.2 \end{bmatrix}^T.$$

To study recursive feasibility of the controller, the explicit control law is computed using MPT (Multi-parametric Toolbox, (Kvasnica, Grieder, Baotić, & Morari, 2004)), and the sequence $(\mathcal{F}_i, \mathcal{I}_i)$ is computed using a variant of the invariance algorithm in Baotić (2005). As can be seen in Fig. 1, the sets of interest are highly nonconvex.

3. Conditions for feasibility and recursive feasibility

An MPC controller lacks recursive feasibility and is thus invalidated if it is possible to find a state which is feasible, but where the optimal input moves the state to a point where the MPC problem is infeasible. The tool used throughout this paper to analyze the constraint (5) is Farkas' lemma (Boyd & Vandenberghe, 2004).

Lemma 3.1. *Let $A \in \mathbb{R}^{m\times n}$ and $b \in \mathbb{R}^m$. Then either there is an $x \in \mathbb{R}^n$ such that $Ax \leq b$ or there is a $y \in \mathbb{R}^m$ such that $y \geq 0$, $y^TA = 0$ and $y^Tb < 0$.*

For given x_k and $u_{k|k}^*$ the feasibility question is

$$\exists U_{k+1} : FU_{k+1} \leq b - E(Ax_k + Bu_{k|k}^*). \tag{6}$$

According to Farkas' lemma, infeasibility of this constraint is equivalent to the existence of a vector y satisfying the following constraints

$$y \geq 0, \qquad y^TF = 0, \qquad y^T(b - E(Ax_k + Bu_{k|k}^*)) < 0. \tag{7}$$

Working with this set of equations alone is typically very conservative. For a reasonable analysis it is crucial to use the fact that the variables x_k and $u_{k|k}^*$ are coupled via optimality. Nevertheless, the condition immediately leads to a sufficient condition for strong recursive feasibility, thus ruling out the possibility for a problematic state.

Corollary 3.2 (*Strong Recursive Feasibility Test*)**.** *Let $r = b - E(Ax_k + Bu_{k|k})$. If $\min_{y\geq 0,\, y^TF=0, Ex_k+Fu_{k|k}\leq b} y^Tr$ is non-negative, the MPC controller is strongly recursively feasible.*

Proof. Follows immediately from Farkas' lemma. □

4. A bilevel programming approach to recursive feasibility analysis

The analysis above which treats x_k and $u_{k|k}^*$ as two uncoupled variables is often too weak. For stronger results, the fact that $u_{k|k}^*$ is optimal at the state x_k must be encoded in the problem formulation. Our approach to do this is to pose the whole problem as a bilevel optimization problem.

A bilevel problem in its most general form is composed of an *outer* and an *inner* problem with corresponding outer and inner

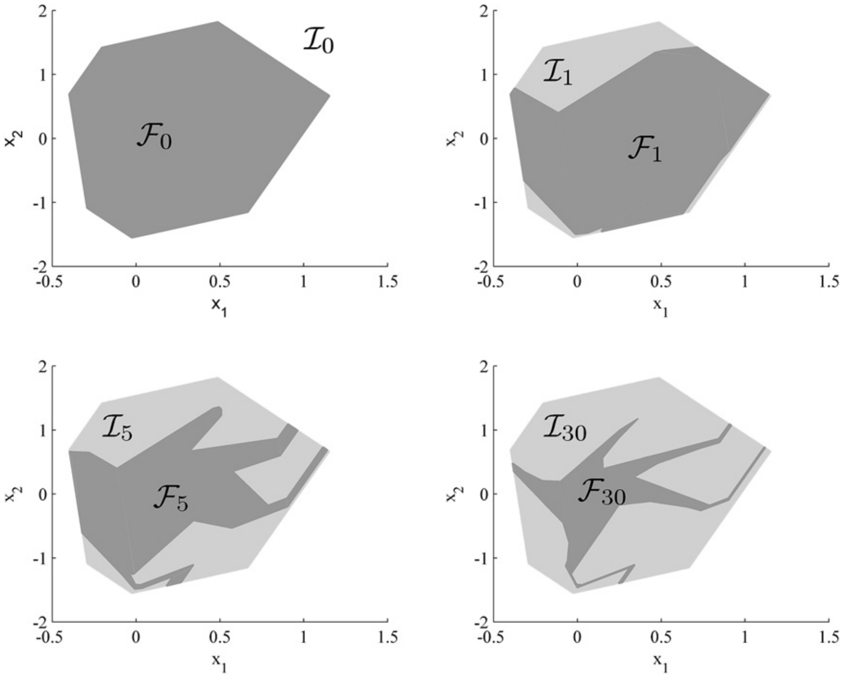

Fig. 1. The top left figure illustrates the set of initially feasible states $\mathcal{F}_0$ and its complement $\mathcal{I}_0$. The top right figure illustrates the set of initial states (soft shaded $\mathcal{I}_1$) which moves to the infeasible region, and the set of states which remain feasible for more than one time update, $\mathcal{F}_1$. The bottom left shows the set of states which become infeasible after five steps and its complement, while the bottom right shows the set of initial states which remain feasible for 30 time updates, i.e., an outer approximation of the recursively feasible set $\mathcal{F}_\infty$.

variables. The key feature which sets bilevel programming apart from standard optimization problems is that the inner variables, here denoted z, are constrained to be optimal with respect to an inner optimization problem which may depend on the outer variables x

$$\begin{aligned}&\min f(x, z^*)\\ &\text{subject to } (x, z^*) \in \mathcal{C}\\ &z^* = \arg\min_z\ h(x, z)\\ &\text{subject to } (x, z) \in \mathcal{D}.\end{aligned}$$

Bilevel problems are notoriously hard to solve, already in the case when both the inner and outer problems are linear programs, which pretty much is the most simple bilevel problem possible (Bard & Moore, 1990; Colson, Marcotte, & Savard, 2005; Jeroslow, 1985). The case when the inner problem is convex for fixed outer variables is somewhat more manageable than the general case, since the optimality condition on the inner variables can be encoded using KKT conditions. In the case of a linear or convex quadratic inner program, the KKT conditions give rise to a set of linear equality and inequality constraints, in combination with a complementarity constraint. Hence, although complementarity problems are NP-hard to solve, the problem at least has a nice structure and can, in principle, be solved using enumeration (Bard & Moore, 1990; Fortuny-Amat & McCarl, 1981; Murty, 1988).

The search for a state x_k and associated optimal input $u^*_{k|k}$ which drives the state to a position where infeasibility arise can thus be posed as the following bilevel program

$$\min_{y,x_k,U_k}\ y^T(b - E(Ax_k + Bu^*_{k|k})) \tag{8a}$$

$$\text{subject to } y \geq 0, \qquad y^T F = 0 \tag{8b}$$

$$U^*_k = \arg\min_{U_k}\ \frac{1}{2}U_k^T H U_k + U_k^T G x_k \tag{8c}$$

$$\text{subject to } Ex_k + FU_k \leq b. \tag{8d}$$

If the optimal objective of the outer problem is negative, we have found a proof of infeasibility. The inner problem is (by assumption on the MPC setup) a convex quadratic program, so we can replace the optimality condition with the corresponding KKT condition. Introduce a non-negative dual variable λ and append the outer problem with the stationarity (9c), and feasibility and complementarity (9d) constraints of the inner problem.

$$\min_{y,x_k,U_k,\lambda}\ y^T(b - E(Ax_k + Bu_{k|k})) \tag{9a}$$

$$\text{subject to } y \geq 0, \qquad y^T F = 0 \tag{9b}$$

$$HU_k + Gx_k + F^T\lambda = 0 \tag{9c}$$

$$0 \leq \lambda \perp b - Ex_k - FU_k \geq 0. \tag{9d}$$

To simplify notation while working with complementarity structures and KKT conditions, the notation $0 \leq \lambda \perp b - Ex_k - FU_k \geq 0$ collects the constraints $\lambda \geq 0$, $Ex_k + FU_k \leq b$ and $\lambda^T(b - Ex_k - FU_k) = 0$.

It goes without saying that the problem above is hard to solve. Not only does it involve the complementarity constraints $\lambda \perp b - Ex_k - FU_k$, but it also involves a bilinear objective function. Nevertheless, we have managed to cast the infeasibility analysis problem as a structured optimization problem.

From a numerical point of view, it is important to note that the problem definition is severely ill-posed. An arbitrary small but negative objective function would prove lack of recursive feasibility, while a solution with objective value 0 easily is constructed using $y = 0$. Such a small separation is dangerous in floating-point arithmetics and numerical optimization. Fortunately, it can be avoided by recognizing that the problem is homogeneous in y. Hence, it is advised to add a normalizing constraint of the type $y^T\mathbf{1} = 1$ ($\mathbf{1}$ denoting a vector of ones) or similarly.

This normalization is not only for purely numerical reasons, but actually increases the scope of the analysis significantly. When solving the optimization problem using a rigorous global solver such as BARON (Tawarmalani & Sahinidis, 2005) or YALMIP (Löfberg, 2004), such a solver typically computes increasingly tighter lower and upper bounds on the achievable objective value. Without the normalizing constraint, a trivial upper bound is 0, and the lower bound is negative and typically only approaches 0 in the limit if the true optimal value is 0. Hence, the lower bound would never allow us to terminate and say that the lower bound proves that the objective cannot be strictly negative. In other words, lack of recursive feasibility is possible to prove (by finding a particular x_k), but proof of recursive feasibility would be problematic. After a normalization though, the optimal value can be a positive number, and the lower bound can on its convergence to the upper bound pass 0, and thus allow us to terminate and conclude that the optimal objective value must be positive and recursive feasibility has to hold.[1]

4.1. Dual interpretation of normalized objective function

With the normalization $y^T\mathbf{1} = 1$, Farka's lemma and the objective function in our optimization problem has an intuitive geometric interpretation. Considering all variables but y as fixed, we have

$$\min_{y} \; y^T(b - E(Ax_k + Bu_{k|k})) \tag{10a}$$

$$\text{subject to } y \geq 0, \qquad y^TF = 0, \qquad y^T\mathbf{1} = 1. \tag{10b}$$

Using strong duality of linear programming, i.e., $\min_{y\geq 0, Ay=b} c^Ty = \max_{c-A^Tx\geq 0} b^Tx$ it follows that the minimization over y corresponds to

$$\max_{z,t} \; t \tag{11a}$$

$$\text{subject to } b - E(Ax_k + Bu_{k|k}) - Fz \geq t\mathbf{1}. \tag{11b}$$

This is nothing but (5) with an added slack-variable t and z taking the role of U_{k+1}. Hence, the normalized objective function, for fixed x_k and $u_{k|k}$, corresponds to the largest possible slack that can be added to all constraints in (5) while keeping feasibility. If this optimal value is negative for some x_k, it means that the constraints are infeasible for $x_{k+1} = Ax_k + Bu_{k|k}$ and the feasible set has to be extended by using a negative slack at that state.

4.2. A linear bilevel alternative

Inspired by the fact that the bilinear objective function disappeared in the dual interpretation above, we derive an alternative formulation of (9) and its normalized form. The important property which we will exploit (and used above) is that the optimization problem (9) is a disjoint bilinear problem (Al-Khayyal, 1992). Introduce the variable $\omega = (x_k, U_k, \lambda)$ and the optimization problem can be written as $\min_{\omega\in\Omega} \; \min_{y\in\mathcal{Y}} y^Tf(\omega)$. Above, we showed that the minimization over y corresponds, by duality, to a maximization over a new set of variables $(z, t) \in \mathcal{Z}$. Hence, we arrive at a minimax problem $\min_{\omega\in\Omega} \; \max_{\mathcal{Z}} t$. This minmax problem is by no means simple, so nothing is won in principle. What we have done however is that we have eliminated the bilinear objective function. The solution procedure we propose at this point is to continue using our tool of the day, bilevel programming. Characterize optimal solutions of the linear program $\max_{\mathcal{Z}} t$ using KKT conditions, thus introducing a new set of dual variables μ.

$$\begin{bmatrix} 0 \\ -1 \end{bmatrix} + \begin{bmatrix} F^T \\ \mathbf{1}^T \end{bmatrix} \mu = 0 \tag{12a}$$

$$0 \leq \mu \perp (b - E(Ax_k + Bu_{k|k}) - Fz - t\mathbf{1}) \geq 0. \tag{12b}$$

Collecting all constraints from (9) and (12) yields the following complementarity problem

$$\min_{x_k, U_k, \lambda, z, t} \; t \tag{13a}$$

$$\text{subject to (9c), (9d), (12a) and (12b).} \tag{13b}$$

Having transformed the problem to a complementarity problem with a linear objective means that we have opened up the possibility to use other alternatives for actually solving the problem. In principle, the complementarity structure allows us to solve the problem using brute-force enumeration and linear programming. In practice, it means that we can use standard mixed-integer reformulations of the problem (Colson et al., 2005).

It is easily seen that we are still solving exactly the same problem as before but in disguise. The introduced complementarity constraint $\mu^T(b - E(Ax_k + Bu_{k|k}) - Fz - t\mathbf{1}) = 0$ simplifies to $t = \mu^T(b - E(Ax_k + Bu_{k|k}))$ if we use $F^T\mu = 0$ and $\mathbf{1}^T\mu = 1$. Hence, μ plays the role of y and all we have done is exploited structure to trade a bilinear objective for a complementarity constraint.

4.3. Extension to systems with disturbances

The problem formulation can be extended to incorporate disturbances[2]

$$x_{k+1} = Ax_k + Bu_k + Mw_k. \tag{14}$$

One problem of interest is to check if there exists any state x_k, optimal input $u^*_{k|k}$ and disturbance $w_k \in \mathcal{W}$ such that the problem loses feasibility. Our prediction will now be different, thus changing our complementarity constraint to

$$\mu \perp (b - E(Ax_k + Bu_{k|k} + Mw_k) - Fz - t\mathbf{1}) \geq 0. \tag{15}$$

Going directly to the formulation with a linear objective, we obtain

$$\min_{x_k, U_k, w_k, \lambda, z, t} \; t \tag{16a}$$

$$\text{subject to (9c), (9d), (12a) and (15)} \tag{16b}$$

$$w_k \in \mathcal{W}(x_k, u_{k|k}). \tag{16c}$$

As one can see, the difference is minor, since the variable w_k is a variable related to the outer problem and does not complicate the KKT conditions. In the problem definition above, the exact representation of $\mathcal{W}$ has been omitted. The typical uncertainty case would be a polytopic model, but there is nothing in the analysis approach that limits the uncertainty model to this. For instance, the uncertainty may depend on the current state and input as indicated above. The only limitation in the uncertainty model is the tractability of actually solving the resulting optimization problem. A linear complementarity model, as in Section 4.2, is obtained if $\mathcal{W}(x_k, u_{k|k})$ is polytopic w_k, x_k and $u_{k|k}$.

By parameterizing a polytopic uncertainty model as $Kw \leq h\gamma$ where γ is a scaling parameter, and using γ as an objective, we can search for the smallest γ such that the problem is feasible with the

[1] Note that a local approach (using, e.g., a general nonlinear solver) might be sufficient in many cases. If the local solver finds a solution with negative objective value, the controller is invalidated. A global solver is only required if we want to prove recursive feasibility by showing positivity of a lower bound, or if we fail to find a problematic state using a local approach.

[2] Note that we are analyzing the impact of a disturbance, and not necessarily using a robust MPC synthesis setup. A robust MPC problem can of course be used to define the control input, assuming it in the end simplifies to a quadratic program of the type (3), possibly involving extra decision variables necessary in order to eliminate the disturbance from the uncertain problem.

constraint $t < 0$. Hence, finding the smallest disturbance set such that recursive feasibility is lost for some disturbance in that set

$$\min_{x_k,U_k,w_k,\lambda,z,t,\gamma} \gamma \tag{17a}$$

$$\text{subject to (9c), (9d), (12a) and (15)} \tag{17b}$$

$$Kw \leq h\gamma, \quad t < 0. \tag{17c}$$

4.4. Relaxing optimality

Almost all stability proofs of MPC are based on strong recursive feasibility and allow a relaxation of optimality by only requiring concurrent optimization problems to have a sufficient decrease in the objective. This is well motivated by practice where it is not always possible to run the optimization process to complete convergence. Instead, the process is terminated at a feasible but sub-optimal stage. In a feasible interior-point approach, this would typically mean that complementarity is violated. Encoding this in the analysis here is straightforward. Instead of using $\lambda \perp b - Ex_k - FU_k$, it is relaxed to, e.g., $\lambda^T(b - Ex_k - FU_k) \leq \epsilon$. Note though, that the linear bilevel alternative (13a) no longer can be used.

5. Numerical illustrations

This section implements the proposed method for a couple of scenarios. All computations were performed on a 3 GHz desktop PC in MATLAB using YALMIP (Löfberg, 2004) and the mixed-integer linear programming solver CPLEX 12 (CPLEX 12.2, 2011).

Example 5.1 (*Initial Example Revisited*). Solving the test (13) for a state violating recursive feasibility one finds, in a fraction of a second, the state $x_k = \begin{bmatrix} -0.2048 & 1.4214 \end{bmatrix}^T$ with associated input $u^*_{k|k} = -1$. Hence, the controller setup can be invalidated and the control engineer should be warned about the possible loss of feasibility.

Example 5.2 (*System with State Disturbances*). Let $A = \frac{9}{10}\begin{bmatrix} \cos\left(\frac{\pi}{3}\right) & -\sin\left(\frac{\pi}{3}\right) \\ \sin\left(\frac{\pi}{3}\right) & \cos\left(\frac{\pi}{3}\right) \end{bmatrix}$, $B = \begin{bmatrix} -1 \\ 1 \end{bmatrix}$. An MPC controller is designed using $N = 3$, $Q = I$ and $R = 1$, with the constraints $-1 \leq x_{k+1+i|k} \leq 1, -1 \leq u_{k+i|k} \leq 1$. The test (13) is solved and returns an optimal value of 0.22, thus proving that recursive feasibility holds.

A process disturbance $w_k \in \mathbb{R}^2$ is now added to the system, $x_{k+1} = Ax_k + Bu_k + w_k$. In a first experiment, the disturbance model was $-\frac{1}{2} \leq w_k \leq \frac{1}{2}$. Solving (16) yields an objective of -0.17, i.e., a combination of initial state x_k and disturbance w_k such that feasibility is lost has been found. To investigate how robust the controller is against process noise, we parameterize the uncertainty set as $-\gamma \leq w_k \leq \gamma$ and find the smallest possible γ such that recursive feasibility is lost, using (17). In less than a second, the optimal value 0.283 is returned. In other words, recursive feasibility is guaranteed as long as the absolute value of the process noise is smaller than 0.283.

Example 5.3 (*Non-convex Uncertainty Models*). As a second robustness test, we use the state-dependent uncertainty $w_k^T w_k \leq 0.1x_k^T x_k$ on the model in the previous example. With this setup, a linear complementarity model is no longer possible. Since the uncertainty description will lead to an indefinite quadratic constraint, we might just as well work with the initial normalized bilinear objective model (9), now involving also the uncertainty and thus leading to the objective $y^T(b - E(Ax_k + Bu_{k|k} + w))$. This bilinear complementarity problem is solved using the spatial branch-and-bound based global solver in YALMIP, with SNOPT (Gill, Murray, & Saunders, 2005) as the local upper bound solver. The branch-and-bound iterations terminate after only two iterations due to an upper bound with negative objective value. Hence, the system is not recursively feasible. The disturbance is now reduced to $w_k^T w_k \leq 0.01x_k^T x_k$. This problem turned out to be more challenging, but after 85 iterations, taking roughly 1 min, the lower bound on the achievable objective value turns positive, thus proving that recursive feasibility holds.

Example 5.4 (*4D Ball-and-plate*). As a final illustration, we revisit the ball-and-plate example from (Borrelli, 2002; Cagienard et al., 2007; Kvasnica, 2008). The system has four states, all constrained, and one constrained input. For simplicity, we disregard the reference signal. An MPC controller is designed, using the same setup as in Kvasnica (2008). In a first experiment, a controller with prediction horizon $N = 1$ is designed. Not surprising, test (13) immediately invalidates this setup and finds a problematic state. A terminal state constraint $x_{k+1|k} \in \mathcal{T}$ is added, where $\mathcal{T}$ is the largest positively invariant set inscribed in the feasible set for the closed-loop system controlled using the associated infinite horizon discrete-time LQ controller. As standard stability theory for MPC suggests, (13) is no longer able to find a complicating initial state. In a second setup, we increase the prediction horizon to $N = 10$, keep the terminal state constraint, but enforce $u_{k+1|k} = u_{k+2|k} = \cdots = u_{k+N-1|k}$. The test (13) is readily adjusted for this setup and returns a problematic state in a couple of seconds, thus numerically supporting the well-known fact that move-blocking can destroy standard stabilizing strategies (Cagienard et al., 2007; Gondhalekar et al., 2009). As an alternative to the proposed test, the explicit solution can be derived and analyzed. The explicit solution, computed in just under one minute using MPT, is comprised of 609 regions. After having obtained the explicit solution, a variant[3] of the invariance algorithm (Baotić, 2005) was applied and showed, in a couple of seconds, that the set $\mathcal{I}_1$ indeed was non-empty.

6. Summary

Feasibility and infeasibility analysis questions in MPC have been addressed using Farkas' lemma, bilevel programming and disjoint bilinear programming. The analysis has been devoted to very standard MPC settings, but can easily be extended to more general scenarios. A generalization to the uncertain case has been proposed and allows us to compute a measure of robustness of an MPC controller.

The proposed analysis approach has been applied to a number of small examples, and current work is focused on improving the computational results in order to handle larger problems. In particular, the global solver in YALMIP does not exploit the bilevel context. YALMIP has a built-in bilevel solver for fairly general bilevel problems, but for the particular class of problems addressed in this paper (nonconvex bilinear outer problem with convex quadratic inner problem) it has not performed well and was thus not used.

In the end, it is our hope that the methods in this paper can be a viable approach for analysis of systems where approaches based on polytopic geometry or brute-force gridding start struggling, say, in the range of 4–10 states.

References

Al-Khayyal, F. (1992). Generalized bilinear programming: part I. models, applications and linear programming relaxation. *European Journal of Operational Research*, *60*(3), 306–314.

Baotić, M. (2005). *Optimal control of piecewise affine systems-a multi-parametric approach-*. Ph.D. thesis, ETH Zurich, 2005.

[3] The algorithm in Baotić (2005) recursively computes the sets $\mathcal{I}_i$ and can be terminated immediately when $\mathcal{I}_1$ becomes non-empty.

Bard, J. F., & Moore, J. T. (1990). A branch and bound algorithm for the bilevel programming problem. *SIAM Journal on Scientific and Statistical Computing*, *11*(2), 281–292.

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

Borrelli, F. (2002). *Discrete time constrained optimal control.* Ph.D. thesis, ETH Zurich.

Boyd, S., & Vandenberghe, L. (2004). *Convex optimization*. Cambridge University Press.

Cagienard, R., Grieder, P., Kerrigan, E. C., & Morari, M. (2007). Move blocking strategies in receding horizon control. *Journal of Process Control*, *17*(6), 563–570.

Colson, B., Marcotte, P., & Savard, G. (2005). Bilevel programming: a survey. *4OR*, *3*(2), 87–107.

CPLEX 12.2. (2011). Available at www.ibm.com/software/integration/optimization/cplex-optimizer.

Fortuny-Amat, J., & McCarl, B. (1981). A representation and economic interpretation of a two-level programming problem. *The Journal of the Operational Research Society*, *32*(9), 783–792.

Gill, P. E., Murray, W., & Saunders, M. A. (2005). SNOPT: an SQP algorithm for large-scale constrained optimization. *SIAM Review*, *47*(1), 99–131.

Gondhalekar, R., Imura, J.-i., & Kashima, K. (2009). Controlled invariant feasibility — a general approach to enforcing strong feasibility in MPC applied to move-blocking. *Automatica*, *45*(12), 2869–2875.

Jeroslow, R. G. (1985). The polynomial hierarchy and a simple model for competitive analysis. *Mathematical Programming*, *32*(2), 146–164.

Kerrigan, E.C. (2000). *Robust constraint satisfaction: invariant sets and predictive control.* Ph.D. thesis, University of Cambridge.

Kvasnica, M. (2008). *Efficient software tools for control and analysis of hybrid systems,* Ph.D. thesis, ETH Zurich.

Kvasnica, M., Grieder, P., Baotić, M., & Morari, M. (2004). Multi-parametric toolbox (MPT), in: Hybrid systems: computation and control, pp. 448–462.

Löfberg, J. (2004). YALMIP: a toolbox for modeling and optimization in MATLAB. In: *Proceedings of the CACSD conference*, Taipei, Taiwan.

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

Murty, K. G. (1988). *Sigma series in applied mathematics*, *Linear complementarity, linear and nonlinear programming*. Helderman-Verlag.

Tawarmalani, M., & Sahinidis, N. V. (2005). A polyhedral branch-and-cut approach to global optimization. *Mathematical Programming*, *103*(2), 225–249.

Johan Löfberg was born in Jönköping, Sweden in 1974. He received his M.Sc. degree in Mechanical Engineering in 1998 and the Ph.D. degree in Automatic Control in 2003, both from Linköping University, Linköping, Sweden. He held a post-doctoral position at the Automatic Control Laboratory at ETH Zurich, Switzerland 2003–2006. Since 2006, he has worked as an Assistant Professor at Linköping University. His research interests include optimal and robust control in a model predictive control framework, and general aspects related to the use of optimization in control. He is the author and developer of the modeling language YALMIP.