INFEASIBILITY HANDLING IN LINEAR MPC SUBJECT TO PRIORITIZED CONSTRAINTS

Jostein Vada * **Olav Slupphaug** * **Bjarne A. Foss** *,[1]

* *Department of Engineering Cybernetics*
Norwegian University of Science and Technology, Norway

Abstract: All practical MPC implementations should have a means to recover from infeasibility. We propose an algorithm designed for linear state-space MPC which transforms an infeasible MPC optimization problem into a feasible one. The algorithm handles possible prioritizations among the constraints explicitly. Prioritized constraints can be seen as an intuitive and structural way to impose process knowledge and control objectives on the controlled process. The algorithm minimizes the constraint violations by solving a series of optimization problems, and the violation of a given constraint is minimized without affecting the higher prioritized constraints. An example shows the effect of implementing this algorithm on a simple process.

Keywords: Model based control, Constraints, Priority, Linear systems.

1. INTRODUCTION

During the last years, model predictive control (MPC) see e.g. (Rawlings *et al.*, 1994), has shown to become an attractive control strategy within the process industry. Important stability results within the area of linear MPC are given in (Rawlings and Muske, 1993). However, to fully exploit this stabilizing property, a means to recover from infeasibility of the associated optimization problem whenever possible is required, since generally, a practical MPC will sooner or later run into infeasibility problems. The infeasibility problems may e.g. be due to disturbances, operator invention, or actuator failure. If the input constraints represent physical limitations (which is often the case) they must be enforced at all times. The state constraints are often desirables and should hence be satisfied whenever possible. Thus, usually, only the state constraints can be relaxed in order to transform the optimization problem into a feasible one in the case of infeasibility.

There exist techniques which transform an infeasible MPC-problem into a feasible one. Rawlings and Muske (1993) proposed to remove the constraints at the beginning of the horizon, i.e. for samples up to some sample number j_1. This feature is also implemented in the QDMC algorithm reported by Garcia and Morshedi (1986), who also proposed a soft constraint solution which minimizes the square of the constraint violations. The use of soft constraints is a way to *avoid* running into infeasibility problems. Zheng and Morari (1995) show that global asymptotic stability can be guaranteed for linear time-invariant discrete time systems with poles inside the closed unit disc subject to hard input constraints and soft output constraints. In (Scokaert and Rawlings, 1998*b*) a method called optimal minimal time approach is proposed, which first minimizes the value of j_1, and then minimizes the size of the violation during the first j_1 samples of the prediction horizon.

Often, the state constraints are not equally important. One way to explicitly express this difference in importance is to give the state constraints different priorities. Imposing priority levels on the constraints is a systematic way to implement certain types of process knowledge, such as "avoid-

[1] Corresponding author, email: Bjarne.Foss@itk.ntnu.no, address: Dept. of Eng. Cybernetics, NTNU, 7030 Trondheim, Norway, phone: +4773594476, fax: +4773594399

ing shut-down is more important than discarding the product for some time, since start-up of the process is very expensive compared to discarding a certain amount of the product". The problem studied in this paper is how to allow control to be continued in the presence of infeasibility of the state constraints taking into account the information contained in the prioritization.

There are some existing techniques which take the prioritization levels into account when recovering from infeasibility. IDCOM-M from Setpoint Inc. provides a means for recovery from infeasibilities which involves prioritization of the constraints (Qin and Badgwell, 1996). When the calculation becomes infeasible, the lowest prioritized hard constraint is dropped, and the calculation is repeated. PCT from Profimatics also uses constraint prioritization when recovering from infeasibility (Qin and Badgwell, 1996).

Meadowcroft *et al.* (1992) have developed a modular multivariable controller (MMC), which is based on the solution of a multiobjective optimization problem using the strategy of lexicographic goal programming where the objectives have different priorities. They proposed a detailed methodology for the design of steady state MMCs (they have left the detailed design of dynamic MMCs for a forthcoming publication).

In (Tyler and Morari, 1997, 1996) it is presented an approach using integer variables for solving infeasible linear MPC problems where the constraints have different priorities. The minimization of the size of the violation of the constraints is done by solving a sequence of mixed integer optimization problems. They compare their methodology with conventional MPC by using weights and slack-variables on an example with 4 prioritized constraints. By trial and error, they find weights which give approximately the same performance for one specific disturbance. When a different disturbance enters their example process, the simulation results show that when using their approach, the two a highest prioritized constraints are fulfilled, while only the highest prioritized constraint is fulfilled when the conventional approach is used. This example supports the statement in (Qin and Badgwell, 1996), that for large problems it is not easy to translate control specifications into a consistent set of relative weights for a single objective function.

The main difference between our approach and the one presented in (Tyler and Morari, 1997, 1996) (the so-called rigorous one) is that the latter approach results in a sequence of mixed integer LP (or mixed integer QP) problems in addition to the original MPC optimization problem, while our approach results in a sequence of LP (QP) problems in addition to the original MPC optimization problem. In both approaches, whether each step in the resulting sequence of optimization problems is an LP problem or QP problem depends on how the slackvariables associated with the constraints are penalized.

The outline of the paper is as follows: The next section presents the problem definition and the MPC formulation used. Then the algorithm which transfers an infeasible MPC optimization problem into a feasible one is presented, followed by a simulation example. The last section contains a discussion and some concluding remarks.

2. PROBLEM DEFINITION

2.1 *MPC formulation*

The notation and MPC-formulation used here is adopted from (Scokaert and Rawlings, 1998*a*). Consider the time-invariant, linear, discrete time system described by

$$x_{t+1} = Ax_t + Bu_t, \tag{1}$$

where $x_t \in \mathbb{R}^n$ and $u_t \in \mathbb{R}^m$ are the state and input vectors at discrete time t. It is assumed that (A, B) is stabilizable. The control objective is to regulate the state of the system optimally to the origin. The quadratic objective is defined over an infinite horizon and is given by

$$\phi(x_t, \pi) = \sum_{j=t}^{\infty} x_{j|t}^T Q x_{j|t} + u_{j|t}^T R u_{j|t}, \tag{2}$$

where $Q \geq 0$ and $R > 0$ are symmetric weighting matrices such that $(Q^{1/2}, A)$ is detectable, $\pi = \{u_{t|t}, u_{t+1|t}, \dots\}$, x^T is the transposed of x, and

$$x_{j+1|t} = Ax_{j|t} + Bu_{j|t}, \quad t \leq j \tag{3}$$

with $x_{t|t} = x_t$. The linear constraints are

$$\begin{aligned} Hx_{j|t} &\leq h,\ t < j \\ Du_{j|t} &\leq d,\ t \leq j \end{aligned}$$

where $h \in \mathbb{R}_+^{n_h}$ and $d \in \mathbb{R}_+^{n_d}$ ($\mathbb{R}_+$ is the positive reals) define the constraint levels, and H and D are the state and input constraint matrices respectively. The MPC optimization problem can now be defined as follows:

$$\min_{\pi} \phi(x_t, \pi)$$

subject to:

$$\begin{array}{lll} x_{t|t} & = x_t & \\ x_{j+1|t} & = Ax_{j|t} + Bu_{j|t}, & t \leq j \\ Hx_{j|t} & \leq h, & t < j \\ Du_{j|t} & \leq d, & t \leq j \\ u_{j|t} & = -Kx_{j|t}, & t + N \leq j \end{array} \tag{4}$$

The constraints need only be satisfied on a finite horizon to guarantee satisfaction on the infinite horizon (Rawlings and Muske, 1993). This form of MPC has Nm decision variables, and can be solved with standard quadratic programming methods. K is discussed below. The performance index in the above equation can be formulated as (Rawlings and Muske, 1993)

$$\phi(x_t, \pi) = \sum_{j=0}^{N-1} (x_{j+t|t}^T Q x_{j+t|t} + u_{j+t|t}^T R u_{j+t|t}) + x_{j+N|t}^T \tilde{Q} x_{j+N|t}$$

where $\tilde{Q}$ is the solution of the matrix Lyapunov equation

$$\tilde{Q} = Q + K^T R K + (A - BK)^T \tilde{Q} (A - BK).$$

The feedback matrix K can be chosen in several ways. In the rest of this paper, $K = 0$ is used. In order to obtain stability, the unstable modi of the predictor, $x_{t+N|t}^u$, are zeroed at the Nth predicted sample (end point constraint), i.e.,

$$x_{t+N|t}^u = 0. \tag{5}$$

The feedback law is defined by receding horizon implementation of the optimal open-loop control. Given the optimal open-loop control strategy $\pi^*(x_t) = \{u_{t|t}^*(x_t), u_{t+1|t}^*(x_t), \dots\}$, the control law is thus given by

$$u_t(x_t) = u_{t|t}^*(x_t). \tag{6}$$

2.2 *Compact problem formulation*

Assume that the system, performance index, and predictor are given by (1), (2), and (3), respectively, and that the MPC problem formulation is given by (4) with $K = 0$ and the additional end point constraint (5). The problem studied in this paper is how to relax the state constraints in an optimal manner subject to their prioritization when the optimization problem defined by the MPC formulation becomes infeasible (e.g. due to a disturbance). The method solving this problem must be computationally implementable, and must not interfere with the control law defined by (4) and (6) when the optimization problem (4) is feasible.

3. SOLUTION APPROACH

3.1 *The algorithm*

When it is impossible to satisfy all state constraints simultaneously, it is desirable to satisfy as many of the highest prioritized constraints as possible. The violations of the other (infeasible) constraints should be minimized, taking their relative prioritization into account. The method described here is an *extension* of the theory presented in (Scokaert and Rawlings, 1998*b*), such that the constraint violations are minimized according to their priorities. Operating at Pareto-optimal points in the "size of violation - duration of violation" space is the goal. The MPC problem defined in (4) (with $K = 0$), can be rewritten as[2]

$$\min_{\pi} \phi(x_t, \pi) \tag{7}$$

subject to:

$$\text{"hard" hard constraints} \begin{cases} x_{t|t} &= x_t \\ x_{t+N|t}^u &= 0 \\ x_{j+1|t} &= A x_{j|t} + B u_{j|t}, \quad t \le j \\ D u_{j|t} &\le h, \quad t \le j \\ u_{j|t} &= 0, \quad t + N \le j \end{cases} \tag{8}$$

$$\text{"soft" hard constraints} \begin{cases} c_1 : \ H^1 x_{j|t} \le h^1, \ t < j \\ \ \vdots \\ c_{n_c} : H^{n_c} x_{j|t} \le h^{n_c}, \ t < j \end{cases} \tag{9}$$

where the constraints marked as "hard" hard constraints cannot under any circumstances be violated, since they are either physical limitations on the process, or related to zeroing the unstable modi at the end of the prediction horizon, or decided by the move horizon N which is assumed to be fixed. The constraint sets $\{c_1, \dots, c_{n_c}\}$ are constructed such that constraint set c_i has higher priority than c_{i+1}. A constraint set is composed of one or more scalar constraints having the same priority. $H_i \in \mathbb{R}^{n_{c_i} \times n}$, $h_i \in \mathbb{R}_+^{n_{c_i}}$, where n_{c_i} is the number of constraints in constraint set c_i. An algorithm solving the problem of infeasibility subject to the prioritization among the constraints is presented next. In the algorithm, a sequence defined as $\{c_l, \dots, c_m\}$, $l > m$, is interpreted as the empty set.

Step 1: Solve the optimization problem defined by (7), (8) and (9). If a feasible solution exists, the optimal solution (π^*) is found - terminate. Else, the problem infeasible. Go to step 2.

Step 2: Check existence of a solution to (8). If there does not exist any solution, the process cannot be stabilized with the given controller. Some kind of extraordinary action has to be taken. Else, if there exist a solution, set $k \leftarrow 1$ and go to Step 3. Note that the integer k is indexing the constraints, and is *not* related to time.

Step 3 Check existence of a solution to (8) and (9), but without constraint sets $\{c_{k+1}, \dots, c_{n_c}\}$. Go to Step 4.

Step 4 If a feasible solution is found, set $k \leftarrow k + 1$, and go to Step 3. Else, if no feasible solution is found, constraint set $\{c_1, \dots, c_k\}$ cannot be satisfied simultaneously. Go to Step 5.

Step 5 Step 4 showed that constraint set c_k cannot be satisfied when $\{c_1, \dots, c_{k-1}\}$ are satisfied. Minimize the violation of constraint set c_k, i.e. compute optimal slack variables $(\Delta h_{j|t}^k)^* \in \mathbb{R}_+^{n_{c_k}}$, such that

---

[2] Detectability of $(Q^{1/2}, A)$, which is a general requirement in Section 2.1, is not nescessary for stability here because of the end point constraint $x_{t+N|t}^u = 0$.

$$c'_k\colon H^k x_{j|t} \le h^k + (\Delta h^k_{j|t})^*, \quad t < j \qquad (10)$$

and $\{c_1, \dots, c_{k-1}\}$ are satisfied. There are several ways to compute the optimal slack variables, according to the control policy, see the discussion at the end of this section. Set $n_s \leftarrow 1$, where n_s is number of softened "soft" hard constraints. If $k < n_c$, i.e. there are more slack variables to be computed, go to Step 6, else go to Step 8.

Step 6: Minimize the violation of constraint set c_{k+n_s}, i.e. compute optimal slack variables $(\Delta h^{k+n_s}_{j|t})^*$, using the same strategy as in Step 5, such that $\{c_1, \dots, c_{k-1}, c'_k, \dots, c'_{k+n_s-1}\}$ are satisfied. Go to Step 7.

Step 7 If $k + n_s < n_c$, i.e. there are more slack variables to be computed, set $n_s \leftarrow n_s + 1$ and go to Step 6, else go to Step 8.

Step 8 At this step, the status is as follows: Constraint sets $\{c_1, \dots, c_{k-1}\}$ are not violated, and $\{c_k, \dots, c_{n_c}\}$ are replaced by $\{c'_k, \dots, c'_{n_c}\}$ such that there exist a solution which fulfills $\{c_1, \dots, c_{k-1}, c'_k, \dots, c'_{n_c}\}$. Now, with the last degrees of freedom (if any), minimize the performance index (7) subject to these constraints.

As stated in Step 5 above, there are several ways to compute the optimal slack variables for a given constraint set. In Section 4 below, the optimal minimal time approach (Scokaert and Rawlings, 1998*b*) is used to compute the slack variables within each constraint set. Considering constraint set c_k, the computation can be described as follows: Let $\kappa^k(x_t)$ denote the least integer such that c_k can be fulfilled when $j \ge t + \kappa^k(x_t)$. Given $\kappa^k(x_t)$, the least (in some sense) $\Delta h^k_t \in \mathbb{R}^{n_{c_k}}_+$, i.e. $(\Delta h^k_t)^*$, is computed such that c'_k defined in (10) is feasible if $(\Delta h^k_{j|t})^* = (\Delta h^k_t)^*$ when $t < j < t + \kappa^k(x_t)$ and $(\Delta h^k_{j|t})^* = 0$ when $t + \kappa^k(x_t) \le j$. Another method is, for each constraint set, to introduce different priorities for every sample. If, for example, the constraints at predicted sample q in constraint set k has higher priority than the constraints at predicted sample $q-1$ in constraint set k, then the minimal duration of the constraint violation is obtained, but the size of the violations will generally differ from the corresponding size of violations resulting from the optimal minimal time approach. Another method for computing $(\Delta h^k_{j|t})^*$ is to minimize $\sum_{j=t+1}^{P} (\Delta h^k_{j|t})^T W_j \Delta h^k_{j|t}$ subject to c_1 to c_{k-1} and $H^k x_{j|t} \le h^k + \Delta h^k_{j|t}$, $t < j \le t+P$, where $W_j \in \mathbb{R}^{n_{c_k} \times n_{c_k}}$ is a weighting matrix, and P is sufficiently large such that $\Delta h^k_{j|t} = 0$, $j > t + P$ is feasible.

3.2 *Stability*

The controller proposed in this paper is based on the controller given in (Rawlings and Muske, 1993), which is shown to be exponentially stabilizing in (Scokaert and Rawlings, 1998*a*), and is essentially an add-on of infeasibility handling in an optimal manner taking priorities into account. This add-on does not interfere with any of the stabilizing properties of the controller given in (Rawlings and Muske, 1993), hence stability is retained.

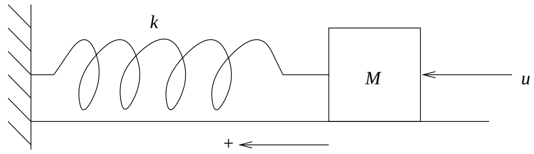

Fig. 1. An idealized mass-spring system.

4. EXAMPLE

4.1 *Process*

The proposed method described above will now be implemented on a simple example, and compared, insofar it is possible, with the optimal minimal time approach described in (Scokaert and Rawlings, 1998*b*). The process used in this example is an idealized mass-spring system which is illustrated in Figure 1. The spring is assumed to be linear, and the mass slides without any friction. There is a force u directed horizontally on the mass. It is assumed that both the position and the velocity of the mass are ideally measured, and that the spring constant k and the mass M both are equal to 1.0. By using exact discretization with sample time equal to 0.5 s, the system is given by the following equations:

$$x_{t+1} = Ax_t + Bu_t,$$

where

$$A = \begin{bmatrix} 0.8776 & 0.4794 \\ -0.4794 & 0.8776 \end{bmatrix}, \quad B = \begin{bmatrix} 0.1224 \\ 0.4794 \end{bmatrix}.$$

The MPC problem is given by (4) and (2) with $K = 0$, $Q = I$, $R = 1$, $N = 5$ and

$$H = \begin{bmatrix} 1 & 0 \\ -1 & 0 \\ 0 & 1 \\ 0 & -1 \end{bmatrix}, \; h = \begin{bmatrix} 0.25 \\ 0.25 \\ 0.25 \\ 0.25 \end{bmatrix}, \; D = \begin{bmatrix} 1 \\ -1 \end{bmatrix}, \; d = \begin{bmatrix} 0.6 \\ 0.6 \end{bmatrix}.$$

The predictor is equal to the process. The input constraint has always to be satisfied, and the prioritization of the state constraints is as follows, in descending order:

(1) Constraints on the mass position, i.e. $|x_{j|t,1}| \le 0.25$, $j > t$.
(2) Constraints on the mass velocity, $|x_{j|t,2}| \le 0.25$, $j > t$.

Using the notation in Section 3.1, the following constraint sets are defined:

$$c_1 : H^1 x_{j|t} \le h^1, \;\; j > t,$$
$$c_2 : H^2 x_{j|t} \le h^2, \;\; j > t$$

where

$$H^1 = \begin{bmatrix} 1 & 0 \\ -1 & 0 \end{bmatrix}, \; H^2 = \begin{bmatrix} 0 & 1 \\ 0 & -1 \end{bmatrix}, \; h^1 = h^2 = \begin{bmatrix} 0.25 \\ 0.25 \end{bmatrix}.$$

4.2 *Simulation: Case 1 - applying the proposed approach*

At time $t = 0$, a state disturbance of $[0.8, -0.4]^T$ enters the system and the approach described in Section 3.1 is used to recover from infeasibility. Step 1 to 4 in the algorithm show that only the "hard" hard constraints can be satisfied at time $t = 0$. In Step 5, the optimal minimal time approach described in (Scokaert and Rawlings, 1998*b*) is used to compute the minimal violation of constraint set c_1: First the minimal time, $\kappa^1(x_t)$, beyond which the constraints c_1 can be satisfied is computed. Given $\kappa^1(x_t)$, the following LP is solved to compute the minimal size of the constraint violation of constraint set c_1:

$$\min_{\pi_t, \Delta h_t^1} S\Delta h_t^1, \text{ subject to}$$
$$\begin{array}{lll} \text{"hard" hard constraints} & & \\ H^1 x_{j|t} \le h^1 + \Delta h_t^1, & t < j < t + \kappa^1(x_t) & \quad (11) \\ H^1 x_{j|t} \le h^1, & t + \kappa^1(x_t) \le j & \end{array}$$

where $S = [1 \; 1]$. At time $t = 0$, the optimal slack variables computed by this LP, are $(\Delta h_0^1)^* = [0.2944, 0.0]^T$, and $\kappa^1(x_0) = 3$. Let c_1' denote the two last constraints in (11) when $\Delta h_t^1 = (\Delta h_t^1)^*$. In Step 6, the minimal violation of constraint set c_2 is computed by first computing the minimal time, $\kappa^2(x_t)$, beyond which the constraints c_2 can be satisfied, subject to the "hard" hard constraints and c_1'. Given $\kappa^2(x_t)$, the minimal size of the constraint violation of constraint set c_2 is computed by a LP problem similar to (11), but with c_1' added to the hard constraints. The optimal slack variables computed by this LP at $t = 0$ are $(\Delta h_0^2)^* = [0.3509, 0.0]^T$, and $\kappa^2(x_0) = 4$. Let c_2' denote the relaxed constraint set c_2. In Step 8, the performance index (7) is minimized subject to the "hard" hard constraints, c_1' and c_2'. The receding horizon implementation using this strategy on the example process with the given disturbance results in the response shown in the left part of Figure 2. It can be observed from the figure that the constraints in constraint set c_1 are satisfied when $j \ge 2$, while $\kappa^1(x_0) = 3$. This difference in open- and closed-loop is due to the receding horizon nature of MPC and finite move horizon (N).

4.3 *Simulation: Case 2- the optimal minimal time approach*

The same disturbance as in Case 1 enters the system at time $t = 0$, but now all slack variables are minimized upon simultaneously by using the optimal minimal time approach presented in (Scokaert and Rawlings, 1998*b*), i.e. there are no prioritization among the constraints. This approach is equal to the approach used in the previous section, but with all constraints collected in constraint set c_1, and gives the response shown in the right part of Figure 2. It can be seen that the violations of constraint set c_1 are less, both in time and size, when the approach used in this paper is applied, compared to the plain optimal minimal time approach. The expense is larger violations of constraint set c_2, this is of course due to the prioritization. It can be observed that in this case, the approach described in this work causes a longer time period with constraint violations, compared to the plain optimal minimal time approach. Table 4.3 shows the number of LP and QP problems needed to be solved in Case 1 and 2. It can be seen that, for the example presented here, the total number of LP and QP problems generated by the approach presented in this paper is about 30% greater than the optimal minimal time approach presented in (Scokaert and Rawlings, 1998*b*). It should be noted that the number of constraints in the QP and LP problems generated by the two approaches are different. At a given sample, the number of constraints in the LP problems generated by our approach are less than or equal to the number of constraints generated by the optimal minimal time approach, since only a subset of the state constraints is present when computing κ^k in our priority handling approach, while all state constraints are present when computing κ in the optimal minimal time approach.

Table 1. Computational load for the simulation examples Case 1 and Case 2.

	# LP problems	# QP problems
Case 1:	16	22
Case 2:	13	16

5. DISCUSSION/CONCLUSION

All practical MPC implementations should have a means to recover from infeasibility, and this paper contain an algorithm which transforms an infeasible hard constrained MPC optimization problem into a feasible one by relaxing those hard state constraints which do not affect the stability of the controlled process. This is done taking into account priorities among the state constraints extending the work (Scokaert and Rawlings, 1998*b*), where priorities are not handled. This assignment is an intuitive and natural means to state objectives on the controlled process, and as shown in this paper, MPC is a suitable framework to impose such objectives. When minimizing the violation of a given constraint, the violations of the higher prioritized constraints are not affected.

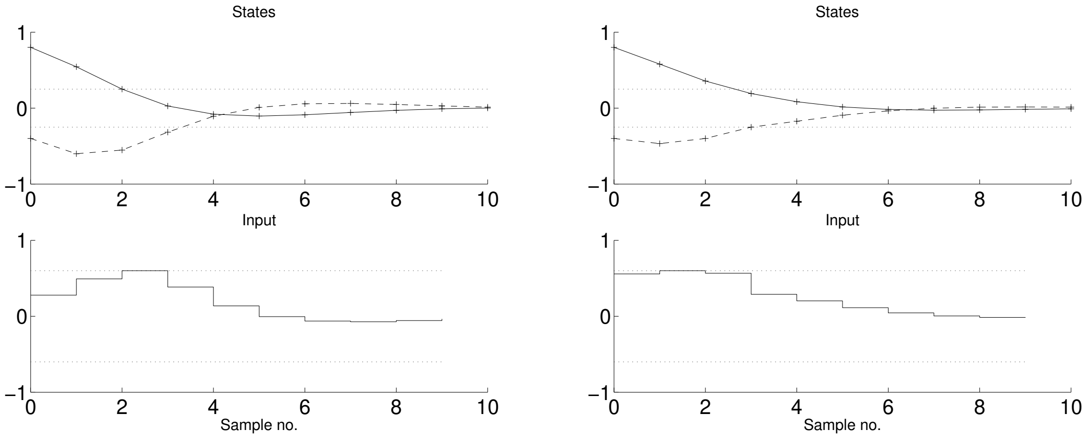

Fig. 2. The left (right) part of the figure shows the simulation results from Case 1 (Case 2). In the upper part of both plots, the solid line shows $x_{1,t}$ and dashed line shows $x_{2,t}$. Dotted lines are the constraint limits.

If some of the input constraint are desirables rather than physical constraints, the infeasibility handling algorithm should also take these constraints into account. Extending the proposed algorithm to include input constraints is trivial.

The example shows how the violations of a constraint are minimized upon at the expense of larger violations of the lower prioritized constraint. In the example, the optimal minimal time approach (Scokaert and Rawlings, 1998*b*) is used to minimize the constraint violations of the constraints which have the same priority. Other approaches may also be used. Consider, for example, a process where large sizes of constraint violations are very expensive. In some cases (such as in non-minimum-phase processes) it will then often be more cost efficient to allow for longer duration of violations at the benefit of smaller sizes of violations. Anyway, as mentioned earlier, obtaining Pareto optimal operation in "the space of duration of violation and size of violation" should always be the goal.

In the example, the number of optimization problems needed to be solved when our approach is used, is about 30% larger than the number of optimization problems needed to be solved when the optimal minimal time approach is used. In other examples, the difference in computational load will probably be different. Whether the computational load required by our approach is acceptable in practise is dependent on the process dynamics and the sampling period of the given process in addition to the computational capacity of the computer where the MPC is installed.

6. REFERENCES

Garcia, C. E. and A. M. Morshedi (1986). Quadratic Programming Solution of Dynamic Matrix Control (QDMC). *Chemical Engineering Communications* **46**, 73–87.

Meadowcroft, T. A., G. Stephanopoulos and C. Brosilow (1992). The Modular Multivariable Controller. 1. Steady State Properties. *AIChE Journal* **38**(8), 1254–78.

Qin, S. J. and T. A. Badgwell (1996). An Overview of Industrial Model Predictive Control Technology. In: *Proceedings of Chemical Process Control - CPC V*. AIChE. Tahoe City, CA.

Rawlings, J B. and K R. Muske (1993). The Stability of Constrained Receding Horizon Control. *IEEE Transactions on Automatic Control* **38**(10), 1512–16.

Rawlings, J. B., E. S. Meadows and K. R. Muske (1994). Nonlinear Model Predictive Control: A Tutorial and Survey. In: *Preprints IFAC Symposium ADCHEM*. Kyoto, Japan.

Scokaert, P. O. M. and J. B. Rawlings (1998*a*). Constrained linear quadratic regulation. *IEEE Trans. Auto. Cont* **43**(8), 1163–1169.

Scokaert, P. O. M. and J. B. Rawlings (1998*b*). Feasibility issues in model predictive control. Submitted for publication in AIChE J, August 1998.

Tyler, M. L. and M. Morari (1997). Propositional Logic in Control and Monitoring Problems. In: *Proceedings of European Control Conference '97*. European Union Control Asssociation. Bruxelles, Belgium. pp. 623–628.

Zheng, A. and M. Morari (1995). Stability of Model Predictive Control with Mixed Constraints. *IEEE Transactions on Automatic Control* **40**(10), 1818–23.