IMPROVED PERFORMANCE OF ROBUST MPC WITH FEEDBACK MODEL UNCERTAINTY

A. L. Warren and T. E. Marlin

Department of Chemical Engineering, McMaster University, Hamilton, ON L8S 4L7 Canada

Abstract: Robust model-predictive controllers use an estimate of model uncertainty in the on-line controller calculation and can be overly conservative for some uncertainty descriptions. This paper discusses the various causes of conservative control with particular emphasis given to the concept of 'closed-loop' probabilistic predictions. A multi-input-multi-output MPC is proposed in which an off-line, non-convex calculation is used to characterize the closed-loop uncertainty a priori. This uncertainty information is incorporated into a convex, quadratic program resulting in a MPC formulation that can be efficiently solved on-line. A distillation column case study demonstrates the benefits of the proposed robust MPC.

Keywords: Robust control, Control algorithms, Closed-loop, Convex optimization.

1. INTRODUCTION

Model-predictive control (MPC) systems have found widespread success in the process industries. The vast majority of these controllers rely upon nominal models, i.e. model uncertainty is not explicitly considered in the on-line controller calculation. Extensive simulation studies and tuning are often required to ensure that these nominal-MPC systems are appropriately robust (Qin and Badgwell, 1996).

Since this situation is not desirable, techniques to create robust MPC systems have been investigated since the late 1980's (see (Badgwell, 1997) for a review). Many of the initial robust MPC systems, such as min-max MPC (Zheng and Morari, 1993), achieved robust stability at the expense of dynamic performance. There are several causes of overly conservative control in robust MPC:

1) *Min-max control strategy* - Min-max controllers are inherently conservative, because they optimize the performance for only the worst-case plant/model mismatch (Bemporad and Morari, 1999).

2) *Time-varying descriptions of process uncertainty* – Several robust MPC systems assume that the process is time-varying (Zheng and Morari, 1993). However, in the process industries many of the processes can be assumed to be time-invariant *within the prediction horizon.* A time-varying description will lead to control that is often too conservative if the actual process is time-invariant.

3) *Open-loop predictions of future system behavior* – An open-loop prediction is one in which the effect of future controller actions is not modeled. An open-loop prediction often overestimates the uncertainty in future process outputs because it does not consider that future controller actions that will respond to plant/model mismatch. This over-estimation of output uncertainty leads to conservative control when the system is near constraints (Mayne, 2000; Kothare et al., 1996).

The controller proposed in this paper addresses these issues by basing the MPC on a closed-loop, time-invariant model of future system behavior. The conservativeness inherent to min-max control is avoided by maintaining the nominal value of the process output near its setpoint while using probabilistic models to avoid output-constraint violations. In addition, the proposed controller uses engineering knowledge of the structure of the process uncertainty to avoid overly conservative uncertainty descriptions. As will be shown in case study, the resulting MPC is robust with respect to output-constraints while avoiding excess conservativeness.

In order to remain computationally feasible for on-line use, the proposed controller is implemented in two stages. In the first stage, the effect of plant/model mismatch on system behavior is captured in off-line studies involving non-convex optimizations. In the second stage, an on-line, convex quadratic program uses the results calculated off-line to predict and to optimize the behavior of the uncertain, closed-loop system. The proposed controller is intended for processes well modeled by multi-input-multi-output (MIMO) linear, time-invariant (LTI) models with no input-constraints.

The remainder of the paper is organized as follows. In Section 2, the rationale behind the various characteristics of the proposed controller will be discussed. This section will outline the derivation of the proposed MPC. Section 3 discusses a method for using Principal Component Analysis (PCA) to improve the uncertainty description of the closed-loop system. Finally, the performance of this new MPC system is explored via a distillation column case study in Section 4.

2. ROBUST MPC UNDER CLOSED-LOOP UNCERTAINTY

2.1 Open-loop vs. Closed-loop Prediction

In unconstrained model-predictive control, the following optimization is solved at each controller execution (Garcia and Morshedi, 1986).

$$\min_{\Delta u} \left\{ (y - y_{SP})^T W (y - y_{SP}) + \Delta u^T Q \Delta u \right\} \qquad (1)$$

$$s.t. \qquad y_i = f(\Delta u, \hat{N}) \qquad \forall i = 1 \ldots n \qquad (1a)$$

Here $y, y_{sp} \in \Re^n$, $\Delta u \in \Re^m$, $W \in \Re^{n \times n}$ and $Q \in \Re^{m \times m}$. The process setpoint is represented by y_{sp}. The matrices, W and Q, are positive definite matrices, typically with the tuning parameters, w and q, on their respective diagonals. These tuning parameters are chosen to achieve the desired compromise between dynamic performance and robustness. Equation (1a) represents a deterministic model of the process with $\hat{N}$ a vector of the predicted value of the process disturbances. In this paper, a linear step-weight model is used and the process is assumed to be open-loop stable or a pure integrator.

The result of this optimization is a vector of input moves, Δu, of which only the first is implemented. At the next controller execution, an updated estimate of the unmeasured disturbance, $\hat{N}$, is calculated, the output prediction is updated, and the procedure begins again.

In an open-loop prediction of uncertainty, the entire vector of Δu is assumed to be known in the prediction of future output uncertainty. This is not an accurate description of a closed-loop, probabilistic system. Through the controller, uncertainty in the future outputs leads to uncertainty in future inputs as the future control actions react to plant/model mismatch. Because open-loop predictions neglect this characteristic of a closed-loop system, such predictions often overestimate the uncertainty in future process outputs and lead to robust MPC that are overly conservative.

In order to perform the required closed-loop prediction, a robust MPC needs a model of the process *and* a model of the future controller actions. In general, the structure of the future control law need not be specified. In this case, the robust MPC problem becomes a special case of the dynamic programming problem. (See Rawlings (1994) for a complete discussion of the relationship between robust MPC and dynamic programming.)

In this paper, the computational issues associated with dynamic programming problem are avoided by assuming that the future control actions are well modeled by the MPC shown in equation (1).

2.2 Overview of Control Strategy

Figure 1 illustrates the general control scheme proposed in this paper.

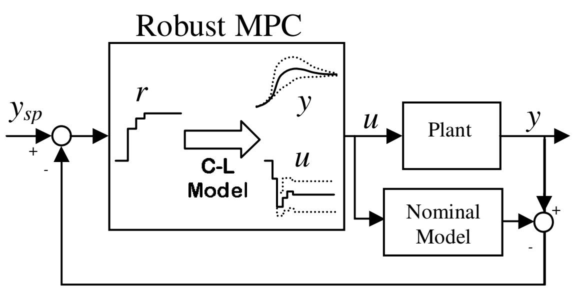

Fig. 1. Conceptual Design for Robust MPC

The controller block depicts the MPC using a closed-loop model of the system to predict the future expected value and upper and lower uncertainty bounds for the inputs, u, and outputs, y. These values are determined by an internal reference trajectory, r. The robust MPC does not directly calculate a vector of input moves as is done in nominal MPC. Instead, it calculates the vector, r. This internal reference trajectory is analogous to a setpoint and by changing this value the robust MPC predicts how a probabilistic closed-loop system will behave in response to a setpoint move. The internal reference trajectory is not a true system setpoint, but represents the desired movement in the future closed-loop system. The term internal refers to the fact that it is a variable used internally by the controller.

The proposed MPC will be implemented as an optimization of the following form.

$$\min_{r, y, \Delta u, \bar{y}, \underline{y}, \bar{u}, \underline{u}} \left\{ (y - y_{SP})^T W (y - y_{SP}) + \Delta u^T Q \Delta u \right\} \qquad (2)$$

$$s.t. \quad y = \left[A_y^{cl} \right]_{nominal} r, \quad \Delta u = \left[A_u^{cl} \right]_{nominal} r \qquad (2a)$$

$$\bar{y} \geq A_y^{cl}(\delta)r, \quad \underline{y} \leq A_y^{cl}(\delta)r, \quad \forall \delta \in \Delta \tag{2b}$$

$$\bar{u} \geq A_u^{cl}(\delta)r, \quad \underline{u} \leq A_u^{cl}(\delta)r, \quad \forall \delta \in \Delta \tag{2c}$$

$$y_{\min} \leq \bar{y}, y, \underline{y} \leq y_{\max} \tag{2d}$$

$$\delta \in \Delta \tag{2e}$$

Here $A_y^{cl}(\delta)$ and $A_u^{cl}(\delta)$ represent the closed-loop models of the system that relate r to y and u. These matrices are functions of the model-mismatch, δ. The nominal inputs and outputs as predicted by a closed-loop model with no plant-model mismatch (i.e. δ=0) are calculated in equation (2a). The upper and lower uncertainty bounds of y and u are represented by the vectors $\bar{y}$, $\underline{y}$ and by $\bar{u}$, $\underline{u}$, respectively. Equations (2b) and (2c) force these values to represent the uncertainty bounds for the worst-case mismatch, assuming $\delta \in \Delta$ where Δ represents the uncertainty set of δ. Section 3 will discuss how this uncertainty set is defined. Equation (2d) ensures that the nominal and uncertainty bounds for y do not violate the desired output constraints. The proposed controller assumes no input-constraints.

The rest of this section will discuss how the various aspects of this control strategy are implemented.

2.3 Closed-loop Predictions

The MPC shown in equation (1) is a natural choice for the model of future controller actions. The Karush-Kuhn-Tucker (KKT) conditions for this unconstrained MPC are linear and can be written as:

$$\left(A^T W^T A + Q^T\right)\Delta u - A^T W^T \left(y_{SP} - \hat{N}\right) = 0 \tag{3}$$

Here the matrix, A, represents a step-weight model of the plant. The linear process model is given by:

$$y = A\Delta u + \hat{N} \tag{4}$$

To create a closed-loop model, equations (3) and (4) are written for each time step within the prediction horizon. This linear set of equations can be combined to derive the closed-loop models, A_y^{cl} and A_u^{cl}, shown in equations (2a) through (2c).

Conceptually, these closed-loop models could be used to determine the predicted uncertainty limits of the future process inputs and outputs (equation (2b) and (2c)). For example, the upper bound in the uncertainty of y for a given r vector could be calculated as:

$$\max_{\delta} \bar{y}_k = 1_k A_y^{cl}(\delta)r \qquad \forall k = 1..n$$
$$s.t. \ \delta \in \Delta \tag{5}$$

Here 1_k is a matrix that selects the k^{th} element in the vector $\bar{y}$. This maximization would find the amount of model-mismatch, δ, that results in the largest possible y at a given time period, k, in the future.

If this optimization could be calculated on-line, the robust MPC could use the calculated uncertainty limit to determine how best to maintain the system at set point while avoiding output-constraints. However, the situation is complicated by the fact that $A_y^{cl}(\delta)$ and $A_u^{cl}(\delta)$ are highly non-linear functions of the amount of plant/model mismatch, δ. Therefore, the optimization shown in equation (5) is non-convex and impractical for on-line implementation. As will be described in the next section, this situation can be avoided if this non-convex minimization is performed off-line a priori.

2.4 Off-line Optimization

The goal of the non-convex optimization is to determine the relationship between the predicted r and the closed-loop uncertainty limits in y and u. These relationships must be summarized so that an on-line, *convex* optimization can make decisions based on this information.

Given an estimate of the model uncertainty, the off-line optimization solves the non-convex optimization problems similar to the one shown in equation (5) and uses the resulting δ to calculate the 'worst-case' $A_y^{cl}(\delta)$ and $A_u^{cl}(\delta)$. Since local optima may be found, several starting points must be used and any results must be checked against Monte-Carlo simulations.

For a MIMO system, the effect of a given δ on future system uncertainty is a function of the direction of r and the directionality of the process. Therefore, the 'worst-case' $A_y^{cl}(\delta)$ and $A_u^{cl}(\delta)$ can be different for different r-directions. Since an infinite number of r-directions exist for any MIMO system, the proposed method uses a representative sampling to estimate this set. For example, for the 2x2 system considered in this paper, 60-different r-directions were considered, each six 'degrees' from another, if one visualizes the set of all possible r-directions as a unit-circle in r_1/r_2 space.

2.5 On-Line Optimization

Naturally, the desired direction of the internal reference trajectory, r, is not known beforehand. Therefore, the on-line optimization must be able to determine the 'worst-case' $A_y^{cl}(\delta)$ and $A_u^{cl}(\delta)$ for any possible r-direction.

The information given in the sampled $A_y^{cl}(\delta)$ and $A_u^{cl}(\delta)$ matrices can be included in the constraints of a convex optimization using the following technique. Assuming a single step change in r at time k=0, the minimization shown in equation (6) can be used to find the largest uncertainty bound for y at given time, regardless of the direction of r.

$$\min_{\bar{y}} \bar{y} \tag{6}$$

$$\begin{aligned} \bar{y} &\geq A_y^{cl}(\delta)_1\, r(0) \\ \bar{y} &\geq A_y^{cl}(\delta)_2\, r(0) \\ &\vdots \\ \bar{y} &\geq A_y^{cl}(\delta)_m\, r(0) \end{aligned} \tag{6a}$$

Here $A_y^{cl}(\delta)_m$ relates r to the largest uncertainty bound of y for a given r-direction. The constraints given in (6a) account for m different directions of r. Each direction could result in a different $A_y^{cl}(\delta)_m$. For a system with 60 different sampled r-directions, m could be as large as 60. However, this is usually not the case, because the $A_y^{cl}(\delta)_m$ are identical for many r-directions. For example, it is very unlikely that the worst-case plant/model mismatch will be different for a change in r of [0 1] and a change of [0.03 0.99]. In case study discussed in Section 4, fewer than 20 different $A_y^{cl}(\delta)_m$ captured the uncertainty limits for all of the tested r-directions.

Within the prediction horizon, the desired direction of r may change several times. Equation (6) can be used to calculate the upper uncertainty bound on y at a given time for a single change in r. In order to calculate the uncertainty bounds for y for a sequence of r-moves, this type of equation must be repeated for each change in the direction of r and the resulting partial uncertainty bounds must be added to give the actual uncertainty bounds. For a system with an input and output horizon of two, these equations would have the following form:

$$\begin{aligned} \bar{y}_0 &\geq A_y^{cl}(\delta)_1\, r(0) \\ &\vdots \\ \bar{y}_0 &\geq A_y^{cl}(\delta)_m\, r(0) \\ \bar{y}_1 &\geq A_y^{cl}(\delta)_1\, r(1) \\ &\vdots \\ \bar{y}_1 &\geq A_y^{cl}(\delta)_m\, r(1) \\ \bar{y} &\geq \bar{y}_0 + \bar{y}_1 \end{aligned} \tag{7}$$

Here r(0) represents the change in r at time, t=0, and $\bar{y}_0$ is a vector representing the upper uncertainty bound for y due to the change in r(0). The true upper limit on the uncertainty of y must be greater than the sum of $\bar{y}_0$ and $\bar{y}_1$ as is shown in the final inequality constraint.

With these linear inequalities, equation (2) can be rewritten as a convex quadratic program.

$$\min_{\substack{r,y,\Delta u,\bar{y},\bar{u},\underline{y},\underline{u} \\ \bar{y}_1..\bar{y}_n,\underline{y}_1..\underline{y}_n \\ \bar{u}_1..\bar{u}_m,\underline{u}_1..\underline{u}_m}} \left\{ (y - y_{SP})^T W (y - y_{SP}) + \Delta u^T Q \Delta u \right\} \tag{8}$$

s.t. Equation (7) and similar equations for $\underline{y}, \bar{u}$, and $\underline{u}$ (8a)

$$y = \left\lfloor A_y^{cl} \right\rfloor_{nominal} r, \quad \Delta u = \left\lfloor A_u^{cl} \right\rfloor_{nominal} r \tag{8b}$$

$$y_{\min} \leq \bar{y}, y, \underline{y} \leq y_{\max} \tag{8c}$$

All output constraints are 'softened' to avoid feasibility and stability issues (Zafiriou, 1990). This quadratic program is solved at each controller execution. The first input move is then applied in a rolling-horizon fashion.

Implementation issues: This optimization is convex, but the size of the problem can create some computational issues. For the 2x2 case study discussed in Section 4, the problem has 1134 decision variables and 3317 inequality constraints.

The size of this QP poses a problem for active set method such as the quadprog program found in Matlab (Coleman et al., 1999). Fortunately, recent progress in the field of interior-point (IP) methods provides a solution. While the theoretical worst-case number of iterations for IP methods is bounded by $O(n^3)$ (Lobo et al., 1998), these methods have been shown to be much more efficient in practice (Andersen and Ye, 1999). Using Andersen's MOSEK interior-point algorithm, the average solution time for our quadratic program averaged only 1.35 seconds on a Pentium IV, 1.8 GHz.

As the number of inputs, outputs, and length of the prediction horizon grows, the set of equations represented by constraint (8a) will reach a point where even interior-point methods will require excessive computing time. Future work will explore how the dimensionality of this problem can be reduced.

3. CLOSED-LOOP UNCERTIANTY DESCRIPTION

The performance of the robust MPC described above depends strongly on the uncertainty description used by the controller. A poor description of the system uncertainty may lead to conservative control.

For example, consider a typical non-linear, binary distillation column from Marlin (2000). This process can be modeled by the following linear system, where X_D and X_B represent the distillate and bottoms compositions of the light key. These variables are controlled by the reflux rate, F_R, and the amount vaporized in the reboiler, F_V.

$$\begin{bmatrix} X_D \\ X_B \end{bmatrix} = \begin{bmatrix} \dfrac{Kp_{11}e^{-\theta_{11}}}{\tau_{11}s+1} & \dfrac{Kp_{12}e^{-\theta_{12}}}{\tau_{12}s+1} \\ \dfrac{Kp_{21}e^{-\theta_{21}}}{\tau_{21}s+1} & \dfrac{Kp_{22}e^{-\theta_{22}}}{\tau_{22}s+1} \end{bmatrix} \begin{bmatrix} F_R \\ F_V \end{bmatrix} \qquad (9)$$

Assume that this linear model is used to represent a non-linear distillation column in which the feed rate, F_f, is unmeasured and not constant. Changes in the feed rate will affect the model parameters and uncertainty in the feed rate leads to uncertainty in these parameters. The feed rate has a nominal value of 10 kmol/min and varies very slowly (with respect to the closed-loop settling time) between 8.5 and 11 kmol/min. Table 1 summarizes the coefficients of the linear model fit at various feed rates with a sampling rate of 2 min^{-1}.

Table 1: Effect of Feed Rate Changes on Model

F_f (kmol/min)	Kp (%/kmol min^{-1})	τ (min)	θ (min)
8.5	$\begin{bmatrix} 0.088 & -0.079 \\ 0.14 & -0.15 \end{bmatrix}$	$\begin{bmatrix} 20.8 & 22.0 \\ 21.6 & 21.2 \end{bmatrix}$	$\begin{bmatrix} 2.60 & 3.76 \\ 3.10 & 2.34 \end{bmatrix}$
10.0	$\begin{bmatrix} 0.075 & -0.067 \\ 0.12 & -0.13 \end{bmatrix}$	$\begin{bmatrix} 17.7 & 18.7 \\ 18.3 & 18.0 \end{bmatrix}$	$\begin{bmatrix} 2.50 & 3.49 \\ 2.97 & 2.33 \end{bmatrix}$
11.5	$\begin{bmatrix} 0.065 & -0.058 \\ 0.10 & -0.11 \end{bmatrix}$	$\begin{bmatrix} 15.3 & 16.1 \\ 15.9 & 15.6 \end{bmatrix}$	$\begin{bmatrix} 2.42 & 3.33 \\ 2.89 & 2.32 \end{bmatrix}$
13.0	$\begin{bmatrix} 0.057 & -0.051 \\ 0.09 & -0.10 \end{bmatrix}$	$\begin{bmatrix} 13.5 & 14.2 \\ 14.0 & 13.7 \end{bmatrix}$	$\begin{bmatrix} 2.35 & 3.18 \\ 2.83 & 2.31 \end{bmatrix}$

One possible uncertainty description for this process is a set of equation such as:

$$\begin{array}{lll} 0.057 \le Kp_{11} \le 0.088, \ldots, -0.15 \le Kp_{22} \le -0.10 & & \\ 13.5 \le \tau_{11} \le 20.8, \quad \ldots & ,14.2 \le \tau_{22} \le 22.0 & (10) \\ 2.35 \le \theta_{11} \le 2.60, \quad \ldots & ,2.31 \le \theta_{22} \le 2.34 & \end{array}$$

However, these box-type uncertainty descriptions are inappropriate. No linear controller will be able to stabilize all of the plants described by equations (10) because the systems do not meet the integral stabilizability test of Grosdidier et al. (1985).

Even if integral stabilizability is not an issue for a given system, the box-type description is unsatisfactory because it ignores the steady-state and dynamic relationships set by the physics of the system. For example, a process uncertainty that affects the process dead-time often also affects the process time-constant and gain. Likewise, there usually exists a relationship between steady-state gains of a MIMO system.

3.1 PCA Uncertainty Description

These structured uncertainty relationships can be captured using the Principal Component Analysis (PCA) technique. PCA is a multivariate statistical method that summarizes the variation within a data set, X, in the fewest possible dimensions, d (Wold, 1987). A score vector, t, a loading vector, p, and a residual matrix, ε, summarize the data as shown in equation (11), where X_{mean} is the column-wise mean of the data.

$$X = X_{mean} + \sum_{i-1}^{d} tp' + \varepsilon \qquad (11)$$

If the information in Table 1 is summarized using PCA where each row of the data matrix represents a different flow rate and each column one of the 12 model coefficients, the majority of the variability can be summarized using a single t-variable. This illustrates the fact that there is one main source of variability within the data set (i.e. the column feed rate.) Using this PCA description, the uncertainty in the process can be summarized as:

$$X = tp' + X_{mean}, \qquad -13 \le t \le 13 \qquad (12)$$

Here p is a 12x1 constant loading vector and the inequality represents a component-wise 95%-confidence interval for t. The uncertainty in this example is summarized in a single score space, but higher dimensional descriptions are possible. In such cases, the inequality constraint shown in equation (12) expands to a multi-dimensional ellipsoid.

This PCA description of uncertainty has several advantages. The dimensionality of the non-convex optimization discussed in section 2.4 is greatly reduced. In addition, the loading and score vectors can be helpful in deciding which sources of uncertainty are important and which can be eliminated from the model.

4. CASE STUDY

The following case study illustrates the ability of the proposed MPC to robustly avoid output-constraint violations while maintaining acceptable dynamic performance. The distillation column discussed above is to be controlled by the proposed robust MPC found in equations (8). The case studies assume that uncertainty is caused by plant/model mismatch only and no disturbances are affecting the plant. This assumption can be relaxed by applying the techniques discussed in Warren (2003).

Figure 2 below shows the performance of the unconstrained system responding to a setpoint change of [1 0] mole percent at time, t=1, from an initial condition of [98 2]. The nominal plant model is given by F_f of 10 kmol/min in Table 1 and the MPC shown in equation (1) is used in the closed-loop model with the following tuning parameters; n=20, m=5, w=[1 1], q=[0.02 0.02].

The thick solid lines in Figure 2 represent the uncertainty limits of the inputs and outputs predicted from time, t=0, using equations similar to equation (7). The dashed lines in Figure 1 represent the closed-loop response of the distillation column at feed rates of 7, 8, 9, 11, 12, 13, and 14 kmol/min. Even though some of these feed flow rates fall outside of the original range for which the robust MPC was designed, the predicted uncertainty bounds are quite accurate. Notice that the closed-loop uncertainty predictions accurately predict that the uncertainty in y will approach zero due to the integral action of the controller and the fact that the closed-loop system is stable.

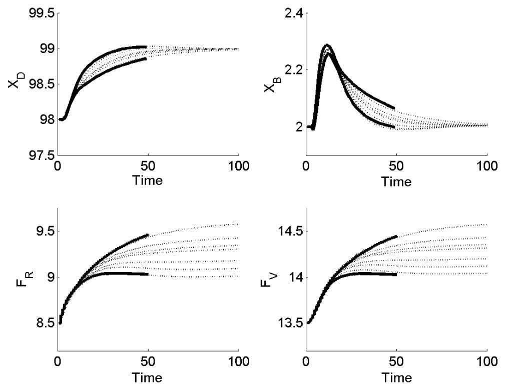

Fig. 2. Closed-Loop Uncertainty Prediction

The proposed robust MPC is able to use these uncertainty predictions to avoid output-constraints. For example, consider the case where the bottoms composition must remain below 2.5 mole percent light key. Figure 3 compares the performance of proposed robust MPC to that of a nominal MPC with softened output-constraints. In this example, the process model used by the controllers is given by F_f of 10 kmol/min in Table 1 while the true process is operating at F_f of 8.5 kmol/min. The robust system successfully avoids the output constraint without becoming overly conservative.

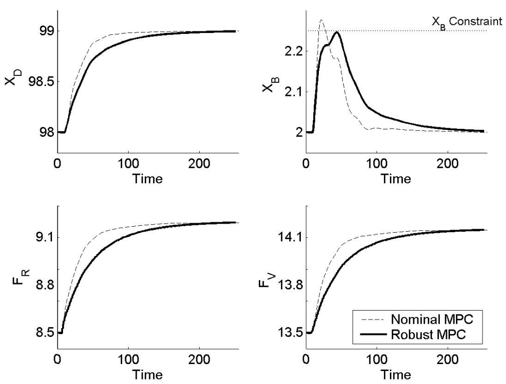

Fig 3. Comparison of Robust and Nominal MPC

5. CONCLUSIONS

This paper has discussed the importance of using an accurate closed-loop description of system uncertainty in robust MPC. A robust MPC system based on a closed-loop system description has been proposed and shown to outperform nominal MPC systems when plant/model mismatch is present.

REFERENCES

Andersen, E. D. and Y. Ye (1999). On a homogeneous algorithm for the monotone complementarity problem, *Mathematical Programming*, **84**(2), pp. 375-399.

Badgwell, T. A. (1997). Robust model predictive control of stable linear systems, *International Journal of Control*, **68**(4), pp. 797-818.

Bemporad, A. and M. Morari (1999). Robust Model Predictive Control: A Survey. *Lecture Notes in Control and Information Sciences,* **245,** Springer-Verlag, pp. 207-226.

Coleman, T., M. A. Branch, and A. Grace (1999). *Matlab Optimization Toolbox: User's Guide*, The Mathworks, Inc., Natick, MA.

Garcia, C. E. and A. M. Morshedi (1986). Quadratic Programming Solution of Dynamic Matrix Control (QDMC), *Chemical Engineering Communications*, **46**, pp. 73-86.

Grosdidier, P., M. Morari, and B. R. Holt (1985). Closed-loop Properties from Steady-State Information, *Industrial and Engineering Chemistry Fundamentals*, **24**, pp. 221-235.

Kothare, M. V., V. Balakrishnan, and M. Morari (1996). Robust Constrained Model Predictive Control using Linear Matrix Inequalities, Automatica, **32**(10), pp. 1361-1379.

Lobo, M. S., L. Vandenberghe, S. Boyd, and H. Lebret (1998). Applications of Second-order Cone Programming, *Linear Algebra & Its Applications*, **284**, pp. 193-228.

Marlin, T. E. (2000). *Process Control: Designing Processes and Control Systems for Dynamic Performance*, Chapter 20, McGraw-Hill, New York, NY.

Mayne, D. (2000). Nonlinear Model Predictive Control: Challenges and Opportunities, *Progress in Systems and Control Theory*, **26**, pp. 23-44.

Qin, S. J. and T. A. Badgwell (1996). An Overview of Industrial Model Predictive Control Technology, Proceedings of Chemical Process Control V, pp. 232-256.

Rawlings, J. B., E. S. Meadows, and K. R. Muske (1994). Nonlinear Model Predictive Control: A Tutorial and Survey, ADCHEM 1994, Kyoto, Japan, pp. 203-214.

Warren, A. L. (2003). Improved Output Constraint-Handling for MPC with Disturbance Uncertainty, *To appear in ACC 2003*, Denver, Colorado.

Wold, S., K. Esbensen, and P. Geladi (1987). Principal Component Analysis, *Chemometrics and Intelligent Laboratory Systems*, **2**, pp. 37-52.

Zafiriou, E. (1990). Robust model predictive control of processes with hard constraints, *Computers and Chemical Engineering*, **14**(4/5), pp. 359-371.

Zheng, A. and M. Morari (1993). Robust control of Constrained Model Predictive Control. Proceedings of American Control Conference, San Francisco, California, pp. 379-383.