CONSTRAINED MPC UNDER CLOSED-LOOP UNCERTAINTY

Adam L. Warren and Thomas E. Marlin*
Department of Chemical Engineering
McMaster University
Hamilton, Ontario Canada L8N 4L7
e-mail: marlint@mcmaster.ca

1. ABSTRACT

The importance of closed-loop uncertainty predictions in robust model-predictive control (MPC) has been discussed by a number of authors in recent years [1-3]. The proposed controllers often rely upon invariant sets and require that input constraints are inactive at the final steady-state [3, 4]. The controller discussed in this paper avoids this limiting assumption while maintaining robust output constraint handling. This paper emphasises the often negative effects of probabilistic input constraints and proposes a method based upon multiple uncertainty regions to deal with these effects. The proposed controller solves a second-order cone program (SOCP) at each execution in order to determine the set of control moves that will optimize the expected performance of the closed-loop system while maintaining the uncertain process outputs and inputs within their allowable bounds. Case studies illustrate the performance of the new controller when plant/model mismatch is present.

2. INTRODUCTION

In recent years, the academic MPC community has proposed a number of MPC systems that explicitly consider uncertainty caused by plant/model mismatch. The most promising of these controllers use a closed-loop prediction of uncertainty [1-7]. Closed-loop uncertainty descriptions differ from open-loop descriptions in that closed-loop descriptions consider the effect of future control actions and, therefore, more accurately predict the future system uncertainty. Robust MPC based upon closed-loop uncertainty predictions have been shown to outperform robust MPC based on open-loop uncertainty [1, 5].

This paper develops a robust model-predictive controller for stable systems with uncertain plant models. The process is assumed to be linear time-invariant (LTI) within the prediction horizon. The controller deals with plant model uncertainty by replacing deterministic constraints, $y_{min} < y < y_{max}$ and $u_{min} < u < u_{max}$, in the MPC formulation with probabilistic constraints of the form:

$$\Pr\{y_{\min} \le y \le y_{\max}\} \ge 1 - \alpha_y \quad (1)$$

$$\Pr\{u_{\min} \le u \le u_{\max}\} \ge 1 - \alpha_u \quad (2)$$

These probabilistic constraints assert that the process output, y, and input, u, remain within their limits with confidence levels of $1-\alpha_y$ and $1-\alpha_u$, respectively.

Many robust MPC proposed in literature [4, 6] use input constraints that are conceptually similar to those shown in equation (2). This type of input constraints can lead to overly conservative control because every probable realization of the process input must lie within the feasible region. Only the input trajectory associated with the worst-case plant/model mismatch is allowed to meet the input constraint. The rest of the possible input trajectories are backed away from the constraint. This 'back-off' does not accurately predict the future behavior of a constrained MPC system and often leads to conservative control.

One of the main contributions of this paper is a method for effectively handling probabilistic input constraints. A method is proposed in which the system uncertainty is split into several uncertainty regions. The uncertainty associated with each region is smaller than the total system uncertainty, allowing each subset to approach the input constraint more quickly. This reduces the 'back-off' caused by the probabilistic input constraints. Case studies illustrate the improved dynamic performance of the multi-region method.

The proposed robust MPC is based on a second-order cone program (SOCP) and is computationally less complex than robust MPC requiring the solution of a semidefinite program (SDP) [4, 7]. See [8] for a comparison of the complexity of interior point algorithms that solve SOCP and SDP.

The rest of the paper is organized as follows. In section 3, a brief review of robust MPC is given. Key concepts include closed-loop prediction of future system uncertainty and the effect of probabilistic input constraints. In section 4, two new robust MPC systems based upon SOCP are presented and explored via continually stirred tank reaction (CSTR) case studies.

3. BACKGROUND

The uncertainty description used by a robust MPC determines much of the dynamic and constraint-handling characteristics of the controller. As discussed below, poor uncertainty descriptions can lead to poor control.

3.1 Time-varying Uncertainty Descriptions

Several robust MPC systems assume that the process is linear time-varying [4, 9]. However, in the process industries many of the processes can be assumed to be

time-invariant *within the prediction horizon*. Because the time-varying uncertainty description includes plants that are very unlikely, a time-varying description is inappropriate in many situations.

3.2 Open-loop Uncertainty Descriptions

In unconstrained model-predictive control, the following optimization is solved at each controller execution [10].

$$\min_{\Delta u} \left\{ (y_{SP} - y)^T W (y_{SP} - y) + \Delta u^T Q \Delta u \right\} \qquad (3)$$

$$s.t. \qquad y_i = f(\Delta u, \breve{N}) \qquad \forall i = 1 \ldots n \qquad (3a)$$

where n is the output prediction horizon, m is the input horizon, and y, y_{sp}, $\breve{N} \in \Re^n$, $\Delta u \in \Re^m$, $W \in \Re^{nxn}$, $Q \in \Re^{mxm}$. The process set point 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 (3a) represents a deterministic model of the process. In this paper, a linear step-weight model is used and the process is assumed to be open-loop stable.

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 disturbance, $\breve{N}$, is calculated, the output prediction is updated, and the procedure is repeated.

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. Uncertainty in the future outputs leads to uncertainty in future inputs as the future control actions compensate for disturbances or plant/model mismatch. Because open-loop predictions neglect this characteristic of closed-loop systems, such predictions often overestimate the uncertainty in future process outputs. Furthermore, open-loop predictions provide no estimate of future input uncertainty. This poor uncertainty model can lead to overly conservative control when the system is operated near input or output constraints [1-5].

3.3 Probabilistic Input Constraints

As discussed in the introduction, the use of probabilistic input constraints can lead to conservative control because only the input trajectory associated with the worst-case plant/model mismatch is allowed to meet the input constraint. Section 4.6 discussed how this effect can be mitigated through the use of multiple uncertainty regions.

4. CONSTRAINED MPC UNDER CLOSED-LOOP UNCERTAINTY

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

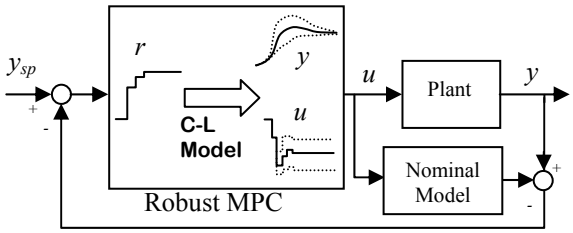

Figure 1: Conceptual Design for Robust MPC

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 limits for the inputs, u, and outputs, y. The robust MPC does not directly calculate a vector of input moves as is done in nominal MPC. Instead, it calculates an internal reference trajectory, r. This internal reference trajectory represents the desired movement in the future closed-loop system and is related to the predictive reference filter [11].

4.1 Closed-loop Uncertainty Prediction

In order to create the required closed-loop prediction, a model of the process *and* a model of the future controller actions are needed. In this paper, the future control actions are modeled by the MPC shown in equation (3). The Karush-Kuhn-Tucker (KKT) conditions for this unconstrained MPC are linear and can be written as:

$$\left(A_M{}^T W^T A_M + Q^T \right) \Delta u - A_M{}^T W^T \left(r - \hat{N} \right) = 0 \qquad (4)$$

where $A_M \in \Re^{nxm}$ and represents the nominal step-weight model of the plant. The vector, r, represents the internal reference trajectory and the tuning parameters, W and Q, have been chosen to provide acceptable dynamic performance and robustness.

The linear process is given by the following equation where A_P represents the true process and is not necessarily equal to A_M.

$$y = A_P \Delta u + \breve{N} \qquad (5)$$

To create a closed-loop model, equations (4) and (5) are written for each time step within the prediction horizon. This linear set of equations can be combined to derive the two closed-loop step-weight models, A_y^{cl} and A_u^{cl} shown in equation (6). These matrices represent the closed-loop relationships between the internal reference trajectory and the process inputs and outputs, such that:

$$\begin{aligned} y &= A_y^{cl} r + \breve{N} \\ u &= A_u^{cl} r \end{aligned} \qquad (6)$$

where y, r, $\breve{N} \in \Re^n$, $u \in \Re^m$, $A_y^{cl} \in \Re^{nxn}$, and $A_u^{cl} \in \Re^{mxn}$.

4.2 Propagation of Closed-loop Uncertainty

Model uncertainty can be characterized using an ellipsoidal bound on the parameters. If $a_{P,i}$ represents a possible

realization of the i^{th} row of the step-weight matrix, A_P, the following equations hold.

$$a_{P,i} \in \left\{ a_{P,i} : \left(a_{P,i} - \bar{a}_{P,i}\right)\Psi_{P,i}\left(a_{P,i} - \bar{a}_{P,i}\right) \le 1 \right\}, \quad \forall i = 1 \ldots n \tag{7}$$

Where $\bar{a}_{P,i}$ is the center of the ellipsoid and $\Psi_{P,i}$ is a positive definite matrix that determines the size and orientation of the ellipsoid.

An equivalent, statistical interpretation of the system is also possible [13]. The vector $\bar{a}_{P,i}$ represents the expected value of the i^{th} row of A_P and the variance-covariance of the i^{th} row is denoted as $V(a_{P,i})$. The relationship between $\Psi_{P,i}$ and $V(a_{P,i})$ is given by equation (8).

$$\Psi_{P,i} = \frac{1}{\beta_\alpha^{\,2}} V^{-1}\left(a_{P,i}\right), \quad \forall i = 1 \ldots n \tag{8}$$

where $\beta_\alpha = \Phi^{-1}(\alpha)$, the value of the inverse cumulative distribution function of the standard normal distribution evaluated at confidence or probability level α.

In order to create a closed-loop uncertainty description, the model uncertainty described in equation (7) must be mapped to the uncertainty associated with the closed-loop models, A_y^{cl} and A_u^{cl} shown in equation (6). For a linear time-invariant (LTI) plant with a fixed controller, this mapping can be accomplished during the controller design phase with an off-line sampling technique. For any plant sampled from the set defined by equation (7), a single realization of A_y^{cl} and A_u^{cl} can be directly calculated by combining equations (4) and (5) for every sample time within the prediction horizon. The uncertainty in A_y^{cl} and A_u^{cl} can then be characterized by determining the variance-covariance and expected values of the sampled realizations.

4.3 MPC under Closed-loop Uncertainty

Once these closed-loop statistics have been estimated, an robust MPC can be derived. The proposed MPC optimizes the performance of the nominal plant while ensuring that the all probable trajectories of y and u remain feasible. This is accomplished by using nominal values of y and Δu in the objective function of the optimization and adding closed-loop probabilistic constraints of the form seen in equations (1) and (2).

The general form of the proposed robust MPC is shown in equation (9). To simplify the presentation, the slack variables required to avoid infeasibility and stability issues are omitted [12].

$$\min_r \left\{ \left(y_{SP} - \bar{y}\right)^T W \left(y_{SP} - \bar{y}\right) + \Delta \bar{u}^T Q \Delta \bar{u} \right\} \tag{9}$$

$$s.t. \quad \bar{y} = \bar{A}_y^{cl} r + \breve{N} \tag{9a}$$

$$\bar{u} = \bar{A}_u^{cl} r \tag{9b}$$

$$\Delta \bar{u}_i = \bar{u}_i - \bar{u}_{i-1}, \quad \forall i = 1 \ldots n \tag{9c}$$

$$\Pr\left\{ y_{\min} \le y_i \le y_{\max} \right\} \ge 1 - \alpha_y, \quad \forall i = 1 \ldots n \tag{9d}$$

$$\Pr\left\{ u_{\min} \le u_i \le u_{\max} \right\} \ge 1 - \alpha_u, \quad \forall i = 1 \ldots n \tag{9e}$$

where the various over-bars denote the expected values of the associated variables.

The proposed MPC can be cast as a second-order cone program (SOCP) if equations (9d-e) are reformulated as conic constraints [8, 13]. Equations (9d-e) can be written as

$$\Pr\left\{ a_{y,i}^{cl} r + \breve{n}_i \le y_{\max} \right\} \ge 1 - \frac{\alpha_y}{2}, \quad \forall i = 1 \ldots n \tag{10}$$

$$\Pr\left\{ y_{\min} \le a_{y,i}^{cl} r + \breve{n}_i \right\} \ge 1 - \frac{\alpha_y}{2}, \quad \forall i = 1 \ldots n \tag{11}$$

$$\Pr\left\{ a_{u,i}^{cl} r \le u_{\max} \right\} \ge 1 - \frac{\alpha_u}{2}, \quad \forall i = 1 \ldots n \tag{12}$$

$$\Pr\left\{ u_{\min} \le a_{u,i}^{cl} r \right\} \ge 1 - \frac{\alpha_u}{2}, \quad \forall i = 1 \ldots n \tag{13}$$

where $a^{cl}{}_{x\,i}$ represents the i^{th} row of associated matrix.

The expected value and variance-covariance of $a^{cl}{}_{y,i}$ and $a^{cl}{}_{u,i}$ are denoted $\bar{a}^{cl}{}_{y,i}$, $\bar{a}^{cl}{}_{u,i}$, $V(a^{cl}{}_{y,i})$, and $V(a^{cl}{}_{u,i})$, respectively. These values are calculated using the off-line sampling technique discussed in section 4.2. From these data, the expected value and variance of equations (10)-(13) can be calculated.

$$E\left(a_{y,i}^{cl} r\right) = \bar{a}_{y,i}^{cl} r, \; V\left(a_{y,i}^{cl} r\right) = r^T V\left(a_{y,i}^{cl}\right) r, \quad \forall i = 1 \ldots n \tag{14}$$

$$E\left(a_{u,i}^{cl} r\right) = \bar{a}_{u,i}^{cl} r, \; V\left(a_{u,i}^{cl} r\right) = r^T V\left(a_{u,i}^{cl}\right) r, \quad \forall i = 1 \ldots n \tag{15}$$

Equations (10)-(13) can then be recast in their standard normal form. For example, equation (10) becomes

$$\Pr\left\{ \frac{\left(a_{y,i}^{cl} r + \breve{n}_i\right) - \left(\bar{a}_{y,i}^{cl} r + \breve{n}_i\right)}{\sqrt{r^T V\left(a_{y,i}^{cl}\right) r}} \le \frac{y_{\max} - \left(\bar{a}_{y,i}^{cl} r + \breve{n}_i\right)}{\sqrt{r^T V\left(a_{y,i}^{cl}\right) r}} \right\} \ge 1 - \frac{\alpha_y}{2}, \quad \forall i = 1 \ldots n \tag{16}$$

Therefore, the probabilistic constraint can be rewritten as

$$\Phi\left(\frac{y_{\max} - \left(\bar{a}_{y,i}^{cl} r + \breve{n}_i\right)}{\sqrt{r^T V\left(a_{y,i}^{cl}\right) r}} \right) \ge 1 - \frac{\alpha_y}{2}, \quad \forall i = 1 \ldots n \tag{17}$$

And equation (10) becomes a second-order conic constraint.

$$\left(\bar{a}_{y,i}^{cl} r + \breve{n}_i\right) + \Phi_{\alpha_y/2}^{-1} \left\| V^{1/2}\left(a_{y,i}^{cl}\right) r \right\|_2 \le y_{max}, \quad \forall i = 1 \ldots n \tag{18}$$

Similar transformations are performed for the remaining probabilistic constraints and equation (9) is reformulated as a convex, SOCP that can be efficiently solved using interior-point algorithms such as Sedumi [14].

4.4 Robust Output Constraints

Consider the following 1st-order, isothermal CSTR where modeling error is present. The true plant, plant model, and variance-covariance of the gain, K_P, time constant, τ, and deadtime, θ, are shown in Figure 2. The model coefficients differ from the true plant coefficients by approximately two standard deviations and lie within the ellipsoidal uncertainty description used by the controller.

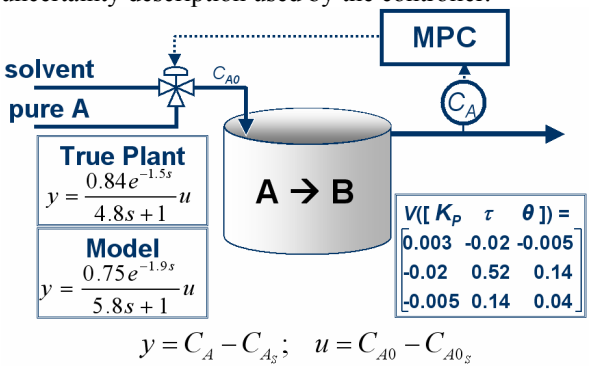

Figure 2: Uncertain Isothermal CSTR System I

This process is controlled near an output constraint of $y \leq 0.1$. No input constraints are present. Figure 3 compares the performance of the robust MPC proposed in this section with a MPC that uses only a nominal model of the process. Both controllers are aggressively tuned with tuning parameters $q = 1.0$ and $w = 1\text{x}10^{-4}$.

Figure 3 shows the reaction of the system to a set point change of -10 away from the constraint and +10 towards the constraint. Notice that the robust and nominal MPC behave almost identically when output constraints are inactive (i.e. set point change of -10). However, when the constraints are encountered, the nominal MPC violates the output constraint. The robust MPC uses an explicit, closed-loop model of uncertainty to observe this y-constraint without becoming overly conservative. Comparable results could be obtained with similar formulations such as [3]. However, such formulations are restricted to cases where input constraints are inactive at stead-state. As discussed in the next section, this restriction does not apply to the proposed MPC.

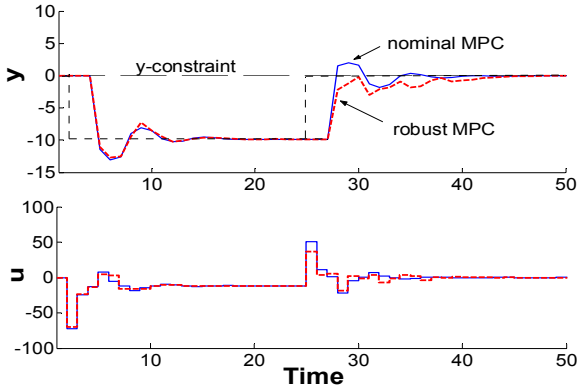

Figure 3: Comparison of Nominal and Robust MPC

4.5 Robust Input Constraints

As discussed in the introduction, probabilistic input constraints can lead to conservative control when the system is operated near input constraints. This situation is illustrated by the following example, in which the CSTR shown in Figure 4 is controlled by a MPC with $q = 1.0$ and $w = 0.4$.

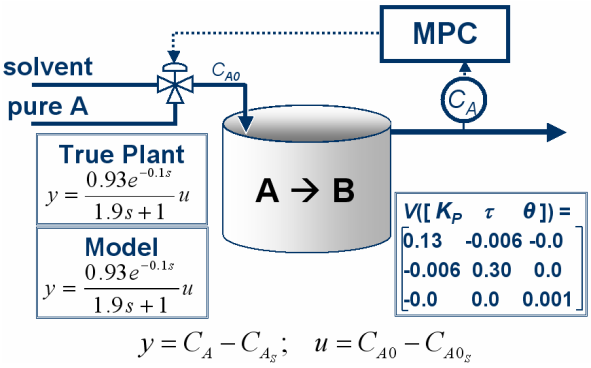

Figure 4: Uncertainty Isothermal CSTR System II

A closed-loop Monte-Carlo simulation of this LTI process is shown in Figure 5. This figure represents the unit set point response of fifty plant realizations sampled from the uncertainty region defined in Figure 4. Notice the relatively large distribution of u. This distribution is modeled by the variance-covariance of the closed-loop model, $A^{cl}{}_u$, as discussed in section 4.2.

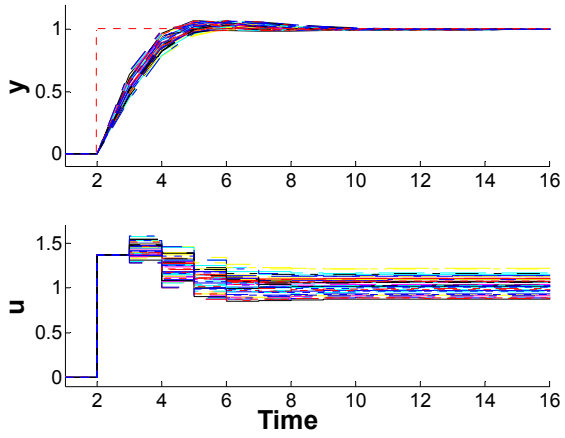

Figure 5: Closed-loop Monte-Carlo of Uncertain CSTR

With this amount of uncertainty present in the closed-loop system, the robust MPC must be conservative when operating near input constraints in order to avoid violation of the probabilistic input constraints seen in equation (9e).

Figure 6 show the reaction of the proposed MPC to a set point change of 1.0 when an input constraint of $u \leq 0.8$ is present.

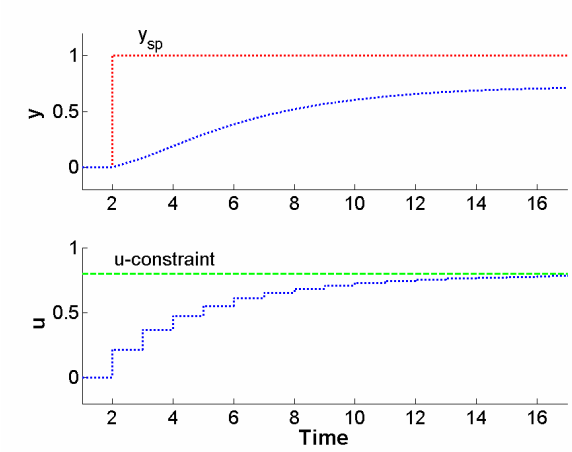

Figure 6: Nominal and Robust MPC with Probabilistic Input Constraints

Clearly, the robust MPC is more conservative than need be. The probabilistic input constraints do not allowing the controller to quickly saturate the system. Instead, the MPC is forced to make a number of relatively small input moves before the input constraint is met.

4.6 Multiple Uncertainty Regions

Improved robust performance can be achieved by dividing the uncertainty region into a number of subsets. The uncertainty associated with each region is smaller than the total system uncertainty, allowing each subset to approach the input constraint more quickly. This uncertainty description more closely matches the true behaviour of a closed-loop, probabilistic system and robust MPC built upon this approach outperform robust MPC that use a single uncertainty region when processes are operated near input constraints.

For example, consider the process shown in Figure 4. The histogram for the distribution of the process gain, K_P, is shown in Figure 7a. This single uncertainty region can be divided into smaller uncertainty sets. Figure 7b-c shows the distribution of K_P within each subset. Each subset has a different expected value of the process, A_P, and different estimates of the variance-covariance of each row of A_P. Naturally, each region of the plant uncertainty description maps to a different uncertain region with respect to the closed-loop matrices, A_y^{cl} and A_u^{cl}, as discussed in section 4.2.

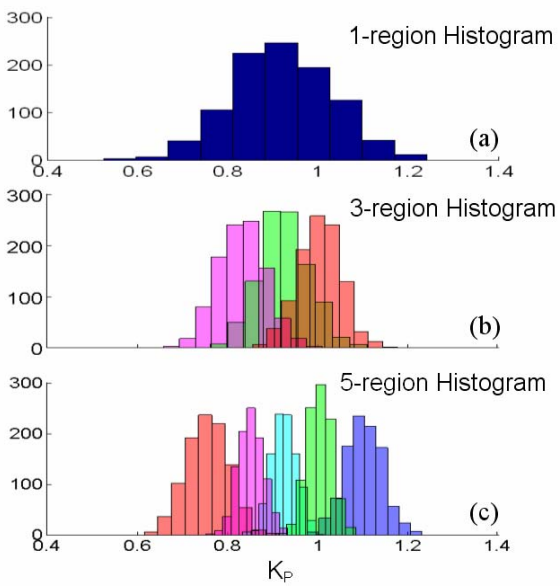

Figure 7: Distribution of Multiple Uncertainty Regions

Once the multiple uncertainty regions have been defined, the robust MPC outlined in section 4.3 can be recast as:

$$\min_{r(1)\ldots r(nc)} \left\{ \sum_{k=1}^{n_c} \left((y_{SP} - \bar{y}_k)^T W_k (y_{SP} - \bar{y}_k) + \Delta \bar{u}_k{}^T Q_k \Delta \bar{u}_k \right) \right\} \quad (19)$$

$$s.t. \qquad \bar{y}_k = \bar{A}_y^{cl} r_k + \breve{N}, \quad \forall k = 1 \ldots n_c \quad (19a)$$

$$\bar{u}_k = \bar{A}_u^{cl} r_k, \quad \forall k = 1 \ldots n_c \quad (19b)$$

$$\Delta \bar{u}_{i,k} = \bar{u}_{i,k} - \bar{u}_{i-1,k}, \quad \forall i = 1 \ldots n, \forall k = 1 \ldots n_c \quad (19c)$$

$$\Pr\{ y_{\min} \le y_{i,k} \le y_{\max} \} \ge \alpha_y, \quad \forall i = 1 \ldots n, \forall k = 1 \ldots n_c \quad (19d)$$

$$\Pr\{ u_{\min} \le u_{i,k} \le u_{\max} \} \ge \alpha_u, \quad \forall i = 1 \ldots n, \forall k = 1 \ldots n_c \quad (19e)$$

$$\Delta \bar{u}_k(1) = \Delta \bar{u}_{k+1}(1), \ldots, \Delta \bar{u}_k(1) = \Delta \bar{u}_{k+n_c}(1) \quad (19f)$$

where the number of regions is denoted n_c. The objective function minimized the weighted average of the predicted nominal trajectories of y and Δu as calculated by equations (19a-c). Equations (19d-e) define the probabilistic constraints for each uncertainty region. As discussed in section 4.3, these equations are reformulated to conic constraints using an off-line sampling technique. Equation (19f) forces the first input move for each of the n_c input trajectories to be equal. It is this first input move that is actually implemented by the controller. Note that this formulation can be viewed as a simplification of the semiinfinite solution MPC solution [13].

4.7 CSTR Case Study – Robust Input Constraints

Consider the CSTR system shown in Figure 4 in which the uncertainty is described by the 1-region, 3-region, and 5-region uncertainty descriptions illustrated in Figure 7. Using equation (19), a MPC system is constructed for each uncertainty description. Figure 8 compares the performance of each of these controllers for a setpoint

change of 1.0 when an input constraint of $u \leq 0.8$ is present. The controller tuning is kept the same for each controller, $q = 1.0$ and $w = 0.4$.

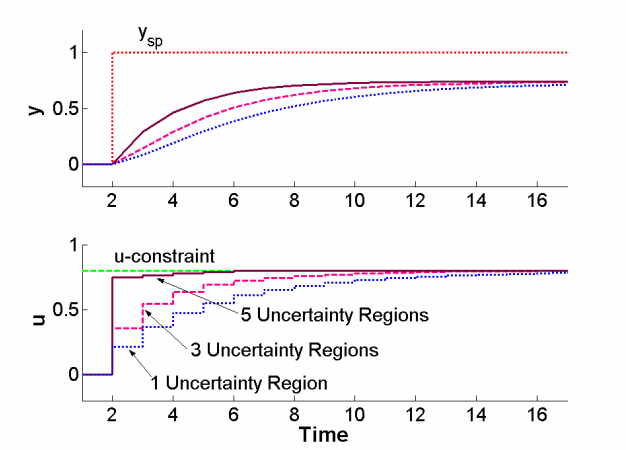

Figure 8: Nominal MPC and Robust MPC with Multiple Uncertainty Regions

Figure 8 illustrates the key benefits of the proposed MPC. The multiple uncertainty region approach creates a robust controller that does not become overly conservative when input constraints are encountered. Nor does this technique require that the system remain unconstrained at steady-state as is required by other robust MPC formulations [3, 4].

4.8 Solution Time per Iteration

The time required to solve a SOCP is clearly a function of the number of linear and conic constraints [8]. As the number of uncertainty regions increase, the number of constraints and the solution time also increase. Figure 9 shows the solution time of the SOCP for the case study discussed in section 4.7 when the number of regions, n_c, vary from one to twenty. The calculations are preformed with Sedumi's interior-point method [14] on a Pentium 4 at 1.8 GHz with 256-meg RAM.

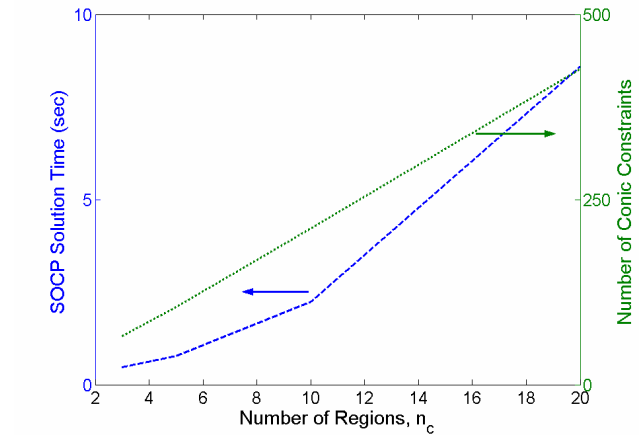

Figure 9: Solution Time vs. Number of Regions

An open question is how to best determine the number and location of the uncertainty regions. As with many other decisions in MPC, this choice is a compromise between model accuracy and computational complexity.

5. CONCLUSIONS

This paper proposes a robust MPC formulation based upon a multi-region, closed-loop uncertainty description. The proposed controller extends robust MPC by avoiding the overly conservative control that can associated with probabilistic input constraints and by allowing the system to remain on an input constraint at steady-state.

The proposed formulation uses a probabilistic, closed-loop description of system uncertainty that is calculated off-line. On-line, the MPC requires the solution of a convex second-order cone program that can be efficiently solved with existing interior-point algorithms.

6. REFERENCES

[1] A. Bemporad, "Reducing conservativeness in predictive control of constrained systems with disturbances," *Proceedings of the IEEE Conference on Decision and Control,* pp. 1384-1389, 1998.

[2] D. Mayne, "Nonlinear Model Predictive Control: Challenges and Opportunities," *Progress in Systems and Control Theory*, 26, pp. 23-44, 2000.

[3] B. Kouvaritakis, J. A. Rossiter and J. Schuurmans, "Efficient robust predictive control," *IEEE Transactions on Automatic Control*, 45, pp. 1545-1549, 2000.

[4] M. V. Kothare, V. Balakrishnan and M. Morari, "Robust Constrained Model Predictive Control using Linear Matrix Inequalities," *Automatica*, 32, pp. 1361-1379, 1996.

[5] A. L. Warren and T. E. Marlin, "Improved Output Constraint-Handling for MPC with Disturbance Uncertainty," *Proceedings of the American Control Conference*, pp. 4573-4578, 2003.

[6] P. O. M. Scokaert and D. Mayne, "Min-Max Feedback Model Predictive Control for Constrained Linear Systems," *IEEE Transactions on Automatic Control*, 43, pp. 1136-1142, 1998.

[7] Y. Lu and Y. Arkun, "Quasi-Min-Max MPC algorithms for LPV systems," *Automatica*, 36, pp. 527-540, 2000.

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

[9] A. Zheng and M. Morari, "Robust control of Constrained Model Predictive Control," *Proceedings of the American Control Conference*, pp. 379-383, 1993.

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

[11] A. Bemporad and E. Mosca, "Fulfilling Hard Constraints in Uncertain Linear Systems by Reference Managing," *Automatica*, 34, pp. 451-461, 1998.

[12] E. Zafiriou and A. L. Marchal, "Stability of SISO Dynamic Matrix Control with Hard Output Constraints," *AIChE Journal*, 37, pp. 1550-1560, 1991.

[13] D. E. Kassmann, T. A. Badgwell and R. B. Hawkins, "Robust steady-state target calculation for model predictive control," *AIChE Journal*, 46, pp. 1007-1024, 2000.

[14] J. F. Sturm, "Using SeDuMi 1.02, a MATLAB toolbox for optimization over symmetric cones," *Optimization Methods and Software*, 11-12, pp. 625-653, 1999.