MODELING, IDENTIFICATION AND CONTROL, 2004, VOL. 25, NO. 2, 67–84
doi:10.4173/mic.2004.2.1

Handling State and Output Constraints in MPC Using Time-dependent Weights

MORTEN HOVD* and RICHARD D. BRAATZ†

Keywords: *Model predictive control; constraints; inverse response; penalty functions*

A popular method for handling state and output constraints in a model predictive control (MPC) algorithm is to use 'soft constraints', in which penalty terms are added directly to the objective function. Improved closed loop performance can be obtained for plants with nonminimum phase zeros by modifying the MPC formulation to include suitably-designed time-dependent weights on the penalty terms associated with the state and output constraints. When the penalty terms are written in terms of the 'worst-case' l_∞-norm, incorporating the appropriate time dependence into the weights provides much better closed loop performance. The approach is illustrated using two multivariable plants with nonminimum phase transmission zeros, where the time-dependent weights cause the open loop predictions to coincide with closed loop predictions, which results in a reduction of output constraint violations.

1. Introduction

Most advanced controllers implemented in the process industries today are Model Predictive Control (MPC) algorithms (Garcia *et al.*, 1989; Rawlings, 1999), and such controllers are becoming increasingly popular in other industries (e.g., see Featherstone *et al.*, 2000; VanAntwerp & Braatz, 2000a, b; and references cited therein). MPC, also known as receding horizon control, uses a process model to predict the future, and then computes the future control trajectory that optimizes a performance objective based on the model predictions. The value for the control move at the current sampling instance is implemented. At the next sampling instance, new measurements are collected and the control calculation is repeated. These steps update the control move calculations to take into account the latest measurement information.

One reason for the popularity of model predictive control is its ability to directly include constraints in the computation of the control moves. This results in a linear program (LP) or quadratic program (QP) to be solved at each sampling instance, with the constraints written directly as constraints in the LP/QP. The introduction of actuator constraints usually poses no significant problems, whereas state and output constraints are more of an issue. Zafiriou showed that explicitly including these constraints in the LP/QP can destabilize otherwise stable closed loop systems controlled by a model predictive controller, as well as lead to some surprises during controller tuning (Zafiriou & Marchal, 1991). Perhaps a more critical issue is with regard to potential infeasibilities in the LP/QP. That is, a combination of actuator,

*Department of Engineering Cybernetics, Norwegian University of Science and Technology, N-7491 Trondheim, Norway, e-mail: morten.hovd@itk.ntnu.no

†University of Illinois, Urbana, Illinois 61801, email: braatz@uiuc.edu

state, and output constraints can easily result in LP/QPs in which there is no feasible solution, so the control algorithm cannot compute any control move to be implemented at that sampling instance. Such behavior cannot be tolerated, so it has become common to use a *soft constraint* formulation to handle state and output constraints, in which penalty terms on the constraints are included in the objective function (de Oliveira & Biegler, 1994; Ricker *et al.*, 1989; Vuthandam *et al.*, 1994; Zheng & Morari, 1995). A soft constraint formulation avoids infeasibility problems by allowing violations in the state and output constraints, but attempts to minimize the violations by placing large enough penalties on constraint violations in the objective function.

It is desirable to select the penalty terms to both avoid extreme violations in the state and output constraints, and to minimize the length of time for which the violations occur. These two objectives are competing for processes with nonminimum phase zeros, with different norms for the penalty terms having different levels of effectiveness in jointly addressing these objectives (Scokaert & Rawlings, 1999). Also, it is desirable for the MPC algorithm to be formulated so that the control weights are tunable in an intuitive manner, since this is critically important for a control algorithm to be effective in practice.

The different norms that have been proposed for quantifying the constraint violations in the penalty functions include:

- The l_1-norm, which is the sum over all time steps in the prediction horizon of the absolute value of the predicted constraint violations.
- The l_2-norm, which is obtained by squaring the predicted constraint violations, and then summing over all time steps in the prediction horizon.
- The l_∞-norm, which is the absolute value of the largest predicted constraint violation within the prediction horizon.

The l_∞-norm has been a popular choice among control theorists (de Oliveira & Biegler, 1994; Zheng & Morari, 1995; Allwright & Papavasiliou, 1992; Campo & Morari, 1986; Dave *et al.*, 1999). One reason for its popularity may be that the l_∞-norm only increases the number of free variables in the optimization by the number of soft constraints, whereas the l_1- and l_2-norms increase the number of free variables by the number of soft constraints multiplied by the number of time steps in the prediction horizon. Also, for some control problems it is appealing from an engineering point of view to minimize the worst-case constraint violation.

In a recent paper, Scokaert & Rawlings (1999) showed that unexpected closed loop behavior could occur, depending on which norm was used in the penalty terms on the constraint violations. One of the key results was that the l_∞-norm can result in very poor closed loop performance for processes with nonminimum phase zeros. Such plants are very common in practice, especially for multivariable processes, which can be nonminimum phase even if each element in the transfer function is minimum phase (Morari & Zafiriou, 1989). Scokaert & Rawlings (1999) reported the following two problems with the l_∞-norm measure of constraint violations:

1. non-intuitive effects of changing the tuning parameters in the penalty function which define how the constraint violations increase the objective function in the optimization, and
2. actual closed loop performance was significantly poorer than the predicted open loop performance.

A very convincing argument was given that these two problems are closely related, that is, it is the fact that the actual closed loop performance in the future can be very different from the predicted performance at a particular time instance that results in non-intuitive tuning.

These problems did not appear when the l_1- and l_2-norms were used. Scokaert and Rawlings therefore recommended the use of the l_1- or l_2-norms as measures of the magnitude of constraint violations, even though the use of these norms results in a larger number of degrees of freedom in the optimization problems, and the norms may be considered to be less appealing for some control problems.

In a recent paper (Hovd & Braatz, 2001), we showed that the l_∞-norm can be used, provided that the MPC algorithm be formulated to include a suitably-tuned time-varying weight on the constraint violations to be penalized in the objective function. From the point of view of the MPC algorithm at a given sampling instance, the weight varies only with respect to the prediction horizon. A simple tuning rule specifies how the time-dependence is selected. The results were illustrated on a single-input single-output (SISO) system with three states, two nonminimum phase zeros, and two highly oscillatory poles.

While inverse response is common in SISO systems, nonminimum phase transmission zeros are probably more prevalent in multivariable systems. This paper extends the approach to multivarible systems, and illustrates the approach by application to two systems. By the appropriate selection of time-varying weights, the predicted open loop responses are consistent with the actual closed loop responses, which results in improved handling of the output constraints.

2. MPC Formulation

The linear discrete-time state space model is

$$\begin{aligned} x_{k+1} &= Ax_k + Bu_k \\ y_k &= Cx_k \end{aligned} \tag{1}$$

which is assumed to be minimal. At timestep k, the model predictive controller solves the optimization

$$\min_{U,\varepsilon} \sum_{i=0}^{N-1} x_{k+i}^T Q x_{k+i} + u_{k+i}^T \, Ru_{k+i}) + x_{k+N}^T S x_{k+N} + f(\varepsilon) \tag{2}$$

subject to

$$\begin{aligned} u_L \leqslant u_{k+i} &\leqslant u_U \\ Cx_{k+i} - W_i\varepsilon_U &\leqslant y_U \\ Cx_{k+i} + W_i\varepsilon_L &\geqslant y_L \\ \varepsilon_U &\geqslant 0 \\ \varepsilon_L &\geqslant 0 \end{aligned} \tag{3}$$

where $U = [u_k, u_{k+1}, \ldots, u_{k+N-1}]^T$ is the vector of control moves computed over the control horizon N. The matrix R is positive definite, whereas Q is positive semidefinite. The 'cost to go', S, is a positive defintite matrix. In this work, it is given by the solution to the discrete-time Lyapunov equation

$$S = (A + BK)^T S(A + BK) + Q \tag{4}$$

where K is the infinite horizon LQ-optimal controller for the weights Q and R. The constraints are enforced over an extended horizon of length $N+j_2$. A sufficiently large value for j_2 ensures that the constraints will be feasible on an infinite horizon if they are feasible up to the horizon $N+j_2$, see Rawlings & Muske (1993) for details. Note that it is necessary to ensure that the manipulated variable constraints are also feasible up to $N+j_2$; since it is assumed that state feedback using K is in effect from N until $N+j_2$ this effectively becomes an additional set of state constraints over that period. The function $f(\varepsilon)$ is the penalty function that increases the optimization criterion when the soft constraints are violated, and

$$\varepsilon = \begin{bmatrix} \varepsilon_U \\ \varepsilon_L \end{bmatrix}^T \tag{5}$$

is the vector of weighted l_∞-norms of the constraint violations in the upper and lower soft constraints (for brevity only output constraints are listed here; the generalization to state constraints is straightforward). The penalty function $f(\varepsilon)$ is given by

$$f(\varepsilon) = \varepsilon^T P \varepsilon + p^T \varepsilon \tag{6}$$

The matrix P is assumed to be positive definite, in order to be able to solve the optimization problem as a standard QP program. The vector p has non-negative elements, and sufficiently large elements in p will ensure that the constraints are enforced as *exact* soft constraints, that is, that constraint violations will only occur when there is no feasible solution with hard constraints (see an optimization book for a discussion on exact penalty functions, e.g. Peressini *et al.*(1988), or de Oliveira & Biegler (1994) for a discussion within the context of MPC).

Instead of including in the objective function the time dependence of the weights for the soft constraints, the non-negative matrices W_i in the constraint equations contain the inverse of that time dependence:

$$[W_i]_{j,j} = \frac{1}{(r_j)^i} \tag{7}$$

with $r_j \geqslant 1$. When evaluating the constraint violation vector for constraint j at different time steps i in equation (3), we find that when a constant magnitude is predicted for the actual constraint violation the required magnitude of ε_j has to increase by a factor of r_j for every time step in order to fulfill the constraints. Thus, predicted constraint violations in the far future are given more weight than violations in the near future. We know that for plants with inverse response, a larger constraint violation is required initially to drive the output back into the feasible region. This is achieved by selecting $r_j > 1$, but would not be optimal according to equation (2) if constraint violations are given the same weight throughout the prediction horizon. Furthermore, this is achieved while using only a single constraint violation variable for each actual constraint over the whole prediction horizon.

The magnitude of r_j is set based on the location of the nonminimum phase zeros in the plant. It was recommended that the value of each r_j should be greater than the magnitude of the absolute value of the nonminimum phase zero farthest from the origin while significantly affecting the process dynamics (Hovd & Braatz, 2001). In extensive simulations, this lower bound on the r_j resulted in an MPC algorithm that did not have either of the two problems observed by Scokaert & Rawlings. That is, the effects of changing the tuning parameters in the penalty function became intuitive, and the actual closed loop performance was close to the open loop predicted performance. The lower bounds were also given a physical explanation in terms of the nonminimum phase influence on the process dynamics.

In this paper, we show that the same tuning criterion applies to multivariable systems with nonminimum phase transmission zeros. The approach is first illustrated using a stable process with one nonminimum phase zero. Then a multivariable process with much more complex nonminimum phase behavior is investigated.

3. Example 1

Consider a process with the 2 × 2 continuous transfer function

$$P(s) = \begin{bmatrix} \dfrac{1}{(5s+1)(10s+1)} & \dfrac{-1}{(5s+1)(10s+1)} \\ \dfrac{-2s+1}{(5s+1)(10s+1)} & \dfrac{4s-2}{(5s+1)(10s+1)} \end{bmatrix} \tag{8}$$

This process has a nonminimum phase transmission zero pinned to the second output. Such pinned zeros are quite common in practice (Skogestad & Postlethwaite, 1996), for example, such a zero would occur if the sensor for the second output demonstrated inverse response.

This process was sampled using zeroth order hold and a sampling interval of 0.5 min. This process was sampled with a sampling interval of 0.5 min. The resulting discrete-time plant has transmission zeros at 1.2869 and -0.9512, two poles at 0.9048, and two poles at 0.9512. The open loop step responses are shown in Figure 1.

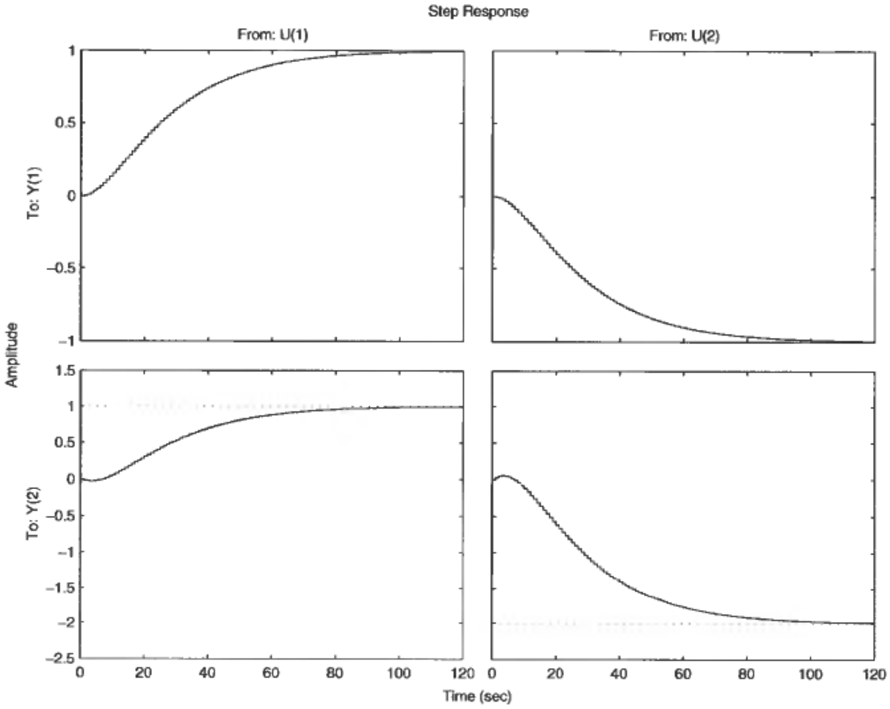

Figure 1. Open loop unit step responses for Example 1.

The inputs and outputs are constrained between ±1. The controller tuning parameters are $Q = C^T C$, $R = I_2$, $N = 20$, $P = 0.01$ and $p = 100$. The relatively large number for p was selected to encourage the MPC algorithm to satisfy the output constraints as quickly as possible. Throughout this example, the excess constraint horizon $j_2 = 70$ is used. This value for j_2 was checked in the simulations and found to be sufficiently large to guarantee feasibility of the constraints on the infinite horizon according to the criterion of Rawlings & Muske (1993).

The initial condition is taken to be

$$x_0 = \begin{bmatrix} 0 \\ 0 \\ -20 \\ -1 \end{bmatrix} \tag{9}$$

so that the output constraint is violated at $t = 0$.

For the case of time-invariant weights ($r_j = 1$), both closed loop output responses are very sluggish, with the output constraints being violated for the first 50 samples (see Figure 2). The second achieved plant output is significantly more sluggish than what is predicted at $t = 0$, and the plant inputs computed at $t = 0$ are very different from the plant inputs that occur when the control is implemented in a receding horizon manner (see Figure 3). This difference in the predicted and achieved plant inputs results in the poor closed loop performance seen in Figure 2.

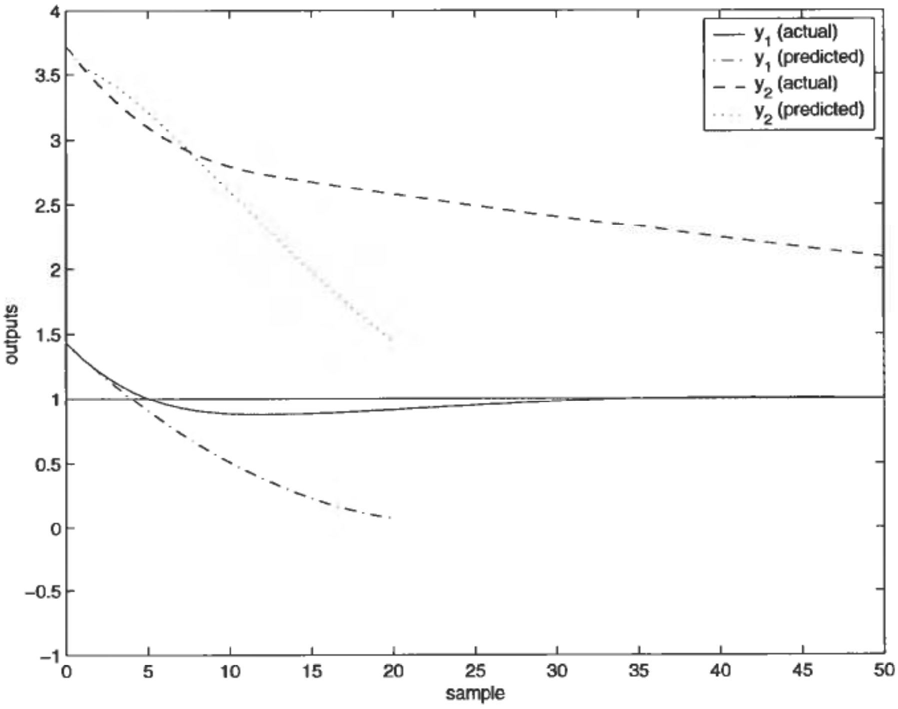

Figure 2. Plant output with $r_j = 1$ for Example 1.

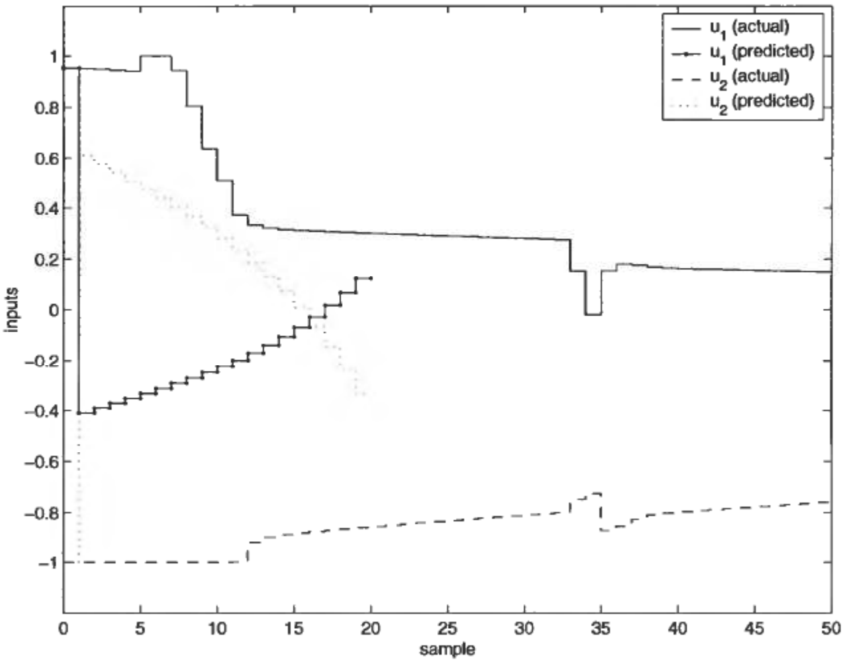

Figure 3. Plant input with $r_j = 1$ for Example 1.

Following the design specification, the rise parameters r_j should be greater than or equal to the magnitude of the nonminimum phase transmission zero. For $r_j = 1.2869$, the predicted and achieved plant inputs are nearly equal for the entire control horizon of 20 (see Figure 4), and the predicted and achieved plant outputs are nearly equal (see Figure 5). The closeness between predictions and what is achieved by the receding horizon control algorithm leads to much better output constraint-handling capability, with both output constraints satisfied within 30 min.

The plant inputs and outputs for $r_j = 2.5$ are shown in Figures 6 and 7. The differences between the predicted and achieved plant inputs and outputs are negligible. The output constraints are satisfied for the second output somewhat sooner than for $r_j = 1.2869$.

For this example the nonminimum phase transmission zero is pinned to the second output (see equation (8)), so only the rise parameter for the second output (r_2) needs to be selected greater than the magnitude of the zero (r_1 can be set to 1 so that the weight on the first output is time-invariant). The plant outputs and inputs for $r_1 = 1$ and $r_2 = 1.2869$ are nearly indistinguishable to the case where $r_1 = r_2 = 1.2869$ (see Figures 8 and 9). Similarly, the closed loop responses for $r_1 = 1$ and $r_2 = 2.5$ are nearly indistinguishable from the responses for $r_1 = r_2 = 2.5$. In contrast, setting the rise parameter for the first output equal to the nonminimum phase zero, and the rise parameter for the second output to one, gives poor performance (see Figures 10 and 11).

Based on these observations, the tuning rule can be modified for zeros pinned to particular outputs. In this case, the rise parameters can be set to one for the outputs unaffected by the pinned zeros. On the other hand, it appears that there may be little change in closed loop performance by selecting all of the rise parameters greater than one.

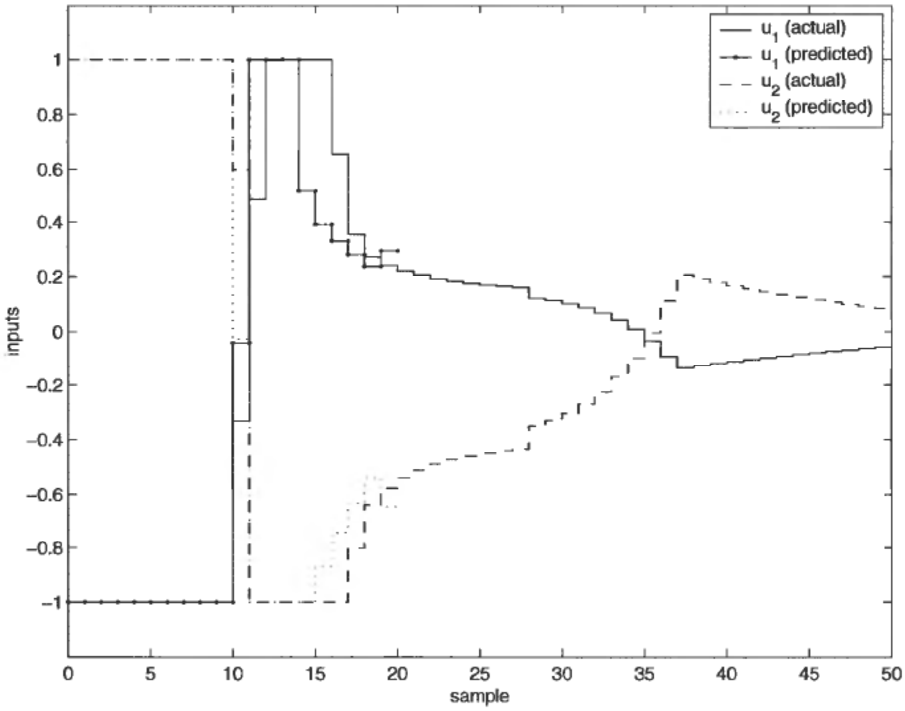

Figure 4. Plant input with $r_j = 1.2869$ for Example 1.

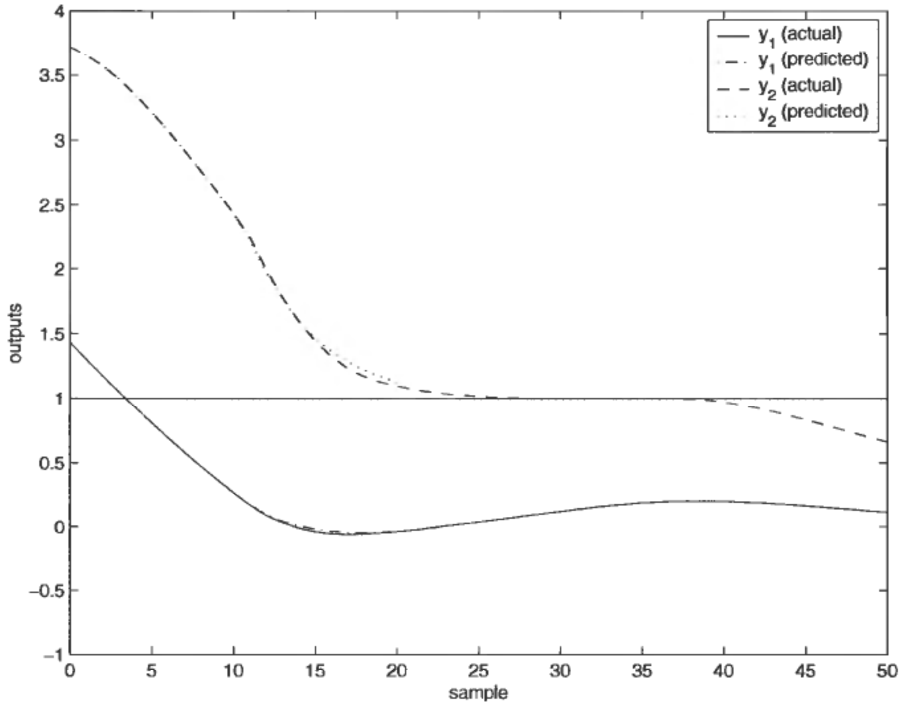

Figure 5. Plant output with $r_j = 1.2869$ for Example 1.

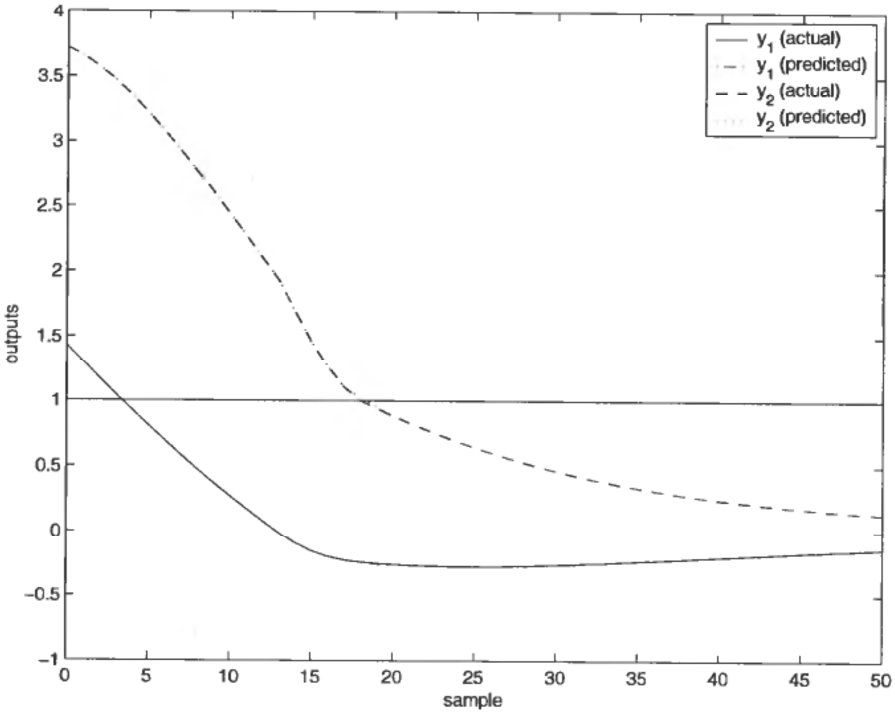

Figure 6. Plant output with $r_j = 2.5$ for Example 1.

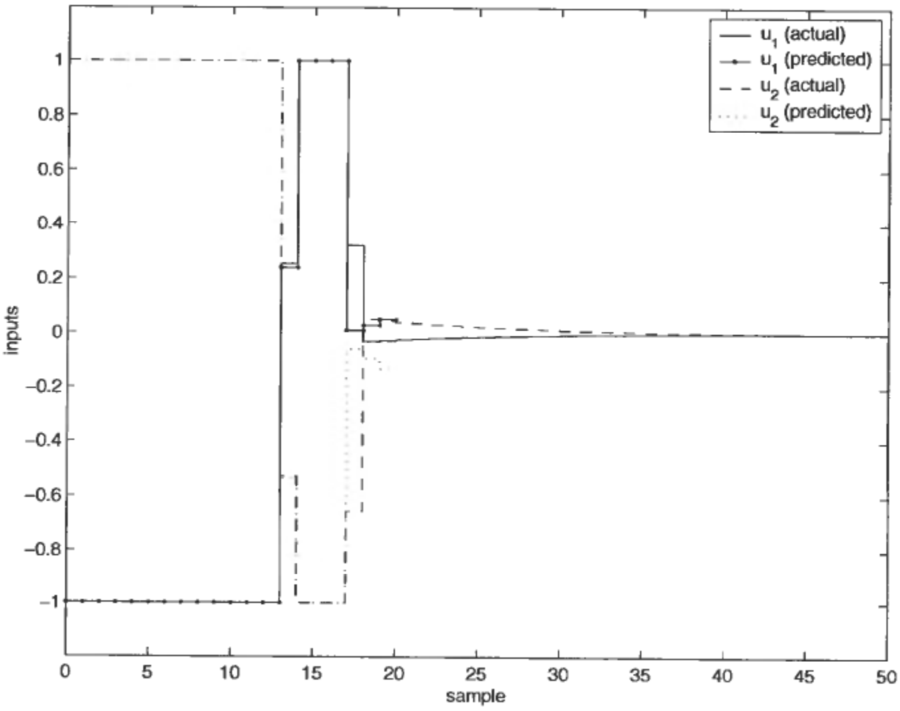

Figure 7. Plant input with $r_j = 2.5$ for Example 1.

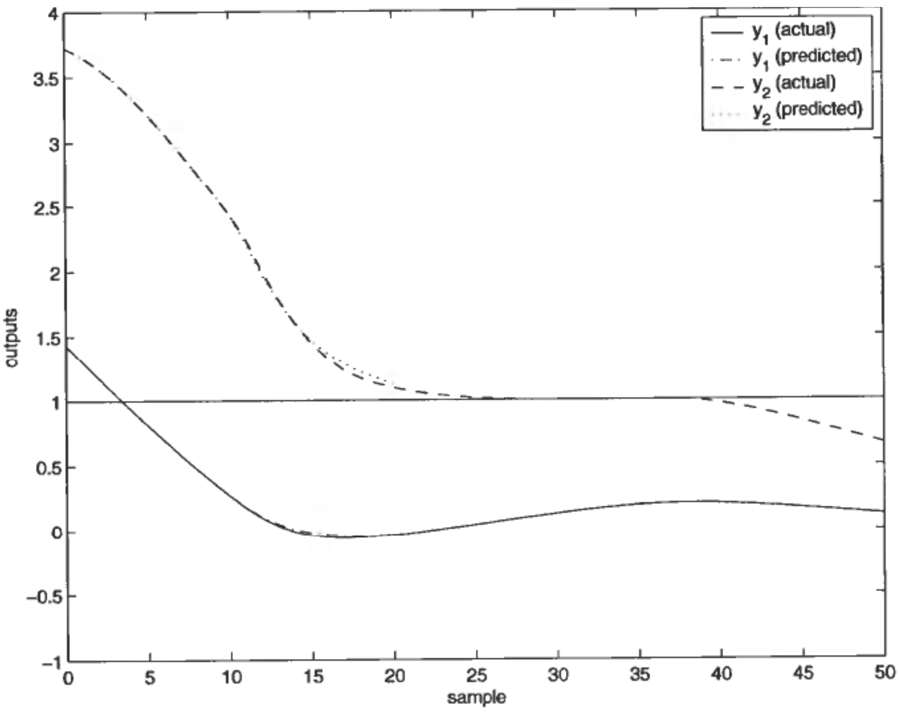

Figure 8. Plant output with $r_1 = 1$ and $r_2 = 1.2869$ for Example 1.

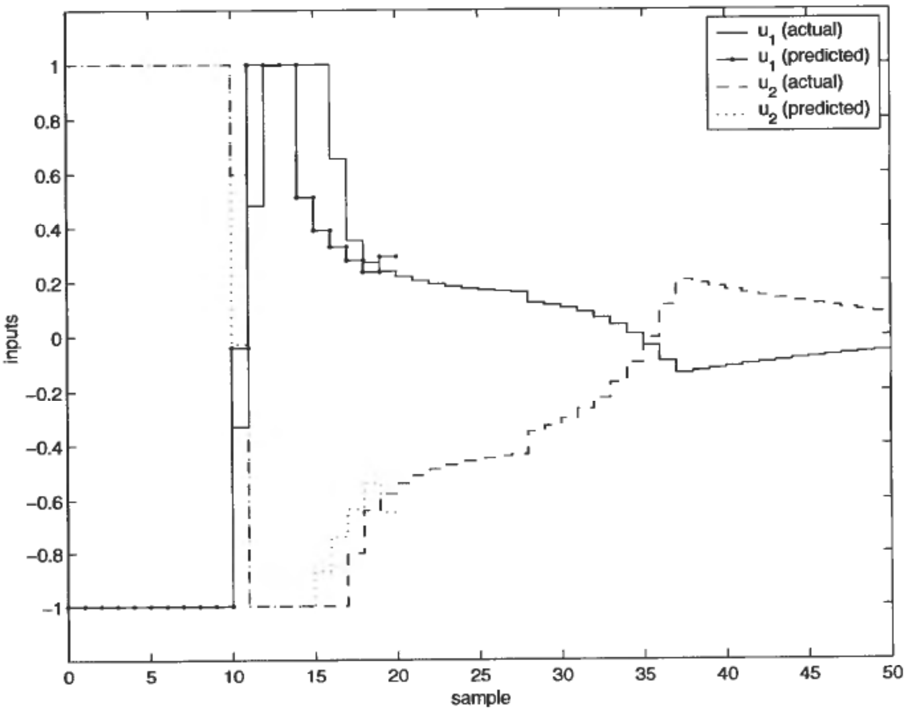

Figure 9. Plant input with $r_1 = 1$ and $r_2 = 1.2869$ for Example 1.

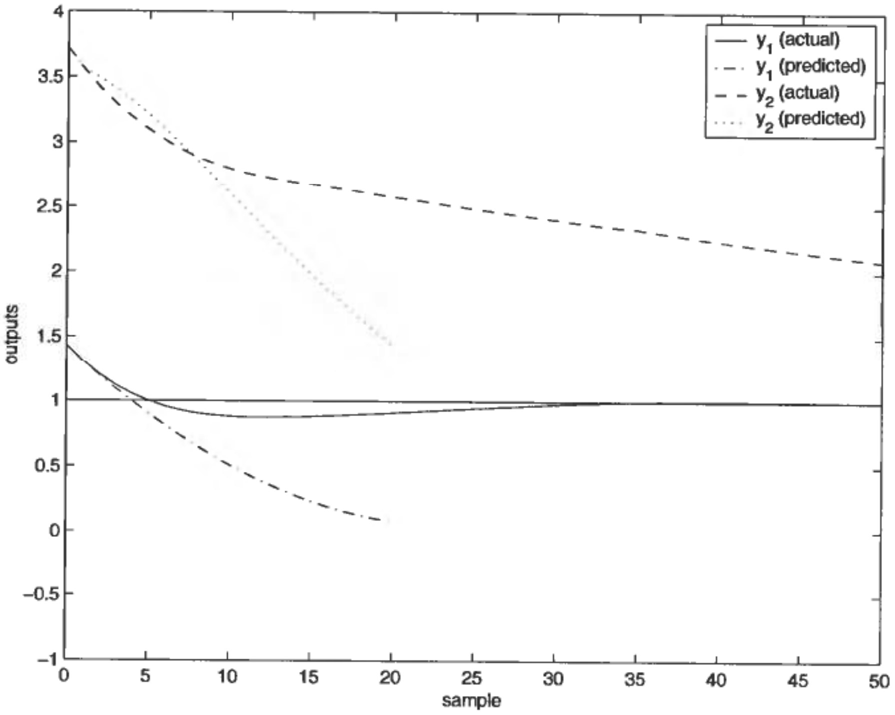

Figure 10. Plant output with $r_1 = 1.2869$ and $r_2 = 1$ for Example 1.

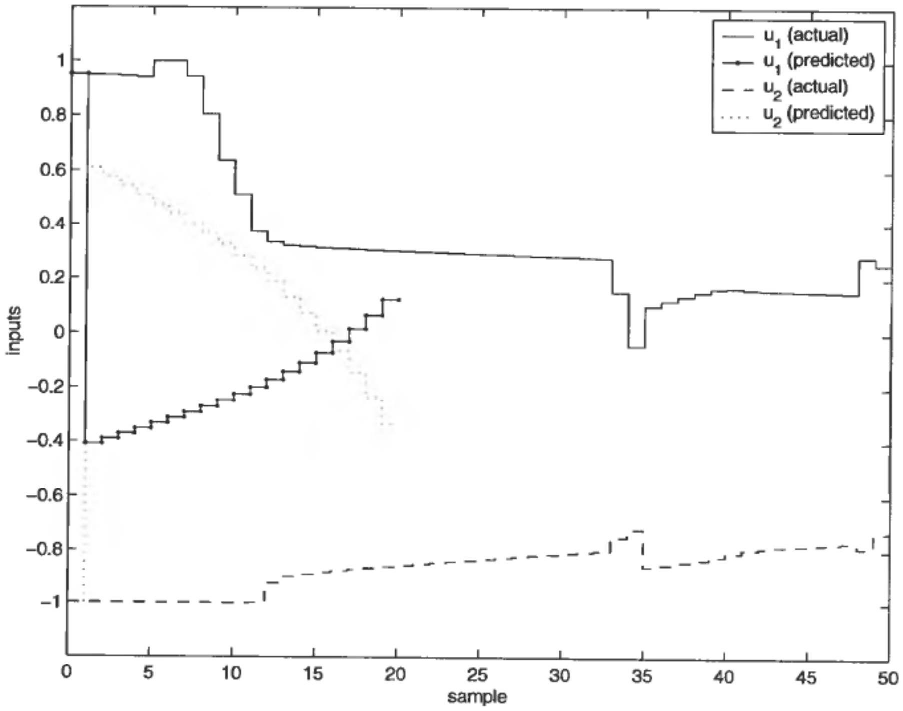

Figure 11. Plant input with $r_1 = 1.2869$ and $r_2 = 1$ for Example 1.

4. Example 2

Consider a process with the 2 × 2 continuous transfer function

$$P(s) = \begin{bmatrix} \dfrac{1.2(-s+5)}{(s+1)(5s+1)} & \dfrac{-2s+5}{(4s+1)(5s+1)} \\ \dfrac{(-2s+5)(-s+5)}{(2s+1)(3s+1)(5s+1)} & \dfrac{-2s+5}{(s+1)(6s+1)} \end{bmatrix} \tag{10}$$

This process was sampled with a sampling interval of 0.25 min. The multivariable poles and zeros for the discrete-time plant are shown in Figure 12. The nonminimum phase transmission zeros are 1.0808, 1.9307, and 4.7261. The open loop step responses are in Figure 13.

The inputs and outputs are constrained between ±1. The controller tuning parameters are $Q = C^T C$, $R = I_2$, $N = 20$, $P = 0.01$ and $p = 100$. Throughout this example, the excess constraint horizon $j_2 = 110$ is used. This value for j_2 was checked in the simulations and found to be sufficiently large according to the criterion of Rawlings & Muske (1993). The initial condition is

$$x_0 = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 2.6350 \\ -1.6484 \end{bmatrix}$$

so that the output constraint is violated at $t = 0$.

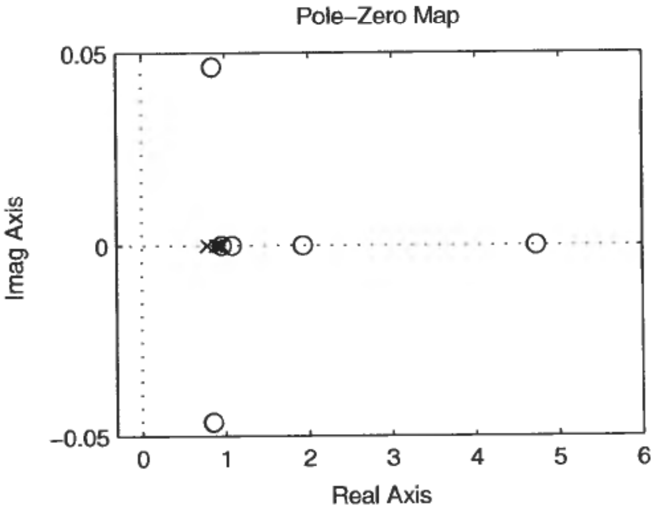

Figure 12. Multivariable plots and zeros for Example 2.

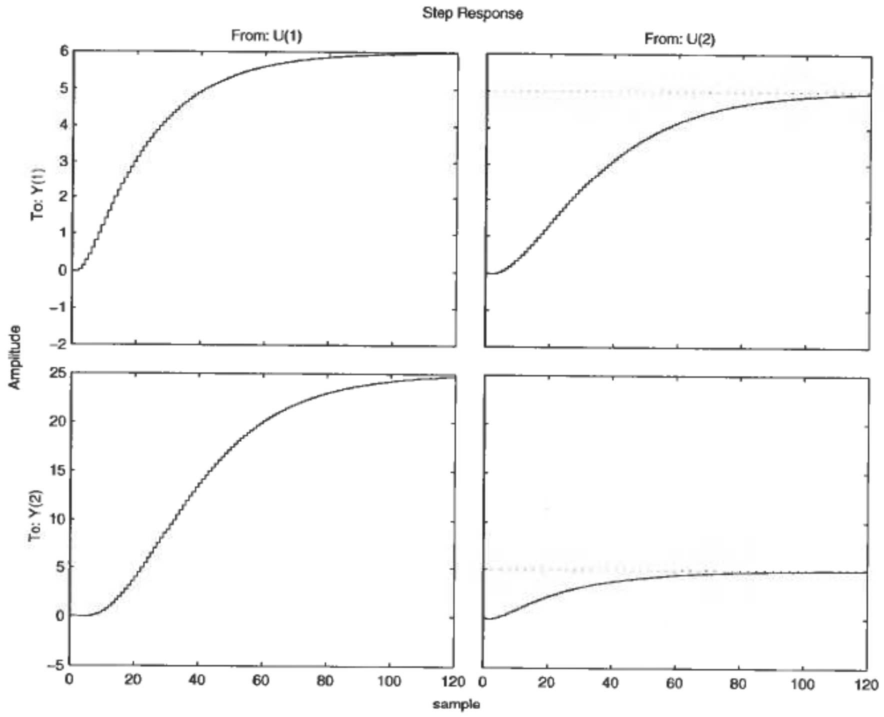

Figure 13. Open loop unit step responses for Example 2.

For the case of time-invariant weights ($r_j = 1$), the second output requires 50% more time to satisfy the output constraint than for its predicted value (see Figure 14). Trying to reduce the time of constraint violation by reducing the magnitude on the control weight resulted in an ill-conditioned Hessian in the quadratic program. The plant inputs computed at $t = 0$ are very different from the plant inputs that actually occur when the control is implemented in a receding horizon manner (see Figure 15). This difference in the predicted and achieved plant inputs leads to the poor closed loop performance seen in Figure 14.

Following the design specification, the rise parameters r_j should be greater than or equal to the magnitude of the nonminimum phase transmission zero. For $r_j = 5$, the predicted and achieved plant inputs are equal for the first 8 sampling instances and qualitatively similar for the first 13 sampling instances (see Figure 16), and the predicted and achieved plant outputs are nearly equal for the first 16 sampling instances (see Figure 17). The closeness between predictions and what is achieved by the receding horizon control algorithm leads to much better output constraint-handling capability, with both output constraints satisfied within 10 sampling instances.

The plant inputs and outputs for $r_j = 2$ are shown in Figures 18 and 19. The responses are similar to those for $r_j = 5$. This would seem to violate the tuning rule that r_j should be greater than the magnitude of the largest nonminimum phase transmission zero. However, the transmission zero at 4.7261 is too large in magnitude to have much of an effect on the dynamics. Thus r_j can be selected to be larger than the magnitude of the largest nonminimum phase transmission zero that significantly affects the dynamics, which is at 1.9307.

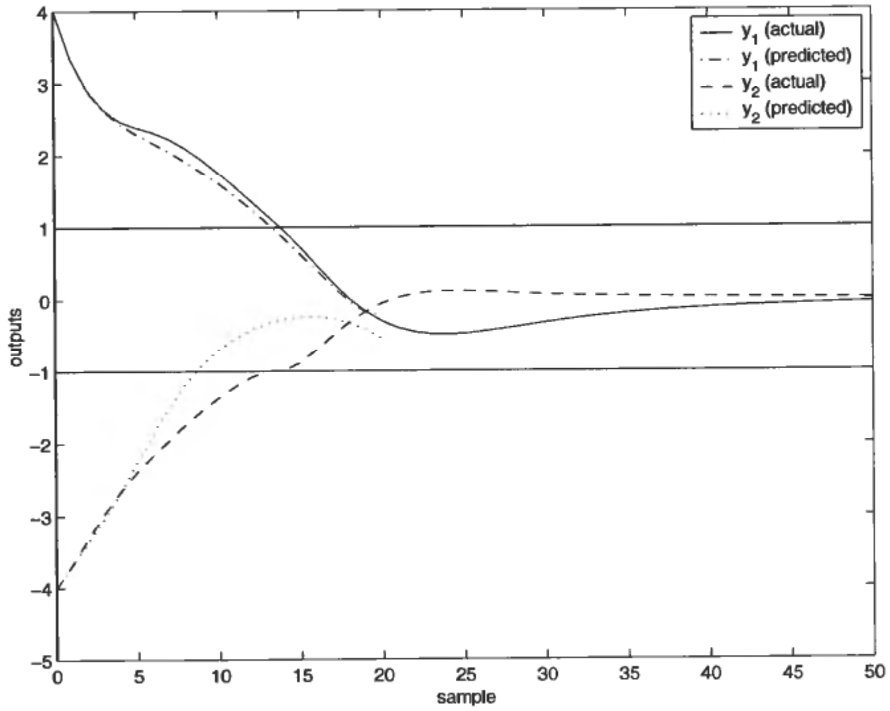

Figure 14. Plant output with $r_j = 1$ for Example 2.

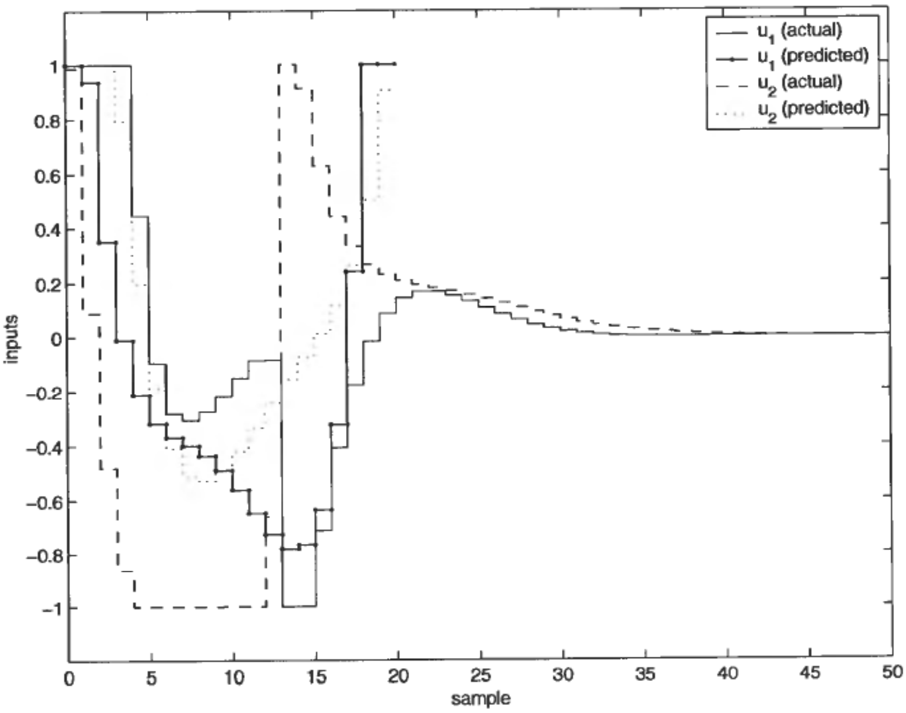

Figure 15. Plant input with $r_j = 1$ for Example 2.

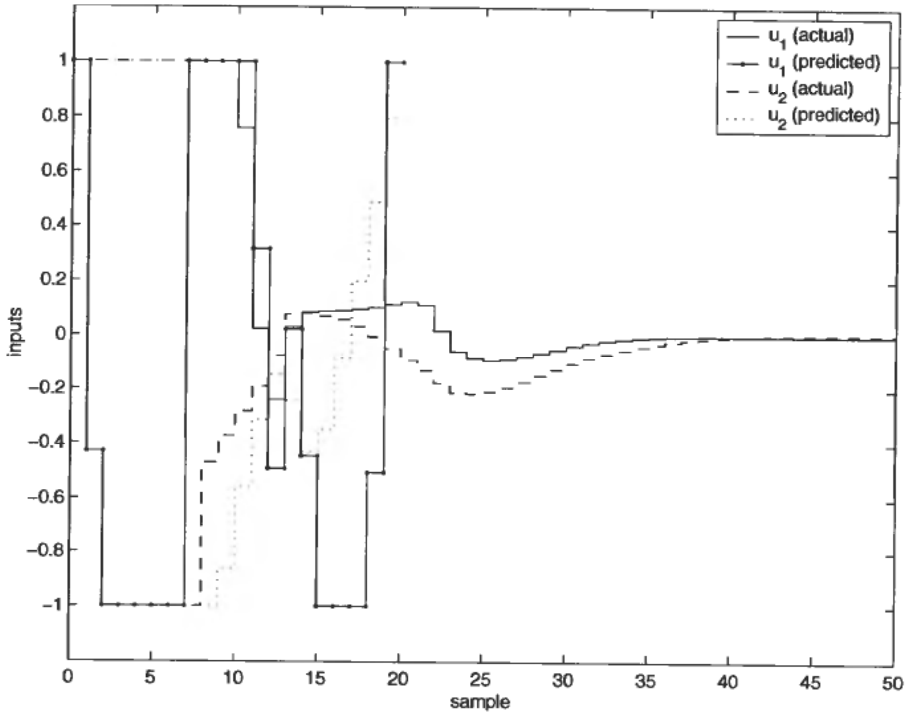

Figure 16. Plant input with $r_j = 5$ for Example 2.

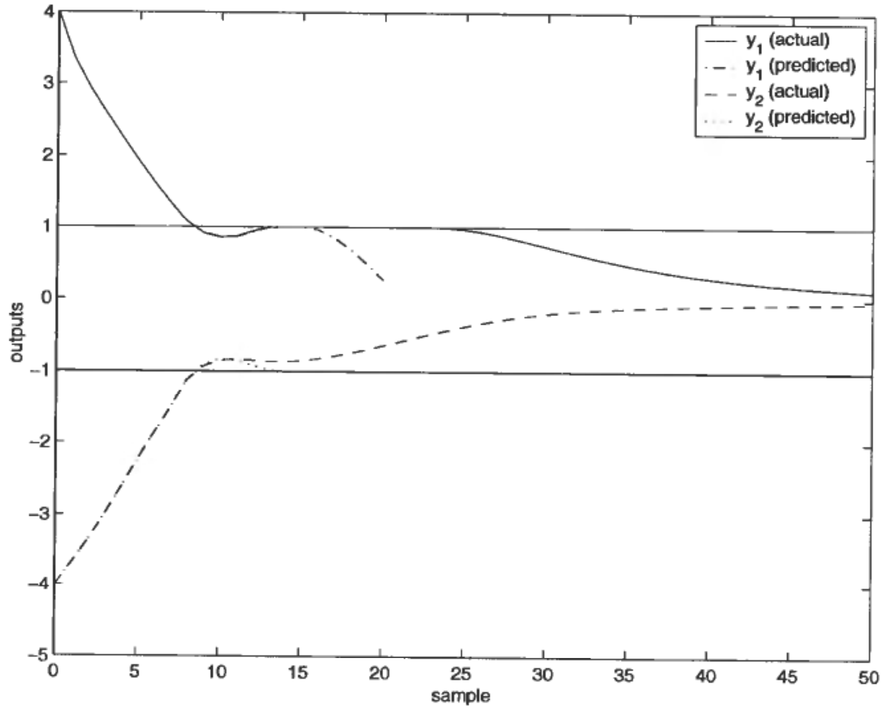

Figure 17. Plant output with $r_j = 5$ for Example 2.

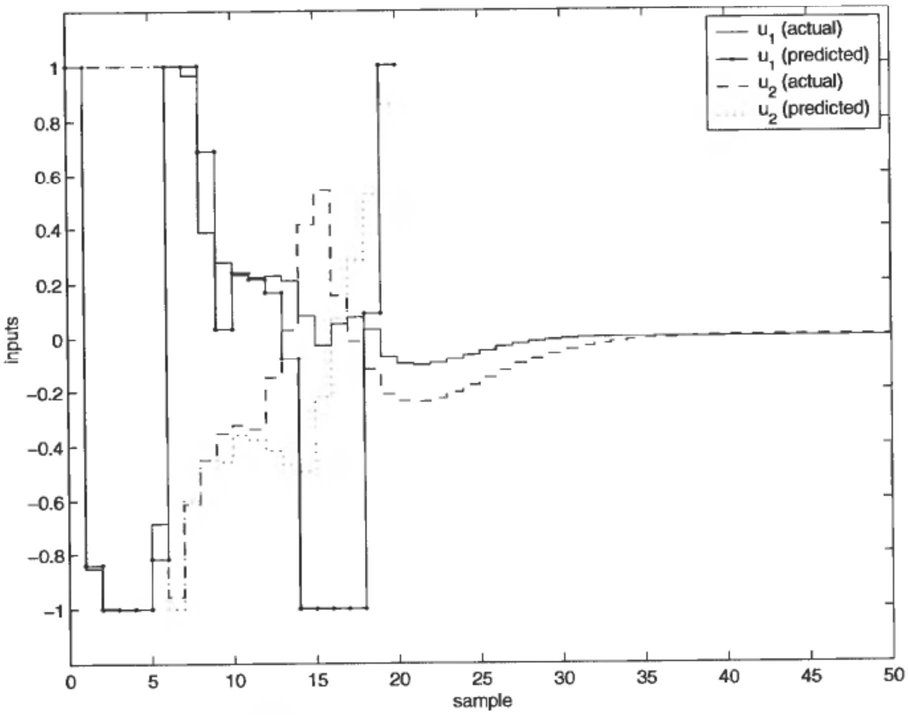

Figure 18. Plant input with $r_j = 2$ for Example 2.

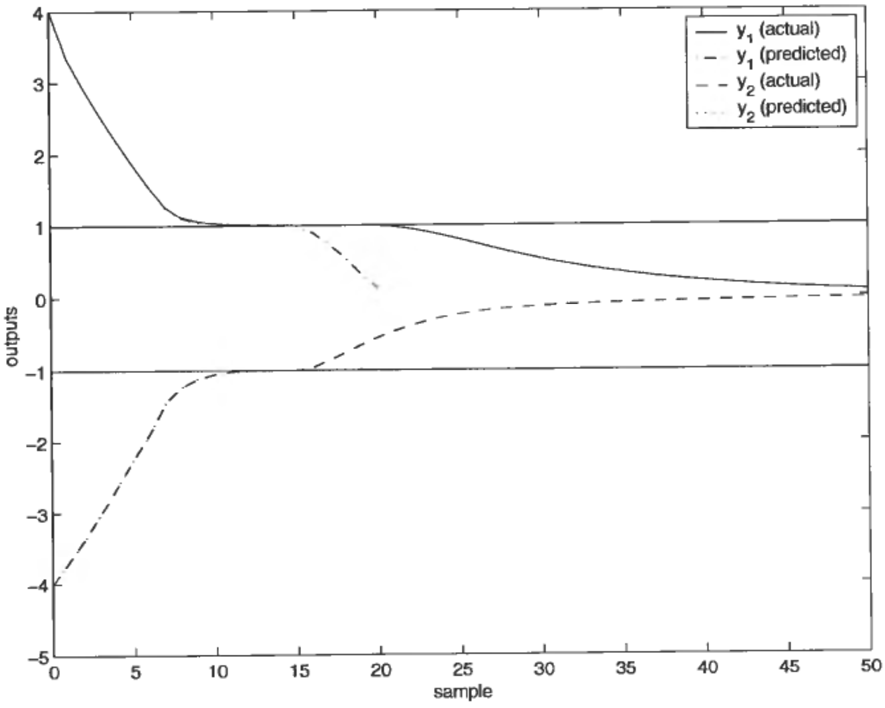

Figure 19. Plant output with $r_j = 2$ for Example 2.

5. Conclusions

The introduction of time-varying output penalty weights into the l_∞-norm MPC formulation removes the drawbacks identified by Scokaert & Rawlings (1999). Two multivariable examples illustrated the effectiveness of the approach. The original tuning rule (Hovd & Braatz, 2001) was modified so to ignore nonminimum phase transmission zeros that are too large to significantly affect the process dynamics.

The tuning can also be modified to take into account the direction of the nonminimum phase transmission zeros. In particular, time-varying weights are not needed for plant outputs that are not significantly affected by the nonminimum phase transmission zeros (for example, as occurs with output pinned zeros). On the other hand, there may not be a significant difference in closed loop performance by selecting all of the rise parameters greater than one.

6. Acknowledgements

The first author acknowledges the support of Fantoft Prosess AS. The second author acknowledges the support of the National Center for Supercomputing Applications.

References

Allwright, J. C. & Papavasiliou, G. C. (1992). On linear programming and robust model predictive control using impulse responses, *Systems and Control Letters*, 18: 159–164.

Campo, P. J. & Morari, M. (1986). Infinity-norm formulation of model predictive control problems, *Proc. of the American Contorl Conf.*, pp. 339–343 (New York: IEEE Press).

Dave, P., Doyle, F. J., III & Pekny, J. F. (1999). Customization strategies for the solution of linear programming problems arising from large scale model predictive control of a paper machine, *J. of Process Control*, 9: 385–396.

de Oliveira, N. M. C. & Biegler, L. T. (1994). Constraint handling and stability properties of model-predictive control, *AIChE J.*, 40: 1138–1155.

Featherstone, A. P., VanAntwerp, J. G. & Braatz, R. D. (2000). *Identification and Control of Sheet and Film Processes* (London, UK: Springer Verlag).

Garcia, C. E., Prett, D. M. & Morari, M. (1989). Model predictive control: Theory and practice—A survey, *Automatica*, 25: 335–348.

Hovd, M. & Braatz, R. D. (2001). On the use of soft constraints in MPC controllers for plants with inverse response, *Proc. of the 6th IFAC Symp. on Dynamics and Control of Process Systems* (Kidlington, UK: Elsevier Science), in press.

Morari, M. & Zafiriou, E. (1989). *Robust Process Control* (Englewood Cliffs, NJ: Prentice-Hall).

Peressini, A. L., Sullivan, F. E. & Uhl, J. J., Jr (1988). *The Mathematics of Nonlinear Programming* (New York: Springer-Verlag).

Rawlings, J. B. (1999). Tutorial: Model predictive control technology, *Proc. of the American Control Conf.* (Piscataway, New Jersey), pp. 662–676, IEEE Press.

Rawlings, J. B. & Muske, K. R. (1993). The stability of constrained receding horizon control, *IEEE Trans. on Automatic Control*, 38(10): 1512–1516.

Ricker, N. L., Subrahmanian, T. & Sim, T. (1989). Case studies of model-predictive control in pulp and paper production, *Proc. of the IFAC Workshop on Model Based Process Control* (T. J. McAvoy, Y. Arkun & E. Zafiriou, eds.), pp. 13–22 (Oxford, UK: Pergamon Press).

Skogestad, S. & Postlethwaite, I. (1996). *Multivariable Feedback Control: Analysis and Design* (New York: John Wiley & Sons).

Scokaert, P. O. M. & Rawlings, J. B. (1999). Feasibility issues in linear model predictive control, *AIChE J.*, 45: 1649–1659.

VanAntwerp, J. G. & Braatz, R. D. (2000a). Fast model predictive control of sheet and film processes, *IEEE Trans. on Control Systems Technology*, 8: 408–417.

VanAntwerp, J. G. & Braatz, R. D. (2000b). Model predictive control of large scale processes, *J. of Process Control*, 10: 1–8.

Vuthandam, P., Genceli, H. & Nikolaou, M. (1994). Analysis and synthesis methods for robust model predictive control, *Proc. of the IFAC Symp. on Advanced Control of Chemical Processes*, pp. 167–172 (Kidlington, UK: Elsevier Science).

Zafiriou, E. & Marchal, A. L. (1991). Stability of SISO quadratic dynamic matrix control with hard output constraints, *AIChE J.*, 37: 1550–1560.

Zheng, A. & Morari, M. (1995). Stability of model predictive control with mixed constraints, *IEEE Trans. on Automatic Control*, 40: 1818–1823.