Model Predictive Control of an Adhesive Coating Process

Introduction Model predictive control (MPC) has become the dominant multivariable control algorithm in the chemical and petrochemical process industries. One of the main reasons for the widespread application of this method is its ability to explicitly address constraints on process variables. This assignment considers the design a model predictive control system for an adhesive coating process. The input-output behavior of this process can be represented by a pure time delay and a multivariable interactions matrix, which allows the control tuning to focus on handling multivariable interactions, disturbances, measurement noise, and constraints without the added complication of treating general process dynamics. This handout describes the formulation of a model predictive control algorithm for this process, following by 5 problems.

Process Consider a process with m manipulated variables and n process outputs where all of the process outputs are measured; a similar derivation holds in the general case but would require a more cumbersome notation. Many processes behave approximately linearly within a normal operating region. Let the manipulated variables be represented by u, the measured variables by y, and the disturbances by d. If the process dynamics can be modeled as a pure delay, then the process outputs at sampling instance t is related to the disturbances and the manipulated variables at the previous sampling instance $t-1$ by

$$y(t) = Pu(t-1) + d(t), \tag{1}$$

where P is a constant $n \times m$ matrix that models the interactions between process inputs and outputs.

The vector d accounts for measurement noise and the effect of all other unmeasured disturbances on the process output (the technique can be generalized to measured disturbances). The disturbance vector d is assumed to be a stochastic variable, that is, $\{d(0), d(1), \cdots, d(t), \cdots\}$ is a sequence of independent random vectors, potentially with non-zero mean (for a more detailed description of stochastic variables, see [1]). The disturbances are chosen to be stochastic because this describes well the apparently random fluctuations of the process.

It will be convenient to express the representation for the true process in terms of changes in the inputs rather than the inputs themselves. For this purpose, subtract (1) evaluated at the previous sampling instance $t-1$ from that at t to arrive at

$$y(t) = y(t-1) + P\Delta u(t-1) + \Delta d(t), \tag{2}$$

where

$$\Delta u(t-1) = u(t-1) - u(t-2); \tag{3}$$

$$\Delta d(t) = d(t) - d(t-1). \tag{4}$$

The variable $\Delta d(t)$ will be a stochastic variable with zero mean. With the above assumptions on unmeasured disturbances $d(t)$, $\Delta d(t)$ is referred to as *white noise* [1]. More general stochastic and/or deterministic disturbances assumptions can be handled (for example, see [2, 3]).

Model Identification Industrial model predictive control algorithms are always coupled with software for process model identification. Let us consider the identification of the plant interactions matrix P from open-loop input-output data. Various changes are made in the manipulated variables u, and the resulting process outputs y are measured. A superscript ^{i} refers to the ith open-loop experiment (e.g., u^i refers to the process input for the ith experiment). The inputs should be chosen as large as possible to diminish the effect of the unmeasured disturbances d, while satisfying any operating constraints. The standard approach in parameter fitting is to minimize the least squares of the errors in the model prediction [1, 4]. This approach will give good models as long as the process interactions are not too highly coupled [5, 6, 7].

Write (2) for each of q experiments, where the time argument is suppressed to simplify notation:

$$\Delta y^i = P\Delta u^i + \Delta d^i, \quad \text{for} \quad i = 1, \ldots, q. \tag{5}$$

Define P_k as the kth column of P, Δu_k^i as the kth element of Δu^i, and I_n as the $n \times n$ identity matrix. To compute P by least squares parameter estimation, write (5) as

$$\begin{aligned}
\Delta y^i &= \begin{bmatrix} \Delta y_1^i \\ \vdots \\ \Delta y_n^i \end{bmatrix} && (6) \\
&= [P_1 \,|\, \cdots \,|\, P_m] \begin{bmatrix} \Delta u_1^i \\ \vdots \\ \Delta u_m^i \end{bmatrix} + \Delta d^i && (7) \\
&= \Delta u_1^i P_1 + \cdots + \Delta u_m^i P_m + \Delta d^i && (8) \\
&= \begin{bmatrix} \Delta u_1^i I_n & \cdots & \Delta u_m^i I_n \end{bmatrix} \begin{bmatrix} P_1 \\ \vdots \\ P_m \end{bmatrix} + \Delta d^i, \quad \text{for} \quad i = 1, \ldots, q. && (9)
\end{aligned}$$

Stacking these equations on f of each other gives

$$\underbrace{\begin{bmatrix} \Delta y^1 \\ \vdots \\ \Delta y^q \end{bmatrix}}_{b} = \underbrace{\begin{bmatrix} \Delta u_1^1 I_n & \cdots & \Delta u_m^1 I_n \\ \vdots & & \vdots \\ \Delta u_1^q I_n & \cdots & \Delta u_m^q I_n \end{bmatrix}}_{A} \underbrace{\begin{bmatrix} P_1 \\ \vdots \\ P_m \end{bmatrix}}_{x} + \underbrace{\begin{bmatrix} \Delta d^1 \\ \vdots \\ \Delta d^q \end{bmatrix}}_{e} \tag{10}$$

where the matrix A is $nq \times nm$, the vector x is $nm \times 1$, and the vectors b and e are $nq \times 1$. The process interactions matrix P can be fit from the data using least squares parameter estimation:

$$\min_P \sum_{i=1}^{q} \sum_{j=1}^{n} \left| \Delta d_j^i \right|^2 = \min_x \; \|e\|^2 = \min_x \; \|b - Ax\|^2, \tag{11}$$

where Δd_j^i is the ith element of the vector Δd^i and $\|e\| \equiv \sqrt{\sum_i e_i^2}$ is the Euclidean norm of e. The least squares estimate of P is calculated from

$$\hat{x} = \begin{bmatrix} \hat{P}_1 \\ \vdots \\ \hat{P}_m \end{bmatrix} = \left(A^\top A\right)^{-1} A^\top b; \quad \hat{P} = \left[\hat{P}_1 \,|\cdots|\, \hat{P}_m\right], \tag{12}$$

where $\hat{P}$ refers to the least squares optimal plant interactions matrix. The model based on the true process (2) is

$$y(t) = y(t-1) + \hat{P}\Delta u(t-1) \tag{13}$$

where the best estimate of the unmeasured disturbances $\Delta d(t)$ in (2) is zero since $d(t)$ has equal probability of increasing or decreasing at each time instance, and its values were assumed to be uncorrelated with time.

Model Correction aka Filter The objective for identifying a model was to predict the effect of the manipulated variables on the process output. The model predictive controller uses this model to compute adjustments that suppress disturbances and track setpoints.

Because the model will not perfectly describe the process, the predicted output from (13) is corrected using the measured output $y_{meas}(t)$:

$$y(t) = (1-\gamma)(y(t-1) + \hat{P}\Delta u(t-1)) + \gamma y_{meas}(t), \tag{14}$$

where $\gamma \in (0,1]$ is a tuning parameter. This equation allows one to compute the corrected output $y(t)$ based on its previous value $y(t-1)$, the previous input move $\Delta u(t-1)$, and the current measurement $y_{meas}(t)$. The corrected output can be initialized with $y(0) = y_{meas}(0)$.

The larger the measurement noise and model uncertainty, the smaller the tuning parameter γ should be. The filter parameter γ is directly related to the desired closed-loop time constant, as used in internal model control or direct synthesis [1, 4]:

$$\tau_{cl} = \frac{\Delta t}{-\ln(1-\gamma)}, \tag{15}$$

or

$$\gamma = 1 - e^{-\Delta t/\tau_{cl}}, \tag{16}$$

where Δt is the time between sampling instances (see section on filtering in the laboratory notebook). A common tuning strategy is to select the desired closed-loop time constant τ_{cl} as small as you think the measurement noise and model uncertainty will allow, implement the controller on the real process, and then retune γ on-line if necessary (e.g., to reduce overshoots).

Prediction In order for the control algorithm to determine the optimal current manipulated variables, there has to be a means for *predicting* the effect of the manipulated variables on the future process outputs y. The predictor is given by writing (13) for the next time step $t+1$:

$$y_p(t+1) = y(t) + \hat{P}\Delta u(t). \tag{17}$$

Unconstrained Control Algorithm The performance criterion is to minimize the Euclidean norm of the difference between the setpoint signal $r(t+1)$ and the predicted process output $y_p(t+1)$. The control problem is expressed as an optimization by combining this objective with the predictor (17):

$$\min_{\Delta u(t)} \|r(t+1) - (y(t) + \hat{P}\Delta u(t))\|^2 \tag{18}$$

where $y(t)$ has been updated at each time instance, and so is known, and the model $\hat{P}$ for the process was previously determined off-line in the process identification step.

If $\hat{P}$ is assumed to have full column rank, then the least-squares solution to the unconstrained control problem is [1]:

$$\Delta u(t) = (\hat{P}^\top \hat{P})^{-1}\hat{P}^\top (r(t+1) - y(t)). \tag{19}$$

The control move $u(t)$ is implemented at time t, then the computer moves on to the next sampling instance where new measurements are taken and a new control move is computed and implemented.

Process Constraints All real world control systems must deal with process constraints. The control system must avoid unsafe operating regimes. In industrial process systems these constraints typically appear in the form of pressure or temperature limits. Further constraints are imposed by physical limitations—valves can only operate between fully open and fully closed, pumps and compressors have finite throughput capacity, surge tanks can only hold a certain volume. Each manipulated variable is subject to lower and upper limits:

$$u_{j,min} \le u_j \le u_{j,max}, \tag{20}$$

which is written in short hand form as

$$u_{min} \le u \le u_{max}. \tag{21}$$

Another type of manipulated variable constraint limits the speed of response of the actuators, for example, valves can open and close at only a finite speed. These constraints are written as

$$(\Delta u)_{min} \le \Delta u \le (\Delta u)_{max}, \tag{22}$$

where $(\Delta u)_{min}$ and $(\Delta u)_{max}$ are vectors specified by the control engineer. The bound on the maximum rate of variation in each manipulated variable is usually the same whether the manipulated variable is increasing or decreasing, so usually $(\Delta u)_{min} = -(\Delta u)_{max}$.

Additional constraints on the outputs or on the states of the process such as pressure and temperature are represented and addressed in a similar manner.

Handling Actuator Constraints Here two methods are discussed for handling actuator constraints most often implemented in industry.

The most direct approach is to explicitly include the constraints in the control algorithm. Then the constrained control problem will be the control objective (18) plus the additional constraints (21) and (22):

$$\min_{\substack{u_{min} \leq u(t) \leq u_{max} \\ (\Delta u)_{min} \leq \Delta u \leq (\Delta u)_{max}}} \|r(t+1) - (y(t) + \hat{P}\Delta u(t))\|^2. \tag{23}$$

This optimization problem can be solved with commercial software packages (for example, [8, 9]).

Another approach to handle the actuator constraints is to include a penalty function in the objective and solve

$$\min_{\Delta u} \|r(t+1) - (y(t) + \hat{P}\Delta u(t))\|^2 + \beta\|\Delta u(t)\|^2. \tag{24}$$

This gives the control move

$$\Delta u(t) = (\hat{P}^\top \hat{P} + \beta I)^{-1}\hat{P}^\top(r(t+1) - y(t)). \tag{25}$$

The disadvantage of this approach is that the added weighted term *always* affect the control action. The weights for these terms must be large enough to keep the control action feasible for large disturbances, but large weights on the control action will substantially slow the control action when the disturbances are small and the extra terms are not needed.

Simulation Example A process model for a pilot-plant adhesive coater is [10]

$$P = k \underbrace{\begin{pmatrix} 0.23 & -0.02 & -0.02 & \cdots & -0.02 \\ -0.02 & 0.23 & -0.02 & \ddots & \vdots \\ -0.02 & \ddots & \ddots & \ddots & -0.02 \\ \vdots & \ddots & \ddots & 0.23 & -0.02 \\ -0.02 & \cdots & -0.02 & -0.02 & 0.23 \end{pmatrix}}_{12 \times 12} \tag{26}$$

where $k = 1$, the process outputs are coating thicknesses, and the process inputs are the valve positions across the width of the coater. Assume that a disturbance occurs at $t = 0$:

$$d = \begin{pmatrix} 0.187 \\ -0.284 \\ -0.381 \\ -0.013 \\ 0.182 \\ 0.016 \\ -0.027 \\ 0.072 \\ 0.041 \\ -0.043 \\ 0.056 \\ 0.194 \end{pmatrix}; \tag{27}$$

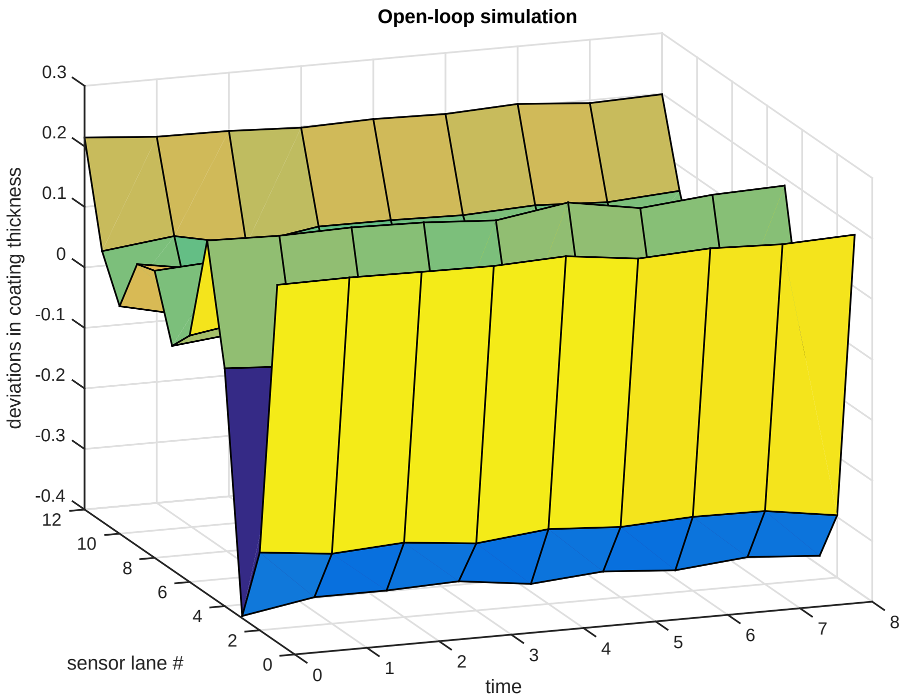

Figure 1: Open-loop response to disturbance d.

and that each coating thickness measurement has random noise with variance 0.000046 at each sampling instance. The sampling time $\Delta t = 1$ min.

Problem: Implement and tune an unconstrained MPC algorithm assuming that there is no model error ($k = 1$). How well does your best tuned MPC algorithm suppress the disturbance if the true process gain $k = 2$ or $k = 0.5$? If necessary, retune the MPC algorithm to give acceptable performance for the whole range of gain uncertainty.

Solution: The open-loop response to the disturbance d is in Figure 1.

For the unconstrained MPC algorithm presented here, the only tuning parameter is γ, which is related to the desired closed-loop time constant τ_{cl} by (15). It seems unlikely that the process model is sufficiently accurate to control the closed-loop system faster than three sampling times, so set $\tau_{cl} = 3$ min. Then the MPC tuning parameter is given by

$$\gamma = 1 - e^{-\Delta t/\tau_{cl}} = 0.2835. \tag{28}$$

The closed-loop response to the disturbance when there is no model error is in Figure 2, with the manipulated variables plotted in Figure 3. Disturbance suppression is speedy, with no overshoot. The coating thicknesses do not reach zero due to the measurement noise. The closed-loop responses when there are errors in model gain are shown in Figures 4 and 5. The closed-loop response is sluggish when the process gain is overestimated and has significant overshoots when the true process gain is underestimated. Select a smaller γ gives a less aggressive control algorithm, which would result in smaller overshoots in Figure 4, but with more sluggish response in Figure 5.

Summary Model predictive control has become the dominant multivariable control technique in the chemical and petrochemical industries. Its main advantage over other techniques is the ease for explicitly addressing process constraints within the control algorithm. A simple process model was used here to illustrate the basic principles of model predictive control. Readers interested in more advanced descriptions of model predictive control which are applicable to processes with more general dynamics and utilize more complex objective functions are referred to [1, 4].

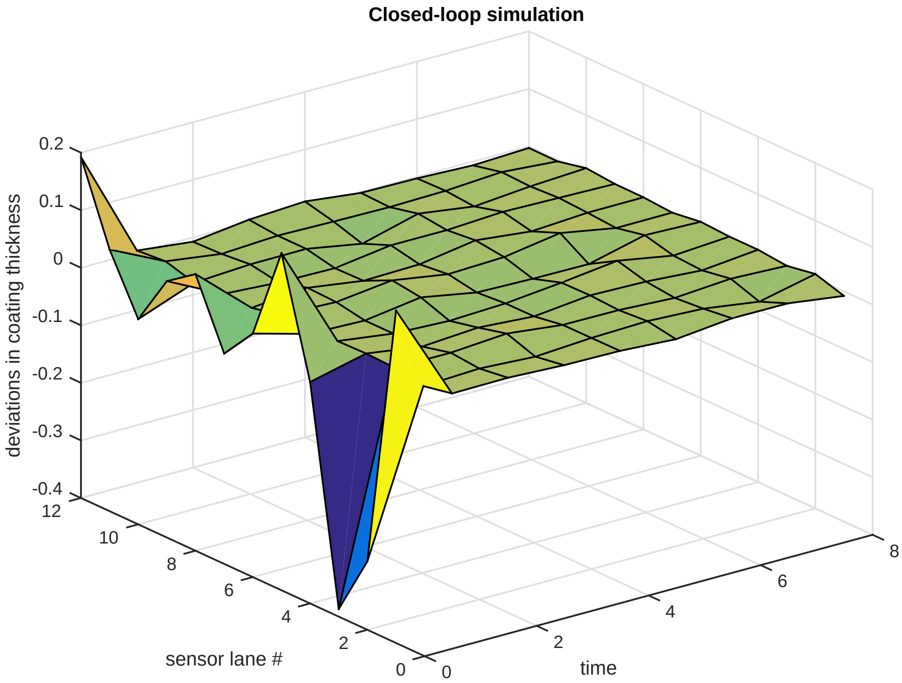

Figure 2: Closed-loop response to disturbance d (with $k = 1$) for unconstrained MPC.

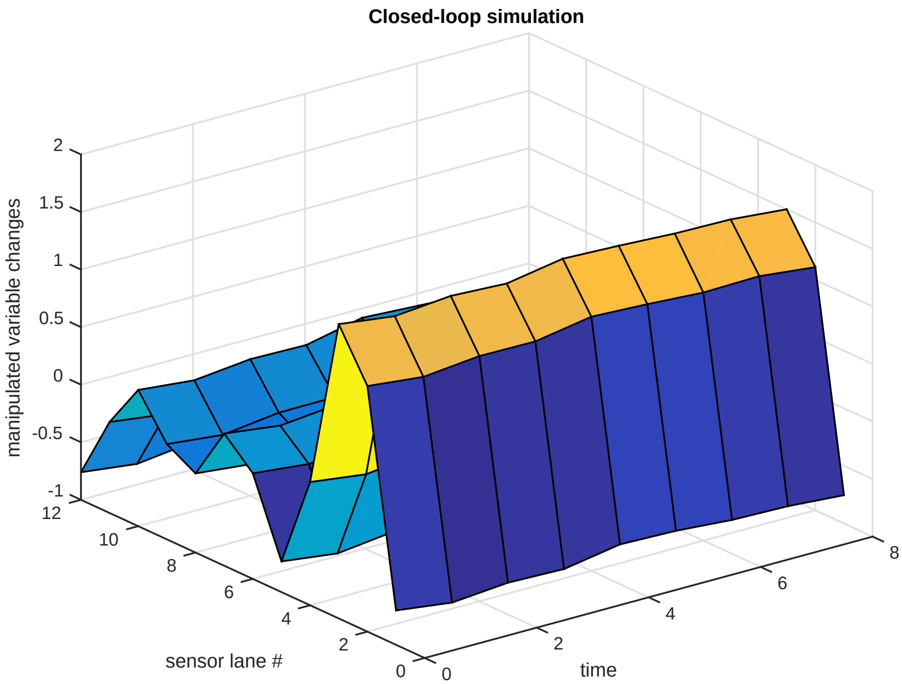

Figure 3: Manipulated variable response to disturbance d (with $k = 1$) for unconstrained MPC.

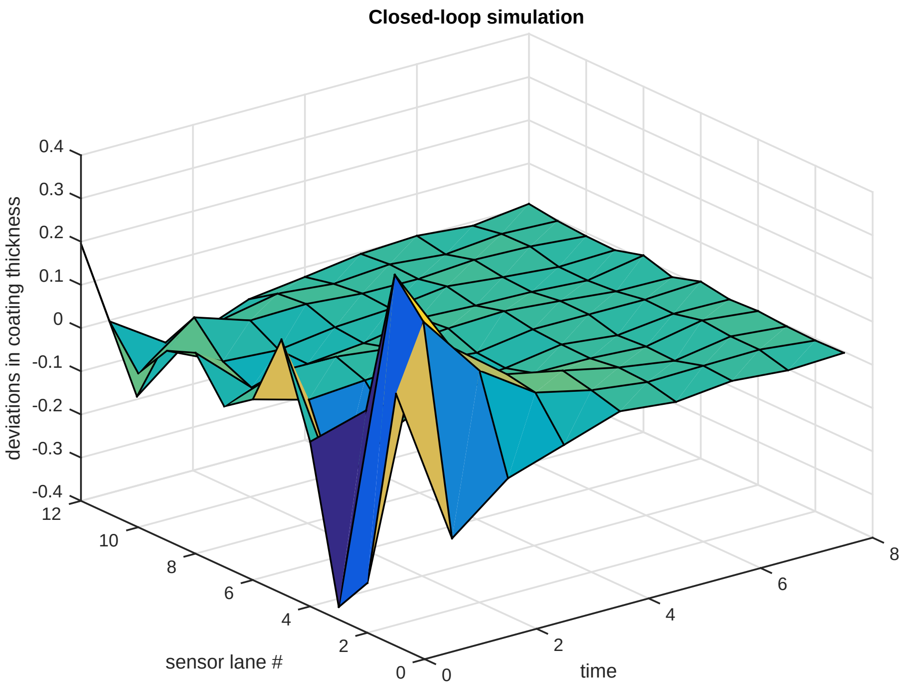

Figure 4: Closed-loop response to disturbance d (with $k = 2$) for unconstrained MPC.

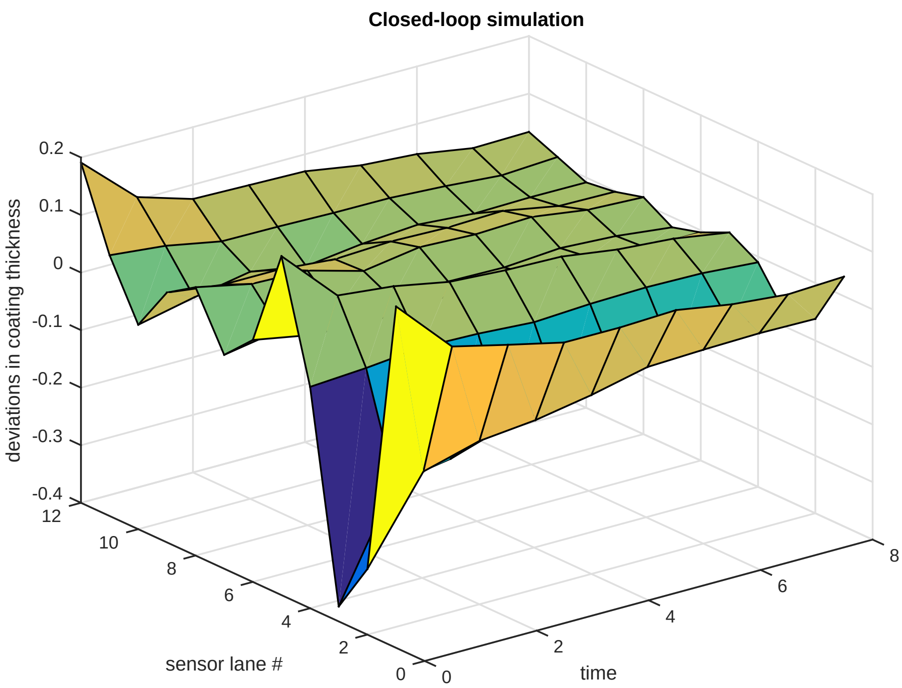

Figure 5: Closed-loop response to disturbance d (with $k = 0.5$) for unconstrained MPC.

Problems Five problems are presented in order from least to most challenging.

1. Using the process model (26) and disturbance (27), tune an unconstrained MPC controller to give acceptable performance for the process model and for interactions matrices with $k = 0.5$ and $k = 2.0$. Quantify the closed-loop performance in all three cases. Which plant limits the closed-loop speed of response? Why?

2. Industrial MPC algorithms have many more tuning parameters than the algorithms described here. One of these tuning parameters is a matrix W, usually diagonal, used to weigh the relative importance of each output in the vector y. In this case, the control objective for the unconstrained MPC problem is:

$$\min_{\Delta u(t)} \|W(r(t+1) - (y(t) + \hat{P}\Delta u(t)))\|^2 \tag{29}$$

Derive an equation for $\Delta u(t)$ that optimizes this objective. Hint: One way to do this is to use matrix algebra, set the derivatives of the objective with respect to each element of $\Delta u(t)$ to zero, and solve for $\Delta u(t)$. A simpler approach is to redefine the process output and the plant interactions matrix and use (19).

3. Select a random disturbance and try to design one unconstrained MPC controller to give acceptable disturbance suppression for the two plants:

$$P = \underbrace{\begin{pmatrix} 0.23 & -0.02 & -0.02 & \cdots & -0.02 \\ -0.02 & 0.23 & -0.02 & \ddots & \vdots \\ -0.02 & \ddots & \ddots & \ddots & -0.02 \\ \vdots & \ddots & \ddots & 0.23 & -0.02 \\ -0.02 & \cdots & -0.02 & -0.02 & 0.23 \end{pmatrix}}_{12 \times 12} \tag{30}$$

$$P = \underbrace{\begin{pmatrix} 0.23 & -0.03 & -0.03 & \cdots & -0.03 \\ -0.03 & 0.23 & -0.03 & \ddots & \vdots \\ -0.03 & \ddots & \ddots & \ddots & -0.03 \\ \vdots & \ddots & \ddots & 0.23 & -0.03 \\ -0.03 & \cdots & -0.03 & -0.03 & 0.23 \end{pmatrix}}_{12 \times 12} \tag{31}$$

Report plots for a wide range of tuning parameter γ. What is causing the strange results? Hint: Calculate the eigenvalues of both plants. Is the control algorithm providing negative feedback for both plants? Also calculate the singular values of both plants. Do the magnitudes of the singular values provide any insight?

4. Consider the simulation example, but include actuator constraints:

$$-1 \leq u_j \leq 1. \tag{32}$$

Answer all questions posed in the simulation example using the penalty function approach to ensure that the actuator constraints are satisfied during the closed-loop response to the disturbance at $t = 0$. Try a variety of control weightings β and filter parameter γ. Select values of β and γ and justify your selection. How would the best value for β change if your disturbance was three times larger than in (27). Three times smaller?

5. Consider the simulation example, but include the actuator constraints (32). Answer all questions posed in the simulation example using the constrained model predictive control approach to handle the actuator constraints. Note: to solve this problem using MATLAB, the student should have access to the quadprog.m program in the optimization toolbox.

References

[1] B. A. Ogunnaike and W. H. Ray. *Process Dynamics, Modeling, and Control*. Oxford University Press, New York, 1994.

[2] M. Morari and J. H. Lee. Model predictive control: The good, the bad, and the ugly. In W. H. Ray and Y. Arkun, editors, *Proceedings of the Fifth International Conference on Chemical Process Control (CPC-IV)*, pages 419–444. Elsevier, 1991.

[3] K. R. Muske and J. B. Rawlings. Model predictive control with linear models. *AIChE J.*, 39:262–287, 1993.

[4] D. E. Seborg, T. F. Edgar, D. A. Mellichamp, and F. J. Doyle III. *Process Dynamics and Control*. John Wiley & Sons, New York, third edition, 2010.

[5] A. P. Featherstone and R. D. Braatz. Integrated robust identification and control of large scale processes. *Ind. Eng. Chem. Res.*, 37:97–106, 1998.

[6] C. W. Koung and J. F. MacGregor. Design of identification experiments for robust control – A geometric approach for bivariate processes. *Ind. Eng. Chem. Res.*, 32:1658–1666, 1993.

[7] W. Li and J. H. Lee. Control relevant identification of ill-conditioned systems: Estimation of gain directionality. *Comp. & Chem. Eng.*, 20:1023, 1996.

[8] IMSL. Visual Numerics, Inc., 1997. computer software.

[9] Gillman and Murray. QPSOL, Stanford University, Office of Technology Licensing, 857 Serra Street, Stanford, California, 94305-6225, 1995. computer software.

[10] R. D. Braatz, M. L. Tyler, M. Morari, F. R. Pranckh, and L. Sartor. Identification and cross-directional control of coating processes. *AIChE J.*, 38:1329–1339, 1992.