An Integrated Framework for Scheduling and Control Using Fast Model Predictive Control

Jinjun Zhuge and Marianthi G. Ierapetritou
Dept. of Chemical and Biochemical Engineering, Rutgers – The State University of New Jersey, 98 Brett Road, Piscataway, NJ 08854

DOI 10.1002/aic.14914
Published online July 14, 2015 in Wiley Online Library (wileyonlinelibrary.com)

Integration of scheduling and control involves extensive information exchange and simultaneous decision making in industrial practice (Engell and Harjunkoski, Comput Chem Eng. 2012;47:121–133; Baldea and Harjunkoski I, Comput Chem Eng. 2014;71:377–390). Modeling the integration of scheduling and dynamic optimization (DO) at control level using mathematical programming results in a Mixed Integer Dynamic Optimization which is computationally expensive (Flores-Tlacuahuac and Grossmann, Ind Eng Chem Res. 2006;45(20):6698–6712). In this study, we propose a framework for the integration of scheduling and control to reduce the model complexity and computation time. We identify a piece-wise affine model from the first principle model and integrate it with the scheduling level leading to a new integration. At the control level, we use fast Model Predictive Control (fast MPC) to track a dynamic reference. Fast MPC also overcomes the increasing dimensionality of multiparametric MPC in our previous study (Zhuge and Ierapetritou, AIChE J. 2014;60(9):3169–3183). Results of CSTR case studies prove that the proposed approach reduces the computing time by at least two orders of magnitude compared to the integrated solution using mp-MPC. © 2015 American Institute of Chemical Engineers *AIChE J*, 61: 3304–3319, 2015

Keywords: integration of scheduling and control, piece-wise affine approximation, fast model predictive control, Multiparametric model predictive control, mixed integer nonlinear programming

Introduction

In chemical process operations simultaneous consideration of scheduling and control leads to and increases in the overall process profitability but meanwhile increases the complexity of modeling and thus the computational requirements of the solution approaches.[1,2] Traditionally, production scheduling and process control problems are studied separately in chemical processes. Scheduling problems address unit assignments, production sequence, timing of tasks but it does not consider the dynamic behavior of units such as reactors and distillation columns. Conversely, control problems focus on the dynamic behavior, such as the transitions between products in continuous processes (continuous stirred-tank reactor (CSTR) or plug flow reactor (PFR)) and the transient characteristics in batch processes. However, scheduling and control are naturally linked and the process operations at scheduling and control levels require data exchange between them.[1] For example, scheduling level transfers batch sizes, start times, and state references to the control level, and the control level provides measured states to the scheduling level. In the models of scheduling and control problems, there are variables that connect them such as the state variables at the starting and ending of the transient periods, durations for the transient periods and amounts of material being processed in units. Therefore, an integrated approach that models the scheduling and control simultaneously addresses the information sharing between scheduling and control levels, leading to the integrated decision making that is overall optimal.[3,4]

In the literature, two approaches are proposed to model the integration of scheduling and control, that is, simultaneous modeling and decoupled modeling. Using simultaneous modeling, the process dynamics are incorporated into the scheduling problems as constraints. This leads to a Mixed Integer Dynamic Optimization (MIDO) problem[5] which is further discretized into a Mixed Integer Nonlinear Programming (MINLP).[6] With decoupled modeling, the scheduling problem (master problem) is modeled as Mixed Integer Linear Programming (MILP) and the control problem (primal problem) as Dynamic Optimization (DO). This complete problem is solved through iterations between the master problem and the primal problem.[7,8] Case studies of polymerization process show that the solution obtained through iterations is close to the global optima.

Online integration of scheduling and control requires updating operation solutions for both scheduling and control levels at real time in the presence of disturbance, thus the online integration requires a repetitive solution of the integrated problem at each time interval.[9] To reduce the computation complexity of the integrated problem, researchers proposed to use Lagrangean Decomposition[10] and Bender Decomposition[11–13] to converge to the optimal solution. Chu and You[14] models the integrated problem using game theory and efficiently solve the resulting bilevel optimization problem. Zhuge and

Correspondence concerning this article should be addressed to M. G. Ierapetritou marianth@soemail.rutgers.edu.

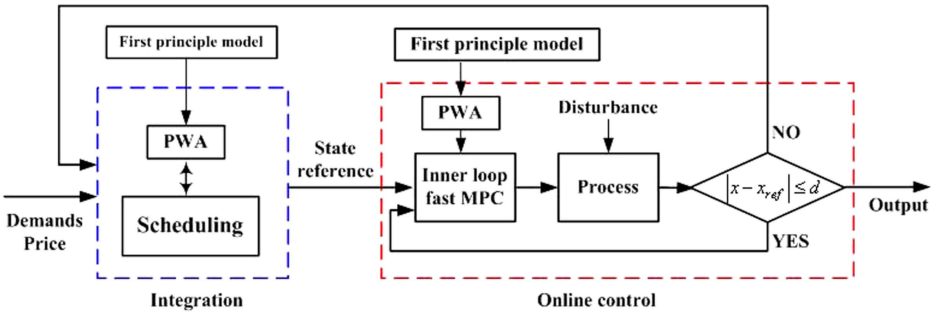

Figure 1. Online integration of scheduling and control.

[Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

Ierapetritou[15] utilizes multiparametric Model Predictive Control (mp-MPC) to handle the control problem and incorporation the explicit control solution with the scheduling level, resulting in a MILP which simplifies the integrated problem.

Model Predictive Control (MPC) is an online optimization technique that involves repetitively solution of an optimization problem over a future time horizon. To reduce the computation burden of conventional MPC, mp-MPC was proposed to solve for the explicit control law offline and thus the online optimization is reduced to simple function evaluations.[16] However, as the problem size increases in terms of state dimension and prediction horizon, the number of polyhedral regions in the state partition increases exponentially and this limits the applicability of mp-MPC to small and medium-sized control problems.[17]

Fast MPC, conversely, is capable in handling large scale problems, and therefore can be used to facilitate the efficient integration of scheduling and control of large-scale processes. Fast MPC for linear systems transforms the MPC problem into a convex quadratic programming problem, for which efficient nonlinear programming methods and computational tools can be used to speed up the computation by exploring the problem structure. Among the existing work in the literature, three solution approaches can be identified: active set method,[18] interior point method,[19,20] and Fast gradient method.[17,21] These methods are specifically elaborated in section "The role of fast MPC." Zavala et al.[22] developed algorithms for fast nonlinear MPC that is based on sensitivity calculation for the Karush–Kuhn–Tucker (KKT) conditions of the NLP derived from differential and algebraic equations (DAE) of the nonlinear MPC. This approach was further investigated and applied in large scale industrial processes in Lopez-Negrete et al.[23]

In this study, we propose an integrated framework that involves two control loops for the online integration of scheduling and control (Figure 1). In the outer loop, we approximate the original process dynamics using a piece-wise affine (PWA) model and incorporate it with the scheduling level. This leads to an integrated problem that is subject to linear constraints. The integrated problem at the outer loop generates both the production scheduling and the state reference for control problem. The scheduling solution and state reference are transferred to the inner loop where the fast MPC tracks the state reference and computes the exact control solution online. Note that these two loops correspond to different models and different time scales. The outer loop uses the integrated model and solves it in the period of days or weeks while the inner loop uses the process dynamic model and updates the solution in seconds. Essentially the inner loop is solved much more frequently than the outer loop does. Following this approach, the outer loop is able to achieve an overall optimality for both scheduling and control levels efficiently, and the inner loop fast MPC is able to respond quickly to the local disturbance. Note that the proposed approach is applicable in a dynamic market environment as the outer loop incorporates the market information such as demand and price. The proposed framework solves an integrated model to guarantee the overall optimality and uses fast MPC to respond timely to process disturbances. While solving the integrated problem the latest market information such as material price and product demands are incorporated into the integrated problem, and the scheduling solutions are updated accordingly.

In this study, we assume that the state deviation is caused by random disturbance on the state or manipulated variables rather than systematic failure. When disturbance is detected (state deviation from the referece is measured), the state information is feedback to the inner fast MPC or outer integrated problem. A threshold is introduced to determine the feedback. If the state deviation is less than the threshold, the state is feedback to inner fast MPC and locally treated by fast MPC. Otherwise if the state deviation is large (higher than the threshold) it is feedback to the integrated problem and the scheduling solution is updated accordingly. The threshold is an empirical value and it is determined to avoid the unecessary changes of the scheduling solution when disturbances are small enough that can be handled efficiently in the control level, whereas updates in the scheduling solution are considered if significant disturbance occurs. In the following, we present an overview of the steps of implementing the proposed framework in Figure 1.

Overview of steps of implementing the integrated framework in Figure 1

Step 1: Transform the nonlinear first-principle model into PWA. The detailed approach of PWA approximation is proposed in sectioin "A PWA identification technique using optimization methods" (i.e., problem (7)).

Step 2: Integrate the obtained PWA with the scheduling level and form an integrated problem. The constraints that incorporate PWA into scheduling level are built in Section "Integrated problem incorporating PWA systems" (i.e., Eqs. 8–38, problem (43)).

Step 3: Solve the integrated problem for a future horizon (i.e., fixed or flexible production cycle), and obtain scheduling and the state reference (dynamic profiles during transitions).

Step 4: Transfer the scheduling solution and state reference to the inner loop which is handled by fast MPC.

Step 5: Solve for fast MPC based on the PWA model obtained in step 1. Fast MPC tracks the state reference and generates the online control solution. The detailed algorithm of fast MPC is proposed in Section "Fast MPC for PWA systems" (i.e., Eqs. 46–50).

Step 6: Input the control solution to the process and measure the states. If the error between measured state and state reference is smaller than the threshold, go to the next sample step and go to step 5 (i.e., stay in the inner loop). If the error is greater than the threshold, go to step 3 (i.e., go to the outer loop).

The rest of the article is organized as follows. In Section "Integration of scheduling and control based on PWA model" we review the PWA model identification techniques and propose a PWA identification technique to approximate nonlinear dynamics using an optimization method. Then we model the integration of scheduling and control incorporating the PWA approximation model of process dynamics. In Section "Fast MPC for PWA systems," we propose fast MPC strategy for PWA model where we design the detailed algorithm and analyze the stability of the control system. To test the proposed integrated model as well as fast MPC strategy, we study two case studies in Section "Case studies," one for SISO CSTR and the other for MIMO CSTR. Finally, conclusions of this study are made.

Integration of Scheduling and Control Based on PWA Model

The main challenge of integration of scheduling and control for chemical processes is that the integrated problem cannot be solved as fast as it is required for the control level but also that the scheduling solution should not be adjusted all the time. As scheduling and control levels correspond to different dynamics and different time scales, an appropriate integration has to capture the process behavior well enough at both scheduling and control levels, while maintaining the appropriate level of integration.[1] Moreover, the integrated problem should not be very computationally expensive thus the integrated solution could be updated timely when disturbance occurs. Typically an integration results in a MIDO which is discretized into a MINLP using collocation point method[6,24–26] or implicit RK4 method.[9] However, the nonlinearity in MINLP poses significant difficulty in the computation method of solving the integrated optimization problem.

In control practice, a standard way to handle the nonlinear dynamics is to linearize the nonlinear model around operating points. However, the linearized model is valid only if the process operates at the vicinity of operating points. It is just an approximation of the actual process dynamics when process experiences transitions where the states are far from the operation points (e.g., steady-state points). In contrast, PWA systems have shown to be an effective approach in dealing with nonlinear systems.[27–29] The basic idea of PWA system is that the nonlinear dynamics can be approximated by a collection of distinct linear (or affine) dynamic approximations with associated regions of validity. Compared to standard linear model, PWA composes a group of linear models and therefore it is capable to address the process dynamics at the entire state domain. PWA models eliminate the nonlinearity while retaining high approximation accuracy.

A brief review of the PWA identification techniques

There are two types of techniques regarding PWA model identification. One is clustering technique where PWA models are obtained by processing K-means algorithm on input-output data. Ferrari-Trecate et al.[30] combined use of clustering, linear identification, and pattern recognition techniques to identify both the affine submodels and the polyhedral partition of the domain. Magnani and Boyd[31] proposed a heuristic method for fitting a convex piecewise linear function to a given set of data. It uses K-means algorithm for clustering, and requires the piecewise linear function to be convex. Researchers[32–37] use K-means clustering technique or variations derived from that. The downside of this approach is that there is no guarantee that the union of regions obtained by clustering is able to cover the whole area of the original domain without gaps where the model is actually defined.

The other type of PWA identification technique is optimization based PWA approximation. This technique applies to the cases where the original nonlinear functions are available. Števek et al.[38] proposed a two-stage optimization-based approach. At the first stage, they fit the input/output data using multivariable nonlinear functions that are represented as a sum of products of functions in single variables. They use neural networks with predefined basis functions to obtain the nonlinear functions. At the second stage they obtain the PWA approximation for the nonlinear functions obtained in the first stage using the approaches in Kvasnica et al.[39] Kvasnica et al.[39] handles one-dimensional (1-D) functions. They treat the partition of the domain and the corresponding linear model as decision variables and build an optimization problem (1) to minimize the error between the PWA and the original functions. For example, PWA approximation $\tilde{f}(x)$ for a nonlinear function $y=f(x)=x^3$ is obtained through solving problem (1) where A and B are coefficients in PWA and r is the partition of the x domain. Note that continuity is enforced by the third constraint. The solutions (PWA $\tilde{f}(x)=A_ix+B_i$) are presented in formula (2) and Figure 2

$$\min_{A_i,B_i,r_i} \int_{\underline{x}}^{\bar{x}} (f(x)-\tilde{f}(x))^2 dx$$

$$s.t. \begin{cases} \tilde{f}(x)=A_ix+B_i \quad if\ x\in[r_{i-1},r_i] \\ \underline{x}=r_1\le\cdots\le r_{N-1}=\bar{x} \\ A_ir_i+B_i=A_{i+1}r_i+B_{i+1} \end{cases} \tag{1}$$

$$\tilde{f}(x)=A_ix+B_i=\begin{cases} 7.431x+7.495, & -2\le x<-1.123 \\ 0.757x, & -1.123\le x<1.123 \\ 7.431x-7.459, & 1.123\le x<2 \end{cases} \tag{2}$$

For two and multiple dimensional functions, Kvasnica et al.[39] investigated two particular cases

Case 1. If the function $f(x_1,x_2,\ldots,x_n)$ is separable, that is, $f(x_1,x_2,\ldots,x_n)=f_1(x_1)+f_2(x_2)+\cdots+f_n(x_n)$, then apply 1-D approximation for each term.

Case 2. If the function $f(x_1,x_2,\ldots,x_n)$ is in a form of product that is, $f(x_1,x_2,\ldots,x_n)=f_1(x_1)f_2(x_2)\cdots f_n(x_n)$, then introduce new variables to transform the product into a summation form in case 1. For example, let $y_1=x_1+x_2$ and $y_2=x_1-x_2$, we have $x_1x_2=\left(y_1^2-y_2^2\right)/4$.

However, for general multiple dimension functions that cannot be put into the form of case 1 and 2, Kvasnica et al.[39] did

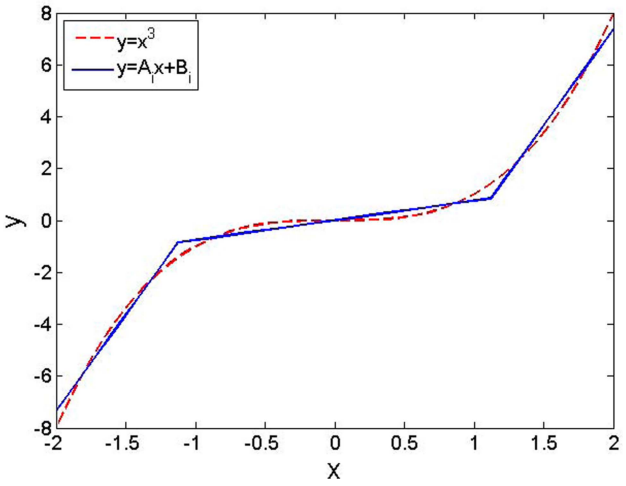

Figure 2. Example of 1-D PWA approximation.

[Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

not provide any approximation approaches. Julian et al.[40] explored a uniform triangular-shape partition for the domain and approximated a nonlinear function by adjusting the grid size of the partition. Zavieh and Rodrigues[41] used heuristic selection of linearization points where a new linearization point is determined by the intersection of tangential lines associated with the old linearization points. Casselman and Rodrigues[42] used a set of linearization points and Voronoi partitions in the approximation. Sontag[43] discussed the stability, controllability, and observability in PWA systems. However, the existing work in PWA identification cannot handle a general form of nonlinear model.

A PWA identification technique using optimization methods

To overcome the shortcomings of the optimization based PWA identification described in previous section, we propose a PWA identification technique that is able to handle general forms of nonlinear function. In two-dimensional (2-D) cases we use rectangular partition of the domain of function $f(x^{(1)}, x^{(2)})$ (Figure 3). Index $i \in I=\{1,2,...,N_i\}$ is associated with $x^{(1)}$, and $j \in J=\{1,2,...,N_j\}$ is associated with $x^{(2)}$. Intermediate points $x^{(1)}_{\text{int},i}$ and $x^{(2)}_{\text{int},j}$ are introduced as partitions for $x^{(1)}$ and $x^{(2)}$, respectively. Therefore, we have constraints: $x^{(1)}_{\text{int},0} \le x^{(1)}_{\text{int},1} \le \cdots \le x^{(1)}_{\text{int},N_i}$ and $x^{(2)}_{\text{int},0} \le x^{(2)}_{\text{int},1} \le \cdots \le x^{(2)}_{\text{int},N_j}$, where $x^{(1)}_{\text{int},0}$ and $x^{(2)}_{\text{int},0}$ are the lower bounds of $x^{(1)}$ and $x^{(2)}$ and $x^{(1)}_{\text{int},N_i}$ and $x^{(2)}_{\text{int},N_j}$ are the upper bounds. N_i and N_j are the number of intermediate points for $x^{(1)}$ and $x^{(2)}$. The subindex "int" represents "intermediate points."

Define the Valid Regions (VR) and the PWA.

$$\text{VR}_{i,j}=\{(x^{(1)},x^{(2)})|x^{(1)}_{\text{int},i-1} \le x^{(1)} \le x^{(1)}_{\text{int},i}, x^{(2)}_{\text{int},j-1} \le x^{(2)} \le x^{(2)}_{\text{int},j}\} \tag{3}$$

$$\hat{f}_{i,j}(x^{(1)},x^{(2)})=a_{i,j}x^{(1)}+b_{i,j}x^{(2)}+c_{i,j}, \text{if}(x^{(1)},x^{(2)}) \in \text{VR}_{i,j} \tag{4}$$

Equation 3 determines that point $(x^{(1)}, x^{(2)})$ belongs to a certain valid region and Eq. 4 provides the Linear Time Invariant (LTI) associated with the regions.

Constraints. The profile of PWA should be continuous at the boundaries of valid regions. In other words, the function values of adjacent LTIs are the same at the boundaries. Note that if the continuity at the intersection points is established, then the continuity at the segment defined by the intersection points is established as well. For example, at point A in Figure 3, the LTIs of the four adjacent valid regions should have the same value. This gives rise to the constraints in Eq. 5

$$\begin{aligned}&\hat{f}_{i,j}(x^{(1)}_{\text{int},i},x^{(2)}_{\text{int},j})=\hat{f}_{i+1,j}(x^{(1)}_{\text{int},i},x^{(2)}_{\text{int},j})\\&=\hat{f}_{i,j+1}(x^{(1)}_{\text{int},i},x^{(2)}_{\text{int},j})=\hat{f}_{i+1,j+1}(x^{(1)}_{\text{int},i},x^{(2)}_{\text{int},j})\end{aligned} \tag{5}$$

More specifically

$$\begin{aligned}&a_{i,j}x^{(1)}_{\text{int},i}+b_{i,j}x^{(2)}_{\text{int},j}+c_{i,j}\\&=a_{i+1,j}x^{(1)}_{\text{int},i}+b_{i+1,j}x^{(2)}_{\text{int},j}+c_{i+1,j}\\&=a_{i,j+1}x^{(1)}_{\text{int},i}+b_{i,j+1}x^{(2)}_{\text{int},j}+c_{i,j+1}\\&=a_{i+1,j+1}x^{(1)}_{\text{int},i}+b_{i+1,j+1}x^{(2)}_{\text{int},j}+c_{i+1,j+1}\end{aligned} \tag{6}$$

Optimization Problem. The proposed PWA identification approach can be modeled as an optimization problem with decision variables that include the partition of the domain $x^{(1)}_{\text{int},i}, x^{(2)}_{\text{int},j}$ and the coefficients of LTI $a_{i,j}, b_{i,j}, c_{i,j}$, and constraints that specify the continuity constraints in (6). The objective of the optimization problem is to minimize the sum of squared error between the PWA and the original function as provided in problem (7)

$$\min_{x^{(1)}_{\text{int},i},x^{(2)}_{\text{int},j},a_{i,j},b_{i,j},c_{i,j}} \sum_i \sum_j \left(\hat{f}(x^{(1)}_{\text{int},i},x^{(2)}_{\text{int},j})-f(x^{(1)}_{\text{int},i},x^{(2)}_{\text{int},j})\right)^2 \quad s.t. \quad (3)\text{–}(6) \tag{7}$$

The optimization problem of identifying PWA from a three-dimensional (3-D) function is provided in Appendix.

The goodness of PWA approximation can be quantified by the sum of squared error as shown in problem (7). If more discretization points are used (Figure 3) the resulting PWA is more accurate. The optimal control problem at control level is an infinite dimensional optimization problem and the discretization using PWA transfers it into a finite dimensional optimization problem with linear constraints.

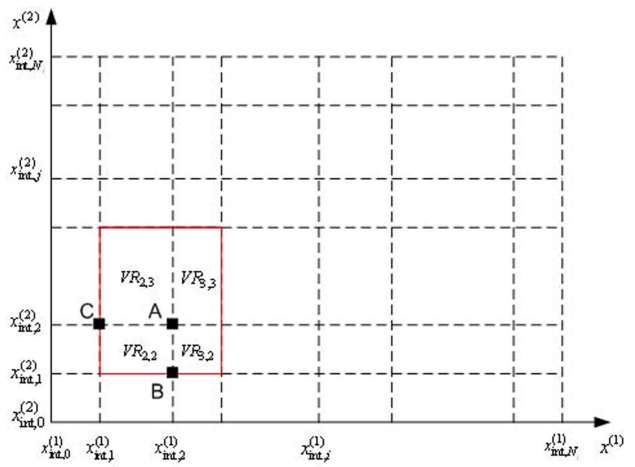

Figure 3. Rectangular partition of the domain of 2-D functions.

[Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

Complexity Analysis. The decision variables of problem (7) include the coefficients of PWA $\left(a_{i,j,k,\cdots}, b_{i,j,k,\cdots}, c_{i,j,k,\cdots}, d_{i,j,k,\cdots}, \cdots\right)$ and the intermediate points of each dimension $\left(x^{(1)}_{\text{int},i}, x^{(2)}_{\text{int},j}, x^{(3)}_{\text{int},k}, \cdots\right)$. If there is an additional dimension, there would be a new coefficient in PWA, a new dimension subindex for each coefficient and a new group of intermediate points. Therefore the number of coefficients of PWA and the number of intermediate points are increasing in the order of o$(N(N+1))$ and o(N), respectively, where N is the number of dimension. However, the number of constraints is increasing exponentially with respect to the number of dimensions, since the number of adjacent valid regions is increasing in the order of o(2^N) and the associated linear functions all have the same value at the intersection point (the continuity constraints). For example for a 2-D function, there are 4 continuity constraints and for a 3-D function, there are 8 continuity constraints (Eqs. 6 and A4).

Integrated problem incorporating PWA systems

In this study we address continuous cyclic production for which the scheduling constraints are adopted from the work of Flores-Tlacuahuac and Grossmann.[6] The constraints at control level are built via the PWA identification proposed in section "A PWA identification technique using optimization methods." The PWA model is incorporated into the scheduling constraints by introducing extra binary variables in a similar way as the previous work in mp-MPC.[15] Since the outer loop MPC has a fixed horizon, cycle time T_c is not a decision variable. Thus the objective is linear and the integrated problem is a MILP.

Constraints at Scheduling Level. ***Product assignment.***

$$\sum_{s=1}^{N_s} y_{p,s}=1,\ \forall p \in S_p \tag{8}$$

$$\sum_{p=1}^{N_p} y_{p,s}=1,\ \forall s \in S_s \tag{9}$$

where $y_{p,s}$ are binary variables indicating the assignment of each product in each slot. $y_{p,s}=1$ if product p is assigned to slot s, otherwise $y_{p,s}=0$. Equations 8 and 9 imply that only one product is produced in one slot and each product can only be produced in one slot.

Production and demands.

$$W_p=G_p\Theta_p,\ \forall p \in S_p \tag{10}$$

$$W_p \geq D_p,\ \forall p \in S_p \tag{11}$$

Inequality (11) implies that the production amount of each product W_p, which is a product of production rate G_p and production time Θ_p (Eq. 10), should satisfy the demand D_p in current production cycle.

Production time.

$$\Theta_{p,s} \leq \Theta_{\max} y_{p,s}, \quad \forall p \in S_p, s \in S_s \tag{12}$$

$$\Theta_p=\sum_{s=1}^{N_s} \Theta_{p,s},\ \forall P \in S_p \tag{13}$$

$$\Theta_s=\sum_{p=1}^{N_p} \Theta_{p,s},\ \forall s \in S_s \tag{14}$$

Inequality (12) implies that the production time $\Theta_{p,s}$ is less than the maximum duration allowed for producing product p in slot s. In Eqs. 13 and 14, Θ_p, Θ_s define the production time of product p and slot s, respectively.

Timing constraints.

$$t^s_1=0 \tag{15}$$

$$t^e_s=t^s_s+\Theta_s+\theta^t_s,\ \forall s \in S_s \tag{16}$$

$$t^s_s=t^e_{s-1},\ \forall s \in S_s \backslash \{1\} \tag{17}$$

$$t^e_s \leq T_c,\ \forall s \in S_s \tag{18}$$

Equation 15 initializes the starting time of the first slot. As described by (16), the ending point t^e_s is equal to the sum of the starting point t^s_s, the production Θ_s and transition times θ^t_s of the current slot. Equation 17 enforces that the starting point in slot s equals the ending point of the previous slot to ensure continuity between slots. Inequality (18) indicates that all slots should end before the end of the cycle.

Constraints at Control Level. ***Selection of LTI in PWA.*** The LTI described by Eq. 19 is valid when state $x_{s,k}$ is in the region of validity Ω_{xi}. We transform this implicit requirement into explicit constraints (Eqs. 20–23) by introducing binary variables $y1_{s,k,i}$. If $y1_{s,k,i}=1$, the region of validity Ω_{xi} is selected by (20) and thus the LTI associated with coefficients A_i, B_i, C_i is selected through the constraints (21) and (22). Equation 23 indicates that only one LTI is selected at each sample step

$$x_{s,k+1}=A_i x_{s,k}+B_i u_{s,k}+C_i,\ \text{if } x_{s,k} \in \Omega_{xi}=\{x : V_i x \leq W_i\} \tag{19}$$

$$-M(1-y1_{s,k,i})+V_i x_{s,k} \leq W_i,\ \forall s \in S_s, k \in S_k, i \in S_i \tag{20}$$

$$x_{s,k+1} \geq A_i x_{s,k}+B_i u_{s,k}+C_i-M(1-y1_{s,k,i}),\ \forall s \in S_s, i \in S_i, k \in (S_k-\{N_k\}) \tag{21}$$

$$x_{s,k+1} \leq A_i x_{s,k}+B_i u_{s,k}+C_i+M(1-y1_{s,k,i}),\ \forall s \in S_s, i \in S_i, k \in (S_k-\{N_k\}) \tag{22}$$

$$\sum_i y1_{s,k,i}=1,\ \forall s \in S_s, k \in S_k \tag{23}$$

Bounds for state and manipulated variables.

$$x_{\min} \leq x_{s,k} \leq x_{\max},\ \forall s \in S_s, k \in S_k \tag{24}$$

$$u_{\min} \leq u_{s,k} \leq u_{\max},\ \forall s \in S_s, k \in S_k \tag{25}$$

$$-\Delta_u \leq u_{s,k+1}-u_{s,k} \leq \Delta_u,\ \forall s \in S_s, k \in S_k \backslash \{N_k\} \tag{26}$$

Inequalities (24) confine the states using lower and upper bounds while the constraints (25) and (26) introduce lower/upper bounds for the manipulated variables and their increments, respectively.

Determination of the end of transitions and evaluation of the transition times. In the integration of scheduling and control for continuous processes, the control action applies to the transitions between products that are distinguished by their steady states. The time points and state values at the beginning and end of the transitions are important variables in connecting scheduling and control levels. In practice, the transitions naturally start from the steady state of the previous product but the end point and the duration of transition are unknown. Therefore in the integrated model we need to build constraints to determine the end of the transition and calculate the transition time accordingly.

As shown schematically in Figure 4, there is a need to determine whether the process states governed by the PWA model

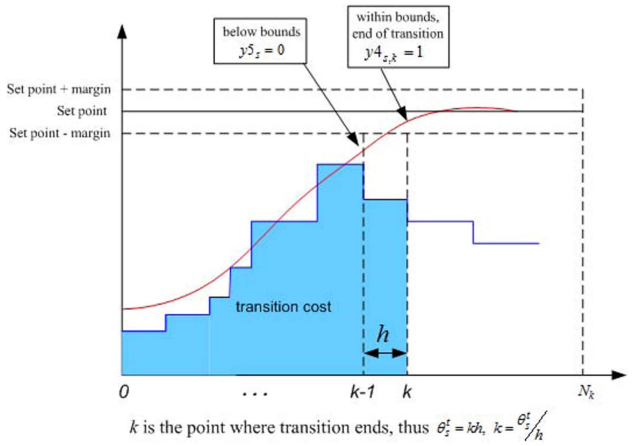

Figure 4. The end of the transition locates at the point inside the bounds with the prior one locating outside the bounds.

[Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

reach the end of transition or not. Thus in this section we develop the constraints required to enable this. Constraints (27)–(30) enforce that the transient states achieve the steady-state values when transitions end.

Due to the existence of some margins around the set point, if the state is between the lower bound $\bar{x}_s - x_{\text{margin}}$ and the upper bound $\bar{x}_s + x_{\text{margin}}$, it is viewed as meeting the set point (steady-state point $\bar{x}_s$). To determine the end of transition we need to identify the point that lies within the bounds while the previous point is outside the bounds. Two types of binary variables are involved in determining the end of transition $y4_{s,k}$ and $y5_s$. $y4_{s,k}$ indicates whether the end of transition state is reached or not while $y5_s$ indicate if the state right before the end of transition is above or below the bounds. More specifically, if $y4_{s,k}=1$, step k is selected as the end of transition and the state at step k is within the bounds. If the state at step $k-1$ is above the upper bound $\bar{x}_s + x_{\text{margin}}$, $y5_s=1$ whereas if it is below the lower bound $\bar{x}_s - x_{\text{margin}}$, $y5_s=0$. Conceptually, $y5_s=0$ means state increases to steady state during transition (as shown in Figure 4) and $y5_s=1$ means state decreases to steady state.

Inequality (27) is based on the definition of $y4_{s,k}$ and (28) ensures that only one of the step points is selected as the end of the transition

$$\bar{x}_s - x_{\text{margin}} - M(1-y4_{s,k}) \le x_{s,k} \le \bar{x}_s + x_{\text{margin}} + M(1-y4_{s,k}), \quad \forall s \in S_s, k \in S_k \tag{27}$$

$$\sum_k y4_{s,k} = 1, \ \forall s \in S_s \tag{28}$$

If $y4_{s,k}=0$, the end of transition is not reached and constraints (27), (29), and (30) are relaxed. If $y4_{s,k}=1$ step k is the end of transition, so constraint (27) enforces that the state at step k ($x_{s,k}$) must be within the quality bounds, and constraint (29) enforces that the state at step $k-1$ ($x_{s,k-1}$) must be below the lower bound $\bar{x}_s - x_{\text{margin}}$ or above the upper bound $\bar{x}_s + x_{\text{margin}}$. The transition time is determined as kh by constraints (30)

$$\bar{x}_s + x_{\text{margin}} - M(2-y4_{s,k}-y5_s) \le x_{s,k-1} \le \bar{x}_s - x_{\text{margin}} + M(1-y4_{s,k}+y5_s), \ \forall s \in S_s, k \in S_k \backslash \{1\} \tag{29}$$

$$kh - M(1-y4_{s,k}) \le \theta_s^t \le kh + M(1-y4_{s,k}), \ \forall s \in S_s, k \in S_k \tag{30}$$

Constraints That Link Scheduling and Control Level. ***Initial values and steady-state values are linked by $y_{p,s}$.***

$$x_{\text{in},s} = \sum_{p=1}^{N_p} x_{\text{ss},p} y_{p,s-1}, \ \forall s \in S_s \backslash \{1\} \tag{31}$$

$$x_{\text{in},1} = \sum_{p=1}^{N_p} x_{\text{ss},p} y_{p,N_s} \tag{32}$$

$$x_{s,1} = x_{\text{in},s}, \ \forall s \in S_s \tag{33}$$

Equations 31 and 32 calculate the initial state value at each slot. Equation 33 defines the initial state value $x_{\text{in},s}$ as the value of the first sample step

$$u_{\text{in},s} = \sum_{p=1}^{N_p} u_{\text{ss},p} y_{p,s-1}, \ \forall s \in S_s \backslash \{1\} \tag{34}$$

$$u_{\text{in},1} = \sum_{p=1}^{N_p} u_{\text{ss},p} y_{p,N_s} \tag{35}$$

$$u_{s,1} = u_{\text{in},s}, \ \forall s \in S_s \tag{36}$$

Equations 34–36 compute the initial value for manipulated variables following the similar way as shown in Eqs. 31–33. Note that $x_{s,1}$ and $u_{s,1}$ are also present in the dynamic PWA in Eq. 19. They are linked to the scheduling variables ($x_{\text{ss},p}$, $u_{\text{ss},p}$, $y_{p,s}$) through (31)–(36). Thus constraints (31)–(36) demonstrate how scheduling and control are linked.

The desired values and steady-state values are linked by $y_{p,s}$.

$$\bar{x}_s = \sum_{p=1}^{N_p} x_{\text{ss},p} y_{p,s}, \ \forall s \in S_s \tag{37}$$

$$\bar{u}_s = \sum_{p=1}^{N_p} u_{\text{ss},p} y_{p,s}, \ \forall s \in S_s \tag{38}$$

Eqs. 37 and 38 compute the desired state values and desired manipulated variables at each slot.

Optimization Problem. To achieve economically optimal operations of chemical processes, we utilize the objective of maximizing profit per unit time, which can be calculated as follows. Profit per unit time (Φ) = (Revenue – Raw material cost – Utility cost)/Cycle time

$$\Phi = \Phi_1 - \Phi_2 - \Phi_3 \tag{39}$$

$$\Phi_1 = \sum_{p=1}^{N_p} \frac{P_p W_p}{T_c} \tag{40}$$

$$\Phi_2 + \Phi_3 = \left(\sum_{s=1}^{N_s} \sum_{k=1}^{\theta_s^t/h_s} (P_r u_{1s,k} + P_u u_{2s,k}) h_s + \sum_{s=1}^{N_s} (P_r \bar{u}_{1s} + P_u \bar{u}_{2s}) \Theta_s \right) \frac{1}{T_c} \tag{41}$$

where Φ_1, Φ_2, and Φ_3 represent the revenue rate, raw material cost rate, and utility cost, respectively. Note that for continuous processes raw material cost and utility cost (heating/cooling) can be combined in (41) where u_1 represents raw material

feeding flow rate and u_2 represents utility flow rate. The total revenue is given as the amount produced (W_p) times the product price (P_r). Raw material cost is composed of two parts, the raw material consumption during production periods (the second term in the parenthesis of (41)) and during the transition periods (the first term in the parenthesis of (41)). Utility cost is calculated following the similar way of raw material cost. Note that the raw material consumed during the transition is the shaded area of Figure 4 which can be calculated as

$$\int_{t=0}^{t=\theta_s^t} u(t)dt \approx \sum_{k=1}^{k=\theta_s^t/h_s} u_{s,k}h_s \tag{42}$$

Combining the objective functions described in (39)–(41) and the constraints at both scheduling level (8)–(18), control level (19)–(30) and the linking constraints (31)–(38) we obtain the optimization model for the integrated problem as shown in (43)

$$\begin{aligned} &\max_{y_{p,s},x_{s,k},u_{s,k},\theta_s^t,\Theta_p,W_p,t_s^s,t_s^e,T_c} \quad \Phi_1-\Phi_2-\Phi_3 \\ &\text{s.t.} \begin{cases} (8)\text{–}(18) & \text{constraints at scheduling level} \\ (19)\text{–}(30) & \text{constraints at control level} \\ (31)\text{–}(38) & \text{linking constraints} \end{cases} \end{aligned} \tag{43}$$

Fast MPC for PWA Systems

The role of fast MPC

The MPC problem for LTI systems with quadratic objective corresponds to a quadratic programming (QP) problem. Fast MPC solves the QP problem online to satisfactory optimality criteria, whereas if mp-MPC is used the calculations are done offline.

One of the shortcomings of mp-MPC however, is that the number of critical regions grows exponentially with respect to the size of MPC problem (e.g., number of input and output, prediction horizon). In contrast, fast MPC can solve large scale problem online. Another advantage of fast MPC is that fast MPC is capable to perform Real Time Optimization (RTO) as described by problem (44) where the reference $\bar{x}_k$ (set point) is a continuous function rather than a fixed value. However, mp-MPC treats the reference trajectory as parameters and the parameter space (the critical regions) is exponentially increasing with respect to the length of prediction horizon (the length of reference track). This makes it impossible to apply mp-MPC in cases with long prediction horizon.

$$\begin{aligned} &\min_{u_k} \sum_{k=1}^{N-1}\left(\|x_{k+1}-\bar{x}_{k+1}\|_2^{Q_k}+\|u_k-\bar{u}_k\|_2^{R_k}\right)+\|x_{k+N}-\bar{x}_{k+N}\|_2^{Q_N} \\ &s.t. \begin{cases} x_1=x^0 \\ x_{k+1}=Ax_k+Bu_k+C, \quad k=1,\ldots,N-1 \\ x_{\min} \le x_k \le x_{\max}, \quad k=1,\ldots,N \\ u_{\min} \le u_k \le u_{\max}, \quad k=1,\ldots,N \end{cases} \end{aligned} \tag{44}$$

The general methodology of fast MPC for linear systems is to transform the MPC problem into a convex quadratic programming, apply nonlinear programming methods and speed up the computation by exploring the problem structure. The detail solution approaches can be categorized as, interior point method (barrier method), active set method and fast gradient method. Using interior point method the MPC problem is built as a QP and the inequality constraints of QP are treated with a log barrier and added to the objective.[19,44] The resulting problem is then solved using Newton's method with the barrier coefficient adjusted in each iteration till convergence. When active set method is applied to MPC problem,[18] equality problems are defined by active set of the original QP and updated when different active constraints are considered in the equality problems. Thus the equality problems are solved sequentially and the iterations are terminated if all KKT conditions are satisfied. Fast gradient method[45] has also been applied in solving the linear quadratic MPC problem where iteration algorithms with derived lower iteration bounds are developed.[17,21] Recently a primal-dual interior point method is applied in solving MPC problems[46] where the authors proposed an efficient solution method for the KKT conditions derived from the linear MPC problem and thus speed up the interior point method.

Fast MPC for PWA systems

Note that the above mentioned fast MPC strategies are targeting linear systems. In the following we propose an algorithm of implementing fast MPC on nonlinear systems (problem (45)). The basic idea is to first derive the PWA model from nonlinear dynamics, and then iteratively locate the LTI in PWA and implement fast MPC at the local LTI so as to quickly drive the states to the next sample step.

$$\begin{aligned} &\min_{x(t),u(t)} \int_{t=0}^{t=t_H} \left(x^TQx+u^TRu\right)dt+x_N^TQx_N \\ &\text{s.t.} \begin{cases} x(0)=x^0 \\ \dot{x}=f(x,u) \\ x_{\min} \le x(t) \le x_{\max}, \ 0 \le t \le t_H \\ u_{\min} \le u(t) \le u_{\max}, \ 0 \le t \le t_H \end{cases} \end{aligned} \tag{45}$$

Step 1: Transform nonlinear dynamics $\dot{x}=f(x,u)$ into PWA using the proposed optimization method in Section "A PWA identification technique using optimization methods"

$$\text{PWA}=\bigcup_i \text{LTI}^{(i)} \tag{46}$$

$$\begin{aligned} &\text{LTI}^{(i)} : x(k+1)=A_ix(k)+B_iu(k)+C_i \\ &\text{if } x(k) \in \Omega_i=\{x : V_ix \le W_i\}, \ i \in S_i=\{1,2,\ldots N_i\} \\ &\text{where } \cup_i \Omega_i=\Omega_x \text{ and } \Omega_{i_1} \cap \Omega_{i_2}=\varnothing \quad \text{if} \quad i_1 \ne i_2 \end{aligned} \tag{47}$$

Step 2: Set the initial state and initial manipulated variables (x^0,u^0)

Step 3: Locate the corresponding LTI for current states

$$\begin{aligned} &\text{if } x \in \Omega_i=\{x : V_ix \le W_i\}, \ i \in S_i=\{1,2,\ldots N_i\}, \\ &\text{then select } \text{LTI}^{(i)} : x(k+1)=A_ix(k)+B_iu(k)+C_i \end{aligned} \tag{48}$$

This step can be computationally very expensive especially when x and u are of high dimensions. If an enumeration method is used someone has to loop over all the LTIs

Table 1. Comparison of Enumeration and Binary Search Based on Their CPU Time Used in Searching for a LTI in a 3-D PWA $\Omega_x \otimes \Omega_u \subset R^3$

Number of discretization	$N_1=N_2=N_3=10$	$N_1=N_2=N_3=100$	$N_1=N_2=N_3=200$
Binary search CPU time	0.016s	0.036s	0.047s
Enumeration CPU time	0.016s	1.922s	14.266s

to determine the position of $(x(k+1), u^*(k))$, leading to complexity that is increasing in the order of $o(\Pi_i N_i)$. To lower the complexity we use binary search method[47,48] with much lower computational complexity $o\left(\sum_i \lg(N_i)\right)$.

When the dimension is high and the number of discretization points is large, binary search is significantly faster. This is shown in (Table 1) with an example of searching for a LTI in a 3-D PWA with varying number of discretization points (10, 100, and 200).

Step 4: Solve the MPC problem (49) for LTI$^{(i)}$ using toolbox FORCES[49] which is based on primal-dual interior-point method,[46] and obtain $u^*(k)$. Primal-dual interior-point method efficiently solves for the KKT conditions derived from the linear MPC problem and thus speed up the interior point method. Note that problem (45) is a general form of MPC for nonlinear systems. Problem (49) is the MPC formulation for PWA systems where a dynamic reference $\bar{x}_{k+k_p}$ is tracked

$$\min_{u_k} \sum_{k_p=1}^{N-1} \left(\|x_{k+k_p} - \bar{x}_{k+k_p}\|_2^{Q_{k_p}} + \|u_{k+k_p-1} - \bar{u}_{k+k_p-1}\|_2^{R_{k_p}} \right) + \|x_{k+N} - \bar{x}_{k+N}\|_2^{Q_N}$$

$$\text{s.t.} \begin{cases} x_k = x^0 \\ x_{k+k_p} = A_i x_{k+k_p-1} + B_i u_{k+k_p-1} + C_i, \quad k_p = 1, \ldots, N \\ x_{\min} \le x_{k+k_p} \le x_{\max}, \quad k_p = 1, \ldots, N \\ u_{\min} \le u_{k+k_p} \le u_{\max}, \quad k_p = 1, \ldots, N \end{cases} \tag{49}$$

Step 5: Evaluate the state transfer using the following equation

$$x(k+1) = A_i x(k) + B_i u^*(k) + C_i \tag{50}$$

Step 6: $k = k+1$, go to step 3.

Stability analysis

Input-output stability implies Bounded Input-Bounded Output (BIBO) stability, that is, in a well behaved system a bounded input should result in a bounded output.[50] As PWA is a discrete system and the stability is relevant to the sample step, we analyze the stability in the context of z-transform and derive the necessary conditions for a stable PWA. The necessary conditions provide a reference when deciding the sample step.

A general form of PWA $x(k+1) = Ax(k) + Bu(k) + C$, where $x(k) \in R^{n\times 1}$, $u(k) \in R^{m\times 1}$, $A \in R^{n\times n}$, $B \in R^{n\times m}$, $C \in R^{n\times 1}$ has an equivalent form

Table 2. Steady-State Information of a CSTR with Cyclic Production

Product	Q(L/h)	C_R(mol/L)	Demand rate (kg/h)	Product price ($/kg)
A	10	0.0967	3	200
B	100	0.2	8	150
C	400	0.3032	10	130
D	1000	0.393	10	125
E	2500	0.5	10	120

$$x'(k+1) = Ax'(k) + Bu'(k) \tag{51}$$

where $x'(k) = x(k) + M_1$ and $u'(k) = u(k) + M_2$. M_1 and M_2 can be obtained through $(A-I)M_1 + BM_2 = C$.

Applying z transform to (51) yields

$$(I - Az^{-1})X(z) = Bz^{-1}U(z) \tag{52}$$

Thus we obtain a transfer function

$$\frac{X(z)}{U(z)} = (I - Az^{-1})^{-1}Bz^{-1} \tag{53}$$

The necessary conditions for BIBO stability of (52) are that the absolute value of all the diagonal terms of A should be less than one

$$|A_{ii}| < 1, \ 1 \le i \le n \tag{54}$$

Case Studies

To demonstrate the feasibility of the proposed PWA identification and solve the integrated problem using PWA and fast MPC, we studied two numerical cases, a SISO CSTR and a MIMO CSTR continuous production.

SISO CSTR

The data for the first case study involving a CSTR is taken from Flores-Tlacuahuac and Grossmann.[6] The reaction $3R \xrightarrow{k} P$ takes place in an isothermal CSTR with reaction rate $-r_R = kC_R^3$, while products A, B, C, D and E, which are differentiated by their concentration C_R (Table 2), are manufactured in a cyclic mode (Figures 5 and 6). The basic dynamic model of the process is shown in Equation 55.

$$\frac{dC_R}{dt} = \frac{Q}{V}(C_0 - C_R) + r_R \tag{55}$$

where C_0 is feed stream concentration, Q is the feed flow rate (i.e., manipulated variable), and C_R is the concentration of the raw material in the outflow (i.e., state variable). Using u and x to represent the manipulated and state variable respectively leads to the following

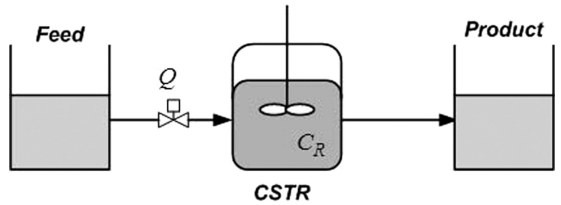

Figure 5. CSTR cyclic production, feeding flow rate is the manipulated variable and raw material concentration is the state variable.

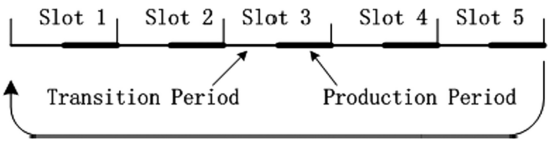

Figure 6. CSTR cyclic production, each slot is composed of a transition period and a production period.

$$\frac{dx}{dt} = \frac{u}{5000}(1-x) - 2x^3 \tag{56}$$

PWA Identification. Following the PWA identification approach presented in Section "A PWA identification technique using optimization methods," we transform the nonlinear dynamics in (56) into PWA by solving problems (7) using GAMS/IPOPT. Using 10 discretization points for x and u results in the solution of problem (7) in 3.4 CPU s. The profile of the original function and PWA are shown in Figure 7 where the smooth and colored sphere represents the original nonlinear dynamic and the grid represents the PWA.

Solving the Integrated Problem. In this case study, we assign five slots, five products, 20 sample steps during transitions. Therefore, $N_p=N_S=5$, $N_k=20$. We solve the integrated problem with PWA and the original integrated problem MINLP using GAMS/SBB and summarize the results in Table 3.

The scheduling solution including production sequence, transition time, production time for each product and the economic performance such as revenue, cost and profit are provided in Table 3. The CPU time of the original integrated problem is higher. This is due to its nonlinearity brought by the discretization using implicit RK4 method. However, the objective values (the profit) of the original integration and the integration using PWA are almost the same.

Implementation of Fast MPC. We implement fast MPC on the obtained PWA system following the algorithm presented in Section "Fast MPC for PWA systems." The original control problem is a nonlinear MPC problem (57), which is transformed through the PWA identification method.

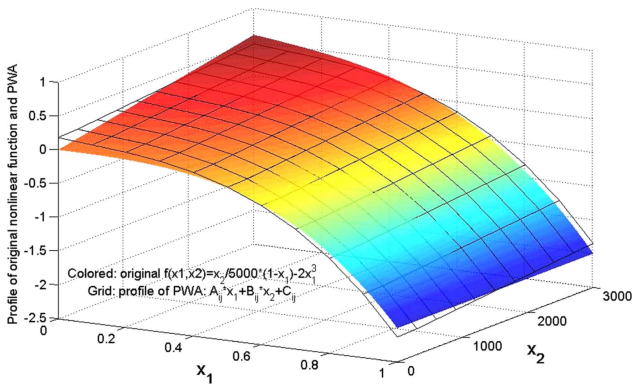

Figure 7. Profiles of the nonlinear dynamic and the PWA.

[Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

Table 3. Results of the Integrated Problem of SISO CSTR Case

Types of integration	Integration using PWA	Original integration MINLP
Number of variables	1601	2986
Number of constraints	4542	4046
CPU time (s)	334.64	785.72
Relative Gap	1.19 %	1.16 %
Optimal sequence	D-E-A-B-C	D-E-C-A-B
Cycle time	116.52	116.02
Transition time in slot 1 to 5 (h)	0.600, 1.000, 52.200, 0.500, 0.600	0.558, 0.832, 1.767, 56.885, 0.500
Production time in slot 1 to 5 (h)	0.198, 0.880, 44.282, 13.750, 2.511	0.243, 0.928, 4.162, 38.525, 11.600
Revenue ($)	4639.03	4458.95
Raw material cost ($)	1174.48	993.12
Profit ($)	3464.55	3465.83

$$\min_u \; J = \int_0^{t_H} |x - \bar{x}_{\text{ref}}|\, dt$$

$$\text{s.t.} \begin{cases} \dfrac{dx}{dt} = \dfrac{u}{5000}(1-x) - 2x^3 \\ x(0) = x^0 \\ 0 \le x(t) \le 1, \;\; 0 \le t \le t_H \\ 0 \le u(t) \le 3000, \;\; 0 \le t \le t_H \\ \Delta u \le 200 \end{cases} \tag{57}$$

The PWA of this case has 100 LTIs. It takes 0.0636 CPU s to solve for the entire transition using a sample step of 0.02 h, and 0.000279 CPU s for the MPC problem over the local LTI at one sample step. The calculations were performed using a PC of 1.86 GHz/4GB RAM. Figures 8 and 9 illustrate two examples of transition (transition from product 2 to 5 and transition from product 5 to 4). Both figures demonstrate the profiles of manipulated variables and state variables. Note that the manipulated variables are subject to a bound of 200 L/h that represents the maximum increasing or decreasing rate within one sample interval. In Figure 8 the manipulated variable feeding flow rate Q increases up to its maximum value 3000 L/h to

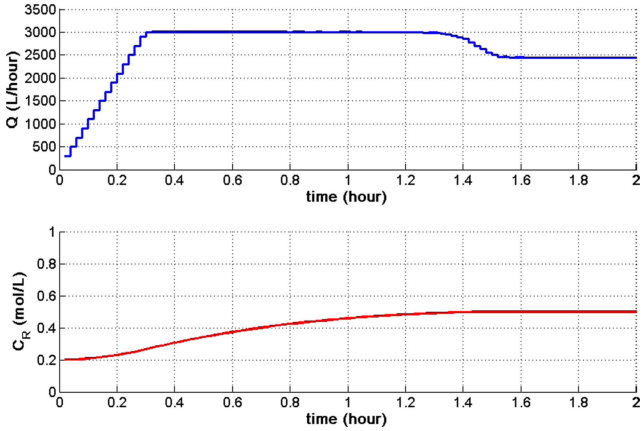

Figure 8. Transition profile from product 2 to product 5 in SISO CSTR, obtained by fast MPC.

[Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

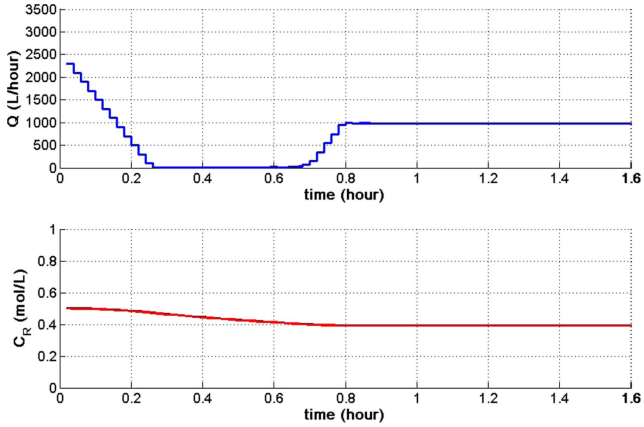

Figure 9. Transition profile from product 5 to product 4 in SISO CSTR, obtained by fast MPC.

[Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

drive the concentration of raw material C_R from one steady state 0.2 mol/L to another steady state 0.5 mol/L in a short time period. Note that Q decreases to the corresponding steady-state value of 2500 L/h before C_R is approaching 0.5 mol/L. This is due to the involvement of MPC that helps to avoid the overshot. Similarly in Figure 9 the feeding flow rate Q decreases to its minimum value 10 L/h to speed up the transition and then adjusts to 1000 L/h to catch the following production period where the steady-state value of C_R is 0.393 mol/L.

The Effects of Threshold at the State Feedback in the Proposed Framework. As describe in the introduction section, a threshold of state deviation is used to determine whether the state information is feedback to the inner fast MPC or outer integrated problem so that the scheduling solution can be updated. As we test many different values of the threshold, we find that a state deviation less than 0.08 mol/L would not disrupt the scheduling solution and thus could be handled by the fast MPC, while in the case of a state deviation greater than that it is more profitable to update the scheduling solution.

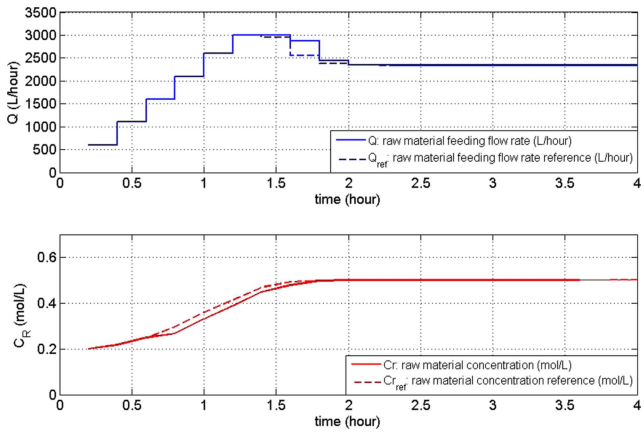

Figure 10. Disturbance with magnitude −0.04 (less than threshold 0.08) at time 0.7 h (in transition), handled by fast MPC, scheduling solution does not change.

[Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

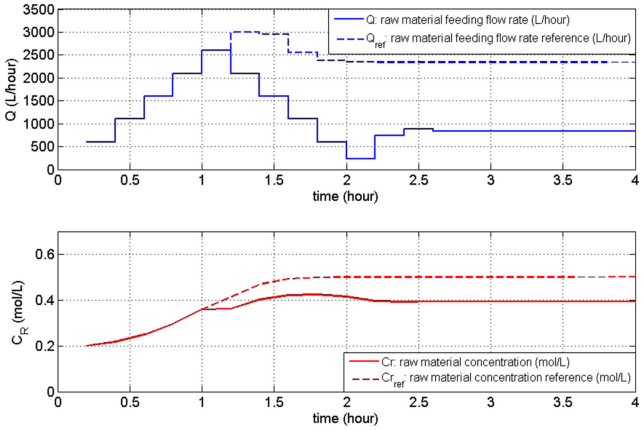

Figure 11. Disturbance with magnitude −0.09 (great than threshold 0.08) at time 1.2 h (in transition), steady-state value (scheduling solution) updated by integrated problem, process goes to product 4 (SS=0.393) instead of product 5 (SS=0.5).

[Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

Therefore we set up the threshold as 0.08 mol/L. Figures 10 and 12 demonstrate two examples of the responses to minor disturbance at transtion and production periods where the scheduling solutions are not interupted. In Figure 10, the disturbance occurs at 0.7 h with magnitude 0.04 mol/L. As this is a minor disturbance the scheduling solution of the integrated problem is not updated, i.e., the process continues with the current transition to product 5. In Figure 12, the disturbance occurs at steady state with magnitude 0.05 mol/L and the control action adjusts the deviated state back to the steady state. Figure 11, however, demonstrates a significant disturbance at transitions and thus the scheduling solution is updated, i.e., the process goes to another production stage. In this case the disturbance occurs at 1.2 h with magnitute 0.09 mol/L. This is a significant deviation and the scheduling solution is updated to keep the optimality of the overall operations. So product 4 is produced next instead of product 5.

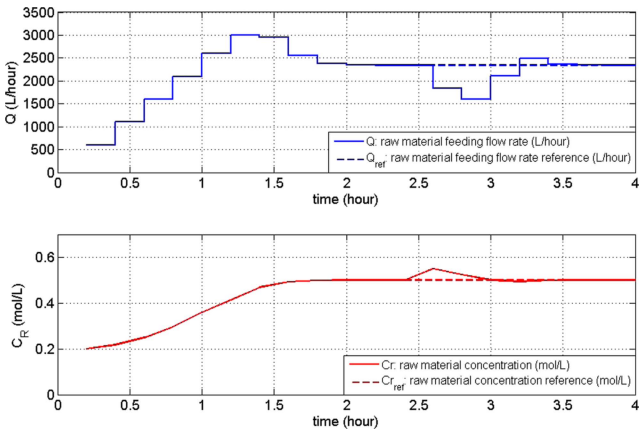

Figure 12. Disturbance with magnitude +0.05 (less than threshold 0.08) at time 2.6 h (in production), handled by fast MPC, scheduling solution does not change.

[Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

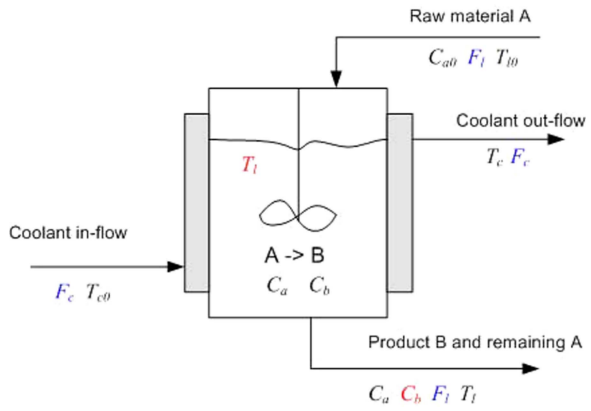

Figure 13. A MIMO CSTR process.

[Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

MIMO CSTR

In this case study we investigate a MIMO CSTR cyclic production where reaction A → B takes place with first order reaction rate: $r = -kC_a$ (Figure 13). Specifications of this process are adopted from Camacho and Bordons Alba.[51] The manipulated variables are F_1 (flow rate of the liquid) and F_c (flow rate of the coolant). States variables are C_b (concentration of product B) and T_1 (liquid temperature). Three steady states are involved in the cyclic production (Table 4). The detailed process specifications are provided in Table 5. The utility price for the coolant F_c is \$3/m^3, and the raw material price is \$10/m^3. The model derived from mass and energy balance are represented by Eqs. 58 and 59, respectively

$$\frac{d(V_1 C_b)}{dt} = V_l k(C_{a0} - C_b) - F_1 C_b \tag{58}$$

$$\frac{d(V_1 \rho_1 C_{pl} T_1)}{dt} = F_1 \rho_l C_{pl} T_{l0} - F_1 \rho_1 C_{pl} T_1 + F_c \rho_c C_{pc}(T_{c0} - T_c) + V_l k(C_{a0} - C_b) H \tag{59}$$

Identifying the PWA from (58) and (59) leads to Eqs. 60 and 61.

$$C_b(k+1) = [A_{11} A_{12}] \begin{bmatrix} C_b(k) \\ T_1(k) \end{bmatrix} + [B_{11} B_{12}] \begin{bmatrix} F_1(k) \\ F_c(k) \end{bmatrix} + C_1 \tag{60}$$

$$T_l(k+1) = [A_{21} A_{22}] \begin{bmatrix} C_b(k) \\ T_l(k) \end{bmatrix} + [B_{21} B_{22}] \begin{bmatrix} F_1(k) \\ F_c(k) \end{bmatrix} + C_2 \tag{61}$$

That can be represented in a matrix format as follows

$$\begin{bmatrix} C_b(k+1) \\ T_1(k+1) \end{bmatrix} = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} \begin{bmatrix} C_b(k) \\ T_1(k) \end{bmatrix} + \begin{bmatrix} B_{11} & B_{12} \\ B_{21} & B_{22} \end{bmatrix} \begin{bmatrix} F_1(k) \\ F_c(k) \end{bmatrix} + \begin{bmatrix} C_1 \\ C_2 \end{bmatrix} \tag{62}$$

The obtained PWA is composed of three LTIs as shown in (63)–(65).

Table 5. Process Specifications of MIMO CSTR

k reaction constant	26/h
H heat of reaction	70000 kJ/kmol
ρ_l liquid density	800 kg/m^3
ρ_c coolant density	1000 kg/m^3
C_{pl} specific heat of liquid	3 kJ/(kg K)
C_{pc} specific heat of coolant	4.19 kJ/(kg K)
V_l tank volume	24 m^3
T_{l0} liquid entering temperature	283 K
T_{c0} coolant in-flow temperature	273 K
T_c coolant out-flow temperature	303 K
C_{a0} initial concentration of reactant	4 mol/L

$$\begin{bmatrix} C_b(k+1) \\ T_1(k+1) \end{bmatrix} = \begin{bmatrix} -0.0164 & 0 \\ -7.4115 & 0.2377 \end{bmatrix} \begin{bmatrix} C_b(k) \\ T_1(k) \end{bmatrix} + \begin{bmatrix} -0.0004 & 0 \\ -0.0029 & -0.0218 \end{bmatrix} \begin{bmatrix} F_1(k) \\ F_c(k) \end{bmatrix} + \begin{bmatrix} 1.7788 \\ 250.7210 \end{bmatrix}, \tag{63}$$

$$\text{if } 0 \le C_b < 1.15$$

$$\begin{bmatrix} C_b(k+1) \\ T_1(k+1) \end{bmatrix} = \begin{bmatrix} 0.2181 & 0 \\ -7.4115 & 0.4722 \end{bmatrix} \begin{bmatrix} C_b(k) \\ T_1(k) \end{bmatrix} + \begin{bmatrix} -0.0005 & 0 \\ -0.0113 & -0.0218 \end{bmatrix} \begin{bmatrix} F_1(k) \\ F_c(k) \end{bmatrix} + \begin{bmatrix} 1.7025 \\ 193.2533 \end{bmatrix}, \tag{64}$$

$$\text{if } 1.15 \le C_b < 1.6$$

$$\begin{bmatrix} C_b(k+1) \\ T_1(k+1) \end{bmatrix} = \begin{bmatrix} 0.4918 & 0 \\ -7.4115 & 0.7459 \end{bmatrix} \begin{bmatrix} C_b(k) \\ T_1(k) \end{bmatrix} + \begin{bmatrix} -0.0008 & 0 \\ -0.0196 & -0.0218 \end{bmatrix} \begin{bmatrix} F_1(k) \\ F_c(k) \end{bmatrix} + \begin{bmatrix} 1.5247 \\ 113.5021 \end{bmatrix}, \tag{65}$$

$$\text{if } C_b \ge 1.6$$

The solution of the integrated problem is provided in Table 6. It takes 14.1 s to solve the integrated problem which maximizes the profit using DICOPT solver in GAMS. The optimal production sequence is C-A-B. The transition time, production time for each product and the economic performance such as revenue, cost and profit are provided in Table 6. It takes 0.028 s for the fast MPC to obtain the entire transitions (sample step is 0.02 h) and 0.00045 s for the MPC problem over the local LTI at one sample step, using a PC of 1.86 GHz/4GB RAM. Figures 14 and 15 illustrate two examples of transition (transition from steady state 3 to 1 and transition from steady state 1 to 2).

A Comparison Between Fast MPC and mp-MPC. Here we compare the performance of fast MPC and mp-MPC in terms of algorithms and computational complexity. We present solution algorithms of implementing fast MPC ((45)–(50) in Section "Fast MPC for PWA systems") and mp-MPC

Table 4. Products at Different Steady States for the Second Case Study

Steady state	C_b (mol/L)	T_1 (K)	F_1 (m^3/h)	F_c (m^3/h)	Demand (L/h)	Price (\$/L)	Inventory (\$/h)
SS1: produce A	1	290	1829	774	110	17	1.7
SS2: produce B	1.3	310	1266	264	120	25	2
SS3: produce C	2	330	610	132	70	32	1.8

Table 6. Scheduling Solutions for the Integrated Problem of MIMO CSTR Case

Number of variables	555
Number of constraints	2712
Solver and CPU time (s)	DICOPT, 14.1s on 3.9GHz/4GRAM
Relative Gap	0.00%
Optimal sequence	C-A-B
Cycle time (h)	10
Transition time in slot 1 to 3 (h)	0.1, 0.15, 0.07
Production time in slot 1 to 3 (h)	1.15, 0.60, 7.93
Revenue ($)	29211.73
Raw material cost($)	12222.46
Utility cost ($)	829.03
Inventory cost ($)	3513.85
Profit($)	12646.39

((67)–(71)) on nonlinear systems, and their detailed computation time as summarized in Table 7 and Table 8.

MPC problem for nonlinear systems (The original problem)

$$\min_{x(t),u(t)} \int_{t=0}^{t=t_H} \left(x^T Q x + u^T R u\right) dt + x_N^T Q x_N$$
$$\text{s.t.} \begin{cases} x(0) = x^0 \\ \dot{x} = f(x,u) \\ x_{\min} \le x(t) \le x_{\max}, \ 0 \le t \le t_H \\ u_{\min} \le u(t) \le u_{\max}, \ 0 \le t \le t_H \end{cases} \quad (66)$$

Algorithms of Implementing mp-MPC on Nonlinear Systems.

Step 1: Transfer nonlinear dynamic $\dot{x} = f(x,u)$ into PWA using the proposed optimization based method

$$\text{PWA} = \bigcup_i \text{LTI}^{(i)} \quad (67)$$

$$\text{LTI}^{(i)} : x(k+1) = A_i x(k) + B_i u(k) + C_i$$
$$\text{if } x(k) \in \Omega_i = \{x : V_i x \le W_i\}, \quad i \in S_i = \{1, 2, \ldots N_i\} \quad (68)$$
$$\text{where } \cup_i \Omega_i = \Omega_x \text{ and } \Omega_{i_1} \cap \Omega_{i_2} = \varnothing \quad \text{if } i_1 \ne i_2$$

Step 2: Solve the following mp-MPC problem offline for the PWA system obtained in step 1.

$$\min_{u_k} \sum_{k_p=1}^{N-1} \left(\|x_{k+k_p} - \bar{x}_{k+k_p}\|_2^{Q_{k_p}} + \|u_{k+k_p-1} - \bar{u}_{k+k_p-1}\|_2^{R_{k_p}} \right) + \|x_{k+N} - \bar{x}_{k+N}\|_2^{Q_N}$$
$$\text{s.t.} \begin{cases} x_k = x^0 \\ x_{k+k_p} = A_i x_{k+k_p-1} + B_i u_{k+k_p-1} + C_i, \ k_p = 1, \ldots, N \\ x_{\min} \le x_{k+k_p} \le x_{\max}, \ k_p = 1, \ldots, N \\ u_{\min} \le u_{k+k_p} \le u_{\max}, \ k_p = 1, \ldots, N \end{cases} \quad (69)$$

Step 3: Set the initial state and initial manipulated variables (x^0, u^0)

Step 4: Locate the corresponding critical region (CR) (Figure 16) for current states and reference states in the prediction horizon, using binary searching method.[47,48]

Step 5: Evaluate the manipulated variables as a function of states and obtain the optimal control inputs.

$$\text{if } x(k) \in CR_j = \{x : H_j x \le K_j\}, \quad j \in J = \{1, 2, \ldots N_j\}$$
$$\text{then } u^*(k) = F_j x(k) + G_j \quad (70)$$

Step 6: Locate the corresponding LTI for the current states using binary searching method.

Step 7: State transfer

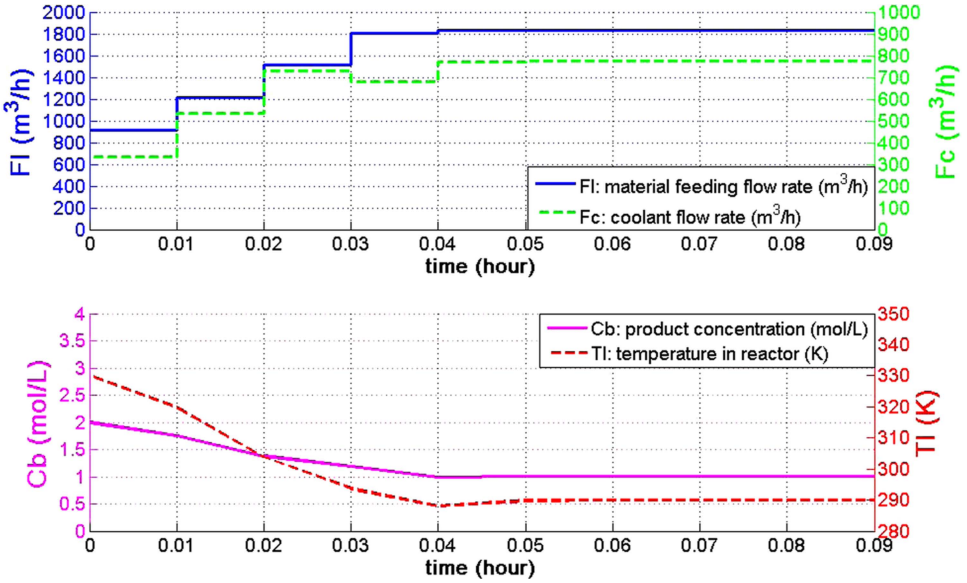

Figure 14. Transition profile from steady state 3 to steady state 1 in MIMO CSTR, obtained by fast MPC.

[Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

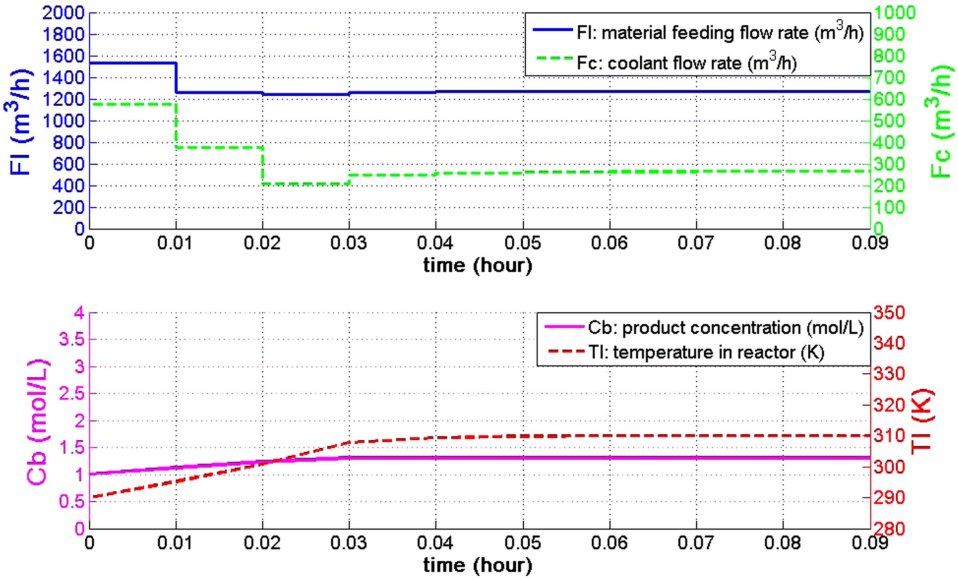

Figure 15. Transition profile from steady state 1 to steady state 2 in MIMO CSTR, obtained by fast MPC.

[Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

Table 7. Analysis of the Computation Time for Fast MPC and mp-MPC, Online Steps Only

CPU time	Fast MPC	mp-MPC
Step 3	Locate LTI: $t_{\mathrm{LTI}}=\varepsilon\sum_{m=1}^{N_x}\lg(N_m)$	N/A
Step 4	Solve QP using primal-dual interior-point method: t_{QP}	Locate CR: $t_{\mathrm{CR}}=\varepsilon\left(\sum_{m=1}^{N_x}\lg(N_m)+\sum_{n=1}^{N_H}\lg(N_n)\right)$
Step 5	State transfer: t_{ST}	Function evaluation: t_{FE}
Step 6	Move to next sample step: t_{MO}	Locate the LTI: $t_{\mathrm{LTI}}=\varepsilon\sum_{m=1}^{N_x}\lg(N_m)$
Step 7	N/A	State transfer: t_{ST}
Step 8	N/A	Move to next sample step: t_{MO}
Total	$t_{\mathrm{fastMPC}}=\varepsilon\sum_{m=1}^{N_x}\lg(N_m)+t_{\mathrm{QP}}+t_{\mathrm{ST}}+t_{\mathrm{MO}}$	$t_{\mathrm{mpMPC}}=\varepsilon\left(2\sum_{m=1}^{N_x}\lg(N_m)+\sum_{n=1}^{N_H}\lg(N_n)\right)+t_{\mathrm{FE}}+t_{\mathrm{ST}}+t_{\mathrm{MO}}$

$$\begin{aligned}&\text{if } x(k)\in\Omega_i=\{x:\ V_ix\le W_i\},\quad i\in S_i=\{1,2,\ldots N_i\}\\&\text{then } x(k+1)=A_ix(k)+B_iu^*(k)+C_i\end{aligned}\tag{71}$$

Step 8: $k=k+1$, go to step 4.

Note that mp-MPC and fast MPC share step 1. In mp-MPC, step 1, 2 are solved off-line, and steps 4 through 8 are calculated online in a loop manner. In fast MPC, step 1 is solved off-line, and Steps 3 through 6 are calculated online in a loop manner.

When comparing the computation complexity of fast MPC and mp-MPC, we focus on the online steps. Table 7 presents the detailed CPU time for each online step of fast MPC and mp-MPC.

We use YALMIP toolbox[52] to solve the MPC problem in (69) and obtain the explicit control solution shown in Figure 17 where both manipulated variables F_1 and F_c are functions of state variables *Cb* and *Tl*. Then we implement the explicit solution following steps 4–8 in the algorithm of implementing mp-MPC and obtain the same transition profiles as those obtained by fast MPC. Table 9 summarizes the computation time of fast MPC and mp-MPC in solving for one sample step as well as the whole transition period. As shown in the table, mp-MPC requires more computational time. This is consistent with the fact presented in Table 7 that mp-MPC takes more time due to its repeatedly locating the critical regions.

Table 8. Notations Associated with Table 7

N_x	Number of states dimension
N_m	Number of discretization point for the mth state
N_H	Prediction horizon
N_n	Number of discretization point for the nth reference state in the prediction horizon
ε	Unit time in binary searching method
t_{LTI}	Time consumed in locating LTI
t_{CR}	Time consumed in locating critical regions
t_{ST}	Time consumed in state transfer
t_{MO}	Time consumed in moving to the next sample step
t_{FE}	Time consumed in function evaluation
t_{fastMPC}	Time consumed in implanting fast MPC in one sample step
t_{mpMPC}	Time consumed in implanting mp-MPC in one sample step

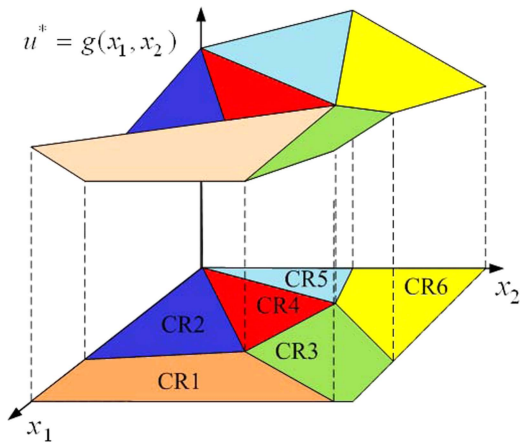

Figure 16. Locate the critical regions and evaluate the explicit solution obtained by mp-MPC.

[Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

Conclusions

In this study, we propose a novel framework for the integration of scheduling and control. This framework aims to simultaneously consider the scheduling and control problem and facilitate the online applications where disturbances are efficiently handled. In this framework, a PWA model is identified from the process's first principle model and incorporated within the scheduling level, leading to an integrated model. More specifically, the PWA is obtained using optimization method that minimizes the error between the original nonlinearity and the proposed PWA composed of a group of local LTIs. Note that the PWA is incorporated as linear constraints with the scheduling level. This eliminates the nonlinear constraints brought by traditional integration that uses collocation point discretization, and thus reduces the computation complexity of the integrated problem.

In the presented framework, we proposed a heuristic determination of the threshold feedback value. For example in the case of SISO CSTR, we tested many cases of disturbances with different magnitudes and found that when the magnitude of disturbance was great than 0.09, there was a high probability that the scheduling solution should be updated to keep the optimality of the process. When the disturbance was less than 0.05, it is very likely that scheduling solution remains the same. Therefore, there is no exact threshold for the feedback. In this study, we choose an empirical value between 0.05 and 0.09 to avoid unnecessarily computing the integrated problem and meanwhile guarantee the optimality when disturbance occurs.

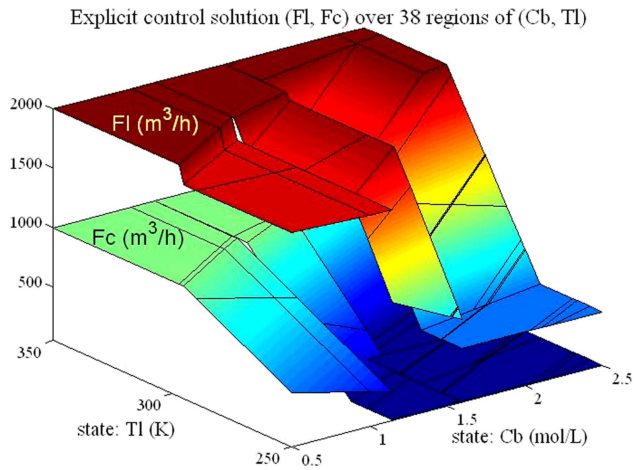

Figure 17. Explicit solution for the control problem in transition from product 3 to 1 of MIMO CSTR case.

[Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

Table 9. Comparison of CPU Time in Implementing Fast MPC and mp-MPC Using a 1.86GHz/4G RAM PC

CPU time	Fast MPC	mp-MPC
One sample step	0.00045s	0.016s
Entire transition	0.028s	0.437s

To facilitate the online application, we use fast MPC to compute the optimal control actions and eliminate the effects of disturbances in a timely manner. Previous investigation of mp-MPC in solving the control problem[15] has shown that the number of critical regions in the explicit solution of mp-MPC increases exponentially with respect to problem size in terms of number of dimensions of state and manipulated variables and the length of prediction horizon. To overcome the dimensionality problem, we propose to use fast MPC in this study. We develop the algorithms of implementing fast MPC and mp-MPC and compare their computing performance. Results show that fast MPC computes the control problem much faster than mp-MPC does, though both of them perform in the speed of msec. Another advantage of fast MPC is its capability to track a dynamic reference. Specifically, the reference of fast MPC is updated when the prediction horizon is moving forward in online computation. This makes it an economic MPC where the dynamic reference is obtained under an economic objective. However, mp-MPC can only handle a fixed reference, as the explicit control solution is solved offline based on a fixed MPC problem. To the authors' knowledge, this is the first attempt in this area to explore the possibility and feasibility of applying fast-MPC in simultaneous scheduling and control problem. In the future, we intend to investigate more computationally efficient PWA approximation techniques and the application of the proposed approach in larger problems.

Acknowledgments

The authors gratefully acknowledge financial support from NSF under grant CBET 1159244.

Notation

Index and sets

p = product
s = slot
k = sample points
k_p = future sample steps of step k
i = LTI in PWA
S_p = set of products, $S_p=\{1,2,\ldots,N_p\}$
S_s = set of slots, $S_s=\{1,2,\ldots,N_s\}$
S_k = set of sample steps, $S_k=\{1,2,\ldots,N_k\}$
S_i = set of polytopes in PWA, $S_i=\{1,2,\ldots,N_i\}$

Parameters

N_p = number of products
N_s = number of slots
N_k = number of sample steps
N_i = number of LTIs in PWA
M = a big positive number
h = sample step
G_p = production rate of product p
D_p = demand of product p

P_p = price of product p
P_r = price of raw material
P_u = price of utility
Θ_{max} = maximum production time
x_{min} = lower bound of state variables
x_{max} = upper bound of state variables
u_{min} = lower bound of manipulated variables
u_{max} = upper bound of manipulated variables
Δ_u = maximum increment of manipulated variables
x_{margin} = the margin of quality bounds
Q = weight matrix for state variables in the quadratic objective of MPC
R = weight matrix for manipulated variables in the quadratic objective of MPC
t_H = prediction horizon of MPC

Variables

$x_{s,k}$ = state variable at sample step k in slot s
$u_{s,k}$ = manipulated variable at sample step k in slot s
$u_{1s,k}$ = raw material feeding flow rate at sample step k in slot s
$u_{2s,k}$ = heating/cooling flow rate at sample step k in slot s
$x_{ss,p}$ = steady-state value of product p
$u_{ss,p}$ = steady manipulated value of product p
$x_{in,s}$ = initial state value in slot s
$u_{in,s}$ = initial manipulated value in slot s
$\bar{x}_s$ = desired state value in slot s(the set point)
$\bar{u}_s$ = desired manipulated value in slot s(the set point)
$y_{p,s}$ = binary variable indicating assignment of product p to slot s
$y1_{s,k,i}$ = binary variable indicating the selection of LTI in PWA
$y4_{s,k}$ = binary variable indicating the end of transitions, equal to 1 if step k is the end of transition
$y41_{s,k}$ = auxiliary binary variable for $y4_{s,k}$, equal to one if state is below the lower bound
$y42_{s,k}$ = auxiliary binary variable for $y4_{s,k}$, equal to one if state is above the upper bound
Θ_s = production time in slot s
Θ_p = production time of product p
$\Theta_{p,s}$ = production time of product p in slot s
θ_s^t = transition time in slot s
t_s^s = starting time of slot s
t_s^e = ending time of slot s
T_c = total production cycle time
W_p = amount produced for product p
VR = valid region where the LTI of PWA is valid
A_i = coefficient of LTI
B_i = coefficient of LTI
C_i = coefficient of LTI
H_j = coefficient of polytope of critical region j
K_j = coefficient of polytope of critical region j
F_j = coefficient of explicit control solution for critical region j
G_j = coefficient of explicit control solution for critical region j
V_i = coefficient of polytope of valid region of LTI
W_i = coefficient of polytope of valid region of LTI

Literature Cited

1. Engell S, Harjunkoski I. Optimal operation: scheduling, advanced control and their integration. *Comput Chem Eng*. 2012;47:121–133.
2. Terrazas-Moreno S, Flores-Tlacuahuac A, Grossmann IE. Simultaneous design, scheduling, and optimal control of a methyl-methacrylate continuous polymerization reactor. *AIChE J*. 2008;54(12):3160–3170.
3. Harjunkoski I, Nystrom R, Horch A. Integration of scheduling and control—theory or practice? *Comput Chem Eng*. 2009;33(12):1909–1918.
4. Baldea M, Harjunkoski I. Integrated production scheduling and process control: a systematic review. *Comput Chem Eng*. 2014;71:377–390.
5. Allgor RJ, Barton PI. Mixed-integer dynamic optimization I: problem formulation. *Comput Chem Eng*. 1999;23(4–5):567–584.
6. Flores-Tlacuahuac A, Grossmann IE. Simultaneous cyclic scheduling and control of a multiproduct CSTR. *Ind Eng Chem Res*. 2006; 45(20):6698–6712.
7. Nystrom RH, Franke R, Harjunkoski I, Kroll A. Production campaign planning including grade transition sequencing and dynamic optimization. *Comput Chem Eng*. 2005;29(10):2163–2179.
8. Nystrom RH, Harjunkoski I, Kroll A. Production optimization for continuously operated processes with optimal operation and scheduling of multiple units. *Comput Chem Eng*. 2006;30(3):392–406.
9. Zhuge J, Ierapetritou MG. Integration of scheduling and control with closed loop implementation. *Ind Eng Chem Res*. 2012;51(25):8550–8565.
10. Terrazas-Moreno S, Flores-Tlacuahuac A, Grossmann IE. Lagrangean heuristic for the scheduling and control of polymerization reactors. *AIChE J*. 2008;54(1):163–182.
11. Chu Y, You F. Integration of production scheduling and dynamic optimization for multi-product CSTRs: generalized Benders decomposition coupled with global mixed-integer fractional programming. *Comput Chem Eng*. 2013;58(11):315–333.
12. Chu Y, You F. Integrated scheduling and dynamic optimization of complex batch processes with general network structure using a generalized benders decomposition approach. *Ind Eng Chem Res*. 2013;52(23):7867–7885.
13. Nie Y, Biegler LT, Villa CM, Wassick JM. Discrete time formulation for the integration of scheduling and dynamic optimization. *Ind Eng Chem Res*. 2015;54(16):4303–4315.
14. Chu Y, You F. Integrated scheduling and dynamic optimization by stackelberg game: bilevel model formulation and efficient solution algorithm. *Ind Eng Chem Res*. 2014;53(13):5564–5581.
15. Zhuge J, Ierapetritou MG. Integration of scheduling and control for batch processes using multi-parametric model predictive control. *AIChE J*. 2014;60(9):3169–3183.
16. Pistikopoulos EN. Perspectives in multiparametric programming and explicit model predictive control. *AIChE J*. 2009;55(8):1918–1925.
17. Richter S, Jones CN, Morari M. Computational complexity certification for real-time mpc with input constraints based on the fast gradient method. *IEEE Trans Automat Control*. 2012;57(6):1391–1403.
18. Ferreau HJ, Bock HG, Diehl M. An online active set strategy to overcome the limitations of explicit MPC. *Int J Robust Nonlinear Control*. 2008;18(8):816–830.
19. Rao CV, Wright SJ, Rawlings JB. Application of interior-point methods to model predictive control. *J Optim Theory Appl*. 1998;99(3):723–757.
20. Wang Y, Boyd S. Fast model predictive control using online optimization. *IEEE Trans Control Syst Technol*. 2010;18(2):267–278.
21. Richter S, Jones CN, Morari M. Real-time input-constrained MPC using fast gradient methods. *Paper Presented at the Decision and Control, 2009 Held Jointly with the 2009 28th Chinese Control Conference*, Shanghai, China, 2009.
22. Zavala VM, Laird CD, Biegler LT. Fast implementations and rigorous models: can both be accommodated in NMPC? *Int J Robust Nonlinear Control*. 2008;18(8):800–815.
23. Lopez-Negrete R, D'Amato FJ, Biegler LT, Kumar A. Fast nonlinear model predictive control: Formulation and industrial process applications. *Comput Chem Eng*. 2013;51:55–64.
24. Flores-Tlacuahuac A, Grossmann IE. Simultaneous cyclic scheduling and control of tubular reactors: single production lines. *Ind Eng Chem Res*. 2010;49(22):11453–11463.
25. Nie Y, Biegler LT, Wassick JM. Integrated scheduling and dynamic optimization of batch processes using state equipment networks. *AIChE J*. 2012;58(11):3416–3432.
26. Chu Y, You F. Integration of scheduling and control with online closed-loop implementation: fast computational strategy and large-scale global optimization algorithm. *Comput Chem Eng*. 2012;47:248–268.
27. Johansson M. *Piecewise Linear Control Systems*. Berlin: Springer-Verlag, 2003.
28. Rodrigues L, How JP. Synthesis of piecewise-affine controllers for stabilization of nonlinear systems. *Paper Presented at the 42nd IEEE Conference on Decision and Control*, Maui, Hawaii, USA, 2003.
29. Rodrigues L, Boyd S. Piecewise-affine state feedback for piecewise affine slab systems using convex optimization. *Syst Control Lett*. 2005(54):835–853.
30. Ferrari-Trecate G, Muselli M, Liberati D, Morari M. A clustering technique for the identification of piecewise affine systems. *Automatica*. 2003;39(2):205–217.
31. Magnani A, Boyd S. Convex piecewise-linear fitting. *Optim Eng*. 2009;10(1):1–17.
32. Roll J, Bemporad A, Ljung L. Identification of piecewise affine systems via mixed-integer programming. *Automatica*. 2004;40(1):37–50.
33. Ohlsson H, Ljung L. Identification of piecewise affine systems using sum-of-norms regularization. *Paper presented at the 18th IFAC World Congress*, Milano, Italy, 2011.
34. Gegúndez ME, Aroba J, Bravo JM. Identification of piecewise affine systems by means of fuzzy clustering and competitive learning. *Eng Appl Artif Intell*. 2008;21(8):1321–1329.
35. Nakada H, Takaba K, Katayama T. Identification of piecewise affine systems based on statistical clustering technique. *Automatica*. 2005; 41(5):905–913.
36. Vasak M, Klanjcic D, Peric N. Piecewise affine identification of MIMO processes. *Paper Presented at the Computer Aided Control*

System Design, 2006 IEEE International Conference on Control Applications, Munich, 2006.
37. Buchan AD, Haldane DW, Fearing RS. Automatic identification of dynamic piecewise affine models for a running robot. *Paper Presented at the 2013 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS),* Tokyo, 2013.
38. Števek J, Szűcs A, Kvasnica M, Fikar M, Kozák Š. Two steps piecewise affine identification of nonlinear systems. *Arch Control Sci.* 2012;22(4):371–388.
39. Kvasnica M, Szucs A, Fikar M. Automatic derivation of optimal piecewise affine approximations of nonlinear systems. *Paper Presented at 18th IFAC World Congress*, Milano, Italy, 2011.
40. Julian P, Desages A, Agamennoni O. High-level canonical piecewise linear representation using a simplicial partition. *IEEE Trans Circuits Syst I Fundam Theory Appl.* 1999;46(4):463–480.
41. Zavieh A, Rodrigues L. Intersection-based piecewise affine approximation of nonlinear systems. *Paper Presented at the 21st Mediterranean Conference on Control & Automation*, Chania, 2013.
42. Casselman S, Rodrigues L. A New methodology for piecewise affine models using Voronoi partitions. *Paper Presented at the 48th IEEE Conference on Decision and Control, Held Jointly with the 28th Chinese Control Conference*, Shanghai, 2009.
43. Sontag ED. Nonlinear regulation: the piecewise linear approach. *IEEE Trans Automat Control.* 1981;26(2):346–358.
44. Wang Y, Boyd S. Fast model predictive control using online optimization. *Paper Presented at the Proceedings of the 17th IFAC World Congress*, Seoul, Korea, 2008.
45. Nesterov Y. A method for solving a convex programming problem with convergence rate O(1/k^2). *Sov Math Dokl.* 1983;27(2):372–376.
46. Domahidi A, Zgraggen AU, Zeilinger MN, Morari M, Jones CN. Efficient interior point methods for multistage problems arising in receding horizon control. *Paper Presented at IEEE 51st Annual Conference on the Decision and Control (CDC)*, Maui, Hawaii, USA, 2012.
47. Tøndel P, Johansen TA, Bemporad A. Evaluation of piecewise affine control via binary search tree. *Automatica.* 2003;39(5):945–950.
48. Jones CN, Grieder P, Raković SV. A logarithmic-time solution to the point location problem for parametric linear programming. *Automatica.* 2006;42(12):2215–2218.
49. Domahidi A. *FORCES: Fast Optimization for Real-time Control on Embedded Systems*, 2012. Available at: http://forces.ethz.ch.
50. Safonov MG. *Stability and Robustness of Multivariable Feedback Systems*. Cambridge, MA: MIT Press, 1980.
51. Camacho EF, Bordons Alba C. *Model Predictive Control.* London: Springer, 2007.
52. Löfberg J. YALMIP: A Toolbox for Modeling and Optimization in MATLAB. *Paper Presented at the IEEE International Symposium on Computer Aided Control System Design*, Taipei, Taiwan, 2004.

Appendix A

As for the method of PWA identification from a 3-D function $f(x^{(1)},x^{(2)},x^{(3)})$ we use cuboids partition as shown in Figure A1. The constraints and objective are similar to the case of 2-D. Here we include an addition dimension $x^{(3)}$. Index $k \in K = \{1,2,\ldots,N_k\}$ is associated with $x^{(3)}$ and the partition of $x^{(3)}$ is $x^{(3)}_{\text{int},0} \le x^{(3)}_{\text{int},1} \le \cdots \le x^{(3)}_{\text{int},N_k}$.

Define the valid regions and the PWA

$$\begin{aligned}\text{VR}_{i,j,k} = \{(x^{(1)},x^{(2)},x^{(3)}) | & x^{(1)}_{\text{int},i-1} \le x^{(1)} \le x^{(1)}_{\text{int},i}, x^{(2)}_{\text{int},j-1} \\ & \le x^{(2)} \le x^{(2)}_{\text{int},j}, x^{(3)}_{\text{int},k-1} \le x^{(3)} \le x^{(3)}_{\text{int},k}\}\end{aligned} \tag{A1}$$

$$\begin{aligned}\hat{f}_{i,j,k}(x^{(1)},x^{(2)},x^{(3)}) = a_{i,j,k}x^{(1)} + b_{i,j,k}x^{(2)} + c_{i,j,k}x^{(3)} + d_{i,j,k}, \\ \text{if } (x^{(1)},x^{(2)},x^{(3)}) \in \text{VR}_{i,j,k}\end{aligned} \tag{A2}$$

Constraints

Equation A3 describes the continuity at the intersection points shared by eight adjacent cuboids (for instance, point P in Figure A1).

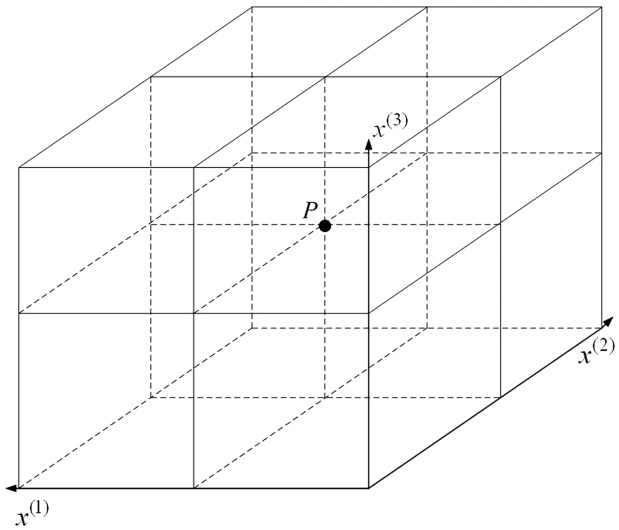

Figure A1. Cuboids partition of the domain of 3-D states.

$$\begin{aligned}&\hat{f}_{i,j,k}(x^{(1)}_{\text{int},i},x^{(2)}_{\text{int},j},x^{(3)}_{\text{int},k}) = \hat{f}_{i+1,j,k}(x^{(1)}_{\text{int},i},x^{(2)}_{\text{int},j},x^{(3)}_{\text{int},k}) \\ &= \hat{f}_{i,j+1,k}(x^{(1)}_{\text{int},i},x^{(2)}_{\text{int},j},x^{(3)}_{\text{int},k}) = \hat{f}_{i+1,j+1,k}(x^{(1)}_{\text{int},i},x^{(2)}_{\text{int},j},x^{(3)}_{\text{int},k}) \\ &= \hat{f}_{i,j,k+1}(x^{(1)}_{\text{int},i},x^{(2)}_{\text{int},j},x^{(3)}_{\text{int},k}) = \hat{f}_{i+1,j,k+1}(x^{(1)}_{\text{int},i},x^{(2)}_{\text{int},j},x^{(3)}_{\text{int},k}) \\ &= \hat{f}_{i,j+1,k+1}(x^{(1)}_{\text{int},i},x^{(2)}_{\text{int},j},x^{(3)}_{\text{int},k}) = \hat{f}_{i+1,j+1,k+1}(x^{(1)}_{\text{int},i},x^{(2)}_{\text{int},j},x^{(3)}_{\text{int},k})\end{aligned} \tag{A3}$$

More specifically

$$\begin{aligned}&a_{i,j,k}x^{(1)}_{\text{int},i} + b_{i,j,k}x^{(2)}_{\text{int},j} + c_{i,j,k}x^{(3)}_{\text{int},k} + d_{i,j,k} \\ &= a_{i+1,j,k}x^{(1)}_{\text{int},i} + b_{i+1,j,k}x^{(2)}_{\text{int},j} + c_{i+1,j,k}x^{(3)}_{\text{int},k} + d_{i+1,j,k} \\ &= a_{i,j+1,k}x^{(1)}_{\text{int},i} + b_{i,j+1,k}x^{(2)}_{\text{int},j} + c_{i,j+1,k}x^{(3)}_{\text{int},k} + d_{i,j+1,k} \\ &= a_{i+1,j+1,k}x^{(1)}_{\text{int},i} + b_{i+1,j+1,k}x^{(2)}_{\text{int},j} + c_{i+1,j+1,k}x^{(3)}_{\text{int},k} + d_{i+1,j+1,k} \\ &= a_{i,j,k+1}x^{(1)}_{\text{int},i} + b_{i,j,k+1}x^{(2)}_{\text{int},j} + c_{i,j,k+1}x^{(3)}_{\text{int},k} + d_{i,j,k+1} \\ &= a_{i+1,j,k+1}x^{(1)}_{\text{int},i} + b_{i+1,j,k+1}x^{(2)}_{\text{int},j} + c_{i+1,j,k+1}x^{(3)}_{\text{int},k} + d_{i+1,j,k+1} \\ &= a_{i,j+1,k+1}x^{(1)}_{\text{int},i} + b_{i,j+1,k+1}x^{(2)}_{\text{int},j} + c_{i,j+1,k+1}x^{(3)}_{\text{int},k} + d_{i,j+1,k+1} \\ &= a_{i+1,j+1,k+1}x^{(1)}_{\text{int},i} + b_{i+1,j+1,k+1}x^{(2)}_{\text{int},j} + c_{i+1,j+1,k+1}x^{(3)}_{\text{int},k} + d_{i+1,j+1,k+1}\end{aligned} \tag{A4}$$

Optimization problem

The objective is to minimize sum of squared errors (problem A5).

$$\min_{\substack{x^{(1)}_{\text{int},i}, x^{(2)}_{\text{int},j}, x^{(3)}_{\text{int},k}, \\ a_{i,j,k}, b_{i,j,k}, c_{i,j,k}, d_{i,j,k}}} \sum_i \sum_j \sum_k \left(\hat{f}(x^{(1)}_{\text{int},i},x^{(2)}_{\text{int},j},x^{(3)}_{\text{int},k}) - f(x^{(1)}_{\text{int},i},x^{(2)}_{\text{int},j},x^{(3)}_{\text{int},k})\right)^2$$

s.t.(A1)–(A4) (A5)

Manuscript received Jan. 26, 2015, and revision received June 18, 2015.