Available online at www.sciencedirect.com

ScienceDirect

Computers and Chemical Engineering 32 (2008) 2233–2242

Computers & Chemical Engineering

www.elsevier.com/locate/compchemeng

Population balance modeling in Simulink: PCSS

Jeffrey D. Ward [*], Cheng-Ching Yu

Department of Chemical Engineering, National Taiwan University, Taipei 106, Taiwan

Received 5 June 2006; received in revised form 17 October 2007; accepted 1 November 2007
Available online 15 January 2008

Abstract

In this work we develop, demonstrate, and distribute the code for a new Simulink block that models the dynamic evolution of the population density function for a physical system which can be modeled by a population balance equation. The name of the block is PCSS, for population balance modeling using the conservation element/solution element method in Simulink. The block interfaces with an auxiliary user-defined function that allows the user to specify arbitrary expressions for growth and generation/loss terms, as well as an arbitrary number of inputs, outputs and auxiliary states for the block. The versatility of the block allows a wide variety of physical systems to be modeled, and the implementation in Simulink facilitates rapid model development and permits the use of pre-existing MATLAB/Simulink packages for system identification and control.

Keywords: Population balance; Process dynamics; Process control; Crystallization

1. Introduction

Mathematical modeling by population balance has become a standard tool for chemical engineers, with applications as diverse as catalysis, particulate processes, and bioengineering. Hulbert and Katz (1964) are generally credited with being the first authors to publish a general formulation of the theory. The standard reference on the subject is the textbook by Ramkrishna (2000). Applications to crystallization are discussed by Randolph and Larson (1988).

A population balance for a solids process such as crystallization with one property coordinate can be described by the partial differential equation:

$$\frac{\partial F(L,t)}{\partial t} + \frac{\partial (G(F,L,t)F(L,t))}{\partial L} = \sum_i \dot{F}_i(F,L,t) \tag{1}$$

where $F(L, t)$ is the population density function, $G(F, L, t)$ is the linear growth rate of the crystals, and $\dot{F}_i$ are generation and loss terms that account for phenomena such as nucleation, attrition, flow into and out of the control volume, etc. In certain special cases, particularly batch processes, Eq. (1) can be solved by the method of moments (Chang & Epstein, 1982; Chung, Ma, & Braatz, 1999; Jones, 1974; Ward, Mellichamp, & Doherty, 2006 and references therein). However, in most realistic cases this method fails because the moments are not closed. In this case, it is necessary to solve Eq. (1) explicitly.

Historically, most researchers who have attempted to solve Eq. (1) have employed the so-termed method of lines, in which the differential equation is discretized into a finite number of ordinary differential equations (Hill & Ng, 1995; Hounslow, Ryall, & Marshall, 1988; Kumar & Ramkrishna, 1996a, 1996b, 1997; Marchal, David, Klein, & Villermaux, 1988; Pathath & Kienle, 2002). Other methods, including statistical methods, are reviewed by Ramkrishna (2000).

Wulkow and coworkers (Gerstlauer, Gahn, Zhou, Rauls, & Schreiber, 2006; Wulkow, Gerstlauer, & Nieken, 2001) have developed and marketed a commercial software package (Parsival) for the dynamic simulation of particle size distributions and flowsheets with solids processes. The software is based on the so-termed Galerkin h–p method (Wulkow, 1996), an adaptive finite element algorithm.

Recently, Motz, Mitrović, and Gilles (2002) proposed that the partial differential equation representing the population balance (Eq. (1)) could be solved using a novel numerical method called the space time conservation element and solution element (CE/SE) method developed by Chang and coworkers (Chang, Wang, & Chow, 1999; Lim, Chang, & Jørgensen, 2004; Yu & Chang, 1997). This method was originally developed for the

[*] Corresponding author. Fax: +886 2 2362 3040.
E-mail address: jeffward@ntu.edu.tw (J.D. Ward).

0098-1354/$ – see front matter © 2007 Elsevier Ltd. All rights reserved.
doi:10.1016/j.compchemeng.2007.11.001

solution of the Navier–Stokes and Euler equations, but can be modified to solve the population balance equation. The method is derived from first principles by enforcing local (and global) numerical flux conservation and, as such, it is well suited to the solution of partial differential equations where numerical diffusion is a concern. It has the further advantages that it is simple to program and has a simple stability criterion.

The issue of automatic control of continuous crystallization processes has also received increased attention recently in the literature. Historically, one significant difficulty in implementing feedback control for crystallization processes was the inability to make online measurements of the crystal size distribution. However, recent improvements in sensor technology have begun to mitigate this problem. Another difficulty is the selection of controlled variables, since the crystal size is a distributed parameter. Eek, Pouw, and Bosgra (1995) investigate the use of proportional-only feedback control. Vollmer and Raisch (2002) and Motz, Mitrović, Gilles, Vollmer, and Raisch (2003) consider H_∞ control. Christofides and coworkers (Chiu & Christofides, 1999; Chiu & Christofides, 2000; El-Farra, Chiu, & Christofides, 2001; Shi, El-Farra, Li, Mhaskar, & Christofides, 2006) have applied a variety of advanced control methodologies to crystallization processes, including robust nonlinear control and predictive control.

The most widely used commercial software program and high-level programming language in the field of automatic control in general and chemical process control specifically is MATLAB (The MathWorks, 2004) and the associated dynamic simulation package Simulink (see, for example, Seborg, Edgar, & Mellichamp, 2004). Various toolboxes have been developed for this platform for control-specific applications, such as model-predictive control, robust control, system identification, neural networking, fuzzy logic, etc. Therefore, it would be useful if it were possible for the engineer to simulate the dynamics of dispersed phase processes such as crystallization in this environment.

The purpose of this work is to develop, demonstrate and distribute the code for a Simulink block which models the time evolution of the particle size distribution function for a crystallization process. The code is based on the CE/SE algorithm, and refers to a user-defined auxiliary function to determine G and $\dot{F}_i$ as a function of L, t, F and other state variables which the user may define.

2. Computer code

The CE/SE algorithm is described in detail elsewhere, and the time marching scheme and its derivation will not be repeated here. Instead we make a few remarks about the Simulink implementation to guide the reader in modeling his or her own process. The MATLAB code for the PCSS block, as well as the auxiliary function files and Simulink block diagrams for the two case study processes, are provided in a web-published supplement to this manuscript with the file PCSS.zip. Some familiarity with the coding of Simulink S-Functions may be helpful in understanding the details presented in this section.

The heart of the PCSS block is the m S-Function PCSS.m This function need not be modified by the user, but it makes use of an auxiliary m function which the user must provide and which specifies all of the model-specific properties, including nucleation and growth rates, block inputs and outputs, etc. The auxiliary function can also establish and update auxiliary states for the block, such as liquid phase concentration, liquid holdup in the crystallizer, crystallizer temperature, etc. The auxiliary function has two inputs, the “block” data structure, and a flag which tells the function what information to return. The auxiliary function has a single output, a variable (usually a data structure) with the requested information.

The block is a discrete time block, with a time step equal to the parameter dt. All state information, including the current values of the crystal size distribution F, its length derivative FL, the growth rate G, and values of auxiliary state variables are stored in Dwork vectors 1 through 4, respectively. Since the entire block data structure is passed to the auxiliary function, the auxiliary function has access to all of the information known about the system, including the values of all states, all inputs, the sample time and the simulation time. Therefore, the user-defined function can be quite general.

Fig. 1 shows the flow of information between the PCSS code and the auxiliary function. Table 1 shows the required output from the auxiliary function. The reader is also advised to refer to the file pcss_ex_1.pdf, which is available as a web-published supplement to this manuscript. This file contains the code for the auxiliary function for the first case study, which is referenced by

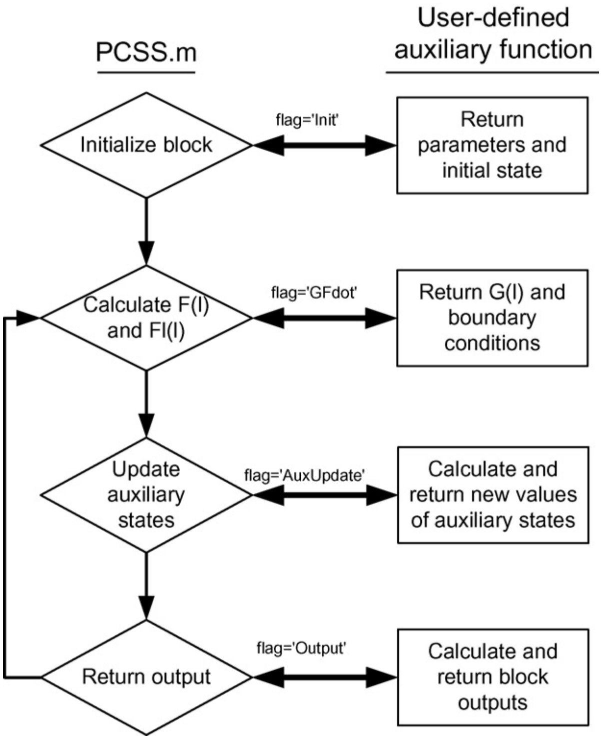
PCSS.m
User-defined auxiliary function
Initialize block
flag='Init'
Return parameters and initial state
Calculate F(l) and Fl(l)
flag='GFdot'
Return G(l) and boundary conditions
Update auxiliary states
flag='AuxUpdate'
Calculate and return new values of auxiliary states
Return output
flag='Output'
Calculate and return block outputs

Fig. 1. Flow diagram illustrating the exchange of information between the PCSS code and the user-defined auxiliary function.

Table 1
Required outputs from auxiliary function

Flag = 'Init'	
Output is a data structure with the following fields	
IPorts	Vector specifying the dimensions of input ports
OPorts	Vector specifying the dimensions of output ports
F0	Vector containing the initial values of the crystal size distribution
FL0	Vector containing the initial values of the length derivative of the crystal size distribution
G0	Vector containing the initial values of the growth rate
AuxInit	Vector containing the initial values of the auxiliary state variables
param	Vector containing the four parameters for the CE/SE method: length step size dL, time step size dt, and parameters α and ε as defined by Chang (1995) param = [dL dt α ε]
Flag = 'GFdot'	
Output is a data structure with the following fields	
G	Vector containing the growth rate at the next time step at each grid point
Fdot	Vector containing generation/loss rate at the next time step at each grid point
GBC	Growth rate at the left boundary, i.e. $G(L=0)$
FBC	Left boundary condition for the population density function at the current time, i.e. $F(L=0)$
Flag = 'AuxUpdate'	
Output is a vector containing new calculated values for the state variables at the current time step	
Flag = 'Output'	
Output is a data structure of length N where N is the number of outputs. The data structure has one field, named 'Data' which specifies the output (scalar or vector) for each port	

line number in this section. More details about the case study can be found in the following section.

If the flag is equal to "Init", then the auxiliary block should return information necessary for the initialization of the block. In the code for Example 1, this occurs on lines 33–45. The output is a data structure 'out' with several fields. The field 'IPorts' specifies the number and dimensions of the input ports. In this example, there is one input port with length 1 (this will be the batch temperature). Likewise, the field 'OPorts' specifies the number and dimensions of the output ports. There are two output ports. The first output is a vector containing the value of the crystal size distribution at the discrete intervals dL; the second output is a scalar: the concentration of solute in the liquid phase. The fields 'F0' and 'FL0' give the initial values of the crystal size distribution and its length derivative FL at the discrete points dL. In this example, the initial condition is a normal distribution with a mean length of 200 μm, a standard deviation of 50 μm, and a total number of crystals equal to 10^5. The field 'G0' gives the initial values of the growth rate at the discrete points dL. The field AuxInit gives initial values of the auxiliary state variables. In Example 1, there are two auxiliary state variables, the liquid phase concentration and the third moment of the crystal size distribution.

If the flag is equal to "GFdot" then the function should return a data structure with three vectors corresponding to the growth rate and the generation/loss rate at each grid point ('G' and 'Fdot') as well as a value for the growth rate and crystal size distribution at the left boundary condition ('GBC', 'FBC'). This occurs on lines 54–68 of the example code.

If the flag is equal to "AuxUpdate" the block should return a vector containing updated values of the auxiliary state variables. This occurs on lines 72–78 of the example code.

Finally, if the flag is equal to "Output" then the block should return a data structure of length n, where n is the number of output ports defined in the initialization step. The data structure should have one field named 'Data' which should contain a scalar or vector with the output for the corresponding port. This occurs on lines 47–50 of the example code. There are two outputs in Example 1; therefore, the data structure has length 2. The first output is the crystal size distribution which is stored in Dwork(1).Data. The second output is the liquid phase concentration, which is stored in Dwork(4).Data(1). (Dwork(4) contains all auxiliary states, and the concentration was specified as the first auxiliary state.)

It is worthwhile to consider some of the limitations of the block. These limitations arise both from inherent limitations in the numerical method and from decisions that we have made in implementing the block in Simulink. One important limitation is the stability criteria for the numerical method: $G\Delta T/\Delta L \leq 1$ at all times (Motz et al., 2002). This has the effect of limiting the magnitude of the time interval Δt for a given value of the resolution of the crystal size distribution ΔL. Another limitation on the Simulink block is that as written it can accommodate only a single property coordinate, whereas some researchers (e.g. Ma & Braatz, 2003) consider multiple property coordinates (such as two characteristic lengths with two independent growth rates). Chang et al. (1999) describe how the CE/SE method can be applied to solve multi-dimensional problems. Another limitation of the code is that as implemented in this work the space–time grid must be uniform, whereas it might be desirable in some circumstances to use a non-uniform grid with (for example) a finer grid at lower values of the length coordinate or in the vicinity of a discontinuity such as the fines or product cutoff in the R–z crystallizer model (Randolph & Larson, 1988). Chang, Wu, Wang, and Yang (2000) discuss a method for local mesh refinement in the CE/SE method.

The following sections present results from two examples that use the PCSS block.

3. Example 1: Batch crystallization

3.1. Mathematical model and numerical parameters

The first example is based on a case study by Chung et al. (1999) for the seeded batch cooling crystallization of potassium nitrate from water. Nucleation is assumed to be by a secondary mechanism, and the nuclei are assumed to be arbitrarily small. The population balance is given by:

$$\frac{\partial F(L,t)}{\partial t} + G(t)\frac{\partial F(L,t)}{\partial L} = 0 \tag{2}$$

subject to the initial condition:

$$F(L,0) = F_0 \tag{3}$$

and the boundary condition:

$$F(0,t) = \frac{B(t)}{G(t)} \tag{4}$$

The growth and nucleation rates are given by:

$$G(t) = k_g S(t)^g \tag{5}$$

$$B(t) = k_b \mu_3 S(t)^b \tag{6}$$

where k_g, k_b, g and b are empirically determined kinetic parameters. μ_3 is the third moment of the crystal size distribution as defined below. The supersaturation is defined as:

$$S = \frac{C - C_{\text{sat}}}{C_{\text{sat}}} \tag{7}$$

where C_{sat} is a function of temperature. Finally, the time evolution of the concentration of solute in the batch is given by a material balance:

$$\frac{\mathrm{d}C}{\mathrm{d}t} = -3\rho_{\text{c}} k_{\text{v}} G(t)\mu_2 \tag{8}$$

Parameters for the potassium nitrate case study are given in Table 2. The moments μ_i of the population density function F are given by:

$$\mu_i(t) = \int_0^\infty L^i F(L,t)\,\mathrm{d}L \tag{9}$$

Table 2
Parameter values for the KNO_3 case study system

Growth rate constant	k_g	1.1612×10^{-4}	m/s
Nucleation rate constant	k_b	4.6401×10^{11}	$\#/\text{m}^3\text{s}$
Growth rate exponent	g	1.32	—
Nucleation rate exponent	b	1.78	—
Saturation concentration	C_{sat}	$0.1286 + 0.00588T + 0.000172T^2$	kg KNO_3/kg H_2O
Volumetric shape factor	k_{v}	1	—
Density of crystal	ρ_{c}	2110	kg/m^3

For this system the moments are closed and if complete knowledge of the time evolution of the population density function is not required, it is possible to obtain approximate information by solving the ordinary differential equations corresponding to the first several moments:

$$\frac{\mathrm{d}\mu_0}{\mathrm{d}t} = B \tag{10}$$

$$\frac{\mathrm{d}\mu_i}{\mathrm{d}t} = iG\mu_{i-1}, \quad i = 1, 2, \ldots \tag{11}$$

Therefore, the time evolution of the moments can be calculated independently by the integration of Eqs. (8), (10) and (11), and this result can be compared to the time evolution of the particle size distribution as determined by solving Eqs. (12) and (8) using the PCSS Simulink block.

3.2. Results and discussion

Ward et al. (2006) introduce the concepts of “early growth” and “late growth” for the selection of operating policies for seeded batch crystallization where secondary nucleation is important (Eq. (6)). Briefly, under an early-growth operating policy the growth rate (supersaturation) is greatest at the beginning of the batch, and under a late growth operating policy it is greatest at the end of the batch. For cooling crystallization, the rate of change of temperature is greatest in an absolute sense at the beginning of a batch for early growth and at the end of the batch for late growth. Ward et al. predict that early-growth operating policies should produce a smaller *number* of nucleus-grown crystals, but a greater *mass* of nucleus-grown crystals. A late-growth operating policy should produce a greater number but a smaller mass of nucleus-grown crystals and at the same time grow the seed crystals to a larger size. The reader is referred to Ward et al. (2006) for a detailed explanation of why this should be the case. This case study will develop further insight into this result by providing the complete population density function at the end of a batch for a seeded batch crystallizer subjected to early-growth and late-growth operating policies.

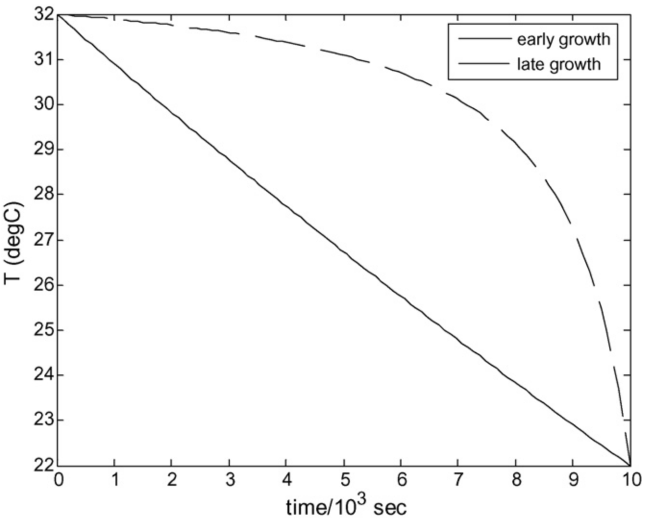

Fig. 2. Early-growth and late-growth temperature trajectories.

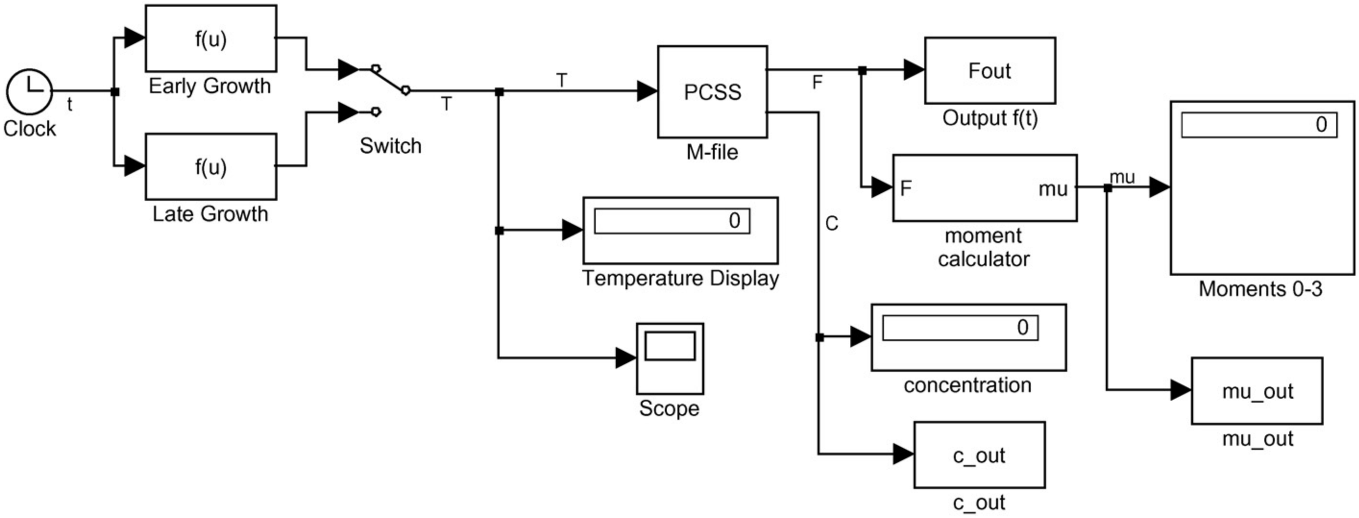

Fig. 3. Simulink block diagram for the KNO_3 process.

Fig. 2 shows the temperature trajectories used in this case study. Note that the early-growth trajectory has a slope greatest in an absolute sense at the beginning of the batch while the late-growth trajectory has a slope steepest in an absolute sense at the end of the batch. Although the early-growth trajectory is nearly linear, it results in a significantly greater supersaturation at the beginning of the batch than at the end of the batch. Fig. 3 shows the Simulink flowsheet that was developed to model this process. The PCSS block has one input for the temperature and two outputs: a vector output for the discretized population density function and a scalar output for the concentration.

Fig. 4 shows the time evolution of the first four moments of the population density function for the early- and late-growth operating policies as calculated using both the PCSS block and the method of moments. The lines overlap almost exactly, suggesting that the PCSS block gives a very accurate description of the time evolution of the population density function. Note that the moments grow rapidly at the end of the batch for the late-growth operating policy, but at the beginning of the batch for the

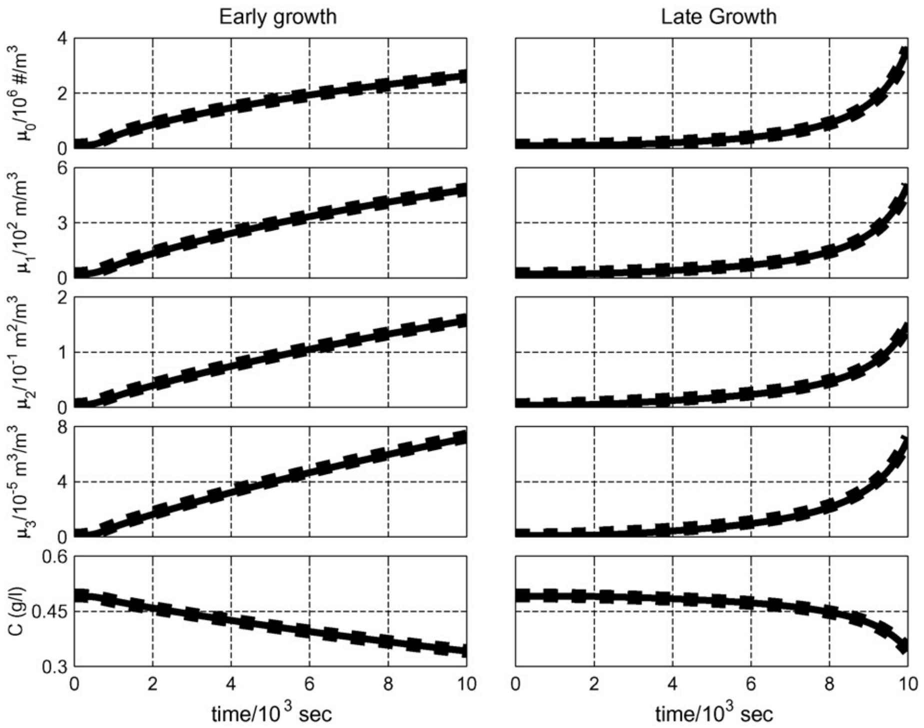

Fig. 4. Time evolution of the solute concentration and the first four moments of the crystal size distribution function for early and late-growth operating policies as calculated by using: (—) PCSS (▬ ▬ ▬) method of moments. The lines lie almost on top of each other.

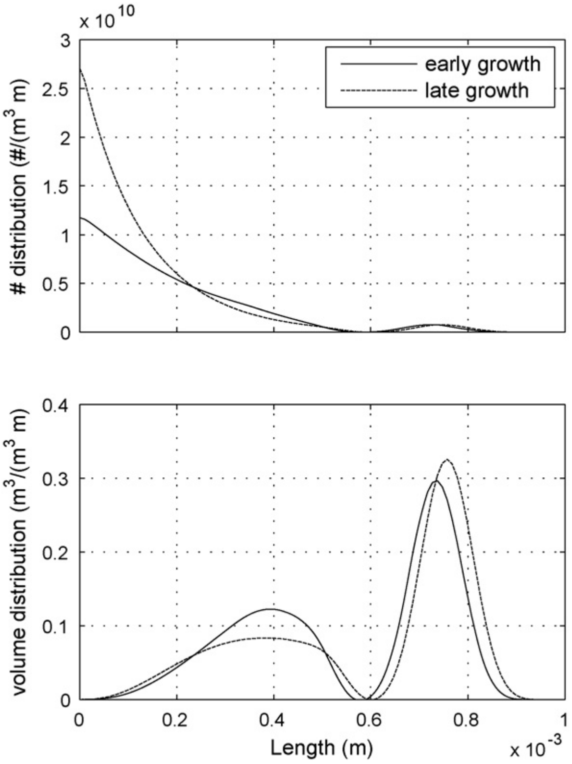

Fig. 5. Number and volume distribution functions for the KNO_3 process subjected to early- and late-growth operating policies.

early-growth operating policy. Note also that the 0th moment is considerably higher at the end of the batch for the late-growth operating policy compared with the early-growth operating policy.

Fig. 5 shows the population density function at the end of the batch for both the early and late-growth operating policies as calculated using the PCSS block. Note that this information cannot be obtained from only the low moments of the crystal size distribution (although the approximate shape of the CSD can be recovered if a large number of moments are calculated). The results shown in Fig. 5 are in agreement with the result published previously by Ward et al. (2006). Consider the lower graph, which shows the volume distribution. The peak near 400 μm corresponds to nucleus-grown crystals, while the peak near 750 μm corresponds to the seed-grown crystals. The integral under these peaks gives the total volume (proportional to mass) of the crystals grown from nuclei and seeds, respectively. From this graph, the late-growth operating policy would seem to be better, because more crystal mass is found in the seed-grown crystals (compared to the early-growth operating policy), and the seeds have grown to a larger size. However, the upper graph, which shows the number distribution function at the end of the batch, would suggest that the early growth operating policy is preferable because the *number* of nuclei produced by the early-growth operating policy is much less. Which of these is actually preferable will depend on the application. Again our primary purpose here is to demonstrate the efficacy of the PCSS Simulink block, and the reader is referred to Ward et al. (2006) for a more detailed discussion of late-growth and early-growth operating policies.

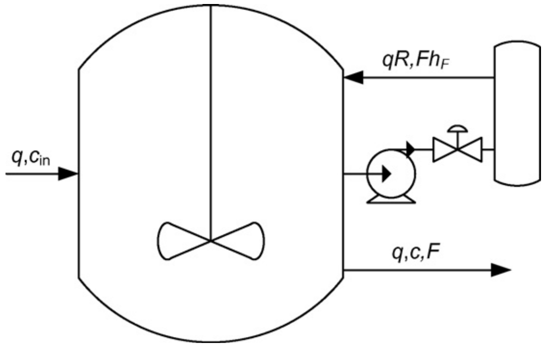

Fig. 6. Process flow diagram for Example 2.

4. Example 2: Continuous crystallizer with fines removal

4.1. Mathematical model and numerical parameters

This example is based on a case study system presented by Pathath and Kienle (2002) for the continuous crystallization of potassium chloride from water. The process flow diagram is shown in Fig. 6. Fresh feed enters the reactor with a volumetric flow rate q and concentration c_{in}. Product with liquid phase concentration c, and particle size distribution F is removed at the same flow rate q. A fines dissolution system is also employed. Material is withdrawn from the reactor at a flow rate equal to Rq where R is the fines dissolution ratio. The fines dissolver is modeled by assuming that all particles smaller than the fines cutoff size L_{f} are dissolved, and other crystals pass through unchanged.

The mathematical expression for the population balance is:

$$\frac{\partial F(L,t)}{\partial t} + G(t)\frac{\partial F(L,t)}{\partial L} = -\frac{q}{V}(1 + h_{\text{f}}(L))F(L,t) \qquad (12)$$

subject to the boundary condition:

$$F(0,t) = \frac{B(t)}{G(0,t)} \qquad (13)$$

V is the crystallizer volume, and h_{f} is the fines dissolution function, which is given by:

$$h_{\text{f}}(L) = R(1 - h(L - L_{\text{f}})) \qquad (14)$$

where h is the unit step function. The nucleation and growth rates are empirical functions of the concentration driving force:

$$G = k_g(c - c_{\text{sat}})^g \qquad (15)$$

$$B = k_b(c - c_{\text{sat}})^b \qquad (16)$$

Finally, a material balance on the solute in the crystallizer gives:

$$M_{\text{A}}\frac{\text{d}c}{\text{d}t} = \frac{q(\rho - M_{\text{A}}c)}{V} + \frac{\rho - M_{\text{A}}c}{\varepsilon}\frac{\text{d}\varepsilon}{\text{d}t} + \frac{q(M_{\text{A}}c_{\text{in}} - \rho)}{V\varepsilon} \qquad (17)$$

where ε is the void fraction given by:

$$\varepsilon = 1 - k_{\text{v}}\int_0^\infty FL^3\,\text{d}L \qquad (18)$$

Table 3
Parameter values for the KCl case study system

Growth rate constant	k_g	3.0513×10^{-2}	mm L/min mol
Nucleation rate constant	k_b	8.357×10^{9}	L^3/min mol^4
Growth rate exponent	g	1	–
Nucleation rate exponent	b	4	–
Saturation concentration	c_s	4.038	mol/L
Feed rate (base case)	q	0.035	L/min
Fines removal cut size	L_F	0.3	mm
Crystal density	ρ_c	1989	g/L
Molar mass KCl	M	74.551	g/mol
Crystallizer volume	V	10.5	L
Recycle ratio (base case)	R	2	–

and:

$$\frac{d\varepsilon}{dt} = -3k_v G \int_0^\infty FL^2\, dL + \frac{q}{V} k_v \int_0^\infty (1 + h_f(L)) FL^3\, dL \qquad (19)$$

The reader is referred to the paper by Pathath and Kienle (2002) for the complete derivation of Eqs. (17)–(19). Numerical values of the physical parameters are given in Table 3. Fig. 7 shows the Simulink block diagram for the process.

4.2. Results

Fig. 8 shows the steady state distribution of the 0th and 3rd moments of the crystal size distribution at steady state calculated using two different methods. The dashed line shows the steady state result for PCSS Simulink block, and the solid line shows the result when Eq. (12) is converted to an ordinary differential equation (by setting time derivatives equal to zero) and solved using the standard MATLAB ODE solving routine ode45. The agreement is excellent, suggesting that the PCSS block provides an accurate solution to the population balance equation.

In order to demonstrate the utility of solving the population balance equation in the MATLAB/Simulink environment, we simulated the automatic control of the process using two different control methodologies: conventional PI feedback control and model-predictive control. In both cases, the manipulated variable was the fines dissolution ratio R and the controlled variable was the third moment of the crystal size distribution μ_3. A 2-min lag was assumed for the measurement of μ_3. The nominal steady state chosen for investigation was with the fines dissolution ratio $R = 2$, which corresponds to $\mu_3 = 0.0451$ mm^3/mm^3. The process was found to be underdamped but stable at this operating point. In addition to a change in the setpoint for μ_3, two disturbances

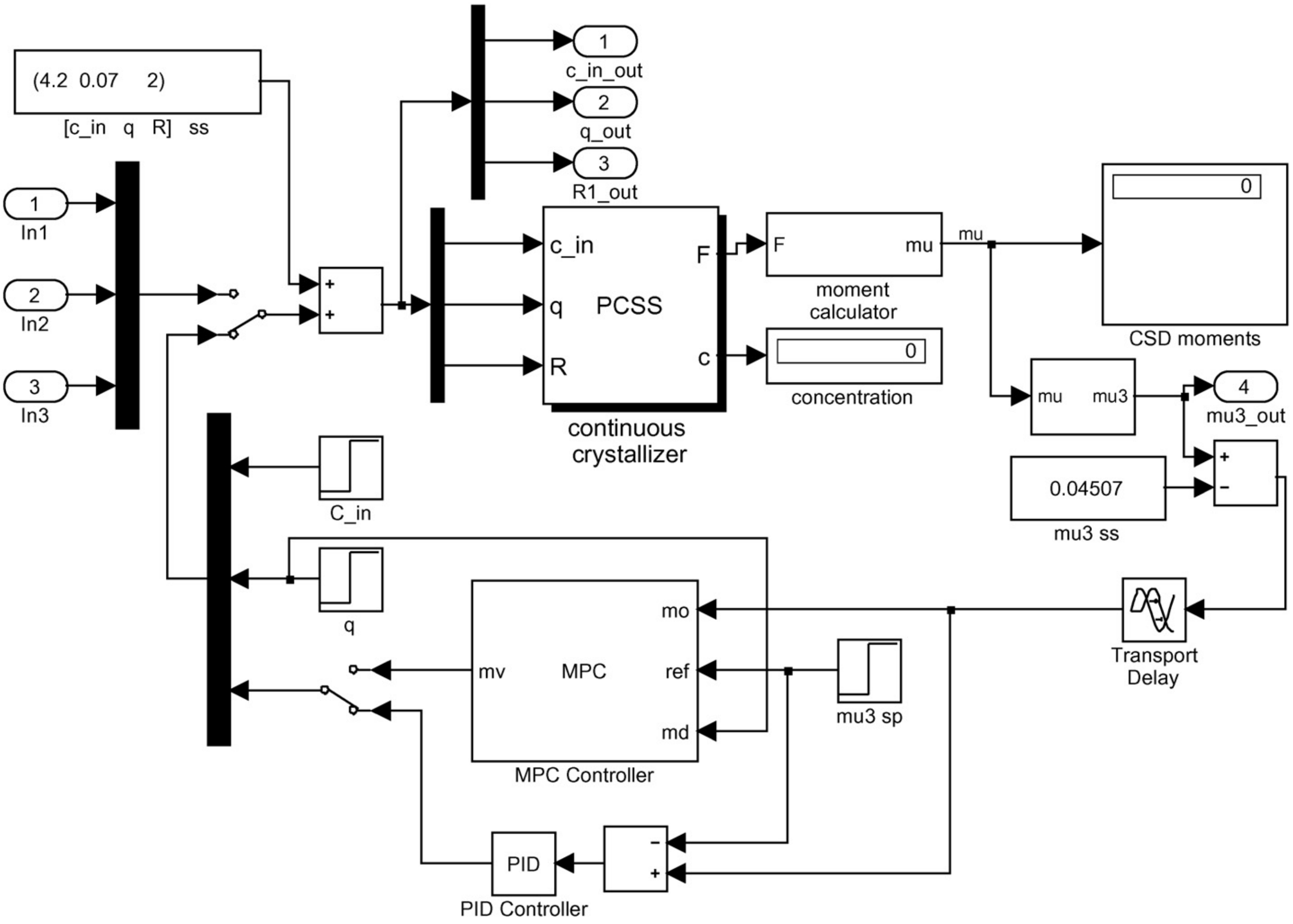

Fig. 7. Simulink block diagram for Example 2.

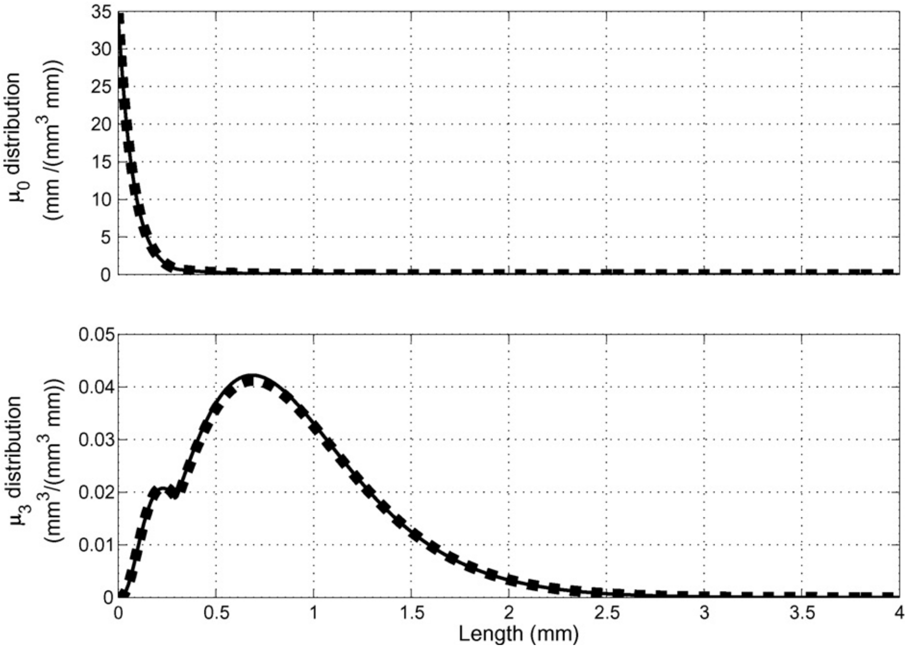

Fig. 8. Comparison of the steady-state distribution of the first and third moments as calculated using (—) PCSS (▬ ▬ ▬) integrating the ODE corresponding to the steady state. The lines lie almost on top of one another.

were considered: a step change in the feed flow rate q and a step change in the feed solute concentration c_{in}. For the case of model-predictive control, the feed flow rate was considered to be a measured disturbance, while the feed concentration was considered to be an unmeasured disturbance.

For PI control, initial values of the tuning parameters were determined by investigation of the step response (not shown), and these were further refined by investigation of the closed-loop response to disturbances and set-point changes. This resulted in a dimensional proportional gain of $1.4\,(\text{m}^3/\text{m}^3\,\text{m})^{-1}$ and an integral gain of $1.2\,(\text{m}^3/\text{m}^3\,\text{m}\,\text{min})^{-1}$ For MPC, a state-space model of the process was identified by exciting the process with pseudo-random binary data generated using the MATLAB System Identification Toolbox function idinput, and analyzing the results using the function n4sid. A model of order 6 (6 state variables) was determined to represent a judicious tradeoff between model complexity and accuracy based on the singular values of the Hankel matrices for models of various orders as calculated by n4sid. A model-predictive controller was designed using the MPC Toolbox GUI and implemented in Simulink using the MPC Simulink block. The controller was tuned manually by investigation of the dynamic responses. The resulting controller had a sampling time of 50 min, a prediction horizon of 10, a control horizon of 2, an output variables weight of 1, and a manipulated variables weight of 0.01.

Figs. 9–11 compare the dynamic response of the process for three different scenarios: a 20% increase in the feed flow rate (Fig. 9), an increase of 0.01 mol/L in the feed concentration (Fig. 10) and a 4.5% increase in the setpoint for μ_3 (Fig. 11). Note that while the magnitude of the step change in feed concentration seems small compared to the steady state value (4.2 mol/L), in fact it is a significant disturbance because it is the *difference* between the feed concentration and the saturation concentration (4.04 mol/L) which provides the driving force for crystal growth. In general, the model-predictive controller outperformed the PI controller, having both a faster initial response and a less oscillation. This is not unexpected because the MPC has considerably more information about the process available to it than does the PI controller. However, it is not possible to make a definitive statement about which control methodology is better, because there is no guarantee that either of the controllers has been tuned in an optimal manner. Furthermore, the process engineer must make a judgment about whether the additional effort required to identify a process model and implement MPC is justified by the improvement in dynamic performance.

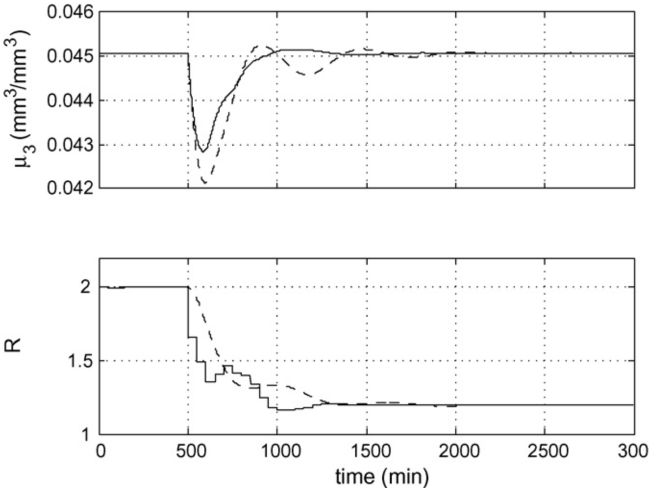

Fig. 9. Dynamic response of the continuous crystallizer to a 20% step increase in the feed flow rate q for PI control (- -) and MPC (—).

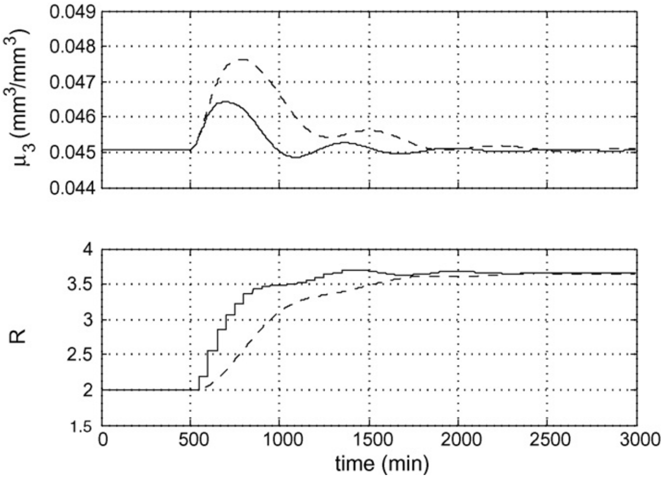

Fig. 10. Dynamic response of the continuous crystallizer to a 0.01 mol/L step increase in the feed concentration c_{in} for PI control (- -) and MPC (—).

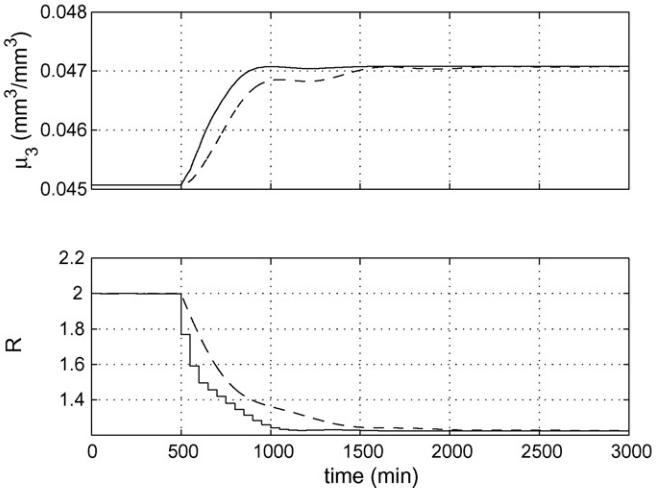

Fig. 11. Dynamic response of the continuous crystallizer to a 4.5% step increase in the set-point for mu$_3$ for PI control (- -) and MPC (—).

5. Conclusions

In this contribution, we have described a new software routine for the solution of population balances in Simulink, with applications to crystallization. We have provided the code freely as a web-published supplement to this manuscript, and it is our hope that other researchers will be able to use the code to model population balance systems quickly and accurately, particularly where it would be useful to employ other MATLAB/Simulink Toolboxes/routines, etc. as part of the research. Our own research group will use the code to investigate more complex crystallization control problems in future publications.

Acknowledgements

This work was supported by the National Science Council of Taiwan and the Fulbright program for US students.

Appendix A. Supplementary data

Supplementary data associated with this article can be found, in the online version, at doi:10.1016/j.compchemeng.2007.11.001.

References

Chang, S.-C. (1995). The method of space-time conservation element and solution element—A new approach for solving the Navier–Stokes and Euler equations. *Journal of Computer Physics*, *119*, 295–324.

Chang, C. T., & Epstein, M. A. F. (1982). Identification of batch crystallization control strategies using characteristic curves. In M. A. F. Epstein (Ed.), *Nucleation, Growth and Impurity Effects in Crystallization Process Engineering*. New York: AIChE.

Chang, S.-C., Wang, X.-Y., & Chow, C.-Y. (1999). The space-time conservation element and solution element method: A new high-resolution and genuinely multidimensional paradigm for solving conservation laws. *Journal of Computer Physics*, *156*, 89–136.

Chang, S.-C., Wu, Y., Wang, X.-Y., & Yang, V. (October 2000). Local mesh refinements in the space-time CE/SE method. *NASA/TM 2000-210516*.

Chiu, T. Y., & Christofides, P. D. (1999). Nonlinear control of particulate processes. *AIChE Journal*, *45*(6), 1279–1297.

Chiu, T. Y., & Christofides, P. D. (2000). Robust control of particulate processes using uncertain population balances. *AIChE Journal*, *46*(2), 266–280.

Chung, S. H., Ma, D. L., & Braatz, R. D. (1999). Optimal seeding in batch crystallization. *Canadian Journal of Chemical Engineering*, *77*, 590–596.

Eek, R. A., Pouw, H. A. A., & Bosgra, O. H. (1995). Design and experimental evaluation of stabilizing feedback controllers for continuous crystallizers. *Powder Technology*, *82*, 21–35.

El-Farra, N. H., Chiu, T. Y., & Christofides, P. D. (2001). Analysis and control of particulate processes with input constraints. *AIChE Journal*, *47*(8), 1849–1865.

Gerstlauer, A., Gahn, C., Zhou, H., Rauls, M., & Schreiber, M. (2006). Application of population balances in the chemical industry—Current status and future needs. *Chemical Engineering Science*, *61*, 205–217.

Hill, P. J., & Ng, K. M. (1995). New discretization procedure for the breakage equation. *AIChE Journal*, *41*(5), 1204–1216.

Hounslow, M. J., Ryall, R. L., & Marshall, V. R. (1988). A discretized population balance for nucleation, growth, and aggregation. *AIChE Journal*, *34*(11), 1821–1832.

Hulbert, H. M., & Katz, S. (1964). Some problems in particle technology: A statistical mechanical formulation. *Chemical Engineering Science*, *19*, 555–574.

Jones, A. G. (1974). Optimal operation of a batch cooling crystallizer. *Chemical Engineering Science*, *29*, 1075–1087.

Kumar, S., & Ramkrishna, D. (1996a). On the solution of population balance equations by discretization. I. A fixed pivot technique. *Chemical Engineering Science*, *51*(8), 1311–1332.

Kumar, S., & Ramkrishna, D. (1996b). On the solution of population balance equations by discretization. II. A moving pivot technique. *Chemical Engineering Science*, *51*(8), 1333–13342.

Kumar, S., & Ramkrishna, D. (1997). On the solution of population balance equations by discretization. III. Nucleation, growth, and aggregation of particles. *Chemical Engineering Science*, *52*(24), 4659–4679.

Lim, Y.-I, Chang, S.-C, & Jørgensen, S. B. (2004). A novel partial differential algebraic equation (PDAE) solver: Iterative space-time conservation element/solution element (CE/SE) method. *Computer Chemical Engineering*, *28*, 1309–1324.

Marchal, P., David, R., Klein, J. P., & Villermaux, J. (1988). Crystallization and precipitation engineering. I. An efficient method for solving population balance in crystallization with agglomeration. *Chemical Engineering Science*, *43*(1), 59–67.

Ma, D. L., & Braatz, R. D. (2003). Robust identification and control of batch processes. *Computer Chemical Engineering*, *43*, 1175–1184.

MATLAB version 7.0.1. (2004). *The Mathworks*, *Inc*.

Motz, S., Mitrović, A., & Gilles, E.-D. (2002). Comparison of numerical methods for the simulation of dispersed phase systems. *Chemical Engineering Science*, *57*, 4329–4344.

Motz, S., Mitrović, A., Gilles, E.-D., Vollmer, U., & Raisch, J. (2003). Modeling, simulation and $H_{\propto}$-control of an oscillating continuous crystallizer with fines dissolution. *Chemical Engineering Science*, *58*, 3473–3488.

Pathath, P. K., & Kienle, A. (2002). A numerical bifurcation analysis of nonlinear oscillations in crystallization processes. *Chemical Engineering Science*, *57*, 4391–4399.

Ramkrishna, D. (2000). *Population balances*. San Diego: Academic Press.

Randolph, A. D., & Larson, M. A. (1988). *The theory of particulate processes*. San Diego: Academic Press.

Seborg, D. E., Edgar, T. F., & Mellichamp, D. A. (2004). *Process dynamics and control* (2nd ed.). Hoboken, NJ: John Wiley & Sons.

Shi, D., El-Farra, N. H., Li, M., Mhaskar, P., & Christofides, P. D. (2006). Predictive control of particle size distribution in particulate processes. *Chemical Engineering Science*, *61*, 268–281.

Vollmer, U., & Raisch, J. (2002). Population balance modeling and H_{∞}-controller for a crystallization process. *Chemical Engineering Science*, *57*, 4401–4414.

Ward, J. D., Mellichamp, D. A., & Doherty, M. F. (2006). Choosing an operating policy for seeded batch crystallization. *AIChE Journal*, *52*, 2046–2054.

Wulkow, M. (1996). The simulation of molecular weight distributions in polyreaction kinetics by discrete Galerkin methods. *Macromolecular Theory and Simulations*, *5*(3), 393–416.

Wulkow, M., Gerstlauer, A., & Nieken, U. (2001). Modeling and simulation of crystallization processes using Parsival. *Chemical Engineering Science*, *56*, 2575–2588.

Yu, S.-T., & Chang, S.-C. (1997). Treating Stiff source terms in conservation laws by the method of space-time conservation element and solution element. In *Proceedings of the sixteenth international conference on numerical methods in fluid dynamics*