ADCHEM 2006

International Symposium on Advanced Control of Chemical Processes
Gramado, Brazil – April 2-5, 2006

STABILITY AND CONTROLLABILITY OF BATCH PROCESSES

B. Srinivasan[1] **and D. Bonvin**[2]

[1] *Department of Chemical Engineering*
École Polytechnique Montreal, Montreal, Canada H3C 3A7
[2] *Laboratoire d'Automatique*
École Polytechnique Fédérale de Lausanne
CH-1015 Lausanne, Switzerland

Abstract: Improving the performance of batch processes requires tools that are tailored to the specificities of batch operations. These include a mathematical representation that explicitly shows the two independent time variables (the run time t and the run index k) as well as the two types of outputs (the run-time and run-end outputs). Furthermore, corrective action can be taken via both on-line and run-to-run control. This paper investigates the important notions of stability and controllability for batch processes, where it is shown that a value rather than a yes-no answer needs to be considered. The tools required for evaluating these properties are readily adapted from the literature. Finally, the various control strategies are illustrated via the simulation of a semi-batch reactor, and references are made to the appropriate tools for evaluating stability and controllability.

Keywords: Batch Processes, Repetitive Processes, On-line Control, Run-to-run Control, Stability, Controllability.

1. INTRODUCTION

The majority of control studies in the literature have dealt with continuous processes operating around around an equilibrium point. In recent years, however, the class of systems where the process terminates in finite time has received increasing attention. An interesting feature is the fact that most of these processes are repeated over time. Many industrial operations, especially in the areas of batch chemical production, mechanical machining, and semiconductor manufacturing do fall under this category.

In a batch process, operations proceed from an initial state to a very different final state. Hence, there exists no single operating point around which the control system can be designed (Bonvin 1998). Also, since batch processing is characterized by the frequent repetition of batch runs, it is appealing to use the results from previous runs to improve the operation of subsequent ones. This has generated the industrially relevant topic of run-to-run control and optimization (Campbell *et al.* 2002, Francois *et al.* 2005). Repetition provides additional degrees of freedom for meeting the control objectives since the work does not necessarily have to be completed in a single run but can be distributed over several runs. This brings into picture an additional type of outputs that need to be controlled, the run-end outputs. The main difficulty is that these outputs are typically only available at the end of the run.

Though a lot of work has been reported recently in the literature on batch process control and optimization (Abel *et al.* 2000, Srinivasan *et al.* 2003, Flores-Cerrillo and MacGregor 2003, Chin *et al.* 2004), there is still a lack of understanding of their system-theoretical properties. Due to the finite-time nature of batch processes, the standard definitions of properties such as stability, controllability and observability cannot be used.

This paper presents definitions and analysis tools for the two important properties of stability and controllability for batch processes. It is important to emphasize that the contribution of this paper is in discussing the various notions of stability and controllability and choosing the right notions for the analysis of batch processes. The analysis tools are then readily adapted from those existing in the literature.

The paper is organized as follows. Section 2 introduces a brief mathematical description of batch processes and discusses the implications of two time scales and two types of output for control. Stability and controllability are analyzed in Sections 3 and 4, respectively. An illustrative example is presented in Section 5, and conclusions are drawn in Section 6.

2. CONTROL OF BATCH PROCESSES

A batch process can be seen as a repetitive dynamical process that is characterized by the presence of a finite terminal time and thus the possibility of having several sequential runs, with each run being dynamic. Batch processes have the following main characteristics: (i) There are two time scales, i.e. the continuous time within the run and the discrete run index k, (ii) the time of a run is limited (finite), (iii) there is no steady-state operating point with respect to t, i.e. the analysis has to be performed around trajectories rather than an equilibrium point, and (iv) two types of measurements are available, i.e. during the run and at the end of the run.

2.1 Terminology and notations

Let $\mathbb{R}$ be used for the space of real numbers and $\mathbb{L}$ for that of functions, and let $\mathbb{Z}_+$ represent the set of positive integers excluding zero. The various elements of a batch process can be defined as follows:

(1) *Run:* One realization of a repetitive process.
(2) *Run time:* The time within a run, t $[0, T]$ $\mathbb{R}_+$, where T is the finite terminal time.
(3) *Run index:* The number of a run, k $\mathbb{Z}_+$.
(4) *Inputs:* The inputs $u_k(t)$ U $\mathbb{R}^m$, evolve with t during run k. The input trajectories for run k are denoted by $u_k[0, T]$ $\mathbb{L}^m$.
(5) *States:* The states, $x_k(t)$ X $\mathbb{R}^n$, evolve with t during run k. x_k^{ic} are the initial conditions at time $t = 0$. The corresponding state trajectories are denoted by $x_k[0, T]$ $\mathbb{L}^n$.
(6) *Outputs:* The outputs are of two types: (i) The run-time outputs, $y_k(t)$ $\mathbb{R}^p$, correspond to the on-line measurements during run k; (ii) the run-end outputs, z_k $\mathbb{R}^q$, include the measurements that become available at the end of run k. The latter might also depend on the state evolution during the entire run, e.g. the average value of a state.
(7) *System dynamics:* They describe the state and output evolutions for a single run. For example, the nonlinear time-invariant model describing the process behavior during run k reads:

$$\dot{x}_k(t) = F(x_k(t), u_k(t)), \quad x_k(0) = x_k^{ic} \qquad (1)$$

$$y_k(t) = H(x_k(t), u_k(t)) \qquad (2)$$

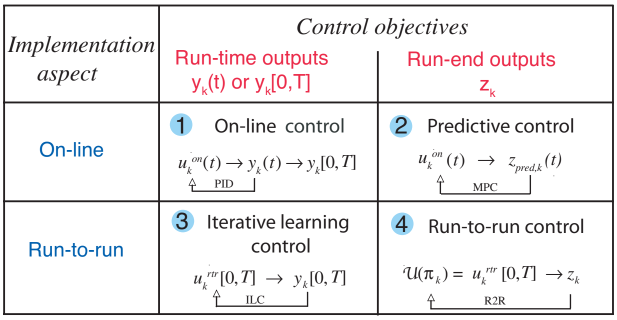
$$z_k = \mathrm{H}(x_k[0, T], u_k[0, T]) \qquad (3)$$

The dynamics over several runs stem from the possibility to update the initial conditions and the inputs on a run-to-run basis.

The system properties will be analyzed around selected reference trajectories, for which the accent $\bar{(\cdot)}$ will be used. For example, the reference state trajectories will be denoted by $\bar{x}[0, T]$, with $\bar{x}(t)$ being the corresponding state values at time t. Perturbations denoted by $(\cdot)$ will be considered, e.g. $\bar{x}[0, T]$ is a perturbation of $\bar{x}[0, T]$.

2.2 Control strategies

There are two types of control objectives (run-time outputs $y_k(t)$ or $y_k[0, T]$, and run-end outputs z_k), and also different ways of reaching them (on-line with $u_k^{on}(t)$ and run-to-run with $u_k^{rtr}[0, T]$). Each objective can be met either on-line or on a run-to-run basis, this choice being dependent on the type of measurements available. The control strategies are classified in Figure 1 and discussed next.



Implementation aspect	*Control objectives* Run-time outputs $y_k(t)$ or $y_k[0,T]$	 Run-end outputs z_k
On-line	1 On-line control $u_k^{on}(t) \rightarrow y_k(t) \rightarrow y_k[0,T]$ PID	2 Predictive control $u_k^{on}(t) \rightarrow z_{pred,k}(t)$ MPC
Run-to-run	3 Iterative learning control $u_k^{rtr}[0,T] \rightarrow y_k[0,T]$ ILC	4 Run-to-run control $\mathcal{U}(\pi_k) = u_k^{rtr}[0,T] \rightarrow z_k$ R2R

Fig. 1. Control strategies resulting from consideration of the control objectives (run-time or run-end outputs) and the implementation aspect (on-line or run-to-run).

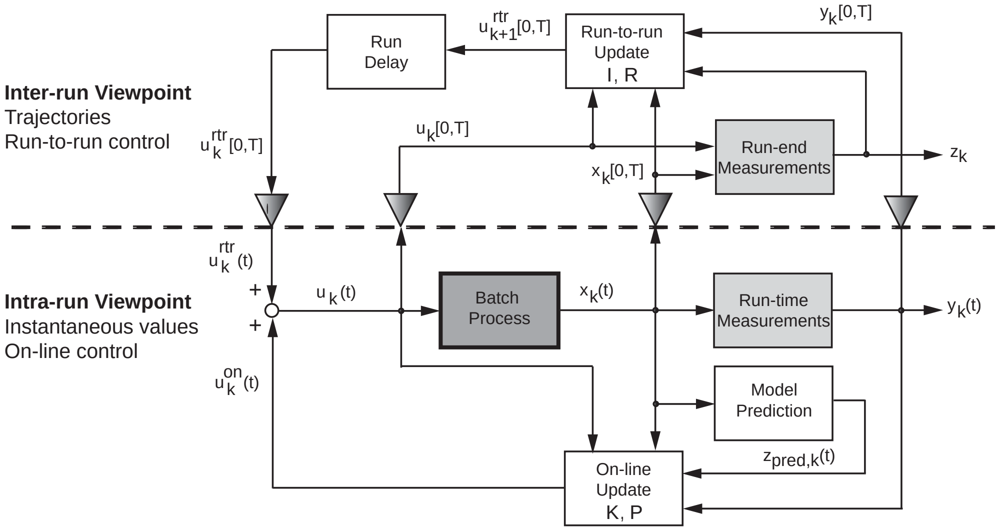


Fig. 2. Batch process with the inputs being updated both on-line (intra-run, use of the run-time measurements $y_k(t)$) and on a run-to-run basis (inter-run, use of the run-end measurements z_k). The symbol is used to indicate a change in viewing the time argument, e.g. from a trajectory to an instantaneous value when going downward and conversely when going upward.

- *On-line control of run-time outputs.* The approach is similar to that used in the traditional control literature. Control is typically done using PID techniques or more sophisticated alternatives whenever necessary. Formally, this controller can be written as

$$u_k^{on}(t) = \mathrm{K}(y_k(t), y_{sp}(t)) \qquad (4)$$

where K is the on-line controller for the run-time outputs $y_k(t)$, and $y_{sp}(t)$ the setpoint.

- *On-line control of run-end outputs.* It is necessary here to predict the run-end outputs based on measurement of the run-time outputs. Model predictive control (MPC) is well suited to that task (Nagy and Braatz 2003). The controller can be written as

$$u_k^{on}(t) = \mathrm{P}(z_{pred,k}(t), z_{sp}) \qquad (5)$$

where P is the on-line controller for the run-end outputs z_k, and $z_{pred,k}(t)$ the prediction of z_k available at time instant t.

- *Run-to-run control of run-time outputs.* In batch processing, key process characteristics such as process gain and time constants can vary considerably. Hence, the need to provide adaptation in a run-to-run manner to compensate the effect of these variations.

  The run-to-run part of the manipulated variable profiles can be generated using Iterative Learning Control (ILC) that exploits information from previous runs (Moore 1993). The controller has the structure

$$u_k^{rtr}[0,T] = \mathrm{I}(y_{k-1}[0,T], y_{sp}[0,T]) \qquad (6)$$

where I is the iterative learning controller for the run-time outputs $y_k[0,T]$. It processes the entire profile of the previous run to generate the entire manipulated profile for the current run.

- *Run-to-run control of run-end outputs.* The input profiles are parameterized using the input parameters π_k, $u_k^{rtr}[0,T] = \mathrm{U}(\pi_k)$. Control is then implemented using simple discrete integral control laws, that is $\pi_k = \pi_{k-1} + K(z_{sp} - z_{k-1})$ (Francois *et al.* 2005). Formally, the controller can be written as

$$u_k^{rtr}[0,T] = \mathrm{U}(\pi_k), \quad \pi_k = \mathrm{R}(z_{k-1}, z_{sp}) \qquad (7)$$

where R is the run-to-run controller for the run-end outputs z_k, and U the input parametrization.

Note that, except for predictive control that involves prediction, all the other control schemes use only measurements and thus do not necessitate a process model for implementation, i.e. a very nice feature for batch processes, where detailed accurate models are seldom available (Bonvin 1998).

By combining strategies for the various types of outputs, the control inputs can have contributions from both run-to-run and on-line updates:

$$u_k(t) = u_k^{rtr}(t) + u_k^{on}(t) \qquad (8)$$

The term $u_k^{rtr}(t)$ stems from the trajectories $u_k^{rtr}[0,T]$ and represent the 'feedforward' operating policies that are not altered within a run. However, $u_k^{rtr}[0,T]$ may change between runs (via run-to-run update), leading to inter-run dynamics. On the other hand, $u_k^{on}(t)$ represents the 'feedback' correction during the run (via on-line update). This combination of strategies is illustrated in Figure 2.

Applying only run-to-run control exhibits the limitations of being open-loop in run time, in particular for run-time disturbances. In general, a combination of these four strategies is used. However, in such a combined scheme, care should be taken that the on-line and run-to-run corrective actions do not oppose each other. Hence, the stability issue is critical.

In formulating the control strategy, controllability is important since it informs whether or not open-loop inputs exist that can provide the desired performance. Once a controller is designed, stability issues are of uppermost importance. Stabilization (and more appropriately finite-time stabilization), which is the issue of designing a controller that achieves stability and desired performance, will not be addressed in this paper.

3. INTRA-AND INTER-RUN STABILITY

Due to the presence of the two time scales t and k, both intra-run (in run time t) and inter-run (in run index k) stability need to be addressed.

3.1 Intra-run stability

Stability in run time t is important for repeatability and reproducibility reasons. The problem addressed therein is whether the trajectories of various runs with initial conditions sufficiently close will remain close during the rest of the run.

System (1) under on-line closed-loop operation using the feedback law (4) or (5) can be written as:

$$\dot{x}_k(t) = \tilde{F}(x_k(t), t), \qquad x_k(0) = x_k^{ic} \qquad (9)$$

The standard definition of Lyapunov stability is typically used around an equilibrium point (Vidyasagar 1978). To extend this definition to finite-time systems without an equilibrium point, it is first necessary to introduce the concept of a tube around the nominal trajectory in the $(n+1)$-dimensional space of states and time.

Definition 1. The trajectories $x_k[0,T]$ are defined to be inside the (a,b)-tube $\mathrm{B}_{a,b}$ around the reference trajectories $\bar{x}[0,T]$, i.e. $x_k[0,T]$ $\mathrm{B}_{a,b}$, if they satisfy $x_k(t) - \bar{x}(t) < ae^{bt}$, t $[0,T]$.

The tube consists of a ball of radius a in the n-dimensional state space at time $t = 0$, which shrinks or expands with time at a rate determined by b.

Definition 2. System (9) is locally intra-run β-**tube stable** around the trajectories $\bar{x}[0,T]$ if there exists a $\delta > 0$ such that, for all $x_k^{ic} = \bar{x}(0) +$ $\bar{x}(0)$ with $\bar{x}(0)$ $< \delta$, the state evolution $x_k[0,T]$ $\mathrm{B}_{\delta,\beta}$.

A diverging (converging) system has a positive (negative) value of β. Note that a system that initially diverges to eventually converge has a positive β. In addition to its sign, the value of β is quite useful since, with finite-time systems, the dividing line between stability and instability is not whether the trajectories converge or diverge, but by how much they come together or grow apart in the interval of interest. Hence, in the context of batch processes, stability is not a yes-no result, but rather a measure quantified by β.

Definition 3. System (9) is locally intra-run α-**terminal-time stable** around the trajectories $\bar{x}[0,T]$ if there exists a $\delta > 0$ such that, for all $x_k^{ic} = \bar{x}(0) +$ $\bar{x}(0)$ with $\bar{x}(0)$ $< \delta$, the terminal states statisfy $x_k(T) - \bar{x}(T)$ $< \alpha\delta$.

Terminal-time stability is the counterpart of asymptotic stability for finite-time systems. Again, stability is not simply determined by whether α is greater or less than 1, but instead it is quantified by the value of α.

It is possible to give results similar to the two theorems of Lyapunov (one based on linearization and the other on the existence of a non-increasing Lyapunov function) for tube stability.

Theorem 1. Let $\dot{x}_k(t) = A(t)$ $x_k(t)$ with the initial conditions $x_k(0) =$ $\bar{x}(0)$ be a bounded linearization of System (9) along $\bar{x}[0,T]$ for run k. Let $\sigma_{max}(t)$ be the maximum of the real parts of the eigenvalues of the time-dependent matrix $\frac{1}{t}\int_0^t A(\tau)d\tau$. Also, let $\bar{\sigma}_{max} = \max_t \sigma_{max}(t)$. Then, System (9) is tube stable around $\bar{x}[0,T]$ with $\beta = \bar{\sigma}_{max}$. Furthermore, the system is locally terminal-time stable around $\bar{x}[0,T]$ with $\alpha = e^{\sigma_{max}(T)T}$.

The proof of the theorem uses Bellman-Gronwall's Lemma (Vidyasagar 1978). Note that the eigenvalues of the integral of A are studied rather than the eigenvalues of themselves. In most optimally operated finite-time systems (e.g. using a finite-time linear quadratic regulator), though the eigenvalues of the integral are negative, some of the eigenvalues of might become positive toward the end of the run. This phenomenon caused by on-line control of z_k is referred to as the 'batch kick' in the optimization of batch processes. Intuitively, this means that little can go wrong toward the end since the 'time-to-go' is small.

Turning to the second Lyapunov method, the following result can be stated.

Theorem 2. Let $V(x,t) : \mathbb{R}^n \times \mathbb{R}_+ \quad \mathbb{R}$ be a continuously differentiable function such that $V(\bar{x},t) = 0$ and $V(x,t) > 0$ for all $x(t) = \bar{x}(t)$, t. If $\dot{V}(x,t) \quad \sigma(t)V(x,t)$ along the system trajectories for all $x(t) = \bar{x}(t) + \quad \bar{x}(t)$, t, $\bar{x}(t) < \delta$, then System (9) is tube stable with $\beta = \max_t \frac{1}{t}\int_0^t \sigma(\tau)\, d\tau$.

Note that the definition of stability presented by (Lohmiller and Slotine 1998) using contraction of deviations around pre-specified trajectories is a special case of Definition 2 above and requires contraction at every time instant, i.e. $\sigma(t) < 0$ for all t. This measure is clearly inadequate for batch systems that exhibit a batch kick. Information regarding the overall performance is better related to the integral of σ as given in Theorems 1 and 2 than to its instantaneous value.

3.2 Inter-run stability

The interest in studying stability in run index k arises from the necessity to guarantee convergence of run-to-run adaptation schemes. Here, the standard notion of stability applies as the independent variable k goes to infinity. The main conceptual difference with the stability of continuous processes is that 'equilibrium' refers to entire trajectories. Hence, the norms have to be defined in the space of functions $\mathbb{L}$ such as the integral squared error $\mathbb{L}_2$.

For studying stability with respect to run index k, System (1) is considered under closed-loop operation. At the k^{th} run, the trajectories of the $(k-1)^{st}$ run are known, which fixes $u_k^{rtr}[0,T]$ according to (6) or (7). These input profiles, along with the on-line feedback law (4) or (5), are applied to (1) to obtain $x_k(t)$ for all t and thus $x_k[0,T]$. All these operations can be represented formally as:

$$x_k[0,T] = \tilde{F}(x_{k-1}[0,T]),\quad x_0[0,T] = x_{init}[0,T] \quad (10)$$

where $x_{init}[0,T]$ are the initial state trajectories. Inter-run stability is considered around the equilibrium trajectory computed from (10), i.e. $\bar{x}[0,T] = \tilde{F}(\bar{x}[0,T])$.

Definition 4. System (10) is locally inter-run Lyapunov stable around the equilibrium trajectories $\bar{x}[0,T]$ if there exist $\delta > 0$ and $\epsilon > 0$ such that, for all $x_0[0,T] = \bar{x}[0,T] + \quad \bar{x}[0,T]$ with $\bar{x}[0,T] < \delta$, $x_k[0,T] - \bar{x}[0,T] < \epsilon$, k. If, in addition, $\lim_{k\to\infty} x_k[0,T] - \bar{x}[0,T] = 0$, then the system is locally inter-run asymptotically stable.

This stability definition is fairly standard but in a discrete setting. Thus, in principle, either one of the two Lyapunov methods (via linearization or Lyapunov function) can be used to analyze stability. However, the linearization method has problems since differentiation has to be performed in the space of functions. The Lyapunov-function method can be used once a norm is appropriately defined (Vidyasagar 1978).

Theorem 3. Let $V : \mathbb{L}^n \quad \mathbb{R}$ be a continuously differentiable functional such that $V(\bar{x}[0,T]) = 0$ and $V(x[0,T]) > 0$ for $x[0,T] = \bar{x}[0,T]$.

System (10) is locally inter-run Lyapunov stable if, for all $x_0[0,T] = \bar{x}[0,T] + \quad \bar{x}[0,T]$ with $\bar{x}[0,T] < \delta$, $V(x_{k+1}[0,T]) \quad V(x_k[0,T])$, k.

If, in addition, $\bar{x}[0,T]$ is the largest invariant set satisfying $V(x_{k+1}[0,T]) = V(x_k[0,T])$, then the system is locally inter-run asymptotically stable.

Again, the choice of a Lyapunov function is a major difficulty. The norm of the input error $u[0,T] - \bar{u}[0,T]_{\mathbb{L}_2}$ has served as a useful Lyapunov function in some of our studies, although the output error has been widely used in the literature.

4. CONTROLLABILITY OF RUN-TIME AND RUN-END OUTPUTS

One of the definitions of controllability for infinite-time dynamic systems requires that there exists an input vector $u[t_0,\tau]$ with which the equilibrium state can be reached from any arbitrary state $x(t_0)$ in the neighborhood of the equilibrium.

There are two difficulties with extending this definition to batch processes. Firstly, the controllability of finite-time systems needs to be defined around trajectories. Therein, the relevant question is whether or not some neighborhood of given trajectories can be reached. Clearly, not all state trajectories can be fixed independently because the state vector $x[0,T]$ contains a lot of redundant information. For example, since a position trajectory enforces the velocity, the trajectories of position and velocity cannot be chosen independently of each other[1]. Hence, only controllability in terms of *independent output* trajectories can be investigated (y-controllability).

Secondly, the above definition of controllability mentions the existence of a time τ, which however might be larger than the terminal time T. This aspect becomes important when considering the

[1] In contrast, when instantaneous values are considered, arbitrary position and velocity values can be specified.

controllability with respect to the run-end outputs (z-controllability).

Here, controllability addresses the problem of the existence of inputs that can implement the desired action and thus is independent of whether the correction is made on-line or on a run-to-run basis.

4.1 Controllability of run-time outputs

Let y_k^i, $i = \{1, \cdots, p\}$, be the i^{th} run-time output of System (1)-(2) and let its relative degree[2] be r^i, i.e. $\frac{\partial}{\partial u_k} \frac{d^j y_k^i}{dt^j} = 0, \;\; j < r^i$.

Definition 5. System (1)-(2) is locally y-**controllable** around the arbitrary trajectories $\bar{y}[0,T]$ if there exists a $\delta > 0$ such that, for all $\bar{y}[0,T] < \delta$, $\bar{y}^i[0,T] \;\; \mathbb{C}^{(r^i-1)}$ for $i = \{1, \cdots, p\}$, there exists $u_k[0,T]$ U that leads to $y_k[0,T] = \bar{y}[0,T] + \bar{y}[0,T]$.

Note that if the first $(r^i - 1)$ derivatives of $\bar{y}^i$ are discontinuous, Dirac impulses are required at the inputs to meet the outputs. Thus, the perturbations $\bar{y}^i$ that are considered cannot have discontinuities in their first (r^i-1) derivatives, i.e. $y^{i-} \;\; \mathbb{C}^{(r^i-1)}$, where $\mathbb{C}^r$ denotes the space of functions that have continuous derivatives up to order r.

Note also that the trajectories $\bar{y}[0,T]$ are assumed feasible, i.e. they respect the initial conditions and they can be implemented through $\bar{u}[0,T]$ (the condition under which $\bar{u}[0,T]$ exist for a given $\bar{y}[0,T]$ is not addressed here). The question asked in this definition regards only the neighboring trajectories. This is clearly a local inversion problem for which standard conditions for inverting a multi-input multi-output system can be used (Hirschorn 1979).

Theorem 4. Let u_k^j, $j = \{1, \cdots, m\}$, be the j^{th} input of System (1)-(2). Let the relative degrees r^i, $i = \{1, \cdots, p\}$, remain constant around $\bar{y}[0,T]$, and M (t) be defined as M $_{i,j}(t) = \frac{\partial}{\partial u_k^j} \frac{d^{r^i} y_k^i}{dt^{r^i}}$. If M (t) is of rank p, t, then System (1)-(2) is locally y-controllable around $\bar{y}[0,T]$.

4.2 Controllability of run-end outputs

A similar definition can be provided for system controllability in terms of reaching specified run-end outputs.

Definition 6. System (1,3) is locally z-**controllable**, from time t_0 on, around an arbitrary operating point $\bar{z}$ if there exists a $\delta > 0$ such that, for all $\bar{z} < \delta$, there exists $u_k[t_0,T]$ U that leads to $z_k = \bar{z} + \bar{z}$.

Here, the notion of controllability is linked to a given time t_0. The question asked is the following: Is it possible to change the outcome of the run if, at time instant t_0 in the run, one wishes so? To answer this question, consider the linearization of System (1,3) around a trajectory, resulting in the linear time-varying system (Friedland 1986):

$$\dot{x}_k = A(t) \;\; x_k + B(t) \;\; u_k, \quad x(t_0) = 0 \quad (11)$$

$$z_k = C(t) \;\; x_k \quad (12)$$

Theorem 5. Consider the output controllability Grammian G (t) for System (11)-(12):

$$P(\tau) = C(\tau) e^{\int_{t_0}^{\tau} A(\kappa)\, d\kappa} B(\tau)$$

$$\mathrm{G}(t_0) = \int_{t_0}^{T} P(\tau) P^T(\tau)\, d\tau \quad (13)$$

If G (t_0) is of rank q, then System (1,3) is locally z-controllable from time t_0 on.

For on-line control of run-end outputs, Theorem 5 can be used to indicate until what time t_0 in the batch the control of run-end outputs is feasible.

For run-to-run control of run-end outputs, it is important to study the case where the inputs are parameterized. Consider the parameterization $u_k[0,T] = \mathrm{U}(\pi_k)$, where $\pi_k \;\; \mathbb{R}^{n_\pi}$ are the input parameters. This way, the batch process can be seen as a static map between the input parameters π_k and the run-end outputs z_k. To assess controllability, the transfer matrix between π_k and z_k needs to be computed. The equivalent of Theorem 5 using input parametrization is given next.

Theorem 6. Consider the $q \times n_\pi$ transfer matrix between π and z calculated for System (11)-(12):

$$\mathrm{T}(t_0) = \int_{t_0}^{T} C(\tau) e^{\int_{t_0}^{\tau} A(\kappa)\, d\kappa} B(\tau) \frac{\partial \mathrm{U}}{\partial \pi}\, d\tau \quad (14)$$

If T (t_0) is of rank q, then System (1,3) with the parametrization $u_k[0,T] = \mathrm{U}(\pi_k)$ is locally z-controllable from time t_0 on.

Note that run-to-run control requires only the evaluation of the matrix T (0). The rank condition (or invertibility) of G or T follows from the

[2] The relative degree of an output is the minimal degree of its time derivative for which at least one input appears.

fact that the inputs that can create the necessary change in the run-end outputs are obtained by inversion. However, note that as t_0 approaches T, the Grammian approaches singularity, with G$(T) = 0$. Similarly, if a piecewise parametrization is used, after a certain time, some of the parameters will have no influence on the outputs, thus making a few columns zero. As t_0 proceeds toward T, more and more columns will become zero. Hence, as $t \quad T$, inverting G or T requires larger and larger inputs for control. Also rank deficiency may occur, and the system may lose controllability.

5. ILLUSTRATIVE EXAMPLE

Consider the scale-up, from the laboratory to production, of a semi-batch reactor in which several reactions take place. The desired and main side reactions are

$$A + B \qquad C, \qquad 2B \qquad D$$

with C the desired product and D an undesired side product. The reactions are fairly exothermic and the reactor is equipped with a jacket for heat removal. The control objective is twofold: (i) Operate isothermally at 50°C by manipulating the jacket temperature, and (ii) match the final concentrations that have been obtained in the laboratory, $c_B(T) = c_{B,max}$ and $c_D(T) = c_{D,max}$, by manipulating the feed rate of reactant B.

The control structure used is illustrated in Figure 3. It implements on-line feedback temperature control. In addition, the feedforward profile for the jacket temperature $T_j^{ff}[0,T]$ is adjusted on a run-to-run basis by means of ILC. In this case, M $= \frac{d\dot{T}_r}{dT_j}$ is a constant non-zero scalar irrespective of the trajectory chosen (hence, satisfies y-controllability - Theorem 4). The controller reads

$$T_{j,k}(t) = T_{j,k}^{ff}(t) + K_R e_k(t) + \frac{K_R}{\tau_I}\int\limits_0^t e_k(\tau)d\tau,$$

$$T_{j,k+1}^{ff}[0, T - \quad] = T_{j,k}^{ff}[\quad, T] + K_{ILC}\, e_k[\quad, T],$$

with $e_k(t) = T_{r,ref}(t) - T_{r,k}(t)$, K_R the proportional gain and τ_I the integral time constant of the PI master controller. It can be easily verified that the system is tube stable with a negative β. K_{ILC} is the gain of the ILC controller and $\quad 0$ the value of the input shift. The second equation allows adapting the feedforward term for the jacket temperature setpoint on a run-to-run basis based on ILC with input shift. In Theorem 3, the integral squared output error $\int_0^T e_k^2(\tau)\, d\tau$ is used as the Lyapunov function in run index k. The value of the input shift is tuned for convergence (Welz *et al.* 2004). Due to the presence of the shift, the error does not converge asymptotically to zero.

In addition, the feed rate profile $u[0,T]$ is parameterized using the two feed-rate levels u_1 and u_2, each valid over half the batch time. The final concentrations $c_B(T)$ and $c_D(T)$ are met, on a run-to-run basis, by adjusting the two parameters $\pi = \{u_1, u_2\}$. The transfer matrix T is evaluated around the current operating point using (14), with $\frac{\partial \mathcal{U}}{\partial \pi} = [1\ 0]^T$ during the first half of the batch and $\frac{\partial \mathcal{U}}{\partial \pi} = [0\ 1]^T$ in the second half. With the matrix T being full rank (satisfies z-controllability - Theorem 6), the discrete integral control law reads

$$\pi_{k+1} = \pi_k + \mathrm{T}^+ K_{R2R}\,[z_{ref} - z_k], \qquad (15)$$

where T$^+$ is the pseudo-inverse of T, and K_{R2R} the gain of the run-to-run controller. The run-to-run convergence of this scheme can be shown using Theorem 3 with the squared input error $\pi - \pi^{*}\ ^2$ as the Lyapunov function in run index k (Francois *et al.* 2005).

The evolution of the manipulated and controlled variables are illustrated in Figures 4.

6. CONCLUSIONS

The control of batch processes is characterized by run-time and run-end objectives on the one hand, and by actions that can be implemented on-line and on a run-to-run basis on the other. It has been shown that the concepts of stability and controllability, which are well understood for infinite-time systems operating around an equilibrium point, are not directly applicable to finite-time batch processes.

With regard to stability, the concept of tube stability, by which the state trajectories remain within a given tube, has been introduced. The special case of terminal-time stability has also been discussed. Two theorems that help evaluate tube stability have been proposed.

As for controllability with respect to specified trajectories, it was observed that the entire state space cannot be studied due to the fact that there is considerable redundancy in the state trajectories. Hence, only controllability with respect to two types of outputs have been addressed. Controllability was studied from the point-of-view of inversion, and results were adapted from the existing literature.

REFERENCES

Abel, O., A. Helbig, W. Marquardt, H. Zwick and T. Daszkowski (2000). Productivity optimiza-

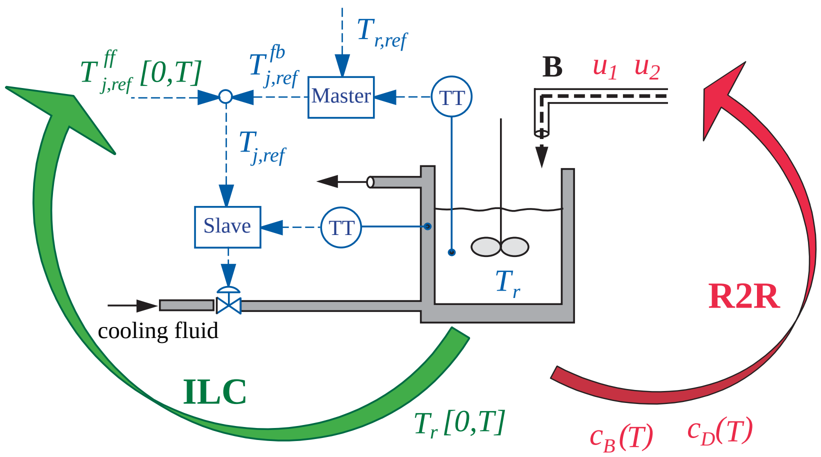


Fig. 3. On-line and run-to-run strategies to control the reactor temperature and the final concentrations.

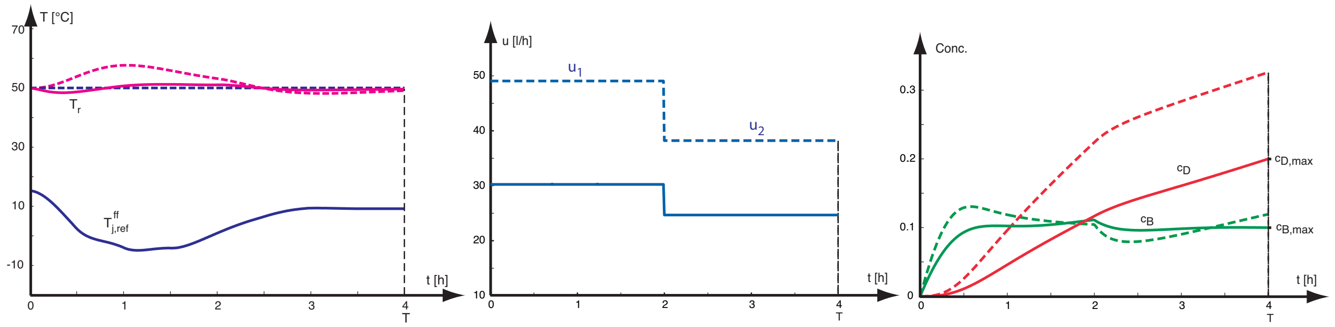


Fig. 4. Evolution of the reactor and jacket temperatures (left), of the feed rate (middle) and of the concentrations c_B and c_D (right), initially (dotted lines) and after 3 iterations (solid lines).

tion of an industrial semi-batch polymerization reactor under safety constraints. *J. Process Contr.* **10**(4), 351–362.

Bonvin, D. (1998). Optimal operation of batch reactors - A personal view. *J. Process Contr.* **8**(5–6), 355–368.

Campbell, W.J., S.K. Firth, A.J. Toprac and T.F. Edgar (2002). A comparison of run-to-run control algorithms. In: *American Control Conference*. Anchorage, Alaska. pp. 2150–2155.

Chin, I.S., S.J. Qin, K.S. Lee and M. Cho (2004). A two-stage iterative learning control technique combined with real-time feedback for independent disturbance rejection. *Automatica* **40**(11), 1913–1922.

Flores-Cerrillo, J. and J.F. MacGregor (2003). Within-batch and batch-to-batch inferential-adaptive control of semibatch reactors: A partial least squares approach. *Ind. Eng. Chem. Res.* **42**, 3334–3335.

Francois, G., B. Srinivasan and D. Bonvin (2005). Use of measurements for enforcing the necessary conditions of optimality in the presence of constraints and uncertainty. *J. Process Contr.* **15**(6), 701–712.

Friedland, B. (1986). *Control System Design – An Introduction to State-Space Methods*. McGraw-Hill, New York.

Hirschorn, R.M. (1979). Invertibility of multivariable nonlinear control systems. *IEEE Trans. automat. Contr.* **24**, 855–865.

Lohmiller, W. and J.J.E. Slotine (1998). On contraction analysis for nonlinear systems. *Automatica* **34**(6), 683–696.

Moore, K.L. (1993). *Iterative Learning Control for Deterministic Systems*. Springer-Verlag, Advances in Industrial Control, London.

Nagy, Z.K. and R.D. Braatz (2003). Robust nonlinear model predictive control of batch processes. *AIChE Journal* **49**(7), 1776–1786.

Srinivasan, B., D. Bonvin, E. Visser and S. Palanki (2003). Dynamic optimization of batch processes: II. Role of measurements in handling uncertainty. *Comp. Chem. Eng.* **44**, 27–44.

Vidyasagar, M. (1978). *Nonlinear Systems Analysis*. Prentice-Hall, Englewood Cliffs.

Welz, C., B. Srinivasan and D. Bonvin (2004). Iterative learning control with input shift. In: *IFAC Symp. DYCOPS-7*. Boston, MA. pp. 187–192.