CONTROL OF INTERMITTENT PROCESSES

Madhukar Pandit and Heiko Hengen
Control Systems and Signal Theory Group, University of Kaiserslautern, Germany

Keywords: Repetitive Control, Iterative Learning Control, Discrete time Models, Cyclic Processes

Contents

Summary

Many processes in industry belong to the class of intermittent processes – the iterative learning control (ILC) and repetitive control (RC) schemes are suited to a special subclass of these processes, the cyclic processes. Whereas normal controllers do not benefit from the repetition of the trajectories, the two learning control schemes improve the system output from cycle to cycle. ILC and RC can be employed in practically every cyclic process. This chapter gives a brief overview of the different techniques and control strategies, the required system models, the fields of application and the use in real industrial setups.

1. Introduction

The control of intermittent processes is a task often encountered in a variety of situations. Every real life process being of finite duration, any control task which has anything to do with practical implementation indeed involves the control of an intermittent process. We here, however, are concerned with a special class of intermittent process, viz. with intermittent processes in which one or more process variables occur cyclically. We subsume such processes under the heading cyclic processes. Of interest are the variables of the process over a sequence of individual cycles. Cyclic processes often met are characterized by the following properties:

a. The reference inputs to the process are identical for the individual cycles.
b. The process starts every cycle with the same initial conditions.
c. The interval between any two consecutive cycles need not necessarily be the same.

   Examples of processes with these properties are

   - robot arm movements in a manufacturing plant
   - the transition of a process to its operating point whereby various variables of the process are required to follow prescribed runs
   - discontinuous extrusion of metals and plastics
   - batch processes in bio technological reactors.

   Processes with the above properties can be very effectively dealt with employing iterative learning control (ILC) which is the main theme of this chapter.

   A second class of cyclic systems is of cyclic systems in which

d. the process starts a cycle with initial conditions which are the end conditions of the previous cycle. Control methods employed for these systems are termed repetitive control.

   If furthermore

e. the interval between two consecutive cycles is a constant, one has periodic processes. An example of a periodic process is the motion of a mirror in an infra-red scanner camera to be described later in this chapter.

The chapter is organized as follows. In Section 2 we give the definitions and physical and mathematical models concerning cyclic processes and their controls. Section 3 gives control schemes which can be employed for cyclic processes and their relative merits. It becomes clear that ILC is the method of choice for all processes with a nonzero inter-cycle time. Repetitive control is more suited to the process classes d) and e). Section 4 gives a scheme for designing ILC for real world applications. Robustness issues are discussed in Section 5. In Section 6 case studies of the application of ILC for

a) the design of temperature control in extruders and

b) the control of the transient phase of a multivariable process control and
c) repetitive control of a scanner mirror of an infra-red camera

are given. Furthermore, results of experiments conducted with industrial systems employing ILC are cited. These show the power of the ILC and RC control methods.

Many industrial processes have a cyclic character: they are repeated in more or less the same manner. Iterative learning and repetitive control systems take advantage of this kind of behavior: every process cycle i.e. the process input and output signals over a finite horizon of time can be considered as an observation of the process behavior.

The exploitation of this surplus of information (compared to a standard feedback control system) yields a higher system performance from process cycle to process cycle.

2. Definitions, Physical and Mathematical Models

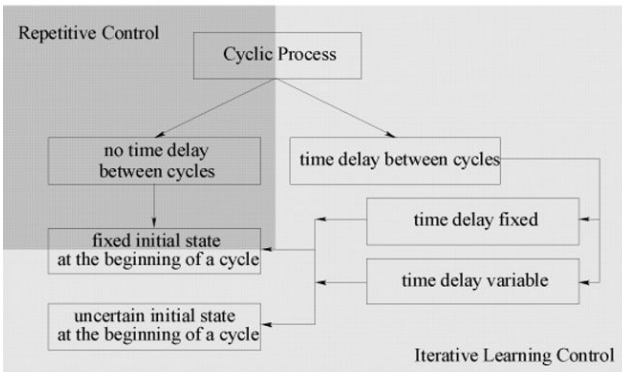

Figure 1: Classes of cyclic processes

2.1. Classes of Cyclic Processes

In order to elucidate the learning schemes of iterative learning and repetitive control schemes, one has to understand the nature of a "cyclic" process. We use definition 2.1 for a cyclic process:

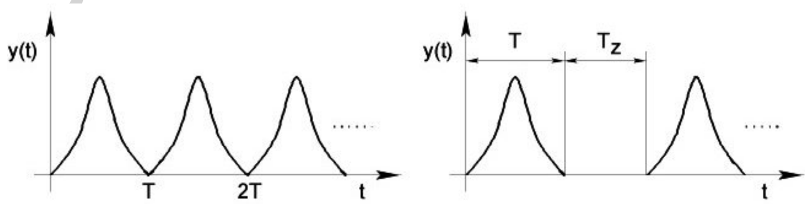

Figure 2: Left: cyclic process with no time delay between the cycles and without variations in the initial state at the beginning of a cycle, right: cyclic process with time delay between the cycles and without variations in the initial state at the beginning of a cycle

Definition 2.1 (Cyclic process): A process (or dynamical system) is called cyclic if one of the process variables (e.g. the reference input $\mathbf{y}_d(t)$) of duration T_k is repeated similarly over the cycles:

$$\mathbf{y}_d(t) = \sum_{k=0}^{\infty} \mathbf{y}_{d_0}(t - \tau_k) \tag{1}$$

$$\mathbf{y}_d(t) = \begin{cases} \mathbf{y}_d(t) & \text{for } t \in [0, T_k] \\ \overline{\mathbf{y}}_d(t) & \text{otherwise} \end{cases}$$

(with $\mathbf{y}_d(t)$ the run of the desired value during the cycle and $\overline{\mathbf{y}}_d(t)$ the desired value between the cycles

-in many cases $\mathbf{0}$ or unspecified). In such a case that output $\mathbf{y}(t)$ has similar runs over the individual cycles. Then we have:

$$\mathbf{y}(t) = \sum_{k=0}^{\infty} \mathbf{y}_k(t - \tau_k) \text{ with}$$

$$\mathbf{y}_k(t) = \begin{cases} \mathbf{y}(t) & \text{for } t \in [0, T_k] \\ \overline{\mathbf{y}}(t) & \text{for } t \in]T_k, \tau_{k+1} - \tau_k[\quad \mathbf{y}_k, \overline{\mathbf{y}}_k(t) \in \mathbb{R}^q, t, \tau_k, T_k \in \mathbb{R}, k \in \mathbb{N}_0. \end{cases} \tag{2}$$

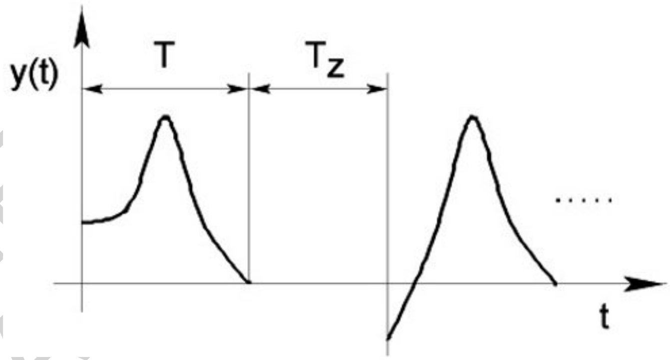

Figure 3: Cyclic process with time delay between the cycles and variations in the initial state

Furthermore

$$\tau_{k+1} \geq \tau_k + T_k. \tag{3}$$

The time T_k is called duration of the k -the cycle.

Ideally, $\mathbf{y}(t)$ would converge fast to $\mathbf{y}_d(t)$ for some k. Figure 2 and 3 give an overview of the different types of cyclic processes.

2.2. System Models

For the study of ILC systems and especially for the design of a proper ILC controller (i.e. operator),a system model is of advantage. In fact, the more information about the system one can provide, the better will be the results of employing the ILC-scheme.

The most general form of a system model is given by Def. 2.2. Repetitive and iterative learning control systems are necessarily sampled data systems, as the samples of the in/- and output-trajectories have to be stored. Hence modeling the systems in the sampled data domain is the appropriate approach.

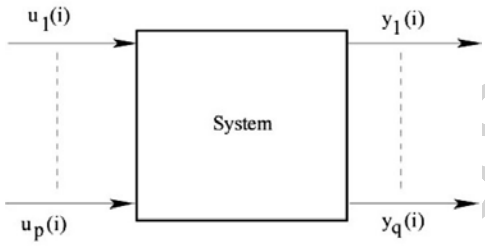

Figure 4: System with multiple in-and outputs

Definition 2.2 (Dynamical Sampled Data System): The response $\mathbf{y}(i)$ of a dynamical system to the input $\mathbf{u}(i)$ can be described using the following operator equation:

$$\mathbf{y}(i) = \mathcal{S}\{\mathbf{u}(i)\} \tag{4}$$

with

$$\mathbf{y}(i) = \begin{bmatrix} y_1(i) \\ \vdots \\ y_q(i) \end{bmatrix}, \ \mathbf{y}(i) \in \mathbb{R}^q, \quad \mathbf{u}(i) = \begin{bmatrix} u_1(i) \\ \vdots \\ u_p(i) \end{bmatrix}, \ \mathbf{u}(i) \in \mathbb{R}^p \tag{5}$$

the output vector $\mathbf{y} \in \mathbb{R}^q$ and $\mathbf{u} \in \mathbb{R}^p$ the input vector. The system operator $\mathcal{S}\{\cdot\}$ can be linear, nonlinear, or time-variant.

The system operator given in Def. 2.3. can be generally expressed in terms of the state equations:

$$\mathbf{x}(i+1) = \mathbf{f}(\mathbf{x}(i), \mathbf{u}(i), i) \tag{6}$$

$$\mathbf{y}(i) = \mathbf{g}(\mathbf{x}(i), \mathbf{u}(i), i), \quad \mathbf{y} \in \mathbb{R}^q, \mathbf{u} \in \mathbb{R}^p, \mathbf{x} \in \mathbb{R}^M . \tag{7}$$

Nonlinear state equations which are obtained by considering the physical laws governing the process are well suited for system simulation. For most applications however, a linear approximate description of the system is sufficient and more convenient.

2.2.1. Transfer Function Models

Most control engineers are used to employing transfer function models for controller design. In the continuous time case, the differential equation of a linear technical SISO system is given by

$$y(t)+a_1\frac{dy(t)}{dt}+\ldots a_m\frac{d^m y(t)}{dt^m}=b_0u(t)+b_1\frac{du(t)}{dt}+\ldots+b_m\frac{d^m u(t)}{dt^m}. \qquad (8)$$

The transfer function is obtained by applying the $\mathcal{L}$-Transform:

$$G(s)=\frac{b_0+b_1s+\ldots+b_m s^m}{1+a_1s+\ldots+a_m s^m}. \qquad (9)$$

Transfer functions in the *s*-domain are well known and widely used models. They will de used for the introduction of the repetitive and iterative learning control methodologies (see Section 3). For the sampled data case, a transfer function model (in the *z*-domain) represents the $\mathcal{Z}$-transform of the following difference equation:

$$\begin{aligned} y(i)&+a_1y(i-1)+\ldots a_m y(i-m)\\ &= b_0u(i)+b_1u(i-1)+\ldots+b_m u(i-m)\end{aligned}. \qquad (10)$$

The transfer function is then given by:

$$G(z^{-1})=\frac{b_0+b_1z^{-1}+\ldots+b_m z^{-m}}{1+a_1z^{-1}+\ldots+a_m z^{-m}} \qquad (11)$$

2.2.2. Finite Horizon Operator Models

For dealing with iterative learning control systems, a finite horizon operator model is well suited. This is because the control operator can be directly calculated by solving a least squares problem.

Then the input relations are depicted as follows: the trajectories are considered over an interval $[0,(N-1)\cdot T_s]$,with T_s the sampling time. The sampled data of the system input-, output- and desired trajectory is expressed in vectorial form:

$$\mathbf{y}_d = \begin{bmatrix} y_d(0) \\ y_d(1) \\ y_d(2) \\ \vdots \\ y_d(N-1) \end{bmatrix}, \ \mathbf{y}_k = \begin{bmatrix} y_k(0) \\ y_k(1) \\ y_k(2) \\ \vdots \\ y_k(N-1) \end{bmatrix}, \ \mathbf{u}_k = \begin{bmatrix} u_k(0) \\ u_k(1) \\ u_k(2) \\ \vdots \\ u_k(N-1) \end{bmatrix}. \qquad (12)$$

The elements of the vectors $\mathbf{u}_k, \mathbf{y}_k$ and $\mathbf{y}_d$ are the sampled values of the time functions $u_k(t), y_k(t)$ and $y_d(t)$ (with k the cycle index). Assuming a cycle duration $T_k = N \cdot T_s$ with N samples and the sample time T_s, one obtains vectors of the dimension $\mathbb{R}^{N\times 1}$. In the case of a multiple input multiple output (MIMO) system, one obtains block vectors $\mathbf{u}_k, \mathbf{y}_k$ and $\mathbf{y}_d$. Now the relation between input and output is given by

$$\mathbf{y} = \mathbf{G}\mathbf{u} + \mathbf{f}\,. \qquad (13)$$

To be precise, we have the following:

Definition 2.3 (System model for ILC design): A finite horizon operator model for ILC design describes the system output composed of the forced response dependent on the system input and the free response which is only dependent on the initial state $\mathbf{y} = \mathcal{S}_u\{\mathbf{u}, \mathbf{y}\} + \mathbf{f}$. $\mathcal{S}_u$ is the system operator describing the forced dynamics, $\mathbf{f}$ depicts the free response. In the case of a linear system model one obtains $\mathbf{y} = \mathbf{G}\mathbf{u} + \mathbf{f}$.

The model $\mathbf{G}\mathbf{u} + \mathbf{f}$ (impulse response model), can be derived in a straightforward manner by solving the state equations iteratively. For a linear state space model (with M the number of states $\mathbf{x} \in \mathbb{R}^{M\times 1}, \mathbf{A} \in \mathbb{R}^{M\times M}$, q the number of system outputs $\mathbf{C} \in \mathbb{R}^{q\times M}, \mathbf{D} \in \mathbb{R}^{q\times p}$, p the number of system inputs $\mathbf{B} \in \mathbb{R}^{M\times p}$), given by the following state equations,

$$\begin{aligned} \mathbf{x}(i+1) &= \mathbf{A}\mathbf{x}(i) + \mathbf{B}\mathbf{u}(i) \\ \mathbf{y}(i) &= \mathbf{C}\mathbf{x}(i) + \mathbf{D}\mathbf{u}(i)\,, \end{aligned} \qquad (14)$$

the expansion of the state equations w.r.t. the system output $\mathbf{y}$ over a horizon of N samples yields:

$$\mathbf{y} = \mathbf{G}_0\mathbf{x}(0) + \mathbf{G}\mathbf{u} \qquad (15)$$

$$\mathbf{G}_0 = \begin{bmatrix} \mathbf{C} \\ \mathbf{CA} \\ \mathbf{CA}^2 \\ \vdots \\ \mathbf{CA}^{N-1} \end{bmatrix}, \quad \mathbf{G} = \begin{bmatrix} \mathbf{D} & 0 & \dots & 0 & 0 \\ \mathbf{CB} & \mathbf{D} & \dots & \dots & 0 \\ \mathbf{CAB} & \mathbf{CB} & \mathbf{D} & \ddots & 0 \\ \vdots & \ddots & \ddots & \ddots & 0 \\ \mathbf{CA}^{N-2}\mathbf{B} & \dots & \dots & \mathbf{CB} & \mathbf{D} \end{bmatrix} \tag{16}$$

The matrix $\mathbf{G}$ is the impulse response matrix of the system. The free response can be calculated using $\mathbf{f} = \mathbf{G}_0\mathbf{x}(0)$

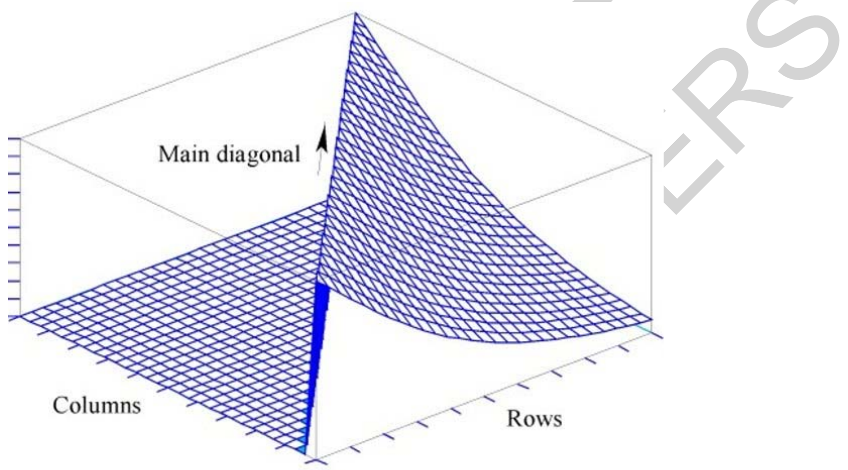

Figure 5: The impulse response matrix $\mathbf{G}$ as a 3-D plot

-
-
-

Bibliography

Chew, K. K and M. Tomizuka. (1990). Digital Control of Repetitive Errors in Disk Drive Systems.IEEE Control Systems Magazine, Vol.10, pp.16-20.

Dick De Roover and Okko H. Bosgra. (2000). Synthesis of robust multivariable iterative learning controllers with application to a wafer stage motion system. International Journal of Control, Vol.73, No. 10, pp.968-979. [Robustness aspects, synthesis of robust ILC]

Heiko Hengen, Stefan Hillenbrand and Madhukar Pandit. (2000). Algorithms for Iterative Learning Control of Nonlinear Plants using Time variant System Descriptions. Proceedings of the 2000 IEEE

CCA/CACSD Joint Conference, CCA 00-1059, TM4-5. IEEE. [Application of time-variant system models for the design of ILC]

Lee, H.- S. and Bien, Z. (1996). Study on robustness of iterative learning control with non-zero initial error. International Journal of Control, Vol. 64, No. 3, pp345-359.

Masayoshi Tomizuka, Tsu Chin Tsao and Kok Kia Chew. (1989). Analysis and synthesis of discrete--time repetitive controllers, Journal of Dynamic Systems, Measurement, and Control, Vol.111, pp.353-358.

N. Amann, H. Owens and E. Rogers. (1996). Iterative learning control for discrete-time systems with exponential rate of convergence. IEE Proceedings-Control Theory and Applications, Exeter, UK, Vol.143, No.2, pp. 217-224.

S. Arimoto and S. Kawamura and F. Miyazaki. (1984). Bettering operation of robots by learning. Journal of Robotic Systems, Vol. 1, No.2, pp123-140. [One of the basic papers]

S. Arimoto, S. Kawamura and F. Miyazaki. (1984). Bettering Operation of Dynamic Systems By Learning: A New Control Theory for Servomechanism or Mechatronic Systems, Proceedings of the 23rd IEEE Conference on Decision and Control, Las Vegas, Nevada, pp.1064-1069.

S. Arimoto, S. Kawamura, F. Miyazaki and S. Tamaki. (1985). Learning Control Theory for Dynamical Systems. Proceedings of the 24th IEEE Conference on Decision and Control, Ft. Lauderdale, Florida, pp. 1375-1380.

S. Arimoto, T. Naniwa and H. Suzuki. (). Robustness of P-Type learning control with a forgetting factor for robotic motions. Proceedings of the 29th IEEE Conference on Decision and Control, No.29, IEEE, pp. 2640-2645. [Robustness analysis of ILC]

S. Hillenbrand and M. Pandit. (1999). A discrete time iterative learning control law with exponential rate of convergence, Proceedings of the IEEE Conference on Decision and Control, No.38.[Selection of a suitable sampling rate as a means of solving ill conditioned ILC problems]

S.~S. Saab. (1995). A discrete-time learning control algorithm for a class of linear time--invariant systems. IEEE Transactions of Automatic Control, Vol.40, No.6, pp1138-1141.

T. Ishihara and K. Abe and T. Takeda. (1992). A discrete-time design of robust iterative learning controllers. IEEE Transactions on Systems, Man and Cybernetics, Vol.22, No.1, pp. 74-84. [Least squares design of discrete time ILC]

T. Sugie and T. Ono (1987). An iterative learning control law for dynamical systems. Proceedings of the 10th Triennal World Congress. München, pp339-344.

T.-J. Jang and H.-S. Ahn and C.-H. Choi. (1994). Iterative learning control for discrete--time nonlinear systems. International Journal of Systems Science. Vol. 25, No.7, pp.1179-1189.

Zeungnam Bien and Jian--Xin Xu (eds.). (1998). Iterative Learning Control - Analysis, Design, Integration, and Applications , Boston, Dordrecht, London: Kluwer Academic Publishers

Biographical Sketches

Madhukar Pandit, born on 14/09/1938 in Bangalore/India, studied electrical engineering in Bangalore and Karlsruhe. He obtained the Dr.-Ing. degree in Control Systems in the Technische Hochschule Karlsruhe and the Habilitation in the Kaiserslautern University. He worked at the National Aeronautical Laboratory, Bangalore, Brown Boveri and Cie in Mannheim. Since 1978, he has been professor of Control Systems and Signal Theory at the Kaiserslautern University. His group is active mainly in the areas of process control and image processing applied to medical imaging and quality control.

Heiko Hengen, born on 06/09/1975 in Karlsruhe/Germany, studied electrical engineering in Kaiserslautern. He obtained the Dipl.-Ing. degree in 1999 and joined the Control Systems and Signal Theory group. His focus of research has been iterative learning control of real world systems, especially the system- and signal oriented design of iterative learning controllers; in 2002, he obtained the Dr.-Ing. degree and is currently involved in research projects in the field of medical image processing.