Available online at www.sciencedirect.com

ScienceDirect

Control Engineering Practice 16 (2008) 623–631

CONTROL ENGINEERING PRACTICE

www.elsevier.com/locate/conengprac

Design of a dynamic threshold generator for λ-tuned control loops

Magnus Berndtsson, Andreas Johansson*

Control Engineering Group, Luleå University of Technology, SE-97187 Luleå, Sweden

Received 2 September 2006; accepted 26 July 2007
Available online 20 September 2007

Abstract

A dynamic threshold generator is employed for detecting faults in λ-tuned control loops. To this end, an optimization algorithm for dynamic threshold generators is proposed. The *a priori* information from λ-tuning is used in designing a state estimator with integral action. A dynamic threshold generator for the residual of this state estimator is derived and the optimization algorithm is applied. Simulations with measurement data from an experimental water tank setup show that the method is capable of detecting a small fault without generating false alarms.

Keywords: PI controllers; Fault detection; Optimization; Uncertain linear systems

1. Introduction

Poorly operating control loops is a problem in many industries worldwide. Performance monitoring has therefore been an active area of interest for several decades, see Jelali (2006) and references therein. Problems with poorly operating control loops are often manifested as oscillations. The reason for the oscillations may be bad controller tuning (Hägglund, 1995) but could also be physical problems e.g. sticking valves. There are several commonly used methods for detecting this kind of oscillations, see Thornhill and Horch (2007). To be able to distinguish the difference between good and bad performance of a control loop, it is necessary to know what kind of behavior characterizes the good performance. This kind of *a priori* information is time consuming and thus expensive to obtain. A way to deal with this problem is to use already known information, e.g. from the controller tuning, to develop a controller supervision algorithm. The λ-method is a well-known tuning algorithm for PI controllers (Sell, 1995) that has been developed into a standard in the Swedish pulp and paper industry. The method is based on the internal model control (IMC) principle and holds for first order systems with time delay. It is easy to apply and is intended to be used by ordinary control-room operators. The method has become very popular in the pulp and paper industry because of its simplicity and is now spreading to other industries.

An example of a monitoring tool that uses the information obtained from λ-tuning is presented in Ingimundarson and Hägglund (2005). Ingimundarson has developed a method based on the Harris index, which compares the controller behavior to the minimum variance controller (Harris, 1989), but takes advantage of the known properties of the process. The minimum variance controller is in some sense an optimal controller and in a plant with hundreds, maybe even thousands of control loops, optimal performance may not be a relevant comparison. To make every controller behave close to the optimum will be a time consuming task for the staff of the plant.

The idea behind this work is to use the known properties of the process to create a model-based fault detection algorithm. Unlike the Harris index the goal is not to compare the existing controller to an optimal one but to detect changes in the process parameters (from the λ-tuning). Faults that are relevant to detect are, for example, faulty sensors, stiction or leakage. All of those faults will appear as changes in the process properties. Faulty sensors may appear as measurement bias or noise or changed process gain. Stiction will introduce nonlinearities

*Corresponding author. Tel.: +46 920 492334.
E-mail address: andreas.johansson@ltu.se (A. Johansson).

0967-0661/$ - see front matter
doi:10.1016/j.conengprac.2007.07.003

which will make the process differ from the identified one. This way many faults can be considered as changes in the process parameters.

Model-based fault detection (Chiang, Russell, & Braatz, 2001) is based on using the redundancy in the information obtained from the measurements in combination with a process model. If the measured output does not match the expected output produced by a process model, then the presence of a fault can be deduced. Provided an analytical process model, a fault detection algorithm essentially consists of two steps, the residual generation and residual evaluation. The purpose of the residual generation is to generate a signal which is nonzero when there is a fault and zero otherwise. However, due to disturbances and uncertainties the residual generally is nonzero even if no fault is present. This fact necessitates the second step of the fault detection algorithm, the residual evaluation, which consists of comparing some function of the residual to a threshold and raising an alarm if the former exceeds the latter. If the model describes the process perfectly then the residual will only be affected by the nonmeasured disturbances, otherwise the residual will also depend on known input signals. A way to deal with the problem of inaccurate process parameters is to assume a process model with time-varying parameter uncertainties and to use these uncertainties when calculating the threshold as in e.g. Johansson, Bask, and Norlander (2006) (Section 2) and Ding, Frank, and Ding (2002). The prime drawback of such an approach is that bounds for the uncertainties must be known. In this paper, the method suggested in Bask and Johansson (2005) and Bask (2005) to use measurement data for finding values for these parameters is generalized (Section 3) and applied to the problem of monitoring λ-tuned control loops (Section 4). The resulting method is tested with measurement data from an experimental scale water tank system (Section 5).

1.1. λ-Tuning

λ-Tuning is a method of tuning of PI controllers based on the IMC principle. It is a guideline that holds for processes with a time delay and a strong time constant.

$$G(s) = \frac{k}{1+Ts} e^{-\tau s}. \tag{1}$$

The tuning consists of an identification part where the parameters of system (1) are identified using an open loop step response according to Fig. 1. The process gain is $k = \Delta PV/\Delta OUT$ while T is the process time constant, calculated as the time it takes for the process to reach 63% of ΔPV. τ is the time delay of the process, i.e. the time it takes from the step input until the output responds.

The controller is a PI:

$$C_\lambda(s) = k_c\left(1 + \frac{1}{T_i s}\right). \tag{2}$$

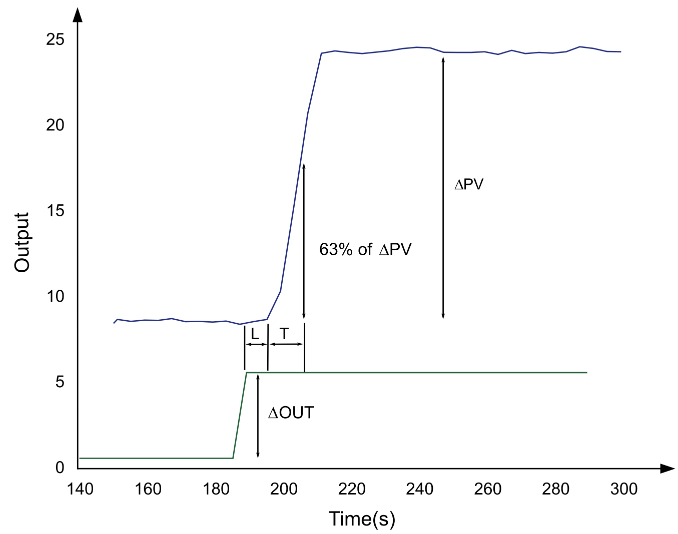

Fig. 1. A step response for λ-tuning.

The controller parameters should be chosen as

$$T_i = T,$$
$$k_c = \frac{T_i}{k(\lambda+\tau)}.$$

The parameter λ is closely related to the closed loop time constant, and is to be chosen in the interval $T \leqslant \lambda \leqslant 3T$. If $\lambda = T$ the tuning is considered aggressive and $\lambda = 3T$ is considered robust.

1.2. Notation and preliminaries

In the following, $|\cdot|$ denotes element-wise absolute value when applied to a matrix. Inequalities between matrices is also to be interpreted element-wise. The notation I_n represents the $n \times n$ identity matrix while 1_n is a column vector of ones of dimension n and $0_{m\times n}$ represents a matrix of zeros with the dimension $m \times n$. The notation $\otimes$ represents the Kronecker product, for which two useful properties are:

Property 1. *Let $A \in \mathbb{R}^{n\times m}$, $B \in \mathbb{R}^{p\times q}$, $D \in \mathbb{R}^{m\times r}$, $E \in \mathbb{R}^{q\times s}$, for arbitrary natural numbers m, n, p, q, r and s. Then $(A \otimes B)(D \otimes E) = (AD) \otimes (BE)$.*

Property 2. *Let $x \in \mathbb{R}^n$ and $y \in \mathbb{R}^m$. Then $x \otimes y = (x \otimes I_m)y = (I_n \otimes y)x$.*

(For Property 1 see Brewer, 1978. Property 2 follows from Property 1.)

The pseudoinverse of a matrix $A \in \mathbb{R}^{n\times m}$ is denoted by $A^+ \in \mathbb{R}^{m\times n}$. A basic property of the pseudoinverse is that $AA^+ = I_n$ if $\mathrm{rank}(A) = n$.

A linear operator defined by convolution by a weighting function is denoted by the symbol of the weighting function written in bold-face font, thus e.g.

$$\mathbf{F}G(t) \triangleq (F * G)(t) = \int_0^t F(t-\tau)G(\tau)\,d\tau.$$