A segmentation-based approach for detection of stiction in control valves

Anders Stenman[†,*], Krister Forsman[‡], Fredrik Gustafsson[†]

[†]Department of Electrical Engineering, Linköpings universitet
SE-581 83 Linköping, SWEDEN. Fax: +46 13 28 26 22

[‡]ABB Automation Systems AB
SE-257 33 Rydebäck, SWEDEN

December 4, 2002

Abstract

A method for detecting static friction (stiction) in control valves is proposed. The method is model-based and is inspired by ideas from the fields of change detection and multi-model mode estimation. Opposed to existing methods only limited process knowledge is needed and it is not required that the loop has oscillating behavior. The advantage of the method is illustrated in both numerical simulations and evaluations on real loop data.

Keywords: valve diagnosis, friction, stiction, multi-model mode estimation

1 Introduction

When a valve is used for controlling a flow in a pipe, a commonly encountered problem is that the valve sticks in certain positions, causing poor control performance. This is usually known as *stiction* or *stick-slip motion*.

A number of methods that diagnose stiction has been suggested in the literature, see, e.g., Deibert (1994), Taha et al. (1996), Ogawa (1998), Horch and Isaksson (1998) and Horch (1999). However, the methods require either detailed process knowledge or assume that the control loop oscillates in a certain way (which is not always the case). Here we propose a model-based method that removes these requirements. It utilizes statistical criteria and matched filters and is inspired by tools and theory from the fields of *change detection* and *segmentation* (Gustafsson, 2000).

The outline of the paper is as follows: Section 2 describes the basic set-up and defines the stiction detection problem. In Section 3 a simplified stiction model suitable for detection is established. Section 4 describes how this model can be used in a filter bank segmentation framework. Section 5 illustrates the proposed method in numerical simulations, and in Section 6 the conclusions are drawn.

2 Stiction detection

The work has been motivated by problems in flow control applications in pulp and paper mills. A typical such control loop is depicted in Figure 1. The controller is normally implemented in a hierarchical/distributed way. The high level (i.e., computer implemented) control system takes the setpoint r_t and the measured flow y_t as inputs and computes a control signal u_t using a PID-type control law. The valve is in turn controlled

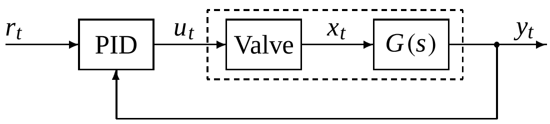

Figure 1: A flow control loop.

by a low level hydraulic or pneumatic control system. Now, if the valve suffers from stiction, the actual valve position x_t will most of the time be different from the commanded u_t which of course results in bad control performance.

It should be pointed out that if it is possible to measure x_t and/or perform special experiments, it would be rather easy to determine the degree of stiction in the valve. Unfortunately, this is rarely the case in practice. Therefore, it is very interesting to have at hand

[*]Now with NIRA Dynamics AB, Teknikringen 1F, SE-58330 Linköping, Sweden

algorithms that automatically detect and classify valve stiction using loop data collected from the high level control system under normal operating conditions. Motivated by this we formulate the stiction detection problem as follows: Given a dataset of sampled process outputs and control signals,

$$Z^N \triangleq \{(y_1, u_1), \ldots, (y_N, u_N)\}, \tag{1}$$

determine the degree of stiction present in the valve.

In this contribution we utilize a model-based approach for the detection problem. However, in contrast to previously proposed observer-based methods by Deibert (1994) and Horch and Isaksson (1998), which require detailed process knowledge (and thus are hard to tune), we here base our design on the relatively simple valve model

$$x_t = \begin{cases} x_{t-1}, & \text{if } |u_t - x_{t-1}| \leq d, \\ u_t, & \text{otherwise.} \end{cases} \tag{2}$$

which was suggested by Hägglund (1999). This model will form the basis for the stiction model derived in the the next section.

3 Stiction Model

According to the simple model (2), the valve position x_t is piecewise constant regarded as a function of time. A suitable model for detecting stick-slip behavior in the valve is thus

$$x_t = (1 - \delta_t)x_{t-1} + \delta_t u_t \tag{3}$$

where δ_t is introduced as a binary *mode parameter* with the property

$$\delta_t = \begin{cases} 1, & \text{if a jump in } x_t \text{ occurs,} \\ 0, & \text{otherwise.} \end{cases} \tag{4}$$

By assuming linear pipe dynamics we get the complete stiction model from u_t to y_t according to Figure 2, where $G_T(q)$ is the sampled version of $G(s)$. In terms

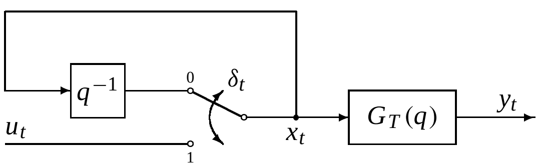

Figure 2: The assumed stiction model.

of this model, the diagnosis problem is to investigate the magnitudes of the jumps in x_t: The larger the jumps are, the more fatal is the stiction in the valve. In Section 4 we will show how this can be done using tools and theory from the fields of *change detection* and *segmentation*.

In most situations the process dynamics $G_T(q)$ are unknown and have to be estimated from data. We here assume that it can be modeled as a (possibly time-varying) linear regression

$$y_t = \varphi_t^T \theta + e_t, \tag{5a}$$

with

$$\theta = (a_1, \ldots, a_{n_a}, b_1, \ldots, b_{n_b})^T \tag{5b}$$

and

$$\varphi_t = (-y_{t-1}, \ \ldots, -y_{t-n_a}, \\ x_{t-n_k}, \ \ldots, x_{t-n_b-n_k+1})^T. \tag{5c}$$

That is, an ARX model with orders n_a, n_b and n_k. Furthermore, the measurement errors $\{e_t\}$ are assumed to be i.i.d. with zero means and variances λ. If x_t is known the system parameters θ can be easily estimated via standard least squares;

$$\hat{\theta}_t = R_t^{-1} f_t, \tag{6}$$

where

$$R_t \triangleq \sum_{k=1}^{t} \mu^{t-k} \varphi_k \varphi_k^T \quad \text{and} \quad f_t \triangleq \sum_{k=1}^{t} \mu^{t-k} \varphi_k y_k,$$

and μ is an exponential forgetting factor satisfying $0 < \mu \leq 1$.

4 Segmentation using filter banks

From the stiction model established in Section 3 it is clear that a major step of the algorithm will be to estimate the mode sequence

$$\delta^N \triangleq (\delta_1, \ \ldots, \delta_N). \tag{7}$$

In the change detection literature (see, e.g., Gustafsson (2000)) this is usually referred to as *segmentation*. In order to solve the segmentation problem though we need to formulate an optimality criterion.

4.1 Optimality Criterion

A naive solution to the segmentation problem is to consider all possible mode sequences and choose the one that maximizes some optimality criterion, i.e.,

$$\widehat{\delta^N} = \arg\max_{\delta^N} \mathcal{V}(\delta^N). \tag{8}$$

From a multi-model viewpoint it is natural to define $\mathcal{V}(\cdot)$ through the *a posteriori probability function*,

$$p(\delta^N|y^N) = p(y^N|\delta^N)\frac{p(\delta^N)}{p(y^N)}, \tag{9}$$

where $p(y^N|\delta^N)$ is the probability that we have observed $y^N = (y_1, \dots, y_N)$ given δ^N, $p(\delta^N)$ is a prior on the mode sequence, and $p(y^N)$ is a normalizing constant (see Appendix A for a derivation of the probabilities). The estimate (8) will thus be referred to as the *maximum a posteriori*, MAP, estimate.

Maximizing (9) is equivalent to maximizing its logarithm. Under the assumption of a fixed jump probability q, i.e.,

$$p(\delta^N) = q^n(1-q)^{N-n},$$

we end up with (see Appendix A)

$$\begin{aligned}\mathcal{V}(\delta^N) &= \log p(\delta^N|y^N) \\ &\approx -2n\log\frac{1-q}{q} + \log\det P_N(\delta^N) \\ &\quad - (N-2)\log V_N(\delta^N), \end{aligned}\tag{10}$$

where n is the number of jumps, $P_N(\delta^N) \triangleq \lambda R_N^{-1}$ and

$$V_N(\delta^N) \triangleq \sum_{k=1}^{N}(y_k - \varphi_k^T\hat{\theta})^2.$$

However, it is of course possible to use a more general prior $p(\delta^N)$ which may depend on both u_t and x_t.

The statistics P_N and V_N used in the criterion (10) can be computed recursively using the RLS (recursive least squares) algorithm (Ljung, 1990),

$$K_t = \bar{P}_{t-1}\varphi_t\left(\varphi_t^T\bar{P}_{t-1}\varphi_t + \mu\right)^{-1}, \tag{11a}$$

$$\hat{\theta}_t = \hat{\theta}_{t-1} + K_t\left(y_t - \varphi_t^T\hat{\theta}_{t-1}\right), \tag{11b}$$

$$\bar{P}_t = \bar{P}_{t-1} - K_t\varphi_t^T\bar{P}_{t-1}, \tag{11c}$$

and

$$V_t = V_{t-1} + \left(y_t - \varphi_t^T\hat{\theta}_t\right)^2, \tag{11d}$$

where, as before, μ is the exponential forgetting factor and $\bar{P}_t \approx P_t$.

4.2 Local search for optimum

The optimal segmentation is obtained by solving (8). However, direct, "brute force" optimization of the MAP criterion (10) is intractable because of the exponential complexity (we have 2^N different mode sequences to choose among). We thus have to resort to some kind of approximation. Here we have adopted the local tree search approach described in Gustafsson (2000).

The evolution of the mode sequence can be illustrated by the growing tree in Figure 3. At each time instant, every branch in the tree might split into two new branches where 0 means no jump and 1 corresponds to a jump in the valve state. The basic idea in the search approach is now to only consider the most likely sequence among the ones that jump according to the following algorithm:

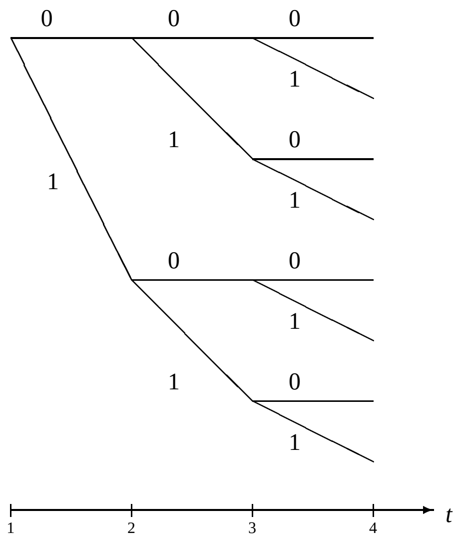

Figure 3: The growing tree of possible mode sequences. A path labeled 0 means that the valve is keeping its old position whereas a path labeled 1 corresponds to a change.

1. Compute recursively the a posteriori probabilities (9) using a bank of M estimators, each one matched to a particular mode sequence.
2. Maintain the fixed number of sequences (M) using the following heuristics:
   (a) Let only the most probable sequence split.
   (b) Cut off the least probable sequence, so that only M ones are left.

Useful restrictions on step 2 above are to assume a *minimum segment length* (i.e., the most probable sequence is allowed to split only if it is not too young), and to guarantee that all sequences have a *minimum life-length* (i.e., assure that they are not cut off immediately after they are born).

The algorithm is currently run in off-line mode. However, due to the inherent nature of the above search procedure, it would be straightforward to convert the algorithm to an on-line equivalent.

4.3 Tuning parameters

The search algorithm above does only require minor a priori knowledge. All that is essentially needed is a recorded dataset Z^N. The main tuning parameters are the number of filters M, the jump probability q, the minimum segment length *MINSEG*, the minimum life-length LL and the model orders n_a, n_b and n_k. In most situations it will however be sufficient to consider models of low orders.

4.4 Tuning guidelines

Tuning guidelines goes here!

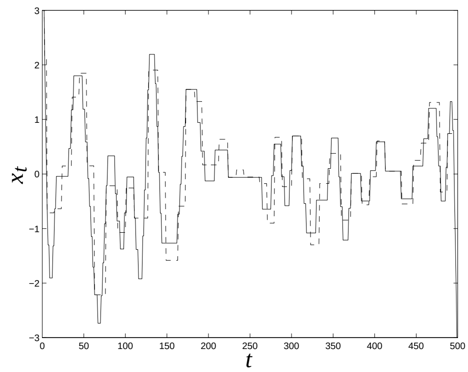

Figure 4: A subset of true (solid) and estimated (dashed) valve positions.

5 Numerical Evaluations

In order to evaluate our proposed detection scheme, we have applied it to both simulated and real control loop data.

5.1 Simulated Data

As a first example we consider simulated data. A band-limited white noise sequence of length $N = 5000$ was used as input $\{u_t\}$ and was fed into the valve model (2) assuming a dead band of $d = 0.5$. The measured output y_t was then formed by filtering the valve signal x_t through a second order linear filter (chosen randomly as)

$$G_T(q) = \frac{0.7q^{-1} + 0.6q^{-2}}{1 - 0.5q^{-1} + 0.2q^{-2}},$$

and adding normally distributed measurement noise e_t with zero mean and variance 0.1 to the filter output.

The detection algorithm with $M = 12$, $q = 0.5$, $LL = 8$, $MINSEG = 8$ and the true model order was applied to the dataset Z^N. A subset of the true and estimated valve positions are shown in Figure 4. We see that the estimated valve position follows the true one quite well. In this case it might however be more interesting to investigate the distribution of the estimated jumps. A histogram over those is shown in Figure 5 (a). We see that we get a well-defined peak for $d \approx 0.5$ which indicates that this jump size dominates in $\hat{x}_t$. This is consistent with the generated data set.

Since the valve model (3) assumes that the valve position is piecewise constant, the peak in the histogram will be more "un-sharp" the smaller the level of stiction becomes. This has also been verified by simulations. Figure 5 (b) shows what happens if the above experiment is repeated with dead band $d = 1$. We see that the peak in the histogram now is sharper and more well-defined.

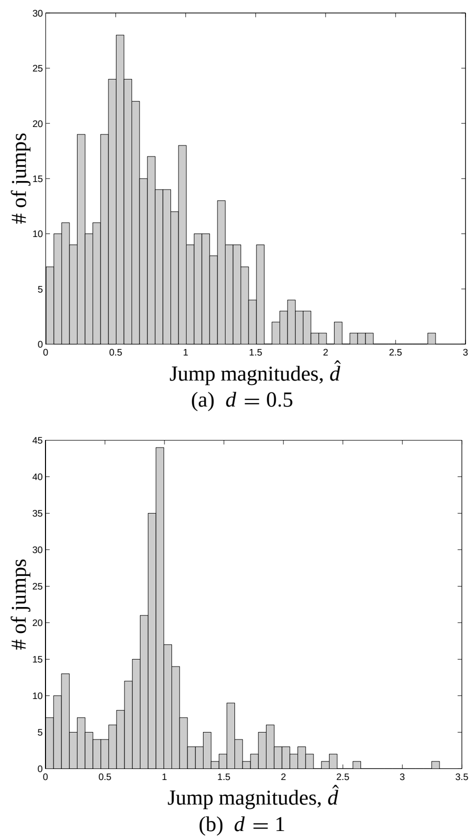

Figure 5: Histogram over jump magnitudes in $\hat{x}_t$.

5.2 Measured Data

The proposed algorithm has also been applied to the dataset of Figure 6, which have been collected from a steam pressure loop in a paper mill. Here we observe

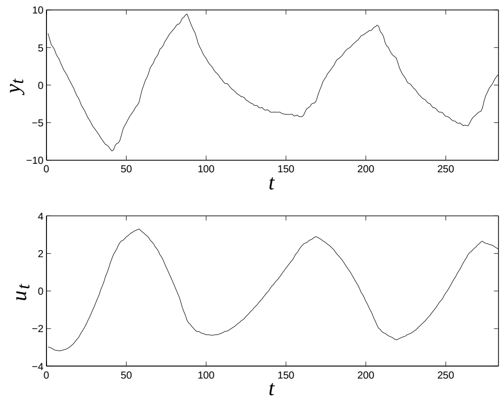

Figure 6: Data from a steam pressure loop.

that although the controller signal u_t is smooth, the process output y_t seems to have a discontinuous derivative. This might be a symptom of stiction. It is thus interesting to find out what the proposed detection algorithm will tell us. Figure 7 shows the result obtained using parameter values $M = 10$, $q = 0.5$, $LL = 6$, $MINSEG = 6$ and a first order ARX model with delay $n_k = 10$. We see that the estimated valve position x_t shows large jumps. This is a clear indication of stick-slip motion.

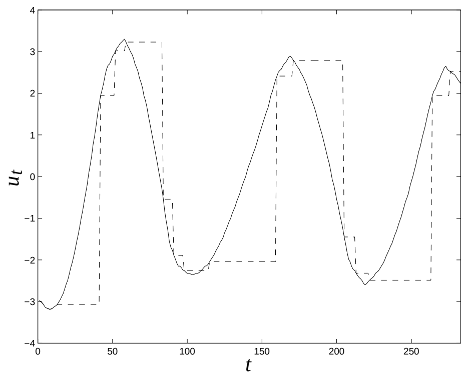

Figure 7: Control signal (solid) and estimated valve position (dashed) for the steam pressure loop.

6 Conclusions

A method for detecting static friction (stiction) in control valves has been proposed, inspired by multi-model mode estimation techniques. Opposed to existing methods only limited process knowledge is needed and it is not required that the loop has oscillating behavior. Numerical evaluations have verified that the method seems to perform well on both simulated and real loop data.

The algorithm is currently run in off-line mode. However, since the search for optimality actually is performed in a local manner, it would be straightforward to convert the algorithm to an on-line equivalent. All that is needed is an alarm device that raises an alarm when a distinct peak is found in the histogram of the valve jumps.

Extensions of the method is to consider more advanced models for the jump probability $p(\delta^N)$. Up til now we have only considered fixed probabilities, but it is obvious that it would be advantageous to let these be dependent of u_t (and $\hat{x}_t$). Another problem that has to be addressed is the common use of dead bands in the analog input module of control systems. This will of course give rise to the same type of discontinuity in data as stiction does, and we must ensure that the estimation routine is capable of distinguishing between the two cases.

Acknowledgments

This work has been carried out as part of the VINNOVA competence center for integrated control and information systems (ISIS) at Linköpings universitet.

References

R. Deibert. Model based fault detection of valves in flow control loops. In *Preprints of the IFAC Symposium on fault detection, supervision and safety for technical processes: Safeprocess 94, Espoo, Finland*, volume 2, pages 445–450, 1994.

F. Gustafsson. *Adaptive Filtering and Change Detection*. John Wiley & Sons, Ltd, 2000.

T. Hägglund. Private communication, 1999.

A. Horch. A simple method for detection of stiction in control valves. *Control Engineering Practice*, 7(10): 1221–1231, 1999.

A. Horch and A.J. Isaksson. A method for detection of stiction in control valves. In *Preceedings of the IFAC Workshop on On-line Fault Detection and Supervision in the Chemical Process Industry, France*, 1998.

L. Ljung. *System Identification – Theory for the User*. Prentice Hall, New Jersey, 2nd edition, 1990.

S. Ogawa. A data analysis and graphical presentation system for control loop performance assessment. In *TAPPI Process Control Conference 98, Vancouver, Canada*, 1998.

O. Taha, G.A. Dumont, and M.S. Davies. Detection and diagnosis of oscillations in control loops. In *Preceedings of the 35th IEEE Conference on Decision and Control, Kobe, Japan*, pages 2432–2437, 1996.

A Statistical Criteria

First note from the model (3) that if δ^N is known then so is also x^N. By assuming Gaussian noise, the likelihood for observing data y_t given δ^N, θ and noise variance λ is

$$p(y_t|\delta^N,\theta,\lambda) = \frac{1}{\sqrt{2\pi\lambda}} \exp\left(-\frac{1}{2\lambda}\left(y_t - \varphi_t^T\theta\right)^2\right)$$

The total likelihood is thus

$$p(y^N|\delta^N,\theta,\lambda) = \prod_{t=1}^{N} p(y_t|\delta^N,\theta,\lambda). \tag{12}$$

The nuisance parameters θ and λ can be removed by means of *marginalization* where (12) is integrated with respect to the prior distributions of those parameters, i.e.,

$$p(y^N|\delta^N) = \int_{\theta,\lambda} p(y^N|\delta^N,\theta,\lambda) \times p(\theta|\lambda)\,p(\lambda)\,d\theta d\lambda. \tag{13}$$

The a posteriori probability function then follows from Bayes' theorem,

$$p(\delta^N|y^N) = p(y^N|\delta^N)\frac{p(\delta^N)}{p(y^N)}. \tag{14}$$

Let us derive (13) for the stiction model. By assuming a flat prior, $p(\theta) \sim 1$, we have that

$$\begin{aligned} p(y^N|\delta^N,\lambda) &= \int_\theta p(y^N|\delta^N,\theta,\lambda)\,d\theta \\ &= (2\pi)^{(d-N)/2}(\det P_N)^{1/2}\lambda^{-N/2}\exp\left(-\frac{1}{2\lambda}V_N\right) \end{aligned}$$

where $P_N = \operatorname{Cov}\hat{\theta} = \lambda R_N^{-1}$ and $V_N = \sum_{t=1}^N (y_t - \varphi_t^T\hat{\theta})^2$. Again assuming a flat prior, $p(\lambda) \sim 1$, and utilizing the *inverse Wishart* probability density function,

$$p(\lambda|V,m) = \frac{V^{m/2}\exp\left(-\frac{1}{2\lambda}V\right)}{2^{m/2}\Gamma(m/2)\lambda^{(m+2)/2}},$$

we get

$$\begin{aligned} p(y^N|\delta^N) &= \int_\lambda p(y^N|\delta^N,\lambda)\,d\lambda \\ &= (2\pi)^{(d-N)/2}(\det P_N)^{1/2}\frac{2^{(N-2)/2}\cdot\Gamma\left(\frac{N-2}{2}\right)}{V_N^{(N-2)/2}}. \end{aligned}$$

The prior $p(\delta^N)$ is a user's choice. A simple approach is to assume a fixed probability q for a jump at each time instant. This gives

$$p(\delta^N) = q^n(1-q)^{N-n} = \frac{q^n}{(1-q)^n}(1-q)^N.$$

where n is the number of jumps. Taking the logarithm of the a posteriori probability (14) thus yields

$$\begin{aligned} 2\log p(\delta^N|y^N) = C - 2n\log\frac{1-q}{q} + \log\det P_N \\ - (N-2)\log V_N, \end{aligned}$$

where C contains all terms that are independent of δ^N.