DEVELOPMENT OF A TECHNIQUE FOR PERFORMANCE EVALUATION OF INDUSTRIAL CONTROLLERS

M. Rossi[1], C. Scali[1*], M. Amadei[2]
[1] *Department of Chemical Engineering - CPCLab - University of Pisa (Italy)*
[2] *Research and technology -- Polimeri Europa - Milano (Italy)*

Abstract: The paper describes a method to account for different issues of performance monitoring of industrial control systems, under SISO control: detection of poorly tuned loops, process identification, controllers retuning and evaluation of perfomance improvements. The procedure can be completely automated and applied on-line or off-line; it starts from the analysis of plants data and ends up with a suggestion to the operator about the new controller settings. Characteristics and effectiveness of the technique are illustrated by simulations results and application to industrial plant data.

Keywords: Performance Monitoring, Identification, Controller Tuning, Industrial Control Systems

1. INTRODUCTION

Performance monitoring in industrial plants is an aspect of increasing importance nowadays, as witnessed by large efforts in advanced academic research and in plants applications. Several aspects still must be resolved in a systematic way: both theoretical (metrics to be used to evaluate control performance, extension to MIMO systems) and practical (automated application on industrial plants, minimisation of perturbation on plant operation, interaction with operators). An updated overview of these topics can be found in Thornhill and Seborg (2002).

A control loop can perform poorly for several reasons, as valve stiction, sensors failures, incorrect tuning of controllers. In the perspective of developing a global tool accounting for different causes, the issue of performance evaluation of loops controlled by PI(D) regulators is addressed in this paper.
This aspect has large importance, owing to their large diffusion in industrial plants; these controllers are not tuned at their best (Ender, 1993), because of the general tendency to avoid oscillations (synonym of instability) and therefore to apply conservative tuning, which results in slow responses. This is also a consequence of the fact that standards about the procedure are not strictly established on the plant and tuning is left to operators' skill; very often, the company policy, while asking for best use of available technologies, assigns few resources to this important task.

* e-mail: scali@ing.unipi.it

In addition, even a perfectly tuned controller may become inadequate when the process undergoes large variations, due to changes in operating condition or to the inlet of external perturbations. Process changes can be detected by performing periodic identification (step and relay tests can be used for this purpose), followed by a new design / tuning of controllers (Åström and Hägglund., 1995).
Even though these procedures can be completely automated (leading to adaptive control / autotuning), there are some drawbacks, because:

- explicit perturbations must be introduced in the plant (even though the amplitude can be somehow controlled),
- may become time consuming, owing to slow process dynamics or reiteration of experiments (Yu' 2000, Marchetti and Scali, 2000).

Without the need of introducing any additional perturbation, all the required information about the behaviour of the controlled process, can be extracted from plant data, which are continuously recorded and archived in real-time data-bases, then available for further analysis.
Several different performance index have been developed for specific purposes; among the most known and widely used: Harris index (Harris, 1989), which allows to compare actual controller with optimal (minimum variance) controller; Idle index (Hägglund, 1999), which allows to detect sluggish responses or the Oscillation Detection technique (Hägglund, 1995), which allows to detect too oscillating responses. These indexes (or their modifications) can be applied for on-line or off-line analysis; the developments of related software tools, as well as their implementation (DCS or external

computers) is still under investigation (Hägglund, 2002).

It is highly desirable that the performance analysis procedure, with indication of poorly acting loops, is associated with a new design/tuning of controllers and evaluation of achievable improvements. Therefore the development of software tools is called for, able to assist the operator in taking key decisions, with possible automated applications on the plant, after a suitable time of training on plant data. With this short introduction to the problem, as first this paper will briefly review basic aspects of two performance indexes to detect poor behaviour of the controller. Then, the different steps of the procedure, which goes through the steps of performance monitoring, process identification, controller tuning and improvement evaluation, will be illustrated. Finally, the effectiveness of the technique is illustrated by simulations results and application to real plant data, drawing some conclusions and indications for future work.

2. INDEXES TO MONITOR PLANT PERFORMANCE

In the reference scheme of a SISO control loop (Figure 1), *P* and *Pd* indicate the effect of manipulated (*u*) and disturbance (*d*) on the controlled (*y*) variable, effects which can be different in the general case.

The application of performance indexes is able to evaluate from the analysis of plant data when a controller gives too weak action, (slow closed loop response), or too strong action (oscillating response). The computation of these indexes should be as simple as possible, requiring few information, (for instance based only on values of controlled and manipulated variables), and giving rise to a clear classification of controller behaviour.
Slow responses can be detected by means of the Idle Index (Hägglund, 1999). The computation of this index is based on two characteristic times: t_{pos}, time interval when the product of the two gradients of controlled (CV) and manipulated (MV) is positive, and t_{neg}, (negative product of gradients).
Then it is possible to get a normalised index in the range [-1÷1], which for slow disturbance suppression approaches 1, while for faster responses assumes negative values:

$$Ii = \frac{t_{pos} - t_{neg}}{t_{pos} + t_{neg}} \qquad (1)$$

Too oscillating responses can be put into evidence by the oscillation detection technique (OD), presented by (Hägglund, 1995). The analysis can be split into two parts: detection of significant perturbations (*anomalies*) and detection of persistent oscillations.

Table 1: Classification of the perturbance response

OD	Ii ∈ [-1; -0.7]	Ii ∈ [-0.7; -0.4]	Ii ∈ [-0.4; 0.4]	Ii ∈ [0.4; 0.7]	Ii ∈ [0.7; 1]
1÷5	Good	Good	Good	Poor	Bad
6÷10	Accept.	Accept.	Poor	Poor	Bad
>10	Bad	Bad	Bad	Bad	Bad

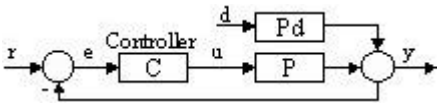


Fig.1: The reference scheme

Each single oscillation is characterised by its IAE:

$$IAE = \int_{t_{i-1}}^{t_i} |e(t)| dt \qquad (2)$$

where t_{i-1} e t_i are successive times where e(t)=0. To be significant, the value must be above the IAE of a half-period of a sinusoidal oscillation having a defined amplitude and frequency (amplitude equal to 1% of control range and frequency equal to the estimated critical frequency of the process). To detect the presence of a persistent oscillation, it is necessary to detect a significant number of oscillations n_{lim} over a supervision time T_{sup}. In this case the reference value is n_{lim} 10 in the time window T_{sup}.

By adopting the Ii and OD indexes it is possible to achieve a quantitative evaluation of the closed loop response to a perturbation, as reported in Table 1.

3. THE PROPOSED TECHNIQUE

As anticipated in the introduction, the proposed technique and the associated software tool developed for its implementation, has the objectives of performing different tasks of:

1) performance monitoring, with detection of "*anomalies*" (poorly performing loops),
2) identification of process and disturbance dynamics (Fig. 1)
3) controller tuning according to a desired performance criterion,
4) evaluation of expected improvement with the adoption of the new controller.

This architecture is depicted in Figure 2.

In the sequel these aspects will be fully illustrated, putting into evidence also implementation issues (DCS or External Computer, on-line or off-line) and interaction with the operator. About this point, while the final goal is a complete automated system (able to work by default from plant data to final tuning), interaction with the operator can always be introduced to force (improve) computer operations on the basis of specific experience on the plant.

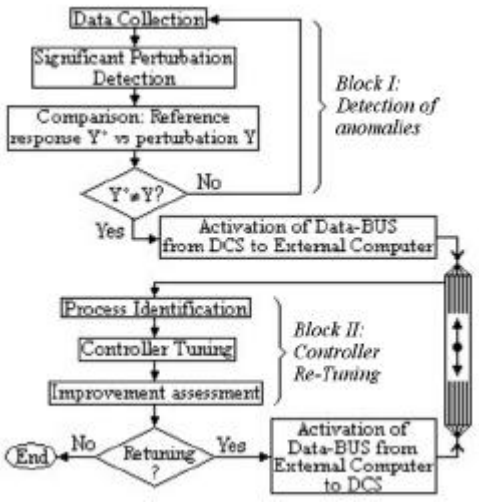


Fig. 2. The architecture of the proposed tool

3.1. Performance Monitoring

Plant data are acquired by the DCS system at given sampling times according to the usual procedure; they consist in values of controlled (CV) and manipulated variables (MV), set points, controller parameters, information about manual/automatic operation. In addition to routine elaboration (Figure 2, Block I):

a) data are analysed according to the OD technique and the presence of a "significant perturbation" in the controlled variable (Y) is detected,
b) this perturbation is compared with a reference response (Y°); if noticeable variation are found, the remaining part of the procedure is activated.

To be noted that:

- the presence of an *"anomaly"* does not mean necessarily that it is possible to get improvements: therefore it is compared with the last good response (Y°), which must be recorded in the system;
- both computations to detect the "*anomaly"*, and data about Y°, do not require heavy additional load (computation and memory); they can be performed in the DCS, in agreement with Hägglund, (2002).

3.2. Process Identification

Once an "*anomalous response*" (which can be improved) has been detected by the DCS, the procedure of process identification is activated in the external computer, which receives all the necessary data (First step of block II, Figure 2). All elaboration of data for this and following stages can be accomplished by using standard Matlab and Simulink routines.
The output response of a controlled loop to a disturbance can be expressed as (Figure 1):

$$Y(d) = \frac{Pd}{1+PC}\,d \qquad (3)$$

The controller C is known, the disturbance d is assumed as unitary step; for P and Pd, models represented by Second Order Plus Time Delay (SOPTD) transfer functions have been assumed:

$$\tilde{P}(s) = \frac{K_p}{t_p{}^2 s^2 + 2t_p x_p s + 1} \cdot e^{-\theta_p s} \qquad (4)$$

$$\tilde{P}d(s) = \frac{K_d}{t_d{}^2 s^2 + 2t_d x_d s + 1} \cdot e^{-\theta_d s} \qquad (5)$$

Both $\tilde{P}$ and $\tilde{P}d$ are identified, as the output response for a given controller may become suboptimal when P or Pd change. The values of parameters are determined by minimising the sum of mean square error (SMSE) between plant (Yi) and model (i) response (N is the number of data):

$$SMSE = \frac{1}{N} \cdot \sqrt{\sum_{i=1}^{N} \left[Y_i - \tilde{Y}_i\right]^2} \qquad (6)$$

Minimisation is carried out by a modified Simplex method (Nelder and Mead, 1965), based on the Matlab function "fminsearch.m". The simplex method may fail to converge or may converge to a suboptimal solution, owing to the presence of local minima, especially when a large number of parameters are present.

About the number of parameters, it must be noted that K_d and θ_d are not real unknowns, as they influence only the amplitude of output response (and not the shape), and the time when the perturbation shows up: unknowns can be reduced to 6.

About the problem of convergence, the initial guess on the values of parameters is very important and then initial knowledge on the process plays an important role. Starting from the values of previous models (memorised until new models are computed), can be a good choice in the case of not too large variations in process parameters. More in general, an estimate of initial values for the parameters K_p and θ_p, (and from them t_p, t_d, x_p, x_d), can be given on the basis of CV and MV values; (details in Rossi 2002).

The number of points and characteristic times of the response come from the DCS; among them: time of inlet of a perturbation (t_{in}), time of detection of a perturbation (t_{det}), time to reach a new steady state (t_{ss}). This is the system default: the operator can decide to start the retuning procedure before t_{ss}, acting with a new controller to suppress the disturbance and this will bring advantages for long lasting perturbations (details in Figure 4).

3.3. *Retuning and Evaluation of Performance Improvement*

Controller retuning is performed assuming an ITAE objective; for PI controllers:

$$\min_{C}\left(ITAE\right)=\min_{Kc,ti}\left(\int\left|e(t)\right|\cdot t\cdot dt\right) \qquad (7)$$

The ITAE choice can be desirable in industrial process control owing to its characteristics of reducing tails in output response; (other objective functions can be adopted in the tool, if the case). To evaluate the effectiveness of a retuning, in terms of achievable performance, the auxiliary index RITAE has been defined as:

$$RITAE=\frac{ITAE_{new}-ITAE_{\min}}{ITAE_{old}-ITAE_{\min}} \qquad (8)$$

where: *new/old* stand for controller after/before retuning, *min* stands for ideal controller (minimum error). Values of RITAE: are strictly positive; in the range 0÷1, indicate improvements of performance; larger advantages are expected for values closer to 0. An illustrative example is reported in Figure 3 and in Table 2.

Figure 3 and table 2 also summarise information communicated to the operator at the end of the global procedure: old and new time responses and controller settings, minimum error response, values of the performance index (Ii, OD, RITAE). One additional information regards values of the actual model compared with old one and allows to distinguish performance deterioration due to change in the process or in the perturbation dynamics. At this point the operator has all the necessary information; in off-line operation: to evaluate (and grade) controller behaviour; in on-line operation to decide if it is worth to change controller parameters, something that can be accomplished manually or automatically, after operator consent. As final objective, once the procedure has been fully tested on plant data, changes in controllers parameters will be accomplished directly, with the supervising operator having the option of interrupting the automatic operation in every moment.

4. SIMULATION RESULTS

The technique has been tested by simulation on processes (P) and perturbations (Pd), represented by different transfer functions of First, Second, Higher order, Plus Time Delay. By varying values of parameters in a large range, a wide class of dynamics of possible interest in industrial applications have been analysed.
The key step of the procedure is the identification. The goodness of identification has been evaluated by

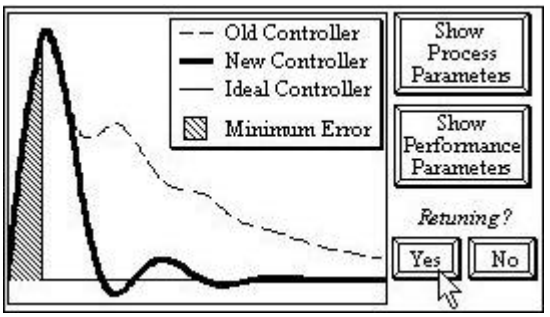


Fig. 3: Examples of output responses, with old, new and ideal controller (minimum error)

Table 2: Performance Parameters

C	Kc	τi	Ii	OD	RITAE
Old	0.58	17.5	+0.48	1	0.15
New	0.43	4.37	-0.11	3	

comparing closed loop responses obtainable by PI controllers based on the models ($\tilde{P}$,$\tilde{P}_d$) and on a perfect knowledge of the process (P, P_d). The visual comparison of responses and an analytical index (DITAE, defined as (percentage) Difference of ITAE), show that identification is able to capture the essential dynamics for control purposes (details in Rossi, 2002).

Some general results can be pointed out:

- The technique based on the modified simplex method showed very good convergence properties.
- Best identification (time, accuracy) is obtained when process dynamics are the same of model (SOPTD), but a good fitting is obtained for a wide class of processes.
- Also, delay dominant dynamics ($\theta/\tau \gg 1$), as well as underdamped responses, are *easier* to identify, in terms of duration and accuracy.
- Only in the case of very high order (overdamped) disturbance dynamics, the technique may fail and this is communicated to the operator, who can change some default settings.

An example of typical simulation results is reported in Figure 4. Perturbation 1 and 2 can not be improved (Y=Y°); perturbation 3 (a decrease in the feed flow rate), causes a more oscillating response which is detected by a comparison between Y and Y° (the ratio in this case). By default the system waits until new steady state conditions are reached and then identifies changes, retunes the controller and, for subsequent perturbations, is able to improve performance according to response 4.
The operator can decide to act before the new steady state has been reached (*Start elaboration*); this way, after the very short elaboration time and controller retuning (*End elaboration*), it is possible to act on

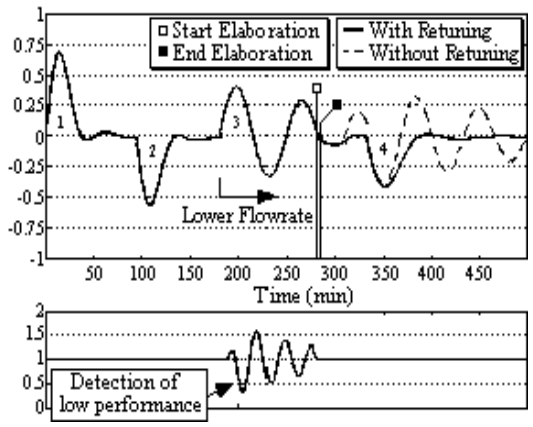


Figure 4: Example of typical simulation results

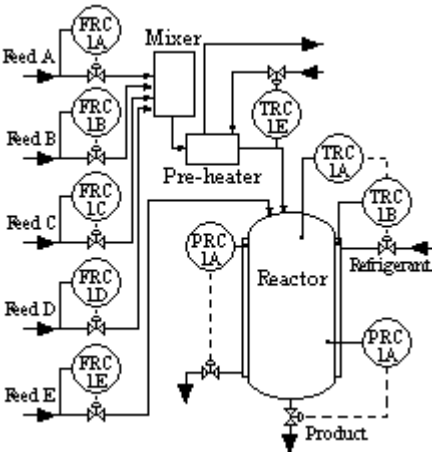


Fig. 5: Plant scheme

the perturbation, suppressing last oscillations. It is evident that the elaboration of data can start only when the response dynamics is sufficiently developed, in order to allow a correct identification; in this stage a crucial role is played by operator experience; anyway the system can work by default, without any assistance.

The robustness of the technique to the noise on plant data has been investigated. The effect of noise has been taken into account by adding random errors to clean data; internal parameters to model the effect of noise have been selected in order to make it as similar as possible to the noise present on industrial data (see section 5).
The ratio between noise and signal amplitude (N/S) has been changed in order to analyse the deterioration of results and evaluate a maximum allowable amplitude.
The presence of noise on plant data influences the stages of detection of a significant *anomaly*, through the comparison between Y and Y°, and the stage of identification.
Errors due to noise are reflected in achievable closed loop performance by the regulator based on the model. A systematic evaluation of performance deterioration has been accomplished for different process dynamics by means of the DITAE index. Assuming a maximum DITAE error equal to 10%, for acceptable performance, a ratio N/S=30% is allowed; then the technique can be considered sufficiently robust for applications on industrial data.

5. APPLICATION TO PLANT DATA

The method has been tested for an off-line application on industrial data, kindly made available by Polimeri Europa s.p.a., from a polybutadiene plant for the production of SBR.

The plant section under analysis consists of a mixer of reagents, a pre-heater and the reactor cooled by an external jacket. Several sources of perturbation can affect the polymerisation reactor: changes in reagents/product properties, fouling of heat exchange surfaces, different stages of the batch process and changes in environmental conditions.

Pre-heating is necessary to make easier to maintain the reaction temperature at the optimal value. This variable is the most critical to control both for safety and high quality control: only 3 °C of deviation from set point values are allowed: larger values will cause the activation of alarms and eventually shut down of the plant. Several sources of perturbation can affect polymerisation: changes in reagents/product properties and environmental conditions, fouling of heat exchange surfaces, different stages of the batch process.

For this reason, pre-heater outlet and reactor temperatures are controlled by one PID and two cascaded PI controllers, respectively, while all other loops are controlled by simple PI controllers (Fig.5):

The evaluation of performance has been carried out following the logical scheme reported in Figure 2; the technique has been extended to cascade control loops, without substantial modifications (Rossi, 2002). When the perturbation enters in the inner loop, the method is applied as such; when in the outer loop, the process dynamics is changed to include also the inner loop (process plus controller).

The following data were available (Excel format):

- Controlled variable: Y(t)
- Manipulated variable: u(t)
- Set-point value: SP
- Controller parameters k_c, τ_i, τ_d
- Controller status (Manual/Automatic)
- Valve opening (%)

Data refer to 5 flow, 2 pressure, 3 temperature (one cascade) control loops. From a preliminary analysis, the reaction can been indicated as the plant section more perturbed by external disturbances. In 24 hours of operation 3 significant *anomalies* have been

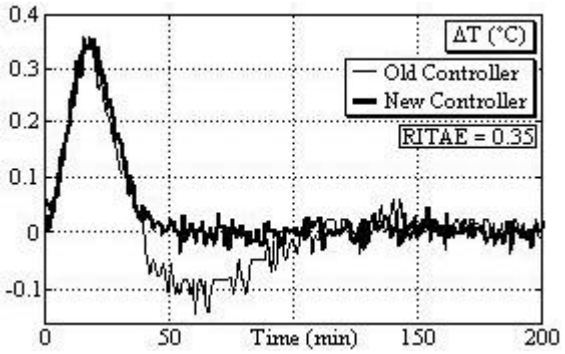


Fig. 6: Suppression of disturbance #1

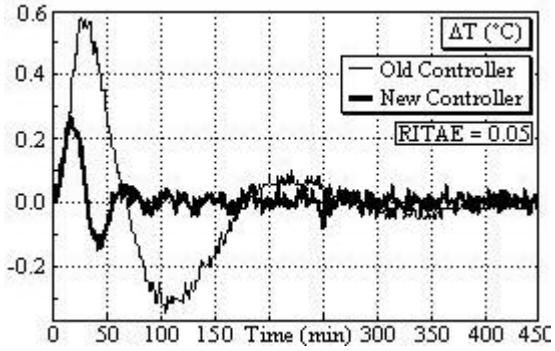


Fig. 7: Suppression of disturbance #2

detected by the analysis of the recorded plant data: the amplitude of deviations from set-point values is rather small, but it should be recalled that the a very tight control is required and that small deviations in reactor temperature propagate as larger perturbations in jacket temperature and flow rates.

For one case the technique was not able to identify the detected *anomaly* and this was explained in terms of two different perturbations acting simultaneously. For the other two cases, the application of the technique allows to improve the response: the perturbation is suppressed in a much lower time with the new controller and the RITAE index assumes values closer to zero (Figure 6 and 7).

6. CONCLUSIONS

The proposed method presents a complete approach to performance monitoring of SISO control loops, including different issues of detection of *anomalies*, process identification, controller retuning and improvements evaluation.

Only the first stage (detection) is accomplished in the DCS, with relatively low additional computational load, while remaining stages are developed in a parallel computer.

The capability of the proposed technique has been validated by simulation and by first off-line application to industrial data.

The technique shows to be very flexible, as it can be applied both off-line and on-line, can work complete automated procedure by using default settings or accept interaction with the operator in some key points.

Further work will be devoted to different objectives: introduction of constraints on control actions and of different objective functions in the retuning block; extension of the software package to detect valve stiction and sensor failures; on line applications to address implementation issues and to better define different levels operator interactions.

7. REFERENCES

Åström, K.J. and Hägglund T. (1995): "PID Controllers: Theory Design and Tuning"; *ISA, Research Triangle Park, N.C.(USA).*

Ender, D.B. (1993): "Process control performance: Not as good as you think.", *Control Engineering*, **40:10**, pp. 180-190.

Hägglund T. (2002): "A friction compensator for pneumatic control valves", *Journal of Process Control*, Vol. 12, pp. 897-904.

Hägglund T. (1999): "Automatic detection of sluggish control loops", *Control Engineering Practice*, Vol. 7, pp. 1505-1511.

Hägglund T. (1995): "A control loop performance monitor", *Control Engineering Practice*, Vol. 3, pp. 1543-1551.

Hägglund, T. (2002). Industrial Applications of Automatic Performance Monitoring Tools*; Proc. 15th IFAC World Congress; T-Mo-A11-1*, Barcelona, Spain.

Harris T.J. (1989): "Assessment of Control loop performance", *The Canadian Journal of Chemical Engineering*, Vol. 67, pp. 856-861.

Marchetti, G. and Scali C. (2000). Use of Modified Relay Techniques for the Design of Model Based Controllers for Chemical Processes. *Ind. Eng. Chem. Res.*, **39**, 3325-3334.

Nelder J.A. and Mead R. (1965): "A simplex method for function minimization", *The Computer Journal*, Vol. 7, Issue 4, pp. 308-313.

Rossi M. (2002): Monitoring and Performance Evaluation of Controlled Loops; Ms Thesis; Chem. Eng. Dept. (University of Pisa).

Thornhill N.F., Seborg D.E. (2002): "New Directions in Control System Performance Monitoring", *T-Mo-A11, Session of 15th IFAC World Congress*, Barcelona.

Yu, C.C. (1999): "Autotuning of PID Controllers", Springer-Verlag, London (UK).

PERFORMANCE ASSESSMENT OF CONSTRAINED CONTROLLERS

Lilong Huang [*,1] **Christos Georgakis** [**]

[*] *Chemical Process Modeling and Control Research Center and Department of Chemical Engineering Lehigh University, Bethlehem, Pennsylvania 18015*
[**] *Department of Chemical Engineering, Chemistry, and Material Science, 728 Rogers Hall, Polytechnic University Six Metrotech Center, Brooklyn, NY 11201 USA Phone: (718) 260- 3579; Fax : (718) 260-3125*

Abstract: The paper presents a new deterministic framework for assessing constrained control loop performance. The proposed dynamic performance index is based on the dynamic operating work of Uzturk and Georgakis (2002). It focuses on the time related characteristics of controllers' response to set-point changes or step-type disturbances. It explicitly takes into account the existence of constraints on manipulated variables. These constraints include minimum and maximum values as well as an upper limit on the rate of change of the input variables.

Keywords: Constrained Controller, Performance Assessment, Time Optimal control, Minimum Settling Time, Deterministic Disturbances.

1. INTRODUCTION

The controller design task aims to find a suitable controller given a model of the system to be controlled and a set of design goals. A well designed control system should satisfy both performance and robustness specifications. Performance specifications include stability, disturbance regulation, set-point tracking, transient response, and constraints (e.g. Boyd and Barratt, 1991). In practice, unfortunately, controllers seldom work as initially designed due to inapt assumptions and compromises made during design, improper controller tunings, and unaccounted model-plant mismatch, etc.

For reliable and profitable process control applications, the chemical industry is in need of effective controller performance monitoring and diagnosis technology. Harris (1989) reported the estimation of the minimum achievable variance of SISO controlled variable from 'normal' closed-loop data. Since then, *Minimum variance control* (Åström, 1970) has been widely used as a benchmark for assessing control loop performance (e.g. Desborough and Harris, 1992). Extensions of SISO control loop performance assessment techniques to MIMO cases were first addressed by Huang *et al.* (1997) and Harris *et al.* (1996).

Minimum variance control provides a lower stochastic bound on the achievable performance of any feedback controller if (i) the control objective is to minimize the steady-state output variance, (ii) the time delay is the leading performance limiting factor, (iii) and the disturbance acting on the process can be reduced to a filtered white noise. However, in most practical applications, it is the constraints on the manipulated variables, along with the time

[1] The financial support of the industrial sponsors of the Chemical Process Modeling and Control Research Center at Lehigh University is greatly appreciated.

delay and the inverse response dynamics, that place an upper limit on the achievable performance. With the wide availability of controllers that explicitly take into account constraints, the assessment of control loop performance needs to consider the effects of constraints as well. Ko and Edgar's (2001) approach deserves notice as it is the first attempt to explicitly account for constraints in control performance assessment.

The present paper describes a new deterministic framework for assessing constrained control loop performance. The proposed dynamic performance index focuses on the time related characteristics of the controller's response to set-point changes or step disturbances. It explicitly accounts for the constraints on manipulated variables, including magnitude and rate of change limits. The paper is organized as follows. In section 2, we review the minimum time-optimal control benchmark. It is used as the basis for the proposed performance index described in section 3. The demonstration examples are given in section 4, and section 5 provides the conclusions.

2. TIME DOMAIN CONSTRAINED APPROACH

The objective of this paper is to develop a performance index to assess controller performance of constrained systems with respect to deterministic disturbances. The motivations are as follows. As pointed out by MacGregor *et al.* (1984), in most chemical engineering processes, the major disturbances are not stochastic disturbances, but deterministic disturbances such as sudden loads on the system and set-point changes made by operators. Furthermore, Eriksson and Isaksson (1994) have shown, through a very interesting example, that the minimum variance control based performance assessment technique (e.g. Desborough and Harris, 1992) gives an inadequate measure of performance if the control objective is set-point tracking.

To develop a time-domain controller performance criterion, the minimum time optimal control is adopted as the benchmark to evaluate control loop performance. Minimum time optimal control explicitly takes into account the input constraints and provides a time domain upper bound of the achievable control performance. It is independent of the feedback control structure one might use on-line and reflects the inherent performance limitations of the process.

2.1 Minimum-time Optimal Control

Minimum-time optimal control aims to drive the process to the desired set-point within minimum settling time t_s, given process dynamics and constraints on controlled variables and manipulated variables. The final time constraints incorporated in the formulation of minimum-time optimal control problem ensure that the system reaches the set-point at the minimum settling time and stays there afterwards. In addition, set-points and disturbances entering the process are assumed to be in step form. The solution of the minimum time control problem can be computationally intensive. Linear programming (LP) technique is commonly used to solve the problem in the case of linear systems. (Refer to Uzturk and Georgakis (2002) for more details.)

2.2 Approximate Equivalence between Minimum Time Optimal Control and Minimum IAE or ISE Control

Simulations of several model SISO systems have shown the set-point response under minimum time optimal control to be overdamped. Very interestingly, minimum integral absolute error (IAE) control or minimum integral square error (ISE) control, if modified to only allow overdamped responses, perform almost identically to minimum time optimal control. This equivalence in performance is evaluated under any of the three criteria (settling time, IAE, and ISE) [2]. In the following sections, we refer as optimal control to any of these three control schemes.

The above equivalence among minimum time optimal control, modified minimum IAE control, and modified minimum ISE control shows that settling time, IAE, and ISE are similar performance measures. Hence, any of them can be applied as a candidate measure for performance assessment. In addition, for system models of order higher than two, we can estimate the minimum settling time by solving a modified minimum IAE or ISE control problem, which is computationally easier and involves one pass LP or QP optimization.

3. CONSTRAINED PERFORMANCE INDEX

The performance index can be defined as the ratio of the performance measure (settling time, IAE, ISE) under optimal control to that under present control. However, as reported by Uzturk and Georgakis (2002), for constrained controllers, the optimal achievable performance measure depends on operating conditions such as the initial operating points, desired set-points, and disturbances (see Figure 2 in Uzturk and Georgakis (2002)). As a result, the performance index used

[2] Due to space limitation, demonstration plots are not presented.

might vary with operating conditions. The major issue is whether the settling time, IAE, or ISE alone is a sufficient measure of controller performance. An additional concern comes when a process model is required for the estimation of the optimal performance criteria.

3.1 Proposed Performance Index

A new constrained performance index is proposed in this section which consists of three components related to integral absolute error, overshoot, and response time:

$$\eta \equiv \frac{1}{3}\left(\frac{\mathrm{IAE}^{\mathrm{ref}}}{\mathrm{IAE}} + \frac{r^{\mathrm{d}}}{r^{\mathrm{d}} + y^{\mathrm{os}}} + \frac{t_{\mathrm{s}}^{\mathrm{ref}}}{t_{\mathrm{s}}}\right) \qquad (1)$$

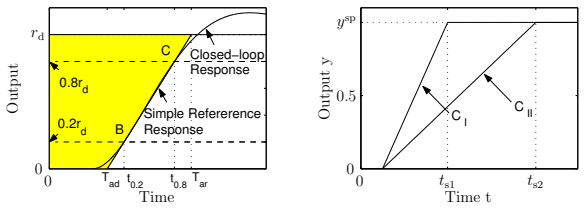
The first term is defined as the ratio of a reference integral absolute error (IAE) to that of the set-point response. The second term penalizes the overshoot, where r_{d} is set-point change and y^{os} is output overshoot (absolute value). The third term evaluates the ratio of a reference settling time to the actual settling time t_{s} of the set-point response. In the above definition, the settling time t_{s} is defined as the time when the set-point response reaches and remains inside a band which is equal to $\pm 3\%$ of the set-point change r_{d}.

If a process model is available, the reference IAE and settling time can be calculated by solving a constrained optimization problem.

Without a process model at hand, the reference settling time and reference IAE are estimated directly from the set-point response:

$$t_{\mathrm{s}}^{\mathrm{ref}} = t_{\mathrm{r}}; \qquad \mathrm{IAE}^{\mathrm{ref}} = \frac{1}{2} r_{\mathrm{d}}(T_{\mathrm{ad}} + T_{\mathrm{ar}}) \qquad (2)$$

The rise time t_{r}, defined as the time when the set-point response reaches the value $0.97 r_{\mathrm{d}}$, is adopted as the reference settling time. The reference IAE is the integral absolute error of a simple reference response, which is characterized by three parameters: the set-point change r_{d}, the apparent rise time T_{ar}, and the apparent dead time T_{ad} (see Fig. 1(a)). The main advantage with this choice of reference settling time and reference IAE is that



(a) Definition of η (b) Drawback of η

Fig. 1. Demonstration diagram for definition & drawback of proposed performance index η

there is no requirement for a process model to estimate the performance index η. In addition, the first and third terms in Eq. (1) implicitly evaluate the contribution of the oscillating part of the set-point response to the IAE and the settling time, respectively.

The parameters (r_{d}, T_{ar}, and T_{ad}) can be determined graphically from the set-point response (see Fig. 1(a)). The set-point change (r_{d}) can be determined from the final steady-state level of the process output. $t_{0.2}$ is the time when the set-point response reaches the value $0.2 r_{\mathrm{d}}$ (point B), and $t_{0.8}$ is the time when the set-point response reaches the value $0.8 r_{\mathrm{d}}$ (point C). The intercept of line (BC) with the horizontal axes gives the dead time T_{ad}. The line BC intersects the line $y(t) = y^{\mathrm{sp}}$ and provides the response time T_{ar}. Furthermore, from Fig. 1(a), we can see that the following equality holds: $T_{\mathrm{ar}} + T_{\mathrm{ad}} = t_{0.8} + t_{0.2}$. Note, however, the rise time t_{r} and the apparent rise time T_{ar} are two different concepts.

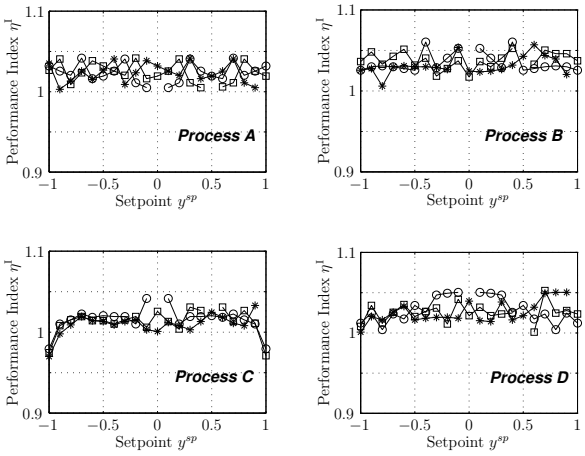


Fig. 2. Characteristics of proposed performance index under optimal control: curve with ($\circ$) corresponds to initial condition $u_0 = 0$, $y_0 = 0$; curve with ($\square$) corresponds to initial condition $u_0 = 0.5$, $y_0 = 0.5$; curve with ($*$) corresponds to initial condition $u_0 = 1$, $y_0 = 1$.

Simulation results have shown that, under optimal control, the first term ($\eta^{\mathrm{I}} = \mathrm{IAE}^{\mathrm{ref}}/\mathrm{IAE}$) defined in (Eq. 1) is very close to unity 1 (see Fig. 2) [3]. This is true for any initial conditions, final conditions, constraints, or process dynamics. In addition, the second term ($\eta^{\mathrm{II}} = r^{\mathrm{d}}/(r^{\mathrm{d}} + y^{\mathrm{os}})$) in Eq. (1) is equal to 1 as the set-point response under optimal control is overdamped, monotonic, and without steady-state offset. With the rise time and settling time defined above, the third term ($\eta^{\mathrm{III}} = t_{\mathrm{r}}/t_{\mathrm{s}}$) is also unity for optimal

[3] In the simulation, all the processes considered (see Table 1) are normalized with unit gain. In addition, the input values are constrained in $|u(k)| \leq 1$, $\Delta u(k)| \leq 0.2$.

Table 1. Processes considered

Process A	$G(s) = 1 \times e^{-8s}/(232s^3 + 37s^2 + s)$
Process B	$G(s) = 1 \times e^{-3s}/(s^2 + s + 1)$
Process C	$G(s) = 1/(2s + 1)(4s + 1)(6s + 1)$
Process D	$G(s) = 1e^{-0.8s}/(2s + 1)^8$

controllers. Therefore, the performance index η defined in Eq. (1) will be close to unity for optimal controllers w.r.t. any operating conditions.

3.2 Drawbacks

However, some drawbacks exist for the proposed performance index η (Eq. 1). For simplicity, consider a hypothetical process which provides two set-point response curves in Fig. 1(b) corresponding to two different controllers C_{I} and C_{II}. For both controllers, the performance indices are close to unity, which means both controllers are almost optimal. However, this conclusion is not true because the closed-loop process with controller C_{I} responds faster than that with controller C_{II}.

To accommodate this drawback, it is desirable to utilize the minimum settling time and minimum IAE as the reference settling time and reference IAE in the definition of performance index η rather than their approximation estimated from the closed-loop set-point response (Eq. 2). Certainly, this requires the availability of a process model. This issue will be discussed in the next section.

3.3 Performance Assessment Framework

We propose our controller performance assessment framework for constrained systems as follows:

Step 1: Estimate the steady state output deviation from the desired reference set-point. If this offset is beyond certain tolerance, either the controller needs improvement or there exists operability issue. Else, go to the next step.

Step 2: From closed-loop set-point response, estimate $t_{0.2}$ and $t_{0.8}$. Then calculate the performance index η. No model is necessary at this stage. If $\eta \ll 1$, it means the current controller does not perform well. On the other hand, if $\eta \simeq 1$, it does not necessarily mean that the current controller is almost optimal (see Section 3.2). In this case, we do need to move to a third step for further evaluation.

Step 3: In this step, different from step 2, minimum settling time and minimum IAE are calculated and employed as the reference settling time and reference IAE in the estimation of the performance index η. If the newly calculated performance index η is also close to 1, it confirms that the controller under evaluation performs well. However, the calculation of the above minimum settling time and minimum IAE demands the knowledge of an approximate process model, which is discussed in the next section.

3.4 Estimation of Minimum Settling Time and Minimum IAE

We assume that the process can be approximated as a first order plus time delay model: $G(s) = K_{\mathrm{p}}e^{-\mathrm{d}s}/(\tau s + 1)$. The parameters can be easily estimated from closed-loop data using strategies such as linear regression.

The advantage of this approximation lies in the fact that the minimum settling time of a system with delay is equal to the time delay plus the minimum settling time corresponding to the delay free system. In addition, an analytical solution of optimal settling time exists for a first order system if there are magnitude constraints on manipulated variables (Uzturk and Georgakis, 2002). If, on the other hand, there are constraints on the rate of change of the manipulated variables as well, then we need to estimate the minimum settling time numerically.

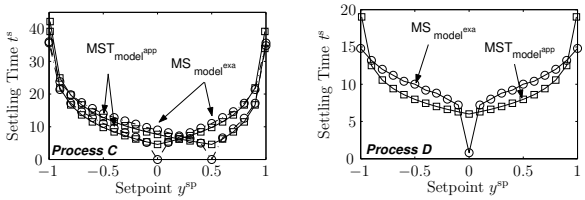


Fig. 3. Comparison of minimum settling time (MST) obtained via approximate model and exact model

Simulation results (see Fig. 3) have shown that the minimum settling time obtained from approximate first order plus time delay models is quite close to that obtained from the exact models. Processes considered are process C and D in Table 1. This is true for different initial conditions, set-point changes, and for different process dynamics. The same results also hold for the estimation of the minimum IAE, calculated in "open-loop". However, this approximation is not very satisfactory if the operating conditions are far from the initial operating point. To handle this inefficiency, one could use more accurate models, for example, a second order plus time delay model.

4. EXAMPLES

In this section, we will use two examples to illustrate the proposed method.

4.1 Example I

For simplicity, consider the integral process ($G(s) = 0.104 \times e^{-3s}/s$). PI controllers with different tunings (Ziegler-Nichols (ZN) and Tyreus-Luyben (TL) tunings), both with and without anti-reset windup (ARWU) schemes, are selected for comparison (see Table 2). τ_t in Table (2) is the tracking time constant (Ogunaike and Ray, 1994, p. 585). Only magnitude constraints on manipulated variables are considered: $|u| \leq 1$.

Table 2. PI tunings for example I

Tuning	K_c	τ_I	τ_t
TL	1.58	26.4	3.5
ZN	2.30	10.0	1.7

Closed-loop set-point step responses corresponding to different controllers are shown in Fig. (4).

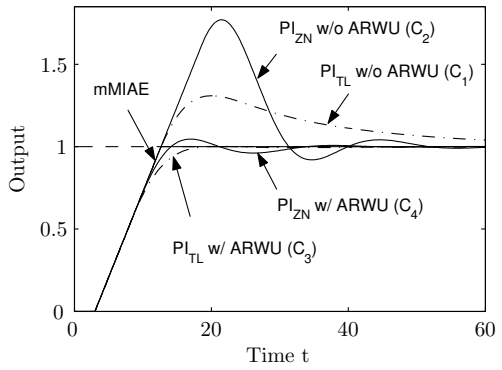


Fig. 4. Comparison of closed-loop response under different controllers

Set-point responses show that controller C_2 performs the worst because it gives the largest overshoot and longest settling time. Controller C_1 is a bit better as it has smaller overshoot. However, both controllers have comparable performances because they have much the same IAE and settling time values. Controllers C_3 and C_4 have performances very close to that of the modified minimum IAE (mMIAE) controller. However, controller C_3 is a bit better than controller C_4 as it has fewer oscillations and shorter settling time.

Based on the above qualitative analysis of the controller responses, we expect that $\eta_{C_2} <\approx \eta_{C_1} \ll 1$ and $\eta_{C_4} < \eta_{C_3} \simeq 1$. The performance index η given in Table (3), which is calculated without a process model, agrees very well with this expectation.

We can easily determine the poor performance of both controllers C_2 and C_1 with the index estimated without the knowledge of a process model. As performance index η is very close to 1 for controllers C_3 and C_4, further evaluation is necessary. The minimum settling time and minimum IAE are

Table 3. Results and Comparison

	C_1	C_2	C_3	C_4
η w/o model	0.48	0.43	0.97	0.77
η w model	0.48	0.43	0.89	0.76

employed as the reference values in the calculation of performance index η. For simplicity, minimum settling time and minimum IAE are determined using the exact model in this example. The newly calculated performance index, given in Table (3), confirms that both controllers have very good performances. Controller C_3 has higher index than C_4 because it allows shorter settling time.

4.2 Example II

In this example, we consider the third order process (process C given in Table 1). The following unconstrained PI tunings are chosen for comparison purpose.

Table 4. PI tunings for example II

	PI_1	PI_2	PI_3
K_c	0.99	2.48	0.25
τ_I	9.54	8.03	5.52

Closed-loop set-point responses under different PI tunings are shown in Fig. (5). The constraints on manipulated variables considered in this example are $|u| \leq 1$, $|\triangle u| \leq 0.2$. Fig. 6(a) and (b) present the performance index calculated without and with an approximate model (1st order plus delay), respectively.

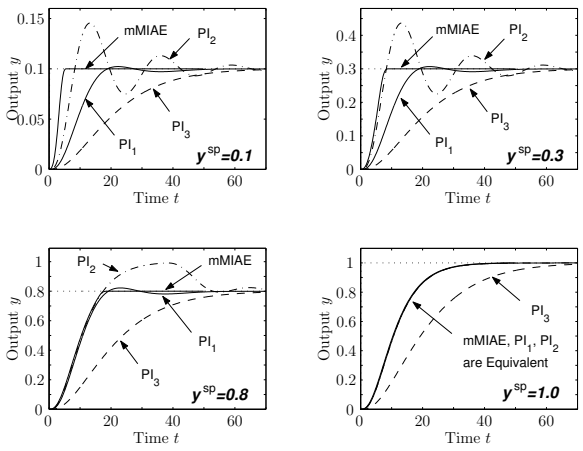
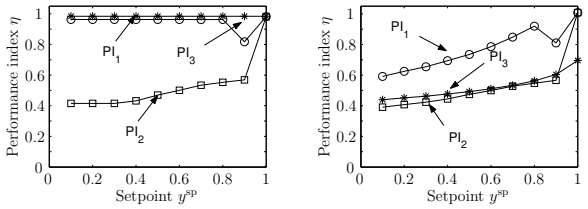


Fig. 5. Comparison of closed-loop set-point responses under different controllers

Fig. 6(a) indicates that the controller PI_2 is inadequate for $y^{sp} \leq 0.9$ as its performance index is very small. This is in accordance with the fact that PI_2 results in significantly oscillating set-point response (Fig. 5) with very large overshoot and long settling time.



(a) w/o model (b) w/ approx. model

Fig. 6. Comparison of performance index η of three different controllers (see Table 4)

Fig. 6(b) confirms that the controller PI_1, which has high performance index value in Fig. 6(a), is an acceptable one. This can be explained by the rather satisfactory set-point responses (Fig. 5) achieved by controller PI_1. On the other hand, Fig. 6(b) tells that controller PI_3 has very poor performance, although it leads to comparably high performance index as controller PI_1 given in Fig. 6(a). This is in agreement with the very sluggish set-point responses (Fig. 5) caused by PI_3. Moreover, Fig. 6(b) shows and verifies that, for maximum available set-point change, controllers PI_1 and PI_2 perform similarly as the optimal controller. We see from Fig. (5) that the set-point responses realized with controllers PI_1 and PI_2 coincide with that under optimal control for $y^{sp} = 1$.

This example clearly illustrates the dependence of controller's performance upon the set-point changes. By comparison with the optimal controller, constrained PI controller's performance turns to be better with the increase of set-point changes. This is due to the fact that more of the available forcing power of the manipulated variables are being exploited. Secondly, the performance index does not change much with set-point magnitudes when the set-point changes are relatively small. This is because the constraints are almost inactive in such cases, and, therefore, the set-point responses are similar.

5. CONCLUSIONS

In this paper, we introduced a new performance index for constrained controller performance assessment w.r.t. deterministic disturbances. It involves integral absolute error (IAE), overshoot, and response time.

A three-step framework is proposed. Steady-state offset is the concern of the first step. In the second step, the performance index is calculated directly from the closed-loop set-point response. No process model is required. If the performance index η is far less than 1, it indicates the controller has poor performance. However, if the index η is close to one, minimum settling time and minimum IAE benchmarks are necessary for further evaluation. In such cases, an approximate process model is required. We have shown that a simple model such as first order plus time delay provides adequate estimate for the needed minimum IAE and minimum settling time.

Remark: Current research is focused on set-point step responses only. We need and will deal with disturbance response in the future.

REFERENCES

Åström, K. J. (1970). *Introduction to Stochastic Control Theory*. Mathematics in Science and Engineering. Academic Press. NY 10003.

Boyd, S. P. and C. H. Barratt (1991). *Linear Controller Design: Limits of Performance*. Prentice Hall information and system sciences series. Prentice-Hall. NJ 07632.

Desborough, L. and T. J. Harris (1992). Performance assessment measures for univariate feedback control. *Can. J. Chem. Eng.* **70**, 1186–1197.

Eriksson, P. G. and A. J. Isaksson (1994). Some aspects of control loop performance monitoring. In: *IEEE Conference on Control Applications*. Vol. 2. Glasgow, Scotland, UK. pp. 1029–1034.

Harris, T. J. (1989). Assessment of control loop performance. *Can. J. Chem. Eng.* **67**, 856–861.

Harris, T. J., F. Boudreau and J. F. MacGregor (1996). Performance assessment of multivariable feedback controllers. *Automatica* **32**(11), 1505–1518.

Huang, B., S. L. Shah and E. K. Kwok (1997). Good, bad or optimal? performance assessment of multivariable process. *Automatica* **33**(6), 1175–1183.

Ko, B. S. and T. F. Edgar (2001). Performance assessment of constrained model predictive control systems. *AIChE J.* **47**(6), 1363–1371.

MacGregor, J., T. J. Harris and Wright J. (1984). Duality between the control of processes subject to randomly occurring deterministic disturbances and arima stochastic disturbances. *Technometrics* **26**(4), 389–397.

Ogunaike, B. A. and W. H. Ray (1994). *Process Dynamics, Modeling, and Control*. Topics in chemical engineering. Oxford university press. New York, Oxford.

Uzturk, D. and G. Georgakis (2002). Inherent dynamic operability of processes: General definitions and analysis of siso cases. *Ind. & Eng. Chem. Res.* **41**(3), 421–432.

PERFORMANCE ENVELOPES OF PROCESS INTENSIFIED SYSTEMS

Syamsul Rizal Abd Shukor, Ming T. Tham

School of Chemical Engineering and Advanced Materials, Merz Court,
University of Newcastle upon Tyne, NE1 7RU, United Kingdom

Abstract: Intensified processes may have response times orders of magnitudes faster than conventional units. Thus, the dynamics of control loop elements such as valves and measurement devices may no longer be negligible. This paper presents the results of an investigation into how the dynamics of control loop components influence the performances of controlled intensified processes. By adopting a particular controller design methodology, it was found that only the process delay and the time-constant of the feedback transmitter affect closed-loop performances. Further analysis showed that there are threshold values for these two parameters, beyond which closed-loop behaviour can be severely degraded, even in the nominal case. In terms of operability, the degree to which a process is intensified may therefore be limited. The results also reveal that advanced control techniques may be necessary if acceptable control of intensified systems is to be achieved.

Keywords: process intensification, control schemes, dynamics, control equipment, control system analysis

1. INTRODUCTION

Advances in Process Engineering have led to numerous new findings and technologies that concentrate on minimising the sizes of unit operations as well as improving the overall speed of production whilst maintaining the throughput of the processes. This concept of "process intensification" was pioneered by ICI in the late 1970's and has since developed rapidly, particularly over the last decade.

Process intensification (PI) is loosely described as a strategy that aims to achieve dramatic reductions in plant volume whilst maintaining production objectives (Ramshaw, 1999). This radical approach to process design is gaining momentum because of drivers that include improved intrinsic safety, increased energy efficiency, reduced plant fabrication costs and easier scale up (Ramshaw, 1983; Jachuck, *et al.*, 1997; Fell, 1998; Ramshaw, 1999; Stankiewicz, and Moulijn, 2000).

Currently, work on advancing PI technologies seems to focus mainly on proving the feasibility of concepts and ideas, as well as attempting to establish key design parameters of various process units. To the best of our knowledge, little investigations have been carried out to study the operation and control of intensified process units.

Minimising the sizes of unit operations inevitably means that process residence times will be much less than conventional sized units. To realise the perceived benefits of PI technology, it is essential that intensified units are coupled with process monitoring and control systems that can cope with the very fast response times so that regulation of environmental variables, product quality, and operational safety can be ensured.

2. STATEMENT OF THE PROBLEM

Reducing the physical sizes of process units whilst maintaining the same throughput means that these units will have shorter residence times, i.e. the dynamics of the systems will be much faster than those encountered in conventional scale units. Adopting the PI design philosophy could lead to an

order of magnitude change in equipment capacity, and would probably bring the response times of intensified systems down to milliseconds rather than the more usual tens of minutes encountered in conventional units. Under such circumstances, current instrumentation may be too slow for intensified processes to be controlled by conventional feedback strategies. Measurement delays that may be tolerable in conventional units may be too large and unacceptable for intensified systems, making the control problem more difficult. Hence, fast responding process sensors are needed in order to achieve automatic feedback control. As the philosophy of process intensification is to reduce equipment sizes without compromising on throughputs, actuators of the same size as those employed in conventional units will continue to be utilised. Thus, actuator dynamics could present problems, as they could be orders of magnitude slower than those of the manufacturing unit. Furthermore, interactions between process states and between process units are also aspects that could lead to further difficulties. There are therefore many factors to be considered in realising automatic control of intensified systems.

Components that make up the control loop must be dynamically compatible with the controlled process for acceptable closed-loop performance. This issue has been largely neglected when designing controllers for conventional process systems, since the time-constants of such processes are significantly larger than those of associated actuators and instrumentation. Nonetheless, given a particular strategy, a deep appreciation of the influences of each component of an intensified system control loop is crucial before good control performances can be assured. Apart from time-delay to time-constant ratios, there also appears to be no published accounts of measures that will indicate when a particular mix of dynamic characteristics could lead to unacceptable closed-loop performances.

This paper presents the results of the preliminary stage of a programme of studies into the control and operation of intensified units. One of the objectives is to attempt to define the "performance envelopes" of fast responding processes under different sensor and actuator conditions. If this can be achieved, then guidelines for the selection of control loop components and control strategies for process intensified systems could be extracted. Knowledge of performance limitations can also be used to determine the extent to which a process should be intensified or miniaturised. The work reported here investigates the effects of instrumentation and actuator dynamics on conventional closed-loop performances.

The paper, which focuses on SISO systems, is structured as follows. First, a controller is designed via the "synthesis equation" method. Note that it is not the aim of this contribution to explore new controller designs. Rather, the synthesis equation method was adopted because it is intuitive, but more importantly, the methodology reveals explicitly, the parametric contributions of the loop components on closed-loop behaviour. The Integral of Absolute Error (IAE) between set-point and controlled output is then analytically defined for the corresponding closed-loop system. For simplicity, all components are taken to be linear. Results are then presented, followed by discussions and conclusions.

3. CONTROLLER DESIGN

Consider the closed-loop system shown in Figure 1.

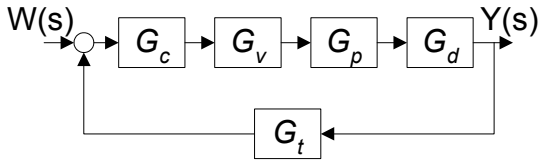


Fig. 1. Closed-loop System

G_c, G_v, G_p, G_d and G_t represent the controller, valve, process, process delay and the feedback transmitter respectively. For simplicity, these components of the closed-loop are assumed linear and G_v, G_p, G_t and G_d have the following forms:

$$G_v = \frac{k_v}{\tau_v s + 1},\ G_p = \frac{k_p}{\tau_p s + 1},\ G_t = \frac{k_t}{\tau_t s + 1}, \text{ and}$$

$$G_d = e^{-\theta s}$$

A standardised controller design methodology is used to develop a general form of the controller. The approach adopted here is based on the Synthesis Equation. It uses the closed-loop expression to determine the controller that will yield a specified closed-loop response. A reason for adopting this approach is to ensure that controllers used under different scenarios are designed from the same basis.

The closed-loop transfer function of the above system is

$$\frac{Y}{W} = \frac{G_c.G_v.G_p.G_d}{1 + G_c.G_v.G_p.G_d.G_t} \quad (1)$$

Rearrangement yields

$$G_c = \frac{1}{G_v.G_p.G_d} \cdot \frac{\frac{Y}{W}}{\left(1 - G_t \frac{Y}{W}\right)} \quad (2)$$

The control objective is to have the closed-loop behave according to:

$$\frac{Y}{W} = \frac{e^{-\theta s}}{\lambda s + 1} \quad (3)$$

It can be seen that the controller comprises the inverses of the valve and process models, and, λ, which is the user specified closed-loop time-constant.

Without loss of generality, the gains of the process, the valve and the transmitter, k_p, k_v and k_t, can be set equal to 1. Substitution of equation (3) into equation (2) and approximating the delay as $G_d \approx 1\text{-}\theta s$, yields

$$G_c = \frac{(\tau_v s+1)(\tau_p s+1)(\tau_t s+1)}{\lambda\tau_t s^2 + (\lambda+\tau_t+\theta)s} \qquad (4)$$

Expanding equation (4) and re-arrangement produces a controller of the form:

$$G_c = G_f.K_c\left(T_{d^2}s^2 + T_d s + \frac{1}{T_i s} + 1\right) \qquad (5)$$

where $G_f = \dfrac{1}{\left(\dfrac{\lambda\tau_t}{\lambda+\tau_t+\theta}s+1\right)}$, $K_c = \dfrac{\tau_v+\tau_p+\tau_t}{(\lambda+\tau_t+\theta)}$,

$T_{d^2} = \dfrac{\tau_v\tau_p\tau_t}{\tau_v+\tau_p+\tau_t}$ $T_d = \dfrac{\tau_v\tau_p+\tau_v\tau_t+\tau_p\tau_t}{\tau_v+\tau_p+\tau_t}$ and

$T_i = \tau_v+\tau_p+\tau_t$

It can be seen that the controller settings take into account actuator and transmitter dynamics.

Notice too that when transmitter dynamics are negligible, the controller reduces to the normal Proportional-Integral-Derivative (PID) form. Otherwise, the design yields an algorithm equivalent to a Proportional-Integral-Derivative-Derivative (PIDD) controller in series with a first-order filter. This leads to an interesting interpretation. When the dynamics of the transmitter are significant, then further anticipatory action is required, leading to the second-order derivative term. In this case, a low-pass filter would be required to mitigate the effects of high order derivative action.

4. SIMULATION WORK

Evaluation of the performances of the controller under different transmitter and valve dynamics were initially carried out via numerical simulation, using SIMULINK™. The IAE between the set-point and controlled output was used as the measure of control performances. The results were very inconsistent however, especially when the time-constants of the valve and/or transmitter were significantly different from that of the process. With hindsight, this should have been expected; under these conditions, the set of equations describing the system becomes "stiff". Therefore, performance investigation of the above closed-loop system was eventually carried out analytically, with the help of the Symbolic toolbox in the MATLAB™ environment, thus avoiding numerical problems.

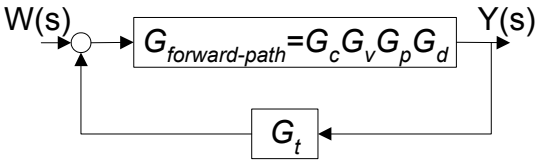


Fig. 2. Simplified diagram of the closed-loop system

From Figure 2, the error response transfer function was established for a unit step change in the input, *W(s)* as:

$$error(s) = \frac{W(s)}{1+G_{forward\text{-}path}.G_t} \qquad (6)$$

where

$$G_{forward\text{-}path} = \frac{(\tau_t s+1)(1-\theta s)}{\lambda\tau_t s^2 + (\lambda+\tau_t+\theta)s} \qquad (7)$$

Equation (7) shows that the specified closed-loop time-constant, the transmitter time-constant and the process delay are the only parameters that have significant influence on overall control performance. It follows from equation (4) that the controller cancels the poles of G_v and G_p, resulting in equation (7) above. In other words, the effects of process and the valve time-constants on closed-loop behaviour have been removed as a result of the controller design adopted.

Substituting equation (7), G_t and $W(s) = \dfrac{1}{s}$ into equation (6) yields

$$error(s) = \frac{\lambda\tau_t s + (\tau_t+\lambda+\theta)}{(\lambda s+1)(\tau_t s+1)} \qquad (8)$$

The IAE can then be determined by integrating the absolute value of the inverse Laplace transform of equation (8), from time 0 to time *t*, which yields

$$\text{IAE} = \left|\frac{\lambda(\lambda+\theta)}{\lambda-\tau_t}\left(1-\exp\left(-\frac{t}{\lambda}\right)\right) - \frac{\tau_t(\tau_t+\theta)}{\lambda-\tau_t}\left(1-\exp\left(-\frac{t}{\tau_t}\right)\right)\right| \qquad (9)$$

5. RESULTS AND DISCUSSION

To study the effects of λ, τ_t and θ on closed-loop performances, different τ_t and θ values were considered for three desired closed-loop responses (λ = 0.5, 1.0 and 1.5). In each case, the closed-loop had to track a unit step change in set-point, and the IAE was determined using equation (9).

The performance evaluation results are summarised in the 3-D plot shown in Figure 3 below, where $1/\tau_t$ and $1/\theta$ were used to identify the "performance envelope" for the system under investigation.

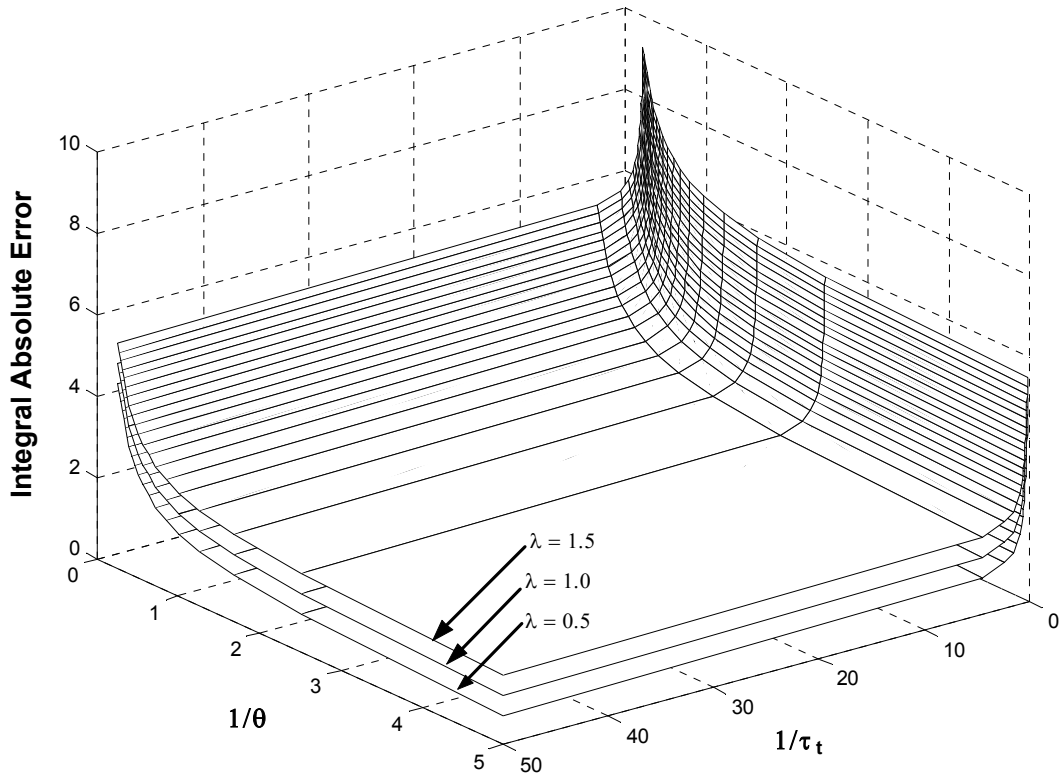


Fig. 3. IAE values for different $1/\tau_t$ and different $1/\theta$ at $\lambda = 0.5$, 1.0 and 1.5

As expected, smaller values of the closed-loop time-constant, λ, produce better overall control performances as indicated by the lower IAE values.

A particular consequence of using the controller given by equation (5) is that closed-loop performance is independent of actuator dynamics and the dynamics of the controlled process (see equation (8)). This is because the controller cancels the forward path dynamics and, as a result, the response speeds of the closed-loop components (relative to the time-constant of the process) have no bearing on performance.

The presence of process time-delays will however degrade control performance, and this can be clearly seen from the IAE values in Figure 3. In the case of this example, overall controller performance degrades rapidly when θ becomes greater than 1. Recall that the approximation $G_d \approx 1\text{-}\theta s$ was used in the formulating the controller, so that the resulting structure bears some resemblance to the familiar PID controller. If this requirement is relaxed, it is possible that the detrimental effects of time-delays will not be as marked. There is definitely scope for improving control performances, in particular, by incorporating time-delay compensation into the control system. Nevertheless, it does point to the fact that advanced control algorithms would be necessary.

The solution to the problem of large transmitter response times is not so straightforward, however. Typically, transmitter time-constants range between 0.2s - 2.0s. Figure 3 shows that, for any particular value of θ, control performance may deteriorate severely even if the fastest transmitter is used. If the controller design outlined in this paper is employed, the problem of slow dynamics in the feedback path will be compounded if the measurement device has an associated delay. Then, effective control may only be achieved through the use of "soft-sensors" (e.g. Tham, *et al.*, 1989). Again, advanced techniques may be necessary.

Finally, it is also interesting to observe that for $1/\tau_t > 5$ and $1/\theta > 1$, control system performances were almost identical. This indicates that the controller is relatively insensitive to variations in transmitter time-constant and process delay, provided $\tau_t < 0.2$ and $\theta < 1$. Beyond these thresholds however, overall control performance deteriorated exponentially, as $1/\tau_t$ and $1/\theta$ become smaller, i.e. when the time-constant of the transmitter and the time-delay become larger.

6. CONCLUSION

The results presented here provide further insights into how the dynamics of control loop components influence closed-loop performance, with particular reference to intensified systems. The small volumes and large throughputs of such systems lead inescapably to smaller residence times. In some cases, their response times are of the same order of magnitude as those of actuators and measurement transmitters, perhaps even faster. At first, it was thought that this could potentially cause control problems.

However, with the example system studied, it was found that the process delay and transmitter dynamics were the only parameters that influence control performance. This is because the Synthesis Equation design methodology yields controllers that cancel the dynamics of both the process and the actuator. Since forward path dynamics no longer feature, the result applies not only to intensified systems, but also to all closed-loop systems with controllers designed via the same approach. This dispels, to a certain extent, initial reservations about the ability of existing actuators to cope with the demands of controlling intensified systems.

As for the influence of process delay and time-constant of the transmitter, the study identified threshold values beyond which closed-loop performances were severely affected. Therefore, with regard to operability, there are limits to the degree of intensification, especially if high throughputs are to be maintained.

One particular difficulty was experienced while performing this study. Due to the stiffness of the resulting set of system equations, brought about by loop components with very different time-constants, numerical simulation results were very inconsistent. Therefore, it is suggested that studying the dynamic behaviour of intensified systems be best done analytically, aided by symbolic mathematical tools.

The work presented in this paper assumes no process-model mismatch and the results were obtained using a non-conventional PIDD type controller. It would be interesting to assess the robustness of the strategy when there are uncertainties associated with the parameters of the loop components used for controller design; and when the controller is constrained to have the standard PID structure. These are the subjects of current work.

ACKNOWLEGEMENTS

The financial support provided by Universiti Sains Malaysia granted to Syamsul Rizal Abd Shukor is gratefully acknowledged.

REFERENCES

Fell, N. (1998). Innovation Offers A New Spin On Drug Production. *Chemical Engineer,* 23-25.

Jachuck, R. J., C. Ramshaw, K. V. K. Boodhoo, and J. C. Dagleish. (1997). Process Intensification: The Oppurtunity Presented By Spinning Disc Reactor Technology. *Institution of Chemical Engineers Symposium Series*. (141), 417-424.

Ramshaw, C. (1983). HIGEE - An Example of Process Intensification. *Chemical Engineer,* 13.

Ramshaw, C. (1999). Process Intensification And Green Chemistry. *Green Chemistry,* 15-17.

Stankiewicz, A. I. and J. A. Moulijn (2000). Process Intensification: Transforming Chemical Engineering. *Chemical Engineering Progress,* 22-34.

Tham, M. T., A. J. Morris, and G. A. Montague (1989). Soft-Sensing: A Solution To The Problem of Measurement Delays. *Chemical Engineering Research & Design,* **67** (6), 547-554.

FAULT DIAGNOSIS BASED ON LIMIT MEASUREMENTS OF PROCESS VARIABLES.

Heinz A Preisig [*], **Yun Xia Xi** [**] and **Khiang Wee Lim** [**]

[*] *Dept of Chemical Engineering, Norwegian University of Science and Technology (NTNU), N-7491 Trondheim, Norway.*
Heinz.Preisig@chemeng.ntnu.no
[**] *National University of Singapore, Singapore.*
elelimkw@nus.edu.sg

Abstract: Industry uses mostly information on measurements passing limits as inputs to diagnostic systems. Asking the question on what can be achieved by such systems, we aim at an optimal design of diagnostic systems. The approach is, in contrast to the currently available techniques, based on continuous models that are mapped into discrete-event dynamic systems in the form of nondeterministic automata, with faults being constraint to have event-dynamic, that is, they occur and persist. We compute domains in which faults can be isolated or detected and discuss guidelines on how to design an application-optimal fault detection and isolation mechanisms.

Keywords: Fault analysis, fault detection, fault isolation, reliability, safety

1. THE PROBLEM BEING STUDIED

Industrial diagnostic systems use as input mostly measurements indicating passing of limits placed on process variables, such as temperature has exceeded value x or pressure is lower than value y. These measurements indicate the limit that is crossed and also the direction in which the process did cross the limit. Currently industry uses mostly diagnostic systems that are built carefully based on process knowledge and experience of the operators. These systems are invariably quite complex and far from easy to build or maintain. The past has seen a number of efforts to improve the situation, which reflects into a rich literature. A wide selection of different methods are reviewed in Gertler (1988), Frank (1990), Patton et al. (1989), Pouliezos and Stavrakakis (1994), Isermann (1997). One of the most common techniques is based on fault tree analysis. Today, it is almost a traditional technique and its development has a rich history Lapp and Powers (1977), Ulerich and Powers (1988), Vries de (1990). With the evolvement of on-line filtering techniques and their extension to parameter estimation, their mostly high sensitivity of the parameters to faults has been utilised for the construction of diagnostic system Isermann (1984), Isermann (1993), Chow and Willsky (1984). The dawn of discrete-event dynamic added over time another viewpoint that yielded alternative design methods, of which examples are reported in Lin (1994), Bavishi and Chong (1994), Sampath et al. (1995), Sampath et al. (1996), Cassandras and Lafortune (1999). Knowledge-based systems are also very popular as they provide a systematic method to capture people's knowledge about the process, experimental or theoretical, into a easy-to-program structure and neural nets provided a matching modelling technique that did not have to rely on mechanistic process knowledge, which was considered too expensive to develop Hoskins and Himmelblau (1988), Venkatasubramanian and Chan (1989), Venkatasubramanian et al. (1990), Maki and Loparo (1997).

Several years ago, we took an alternative route, which grew out of two efforts: one on computer-aided modelling that provides a systematic and easy method to construct mechanistic

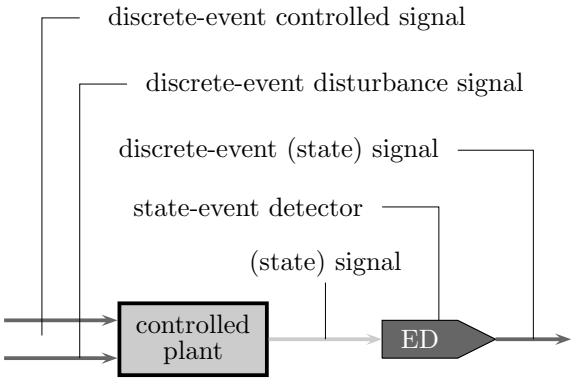


Fig. 1. *The generic plant set-up with a continuous plant, a state-event detection mechanism providing the primitive measures and two discrete-event signal inputs one controlled and the other one not.*

process models, at least to the extent possible, and second the need for safe supervisory control, which was a project that aimed at the design of supervisory systems that are guaranteed complete, meaning including all aspects of the process behaviour on the supervisory level. Quite obviously, mainly the latter induced the question on how to handle faults in such systems, as supervisory control systems are directly linked to diagnostic systems for handling shut-down or recovery. Consequently we posed the question on what can be achieved having only available such limited information as limit crossing and the direction of the crossing; and: can we find design methods for diagnostic systems, a problem that turned out to be at least equally challenging than designing supervisory control systems Philips (2001).

Figure (1) shows the context of plants that we consider, namely a continuous plant, which is affected by fast-switching unobserved disturbances and commands that change the parameters of the controllers in the plant and their setpoints or ask to switch flows or the like.The scheme assumes that the plant is operating in continuous time and that it provides information about the state. Both may raise some questions as the plant's signals are usually sampled and it is not the state that is available, but an estimate of the state being reconstructed using an observer. However, we take the view of a larger time scale in which, for all practical purposes, the plant and its associated measurement and low-level control systems provide essentially continuous state information. The obvious conditions on the relative sampling rate and the delays caused by filtering and reconstruction must be satisfied.

2. WHERE CAN WE DETECT WHAT

A diagnostic system should provide information on faults. Obviously one would like to detect these faults whenever they occur. But then, it is also obvious that this will not be possible for various reasons but mostly because of accuracy of measurements and accuracy of knowing the systems behaviour under different operational modes (normal or various different faults). We shall certainly come back to some of these questions later in this exposition. But first, we shall focus on the ideal world, just to learn on what would be possible and analyse a process model. When we aimed at the design of supervisory control systems, we took a similar approach and derived a method to map continuous plants that are, on the high level, controlled by commands and which generate as output limit crossing signals generated by event detectors. Thus we describe a hybrid system, but choose to map the continuous part controlled by commands and generating discrete event signals as a discrete-event dynamic system Preisig et al. (1997),Philips (2001). In its core, the method maps the continuous state space of the plant into a discrete equivalent, which is a space of hypercubes defined by the limits, which, in turn, define the event detectors operation.

Mathematically, let us choose the plant as being described by:

$$\begin{aligned}\dot{\underline{\mathbf{x}}} &:= \underline{\mathbf{f}}(t, \underline{\mathbf{x}}, \underline{\mathbf{u}}_c, \underline{\mathbf{d}}_c) \\ &\underline{\mathbf{x}} \in \mathbb{R}^n, \quad \underline{\mathbf{d}}_c \in \mathbb{R}^m, \quad \underline{\mathbf{u}}_c \in \mathbb{R}^p\,. \end{aligned} \tag{1}$$

with the two vector signals $\underline{\mathbf{u}}_c$ and $\underline{\mathbf{d}}_c$ being piecewise constant signals for the command input and the persistent fault signals and $\underline{\mathbf{f}}(t, \underline{\mathbf{x}}, \underline{\mathbf{u}}_c, \underline{\mathbf{d}}_c)$ a vector of analytical functions. The plant is observed by a set of event detectors attached to the individual state measurements, or reconstruction in the case they are not directly attachable, each defined by a set of limits:

$$\mathcal{B}_i := \left\{\beta_i^1, \ldots, \beta_i^{b_i}, \ldots \beta_i^{n_i}\right\}$$

then the continuous state space is mapped into a discrete space of hypercubes representing the discrete space:

$$\mathcal{H}(x) := \left\{[x_i]_{\forall i} \mid \beta_i^{b_i - 1} \leq x_i \leq \beta_i^{b_i}\right\}$$

An event is, in this context, defined as a crossing of a face of a hypercube, that is, a boundary is crossed. The dynamics of the discretely-controlled and discretely-observed plant is a non-deterministic automaton Preisig et al. (1997), Philips (2001). The crossing of the boundary may only occur if there exists at least one point on the face having a gradient that points across the boundary. Requesting a reasonable kind behaviour of the system, that is in the simplest case, analytical functions describing the dynamics, the directionality of the gradient in a particular direction changes on a hypersurface defined by the particular component being zero. This hyper surface is defined by the expression:

$$\dot{x}_i := f_i(t, \underline{\mathbf{x}}, \underline{\mathbf{u}}^k, \underline{\mathbf{d}}^l) := 0\,. \tag{2}$$

Assuming, which is not very limiting, that these hypersurfaces are reasonably kind, preferably monotone, one sees quickly that they split the state space into two parts, one in which all boundaries defined for this state variable are crossed in positive direction, and one in which it is crossed in negative direction, which are the measurements we have available.

3. DIAGNOSABILITY

With this result, we know where a particular measurement, the signal indicating limit crossing and its direction, gives us information about the dynamics of the process. Diagnosability tries to distinguish between different plant behaviours, which above we termed modes of operation. The task now is to find the piece of the state space in which such rudimentary process information yields information about the mode of operation. Mathematically spoken we seek a subspace in which the behaviour of the plant operating under *mode a* behaves differently from the plant operating under *mode b* such that it is uniquely observable with the given measurement. All combinations of inputs are defined as instances of the discrete command and disturbance vectors. For the command (control) vector the running index $k \in \mathcal{K} \subset \mathbb{N}^+$ is used and for the disturbances it is the letter $l \in \mathcal{L} \subset \mathbb{N}^+$ both shown as superscript, in distinction to the vector component index which is shown as subscript. Let $\mathcal{N}_i$ be the index set of the states being coupled with state i and $\mathcal{L}_i$ the index set of the disturbances being coupled with state i we first define the space:

$$\mathcal{X}_{i,s}(k,l) := \left\{ \underline{\mathbf{x}} \in \mathcal{V} \mid s := \text{sign}\left(f_i(t, \underline{\mathbf{x}}, \underline{\mathbf{u}}^k, \underline{\mathbf{d}}^l) \right) \right\} \tag{3}$$

in which the i^{th} component of the gradient assumes the sign s and next

$$\begin{aligned} \mathcal{O}_{i,s}(k, \mathcal{A}_r) := & \left(\bigcap_{\forall j \in \mathcal{A}_r} \mathcal{X}_{i,s}(k,j) \right) \cap \\ & \cap \left(\bigcap_{\forall j \in \{\mathcal{L}_i - \mathcal{A}_r\}} \mathcal{X}_{i,-s}(k,j) \right) \\ & \forall i; \forall r\,. \end{aligned} \tag{4}$$

with

$$\begin{aligned} \mathcal{A} := \{\mathcal{A}_r\} &:= \{\mathcal{T}_0, \mathcal{T}_j\} \cup \mathcal{F}\,, \\ \mathcal{F} := \{\mathcal{F}_r\} &:= \{\{\mathcal{T}_{j_1,j_2}\}, \{\mathcal{T}_{j_1,j_2,j_3}\}, \dots\}\,, \\ \mathcal{T}_{j_1,j_2} &:= \{j_1, j_2 | j_1 \neq j_2; j_1, j_2 \in \mathcal{L}_i\}\,, \\ \mathcal{T}_{j_1,j_2,j_3} &:= \{j_1, j_2, j_3 | j_1 \neq j_2 \neq j_3; j_\cdot \in \mathcal{L}_i\}\,, \\ \dots &:= \dots\,, \\ \mathcal{T}_0 &:= \{0\}\,, \\ \mathcal{T}_j &:= \{j \in \mathcal{L}_i\}\,, \end{aligned}$$

with

$$\mathcal{L}_i := \left\{ j \;\middle|\; \frac{\partial f_i(t, \underline{\mathbf{x}}, \underline{\mathbf{u}}^k, \underline{\mathbf{d}}^l)}{\partial {d_j}^l} \neq 0 \right\}. \tag{5}$$

being the set of persistent fault inputs that are coupled with state i. The index set $\mathcal{T}_0$ represents the no-fault case, whilst the sets $\mathcal{T}_j$ each having only one element are for the jthfaults. The other test sets $\mathcal{T}_{\{j_1,\dots\}}$ are for groups of faults. The non-empty subspaces $\mathcal{O}_{i,s}(k, \mathcal{A}_r)$ have the properties that gradient information in the x_i-direction is sufficient information to diagnose the plant for the cases listed in Table 1, which is what we were seeking.

		s detected	-s detected
$\exists$	$\mathcal{O}_{i,s}(k, \mathcal{T}_0) \neq 0$	no fault	$\{j\} \subset \mathcal{L}_i$
$\exists$	$\mathcal{O}_{i,s}(k, \mathcal{T}_j) \neq 0$	fault j	not fault j
$\exists$	$\mathcal{O}_{i,s}(k, \mathcal{F}_r) \neq 0$	$\{j\} \subset \mathcal{F}_r$	$\{j\} \subset \mathcal{L}_i - \mathcal{F}_r$

Table 1. *Different types of overlapping subspaces for the i^{th}component*

Now that tells us on what we can achieve, our main result. How about, though, the design of diagnostic units operating on such measurements?

4. DESIGN ISSUES

If the automaton to be constructed serves the purpose of fault detection & and fault isolation, then the boundaries that make the state-event detector, are to be place into the subspaces $\mathcal{O}_{i,s}(k, \mathcal{A}_r)$ for each component. The computation of the automaton is solved, as the procedure for the computation of the non-deterministic automaton is given (Preisig et al. (1997), Philips (2001)). Remains the question on how precisely to place the boundaries into these subspaces. The question cannot be answered in a deterministic manner, as it is a true design issue. Why? The automaton operates in a square world, with the size of the hypercubes defining a type of resolution for the detection. If we make the resolution high, the detection will operate on this high resolution and if it is low it will be correspondingly operating on the low resolution. It is the designer's choice and must be based on the dynamics of the process and the significance of the fault to be detected. It involves such questions as how quickly should it

be detected and how much of faulty behaviour can be tolerated under various conditions. De facto, the designer gets the information on which part of the state space he is to approximate with the automaton and he is left with the decision on the fidelity of the approximation. The automaton, as it is described in Preisig et al. (1997) and Philips (2001) does not provide any timing information but first results on computing minimal and maximal transition times are now available Preisig et al. (2002), though for monotone plants only or for parts of the state space that exhibit monotone behaviour. Work on non-monotone plants is currently in progress.

There are two additional obvious issues to be mentioned. Firstly, the design of a diagnostic system is based on a process model. Consequently the diagnostic system will not be designed to detect faults that are not modelled, but then this is not the case with any of the diagnostic system; none can operate and act on information it does not have. The second limitation is the fact that the measurement is sensitive to noise. It must be assumed that the detection of the event and consequently the detection of the direction is essentially done with certainty thus with a probability that is near to 1. This raises certainly questions on the implementation, but then effects of noise are always present independent on the behaviour of the diagnostic system. We prefer to separate these question, being aware that we deal with dynamics in a certain time scale, which is limited and which allows us to design appropriate state reconstruction and filters.

Combination of faults can easily be handled as they can simply be defined as additional operating modes. The set-up of the initial description is design to handle this complexity. Further, the knowledgeable reader may point out that the dimensionality of the discrete state space grows combinatorially with the number of state variables and the number of boundaries being defined for the individual state variables. Whilst the basic observation is correct, the reality shows that the systems are almost always very sparse and the at the individual functions (equations 2) are very weekly coupled in the state space and the input spaces indeed. Thus since the automaton is based on these local measurements, it is always locally of small dimension: the two sets $\mathcal{N}_i$ and $\mathcal{L}_i$ are usually rather small. Thus the state explosion argument is not applicable.

5. EXAMPLE

The sample plant consists simply of two tanks standing side by side with a feed of fluid that is driven by a pump into the first tank and a pipe connecting the two tanks at the bottom. Once the pump is running, the plant is obviously not stable as it consists of two coupled, pure integrators.

For a given set of parameters, the subspaces are shown in Figure 5 with the levels (volumes or masses) of the two tanks as the respective co-ordinates.Table 2 and the Table 3 list the

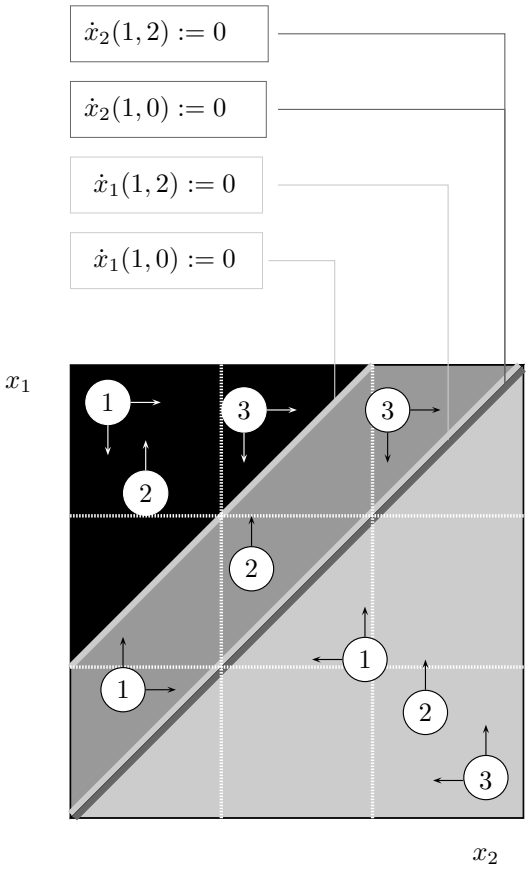


Fig. 2. *Phase diagram of two tank system for case 1-3. (Case 1: pump on, pipe open, pump running; case 2 : pump on, pipe blocked, pump running; case 3: pump on, pipe open, pump failing.)*

component-equilibrium surfaces, here lines, and the different subspaces for the two signs and the two co-ordinates. From the analysis, it is evident that only one co-ordinate (level, volume, mass of tank 1) provides information for diagnoses if one does not use transition time in which case the fault of a blocked pipe could be detected as not generating a transition in the x_2 direction. Otherwise, it is the stripe in the middle in which it is case 3 that shows a negative sign whilst the other two cases would move in positive direction. Moving across a boundary in the negative x_1-direction isolates the fault of the pump not being on.

The second fault that can be isolated using directional information on state component x_1 is detectable in the upper-left triangle in which it is case 2, which is the only one that moves in positive direction for this operation mode,

case	k	l	u_1	d_1	d_2	x_1^0	dyn	x_2^0	dyn
1	1	-	1	0	0	$x_2 + \frac{p_2}{p_1}$	+	x_1	+
2	1	1	1	1	0	$x_1 \rightarrow x_1^+$	−	$x_2^- \leq x_2 \leq x_2^+$	0
3	1	2	1	0	1	x_2	+	x_1	+
4	2	-	0	0	0	x_2	+	x_1	+
5	2	1	0	1	0	$x_1^- \leq x_1 \leq x_1^+$	0	$x_2^- \leq x_2 \leq x_2^+$	0
6	2	2	0	0	1	x_2	+	x_1	+

Table 2. *Component equilibrium surfaces for all cases (+ indicates stable, - unstable and 0 no dynamics for the respective component)*

case	k	l	u_1	d_1	d_2	$\mathcal{X}_{1,-1}(k,l)$	$\mathcal{X}_{1,+1}(k,l)$	$\mathcal{X}_{2,-1}(k,l)$	$\mathcal{X}_{2,+1}(k,l)$
1	1	-	1	0	0	$x_1 > x_2 + 1$	$x_1 < x_2 + 1$	$x_2 > x_1$	$x_2 < x_1$
2	1	1	1	1	0	0	$x_1 > {x_1}^-$	0	0
3	1	2	1	0	1	$x_1 > x_2$	$x_1 < x_2$	$x_2 > x_1$	$x_2 < x_1$
4	2	-	0	0	0	$x_1 > x_2$	$x_1 < x_2$	$x_2 > x_1$	$x_2 < x_1$
5	2	1	0	1	0	0	0	0	0
6	2	2	0	0	1	$x_1 > x_2$	$x_1 < x_2$	$x_2 > x_1$	$x_2 < x_1$

Table 3. *Subspaces for both co-ordinates, each case and both signs*

whilst for the other two operation modes the gradient in this direction is negative. Thus a move across a boundary in this the upper-left triangular subspace in positive x_1 direction indicates the fault "blocked pipe".

6. CONCLUSIONS

Insight gained on modelling the discretely-controlled and discretely-observed continuous plant section of a hybrid system as a non-deterministic automaton gives valuable insight into what can be achieved with simple measurements of state-variable limit crossing and direction of the crossing.

The key is to analyse the flow of the dynamic behaviour in the continuous domain under the different operating modes and seek the parts of the state space in which directional information is sufficient to diagnose faulty behaviours.

The design of diagnostic systems focusing on individual or any combination of faults are only constraint by the ability to model the plant behaviour under any combination of faulty conditions and the accuracy with which the fault is to be detected, both in state space as well as in time.

Since the dynamic equations are sparsely coupled and since we analyse the behaviour component by component, the state dimension remains quite small and the often in this context cited dimension-explosion problem does simply not occur.

References

S Bavishi and K P E Chong. Automated fault diagnosis using a discrete event systems framework. *IEEE Int Symp Intelligent Control*, pages 213–218, 1994.

C G Cassandras and S Lafortune. Discrete event systems: The stae of the art and new directions. *Applied and Computational Control Signals and Circuits*, 1:34–42, 1999.

E Y Chow and A S Willsky. Analytical redundancy and the design of robust faulure detection systems. *IEEE Transaction on Automatic Control*, pages 603–614, 1984.

P M Frank. Fault diagnosis in dynamic systems unsing analytical and knowledge-based redundancy – a survey and some new results. *Automatica*, 26(3):459–474, 1990.

J J Gertler. Survey of model-based failure detection and isolation in complex plants. *IEEE Control System Magazine*, 6(8):3–11, 1988.

J C Hoskins and D M Himmelblau. Artificial neural network models of knowledge representation in chemical engineering. *Computers & Chemical Engineering*, 12:881–890, 1988.

R Isermann. Process fault detection based on modelling and estimation methods – a survey. *Automatica*, 20(4):387–404, 1984.

R Isermann. Fault diagnosis of machines via parameter estimation and knowledge processing. *Automatica*, 29(4):815–835, 1993.

R Isermann. Supervision, fault-detection and fault-diagnosis methods – and introduction. *Control Engineering Practice*, 5(5):639–652, 1997.

S Lapp and G Powers. Computer-aided synthesis of fault trees. *IEEE Transaction Reliability Engineering*, 26:2–13, 1977.

F Lin. Diagnosablility of discrete event systems and its application. *Journal of Discrete Event Dynamic Systems*, 4(2):197–212, 1994.

Y Maki and K A Loparo. A neural network approach to fault detection and diagnosis in

industrial processes. *IEEE Transaction Control System Technology*, 5(6):529–541, 1997.

R Patton, P M Frank, and R Clark. *Fault diagnosis in dynamic systems: theory and applications*. Prentice Hall, Englewood Cliffs, NJ, 1989.

P P H H Philips. *Modelling, Control and Fault Detection of Discretely-Observed Systems*. PhD thesis, TU Eindhoven, Eindhoven, The Netherlands, 2001.

A D Pouliezos and G S Stavrakakis. *Real-time fault monitoring of industrial processes*. Kluwer Academic Publishers, Dordrecht, The Netherlands, 1994.

H A Preisig, K W Lim, and Y X Xi. Computation of min max transition times in automata represting discrete-event observed continous, monotone plants. *ASCC 2003*, (TM 9-23):1658, 2002.

H A Preisig, M J H Pijpers, and M Weiss. A discrete modelling procedure for continuous processes based on state- discretization. *MATHMOD 2*, pages 189–194, 1997.

M Sampath, R Sengupta, S Lafortune, K Sinnamohideen, and D Teneketzis. Diagnosablility of discrete event systems. *IEEE Transaction on Automatic Control*, 40(9):1555–1575, 1995.

M Sampath, R Sengupta, S Lafortune, K Sinnamohideen, and D Teneketzis. Failure diagnosis using discrete-event models. *IEEE Transaction Control System Technology*, 4(2):105–124, 1996.

N Ulerich and G Powers. On-line hazard aversion and fault diagnosis in chemical processes: The digraph + fault-tree method. *IEEE Transaction Reliability Engineering*, 37(2):171–177, 1988.

V Venkatasubramanian and K Chan. A neural network methodology for process fault diagnosis. *AIChE Journal*, 35(‘1):1993–2001, 1989.

V Venkatasubramanian, R Vaidyanathan, and Y Yamamoto. Process fault detection and diagnosis using neural networks. *Computers & Chemical Engineering*, 14(7):699–712, 1990.

R Vries de. An automated methodology for generating a fault tree. *IEEE Transaction Reliability Engineering*, 39:76–86, 1990.

OPTIMAL EXPERIMENTAL DESIGN FOR TRAINING OF A FAULT DETECTION ALGORITHM

Shi Jin Lou, Thomas Duever[1], Hector Budman

Department of Chemical Engineering
University of Waterloo, Waterloo, Ontario, Canada, N2L 2G1

Abstract: This paper focuses on Optimal Experimental Design to train a Projection Pursuit Regression (PPR) model used for fault detection. A novel experimental design method, referred to as Gaussian Probability Design, is proposed and compared with the conventional Factorial Design. The Gaussian Probability Design automatically searches for the sparseness of the data, and adds pairs of training data on both sides of a class boundary in areas where the data density is the lowest. This design method outperforms the Factorial Design in reducing the fault misclassification more effectively with the same amount of new training data.

Keywords: Optimal Experimental Design, fault diagnosis, Projection Pursuit Regression

1. INTRODUCTION

In most model learning problems, the learner has the ability to act on its environment and gather specific experimental data that will minimize the model errors. A special case of a model learning problem is the training of a fault detection algorithm. In most of the literature dealing with fault detection, the learner has been treated as a passive recipient of data and its active role in determining optimal data for training has been ignored. More specifically, the fault detection algorithms were generally trained on specific faults on the assumption that only these faults will occurred in the future. Thus, research work on optimal experimental design (OED) for training of fault detection techniques has not been extensively considered and reported in the literature. This work addresses the generalization problem where future faults may occur that did not occur during training. Therefore, optimal experimental design may be a venue to minimize the misclassification of faults that were not observed during training of the fault detection algorithm.

In this work, the Projection Pursuit Regression (PPR) technique has been chosen as the basis for the design of the fault detection algorithm. PPR is a multivariate statistical technique [1,3,4], ideally suited for nonlinear systems and has been applied to fault detection [2]. Similar to other multivariate methods it is based on the decomposition of the inputs along principal components. However the basis functions referred to as hidden functions are not fixed a priori but determined by the training data and the output calculation is based on a nearest neighbourhood approach applied in the hidden functions space. In a previous work by Lou et al [5], PPR has been found to be a good trade-off as compared to other techniques from the point of view of extrapolation errors due to insufficient training data and noise rejection. This paper deals with a novel method to design optimal experiments for the training of a PPR-based fault detection algorithm. The objective is to design experimental data in some predetermined window of operating conditions to minimize the fault misclassifications during testing. The incentive is to minimize the number of experiments for the sake of economy. Three different sets of data will be considered: training set #1: obtained using the conventional Factorial Design to obtain a first model, training set #2: added based on the knowledge of the first data set, and a testing set, to assess the classification accuracy of the algorithm trained with the two training data sets. The current work presents a methodology to design the second training data set mentioned above. Thus, this study follows a sequential approach where the optimal design of training set #2 is based on a priori knowledge of training set #1.

Cohn (1996) has applied techniques from OED to guide the query selection of a neural network learner [6]. Cohn demonstrated that these techniques allow the learner to minimize the generalization error by minimizing its variance.

[1] Corresponding author. Tel: +1-519-888 4567 ext. 2540; Fax: +1-519-746-4979. E-mail: tduever@cape.uwaterloo.ca

His OED approach is claimed to be applicable to any network architecture whose output is differentiable with respect to its parameters and may be used for both regression and pattern classification problems. However, for PPR models, the derivatives of the output with respect to the parameters are actually the outputs of the hidden functions. Since for PPR models, each class corresponds to the same hidden output value in each hidden function, all data points belonging to the same class will have the same derivative values with respect to the parameters. Thus, Cohn's method will not be useful for distinguishing between better or worse experimental points belonging to the same class.

In this work a method, referred as to Gaussian Probability Design, is proposed for designing the second set of training data mentioned above. In order to test the efficiency of the proposed design, the design of training set #2 will be compared to a conventional factorial design of this set. Finally, the methods are illustrated and compared for a CSTR case study. This case study, previously used [7] for testing fault detection algorithms, considers faults as extreme values in the inlet temperature and concentration conditions. These conditions are inferred from the measurements of reaction temperature and concentration.

2. CONCEPTION OF EXPERIMENTAL DESIGN METHODS

In a pattern classification problem, Optimal Experimental Design depends on both the design problem and the property of the modelling tool, such as the generalisation/localization ability. Since PPR is based on a nearest neighbourhood idea, if the problem consists of a single straight class boundary, a symmetric pair of training data with one datum on each side of this boundary is sufficient to determine the boundary in the middle of the two points. For example, for the simple detection problem: $y = x + n$, where x is the input, y is the output(measurement), n is a noise term and class I:$x<0.5$,class II:$x>0.5$. In a simple Factorial Design, the designer can easily produce a pair of data symmetric with respect to the boundary, x+-Delta. A pair of data points symmetric to the boundary will be referred as a conjugate pair. On the other hand when the class boundary is curvilinear, many conjugate pairs of training data may be needed, in order to identify the true boundary. An example for this situation is shown in Figure 1 where a curvilinear boundary, i.e. composed of different straight portions, separates two fault classes to be identified. The output of class I given by the rhomboids is equal to 1 whereas the output corresponding to class II given by the triangles is equal to 2. Thus, the problem consists of inferring the class from the output values. At least 8 training data points are needed, as shown in Figure 2, in order to accurately determine the class boundary between the two classes using a PPR algorithm.

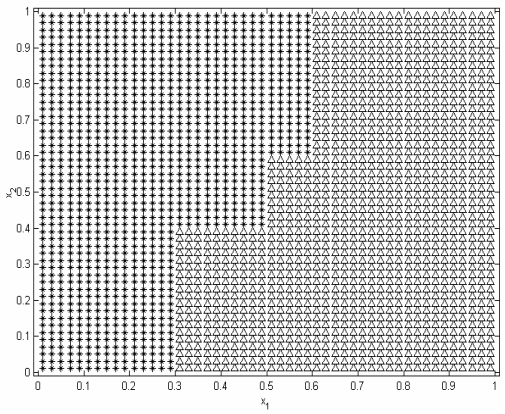


Fig. 1. True classification of a data pattern

To illustrate this point, let us assume in Figure 2, an initial training set #1 is given by points *a* and *b*. Then training set #2 will be selected from either one of the conjugate pairs defined by coordinates (*c, d*), or (*e, f*), or (*g, h*) to test which pair result in the best model, in terms of the smallest class misclassification. The test set consists of all the grid points shown in Figure 1. If *c* and *d* are used as new training data, the PPR method produces a pattern classification as presented in Figure 3. The misclassification rate is as high as 26%. If *e* and *f* are used as the new training data, the corresponding data classification is shown in Figure 4. The misclassification rate is then reduced to only 4.5%. Finally, if the conjugate pair *g* and *h* are applied as the new training data, the misclassification rate will be further reduced to 3.9%.

The above example shows that the design of training data set #2 will result in the best model, if the new data is added to the area with the lowest data density. A major problem with Factorial Design is that it does not automatically search the sparseness of training data in the data domain. Therefore, an algorithm, which can automatically search the data domain and exploit the sparseness of training data, is expected to give better results with the same amount of data. This observation inspired a novel experimental

design method for fault detection, which will be introduced in the Section 4.2.

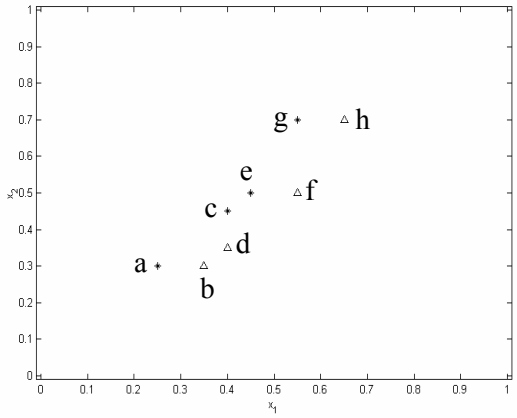


Fig. 2 Minimum training data needed to identify class boundary accurately

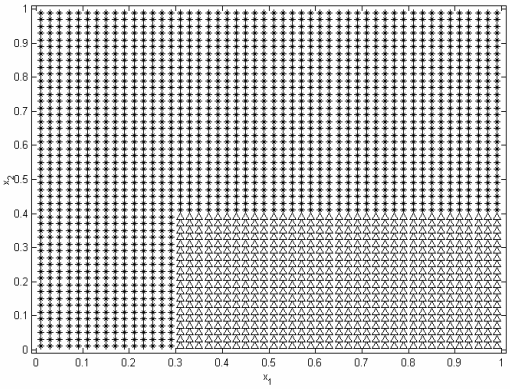


Fig. 3. Classification results with new training data *c* and *d*

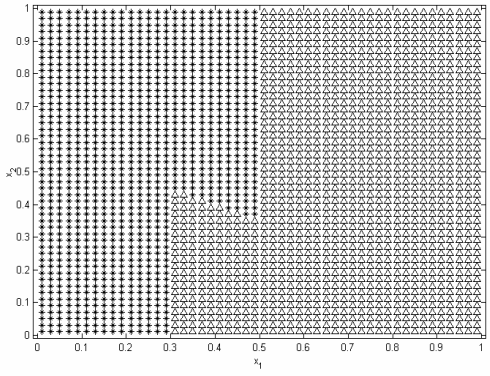


Fig. 4 Classification results with new training data *e* and *f*

3. TRAINING DATA SET #1

The CSTR model used in this case study has been originally used by Venkatasubramanian to study a fault detection algorithm based on a Radial Basis Functions Neural Network [7] and it is based on conventional component and energy balances. Two outputs are measured: the outlet temperature and the outlet concentration. These variables are then used to identify the faults in two process inputs: the inlet temperature and the inlet concentration. The magnitude of the inlet temperature, T_o, is allowed to change from the normal steady state value to 3.0 times of that value. The magnitude of another process input, the inlet concentration C_{Ao}, is allowed to change from the normal steady state value to 1.6 times of that value.

As mentioned before, the experimental design methods discussed in this case study are based on an initial set of training data which is generated by a Factorial Design. The design is carried out at 3 levels: 0, 5, 10. There are two factors: the inlet temperature, and the inlet concentration. For a full Factorial Design with 2 factors and 3 factorial levels, there are 9 possible experimental conditions, as shown in Table 1.

Table 1. Factorial Design for training data set #1

Experiment run	Factorial level 1	Factorial level 2
1	0	0
2	0	5
3	0	10
4	5	0
5	5	5
6	5	10
7	10	0
8	10	5
9	10	10

The magnitude of inlet temperature, with respect to its normal steady state value, is decided according to Equation (1):

$$Magnitude\ 1 = factorial\ level\ 1 \times 0.2 + 1.0 \quad (1)$$

The magnitude of inlet concentration, with respect to its normal steady state value, is decided by Equation (2):

$$Magnitude\ 2 = factorial\ level\ 2 \times 0.06 + 1.0 \quad (2)$$

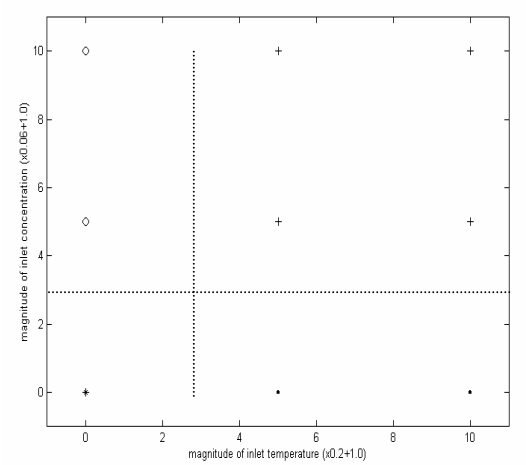
Based on these definitions the faults are defined as shown in Table 2. The generated training data is plotted in the process input space in Figure 5.

Corresponding to the data pattern in the process inputs in Figure 5, the process outputs, i.e., the reactor temperature and concentration, are generated using the CSTR model, to train a PPR model. In order to test the performance of the PPR model, a group of testing data has been designed. The testing data are also generated by a Factorial Design on 2 factors: the inlet temperature and the inlet concentration, but at 20 factorial levels: 0.25, 0.75, 1.25,…, 8.75, 9.25, 9.75; i.e., from 0.25 to 9.75, with an interval of

Table 2. Definitions of Faults for the CSTR example

Normal operation (Class 1):	T_o < 1.6 times its normal steady state value; C_{Ao} < 1.18 times its normal steady state value;
high inlet concentration fault (Class 2):	T_o < 1.6 times its normal steady state value; C_{Ao} ≥ 1.18 times its normal steady state value;
High inlet temperature fault (Class 3):	T_o ≥ 1.6 times its normal steady state value; C_{Ao} < 1.18 times its normal steady state value;
Concurrent faults (Class 4):	T_o ≥ 1.6 times its normal steady state value; C_{Ao} ≥ 1.18 times its normal steady state value;

0.5. A full Factorial Design on two factors at 20 factorial levels results in 400 experimental conditions. The percentage of misclassification is calculated to be 38.5%. The misclassifications are mainly due to the insufficiency of training data.



✱Class 1; ○Class 2; •Class 3; +Class 4; — true class boundary

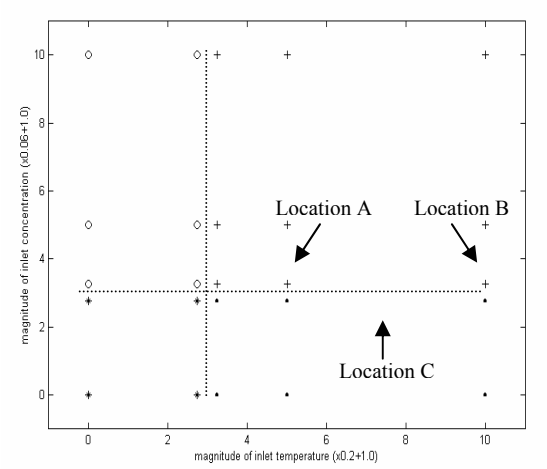
Fig. 5. Original training data in process input space,

4. COMPARISON OF EXPERIMENTAL DESIGN METHODS FOR TRAINING DATA SET #2

4.1 Factorial Design

To generate the new training data #2 using Factorial Design, two factorial levels are added on each of the two factors. Now each factor has a total of five factorial levels: 0, 2.75, 3.25, 5, 10. After the addition of the new data, the training data in the process input space has a pattern as shown in Figure 6. Compared to the original training data, 16 new data have been added.

The PPR model trained with these new data results in 23.25% misclassification, when tested on the testing data set defined above. Checking the locations of the misclassification, it is found that the new training data generated by the Factorial Design does not reduce the misclassification in location *C* in Figure 6. The figure shows that the Factorial Design adds new training data to location *A* and *B*, because sufficient data already exist there. Location *C* has a much lower density of training data than location *A* and *B*. But the Factorial Design does not add any new training data in location *C*, because it does not search for the sparseness of the training data. A combed effect of the sparseness of training data and the curvilinear shape of the class boundary in the process output space results in high misclassification in location *C*.



✱Class 1; ○Class 2; •Class 3; +Class 4; — true class boundary

Fig. 6. Training data in the process input space, after Factorial Design

4.2 Gaussian Probability Design

As for the Factorial Design, the Gaussian Probability Design is also carried out, based on the original training data in the process input space, which is shown in Figure 5. The design is implemented in the following steps.

Step 1: The Gaussian probability is calculated on a set of experimental data, with respect to their process inputs. The experimental data to be

considered are the grid points next to the class boundary. The sampling rates in all dimensions of this grid are decided based on a trade-off between the design accuracy and the computational expense. The Gaussian probability of a datum is calculated by the following equation.

$$p_j = \sum_{i=1}^{N_T} e^{-(\frac{x_{j,1}-x_{i,1}}{s_1})^2-(\frac{x_{j,2}-x_{i,2}}{s_2})^2\cdots-(\frac{x_{j,d}-x_{i,d}}{s_d})^2} \quad (3)$$

Where, p_j represents the Gaussian probability of the j^{th} experimental datum, with respect to all the training data in data set #1; N_T is the total number of the original training data in data set #1. For the i^{th} training datum $X_i = \begin{bmatrix} x_{i,1} & x_{i,2} \cdots x_{i,d} \end{bmatrix}^T$, $x_{i,d}$ represents the process input of the i^{th} training datum, in the d^{th} dimension; s_d is the sampling rate of the process input, in the d^{th} dimension.

Step 2: An experimental datum is accepted as a new training datum, if it has the lowest probability value, p_{min}.

$$p_{\min} = \min_{1\le j\le N_E} \{ p_j \} \quad (4)$$

Where, N_E is the total number of experimental data.

Step 3: Once a new training datum is selected, its conjugate point on the other side of a class boundary is also selected as a new training datum. A conjugate point can further generate a new one with respect to another individual class boundary. This procedure is carried on, until each new training datum has a conjugate point on the other side of a class boundary, to form a conjugate pair. Conjugate pairs of training data, across the class boundary, are sought, because they obviously provide equal amount of class information, on the two sides of the boundary. Unequal amount of training data on the two sides of the class boundary will cause the PPR model to identify a class boundary that is biased towards the side with less training data.

This proposed design method can be carried out in a sequential approach manner. If the design objective, such as the misclassification rate in testing, is not met in the previous run of design, additional training sets can be added according to steps 1-3 above.

For the CSTR example the experimental data to be considered in step #1 are shown in Figure 7. The Gaussian probability is calculated on each experimental datum, according to Equation (3). New training data are then selected from the experimental data following Equation (4). The following three data points have been found to locate in the areas with the (equally) lowest density: [2.75 2.75], [7.75 2.75], [2.75 7.75]. The conjugate points of these new training data are also selected, according to Step 3 above. For example, the data point '*a*' in Figure 8 is a new training datum selected in the previous design steps. The data point '*b*' is accepted as a new training datum, because it is the conjugate point of '*a*', with respect to class boundary *1*. The data point '*d*' is another conjugate point of '*a*', with respect to class boundary *4*. Since data point '*c*' is a conjugate point of both '*b*' and '*d*', with respect to the class boundary *2* and *3* respectively, it is also taken as a new training datum, following Step 3. Finally, the new training data, together with the original ones, are presented in Figure 8.

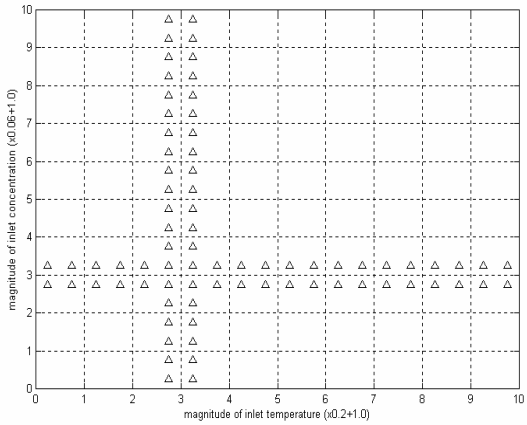


Fig. 7. Experimental data in Gaussian Probability Design,

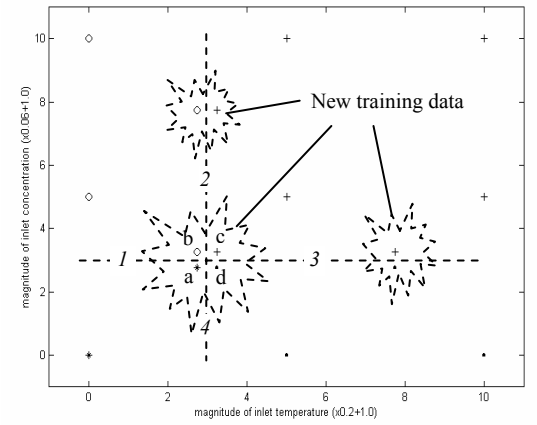


Fig. 8. New training data, together with original ones, in process input space,

Based on the training data set in Figure 8, a PPR model is obtained that gives only 16.25% misclassification in the testing. This result is better than 23.25% misclassification by the Factorial Design. Furthermore, the Gaussian

Probability Design achieves this higher accuracy with only 8 new training data, significantly less than 16 new training data calculated by the Factorial Design. In summary, the Factorial Design adds training data to locations *A* and *B*, which are already dense with training data, but leave location *C* blank. On the other hand, the Gaussian Probability Design automatically searches for the data void, and adds new training data to location *C*, where the data density is the lowest. The difference in the location of new training data shows the advantage of the Gaussian Probability Design, in automatically searching for data voids.

5. CONCLUSION

A novel experimental-design method, referred to as Gaussian Probability Design, is proposed and compared to the conventional Factorial Design. The comparison is carried out on a CSTR process. The simulation results are summarised in Table 3. The results show that, for PPR, the Gaussian Probability Design provides a more accurate classification of faults as compared to the Factorial Design.

Table 3. Comparison results of experimental design methods

	Misclassification
Original training data	*38.50%*
Factorial Design *(16 new data)*	*23.25%*
Gaussian Probability Design *(8 new data)*	*16.25%*

The key disadvantage of Factorial Design for training a fault detection algorithm is that it can not take into account the sparseness of the data. The location of new training data is decided by a combination of the factorial levels. The design adds training data not only on the class boundary, but also inside a class as well. This is often unnecessary for a model, which can make reasonable generalisation, such as PPR. For these models, a training datum near the class boundary is more important than one inside the class. The classification of a datum inside a class can be correctly determined by the PPR technique through interpolation, based on the data located near the class boundary. When the class boundary is of complicated shape, or the dimension of the problem is high, i.e. it has a large number of inputs and outputs, Factorial Design may lead to a prohibitively large amount of training data.

Our simulation results show that using more training data does not necessarily mean less misclassification. This further illustrates that a ***random design*** of training data may lead to poor generalization results.

The Gaussian Probability Design investigates the sparseness of the training data, and adds training data to the class boundary area, where the data density is the lowest. Since low density indicates insufficiency of training data, this method is *efficient* in filling out the data void. The rationale for using the Gaussian probability distribution based on assessing the probability of points in the neighbourhood of the boundary to belong to a specific class based on an initial set of training data. Finally, the Gaussian Probability design results in smaller number of experiments as compared to Factorial Design, since it adds points only in the neighbourhood of the class boundary. It does not add unnecessary data points away from the class boundary that can be easily interpolated by the PPR algorithm based on the points located at the boundary.

REFERENCE

[1] Friedman, J.H. and W. Stuetzle, (1981) Projection Pursuit Regression, *Journal of the American Statistical Association*, no.376, p:817

[2] Flick, T.e., L.K. Jones, R.G. Pries, and C. Herman, (1990) Pattern Classification using Projection Pursuit, *Pattern Recognition*, Vol. 23, No. 12, pp. 1367-1376

[3] Utojo, U., B. R. Bakshi, (1995) Connections between Neural Networks and multivariate Statistical Methods: An Overview, *Neural Networks in Bioprocessing and Chemical Engineering*, Academic Press

[4] Hwang, J.N., S. R. Lay, M. Maechler, R. D. Martin, J. Schimert, (1994) Regression Modeling in Back-propagation and Projection Pursuit Learning, *IEEE Transactions on Neural Network*, vol.5, no.3

[5] Lou, S. J., H. Budman, and T. A. Duever, (accepted in August 2002), Comparison of fault detection techniques: Haar Wave-Net versus Projection Pursuit Regression, *Journal of Process Control*

[6] Cohn, David A., (1996) Neural network exploration using optimal experimental design, *Neural Networks*, vol.9, no.6, pp1071-1083

[7] Baughman, D. R.; Liu, Y. A, (1995) *Neural Networks in Bioprocessing and Chemical Engineering*, Academic Press

Fault Diagnosis and Fault Identification for Fault-Tolerant Control of Chemical Processes

Kap-Kyun Noh* **and En Sup Yoon**

School of Chemical Engineering, Seoul National University, Seoul 151-742 Korea
*(*Email: kknoh@pslab.snu.ac.kr)*

Abstract: Fault-tolerant control (FTC) of nonlinear systems is presented within an adaptive control framework. FTC can be accomplished by three subtasks; fault-diagnosis, fault identification and adaptive nonlinear control. In order to diagnose a fault at a time, a set of residual generators for fault diagnosis is designed by means of unknown input observers. When disturbances exist, disturbance-decoupled model could be derived for reliable diagnosis. Fault identification following fault diagnosis is an analogue to control task; the diagnosed fault is regarded as a control input and found out such that the residual from residual generator incorporating identification task is driven to reference zero. And, feedback linearizing control is liked with fault diagnosis and fault identification to compensate for a fault to the process. A three-tank system is taken as an example for demonstration of the presented FTC.

Keywords: fault diagnosis, fault identification, fault-tolerant control, nonlinear systems.

1. INTRODUCTION

Automated chemical process has yield to high quality and high efficiency of normal operation, but has become more complicated and more vulnerable to faults because processes have been integrated into wider operation platform and operation algorithms may be another fault sources. Furthermore, a simple fault could be amplified by the control system and developed into malfunction of the loop, even into a failure at the plant level. This requires advanced fault diagnosis and supervision to improve reliability and safety. A cost-effective way to achieve the goal is by means of a fault-tolerant control (FTC) (Blank et al., 2001).

The FTC could be achieved by merging the fault information obtained from fault diagnosis and fault identification into the control system for fault accommodation. Fault diagnosis scheme has to efficiently detect and identify a fault even when the process is under closed-loop control and varies over wide range, and following fault diagnosis, fault identification estimates time-varying behaviors of the diagnosed fault which then is reflected into the control law to accommodate the fault.

Nonlinear observer-based fault diagnosis where the research that has been made around a linear system has been lately extended to nonlinear system (Frank and Ding, 1997) is presented. The model used for observer design could be decoupled from disturbances and/or a dedicated fault by means of state transformation, and so resulting observer is unaffected by disturbances and possess structured sensitivities to the faults. The state transformation is based on the concept of fault (disturbance) detectability index and so it contains outputs and their derivatives. A bank of residual generators providing generalized residuals set and decisions function such as fixed threshold are needed to diagnose a fault. The design method and conditions of such nonlinear observers will be presented.

When a fault is detected and localized, its magnitude and time-evolving behavior should be identified to take a countermeasure keeping process performances. The same model and the concept of fault relative order as those of fault diagnosis are utilized to obtain design model which allows to reconstruct the fault and on which the residual generator is designed. Fault

identification is formulated into control task problem such that the residual from the residual generator driven by identified fault is forced to zero. The method is based on the conditions of input observability (Hou and Patton, 1998; Kabore and Wang, 1999); the residual responds to only a specific fault and reconstructs the actual fault.

Fault-tolerant control is achieved by combining input-output feedback linearizing control law based on the fault-parameterized model with fault informations obtained from fault diagnosis fault identification, which is reduced to the adaptive linearizing control (Sastry and Isidori, 1989).

2. ON-LINE FAULT DIAGNOSIS

Consider nonlinear systems affine in control, disturbance and fault mode;

$$\dot{x} = g_0(x) + \sum_{s=1}^{m} g_s(x)u_s(t) + \sum_{s=1}^{n_d} D_s(x)d_s(t) + \sum_{s=1}^{n_f} e_s(x)f_s(t) \quad (1a)$$

$$y = h(x) \quad (1b)$$

where $x \in \Gamma \subset R^n$, $u \in \Omega_u \subset R^m$, $y \in R^p$ are the state vector, the control input vector, and the output vector. $d \in \Omega_d \subset R^{n_d}$ is the disturbance vector including model errors, and $f \in \Omega_f \subset R^{n_f}$ is the fault mode vector for component faults and actuator faults, and both vectors consist of unknown time-varying functions. $g_s(x), s = 0,1,..,m$; $D_s(x), s = 1,..,n_d$ and $e_s(x), s = 1,..,n_f$ are smooth vector fields, and $h_s(x), s = 1,..,q$ are smooth scalar fields, respectively, on Γ and they are known. Here, Γ is a physically feasible and bounded set, and Ω with subscripts are bounded sets for corresponding inputs.

FDI Strategy: When the process is subject to disturbances, model-based FDI strategy may give misleading analytical redundancies. Thus, to improve the FDI performance, a means of creating analytical redundancy not affected by disturbances should be devised as in the following steps;

S1: Obtain a reduced model decoupled from disturbances, but still affected by faults.

S2: For the detection of faults remained at step S1, produce a residual generator based on disturbance-decoupled model obtained at step S1.

S3: For fault isolation among the faults at step S1, partition faults into isolable fault subsets and generate a bank of residual generators in which each one is dedicated to each fault subset. It is based on a fault-added disturbance-decoupled model and provides generalized residuals set giving different sensitivities to different fault subsets. This feature enables to uniquely isolate a fault subset by checking the values of all residuals.

Fault Detectability: A fault is said to be detectable if a fault affects at least one of observable outputs. From this point of view, the detectability of a fault can be characterized by using the concept of relative order known as a fault detectability index in fault diagnosis field (Liu and Si, 1997), which is defined as the smallest integer, r_{f_j}, such that

$$L_{e_j} L_{g_0}^{r_{f_j}-1} h_i(x) \neq 0 \qquad \exists i \in [1,.., p], \forall x \in \Gamma \quad (2)$$

where $e_j(x)$ is a fault vector field of a fault, f_j and $h_i(x)$ is a scalar function of the output, y_i. If fault relative order, r_{f_j}, is less than or equal to n, the fault is detectable. Otherwise, the fault is not detectable. Disturbance relative order, r_{d_j}, for the disturbance, d_j, can be defined in the same way.

Disturbance-Decoupled Model: Residual generators will be obtained through the design of observers based on disturbance-decoupled or a fault-added disturbance-decoupled models. The conditions that guarantee a state transformation inducing disturbance-decoupled nonlinear model are based on the concept of well-defined disturbance relative order and are analogies to those of feedback linearization in nonlinear control theory (Isidori, 1989).

Consider the system with only disturbances;

$$\dot{x} = g_0(x) + \sum_{s=1}^{n_d} D_s(x)d_s(t) \quad (3a)$$

$$y = h(x) \quad (3b)$$

For above system, if conditions below are met,

C1. The relative order, r_{d_i}, of the output, $y_i, i = 1,..,l$ $(l < p)$, with respect to disturbance vector, d, is well defined.

C2. The characteristic matrix, $C_D(x)$, formed at the r_{d_i} th times derivative of each output, y_i, before the disturbance vector, d, has full row rank.

$$C_D(x)=\begin{bmatrix} L_{D_1}L_{g_0}^{r_{d_1}-1}h_1 & \cdots & L_{D_{n_d}}L_{g_0}^{r_{d_1}-1}h_1 \\ \vdots & \vdots & \vdots \\ L_{D_1}L_{g_0}^{r_{d_{n_d}}-1}h_{n_d} & \cdots & L_{D_{n_d}}L_{g_0}^{r_{d_{n_d}}-1}h_{n_d} \end{bmatrix}(x) \quad (4)$$

C3. The distribution, $\Delta(x)=span\{D_1(x),..,D_{n_d}(x)\}$, identified and spanned by disturbance vector fields has constant rank, $\bar{q}(\le q)$ and it is involutive, i.e., Lie brackets of any pair of vector fields belonging to $\Delta(x)$ belong to $\Delta(x)$ again.

then, there exists a state transformation

$$z=\Phi(x)=\begin{bmatrix} x^i \\ \bar{h} \end{bmatrix}=\begin{bmatrix} y_i \\ \vdots \\ y_i^{(r_i-1)} \\ \bar{h} \end{bmatrix}(x)=\begin{bmatrix} h_i \\ \vdots \\ L_{g_0}^{r_{d_i}-1}h_i \\ \bar{h} \end{bmatrix}(x) \quad \forall i\in[1,..,p] \quad (5)$$

where $x=\left[x^i, \ .. \ ,x^l\right]^T$, $x^i=\left[x_1^i, \ .. \ ,x_{r_{d_i}}^i\right]^T=\left[y_1,..,y_1^{(r_{d_i}-1)}\right]^T$, and $\bar{h}\in R^{n-r_d}$ with $r_d=r_{d_1}+..+r_{d_{n_d}}$ consists of scalar fields, $\bar{h}_i$, such that $L_D\bar{h}_i(x)=0$ and makes the state transformation locally invertible (Isidori, 1989);

$$x=\Phi^{-1}(z)=\Phi^{-1}\left(\left[x^i, \ \bar{h}\right]^T\right) \quad \forall i\in[1,..,l] \quad (6)$$

■

The system in the transformed state is

$$\dot{x}_{r_{d_i}}^i=L_{g_0}^{r_{d_i}}h_i(x)+\left[L_{g_1}L_{g_0}^{r_{d_i}-1}h_i \ \cdots \ L_{g_m}L_{g_0}^{r_{d_i}-1}h_i\right]^T(x)u+\left[L_{D_1}L_{g_0}^{r_{d_i}-1}h_i \ \cdots \ L_{D_{n_d}}L_{g_0}^{r_{d_i}-1}h_i\right]^T d$$
$$+\left[L_{e_1}L_{g_0}^{r_{d_i}-1}h_i \ \cdots \ L_{e_{n_f}}L_{g_0}^{r_{d_i}-1}h_i\right]^T(x)f\big|_{x=\Phi^{-1}(x,h)} \quad (7a)$$

$$\begin{bmatrix} \dot{x}_1^i \\ \vdots \\ \dot{x}_{r_{d_i}}^i \end{bmatrix}=A_i\begin{bmatrix} x_1^i \\ \vdots \\ x_{r_{d_i}}^i \end{bmatrix}+B_i x_{r_{d_i}}^i+\begin{bmatrix} L_{g_1}h_i & \cdots & L_{g_m}h_i \\ \vdots & \cdots & \vdots \\ L_{g_1}L_{g_0}^{r_{d_1}-2}h_i & \cdots & L_{g_m}L_{g_0}^{r_{d_{n_d}}-2}h_i \end{bmatrix}(x)u$$
$$+\begin{bmatrix} L_{e_1}h_i & \cdots & L_{e_{n_f}}h_i \\ \vdots & \cdots & \vdots \\ L_{e_1}L_{g_0}^{r_{d_{n_f}}-2}h_i & \cdots & L_{e_{n_f}}L_{g_0}^{r_{d_{n_f}}-2}h_i \end{bmatrix}(x)f\big|_{x=\Phi^{-1}(x,h)} \quad (7b)$$

$$\dot{\bar{h}}=\bar{h}(x,\bar{h},u,f) \quad (7c)$$

where A_i is the $(r_i-1)\times(r_i-1)$ matrix and B_i is the $(r_i-1)\times 1$ vector and (A_i,B_i) is in a canonical form. The transformed system is divided into two subsystems according to explicit dependence on the disturbances. Disturbance-decoupled model consists of lower subsystems (7b)(7c) and it will be utilized for the design of nonlinear observers that are robust to disturbances but sensitive to faults.

Remark: The model is driven by the faults and as well $x_{r_{d_i}}^i$ as new inputs, not directly available. Its estimation is to use a differentiator filter. But, since each filter is driven by a measured output, the effects of the faults are reflected into estimated output and its successive derivatives, and the disturbance decoupled model actually useful for FDI is limited to the (7c). But, their estimates from filters will be used for provision of unavailable states with (7c). This means that the faults whose relative orders are more than two cannot be separable through the resulting decoupled model unless all states are available.

Fault Detection: The design of a nonlinear observer for FD (Fault Detection) is performed on the model (7c) and various design methods of an observer in the literatures can be considered. If state estimates from differentiator filters are sufficiently accurate, the basis model becomes;

$$\dot{\bar{h}}=\bar{h}(\bar{h},x,u,f) \quad (8a)$$

$$Y_{ex}=C\bar{h} \quad (8b)$$

Under some assumptions below,

A1 Extra outputs, Y_{ex}, not involved in disturbance decoupling are available and linear in the state.

A2 The basis model is observable from extra outputs.

A3 The basis model can be put into a time-varying linear system with a Lipschitz nonlinear perturbation;

$$\dot{\bar{h}}=Q(x,u)\bar{h}+N(\bar{h},x,u)+E(\bar{h},x,f) \quad (9)$$

$$\left\|N(\bar{h},x,u)-N(\hat{\bar{h}},x,u)\right\|\le\bar{h}_o\left\|\bar{h}-\hat{\bar{h}}\right\| \quad (10)$$

where $\bar{h}_o$ is a constant.

A4 Constant matrix, K, can be chosen such that

$$l\left[\left(Q(x,u)-KC\right)^T+\left(Q(x,u)-KC\right)\right]<-a \quad \forall x,u \quad (11)$$

where $l[t;\cdot]$ denotes the eigenvalues of time-varying matrix at time t and $a>0$ is a constant.

A candidate observer can be taken as

$$\dot{\hat{\bar{h}}} = Q(x,u)\hat{\bar{h}} + N(\hat{\bar{h}},x,u) + KC(\bar{h} - \hat{\bar{h}}) \quad (12)$$

and then the error dynamics with a residual is

$$\dot{e} = (Q(x,u) - KC)\, e + N(\bar{h},x,u) - N(\hat{\bar{h}},x,u) + E(\bar{h},x,f) \quad (13a)$$

$$r = \left(e^T C^T C e\right)^{1/2} \quad (13b)$$

where $e = \bar{h} - \hat{\bar{h}}$.

Assumption A3 can be easily met since the basis model is reduced and a nonlinear perturbation term is allowed, and when the model is defined over bounded domain. Assumption A4 makes the error dynamics stable if the design matrix, K, can be chosen such that for $a > 2\bar{h}_0$ it is satisfied (Slotine and Li, 1984).

In the absence of any fault ($f = 0$), the error dynamics are made stable around the equilibrium, $e = 0$ and so the residual is decaying to zero, indicating no fault. But, in the presence of any fault ($f \neq 0$), the error no longer stays at zero due to nonlinear nonvanishing effects by a fault and so resulting nonzero residual indicates the occurrence of a fault.

Fault Isolation: Due to the allowance for disturbance decoupling, the generalized observer scheme (GOS) (Chen and Patton, 2000) providing a generalized residuals set for fault isolation is adopted. To implement the GOS, the model decoupled from disturbances and as well a fault subset is needed, and it can be obtained in the same way as before for augmented disturbances. Robust fault isolation between two faults in sense of the GOS can be checked by;

$$rank\left(\left(\frac{\partial \bar{h}}{\partial x}\right)(e_i(x), e_j(x))\right) = rank\left(e_i(x), e_j(x)\right) = 2 \quad (14)$$

where $e_i(x)$ and $e_j(x)$ are vector fields of considered faults. This condition makes sure that two faults are not only reflected into disturbance-decoupled model but also not decoupled at a time. The observability of a fault-added disturbance-decoupled model is assumed.

The design of residual generators for fault isolation is based on a designated fault and disturbance decoupled model and their designs proceed as before. When one fault occurs at a time, the right fault isolation can be done by checking values of all residuals.

3. FAULT IDENTIFICATION

When the focus is made on the design of fault tolerant control, in addition to fault detection and isolation, fault identification identifying the size of the fault and its time varying behavior has to be solved. Faults to be identified are limited to the faults isolated by fault diagnosis.

Fault identification problem can be reformulated into control task problem. In this approach, fault signal is regarded as a control input to the system and a feedback control is found such that the residual tracks a zero reference. Since the residual is usually given as the difference between measured output and estimated output, the fault forces the estimated outputs from residual generator to track measured outputs. When there is no fault, the feedback control will decay to zero, indicating no fault, while in the presence of a fault, the feedback control will provide a control input which is actually an estimate of the fault.

Consider the system with single fault vector;

$$\dot{x} = g_0(x) + \sum_{s=1}^{m} g_s(x)u_s(t) + \sum_{s=1}^{n_{fi}} e_s(x)\, f_s(t) \quad (15a)$$

$$y = h(x) \quad (15b)$$

where f_s is one element of fault vector, $\overline{f}_i$, and descriptions of variables, x, y, u , and functions, g_s, e_s, h_j, are the same as those of the system (1).

A1 the fault detectability index, k_j, for the fault vector, $\overline{f}_i$, such that

$$L_{e_s} L_{g_0}^{k_j - 1} h_j(x) \neq 0 \quad s \in \{1,..,n_{f_i}\}, j \in \{1,..,\overline{p}\}, \forall x \in \Gamma \quad (16)$$

is well defined.

A2. the matrix,

$$E_i(x) = \begin{pmatrix} E^i_{1,k_1}(x) \\ E^i_{2,k_2}(x) \\ \vdots \\ E^i_{\overline{p},k_{\overline{p}}}(x) \end{pmatrix} = \begin{pmatrix} \sum_{s=1}^{n_{fi}} L_{e_s} L_{g_0}^{k_1 - 1} h_1(x) \\ \sum_{s=1}^{n_{fi}} L_{e_s} L_{g_0}^{k_2 - 1} h_2(x) \\ \vdots \\ \sum_{s=1}^{n_{fi}} L_{e_s} L_{g_0}^{k_{\overline{p}} - 1} h_{\overline{p}}(x) \end{pmatrix} \quad (17)$$

is a full column rank for all $x \in \Gamma$.

Then, the following state coordinate can be taken

$$x^j = \begin{pmatrix} x_1^j \\ x_2^j \\ \vdots \\ x_{k_j}^j \end{pmatrix} = \begin{pmatrix} h_j(x) \\ L_{g_0} h_j(x) \\ \vdots \\ L_{g_0}^{k_j-1} h_j(x) \end{pmatrix}, \quad j = 1,.., \bar{p} \tag{18}$$

and the system is transformed into following form, which is the same as (7);

$$\dot{x}^j = A_j^i x^j + B_j^i G_{0,k_j}^1(x,V) + \sum_{s=1}^{m} G_{s,j}^1(x,V) u_s + E_j^i(x,V) \bar{f}_i \tag{19a}$$

$$V = G_0^2(x,V) + \sum_{s=1}^{m} G_s^2(x,V) u_s \tag{19b}$$

$$y_j = C_j^i x^j = x_1^j$$

where $x^i = (x_1^j,..,x_{k_j}^j)$, $x = (x^1,..,x^{\bar{p}}) = \Phi^1(x) \in R^k$ with $k = \sum_{s=1}^{\bar{p}} k_s$, $V = \Phi^2(x) \in R^{n-k}$ such that $h = (x,V)^T$ is invertible. $G_{s,j}^1 = \left[0,..,L_{g_s} L_{g}^{k_j-1} h_j(x)\right]^T, s = 0,1,.., \bar{p}$,

$E_j^i = \left[0,..,\sum_{s=1}^{n_{f_i}} L_{e_s} L_{g_0}^{k_j-1} h_j(x)\right]^T$, and (A_j^i, C_j^i) is in a observable canonical form and (A_j^i, B_j^i) is in a controllable canonical form.

Based on the structure of the above systems, a candidate residual generator incorporating fault identification is taken as (Kabore and Wang, 1999):

$$\dot{\hat{x}}^j = A_j^i \hat{x}^j + B_j^i G_{0,k_j}^1(\hat{x},\hat{V}) + \sum_{s=1}^{m} G_{s,j}^1(\hat{x},\hat{V}) u_s + E_j^i(\hat{x},\hat{V}) \hat{\bar{f}}_i - K_j^i(C_j \hat{x}^j - y_j) \tag{20a}$$

$$\dot{\hat{V}} = G_0^2(\hat{x},\hat{V}) + \sum_{s=1}^{m} G_s^2(\hat{x},\hat{V}) u_s \tag{20b}$$

$$r_j^i = C_j^i x^j - y_j \tag{20c}$$

$$\hat{\bar{f}}_i = E_i^{-1}(\hat{x},\hat{V}) \left[v - \begin{pmatrix} G_{0,k_1}^1(\hat{x},\hat{V}) + \sum_{s=1}^{m} G_{s,k_1}^1(\hat{x},\hat{V}) u_s \\ G_{0,k_2}^1(\hat{x},\hat{V}) + \sum_{s=1}^{m} G_{s,k_2}^1(\hat{x},\hat{V}) u_s \\ \vdots \\ G_{0,k_{\bar{p}}}^1(\hat{x},\hat{V}) + \sum_{s=1}^{m} G_{s,k_{\bar{p}}}^1(\hat{x},\hat{V}) u_s \end{pmatrix} \right] \tag{21}$$

where v is a proper new input and taken as

$$v = \begin{pmatrix} v_1 \\ v_2 \\ \vdots \\ v_{\bar{p}} \end{pmatrix} = \begin{pmatrix} y_1^{(k_1)} \\ y_2^{(k_2)} \\ \vdots \\ y_{\bar{p}}^{(k_{\bar{p}})} \end{pmatrix} \tag{22}$$

and observer gain, K_j^i, is chosen such that each observable form may be stable and makes the error exponentially converge to zero.

4. FAULT-TOLERANT CONTROL

To accommodate identified faults, the linearizing control law can be linked with fault identification, which is reduced to adaptive linearzing control (Sastry and Isidori, 1989; Teel et. al., 1991; Hu, 1999). When the resulting control law is applied to the faulty process, quasi-linear system results in.

$$\dot{e} = A_c e + W(e,h,u_a)\Theta \tag{23}$$

where A_c is a $(r+1)\times(r+1)$ Hurwitz matrix, u_a is approximately linearizing control law and e is error coordinate. Θ is the fault error and the matrix, $W(\cdot)$ is nonlinear functions before the fault error. As fault error is decaying to zero, the quasi-linear system becomes asymptotically linearized. The stability of the perturbed system is ensured if the perturbing term is bounded over the domain and the poles of the Hurwitz linear system are placed sufficiently deep into the left half of s-plane (Zak, 1990).

5. APPLICATION

Figure1 shows a schematic of a three-tank system. Using the mass balance and mass flows by Torricell's law, the system can be described as;

$$\dot{x}_1 = (u_{10} - a_1 s_{13} \operatorname{sgn}(x_1 - x_3)\sqrt{2g|x_1 - x_3|})/A$$

$$\dot{x}_2 = (u_{20} - a_3 s_{32} \operatorname{sgn}(x_2 - x_3)\sqrt{2g|x_2 - x_3|} - a_2 s_{20}\sqrt{2g x_2})/A$$

$$\dot{x}_3 = (a_1 s_{13} \operatorname{sgn}(x_1 - x_3)\sqrt{2g|x_1 - x_3|} - a_3 s_{32} \operatorname{sgn}(x_3 - x_2)\sqrt{2g|x_3 - x_2|})/A$$

where the state, x, is the level of each tank and u_{10}, u_{20} are mass inflows. A is the cross-section of tank and s_{13}, s_{32}, s_{20} are the cross sections of interconnected and outlet pipe, respectively. a_i is scaling constants and g is the gravity constant. f_i denotes faults caused by various reasons such as leaks, clogging and pump failures. Levels are available. Fault distribution matrix, $E(x)$, to the fault, f, is given as;

$$E(x) = \begin{bmatrix} -\frac{\sqrt{2g x_1}}{A} & 0 & 0 & \frac{u_{10}}{A} & 0 & \frac{\operatorname{sgn}(x_1 - x_3)\sqrt{2g|x_1 - x_3|}}{A} & 0 & 0 \\ 0 & -\frac{\sqrt{2g x_2}}{A} & 0 & 0 & \frac{u_{20}}{A} & 0 & \frac{\operatorname{sgn}(x_2 - x_3)\sqrt{2g|x_2 - x_3|}}{A} & \frac{\sqrt{2g x_3}}{A} \\ 0 & 0 & -\frac{\sqrt{2g x_3}}{A} & 0 & 0 & -\frac{\operatorname{sgn}(x_1 - x_3)\sqrt{2g|x_1 - x_3|}}{A} & \frac{\operatorname{sgn}(x_3 - x_2)\sqrt{2g|x_3 - x_2|}}{A} & 0 \end{bmatrix}$$

Unknown disturbance is not considered. And, all modeled faults have the fault relative order of one. Outputs carry with measurement noises.

Residual generator for fault detection will be designed based on the whole model (Figure2) and isolable fault subsets for fault isolation are as follows;

$$S_1 = \{f_1, f_4\},\ S_2 = \{f_2, f_5, f_8\},\ S_3 = \{f_3\},\ S_4 = \{f_6\},\ S_5 = \{f_7\}$$

Generalized residuals set, $\{r_1, r_2, r_3, r_4, r_5\}$, is generated from residual generators via observers based on the models in new states, h_j such that $L_{e_i} h_j = 0$ where e_i is all fault vector fields belonging to a fault subset, S_i (Figure 3).

As for fault identification, only one fault at a time is estimated and its fault relative order is one. The state of $\bar{V}$ is available from extra outputs and the first derivative of the output, y_i, is obtained by a differentiator filter. The results are shown in Figure4.

Performance of linearizing control linked with fault identification is compared with simple linearizing control as shown in Figure5.

ACKNOWLEDGEMENT

We acknowledge the financial aid for this research provided by the Brain Korea 21 Program supported by the Ministry of Education. In addition, we would like to thank the Automation & Systems Research Institute and Institute of Chemical Engineering of Seoul National University.

REFERENCES

Blanke, M., Staroswieki, M. and Wu, N.E. (2001) Concepts and Methods in Fault-Tolerant Control. Proc. American Control Conference, Arlington, June, 2606-2620.

Frank, P. M. and Ding, X. (1997) Survey of Robust Residual Generation and Evaluation Methods in Observer-Based Fault Detection Systems, *J. Proc. Contr.* **7**(6), 403-423.

Hou, M. and Patton, R.J. (1998) Input Observability and Input Reconstruction. *Automatica.* **34**(6), 789-794.

Kabore, P. and Wang, H. (1999) On the Design of Fault Diagnosis Filters and Fault Tolerant Control. American Control Conference, San Diego, California, June.

Liu, B. and Si, J. (1997) "Fault Isolation Filter Design for Linear Time-Invariant Systems," *IEEE Trans. Autom. Control*, **42**, 704-707.

Isidori, A. (1989) *Nonlinear Control Systems*, 2nd, Springer-Verlag.

Sastry, S. and Isidori, A. (1989) "Adaptive Control of Linearizable Systems," *IEEE Trans. Autom. Control*, **34**, 1123-1131.

Chen, J. and Patton, R. J. (2000) *Robust Model-based Fault Diagnosis for Dynamic Systems*. Kluwer Academic Publishers, Boston/Dordrecht/London.

Slotine, J-J. E. and Li, W. (1991) *Applied Nonlinear Control*, Prentice Hall, New Jersey.

Hu, Q. and Rangaiah, G.P. (1999), "Adaptive Internal Model Control of Nonlinear Processes," *Chem. Engng. Sci.*, **54**, 1205-1220. (6.7)

Zak, S.H. (1990) On the Stabilization and Observation of Nonlinear/Uncertain Dynamic Systems. *IEEE Trans. Automatic Control.* **35,** 604.

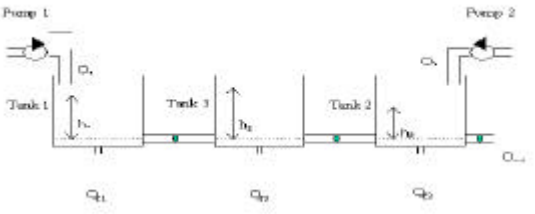


Figure1. Schematic of a three-tank system

Figure2. Residual for fault detection

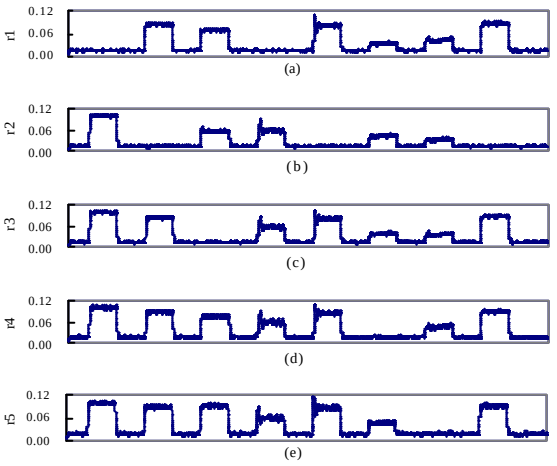


Figure3. Generalized residuals set ($r_i, i = 1,..,5$) for fault isolation

Figure4. Fault identification of fault modes

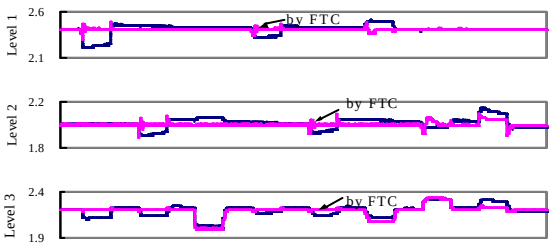


Figure5. Performance by FTC and simple nonlinear control