

SUPERVISORY CONTROL OF MODE-SWITCH PROCESSES: APPLICATION TO A FLEXIBLE BEAM

R.A. Hilhorst*, J. van Amerongen*, P. Löhnberg* and H.J.A.F. Tulleken**

*Control, Systems and Computer Engineering Group, Department of Electrical Engineering, University of Twente, and Mechatronics Research Centre Twente, P.O. Box 217, 7500 AE Enschede, The Netherlands

**Department of Mathematics and Systems Engineering, Operational Analysis and Optimization Group, Koninklijke/Shell-Laboratorium (Shell Research B.V.), P.O. Box 3003, 1003 AA Amsterdam, The Netherlands

Abstract. Many processes operate only around a *limited* number of operation points. In order to have adequate control around each operation point, an adaptive controller could be used. Then, if the operation point changes often, a large number of parameters would have to be adapted over and over again. This prohibits application of conventional adaptive control, which is more suited for processes with slowly changing parameters. Furthermore, continuous adaptation is not always needed or desired. An extension of adaptive control is presented, in which for each operation point the process behaviour can be stored in a memory, retrieved from it and evaluated. These functions are coordinated by a "supervisor". This concept is referred to as *supervisory control*. It leads to an adaptive control structure which, after a learning phase, quickly adjusts the controller parameters based on retrieval of old information, without the need to fully relearn each time. This approach has been tested on an experimental set-up of a flexible beam, but it is directly applicable to processes in e.g. the (petro)chemical industry as well.

Keywords. Adaptive control; Automatic tuning; Learning systems; Mode-switch processes; Time-varying systems; Supervisory control.

1. INTRODUCTION

Many processes cannot be controlled adequately by a *fixed* controller. Then for appropriate control, an adaptive controller or even a variable controller structure is needed. When the process operates in a limited number of operating points, a limited number of controllers suffices. In practical situations a controller will not only yield satisfactory control performance in the operation point, but also in the neighbourhood of this operating point. The set of operating conditions where one controller performs well, is called a *mode*. Processes which frequently return to an earlier seen mode will be referred to as *mode-switch processes* (Hilhorst et al., 1991a). In practice there are several processes which exhibit this behaviour and operate in a limited number of modes only. Such processes are common in e.g. the process industry and in robotics. For instance, this mode-switch behaviour is encountered in a chemical reactor in which the yield and quality of the product has to be optimized to meet market demands, or in a robot which has to transport a limited number of payloads with different masses.

In order to meet the control demands in each operating point, the use of a conventional adaptive controller (Åström and Wittenmark, 1989) could be considered. However, for mode-switch processes the

time needed for adaptation may be too long, i.e. larger than the average residence time in a process mode. For instance, because the closed-loop process signals are not sufficiently exciting. Although the addition of test signals can increase the adaptation speed, it obviously disturbs the process and hence induces performance loss. On the other hand, it seems not to be necessary to repeat the whole adaptation cycle each time the process returns to a certain process mode. The problem is that conventional adaptive controllers forget the useful information which was available before.

A new solution to the problems described above is to exploit the mode-switch behaviour of processes. For this purpose it is attractive to store information related to each previously encountered operation condition in a *memory* and to retrieve it when necessary. When the process enters a new mode, model identification and subsequent controller design are carried out. The model and the controller together should be stored in the memory. These functions are coordinated by a *supervisor*. This approach has the advantage that only recognition of the new mode, and no identification of the process is needed when the process returns to an earlier visited recognized mode of operation or when the control criterion has changed. A performance monitor can take care of restarting the adaptation whenever necessary.

In this paper the supervisory control of mode-switch processes will be exemplified with an experimental flexible beam (Kruise, 1990) which has to transport different payloads with different masses. The main emphasis is on the detection of new and old modes. The paper is organized as follows. In Section 2 the modeling of the flexible beam is presented. In Section 3 a supervisory structure for detection of new and old modes is discussed. In Section 4 a detailed discussion about mode recognition is presented. In Section 5 the results of applying supervisory control to the flexible beam is shown. These results are compared with robust control. Finally in Section 6 conclusions are drawn.

2. FLEXIBLE BEAM

A good example of a mode-switch process is the experimental set-up of a flexible beam. In this experimental set-up the flexible beam has to transport a number of payloads. This process clearly shows the mode-switch behaviour, as the dynamics change with the mass. This implies that changes in dynamics occur only at the instants of a mass change.

2.1. Description of the process

In Fig. 1 the experimental set-up of the flexible beam is shown.

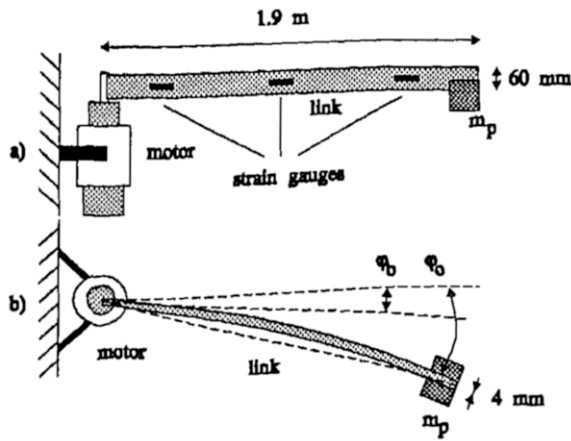


Fig.1 The flexible beam a) side view b) top view

The link rotates in the *horizontal* plane and is free at one end. The other end is clamped to the vertical shaft of a DC-motor. A payload can be attached at the free end of the link. The mass of this payload can be varied between 0 and 0.5 kg. The weight of the beam itself is about 1.2 kg, so the ratio between payload mass and link mass is relatively high compared to other more common robots. Strain gauges are used to measure the bending in the link, and a resolver is used to measure the angle φ_b of the motor axis. By use of these measurements, the tip angle φ_o can be determined.

In a flexible beam with one end mounted to a motor shaft, torsional, longitudinal and transverse vibrations occur. Due to the geometry of the beam (length $l_b = 1.9m$, width $w_b = 4.0mm$ and height $h_b = 60mm$), the torsional and vertical vibrations are small and do not affect the horizontal vibrations

(Kruise, 1990). Therefore, only the transverse vibrations have to be modelled. Kruise (1990) indicated that only the first two transverse vibration modes have to be taken into account to obtain a good model of the process. Kruise showed also that the Coulomb friction can be disregarded for the controller design. In that case, for each payload mass a suitable linear model M_i can be obtained. This linear model of the flexible beam is given by

$$\ddot{q} = (I + F)^{-1} (\alpha_b \ddot{\varphi}_b - R\dot{q} - Wq) \text{ (vibrat. modes)} \quad (2.1a)$$

$$\dot{\varphi}_b = K_a u - \beta_b q \quad \text{(motor)} \quad (2.1b)$$

$$\varphi_o = \varphi_b + (q_1 - q_2)/l_b \quad \text{(tip angle)} \quad (2.1c)$$

where α_b is the coupling vector from base angle to bendings, β_b is the coupling vector from bendings to base angle, φ_b is the base angle, φ_o is the tip angle, F is the coupling matrix between vibration modes, I is the unity matrix, K_a is the motor constant, l_b is the length beam, q is the vector of vibration modes $[q_1 \ q_2]^T$, R is the diagonal damping matrix, u is the control signal, $u \leq 15V$, and W is the diagonal matrix of resonance frequencies. For the description of the various matrices, the reader is referred to Kruise (1990, page 49-50). The matrix F and vectors α_b and β_b are dependent on the payload mass m_p . For instance, when no payload is attached to the tip, the matrix $F=0$. In that case there is no direct coupling between the vibration modes, because R and W are diagonal.

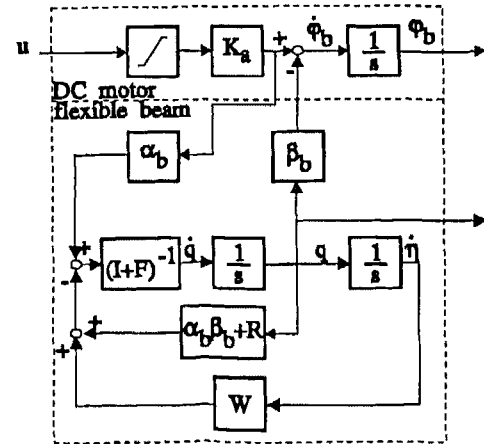


Fig. 2 Block diagram of beam for the new choice of states

The states are given by $x = [\varphi_b \ \dot{q} \ q]^T$. In order to have a correct simulation of the system (2.1), the dynamics of the beam were integrated once. This results in the state-vector $x = [\varphi_b \ q \ \dot{\eta}]^T$, where

$$\dot{\eta} = \int_0^t q(\tau) d\tau \quad (2.2)$$

In Fig. 2 a block diagram of the flexible beam for this choice of the state vector is shown.

3. SUPERVISORY CONTROL

The control objective is to realize a fast settling time and no overshoot of the tip for a limited number of different payloads of different masses. If the payload

mass can be measured or if it is known in advance, by use of a look-up table it can be searched whether this payload has been transported earlier or not. If the payload has not been transported earlier, a model identification cycle can be started and subsequently a controller can be designed. The controller can be stored in a memory together with the payload mass. Subsequently, the controller is installed in the closed loop. If the payload has been transported earlier, the related controller can be retrieved from memory.

When the payload is not known and cannot be measured this gain scheduling approach is inapplicable. Another approach is to detect whether the process behaviour has changed, based on the available signals. If the present behaviour corresponds with the behaviour of one of the models in the memory, the controller related to that model can be retrieved and installed in the closed loop. If the performance of this controller is still insufficient, then probably the mass transported by the beam is new. Hence then learning has to take place. This implies that a model of the beam with this payload mass is constructed and added to the memory. Subsequently a new controller is designed and installed in the closed loop. The monitoring of the closed-loop performance is carried out by a *performance monitor*. Based on the performance criterion stated by the user and on the measured performance, a *supervisor* can detect whether the closed-loop control performance is adequate or not. In the latter case, the supervisor will propose to start a new identification cycle. The introduction of a performance monitor and supervisor results in a third feedback loop, as is shown in Fig. 3. The performance monitor, supervisor, mode detector and memory together form the supervisory structure.

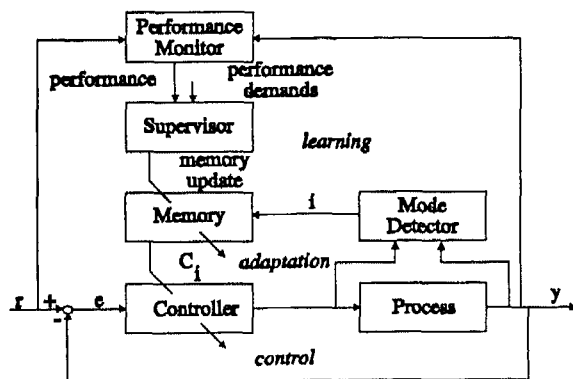


Fig. 3 Control system with supervisory structure

Figure 3 shows that the lower loop is the normal *process feedback loop*, which consists of a feedback controller and the process. The second loop is the *adaptation loop*, which consists of mode detector, memory and controlled system. Based on the mode i detected, controller C_i is selected from the memory and installed in the closed loop. The third feedback loop is the *learning loop*, which consists of performance monitor, supervisor and memory. Based on the measured performance and on the performance de-

mands, the supervisor adds models to the memory or changes existing models.

4. MODE DETECTOR

4.1. Definitions and goals

Mode recognition is the task of identifying the current mode of operation. *Mode-switch detection* is a subsequent task, which establishes whether or not the current process mode differs from the previous process mode. These two tasks are performed by the *mode detector* shown in the supervisory structure of Fig. 3. On the basis of the information provided by the mode detector, the supervisor decides whether the controller parameters should be adapted. If the method is quick in detecting mode switches, then noise may often give rise to wrong detections. On the other hand, if the method is insensitive to noise, mode switches may not be detected fast enough. As both situations are undesirable, a good balance between noise sensitivity and mode tracking is called for.

Finally, when the mode detector establishes a mode switch, the controller parameters should be adjusted. In order to prevent bumps in the control signal, *bumpless transfer* was applied.

In order to apply this idea to the mode recognition problem, a distinction has to be made between *finite mode-switch processes*, i.e. processes which can be described by a finite number of linear models such as the flexible beam, and *generalized mode-switch processes* which can be *approximated* by a finite number of linear models. In this paper we restrict ourselves to finite mode-switch processes. For a description of generalized mode-switch processes is referred to Hilhorst (1992).

4.2. Mode recognition

For a good balance between noise-insensitivity and mode-tracking, effective use can be made of the ideas developed by Fortescue et al. (1981) in the field of least squares estimation. Furthermore, in order to make a good distinction between the models, the models were run in a series-parallel structure with the process. The approach taken has similarities to the approaches taken by Willsky (1976), Isermann (1984), and Tzafestas and Watanabe (1990) in the field of process fault detection. However, those approaches have severe drawbacks as described by Lund et al. (1991) like mode-track stopping and lack of distinction between the different models. These disadvantages have been eliminated in the method proposed here.

If a finite mode-switch process P is in mode i , and if a Gaussian measurement disturbance ψ is present with variance σ_ψ^2 , then the process output y at time instant k can be written as

$$y_k = \hat{y}_k^i + \psi_k \quad (4.1)$$

where \hat{y}_k^i is the output of model M_i . For each model M_i , the error e_k^i between model output \hat{y}_k^i and process output y_k can be determined, i.e.

$$e_k^i = y_k - \hat{y}_k^i \quad (4.2)$$

When the process is in mode i , the error e_k^i is equal to the measurement disturbance, i.e.

$$e_k^i = \psi_k \quad (4.3)$$

The expected norm of the error e_k^i is given by

$$\begin{aligned} E \|e_k^i\|^2 &= E \{(e_k^i)^T (e_k^i)\} \\ &= \begin{cases} \sigma_\psi^2, & \text{if process is in mode } i \\ \sigma_i^2 + \sigma_\psi^2, & \text{otherwise} \end{cases} \end{aligned} \quad (4.4)$$

where σ_i^2 is the additional variance due to the model error between the process and model M_i . Equation (4.4) shows that the expected norm of the error is minimal for the model M_i related to the current process mode i . Hence, one straightforward method for mode recognition is by weighting the sum of errors and to select that model M_i for which the sum is minimal. This can be done by defining a distance $d(P, M_i)$:

$$d(P, M_i) = \Sigma_k^i \quad (4.5)$$

where the weighted sum of squared errors

$$\Sigma_k^i = \lambda_k (\Sigma_{k-1}^i + (e_k^i)^T e_k^i) \quad (4.6)$$

with $0 < \lambda_k \leq 1$ the *forgetting factor*, and $\Sigma_0^i = 0$. The effective number of samples taken into account at an instant k is referred to as the *memory length* χ_k , which is given by

$$\chi_k = \lambda_k (\chi_{k-1} + 1) \quad (4.7)$$

with $\chi_0 = 0$. The expectation of the distance between process P and model M_i at time instant k is given by

$$\begin{aligned} E \{d(P, M_i)\} &= E \{\Sigma_k^i\} \\ &= \begin{cases} \chi_k \sigma_\psi^2, & \text{if process is in mode } i \\ \chi_k (\sigma_i^2 + \sigma_\psi^2), & \text{otherwise} \end{cases} \end{aligned} \quad (4.8)$$

Equation (4.8) shows that the smallest distance $d(P, M_i)$ will converge to $\chi_k \sigma_\psi^2$. Hence, the heuristic idea of Fortescue (1981) can be applied to the mode-recognition problem by keeping the sum Σ_k^i constant for the model with minimal sum Σ_k^i . In the case of weighting both old and new information, the sum Σ_k^i can be kept constant at target Σ_∞ by selecting the forgetting factor

$$\lambda_k = \frac{\Sigma_\infty}{\Sigma_{k-1}^i + (e_k^i)^T (e_k^i)} \quad (4.9)$$

Åström and Wittenmark (1989) motivate that it is required that

$$0 < \lambda_k \leq 1 \quad (4.10)$$

Because $(e_k^i)^T e_k^i \geq 0$, it follows from (4.9) that this condition is fulfilled by $\Sigma_{k-1}^i \geq \Sigma_\infty$ ($1 \leq i \leq n$). In order to have $\Sigma_{k-1}^i \geq \Sigma_\infty$ ($1 \leq i \leq n$), $\Sigma_0^i = 0$ ($1 \leq i \leq n$), and λ_k is taken equal to 1 till $\Sigma_{k-1}^i > \Sigma_\infty$ ($1 \leq i \leq n$).

By the use of the forgetting factor (4.9), the sum Σ_k^i is kept constant. In contrast, the sum Σ_k^j of all other models is variable and may become smaller or larger than the sum Σ_k^i . If a switch from mode i to mode j occurs, then the error e_k^j will increase, and simultaneously the error e_k^i will decrease. Equation (4.9) shows that an increase in error e_k^j results in a small forgetting factor and hence in a small memory length. Due to the drop in memory length and the fact that e_k^j is smaller than e_k^i , the sum Σ_k^j will rapidly decrease and become lower than the sum Σ_k^i . A *mode switch* is detected at the time instant k at which $\Sigma_k^j < \Sigma_k^i$. At that time instant k , model M_j should be selected as the best model, and accordingly the sum Σ_k^j of squared errors should be kept constant. This is performed by taking the new forgetting factor λ_k equal to

$$\lambda_k = \frac{\Sigma_\infty}{\Sigma_{k-1}^j + (e_k^j)^T (e_k^j)} \quad (4.11)$$

This mechanism guarantees an effective balance between mode-tracking and noise-insensitivity.

4.3. Series-parallel structure

Until here the computation of \hat{y}_k^i was not discussed. One method to compute these predictions is to make use of the series-parallel structure. In MRAS literature (Landau, 1979) this structure is called the series-parallel structure, because the reference model (in this case the process model) is placed partly in series with the process and partly in parallel with the process. The advantage of this structure is that drift in the model states can be avoided by regular resetting of the model states to the process state. This is performed by selecting an appropriate observation period T_o . The choice of the observation period T_o will be process-dependent and will therefore be discussed in more detail in section 5.

4.4. Mode-switch and bumpless transfer

Once the mode detector has detected a mode switch, a switch between controllers has to be made. At the instant the parameters of the controller are adapted, a bump in the control signal may be introduced. Bumps in the control signal are unwanted and therefore should be removed. This can be done by applying a *bumpless transfer algorithm*. In this paper bumpless transfer is established by the use of an integrating action with a leakage (Hilhorst et al., 1991b). By taking the leakage time constant (bumpless transfer time constant) τ_b equal to an appropriate value, for instance related to the bandwidth of the control loop, the control signal remains smooth.

5. EXPERIMENTS AND RESULTS

5.1 Experiment design

In order to show the applicability of supervisory control, three experiments with the flexible beam were

carried out. In the first experiment, a step from -90° to 90° was applied as the reference signal ϕ_r for a payload mass $m_p = 0.0 \text{ kg}$. This experiment was repeated for payload masses $m_p = 0.25 \text{ kg}$ and $m_p = 0.5 \text{ kg}$. The control objective is to realize a fast settling time and no overshoot of the tip for each payload mass.

In order to meet this goal for each of the payload masses, the mode centres were selected equal to the payload masses, i.e. $\omega_1 = \{0\}$, $\omega_2 = \{0.25\}$, and $\omega_3 = \{0.5\}$. For each of these mode centers a linear model M_i was obtained by taking the parameter m_p equal to ω_i .

In order to meet the control criterion, Kruse shows that a PID-like controller for the motor axis only is insufficient, and that a state-feedback controller is appropriate. The state-feedback controller is described by

$$u = -K(x - x_r) \quad (5.1)$$

where K is a row vector of control gains, x the process state, and x_r is the desired state. The control gains can, for instance, be found with the pole-placement method.

5.2. Real-time requirements

In order to apply the mode-switch concept to the practical set-up, real-time requirements have to be met. strm and Wittenmark (1989) give a rule of thumb for the selection of the sampling interval T_s . Based on the natural frequency of the dominant closed-loop pole (i.e. 34 rad/s), the sampling interval should be selected between $0.003 \leq T_s \leq 0.015$.

The implementation of supervisory control must be such that these real-time requirements are met. This is performed by exploiting the natural parallelism (Bakkers and Van Amerongen, 1990) in the mode-detection scheme. A PC (286) and a transputer network with four T4's have been used. On each transputer one of the three models was run, and on the fourth transputer the mode-recognition algorithm. By the use of the transputer network a sampling interval $T_s = 0.006 \text{ s}$ could be reached.

5.3. Experimental results

Sufficient distinction between the models could be realized with an observation period T_o of 0.06 s ($= 10 \times T_s$). The asymptotic memory length χ_∞ was selected as 40 samples (that is four times the observation period length T_o). The variance σ_v^2 of the noise was 10^{-4} . As a result the target Σ_∞ was taken equal to 0.004 . At the end of an observation period, the controller related to the model with the smallest distance to the process was installed in the closed loop. Deactivation of the old controller and substitution of the new controller occurred by using the bumpless transfer algorithm described in Section 4. The bumpless transfer parameter τ_b has been selected 0.065 s . This is about twice the inverse of the natural frequency of the fastest closed-loop pole. The results obtained with the mode-switch controller were compared to those of a fixed

controller designed for a payload mass of 0.5 kg , which shows no overshoot for all different payload masses between 0 and 0.5 kg . This controller is referred to as the *robust* controller.

The results obtained from the experiments with payload mass $m_p = 0, 0.25$ and 0.5 kg are shown in Fig. 4 and 5 and 6 respectively. Figure 4 shows that the application of supervisory control results in a shorter rise time and a shorter settling time of the tip angle response compared to the robust controller. This is caused by the fact that the robust controller yields a smaller control signal than the controller C_1 which is optimized for the beam with no payload. Figure 4 also shows that in the beginning (i.e. $0 \leq t \leq 0.25 \text{ s}$) and at the end (i.e. $t \geq 3.0 \text{ s}$) wrong models are selected. Furthermore, after time instant 4 s model M_3 is continuously selected.

The wrong detection at the beginning can be explained by the non-minimum phase behaviour of the different models and the process. Singh (1991) showed that this non-minimum phase behaviour is present in the model related to no payload. He also showed that the non-minimum phase behaviour decreases with increasing mass. However, the real tip response of the beam with no payload shows almost no non-minimum phase behaviour. Therefore, the models related to payload masses 0.25 and 0.5 kg are selected in the beginning.

The wrong model selection at the end can be explained by the Coulomb friction. Figure 4 shows that after time instant 4 s due to a small steady-state error, the control signal u is slightly larger than zero. Because of the presence of the Coulomb friction, a small control signal cannot accelerate the beam, and hence the tip position remains constant. Simultaneously, due to a non-zero control signal, the linear models predict that the link would accelerate. As the predicted amount of acceleration decreases with increasing mass, the tip position of model M_3 related to the largest mass will be closest to the measured tip position. Hence, model M_3 is selected at the end. From this experiment it can be concluded that a good model selection can be made only when the link moves.

Figure 5 shows that compared to the robust controller, the application of the mode-switch controller for the beam with payload mass $m_p = 0.25 \text{ kg}$ results in a shorter rise time and shorter settling time of the tip angle response. Hence, the control performance is improved by supervisory control.

Figure 6 shows that both the control signal and the tip response obtained with supervisory control are equal to the ones obtained with the robust controller.

6. CONCLUSIONS

In this paper the concept of *supervisory control* was the basis for a novel method which can be seen as an attractive alternative for both robust control and conventional adaptive control for *mode-switch processes*. By the use of the mode concept, a controller is obtained which behaves less conservative than a robust controller, and which has the ability to adjust the control

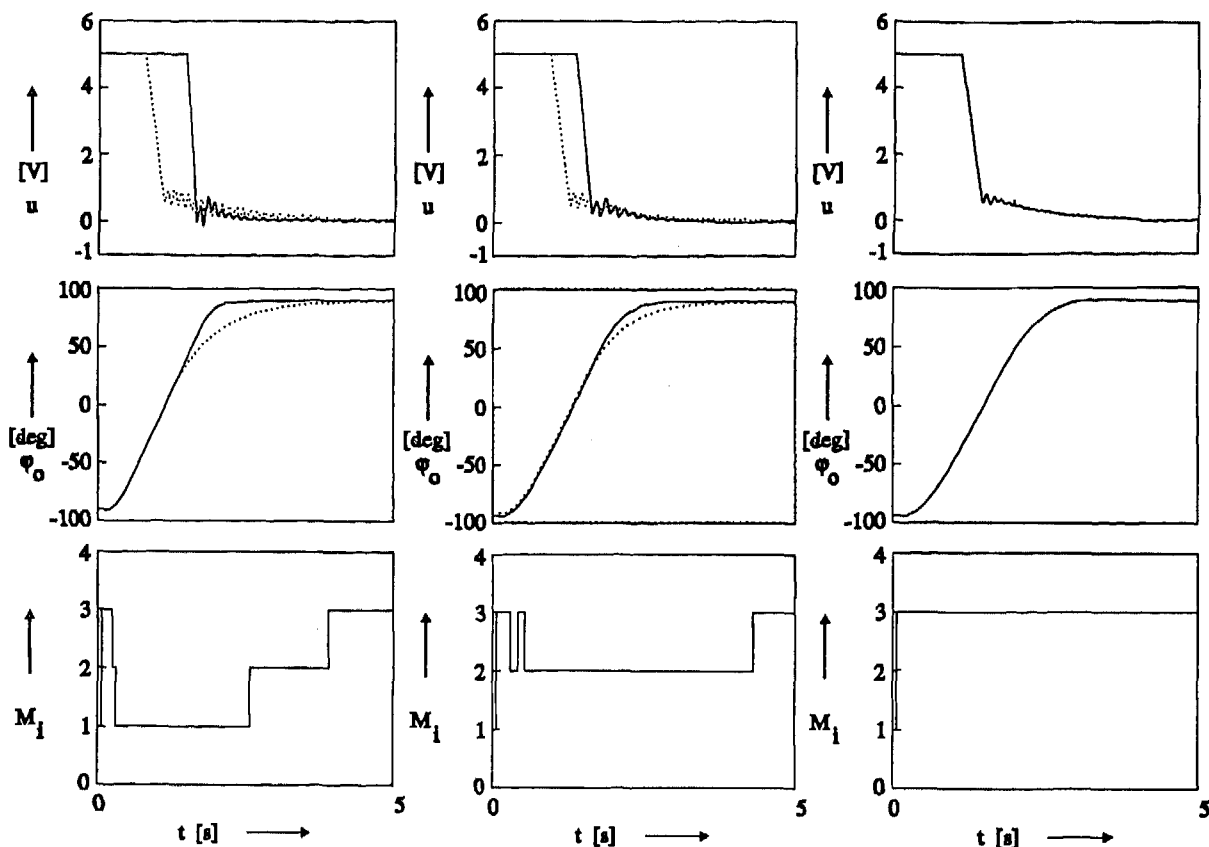


Fig.4 Responses with mass 0 kg Fig. 5 Responses with mass 0.25 kg Fig. 6 Responses with mass 0.5 kg
 u = control signal of supervisory controller (solid) and robust controller (dotted)
 φ_o = tip angle using supervisory controller (solid) and robust controller (dotted)
 M_i = model selected (associated with mode)

parameters fast using process knowledge more effectively than a conventional adaptive controller does. The main problem in this approach is how to detect a mode switch. Various alternatives were considered and tested. The newly developed method of exponential forgetting dedicated for the use on (finite) mode-switch processes has been demonstrated by real experiments.

In this application, supervisory control of the flexible beam results in a better overall performance than what is achieved with a fixed linear controller. That is without overshoot the settling time was smaller and even near minimum time. Problems of measurement noise and Coulomb friction could be solved by adequate tuning of the detection method. The results show that fast adaptation of control parameters can be obtained without deliberately disturbing the process. Therefore, supervisory control should be used as a standard procedure for such processes. Because of the motivation given in Section 1, this conclusion is expected to hold more generally and might extend to the petrochemical processes as well.

REFERENCES

- Åström, K.J. and B. Wittenmark, (1989), Adaptive control, Addison Wesley, New York.
- Bakkers, A.W.P., and J. van Amerongen, (1990), Transputer Based Control of Mechatronic Systems, 11th IFAC World Congress, Tallinn, Estonia, USSR, Vol. 7, 128-133, Pergamon Oxford UK.
- Fortescue, T.R., L.S. Kershenbaum, and B.E. Ydstie, (1981), Implementation of Self-tuning Regulators with Variable Forgetting Factors, *Automatica*, Vol. 17, 831-835.
- Hilhorst, R.A., J. van Amerongen, P. Löhnberg and H.J.A.F. Tulleken, (1991a), Intelligent Adaptive Control of Mode-switch Processes, Proceedings of the IFAC Conference ITAC'91, Vol. 2, Pergamon, Oxford UK.
- Hilhorst, R.A., J. van Amerongen, P. Löhnberg and H.J.A.F. Tulleken, (1991b), Neural Network Based Control of Mode-switch Processes, Proceedings of the IFAC Conference ADCHEM'91, Pergamon, Oxford UK.
- Hilhorst, R.A., (1992), Supervisory Control of Mode-switch Processes, Ph.D. Thesis, University of Twente, Enschede, Netherlands.
- Isermann, R., (1984), Process Fault Detection Based on Modeling and Estimation Methods, A Survey, *Automatica*, Vol. 20, 387-404.
- Kruise, L., (1990), Modeling and Control of a Flexible Beam and Robot Arm, Ph.D. Thesis, University of Twente, Enschede, Netherlands.
- Landau, Y.D., (1979), Adaptive Control, The Model Reference Approach, Marcel Dekker, New York.
- Lund, E.J., J.G. Galchen, and B.A. Foss, (1991), Multiple Model Estimation with Inter-residual Distance Feedback, Proceedings of the 9th IFAC/IFORS symposium on Identification and System Parameter Estimation, July 8-12 Budapest.
- Singh, N., (1991), Control of a Single Flexible Beam based on the Mode-Switch Concept, Report No. 91OR53, Control Laboratory, Control, Systems and Computer Engineering Laboratory, University of Twente, Enschede, Netherlands.
- Tzafestas, S.G., and K. Watanabe, (1990), Modern Approaches to System/Sensor Fault Detection and Diagnosis, *Journal A*, Vol. 31, No.4, 42-57.
- Willsky, A.S., (1976), A Survey of Design Methods for Failure Detection in Dynamic Systems, *Automatica*, Vol. 12, 601-611.