International Journal of Systems Science, 2002, volume 33, number 2, pages 97–105

Robust fault detection based on compensation of the modelling error

G. Klančar†*, Đ. Juričić‡ and R. Karba†

The problem of compensation of modelling errors for the purpose of robust fault detection based on parity relations is addressed. The idea is to approximate unmodelled non-linear dynamics by a neural network model and then to remove the effects of unmodelled dynamics from the primary residuals. The design of such a compensator takes two steps. In the first, a subset of the most informative regressors is selected. The second step entails structure determination and parameter estimation by means of numerical optimization of a criterion function. The criterion reflects a compromise between the quality of approximation and the complexity of the model structure. The results from the study on a three-tank test rig are presented and a comparison between compensated and uncompensated residuals made. It is shown that the compensated residuals represent a good basis for reliable, yet sensitive enough, fault detection and isolation.

1. Introduction

Faults in systems can be viewed as hidden inputs that cannot be directly measured, at least in a vast majority of cases. The best way to assess faults is to check the consistency of data coming from sensing devices. This is done by means of the mathematical model that serves to transform the original process signals into a set of special signals referred to as residuals. Residuals have the unique property of being zero if there is no fault and different from zero as soon as a fault in the system occurs. The problem with this concept is that in practice there are other unanticipated hidden inputs that can affect the residuals in a similar way as faults do. Their origin extends from measurement noise, process disturbances and modelling errors. Consequently, the residuals may be non-zero even if there is no actual fault in the system. This can easily lead to false alarms. Therefore, the key problem in the design of fault detection and isolation (FDI) systems is robustness, i.e. to achieve as high as possible sensitivity to faults and as low as possible sensitivity to other hidden inputs. In further research attention will be focussed on disturbances caused by modelling errors.

Received 6 June 2000. Revised 15 May 2001. Accepted 7 June 2001.

† Faculty of Electrical Engineering, University of Ljubljana, Tržaška 25, 1000 Ljubljana, Slovenia.

‡ Jožef Stefan Institute, Jamova 39, 1000 Ljubljana, Slovenia.

* Author for correspondence. e-mail: gregor.klancar@fe.uni.lj.si

The issue of robustness has received wide attention over the last two decades (Patton 1994). Robustness can be assured using the following two approaches.

The first relies on cautious decision-making about zero/non-zero residual patterns. It is also termed a passive approach and involves various techniques like adaptive thresholds (Hofling and Pfeufer 1994), weighted residuals (Žele and Juričić 1999) and fuzzy decision rules (Schneider 1993).

The second approach to robustness, termed an active approach, focuses on proper residual design. This approach depends on detailed knowledge about distribution matrices, i.e. the way the disturbances affect residuals. If the distribution matrix were known exactly, various decoupling techniques like failure detection filters, unknown input observers and eigenstructure assignment (Patton and Kangethe 1989) would eliminate the effects of uncertainties. Unfortunately, in practice, the distribution matrix is often not precisely known. In such a case, an approximate structure of the uncertainty should be found. A remedy suggested by Patton and Chen (1993) relies on approximating the distribution matrix by a constant such that the changing operating conditions are approximately described in a equally fair manner. However, in the case of modelling errors, the distribution matrix can consist of many, generally time-varying, terms that can make decoupling quite difficult. For the special case of linear models with unmodelled higher frequency dynamics, Patton and Chen (1992) suggested a linear compensator to

International Journal of Systems Science ISSN 0020–7721 print/ISSN 1464–5319 online © 2002 Taylor & Francis Ltd
http://www.tandf.co.uk/journals
DOI: 10.1080/00207720110073172

remove the bias term from the residuals. A similar philosophy has been followed by Zhou and Bennett (1998) for the problem of compensating for the effects of unmodelled non-linearities. They applied a neural network approximator integrated within a conventional linear observer set-up. The key problem they pointed out concerns avoiding incorrect compensation signals during the fault period. Their solution relies on employing different structures in the learning and fault detection stage, which is an important contribution. However, the problem with (closed-loop) observer-based approaches is still the question of trade-off between the sensitivity of the residuals to faults and the dynamics of state reconstruction. Namely, if the eigenvalues of a system's matrix are designed to speed up the reconstruction, then small incipient faults might be left undetected. Therefore, for stable systems, the open-loop parity-space approach can be more appropriate.

A robust open-loop residual generation scheme based on parity relations is addressed here and a solution derived for the problem of compensating modelling errors. Unlike the observer-based (closed-loop) residual generation schemes, it is shown that a unique compensator structure is used in the learning and detection stage. Similarly to Zhou and Bennett (1998), a neural network is used to realize the compensator of unmodelled non-linear effects. The paper is organized as follows. In section 2 the problem formulation is provided. Section 3 discusses the design of a robust residual compensator by means of neural networks. The experimental results obtained on a three-tank system are reviewed in the fourth section. The paper ends with concluding remarks.

2. Problem formulation

Consider a non-linear dynamic system described by the following general form:

$$\boldsymbol{y}(k) = g(\boldsymbol{y}(k-1),\ldots,\boldsymbol{y}(k-d_y),\boldsymbol{u}(k),\ldots,\boldsymbol{u}(k-d_u),\boldsymbol{f}(k),\ldots,\boldsymbol{f}(k-d_f)), \tag{1}$$

where k is current sampling time, $\boldsymbol{y}(k) \in \Re^{n_y}$, $\boldsymbol{u}(k) \in \Re^{n_u}$ and $\boldsymbol{f}(k) \in \Re^{n_f}$ are vectors of process outputs, inputs and faults entry respectively, and n_y, n_u, and n_f are the dimensions of mentioned vectors with d_y, d_u and d_f being the dynamic orders. Furthermore, function $g(\cdot)$ is assumed to be continuously differentiable over the arguments.

Also assume that the effects of faults can be represented by means of additive terms:

$$\boldsymbol{y}(k) = g(\boldsymbol{y}(k-1),\ldots,\boldsymbol{y}(k-d_y),\boldsymbol{u}(k),\ldots,\boldsymbol{u}(k-d_u)) + \sum_{i=1}^{d_f} \boldsymbol{R}_i(k)\boldsymbol{f}(k-i), \tag{2}$$

with

$$\mathbf{y}(k) = [y_1(k) \quad \cdots \quad y_l(k) \quad \cdots \quad y_{n_y}(k)]^{\mathrm{T}},$$
$$\mathbf{u}(k) = [u_1(k) \quad \cdots \quad u_l(k) \quad \cdots \quad u_{n_u}(k)]^{\mathrm{T}},$$

where $\boldsymbol{R}_i$ is the fault distribution matrix and $f(k)$ is a vector of faults, which is an unknown function of time. During normal process behaviour $f(k) = 0$. Matrices $\boldsymbol{R}_i$ are assumed to contain a fixed configuration of zero terms.

The general Nonlinear AutoRegressive model with eXogenous inputs (NARX) process model (2) can be represented in various forms. In our case the most convenient is the following polynomial form (Henson and Seborg 1997):

$$\begin{aligned} y_l(k) = {} & \sum_{i=1}^{n_y\cdot d_y} \theta^1_{(l)i}\mathbf{Y}_k(i) + \sum_{i=1}^{n_u\cdot d_u} \theta^2_{(l)i}\mathbf{U}_k(i) \\ & + \sum_{i=1}^{n_y\cdot d_y}\sum_{j=1}^{i} \theta^3_{(l)ij}\mathbf{Y}_k(i)\mathbf{Y}_k(j) \\ & + \sum_{i=1}^{n_y\cdot d_y}\sum_{j=1}^{n_u\cdot d_u} \theta^4_{(l)ij}\mathbf{Y}_k(i)\mathbf{U}_k(j) \\ & + \sum_{i=1}^{n_u\cdot d_u}\sum_{j=1}^{i} \theta^5_{(l)ij}\mathbf{U}_k(i)\mathbf{U}_k(j) \\ & + \cdots + \sum_{i=1}^{d_f} \mathbf{R}_{il}(k)\mathbf{f}(k-i), \end{aligned} \tag{3}$$

where notation $\boldsymbol{Y}_k(i)$ means ith element of vector $\boldsymbol{Y}_k$ and

$$\begin{aligned} \mathbf{Y}_k = \lfloor & y_1(k-1) \ \cdots \ y_1(k-d_y) \ \cdots \ y_l(k-1) \\ & \cdots \ y_l(k-d_y) \ \cdots \ y_{n_y}(k-1) \ \cdots \ y_{n_y}(k-d_y) \rfloor \\ \mathbf{U}_k = [& u_1(k-1) \ \cdots \ y_1(k-d_u) \ \cdots \ u_l(k-1) \\ & \cdots \ u_l(k-d_u) \ \cdots \ u_{n_u}(k-1) \ \cdots \ u_{n_u}(k-d_u)]. \end{aligned}$$

Here $\theta^m_{(l)ij} \in \Re$ is the parameter accompanying the mth term in expansion, $\boldsymbol{U}_k$ and $\boldsymbol{Y}_k$ are the vectors of process inputs and outputs and also their delayed values.

The nominal model of the plant is usually not as detailed as is suggested in (3) but rather it tends to be as simple as possible. Moreover, in many realistic applications, simple models are good enough to meet user requirements. As the resulting model is only an approximate description of the process, the unmodelled effects, such as higher dynamic terms and non-linearities, remain so that their influence should be suppressed.

Without loss of generality, it will be assumed in the further research that the nominal behaviour of the process can be described by means of a stable multivariate ARX model:

$$\boldsymbol{y}(k) = \boldsymbol{A}_1\boldsymbol{y}(k-1) + \cdots + \boldsymbol{A}_{d_y}\boldsymbol{y}(k-d_y) + \boldsymbol{B}_0\boldsymbol{u}(k) + \cdots + \boldsymbol{B}_{d_u}\boldsymbol{u}(k-d_u). \quad (4)$$

Model (4) is widely applied in practice. It can be obtained either by linearization around an operating point or directly by identification from past data records.

On the basis of model (4), it is easy to design the set of primary residuals $\boldsymbol{r}(k)$ (Gertler 1998) as follows:

$$\begin{aligned}\boldsymbol{r}(k) = \boldsymbol{y}(k) - \hat{\boldsymbol{y}}(k) &= \boldsymbol{y}(k) - \boldsymbol{A}_1\hat{\boldsymbol{y}}(k-1) \\ &\quad - \cdots - \boldsymbol{A}_{d_y}\hat{\boldsymbol{y}}(k-d_y) - \boldsymbol{B}_0\boldsymbol{u}(k) \\ &\quad - \cdots - \boldsymbol{B}_{d_u}\boldsymbol{u}(k-d_u), \end{aligned} \quad (5)$$

where $\boldsymbol{r}(k) = \lfloor r_1(k) \ \cdots \ r_l(k) \ \cdots \ r_{n_y}(k) \rfloor^{\mathrm{T}}$.

Expression (5) represents the so-called computational form of residuals and $\hat{\boldsymbol{y}}(k)$ represents outputs of the model. Supposing that the linear model is the same or at least similar to the first part of equation (3) (ideal case) then residuals in normal process behaviour appear only due to higher terms in expression (3). The internal form resulting from (3) and (5) reads:

$$\begin{aligned} r_l(k) = &\sum_{i=1}^{n_y \cdot d_y} \tilde{\theta}^1_{(l)i}\mathbf{Y}_k(i) + \sum_{i=1}^{n_u \cdot d_u} \tilde{\theta}^2_{(l)i}\mathbf{U}_k(i) \\ &+ \sum_{i=1}^{n_y \cdot d_y}\sum_{j=1}^{i} \theta^3_{(l)ij}\mathbf{Y}_k(i)\mathbf{Y}_k(j) \\ &+ \sum_{i=1}^{n_y \cdot d_y}\sum_{j=1}^{n_u \cdot d_u} \theta^4_{(l)ij}\mathbf{Y}_k(i)\mathbf{U}_k(j) \\ &+ \sum_{i=1}^{n_u \cdot d_u}\sum_{j=1}^{i} \theta^5_{(l)ij}\mathbf{U}_k(i)\mathbf{U}_k(j) \\ &+ \cdots + \sum_{i=1}^{d_f} \mathbf{R}_{il}(k)\mathbf{f}(k-i). \end{aligned} \quad (6)$$

The polynomial description (6) can be represented in compact form:

$$\boldsymbol{r}(k) = \gamma(\boldsymbol{y}(k-1), \ldots, \boldsymbol{y}(k-d_y), \boldsymbol{u}(k), \ldots, \boldsymbol{u}(k-d_u)) + \sum_{i=1}^{d_f} \boldsymbol{R}_{il}(k)\boldsymbol{f}(k-i). \quad (7)$$

Obviously, due to unmodelled dynamics, residuals are never equal to zero, even during fault-free operation. Now focus on fault-free operation, i.e. $\boldsymbol{f}(i) \equiv 0$.

The process outputs can be written as:

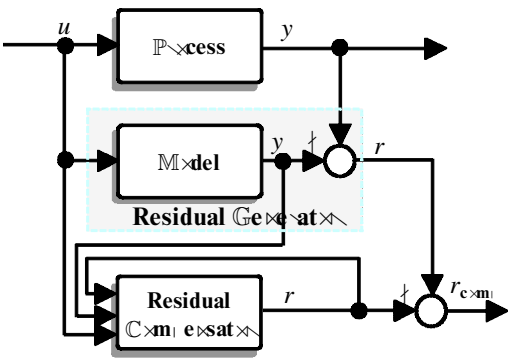


Figure 1. Robust residual design based on the compensator of unmodelled non-linear dynamics.

$$\boldsymbol{y}(k) = \hat{\boldsymbol{y}}(k) + \boldsymbol{r}(k). \quad (8)$$

If expression (8) is substituted in (7) for $\boldsymbol{y}(k-1)\cdots\boldsymbol{y}(k-d_y)$ the following relationships are obtained

$$\boldsymbol{r}(k) = \varphi(\boldsymbol{r}(k-1), \ldots, \boldsymbol{r}(k-d_y), \hat{\boldsymbol{y}}(k-1), \ldots, \hat{\boldsymbol{y}}(k-d_y), \boldsymbol{u}(k), \ldots, \boldsymbol{u}(k-d_u)). \quad (9)$$

To minimize the effects of modelling errors on residuals, the relationships in expression (9) can be approximated by an additional non-linear model. Such a model should rely on expression (9) rather than (7) because in the faulty mode the process output is contaminated by fault entry. Fault modelling is, however, not considered in the design stage. The problem is thus only partly solved because residuals $\boldsymbol{r}(k-i)$ are still influenced by faults. The complete remedy is achieved by using the revised relationship (9), i.e.

$$\hat{\boldsymbol{r}}(k) = \varphi(\hat{\boldsymbol{r}}(k-1), \ldots, \hat{\boldsymbol{r}}(k-d_y), \hat{\boldsymbol{y}}(k-1), \ldots, \hat{\boldsymbol{y}}(k-d_y), \boldsymbol{u}(k), \ldots, \boldsymbol{u}(k-d_u)). \quad (10)$$

The difference between (9) and (10) is only in the type of prediction. In the first case, we have one-step ahead prediction while in the second case, the prediction is k-step ahead. Generally, the second case results in higher prediction error variance than the first case, but that is the price to be paid for getting rid of the influence of faults. The configuration of the residual compensator is shown in figure 1.

3. Design of the robust residual compensator

The structure of residual compensator (10) can be designed by using different approaches such as neural networks, NARX polynomial models, fuzzy models and local linear model trees (Nelles 1996). Generally, the choice of approximator depends on the extent of available *a priori* knowledge. Owing to the fact that in our

case no *a priori* assumption about model structure could be made, Artificial Neural Networks (ANN) were employed. In particular, the perceptron with one hidden layer is used.

The problem of approximation can be stated as the problem of defining the perceptron Nonlinear Output Error (NOE) structure $\hat{\varphi}_{\mathrm{ANN}}$ which approximates the expression (10) such that:

$$\boldsymbol{r}(k) \approx \hat{\boldsymbol{r}}(k) = \hat{\varphi}_{\mathrm{ANN}}([\hat{\boldsymbol{r}}(k-1), \ldots, \hat{\boldsymbol{r}}(k-d_y), \hat{\boldsymbol{y}}(k-1), \ldots, \hat{\boldsymbol{y}}(k-d_y), \boldsymbol{u}(k), \ldots, \boldsymbol{u}(k-d_u)], \hat{\boldsymbol{\theta}}), \tag{11}$$

where $\hat{\boldsymbol{\theta}}$ is the vector of parameters of the network (weights and biases) and $[\cdot]$ is the vector containing the regressors.

The design of ANN (11) consists of two main steps. In the first, the set of inputs into the network must be defined. In the second, it is first necessary to define the number of hidden neurons in the perceptron structure. Then, from training data, the unknown parameters of the model have to be estimated by the aid of optimization techniques.

3.1. *First step: optimal choice of regressors*

The optimal choice of regressors is one of the key problems in neural network design. Operating with the set of all possible regressors is inconvenient. Therefore, it is better to choose a subset of most significant ones that carry the bulk of information required to describe the process output. Thus, the well-known problem of overfitting and accompanying numerical inconveniences originating from ill-conditioned optimization problems can be avoided. Three types of approaches are roughly distinguished.

The first type is based on *ad hoc* selection of regressors. This procedure is typically repeated until the underlying network yields satisfactory quality of fit to the process output.

The second type relies on the use of the techniques of non-linear statistical analysis such as non-linear principal component analysis (Dong and McAvoy 1996). Since they are mainly realized by neural networks, they are time-consuming and the contribution of each input variable to principal components is difficult to evaluate (Atsushi and Stephen 1997).

The third type, however, uses linear techniques. The resulting selection of regressors is suboptimal but the required computational load is significantly lower than in the previous two approaches. Here, it is simply assumed that an input that has a significant non-linear effect on output in fact has a significant linear effect on output. A systematic procedure for eliminating unnecessary data by examining the dynamic characteristics of cause and effect relationship was recently suggested by Atsushi and Stephen (1997). Their idea is followed in the approach below.

The procedure is based on linear principal component analysis. First, take the training set consisting of the matrix of recorded inputs $\boldsymbol{U} = [\boldsymbol{u}_1, \boldsymbol{u}_2, \ldots, \boldsymbol{u}_{n_u}]$ and matrix of recorded outputs $\boldsymbol{Y} = [\boldsymbol{y}_1, \boldsymbol{y}_2, \ldots, \boldsymbol{y}_{n_y}]$, where $\boldsymbol{u}_i = [u_i(k), \ldots, u_i(k+N)]^{\mathrm{T}}$, $\boldsymbol{y}_j = [y_j(k), \ldots, y_j(k+N)]^{\mathrm{T}}$ and N is the number of samples defined by the history of records.

In the first step matrix $\boldsymbol{\Phi}$ is composed from matrices $\boldsymbol{Y}$ and $\boldsymbol{U}$ as $\boldsymbol{\Phi} = \boldsymbol{Y}^{\mathrm{T}}\boldsymbol{U}$. Then singular value decomposition is applied to $\boldsymbol{\Phi}$, resulting in $\boldsymbol{\Phi} = \boldsymbol{W}\boldsymbol{S}\boldsymbol{V}^{\mathrm{T}}$, where $\boldsymbol{W}$ and $\boldsymbol{V}$ are unitary matrices and $\boldsymbol{S}$ is a diagonal matrix containing singular values. According to Turner *et al.* (1996), the input with index λ is considered the most significant, with λ being the biggest element of the first column of matrix $\boldsymbol{V} = [\boldsymbol{v}_{ij}]$ which means $\max\{v_{11}, v_{21}, \ldots, v_{n_u 1}\} = v_{\lambda 1}$.

To determine the second most significant input, the input $\boldsymbol{u}_\lambda$ selected in the first step is removed from $\boldsymbol{U}$. The contribution of the selected input $\boldsymbol{u}_\lambda$ to the outputs $\boldsymbol{Y}$ is removed by linear regression $\boldsymbol{Y}$ $\boldsymbol{Y} - \boldsymbol{u}_\lambda \cdot \boldsymbol{P}^{\mathrm{T}}$, where $\boldsymbol{P}$ is the vector of estimated linear parameters. To find the next important candidates the procedure is repeated in the same manner.

3.2. *Second step: ANN training*

When regressors are sorted out by significance, the main part of a usually heuristically treated problem is solved. For selection of those regressors that essentially contribute to the quality of fit, the simple and most effective procedure of experimenting should be employed. This is achieved by gradually extending the set of ANN inputs by regressors according to the significance deduced above. The addition of regressors is terminated when the trained ANN does not allow visible improvement.

If the resulting static ANN fails to yield the expected quality of fit, the dynamic recurrent ANN structure (11) is to be used. Here external time delay units are used to learn the system dynamics. The number of lags, i.e. order of the model, can either be determined by a trial and error approach or by using *a priori* system knowledge.

The number of hidden sigmoid neurons should be determined gradually. First, a network with one hidden neuron is trained and validated. Then, in each step, the number of hidden neurons is increased until the performance on the validation data set achieves the optimum.

The ANN structure could also be determined by iterative procedures for ANN structure optimisation. These procedures try to automate the usually manual search of

the optimal ANN structure by the use of pruning and constructive algorithms (Kwon and Yeung 1997).

The parameters of the ANN NOE structure are determined as the minimum of the criterion function defined on a training set:

$$\hat{\boldsymbol{\theta}} = \arg\min_{\theta} \sum_{k=1}^{N} (\boldsymbol{r}(k) - \hat{\boldsymbol{r}}(k))^2. \qquad (12)$$

For that purpose the Levenberg–Marquardt training algorithm (Norgaard 1997) was used. Optimization of parameters is performed in two steps. In the first, the structure in figure 2a is applied where real past residuals r enter the NARX model. This step is important in finding the initial parameter guesses. In the second, the training procedure proceeds on model (11), which uses past estimated residuals $\hat{r}$ (NOE model). The main reasons for this are (Norgaard *et al*. 2000):

- models with feedback (non-linear OE models, recurrent networks) tend to have more local minima than the NARX models; hence, without good initialization, it is likely that one will get stuck in a poor minimum; and
- the initial parameters obtained by optimizing an open loop model (one step ahead prediction; figure 2a) guarantee a convergence of the optimisation run in figure 2b.

Learning starts from randomly chosen small weights that are adapted during the training. Therefore, the procedure should be repeated several times to obtain the best results.

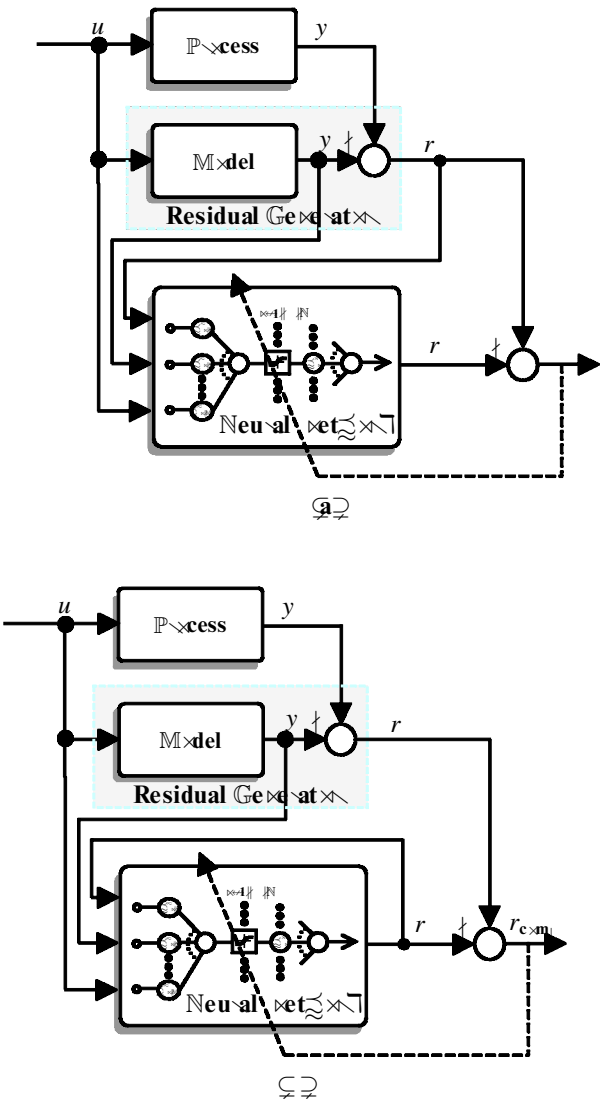


Figure 2. ANN training: (a) initialization phase and (b) refinement phase.

4. Application to a three-tank system

The procedure for robust residual design was applied to a three-tank test rig. Its structure is depicted in figure 3. It consists of three tanks, T_1, T_2 and T_3, connected with flow paths, which serve to supply water from main reservoir T_0. There are two active flow paths available. In the first, the flow is generated by varying the angular speed of pump P_1. In the second case, pump P_2 works at constant speed while the flow is manipulated by the valve V_5. There are two servo-valves in the plant, i.e. V_4 and V_5, driven by DC motors. Valves V_1 and V_2 are on-off while V_3 is a manual valve. The purpose of the latter is mainly to emulate 'real' faults, i.e. leakage of tank T_1. A detailed description of the process (Juričić *et al*. 1997) along with complete simulator in Simulink is available at http:// www - e2.ijs.si / Topics / Projects / Copernicus / 3tank.html. The simulator is mentioned as it is easier to experiment with, but it should be emphasized that the results presented here were obtained from the rig itself.

In the present study, tanks T_1 and T_3 take over the role of buffers for supplying T_2. Contents from T_1 and T_3 are 'mixed' in T_2 and then fed back to the reservoir

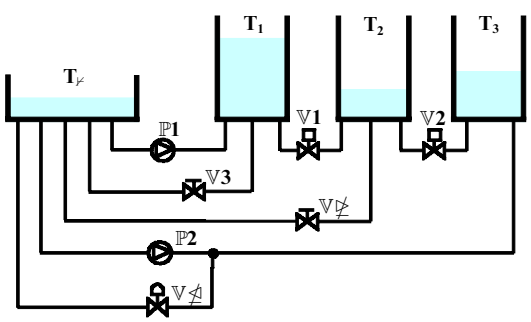


Figure 3. Process structure.

T_0. The level in tank T_1 is controlled by manipulating the reference speed of pump P_1 while the level in T_3 is controlled by manipulating the command signal of valve V_5. The valves V_1 ,V_2 and V_4 are normally open, while V_3 is normally closed. By changing these normal settings and by adding bias to level sensors, various faults can easily be emulated.

The described process has two inputs $\boldsymbol{u}(t) = [\omega\ s]^{\mathrm{T}}$ (speed ω of pump P_1 and position s of servo valve V_5) and three outputs $\boldsymbol{y}(k) = [h_1 h_2 h_3]^{\mathrm{T}}$ in tanks T_1, T_2 and T_3 respectively. The linearized nominal model obtained at a certain operating point is described by the following first-order system:

$$\hat{\boldsymbol{y}}(k+1) = \begin{bmatrix} 0.9718 & 0.0107 & 0.0001 \\ 0.0107 & 0.9056 & 0.0153 \\ 0.0001 & 0.0153 & 0.9640 \end{bmatrix} \cdot \hat{\boldsymbol{y}}(k) + \begin{bmatrix} 0.4946 & 0 \\ 0.0027 & 0.0003 \\ 0 & 0.0397 \end{bmatrix} \cdot \boldsymbol{u}(k). \quad (13)$$

The system is driven in closed loop with two PI controllers which take care of the levels h_1 and h_3. Thus the effects of faults are seen only on process inputs.

The structure of the residual compensator is represented with three MISO recurrent neural networks (figure 4). Inputs to each of them are determined by the selection procedure represented above. The initial set of regressors consists of inputs to the system and model outputs only. The regressors obtained, sorted by significance, are listed in table 1.

Here $\hat{h}_i$ denotes the ith output of the linear model (13). The elements of the rows of table 1 appear as candidate inputs into the ANN. The selection of the important subset of regressors from table 1 is done stepwise. For example, the first ANN starts with $\hat{h}_1$ as the only input. In the next step, two inputs are applied, i.e. $\hat{h}_1$ and ω. If the quality of fit has improved, the third input is added, i.e. $\hat{h}_1$, ω and $\hat{h}_2$ and so on until no progress in the quality of fit is achieved any more. The final subset of regressors for each static ANN is given in table 2.

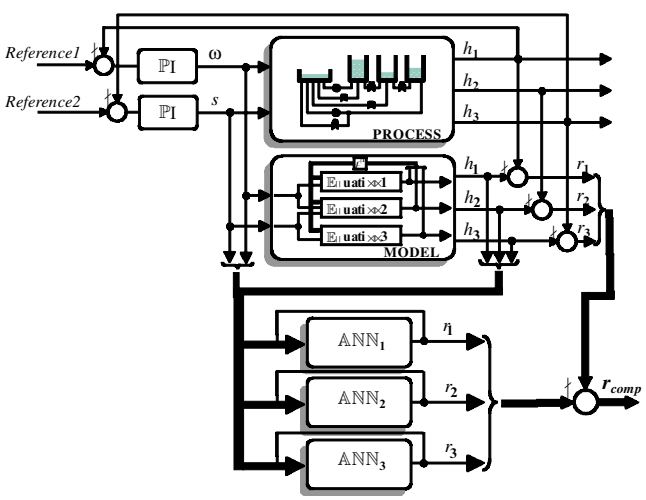


Figure 4. Robust residual design implemented on the three-tank system.

Table 1. Regressors sorted by significance.

Output	Sequence of regressors sorted by significance 1	2	3	4	5
r_1	$\hat{h}_1$	ω	$\hat{h}_2$	$\hat{h}_3$	s
r_2	$\hat{h}_1$	$\hat{h}_3$	s	ω	$\hat{h}_2$
r_3	$\hat{h}_3$	s	$\hat{h}_2$	$\hat{h}_1$	ω

After the regressors are determined, the next step is to choose the proper order of a neural network structure which can either be done by trial and error or by using *a priori* knowledge of the system. The second-order recurrent ANN structure was used in our case. Further increasing the order, i.e. the number of variables lags, is not reasonable as the trade-off between quality of fit and structure complexity becomes less acceptable. The number of hidden neurons was determined according to the success of the ANN training as mentioned above, thus there were five neurons for ANN_1 and ANN_2, and four of them for ANN_3.

The set of training data from normal system operation was obtained by changing the reference values of each controlled tank pulsewise with different durations and different magnitudes covering the whole operating region. Thus, the process dynamics and non-linearity are well described by the training data.

The progress of training can be seen in figure 5. As can be seen, the networks are trained in less than 10 iterations where initialization phase of the training is not included.

Once the networks were trained, a set of validation data from normal system operation was used to check the adequacy of the proposed approach. The set was collected from a smaller region and different from the training set. Figure 6 shows the process inputs and out-

Table 2. Chosen regressors used as inputs to static ANN.

Output	Sequence of final subset of regressors output 1	2	3	4	5
r_1	ω	$\hat{h}_1$	$\hat{h}_2$	/	/
r_2	ω	s	$\hat{h}_1$	$\hat{h}_3$	/
r_3	s	$\hat{h}_3$	/	/	/

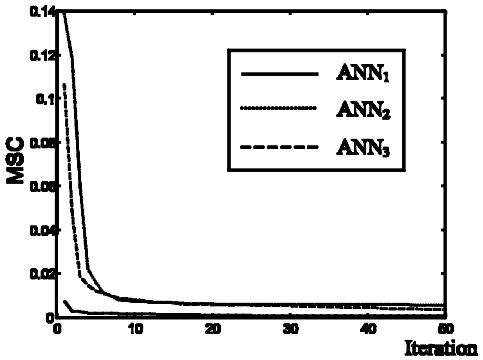
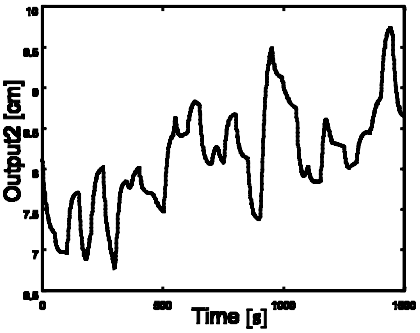


Figure 5. Mean square error criterion (MSE) of training.



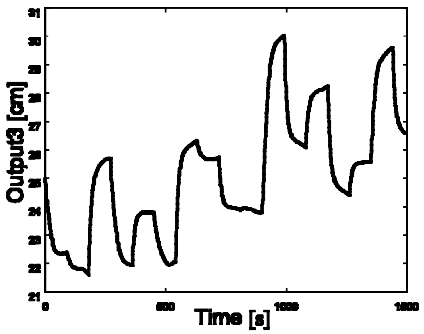
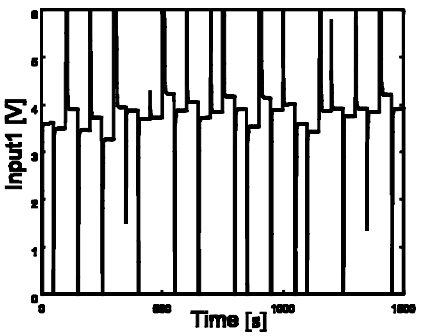
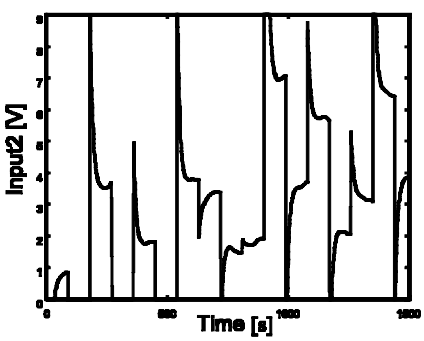
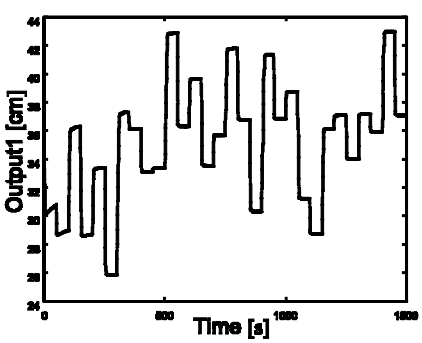


Figure 6. Inputs and outputs of the validation data.







puts while the resulted uncompensated and compensated residuals are shown in figure 7.

Comparison of compensated and non-compensated residuals shows that considerable improvement is achieved as thresholds needed for uncompensated residuals can be decreased by at least three times in order that smaller faults can easily be detected.

In figure 8, the system is subjected to a small fault-leakage of the first tank where the period of fault is marked by a shaded area. The fault affects only the first two residuals (compare figures 7 and 8) while the third remains unaffected. Compensated residuals are close to zero and when the fault occurs, they have a similar average magnitude to the non-compensated ones. Thus, as the fault detection scheme is made robust to modelling errors, the scheme can detect incipient faults although modelling uncertainties are present. From figure 8, it can be seen that compensated residuals have a significant decrease and small thresholds can easily be applied to the residual signals to judge about zero/non-zero patterns. Just from observing the uncompensated primary residuals, it is not clear whether the fault occurs in the system or not.

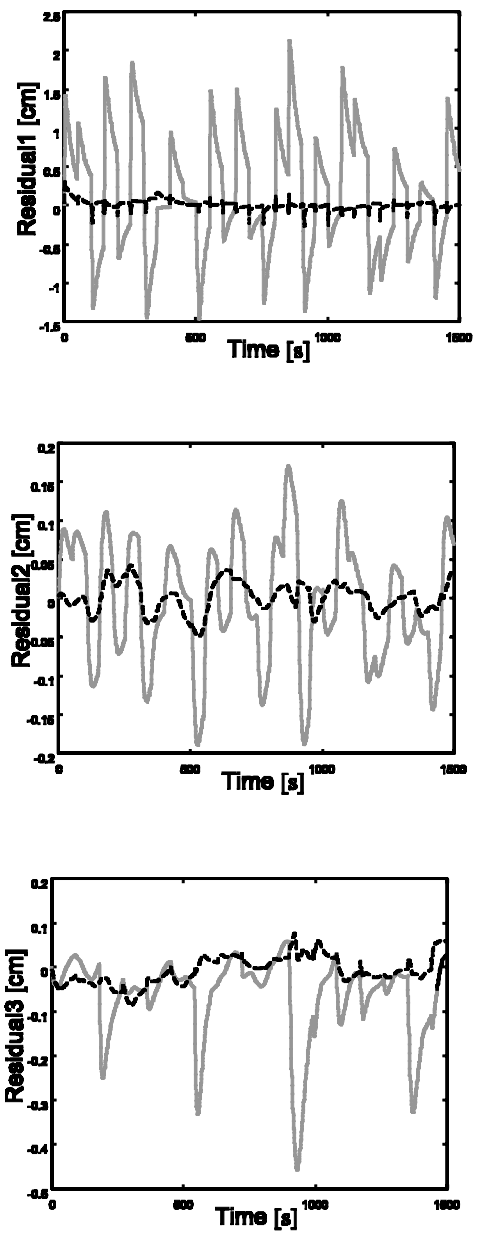


Figure 7. Uncompensated (solid) and compensated (dashed) residuals during normal operation.

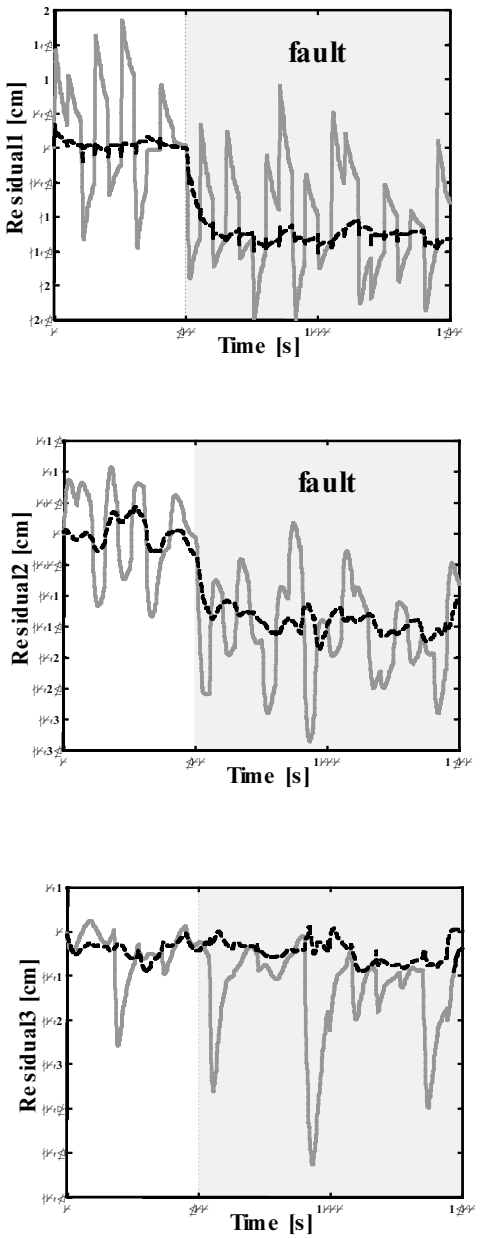


Figure 8. Uncompensated (solid) and compensated (dashed) residuals before and after (shaded area) the fault (leakage from first tank) occurs.

5. Conclusions

An approach to assuring robustness in model-based fault detection, in spite of the presence of unmodelled higher dynamics and unmodelled nonlinearities, is presented. The idea relies on designing an additional compensator structure, that is used to decouple the effects caused by uncertainties from residuals. The suggested procedure combines classical modelling with modelling based on a learning approach resulting in a hybrid model structure. The compensator approximates the structure of modelling uncertainties by developing a neural model, where particular consideration is given to avoiding the generation of an incorrect compensation signal during a fault period. The idea to choose the minimal number of inputs into the neural network by selecting only the most informative ones has proved useful.

The proposed scheme has been applied to the laboratory test rig and the results show that incipient faults can be detected and isolated correctly from compensated residuals while this could not have been done by analysing the non-compensated ones. It is evident that at

the same degree of robustness the sensitivity to faults of the proposed fault detection scheme can be much increased. However, further modelling effort to achieve more efficient or even perfect disturbance decoupling is not reasonable due to the fact that the complicated structure required for a hardly noticeable improvement is rarely justifiable. As an ANN is a relatively complicated model representation of the modelling error, it is of considerable practical importance to search for a simpler representation of expression (6) as, for example, statistical correlation methods. This will be addressed in further research.

Acknowledgements

The Slovenian Ministry of Science and Technology is acknowledged for supporting the work. The authors are grateful to Igor Škrjanc and Magnus Norgaard for helpful comments.

References

Atsushi, A., and Stephen, P. K., 1997, Nonlinear modelling of LPG %C5 content of catalytic reformer debutaniser column. *Computers in Chemical Engineering*, **21,** 1155–1160.

Dong, D., and McAvoY, T. J., 1996, Nonlinear principal component analysis based on principal curves and neural networks. *Computers in Chemical Engineering*, **20,** 65–78.

Gertler, J., 1998, *Fault Detection and Diagnosis in Engineering Systems* (New York: Marcel Dekker).

Henson, M. A., and Seborg, D. E., 1997, *Nonlinear Process Control* (New York: Prentice-Hall).

Hofling, T., and Pfeufer, T., 1994, Detection of additive and multiplicative faults-parity space vs. parameter estimation. *Preprints of IFAC Symposiumon Fault Detection, Supervision and Safety for Technical Processes: SAFEPROCESS'94*, Espoo, Finland, vol. 2, pp. 539–544.

Juričić, D., Dolanc, G., and Rakar, A., 1997, Semi-physical modelling of a FDI benchmark process. *Proceedings of the 2nd MATHMOD, IMACS Symposium on Mathematical Modelling*, Vienna, Austria, pp. 815–820.

Kwon, T., and Yeung, D., 1997, Constructive algoritms for structure learning in feedforward neural networks for regression problems. *IEEE Transactions on Neural Networks*, **8,** 630–645.

Nelles, O., 1996, Local linear model trees for online identification of time-variant nonlinear dynamic systems. *Artificial Neural Networks —ICANN '96, International Conference Proceedings* (Berlin: Springer), pp. 115–120.

Norgaard, M., 1997, *Neural Network Based System Identification Toolbox*. (Lyngby: Technical University of Denmark).

Norgaard, M., Ravn, O., Poulsen, N. K., and Hansen, L. K., 2000, *Neural Networks for Modelling and Control of Dynamic Systems* (London: Springer).

Patton, R. J., 1994, Robust model-based fault diagnosis: the state of the art. *Preprints of IFAC Symposium on Fault Detection, Supervision and Safety for Technical Processes: SAFEPROCESS'94*, Espoo, Finland, vol. 1, pp. 1–27.

Patton, R. J., and Chen J., 1992, On-line residual compensation in robust fault diagnosis of dynamic systems. *Preprints of IFAC Symposium on Artificial Intelligence in Real-Time Control* (Oxford: Pergamon), pp. 221–227.

Patton, R. J., and Chen, J., 1993, Optimal unknown input distribution matrix selection in robust fault diagnosis. *Automatica*, **29,** 837–841.

Patton, R. J., and Kangethe, S. M., 1989, Robust fault diagnosis using eigenstructure assignment of observers. In R. Patton, P. Frank and R. Clark (eds.) *Fault Diagnosis in Dynamic Systems: Theory and Application* (New York: Prentice-Hall), pp. 99–154.

Schneider, H., 1993, Implementation of a fuzzy concept for supervision and fault detection of robots. *Preprints EUFIT'93, 1st European Congress on Fuzzy and Intelligent Technologies*, Aachen, Germany, vol. 2, pp. 775–780.

Turner, P., Montague, G., and Morris, J., 1996, Dynamical neural networks in nonlinear predictive control (an industrial application). *Computers in Chemical Engineering*, **20,** 937–942.

Žele, M., and Juričić, D., 1999, Monitoring the control loop on the basis of a reduced process model. In *Modern Applied Mathematics Techniques in Circuits, Systems and Control* edited by N. E. Mastorakis (World Scientific and Engineering Society Press) (Athens: Greece), pp. 3391–3396.

Zhou, J., and Bennett, S., 1998, Adaptive error compensation for robust fault detection. *International Journal of Systems Science*, **29,** 57–64.

http://www-e2.ijs.si/Topics/Projects/Copernicus/3tank.html