15th Triennial World Congress, Barcelona, Spain

NEURAL NETWORK BASED TIME-DELAY ESTIMATION FOR NONLINEAR DYNAMIC SYSTEMS

Yonghong Tan, Chun-Yi Su[+], Naz Karim[++]

Lab. of Intelligent Systems & Control Engineering
Guilin University of Electronic Technology, 541004 Guilin, China
+Dept. of Mechanical Eng., Concordia University, Montreal, Canada
++Dept. of Chemical Eng., Colorado State University, Fort Collins, USA

Abstract: The estimation for the nonlinear dynamic system with time-varying input time-delay is an important issue for system identification. In order to estimate the dynamics of the process, a dynamic neural network with external recurrent structure is applied to the modelling procedure. In the case where time-delay is time varying, a useful way is to develop on-line time-delay estimation mechanisms to track the input time-delay variation. In this paper, two schemes respectively called direct as well as indirect time-delay estimators are proposed. Finally, two numerical examples are illustrated for the test of the proposed methods. *Copyright © 2002 IFAC*

Keywords: Estimation, time-delay, nonlinear systems, neural networks.

1. INTRODUCTION

It is known that many industrial processes involve input time-delays. Therefore, the identification of the process input time-delay is one of the important issues in the domain of system modelling and identification. Some literatures can be found on time-delay estimation. Reed et. al.(1981) applied LMS algorithm to locate the cross-correlation function so as to estimate the time-delay between input/output signals. Teng and Sirisena (1988) proposed an approach to extend the order of the numerator polynomial function for time-delay estimation. Lim and Macleod (1995) proposed an adaptive time-delay tracking method for IIR filter. Balestrino et. al.(1998) proposed a strategy for steady state time-delay estimation. However, almost all of these approaches are only available for linear systems. It is noted that most industrial systems more or less contain not only time-delay but also non-linearity. Hence, if the non-linearity of the process is significant, it will be necessary to develop the approaches for the modeling of nonlinear processes with time-delay.

During the recent decade, neural networks have been proved to be useful for system modeling and function approximation. In this paper, the dynamic neural network where the input layer has the external recurrent connection with the output of the network is used to model the dynamic nonlinear process. It is noted that the time-delay in some industrial processes may be varying with time. For example, the inlet flow rate, the manipulated variable of a continuous stirred tank reactor, may change with time, it thus causes the variations of the manipulating time-delay.

In this case, the on-line time-delay estimation is necessary if the effect of those variations can not be ignored. In this paper, two neural network based methods for the modelling of a class of nonlinear process with input time-delay is proposed. The first one is called as indirect time-delay estimation method. In this method, the criterion is minimized with respect to the estimated time-delay that is contained in the neural network based model used for the identification of the non-linearity and the dynamic behaviour of the process. The indirect method can be considered as a nonlinear programming problem. The second scheme, on the other hand, is called direct time-delay estimator which is constructed by a neural network. In order to model the time-delay, a neural network is applied. To evaluate the proposed time-delay estimation schemes, two numerical examples are illustrated for comparison.

2. INDIRECT TIME-DELAY ESTIMATION

Suppose the process is described by $f: R^{n+m} \to R$, i.e.

$$y_k = f(Y_{k-1}, U_{k-\tau_k}) \quad (1)$$

where $f(.) \in C^2$, $Y_{k-1} = [y_{k-1},..., y_{k-n}]^T \in R^n$ and $U_{k-\tau_k} = [u_{k-\tau_k},..., u_{k-\tau_k-m}]^T \in R^m$ are respectively the output and input vectors, and τ_k is the time-delay. The neural network based model used to describe the process, i.e.

$$\hat{y}_k = W^T S(x), \quad (2)$$

where $\hat{y}_k$ is the output of the neural model with time-delay, $W = [w_1,..., w_h]^T \in R^h$ is the weight vector connecting the outputs of the hidden layer and the output of the model, $S(x) = [s(x_1),...,s(x_h)]^T \in R^h$ is the output vector of the hidden layer, $s(x) = (1-e^{-x})/(1+e^{-x})$ is the sigmoid function, and the inputs of the sigmoid function in the hidden layer are of the form, i.e.

$$x_i = \sum_{j=1}^{na} w_{ij}\hat{y}_{k-j} + \sum_{j=1}^{nb} w_{i,na+j}u_{k-\hat{\tau}_k-j}, i = 1,..., h \quad (3)$$

where $\hat{\tau}_k$ is the estimate of the time-delay,. na and nb are respectively the lags of the output and input of the neural model, and w_{ij}s are the weights. The introduction of the auto-regression of the model output into the network can be useful to simulate the dynamics of the process. Consider the case where time-delay is varying with time. It is supposed that the time-delay can be separated as integer and fractional parts as well, i.e.

$$\hat{\tau}_k = \hat{d}_k + \delta\hat{\tau}_k, \quad (4)$$

where d_k is the integer part of the time-delay whilst $\delta\tau_k$ denotes the fractal part of the time-delay, which is constrained within the range of one sample period, i.e. (2,3) (Lim and Macleod 1995). Suppose that the time-delay changes slowly so that it means that the time-delay can be considered as constant during one sample period. For the estimation of the fractional part of time-delay, the gradient of the output of the neural model with time-delay respect to $\delta\hat{\tau}$, i.e. $\partial\hat{y}_k/\partial\delta\hat{\tau}$ should be calculated. Considering (2) and (3), it yields

$$\frac{\partial\hat{y}_k}{\partial\hat{\tau}} = \frac{\partial\hat{y}_k}{\partial\delta\hat{\tau}} = \sum_{i=1}^{h} w_i s'(x_i)[\sum_{j=1}^{na} w_{ij}\frac{\partial\hat{y}_{k-j}}{\partial\delta\hat{\tau}} + \sum_{j=1}^{nb} w_{i,na+j}\frac{\partial u_{k-\hat{\tau}_k-j}}{\partial\delta\hat{\tau}}] \quad (5)$$

where $s'(x) = ds(x)/dx = 0.5(1-s(x)^2)$. In (5), the effect of the recurrent connection to the gradient has been considered. Using a method of first-order interpolation can derive the gradient $\partial u_{k-\hat{\tau}_k-j}/\partial\delta\hat{\tau}$. From the first-order Taylor's series expansion, it leads to

$$u_{k-\hat{\tau}_k} \approx u_{k-\hat{d}_k-1} + \delta\tau\frac{u_{k-\hat{d}_k} - u_{k-\hat{d}_k-1}}{k-\hat{d}_k-(k-\hat{d}_k-1)} = u_{k-\hat{d}_k-1} + \delta\tau(u_{k-\hat{d}_k} - u_{k-\hat{d}_k-1}). \quad (6)$$

Hence, the gradient of $u_{k-\hat{\tau}_k}$ with respect to $\delta\tau$ can be derived as

$$\frac{\partial u_{k-\hat{\tau}_k-j}}{\partial\delta\hat{\tau}} = u_{k-\hat{d}_k-j} - u_{k-\hat{d}_k-j-1}. \quad (7)$$

Moreover, the gradients of the output of the neural model with respect to the weights are respectively obtained by

$$\frac{\partial\hat{y}_k}{\partial w_i} = s(x_i), \quad i = 1,...,h; \quad (8)$$

and

$$\frac{\partial\hat{y}_k}{\partial w_{ij}} = \sum_{i=1}^{h} w_i s'(x_i)[\sum_{j=1}^{na} w_{ij}\frac{\partial\hat{y}_{k-j}}{\partial w_{ij}} + \hat{y}_{k-j}] \quad j = 1,...,n_a \quad (9)$$

as well as

$$\frac{\partial\hat{y}_k}{\partial w_{ij}} = \sum_{i=1}^{h} w_i s'(x_i)[\sum_{j=1}^{na} w_{ij}\frac{\partial\hat{y}_{k-j}}{\partial w_{ij}} + u_{k-\hat{\tau}_k-j}] \quad j = 1,...,n_b \quad (10)$$

Define the index

$$Q = 0.5e_k^2 = 0.5(y_k - \hat{y}_k)^2, \quad (11)$$

and the parameter matrices, i.e.

$$\theta = [w_1,..., w_h \mid v_1,..., v_H]^T; \quad \omega = [w_{ij} \mid vij]^T_{(h+H)\times(na+nb+q)}. \quad (12)$$

Then, the estimate of these matrices will be

$$\theta = \theta - \lambda_1\frac{\partial Q}{\partial\theta}, \text{and} \quad \omega = \omega - \lambda_2\frac{\partial Q}{\partial\omega}. \quad (13)$$

where $\lambda_i > 0, \quad (i = 1,2)$ are the optimizing step-sizes. If an optimizing algorithm with second-order convergence, e.g. a modified Levenberg-Marquardt method, is applied, the update of matrices, i.e. θ and ω, becomes

$$\psi(k) = \psi(k-1) - \lambda[H(k) + \alpha I]^{-1}\frac{\partial Q}{\partial\psi} + \beta(\psi(k-1) - \psi(k-2)), \quad \psi = \theta, \omega \quad (14)$$

where $H = (\frac{\partial\hat{y}_k}{\partial\psi})(\frac{\partial\hat{y}_k}{\partial\psi})^T$, $\alpha > 0$ and is adjustable factor. The factor α can be changed from 0 to ∞. When α is zero, the Gauss-Newton algorithm is obtained. If α changes to ∞, the algorithm becomes the gradient descent method. In addition, α has the capability for numerical stabilization if the algorithm is converging towards a saddle point, then $[\partial\hat{y}_k/\partial\Psi]$ may approach zero. In this singular case, $\alpha > 0$ will considerably improve the numerical stability. If the proposed step size is too large, even if the direction of the step is correct, the final result may be worse. By increasing α, it can also reduce the step size, and move more in the direction of the gradient. In order to increase the possibility to escape from some local minimums, a momentum term is embedded into this algorithm and $\beta > 0$ is the momentum factor. When the estimated $\delta\hat{\tau}_k$ is obtained, both the fractional and integer parts will be updated by (Lim and Macleod, 1995)

$$\begin{cases} \hat{d}_k = \hat{d}_k - 1, \\ \delta\hat{\tau}_k = \delta\hat{\tau}_k + 1, \end{cases} \delta\hat{\tau}_k \in (-\infty, k+\eta] \quad \text{or}$$
$$\begin{cases} \hat{d}_k = \hat{d}_k + 1, \\ \delta\hat{\tau}_k = \delta\hat{\tau}_k - 1, \end{cases} \delta\hat{\tau}_k \in [k+1+\eta, \infty) \qquad (15)$$

and

$$\begin{cases} \hat{d}_k = \hat{d}_k, \\ \delta\hat{\tau}_k = \delta\hat{\tau}_k, \end{cases} \quad \delta\hat{\tau}_k \in (k+\eta, k+1+\eta) \qquad (16)$$

where $0 < \eta < 1$ is a very small number. As the indirect time-delay estimation is a procedure of on-line nonlinear programming, it requires much cost on computational load.

3. DIRECT TIME-DELAY ESTIMATION

The procedure of the indirect time-delay estimation illustrated in above section is considered as a problem of nonlinear programming. In this section, the so-called direct time-delay estimation approach will be proposed. A dynamic neural network will be constructed directly for the time-delay estimation. The performance of the estimator depends on the specification of the weights V, the orders of the inputs and the number of the hidden nodes of the network. In this section, the time-delay estimation is formulated as a procedure of system identification. Assume that the time-delay can be separated as integer and fractional parts as well, i.e.

$$\hat{\tau}_k = \hat{d}_k + \delta\hat{\tau}_k, \qquad (17)$$

where d_k is the integer part of the time-delay whilst $\delta\tau_k$ denotes the fractal part of the time-delay.

In order to estimate the time-delay, the estimator of the fractional part of time-delay is proposed as follows:

$$\delta\hat{\tau}_k = g(V_k, I_k), \qquad (18)$$

where $g(.) \in C^2$ realises the mapping $g: R^q \to R$, where $q=q_1+q_2$; V_k is the weight matrix; $I_k = [\delta\hat{\tau}_{k-1},...,\delta\hat{\tau}_{k-q_1}, e_{k-1},...,e_{k-q_2}]^T$, where $e_k = y_k - \hat{y}_k$. The formula (18) can be realised using a neural network, i.e.

$$\delta\hat{\tau}_k = \sum_{i=1}^{H} v_i s(z_i) = \sum_{i=1}^{H} v_i s[\sum_{j=1}^{q} v_{ij} I_{k-j}]. \qquad (19)$$

The gradient of $\hat{y}_k$ with respect to $\hat{\tau}_k$ can be obtained by

$$\frac{\partial\hat{y}_k}{\partial\hat{\tau}} = \frac{\partial\hat{y}_k}{\partial\delta\hat{\tau}} = \sum_{i=1}^{h} w_i s'(x_i)[\sum_{j=1}^{na} w_{ij} \frac{\partial\hat{y}_{k-j}}{\partial\delta\hat{\tau}} + \sum_{j=1}^{nb} w_{i,na+j} \frac{\partial u_{k-\hat{\tau}_k-j}}{\partial\delta\hat{\tau}}] \qquad (20)$$

where

$$\frac{\partial u_{k-\hat{\tau}_k-j}}{\partial\delta\hat{\tau}} = u_{k-\hat{d}_k-j} - u_{k-\hat{d}_k-j-1}. \qquad (21)$$

To determine the weights of the neural network, which is used for the modelling of fractional time-delay, the gradients of $\delta\hat{\tau}$ with respect to v_i and v_{ij} are calculated respectively by:

$$\frac{\partial\delta\hat{\tau}}{\partial v_i} = s(z_i) \qquad i = 1,...,H, \qquad (22)$$

and

$$\frac{\partial\delta\hat{\tau}}{\partial v_{ij}} = \sum_{i=1}^{H} v_i s'(z_i)(I_{k-j} + \sum_{j=1}^{q_1} v_{ij} \frac{\partial\delta\tau_{k-j}}{\partial v_{ij}}] \qquad j = 1,...,q. \qquad (23)$$

The gradients of $\hat{y}_k$ with respect to the weights of the neural network are determined respectively based on (8) - (10). The weights are adjusted using the modified Levenberg-Marquadt method shown in (14).

Then based on the estimated result of the fractional part of time-delay, both the integer part as well as the fractional part of the time-delay are adjusted separately by using the following formulae

$$\begin{cases} \hat{d}_k = \hat{d}_k - 1, \\ \delta\hat{\tau}_k = \delta\hat{\tau}_k + 1, \end{cases} \delta\hat{\tau}_k \in (-\infty, k+\eta] \quad \text{or}$$
$$\begin{cases} \hat{d}_k = \hat{d}_k + 1, \\ \delta\hat{\tau}_k = \delta\hat{\tau}_k - 1, \end{cases} \delta\hat{\tau}_k \in [k+1+\eta, \infty) \qquad (24)$$

and

$$\begin{cases} \hat{d}_k = \hat{d}_k, \\ \delta\hat{\tau}_k = \delta\hat{\tau}_k, \end{cases} \quad \delta\hat{\tau}_k \in (k+\eta, k+1+\eta) \qquad (25)$$

where $0 < \eta < 1$ is a very small number.

The on-line computational load of the direct time-delay estimation will certainly be heavier than the indirect method. However, the direct neural time-delay estimator can be trained either on-line or off-line. Then the training procedure can be cut off if the neural network has been trained well. In this case, the estimator can be implemented with much less on-line computational effort and can be used for fast processes as well.

4. NUMERICAL EXAMPLES

Two numerical examples are illustrated to show the performances of the proposed time-delay estimation approaches.

Example 1: Suppose that the nonlinear process with time-delay is

$$y_k = \frac{y_{k-1} + 0.01}{1 + y_{k-1}^2 + y_{k-2}^2} + 0.5u_{k-\tau}$$

where the time-delay is a continuous time-varying function that is of the form:

$$\tau = \begin{cases} 0.005t + 2, & t < 300 \\ 17, & 300 \le t < 400 \\ 17 - 0.005(t-400), & t \ge 400 \end{cases}$$

Suppose the sampling period is 0.1 second and a neural network with the architecture of single input-single output and five hidden nodes is used for system

modeling. The input signal in the form shown in the following

$$u_k = \sin(k/20) + 2\sin(k/10) + \cos(k/5) + \cos(k/2) + 2\sin(k)$$

is used to stimulate the process. Both the direct and indirect algorithms for time-delay estimation are applied to the modeling procedure respectively. Fig. 1 shows the result of time-delay estimation using direct method. In this method, a neural network with 7 hidden nodes and the input vector of the form:

$$I = [\delta\tau_{k-1}, \delta\tau_{k-2}, \delta\tau_{k-3}, e_{k-1}, e_{k-2}, e_{k-3}, 1]^T$$

is used for the construction of time-delay estimator.

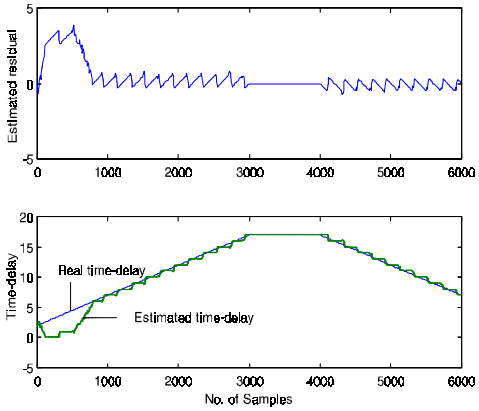

Fig. 1 Time-delay estimation using direct method

For the indirect time-delay estimation method, the parameters for Levenberg-Marquadt algorithm are chosen as: $\lambda = 0.025$, $\alpha = 0.15$ and $\beta = 0.75$. The corresponding time-delay estimation result is illustrated in Fig. 2. The estimation errors are respectively demonstrated in tables 1 and 2.

Table 1 Estimation errors of indirect method

Mean squared error	Maximum error
0.9715	4.4950

Table 2 Estimation errors of direct method

Mean squared error	Maximum error
0.1077	3.8622

From the illustrated results, it is known that the direct approach for time-delay estimation is better that the indirect method. Obviously the direct method results in much smaller estimate residual. The on-line computational cost for the direct method is, however, much more expensive than that of the indirect approach. If the neural time-delay estimator is trained well, then, the training mechanism can be cut off. In this case, the well-trained neural estimator can be implemented with fast speed and much less computational cost. Moreover, the dynamic model validation result is illustrated in Fig. 3. It shows that the system identification based on the on-line time-delay estimation can get satisfactory result.

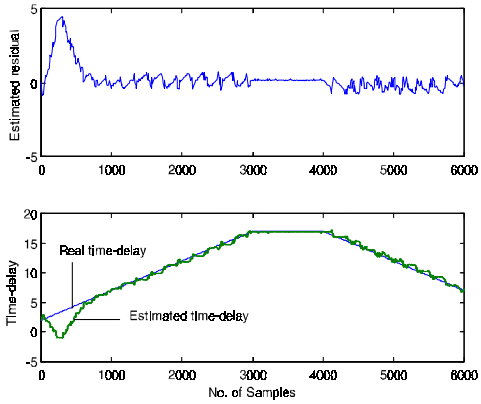

Fig.2 Time-delay estimation using indirect method

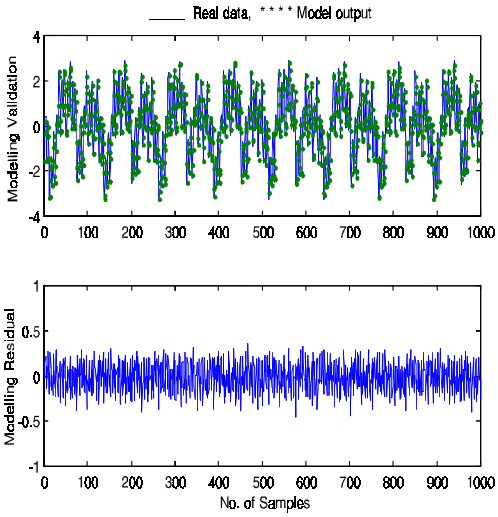

Fig. 3 Dynamic model validation

Example 2: The van de Vusse reactor can be described by[4]

$$\frac{dC_A}{dt} = -k_1 C_A - k_3 C_A^2 + d(t-\tau)(C_{Af} - C_A),$$

$$\frac{dC_B}{dt} = k_1 C_A - k_2 C_B - d(t-\tau) C_B$$

The isothermal series/parallel reactions:

$A \rightarrow B \rightarrow C, \qquad 2A \rightarrow D$

take place in the reactor. In the process, C_A and C_B are respectively the effluent concentration of component A and B, d is the dilution-rate, and τ is the time-delay. The values of the parameters are k_1=50 h^{-1}, k_2=100 h^{-1}, and k_3=10 $l \cdot mol \cdot h^{-1}$. The concentration of A in the feed stream is given by C_{Af} and equals to 10 $mol \cdot l^{-1}$. Initially, the process is at steady state with C_A=0.2143 $mol \cdot l^{-1}$ and C_B=0.1520 $mol \cdot l^{-1}$. The process is sampled at every 0.002 h. The time-varying time-delay τ is caused by the change of the dilution rate flowing through the pipe, i.e.

$$\tau = \begin{cases} 5(1-e^{-0.005t}) + 2; & t < 2h \\ 2 + 5(1-e^{-0.01}) + 3.5(1-e^{-0.015(t-2)}); & 2h \le t < 4h \\ 2 + 5(1-e^{-0.01}) + 3.5(1-e^{-0.03}) - 3.5(1-e^{-0.005(t-4)}); & t \ge 4h \end{cases}$$

Both approaches proposed in this paper are applied to the modeling of the time-delay. Fig. 4 and Fig. 5 respectively demonstrate the time-delay estimate results using the direct as well as the indirect methods.

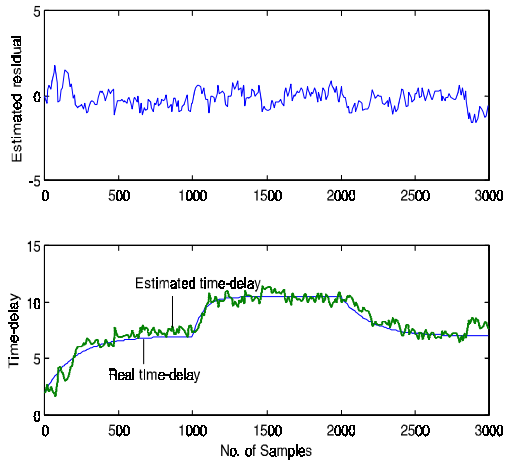

Fig. 4 Time-delay estimate for the van de Vusse reactor using direct method

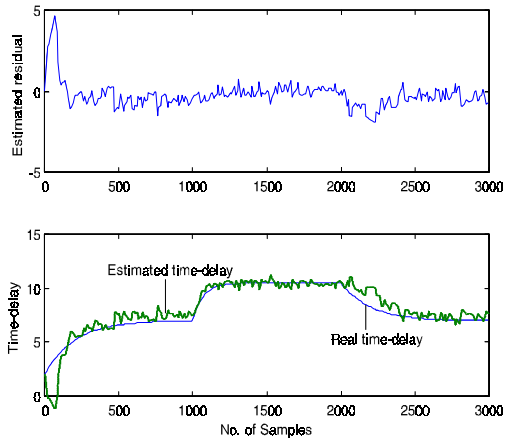

Fig. 5 Time-delay estimate for the van de Vusse reactor using indirect method

It seems that the method of the direct time-delay estimate derives better result than that obtained by using indirect method. In the indirect method, the optimizing step size λ is chosen as 0.275, $\alpha = 0.15$, and $\beta = 0.5$; while in the direct method, the number of the hidden nodes of the neural network is 20, the order of the network inputs are respectively $q_1 = 2$, and $q_2 = 3$. Moreover, the parameters λ, α, β of the training algorithm are respectively 0.95, 0.5 and 0.05.

5. CONCLUSIONS

In this paper, the neural network based direct and indirect time-delay estimation methods for nonlinear dynamic systems with time-varying time-delay are proposed. The proposed indirect approach, based upon a neural model with time-delay to simulate the nonlinear dynamic system with time-delay, can be considered as an on-line nonlinear programming problem. On the other hand, the direct method for time-delay estimation is using a neural network based time-delay estimator to identify the time-delay directly. For the computational cost, the direct method is obviously larger than the indirect method if the on-line training is implemented. However, if the training procedure is finished, the well-trained estimator will have much less computational load than the indirect method since it does not require any on-line optimizing computation in this case. In order to simplify the procedure of time-delay estimation, the technique of dividing the time-delay as both the integer and fractional parts has been applied. The numerical examples show that both the proposed methods can be used to estimate the time-delay for nonlinear systems with time-delay. The direct method however obtained better estimation results.

REFERENCES

Balestrino, A., F. Verona, and A. Landi (1998). On-line process estimation by ANNs and Smith controller design, *IEE Proc., Pt. D. Contr. Theory Appl.*, **145(2)**, 231-235

Lim, T. J. and M. D. Macleod (1995). Adaptive algorithm for joint time-delay estimation and IIR filtering, *IEEE Trans. Signal Processing*, **43(4)**, 841-851

Reed, F., P. Feintuch, and N. Bershad (1981). Time-delay estimation using the LMS adaptive filter-static behavior; dynamic behavior, *IEEE Trans. Acoustics, Speech, and Signal Processing*, **29(3)**, 561-576

Teng, F. C. And H. R. Sirisena (1988). Self-tuning PID controllers for dead time processes, *IEEE Trans. Industrial Electronics*, **35(1)**, 119-125