'92 KACC 1992. 10. 19～21

Identification of Continuous Systems Using Neural Network

Chun-Zhi Jin, Kiyoshi Wada
Dept. of Electrical Engineering,
Faculty of Engineering, Kyushu University, Japan

Setsuo Sagara
Dept. of Communication and Computing Engineering,
Fukuoka Institute of Technology, Japan

Abstract

In this paper an identification of nonlinear continuous systems by using neural network is considered. The nonlinear continuous system is identified by two steps. At first, a linear approximate model of the continuous system with nonlinearity is obtained by IIR filtering approach. Then the modeling error due to the nonlinearity is reduced by a neural network compensator. The teaching signals to train the neural network is gotten by smoothing the measurement data corrupted by noise. An illustrative example is given to demonstrate the effectiveness of the proposed approach.

1 Introduction

Most of real systems are continuous time systems with nonlinearity. However, for convenience of treatment of the problems, the systems are often linearly approximated around the operating point, so the well established linear system theory can be used. But in these cases, the error due to the nonlinearity inevitably exists. While the physical meanings of the parameters of continuous time models are clearer comparing with discrete time models, and the corresponding continuous time models are not uniquely decided when we derive it from the discrete time models. So the continuous time models are more desirable than the discrete time models.

In this paper, deriving a linear approximate model for continuous time systems with nonlinearity, and structuring a neural network compensator to reduce the error due to the nonlinearity are considered. In order to apply the digital computers to the treatment of continuous time systems, an identification model which has the same parameters of the continuous time systems are derived by IIR filtering approach and bilinear transformation. Then the parameters of the linear approximate model are estimated by the recursive LS method from sampled data. To reduce the modeling error due to the nonlinearity, we present a structure of neural network compensator. An illustrative example are given to demonstrate the effectiveness. Finally, some conclusions are presented.

[1]'92 Korean Automatic Control Conference, Oct. 19-21, 1992, Seuol, Korea

2 Statement of the problem

Consider single input single output continuous time systems as the following:

$$\begin{aligned} A(p)x(t) &= B(p)u(t) + N(p, u(t), x(t)), \\ A(p) &= a_0 p^n + \cdots + a_n, \quad a_0 = 1, \\ B(p) &= b_1 p^{n-1} + \cdots + b_n. \end{aligned} \tag{1}$$

$$\begin{aligned} ||A(p)x(t)|| &>> ||N(p, u(t), x(t))|| > \varepsilon \\ ||B(p)u(t)|| &>> ||N(p, u(t), x(t))|| > \varepsilon \end{aligned} \tag{2}$$

where p is the differential operator, $u(t), x(t)$ are the input signal and the output signal of the system on instant t, and $N(p, u(t), x(t))$ is a nonlinear function of $p, u(t), x(t)$. The condition (2) shows that the system can be approximated by appropriate linear model, and the error due to nonlinearity can not be negligible though it can be limited to a small value. Assume that the poles of $A(p) = 0$ are in left complex plane so that the system is stable, and it is irreducible between $A(p)$ and $B(p)$ so that guarantee the identifiability.

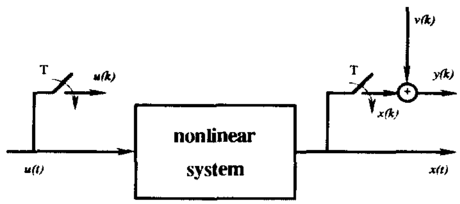


Figure 1: System to be treated

Practically the sampled output measurement is always corrupted by a noise, the output measurement on instant kT (T is the sampling period) is:

$$y(k) = x(k) + v(k) \quad (3)$$

where note $y(kT)$, $x(kT)$, $v(kT)$ respectively as $y(k)$, $x(k)$, $v(k)$, for convenience. $v(k)$ is the measurement noise, it is assumed to be white noise. Such system is shown in figure 1.

In this paper, we estimate the parameters of linear approximate model from input-output sampled data, and structures a neural network compensator to reduce the modeling error due to nonlinearity.

3 Estimation of linear approximated model

The system to be treated is a nonlinear system, so there must be the modeling error due to the nonlinearity when the system is approximated as linear model. Usually, the model order is higher, the accuracy is better. However, the model will become complicated and difficult to be treated. Practically the model order decided by tread-off between the accuracy and the simplicity. Here, we assume that the model order is known.

While when we estimate the continuous time model, the differential value of input-output signal are not often gotten directly from the measurement, we have to do differential operation on input-output signals. So how to reduce the effect of noise due to differentiation is a important point. Here, We estimate the parameters of the systems by the digital low-pass filtering approach.

First, we choice one of the low-pass filters. Here, Butterworth filter (a type of IIR filters) which the cut-off frequency is ω_c, and the order is m ($m \geq$ model order n) is used.

$$F(p) = \frac{1}{(\frac{p}{\omega_c})^m + c_1(\frac{p}{\omega_c})^{m-1} + \cdots + c_m} \quad (4)$$

where $c_i (i = 1, \cdots, c_m)$ are the coefficients of Butterworth filter. Butterworth filter when $m = 2$ is shown as the following:

$$F(p) = \frac{1}{(\frac{p}{\omega_c})^2 + \sqrt{2}(\frac{p}{\omega_c}) + 1} \quad (m = 2) \quad (5)$$

Multiplying both side of the equation (1) by the filter (4), we have

$$\begin{aligned} & F(p)p^n x(t) + \sum_{i=1}^{n} a_i F(p) p^{n-i} x(t) = \\ & \quad \sum_{i=1}^{n} b_i F(p) p^{n-i} u(t) + F(p) N(p, u, x) \end{aligned} \quad (6)$$

Discretizing it by the bilinear transformation:

$$p = \frac{2}{T}\frac{1 - z^{-1}}{1 + z^{-1}} \quad (7)$$

and considering the equation (3), we obtain

$$\begin{aligned} & \xi_{0y}(k) + \sum_{i=1}^{n} a_i \xi_{iy}(k) = \sum_{i=1}^{n} b_i \xi_{iu}(k) \\ & \quad + \mathcal{N}(z, u(k), x(k)) + r(k) \end{aligned} \quad (8)$$

where

$$\begin{aligned} \xi_{iy}(k) &= Q(z^{-1})(\frac{T}{2})^i (1 + z^{-1})^i \\ & \quad \cdot (1 - z^{-1})^{n-i} x(k) \\ \xi_{iu}(k) &= Q(z^{-1})(\frac{T}{2})^i (1 + z^{-1})^i \\ & \quad \cdot (1 - z^{-1})^{n-i} u(k) \\ r(k) &= \sum_{i=0}^{n} a_i Q(z^{-1})(\frac{T}{2})^i (1 + z^{-1})^i \\ & \quad \cdot (1 - z^{-1})^{n-i} v(k) \\ & \qquad (i = 0, 1, \cdots, n) \\ Q(z^{-1}) &= \frac{(\frac{T}{2})^{m-n} (1 + z^{-1})^{m-n}}{(\frac{1 - z^{-1}}{\omega_c})^m + R(z^{-1})} \\ R(z^{-1}) &= \sum_{i=1}^{m} c_i (\frac{1 - z^{-1}}{\omega_c})^{m-i} (\frac{T}{2})^i (1 + z^{-1})^i \end{aligned} \quad (9)$$

and the term $\mathcal{N}(z, u(k), x(k))$ is the one of discretizing $F(p) \cdot N(p, u, x)$. Here given that:

$$\begin{aligned} z(k) &= [-\xi_{1y}(k), -\xi_{2y}(k), \cdots, -\xi_{ny}(k), \\ & \quad \xi_{1u}(k), \xi_{2u}(k), \cdots, \xi_{nu}(k)]^T \\ \boldsymbol{\theta} &= [a_1, a_2, \cdots, a_n, b_1, b_2, \cdots, b_n]^T \\ \delta &= \mathcal{N}(z, u(k), x(k)) + r(k) \end{aligned} \quad (10)$$

then,

$$\xi_{0y}(k) = \boldsymbol{z}^T(k)\boldsymbol{\theta} + \delta \quad (11)$$

δ is a term of error consisted from system nonlinearity and measurement noise. If the input signals are sufficiently rich, then we can estimate the parameters of the linear approximate model by the following LS method:

$$\hat{\boldsymbol{\theta}}(k) = \left[\sum_{k=1}^{N} \rho(k)[\boldsymbol{z}(k)\boldsymbol{z}^T(k)]\right]^{-1} \cdot \left[\sum_{k=1}^{N} \rho(k)\boldsymbol{z}(k)\xi_{0y}(k)\right] \quad (12)$$

where $\rho(k)$ is the forgetting factor, N is the number of data. For the equation (12), the recursive LS algorithm can be described by the following form:

$$\begin{aligned}
\hat{\boldsymbol{\theta}}(k) &= \hat{\boldsymbol{\theta}}(k-1) \\
&\quad + \gamma(k)\boldsymbol{P}(k-1)\boldsymbol{z}(k)\varepsilon(k) \\
\varepsilon(k) &= \xi_{0y}(k) - \boldsymbol{z}^T(k)\hat{\boldsymbol{\theta}}(k-1) \\
\boldsymbol{P}(k) &= \frac{1}{\rho(k)}[\boldsymbol{P}(k-1) - \gamma(k) \\
&\quad \cdot \boldsymbol{P}(k-1)\boldsymbol{z}(k)\boldsymbol{z}^T(k)\boldsymbol{P}(k-1)] \\
\gamma(k) &= \frac{1}{\rho(k) + \boldsymbol{z}^T(k)\boldsymbol{P}(k-1)\boldsymbol{z}(k)}
\end{aligned} \quad (13)$$

The initial values $\boldsymbol{\theta}(0)$, $\boldsymbol{P}(0)$ are choose as:

$$\begin{aligned}
\boldsymbol{\theta}(0) &= \text{any value} \\
\boldsymbol{P}(0) &= \alpha \boldsymbol{I}
\end{aligned} \quad (14)$$

where α is a sufficiently large positive constant. The forgetting factor $\rho(k)$ is[10]

$$\rho(k) = 0.99\rho(k-1) + 0.01, \rho(0) = 0.95 \quad (15)$$

4 Neural Network Compensator

Neural Network

The neural network used in this paper, is a three layers network with n_i inputs, 1 output. The notations of each input-output signal, each weight between neurons and each threshold of neurons are as shown in figure 2. Only the neurons on the hidden layer have sigmoidal function property, the neurons on the output layer have linear property, and the neurons on the input layer only pass the input signals to output. The relationship of the network from input to output is as following equation:

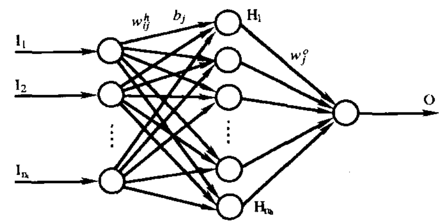


Figure 2: Three layer neural network

$$\begin{aligned}
y_j &= f(\sum_{i=1}^{n_i} w_{ij}^h I_i - b_j) \quad (16) \\
H_j &= \frac{1}{1 + exp(-y_j)} \quad (17) \\
O &= \sum_{j=1}^{n_h} w_j^o H_j \quad (18)
\end{aligned}$$

The weights of the network is adjusted by the backpropagation with teaching signals:

$$\boldsymbol{\Delta\tau}_{(l+1)} = -\eta \left(\frac{\partial E}{\partial \boldsymbol{\tau}}\right)_{(l)} + \beta \boldsymbol{\Delta\tau}_{(l)} \quad (19)$$

where E is the square error of the output of network, η, β are appropriate positive coefficients. $\boldsymbol{\tau}$ is the parameters vector consisted of the weights and thresholds of the network, $\boldsymbol{\Delta\tau}$ is degree of adjustment and l is the number of training times.

Neural Network Compensator

The differential equation of the linear approximate model obtained from section 3, is given as the following:

$$\begin{aligned}
\hat{A}(p)\hat{x}_l(t) &= \hat{B}(p)u(t), \\
\hat{A}(p) &= p^n + \hat{a}_1 p^{n-1} + \cdots + \hat{a}_n \\
\hat{B}(p) &= \hat{b}_1 p^{n-1} + \cdots + \hat{b}_n.
\end{aligned} \quad (20)$$

Discretizing the system (1) and the model (20), we obtain the output of the system and the output of the linear approximate model as the following:

$$\begin{aligned}
x(k) &= f(x(k-1), \cdots, x(k-n), \\
&\qquad u(k-1), \cdots, u(k-n)) \quad (21) \\
x_l(k) &= f_l(x_l(k-1), \cdots, x_l(k-n), \\
&\qquad u(k-1), \cdots, u(k-n)) \quad (22)
\end{aligned}$$

where $f(\cdot)$ and $f_l(\cdot)$ are the discretized system function and the discretized model function respectively. Thus the modeling error $e(k)$ is:

$$
\begin{aligned}
e(k) &= x_l(k) - x(k) \\
&= g(x(k-1), \cdots, x(k-n), \\
&\qquad x_l(k-1), \cdots, x_l(k-n), \\
&\qquad u(k-1), \cdots, u(k-n))
\end{aligned} \tag{23}
$$

where $g(\cdot)$ is the error function. The $e(k)$ contains mainly the error due to the nonlinearity even though it contains another approximating errors. From equation (23), the modeling error $e(k)$ depend on the past values of the input $u(k-1), \cdots, u(k-n)$, the past values of the system output $x(k-1), \cdots, x(k-n)$ and the past values of the output of the linear approximate model $x_l(k-1), \cdots, x_l(k-n)$. Thus using a neural network which has the inputs $x(k-1), \cdots, x(k-n)$, $x_l(k-1), \cdots, x_l(k-n)$, $u(k-1), \cdots, u(k-n)$ and the output $\hat{e}(k)$, if we approximate the relationship of function $g(\cdot)$, and add the output of the network to the output of the linear approximate model, then the modeling error can be compensated.

While the signals used in equation (23) $x(k), x(k-1), \cdots, x(k-n)$ are the true outputs of the system, in practice, only the corrupted output measurements $y(k)$, $y(k-1), \cdots$ can be gotten. If we only replace $x(k), x(k-1), \cdots, x(k-n)$ by $y(k), y(k-1), \cdots, y(k-n)$, we can not get the correct compensating signal even if the neural network represent the function $g(\cdot)$ exactly because the effect of the output measurement noise. While the network which accurately represent the function $g(\cdot)$ can not be gotten when the network is trained by using such signals corrupted by noise. We have to find a way to reduce the noise effect.

The outputs of the linear approximate model $x_l(k-1)$, $\cdots, x_l(k-n)$ are the approximation of the system outputs $x(k-1), \cdots, x(k-n)$. So we replace the outputs of the system $x(k-1), \cdots, x(k-n)$ to the outputs of the linear approximate model $x_l(k-1), \cdots, x_l(k-n)$ in equation (21), and use the past values of inputs $u(k-n-1), \cdots, u(k-2n)$. This means that we hope to infer the system output in some degree by addition of $u(k-n-1), \cdots, u(k-2n)$. Thus we have

$$
\begin{aligned}
x(k) &= f(x(k-1), \cdots, x(k-n), \\
&\qquad u(k-1), \cdots, u(k-n)) \\
&\approx \bar{f}(x_l(k-1), \cdots, x_l(k-n), \\
&\qquad u(k-1), \cdots, u(k-2n))
\end{aligned} \tag{24}
$$

and the error compensating signal is

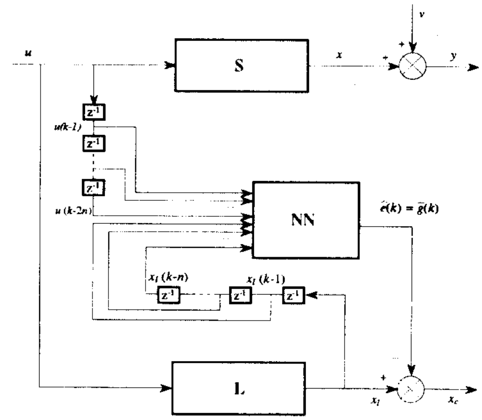


Figure 3: The neural network compensator

$$
\begin{aligned}
e(k) &= x_l(k) - x(k) \\
&= \bar{g}(x_l(k-1), \cdots, x_l(k-n), \\
&\qquad u(k-1), \cdots, u(k-2n))
\end{aligned} \tag{25}
$$

By this equation, a neural network error compensator is structured as shown in figure 3.

The Training of Neural Network

From equation (25), adjustment of the parameters of the neural network should be carried out by regarding the data set $\{e(k), x_l(k-1), \cdots, x_l(k-n), u(k-1), \cdots, u(k-2n)\}$ as teaching signals. However, $e(k)$ is $e(k) = x_l(k) - x(k)$, so the true output of the system $x(k)$ on instant k is required. One of the way is that many output measurement values are stored up, the $x(k)$ is estimated by any proper statistical approach, then some typical teaching signals based on the estimated values are formed and the weights of the network are decided by off-line training. We present another way that is on-line training approach based on data smoothing as the following explanation.

Usually, the low-pass filter, the data smoothing, *etc* are effective to reduce noise. However, the information used in the data smoothing is more than in the low-pass filter, thus the efficiency reducing noise effect better than low-pass filter. So we use one of the smoothing approach called Savizky-Golay method[2] here.

$$
\tilde{y}(k-m) = \frac{1}{W} \sum_{j=-m}^{m} w(j) y(k-m+j) \tag{26}
$$

$$
W = \sum_{j=-m}^{m} w(j) \tag{27}
$$

where $\tilde{y}(k-m)$ is the smoothed measurement of system output on instant$(k-m)$. $y(k-m+j)$ is the output measurement on instant $(k-m+j)$ and $w(j)$ is the coefficient of smoothing. The larger is the value of m, the better is degree reducing noise effect, while the large is wave deformation, so we have to take a tread-off between them.

From equation (26), we can only get the smoothed signals until instant $(k-m)$ on instant k, so the error signal computed on the basis of above can be also gotten until instant $(k-m)$. Namely, the error signal is given by $e(k-m) = x_l(k-m) - \tilde{y}(k-m)$. Therefore, the weights of the neural network have to be adjusted in accordance with the data delayed m instants. Yet, delayed data set $\{e(k-m), x_l(k-m-1), \cdots, x_l(k-m-n), u(k-m-1), \cdots, u(k-m-2n)\}$ satisfy the relationship of the function $\bar{g}(\cdot)$, the neural network can approximate the function accurately if the training is sufficient. The training of the neural network and the error compensating are carried out every instant. On instant k, the weights of the network are adjusted in accordance with data delayed m instants, then the inputs of the adjusted network are changed to $x_l(k-1), \cdots, x_l(k-n), u(k-1), \cdots, u(k-2n)$, and the model output is revised by error compensating signal $\hat{e}(k)$ obtained from the output of the network.

5 Example

Consider a nonlinear system described by the following differential equation:

$$\ddot{x} + 3\dot{x} + 0.3\dot{x}^2 + 4x + 0.8\sin(x) = 4u \tag{28}$$

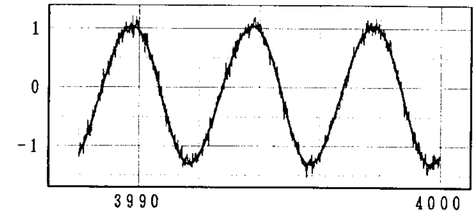
The simulation experiment are carried out with the sampling period $T = 0.02$, N/S=10%. Using the IIR filter approach under the condition: the input signal $u = \sin t + 0.5\sin 3t + 0.3\sin 5t + 0.2\sin 7t + \sin 1.5t + 2\sin 4.5t$, sampling number N=10000, the cut-off frequency of Butterworth filter $\omega_c = 4$, the order $m = 2$, the parameters of the linear approximate model are estimated. the obtained model is:

$$\ddot{x}_l + \hat{a}_1\dot{x}_l + \hat{a}_2 x_l = \hat{b}_1\dot{u} + \hat{b}_2 u \tag{29}$$

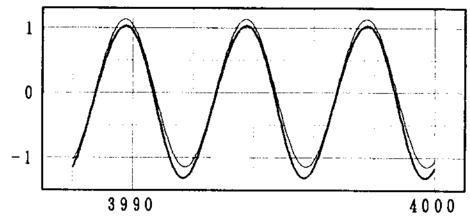
$$\begin{aligned} \hat{a}_1 &= 3.1066, \quad \hat{a}_2 = 4.6569 \\ \hat{b}_1 &= 0.0025, \quad \hat{b}_2 = 3.9582 \end{aligned} \tag{30}$$

When the input is a sinusoid signal $u = \sin(\pi kT)$, the system output measurement and the model output are plotted in figure 4(a), 4(b) respectively. The bold curves are true output of the system.

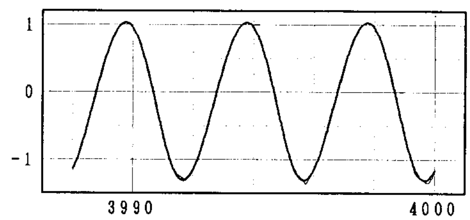
The modeling error is compensated by a neural network with 6 inputs, 1 output, and 3 neuron on hidden layer.



(a) System output and the measurement



(b) System output and model output



(c) compensated model output

Figure 4: Outputs

The inputs are $x_l(k-1), x_l(k-2), u(k-1), u(k-2), u(k-3), u(k-4)$ respectively. To get the error teaching signal required for training, the measurement of the system output is smoothed by using 21 points polynomial smoothing method[2]. The coefficients of smoothing are -171, -76, 9, 84, 149, 204, 249, 284, 309, 324, 329, 324, 309, 284, 249, 204, 149, 84, 9, -76, -171, and the regularization constant is 3059. The 21 points smoothing is carried out delayed 10 instants, so the network is also trained on the same instant. Figure 4(c) shows that model output are compensated by the neural network. Figure 5 shows that the errors before and after the compensation. That is the result after 200000 times training. The compensated effect is clear.

6 Conclusion

In this paper, the approach for identification of nonlinear continuous time systems is proposed. In the approach the parameters of the linear approximate model are estimated from the input-output sampled data of the nonlinear

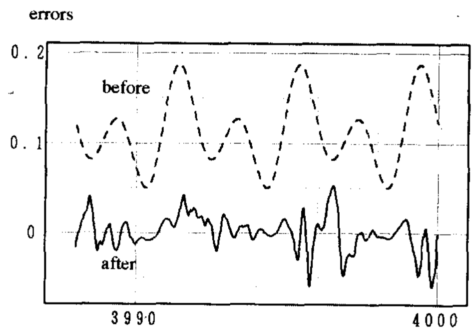


Figure 5: Errors before and after the compensation

systems by low-pass filtering approach, and the modeling error is compensated by using neural network in the cases with the measurement noise. On the training of the neural network, the output measurement corrupted by noise can not be directly used as the teaching signals, so the error teaching signals are gotten by data smoothing method. The training is carried out based on the data delayed some instants because the smoothing method is used. The modeling error is compensated on current instant using the network gotten by the training. Finally, the illustrative example is given to demonstrate the effectiveness of the proposed approach.

References

[1] S. Chen, S. A. Billings and P. M. Grant, "Non-linear system identification using neural networks," *Int. J. Control*, vol.51, no.6, pp.1192-1214, 1991.

[2] S. Minami, "Wave data processing for scientific measurements," CQ, 9 th edit, 1992 (Japanese)

[3] K. S. Narendra and K. parthasarathy, "Identification and control of dynamical systems using neural networks," *IEEE Trans. Neural Networks*, vol.1, pp. 4-27, Mar., 1990.

[4] K. S. Narendra and K. parthasarathy, "Gradient methods for the optimization of dynamical systems containing neural networks," *IEEE Trans. Neural Networks*, vol.2, no.2 pp. 252-262, Mar., 1991.

[5] S. Sagara, C. Z. Jin, Z. J. Yang, "Identification of continuous systems with a nonlinear compensator using neural networks," *Proc. of the '91 Korean Automatic Control Conference*, Vol.2 of 2, pp. 1354-1358, 1991.

[6] S. Sagara, Z. J. Yang, K. Wada, "Consistent parameter estimation for continuous systems via IIR digital filters," *Trans. Soc. Instrum. Control Engrs Japan*, vol.26, no.1, pp.39-45 1990.

[7] S. Sagara, Z. J. Yang, K. Wada, "Identification of continuous systems using low-pass filters," *Int. J. Systems sci.* vol.22, no.7, pp.1159-1176 1991.

[8] H. Unbehauen and G. P. Rao, "Identification of continuous systems," Amsterdam: North-Holland, 1987.

[9] P. J. Werbos, "Backpropagaton through time: What is does and how to do it," *Proc. IEEE*, vol.78, pp. 1550-1560, Oct., 1990.

[10] YANG Z.J.: "Identification of continuous-time systems using digital signal processing techniques," thesis for the doctor's degree, 1991.