Journal of
Marine Science and Technology

Volume 27 | Issue 4 Article 4

PARAMETER IDENTIFICATION USING THE NELDER?€“MEAD SIMPLEX ALGORITHM FOR LOW SIGNAL-TO-NOISE RATIO SYSTEMS IN A FREQUENCY DOMAIN

Chyun-Chau Fuh
Department of Mechanical and Mechatronic Engineering, National Taiwan Ocean University, Keelung, Taiwan, R.O.C.,
f0005@mail.ntou.edu.tw

Hsun-Heng Tsai
Department of Biomechatronics Engineering, National Pingtung University of Science and Technology, Pingtung, Taiwan, R.O.C.

Recommended Citation
Fuh, Chyun-Chau and Tsai, Hsun-Heng (2019) "PARAMETER IDENTIFICATION USING THE NELDER?€“MEAD SIMPLEX ALGORITHM FOR LOW SIGNAL-TO-NOISE RATIO SYSTEMS IN A FREQUENCY DOMAIN," *Journal of Marine Science and Technology*: Vol. 27: Iss. 4, Article 4.
DOI: 10.6119/JMST.201908_27(4).0004
Available at: https://jmstt.ntou.edu.tw/journal/vol27/iss4/4

Journal of Marine Science and Technology, Vol. 27, No. 4, pp. 332-342 (2019)
DOI: 10.6119/JMST.201908_27(4).0004

PARAMETER IDENTIFICATION USING THE NELDER–MEAD SIMPLEX ALGORITHM FOR LOW SIGNAL-TO-NOISE RATIO SYSTEMS IN A FREQUENCY DOMAIN

Chyun-Chau Fuh[1] and Hsun-Heng Tsai[2]

Key words: parameter identification, frequency domain, simplex algorithm, noise.

ABSTRACT

Parameter identification algorithms are very fundamental techniques in system engineering practices. For example, estimating the parameters of the AutoRegresive model with an eXternal input or AutoRegresive Moving-Average model with an eXternal input by using the least squares (LS) method has become a standard approach. However, the estimated parameters may generate extremely erroneous results when the signal is disturbed by large noise, which cannot be effectively filtered. If a frequency response method that scatters the power of a broadband noise over different frequencies is adopted, the effect of noise on the estimated parameters would be relatively reduced. Moreover, estimating whether the plant is a high-order system or is perturbed by a large noise is difficult. The estimated accuracy decreases even after applying the generalized LS method or other modified approaches. To overcome this problem, this study proposed a new technique combining a simplex algorithm and frequency response method for improving the accuracy of the parameter estimation of a dynamic system with a large noise (i.e., an extremely low signal-to-noise ratio) of the system. The algorithm is simple and easy to implement. Moreover, the precision of parameter identification can be increased even when estimated systems suffer from large measurement noises.

I. INTRODUCTION

The principle of system identification is to utilize the measured input-output signals for deducing the mathematical model and related parameters of a dynamic system. Continuous and discrete-time transfer functions, state space models, and process models can be identified using input–output data in a time or frequency domain (Norton, 1986; Ljung, 1987; Johansson, 1993; Bosch et al., 1994; Verhaegen et al., 2007; Keesman, 2011; Nevaranta et al., 2017).

The least squares (LS) or generalized LS (GLS) method is generally employed to estimate dynamic systems constructed using the AutoRegresive model with an eXternal input (ARX) or AutoRegresive Moving-Average model with an eXternal input (ARMAX). However, in systems with a relatively low signal-to-noise ratio (SNR), large estimation deviations are frequently observed (Ljung, 1987).

In the study, the Nelder-Mead simplex method was used to identify the parameters of a system with the characteristics of a low SNR based on a frequency domain. The simplex method was primarily presented by Spendley, Hext, and Himsworth (1962), which was further improved by Nelder and Mead (1965). The algorithm presented by Nelder and Mead (referred to as the N-M simplex method or N-M method in this study) is an easy to implement and underutilized method for calculating the minimum or maximum value of the objective function in a multidimensional parameter space. Furthermore, the objective function cannot be used to derive the proposed N-M method; therefore, it is particularity suitable for the cases, where the objective function is undifferentiable or unanalyzable, or includes noise (Luersen and Riche, 2004; Chelouah and Siarry, 2005; Hedar and Fukushima, 2006; Fuh, 2009).

The results of the numerical simulation demonstrate that the parameters of the dynamic system with a large noise can be evaluated on the frequency domain by applying the N-M simplex method.

II. NELDER-MEAD SIMPLEX ALGORITHM

The parameter identification based on the frequency response method is less recognized than other methods, such as those mentioned in Section 1. However, different types of noise, such

Paper submitted 03/07/19; revised 04/18/19; accepted 06/12/19. Author for correspondence: Chyun-Chau Fuh (e-mail: f0005@mail.ntou.edu.tw).

[1] *Department of Mechanical and Mechatronic Engineering, National Taiwan Ocean University, Keelung, Taiwan, R.O.C.*

[2] *Department of Biomechatronics Engineering, National Pingtung University of Science and Technology, Pingtung, Taiwan, R.O.C.*

as white, color, broadband, low-frequency, and high-frequency noise, may appear in physical, electrical, mechanical, and other real-world systems. The identification of the parameters of a system with a low SNR by using the aforementioned methods often fails. In this research, the N-M simplex method with an objective function of frequency was proposed to estimate the parameters of systems with large measurement noises.

The N-M simplex algorithm is designed to solve unconstrained optimization problems of the following form:

$$\min_{\mathbf{p}\in\mathbf{R}^n} J(\mathbf{p}) \tag{1}$$

where $J(\mathbf{p})$ is an objective function (also be termed a target function, cost function, performance index, etc.), and $\mathbf{p}$ is a vector comprising parameters to be estimated. After determining the form of the objective function, the N-M simplex method generates a sequence of simplexes, where each simplex is defined using $n + 1$ distinct vertices $\mathbf{p}_0, \ldots, \mathbf{p}_n$, with the corresponding function values of $J_0, \ldots, J_n$, respectively. Points $\mathbf{p}_0, \ldots, \mathbf{p}_n$ are sorted such that $J_0 \le \ldots \le J_{n-1} < J_n$, and $\bar{\mathbf{p}}$ represents a centroid of points $\mathbf{p}_0, \ldots, \mathbf{p}_{n-1}$. In each iteration, simplex transformations in the N-M simplex method are controlled using parameters α, β, and γ. They must satisfy the following conditions:

$$0 < \beta < 1, 0 < \alpha < \gamma . \tag{2}$$

The typical values are as follows: $\alpha = 1$, $\beta = \pm 0.5$, and $\gamma = 2$. The values of α, γ, β, and $-\beta$ yield the reflection point $\mathbf{p}_r$, expansion point $\mathbf{p}_e$, outer contraction point $\mathbf{p}_c$, and inner con-
traction point $\mathbf{p}_{cc}$, respectively. The objective function at these four points are denoted as J_r, J_e, J_c, and J_{cc}, respectively. If none of the four points are improved on the current worst point $\mathbf{p}_n$, the algorithm shrinks the points $\mathbf{p}_1, \ldots, \mathbf{p}_n$ toward the lowest $\mathbf{p}_0$, thereby producing generating a new simplex. In the shrinking process, each $\mathbf{p}_j$ is replaced by $0.5(\mathbf{p}_0 + \mathbf{p}_j)$ for $j = 1, \ldots, n$. A new iteration is automatically triggered after accomplishing the shrinking process. This iterative process is continued until the specified termination criteria are satisfied (e.g., when the iterations reach the allowed maximum number and the accuracy of seeking the function value J_0 is higher than the default value).

The geometric phenomenon of the N-M simplex method for a two-dimensional parameter (i.e., $n = 2$) is discussed in this section. A typical N-M simplex algorithm may generate a series of simplexes. Each simplex comprises three vertices $\mathbf{p}_0$, $\mathbf{p}_1$, and $\mathbf{p}_2$ with the corresponding objective function values J_0, J_1, and J_2, respectively, where vertices $\mathbf{p}_0$, $\mathbf{p}_1$, and $\mathbf{p}_2$ are ordered as follows: $J_0 \le J_1 < J_2$, and $\bar{\mathbf{p}}$ denotes the centroid of $\mathbf{p}_0$ and $\mathbf{p}_1$. In each iteration, the N-M simplex method examines one or more of four different ζ values along the line $\bar{\mathbf{p}} + \zeta(\bar{\mathbf{p}} - \mathbf{p}_n)$, ($n$ = 2 in this example). These four values α, γ, β, and $-\beta$ yield the reflection point $\mathbf{p}_r$, expansion point $\mathbf{p}_e$, outer contraction point $\mathbf{p}_c$, and inner contraction point $\mathbf{p}_{cc}$, respectively. The objective function values at these four points are denoted as J_r, J_e, J_c, and J_{cc}. If none of the four points represent an improvement on the current worst point $\mathbf{p}_2$, the algorithm shrinks points $\mathbf{p}_1$ and $\mathbf{p}_2$ toward the optimal point $\mathbf{p}_0$, thereby producing a new simplex. In the shrinking process, each $\mathbf{p}_j$ is replaced by $0.5(\mathbf{p}_0 + \mathbf{p}_j)$ for $j = 1, 2$. After producing a new simplex, a new iteration is automatically triggered. This iterative process is continued until the specified termination criteria are satisfied.

A conventional algorithm for the N-M simplex method is summarized as follows:

Initialization

Step 1. Let $\alpha = 1$, $\beta = 0.5$, and $\gamma = 2$.
Step 2. Give an initial simplex comprising $n + 1$ vertices $\mathbf{p}_0, \ldots, \mathbf{p}_n$.
Step 3. Calculate function values $J_0, \ldots, J_n$ corresponding to $\mathbf{p}_0, \ldots, \mathbf{p}_n$, respectively.

Loop

Step 4. Sort $\mathbf{p}_0, \ldots, \mathbf{p}_n$ such that $J_0, \ldots, J_n$ are in an ascending order.
Step 5. **(Reflection)** Let $\mathbf{p}_r = \bar{\mathbf{p}} + \alpha(\bar{\mathbf{p}} - \mathbf{p}_n)$ and calculate J_r.
Step 6. Let $\mathbf{p}_{\text{new}}$ be undefined.
Step 7. If $J_r < J_0$, go to Step 8; else go to Step 10.
Step 8. **(Expansion)** Let $\mathbf{p}_e = \bar{\mathbf{p}} + \gamma(\bar{\mathbf{p}} - \mathbf{p}_n)$ and calculate J_e.
Step 9. If $J_e < J_r$, $\mathbf{p}_{\text{new}} = \mathbf{p}_e$; else $\mathbf{p}_{\text{new}} = \mathbf{p}_r$. Go to Step 15.
Step 10. If $J_r < J_n$, go to Step 11; else go to Step 13.
Step 11. **(Outer contraction)** Let $\mathbf{p}_c = \bar{\mathbf{p}} + \beta(\bar{\mathbf{p}} - \mathbf{p}_n)$ and calculate J_c.
Step 12. If $J_c < J_r$, $\mathbf{p}_{\text{new}} = \mathbf{p}_c$; else $\mathbf{p}_{\text{new}} = \mathbf{p}_r$. Go to Step 17.
Step 13. **(Inner contraction)** Let $\mathbf{p}_{cc} = \bar{\mathbf{p}} - \beta(\bar{\mathbf{p}} - \mathbf{p}_n)$ and calculate J_{cc}.
Step 14. If $J_{cc} < J_n$, $\mathbf{p}_{\text{new}} = \mathbf{p}_{cc}$. Go to Step 17.
Step 15. If $\mathbf{p}_{\text{new}}$ is undefined, go to Step 15; else go to Step 17.
Step 16. **(Shrink)** Let $\mathbf{p}_j = 0.5(\mathbf{p}_0 + \mathbf{p}_j)$ and $j = 1, \ldots, n$. Go to Step 18.
Step 17. Let $\mathbf{p}_n = \mathbf{p}_{\text{new}}$ and calculate J_n.

Step 18. If the termination criteria are satisfied, end the iteration procedure; else go to Step 4.

III. PARAMETER IDENTIFICATION USING THE NELDER-MEAD SIMPLEX ALGORITHM

Consider an n^{th} order linear time invariant (LTI) dynamical system with the following transfer function:

$$G(s) = \frac{b_{n-1}s^{n-1} + \ldots + b_1 s + b_0}{s^n + a_{n-1}s^{n-1} + \ldots + a_1 s + a_0} \tag{3}$$

where the parameters are unknown but constant, and the parameter vector is defined as follows: $\mathbf{p} = \left[\, b_{n-1}, \ldots, b_1, b_0, a_{n-1}, \ldots, a_1, a_0 \,\right]$. In this section, the estimation of the parameter vector $\mathbf{p}$ is proposed using the N-M simplex algorithm.

Because in actual experiments, the parameter identification is generally performed using a digital computer, Δt (unit: s) was used to represent the sampling time of the signal. Thus, the sampling frequency can be denoted using $f_s = 1/\Delta t$ (unit: Hz) or $\omega_s = 2\pi f_s$ (unit: rad/s).

Before performing parameter identification for a system, the stability of the system in an open-loop mode must be confirmed. First, the open-loop system was assumed to be stable. The algorithm proposed in this study for a stable open-loop system is as follows:

1. Stable System

Step a: Directly inject a sinusoidal exciting signal $u_k(t) = A_k \sin(\omega_k t)$ and $k = 1,..., N$ to the open-loop system.

Step b: Measure the steady output signal $y_k(t) = B_k \sin(\omega_k + \phi_k)$ and $k = 1, \ldots, N$.

Step c: Perform fast Fourier transform (FFT) (Brigham, 1988; Oppenheim, 1989) for $u_k(t)$ and $y_k(t)$ to obtain $U(j\omega_k)$ and $Y(j\omega_k)$, respectively. Define $\hat{G}(j\omega_k) = \dfrac{Y(j\omega_k)}{U(j\omega_k)}$.

Step d: Provide an initial simplex comprising $n + 1$ vertices $\mathbf{p}_0, \ldots, \mathbf{p}_n$.

Step e: Calculate objective function values $J_0, \ldots, J_n$ corresponding to $\mathbf{p}_0, \ldots, \mathbf{p}_n$, respectively. In this study, the definition of the objective function J is crucial, which is thoroughly explained in the following sections.

Step f: Go to Step 4 of the N-M simplex algorithm, which is described in the previous section.

Step g: If the termination criteria are satisfied, but the number of iterations is less than the set value, the optimal estimation $\mathbf{p}^*$ can be obtained. The optimal estimation $\mathbf{p}^*$ may not be the global optimal estimation. If the difference between the Bode diagram of $\hat{G}(\mathbf{p}^*)$ and the actual frequency response is extremely high, reselect another initial simplex and go to Step e.

Consider an LTI system, where an open-loop transfer function is unstable. For this unstable system, a controller must first be designed to stabilize the system through a trial and error process. The parameter estimation method is similar to the parameter estimation process for a stable system. The detailed estimation steps are as follows:

2. Unstable System

Step a: Design an appropriate feedback controller to stabilize the original system.

Step b: Let the reference signal be $r = 0$. Denote the original output signal of the controller by using u_c, which is a sinusoidal exciting signal, as $u_{ext}(t) = A_k \sin(\omega_k t)$, $k = 1, \ldots, N$. Assume $\omega_{k+1} > \omega_k$. The total control signal is written as $u_k = u_c + u_{ext}$.

Step c: Measure the steady output signal and denote the signal as $y_k(t) = B_k \sin\left(\omega_k t + \phi_k\right)$, $k = 1, \ldots, N$.

Step d: Perform FFT for $u_k(t)$ and $y_k(t)$ to obtain $U(j\omega_k)$ and $Y(j\omega_k)$, respectively. Define $\hat{G}(j\omega_k) = \dfrac{Y(j\omega_k)}{U(j\omega_k)}$.

Step e Provide an initial simplex comprising $n + 1$ vertices $\mathbf{p}_0, \ldots, \mathbf{p}_n$.

Step f: Calculate objective function values $J_0, \ldots, J_n$ corresponding to $\mathbf{p}_0, \ldots, \mathbf{p}_n$, respectively.

Step g: If the termination criteria are satisfied, but the number of iterations is less than the set value, the optimal estimation $\mathbf{p}^*$ can be obtained. The optimal estimation $\mathbf{p}^*$ may not be the global optimal estimation. If the difference between the Bode diagram of $\hat{G}(\mathbf{p}^*)$ and actual frequency response is extremely high, reselect another initial simplex and go to Step f.

In this study, the objective function applied in the simplex method is defined as follows:

$$J(\hat{\mathbf{p}}) \equiv w_{mag} \sum_{k=1}^{N/2} \lambda^k \left| \hat{G}(j\omega_k) - \tilde{G}(j\omega_k, \hat{\mathbf{p}}) \right|_{dB} + w_{phase} \sum_{k=1}^{N/2} \lambda^k \left| \angle \hat{G}(j\omega_k) - \angle \tilde{G}(j\omega_k, \hat{\mathbf{p}}) \right| \tag{4}$$

where ω_k denotes the k^{th} frequency. $\sum_{k=1}^{N/2} \lambda^k \left| \hat{G}(j\omega_k) - \tilde{G}(j\omega_k, \hat{\mathbf{p}}) \right|_{dB}$ can be considered a magnitude error function. $\sum_{k=1}^{N/2} \lambda^k \left| \angle \hat{G}(j\omega_k) - \angle \tilde{G}(j\omega_k, \hat{\mathbf{p}}) \right|$ is considered a phase error function. Because the units are different [i.e., decibel (dB) and degree (°)] in order to balance the effects of the two aforementioned functions within the objective function $J(\hat{\mathbf{p}})$, the magnitude error function and phase error function are multiplied by weightings w_{mag} and w_{phase}, respectively. Without loss generality, $w_{phase} = 1$; however, the weighting w_{mag} may be tuned based on different systems. Improved estimation results are generally observed when w_{mag} is higher than 10 according to our experience. Furthermore, because most physical systems (plants) have the common characteristic, such as low-pass filters, it implies a low SNR and high frequency of the exciting signal when the systems are disturbed by white or broadband noises. To improve the accuracy of parameter identification at high frequencies, weighting λ^k (parameter λ must satisfy the following condition: $0 < \lambda < 1$ and $k = 1, 2, \ldots, N/2$) is multiplied with the magnitude error function and phase error function to reduce the effect of high-frequency data. In this study, we choose $\lambda = 0.995$ for all simulations. Sometimes, if the errors of the simulations or experiments are extremely high in the Bode diagram, those data can be eliminated. The data outside the two yellow vertical lines are neglected in this research.

IV. NUMERICAL SIMULATION

To demonstrate the superiority of the proposed method, the simulation results of the proposed method in the frequency domain were compared with the results of the LS method (GLS method) in the time domain. The two methods are as follows:

Step 1. First, create a mathematical model for the real physical system (plant).
Step 2. Inject a white exciting signal $u(\cdot)$ into the system, and simultaneously measure the output signal $y(\cdot)$.
Step 3. Substitute the exciting signal (input signal) $u(\cdot)$ and the output signal $y(\cdot)$ in the LS and generalized LS algorithms to estimate the corresponding discrete transfer functions $\hat{G}(z^{-1})$ of the ARX and ARMAX models, respectively.
Step 4. Convert the discrete transfer function $\hat{G}(z^{-1})$ into a continuous transfer function $\hat{G}(s)$.

Throughout the simulation study, we let the common parameters be phase error weighting $w_{phase} = 1$, sampling points N = 2048, and λ = 0.995. The iterative process of the N-M simplex algorithm is continued until the iterations reach 500. Furthermore, only the estimated results by using the N-M simplex method are compared with the exact system in all the figures. The estimated results are not shown in the figures because the errors are extremely large or are negligible.

Case A: *Fourth-Order Stable LTI System (Relative Degree of 4)*

First, consider the following fourth-order stable LTI system with two different real poles and a pair of conjugate poles but no zeros (relative degree of 4):

$$\frac{1.275\times10^{7}}{s^4+45s^3+2350s^2+45500s+255000} \tag{5}$$

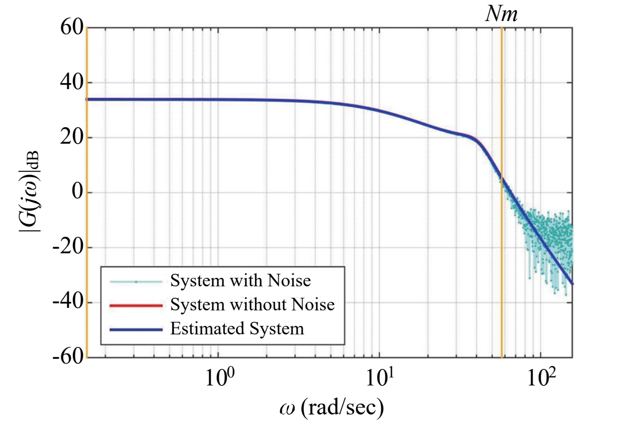

Fig. 1. Magnitude diagram of case A. (The data after the vertical line *Nm* are ignored throughout this research.)

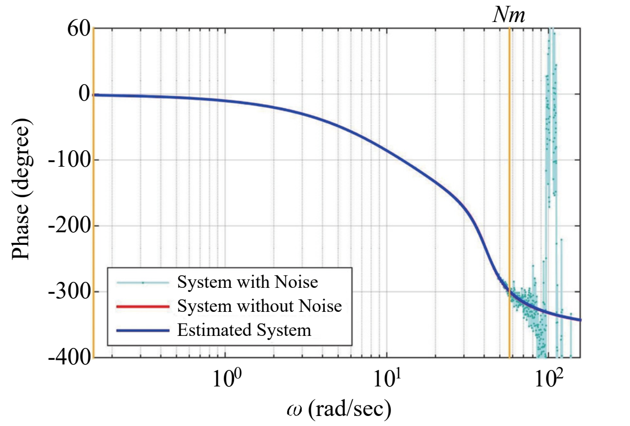

Fig. 2. Phase diagram of case A.

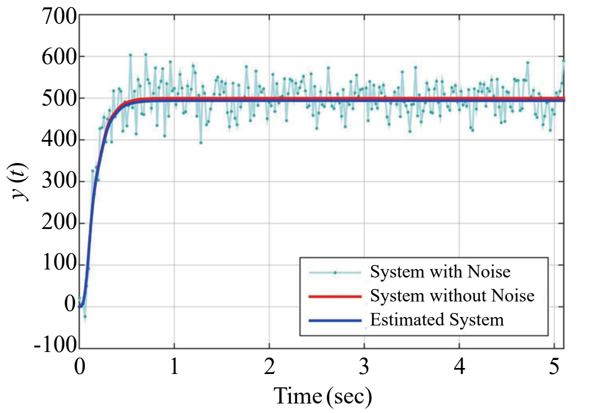

Fig. 3. Step response verification of case A.

A simple MATLAB code for this case is appended at the end of this paper. Table 1 and Figs. 1-3 provide simulation results, which are based on a sampling time of 0.02 s, $w_{mag} = 20$, and an SNR approaching 7. The proposed method is suitable for this example.

Table 1. Estimated results when the identification of ARX and ARMAX fails.

Exact Model	
Transfer function	$\dfrac{1.275\times10^{7}}{s^{4}+45s^{3}+2350s^{2}+4.55\times10^{4}s+2.55\times10^{5}}$
Zeros	-
Poles	$-10+j40$ $-10-j40$ -15 -10
N-M Simplex Method	
Transfer function	$\dfrac{1.266\times10^{7}}{s^{4}+45.99s^{3}+2364s^{2}+4.592\times10^{4}s+2.564\times10^{5}}$
Zeros	-
Poles	$-10.3282+j39.7774$ $-10.3282-j39.7774$ -15.596 -9.7349
ARX with Least-Square Method	
Transfer function	$\dfrac{0.336s^{4}-1988s^{3}+4.377\times10^{4}s^{2}-7.901\times10^{6}s+2.942\times10^{8}}{s^{4}+152s^{3}+2.203\times10^{4}s^{2}+1.165\times10^{6}s+7.344\times10^{6}}$
Zeros	5937.9 $-6.5+j65.9$ $-6.2-j65.9$ 33.9
Poles	$-40.35+j119.10$ $-40.35-j119.10$ -64.08 -7.25
ARMAX with Generalized Least-Square Method	
Transfer function	$\dfrac{0.1526s^{4}-9192s^{3}-1.185\times10^{5}s^{2}-1.543\times10^{7}s+4.486\times10^{8}}{s^{4}+232.9s^{3}+3.467\times10^{4}s^{2}+1.39\times10^{6}s+8.824\times10^{6}}$
Zeros	60243 $-17+j46$ $-17-j46$ 21
Poles	$-89.63+j129.07$ $-89.63-j129.07$ -45.91 -7.78

Table 2. Estimated results of case B, where the identification of ARX and ARMAX fails.

Exact Model	
Transfer function	$\dfrac{2s^2+40s+10000}{s^4+34.29s^3+3.371\times10^4 s^2+3.214\times10^5 s+7.143\times10^7}$
Zeros	$-10+j70$ $-10-j70$
Poles	$-12.94+j176.18$ $-12.94-j176.18$ $-4.2+j47.66$ $-4.2-j47.66$
N-M Simplex Method	
Transfer function	$\dfrac{1.987s^2+40.2s+9960}{s^4+35.53s^3+3.37\times10^4 s^2+3.326\times10^5 s+7.149\times10^7}$
Zeros	$-10.1185+j70.0817$ $-10.1185-j70.0817$
Poles	$-13.42+j176.06$ $-13.42-j176.06$ $-4.35+j47.69$ $-4.35-j47.69$
ARX with Least-Square Method	
Transfer function	$\dfrac{-3.743\times10^{-6}s^4+0.02496s^3+6.435s^2+522.7s+1.367\times10^6}{s^4+619.8s^3+4.131\times10^5 s^2+7.703\times10^7 s+1.01\times10^{10}}$
Zeros	69198 $991+j3276$ $991-j3276$
Poles	$-201.66+j500.45$ $-201.66-j500.45$ $-108.27+j151.53$ $-108.27-j151.53$
ARMAX with Generalized Least-Square Method	
Transfer function	$\dfrac{-3.392\times10^{-6}s^4-0.08342s^3+21.41s^2-1.57\times10^4 s+5.427\times10^6}{s^4+2101s^3+8.895\times10^5 s^2+3.279\times10^8 s+3.612\times10^{10}}$
Zeros	-24857 $-26+j452$ $-26-j452$ 314
Poles	-1680.5 $-134.4+j351.3$ $-134.4-j351.3$ -151.9

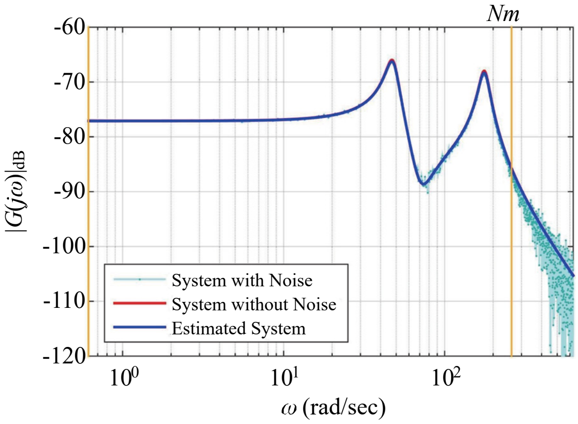

Fig. 4. Magnitude diagram of case B.

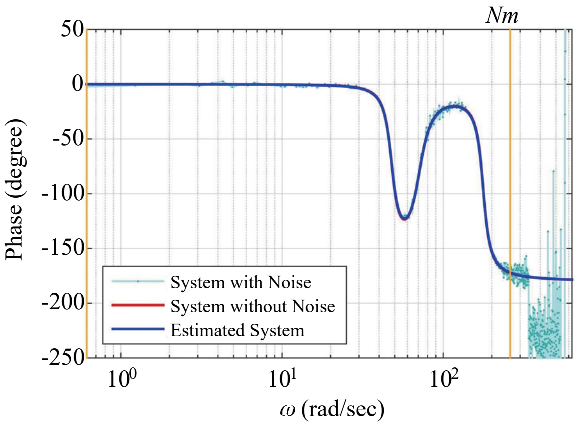

Fig. 5. Phase diagram of case B.

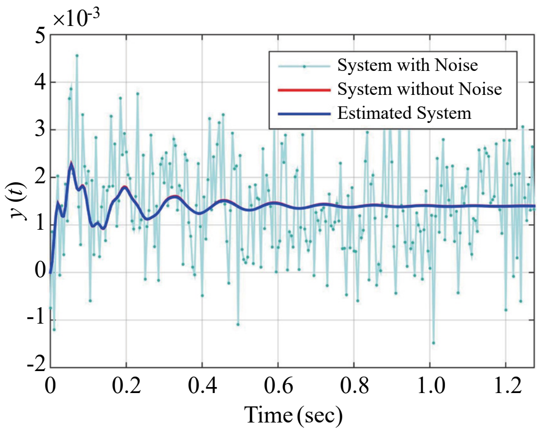

Fig. 6. Step response verification of case B.

Case B: *Fourth-Order Stable LTI System (Relative Degree of 2)*

Consider the fourth-order stable LTI system with the following transfer function, with two pairs of conjugate poles and one pair of conjugate zeros (relative degree of 2). The simulation parameters are as follows: an SNR of approximately 0.9, a sampling time of 0.005 s, and $w_{mag} = 20$.

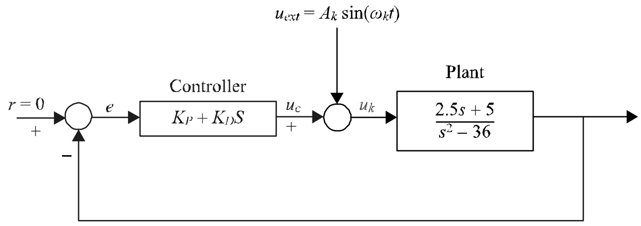

Fig. 7. Block diagram of case C with a feedback controller and sinusoidal exciting signal.

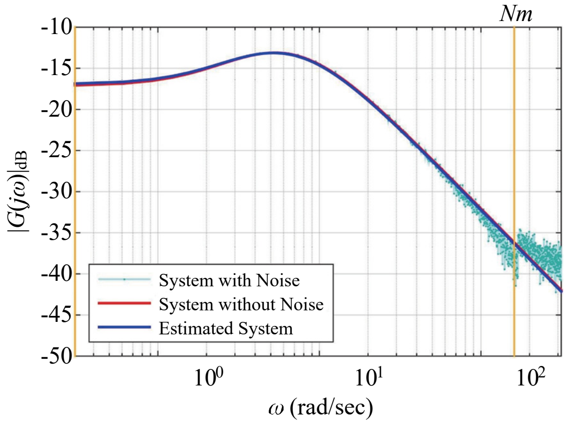

Fig. 8. Magnitude diagram of case C.

$$\frac{2s^2 + 40s + 10000}{s^4 + 34.29s^3 + 3.371\times10^4 s^2 + 3.214\times10^5 s + 7.143\times10^7} \tag{6}$$

Table 2 and Figs. 4-6 show the simulated results. The results indicate that the proposed approach is feasible. For the estimated data of the other two methods, their results are not provided in the figures because they are irrelevant.

Case C: *A Second-Order Unstable LTI System*

Consider a second-order single-input single-output LTI system with the following open-loop transfer function:

$$\frac{2.5s + 5}{s^2 - 36} \tag{7}$$

Because the open-loop system is unstable, its parameters cannot be identified through direct excitation. Therefore, a trial and error method is generally used to design a controller, which can stabilize the system and perform system parameter identification. When the system parameters are estimated, the controller with improved performance or high robustness can be redesigned based on the estimated system model, which is one of the primary aims of system parameter identification.

In this example, the trial and error method was applied to a proportional-derivative controller as $u_c \equiv K_P e + K_D \dot{e} = 3500e + 100\dot{e}$, where $e \equiv y - r$ and reference signal $r \equiv 0$. Fig. 7 shows the closed-loop system.

Table 3. Estimated results of case C, where the identification of ARX and ARMAX fails.

Exact Model	
Transfer function	$\frac{2.5s+5}{s^2-36}$
Zeros	-2
Poles	6 -6
N-M Simplex Method	
Transfer function	$\frac{2.445s+4.959}{s^2-34.98}$
Zeros	-2.0281
Poles	5.8141 -5.9141
ARX with Least-Square Method	
Transfer function	$\frac{0.002852s^3+4.066s^2-104.9s+4.337\times10^5}{s^3+227.7s^2+1.153\times10^5 s+4.879\times10^6}$
Zeros	-1515.9 $45.2+j313.5$ $45.2-j313.5$
Poles	$-91.03+j314.16$ $-91.03-j314.16$ -45.61
ARMAX with Generalized Least-Square Method	
Transfer function	$\frac{0.0006716s^3+1.875s^2+1816s+9.083\times10^5}{s^3+973.9s^2+3.415\times10^5 s+3.918\times10^6}$
Zeros	-1650.1 $-571.1+j702.4$ $-571.1-j702.4$
Poles	$-481.01+j314.16$ $-481.01-j314.16$ -11.87

$u_{ext} = A_k \sin(\omega_k t)$ denotes a sinusoidal exciting signal. The total control signal can be written as $u_k = u_c + u_{ext}$. In this example, the sampling time is 0.01 s, magnitude error weighting $w_{mag} = 10$, and the SNR is less than 1.0. Table 3 and Figs. 8-10 show the estimated results. This example validates that the proposed approach of using the N-M simplex method in the frequency domain is feasible for a system disturbed by large measurement noise.

CONCLUSION

In any identification method, avoiding parameter biases is difficult when the estimated systems are disturbed by measurement noise. When using conventional methods based on LS to identify the systems disturbed by large white/color noise, the estimated parameters may include severe errors and thus, the results become useless. For enabling control engineers to select or design more appropriate (or robust) controllers, obtaining

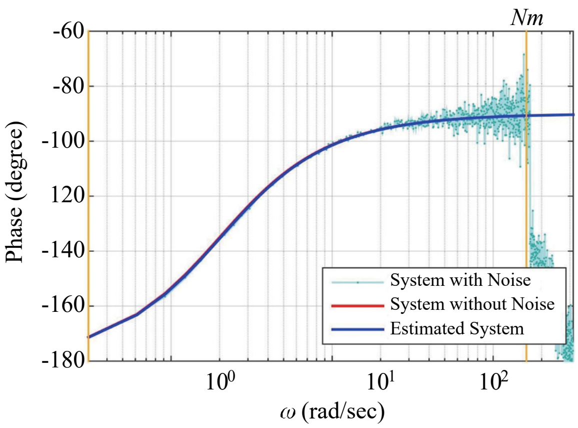

Fig. 9. Phase diagram of case C.

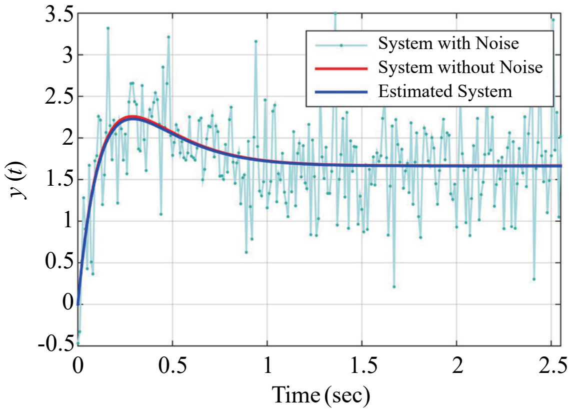

Fig. 10. Step response verification of case C.

accurate characteristics of the plants in advance is essential.

This study proposed an approach combining the N-M simplex method and the frequency response method to estimate systems with large measurement noises (or a low SNR). The simulation results show that by using the simplex method in the frequency domain, more accurate models and estimated systems with large measurement noises can be obtained.

The proposed algorithm can be easily implemented. Because the power of noise is distributed over a wide range, the effect of noise can be reduced to obtain more precise parameters. The N-M method does not require the derivatives of the objective function, and thus, it can be specifically applied to problems with discontinuities or an object function (or the system) that includes noise.

REFERENCES

Brigham, E. O. (1988). The Fast Fourier Transform and Its Application, Prentice-Hall.

Chelouah, R. and P. Siarry (2005). A hybrid method combining continuous tabu search and Nelder-Mead simplex algorithms for the global optimization of multiminima functions, European Journal of Operational Research 161, 636-654.

Fuh, C.-C. (2009). Detecting unstable periodic orbits embedded in chaotic systems using the simplex method. Communications in Nonlinear Science and Numerical Simulation 14, 1032-1037.

Hedar, A. R. and M. Fukushima (2006). Tabu Search directed by direct search methods for nonlinear global optimization. European Journal of Operational Research 170, 329-34.

Johansson, R. (1993). System Modeling and Identification, Prentice-Hall.

Keesman, K. J. (2011). System Identification: An Introduction, Springer Science & Business Media.

Ljung, L. (1987). System Identification, Prentice-Hall.

Luersen, M. A. and Le R. Riche (2004). Globalized Nelder-Mead method for engineering optimization, Computers and Structures 82, 2251-2260.

Nelder, J. A. and R. Mead (1965). A simplex method for function minimization. Computer Journal 7, 308-313.

Nevaranta, N., J.-H. Montonen, T. Lindh, M. Niemelä and O. Pyrhönen (2017). Recursive Parameter Estimation of a Mechanical System in Frequency Domain. IEEE 11th International Symposium on Diagnostics for Electrical Machines, Power Electronics and Drives, 122-128.

Norton, J. P. (1986). An Introduction to Identification, Academic Press.

Oppenheim, A. V., W. Schafer and J. R. Buck (1989). Discrete-Time Signal Processing, 2nd ed., Prentice-Hall.

Spendley, W., G. R. Hext and F. R. Himsworth (1962). Sequential application of simplex designs in optimization and evolutionary operation. Technomelrics 4, 441-461.

Van den Bosch, P. P. J. and A. C. van der Klauw (1994). Modeling, Identification and Simulation of Dynamical Systems, CRC Press.

Verhaegen, M. and V. Verdult (2007). Filtering and System Identification: A Least Squares Approach, Cambridge University Press.

APPENDIX

```
function Parameter_ID_via_Simplex_Method
global N_halves
global n_exact
global w2
global lambda
global db_exp phase_exp
global mag_weighting phase_weighting
global num_zeros den_zeros
N = 2^11;                               % Length of signal
N_halves = N/2;                         % Significant length of FFT
lambda = 0.995.^(1:N_halves)';          % Frequency weighting vector
```

```
ZERO = [];                                                % Zero(s) of the exact system
POLE = [-10 + 40j, -10-40j, -15, -10];                    % Pole(s) of the exact system
GAIN = 50.0;                                              % dc gain of the exact system
num_exact0 = poly(ZERO);                                  % Original numerator polynomial of the exact system
den_exact  = poly(POLE);                                  % Denominator polynomial of the exact system
num_exact  = num_exact0*GAIN*den_exact(end)/num_exact0(end);        % Modified numerator polynomial
                                                                    % of the exact system
dt = 0.02;                                                % (sec) Sampling period
Amp = 10;                                                 % Amplitude of the input sinusoidal signal
sigma = 40;                                               % Standard deviation of the output noise
fs = 1/dt;                                                % (Hz) Sampling frequency
f1 = fs/N;                                                % (Hz) Frequency resolution of the FFT
w1 = 2*pi*f1;                                             % (rad/sec) Frequency resolution of the FFT
w  = w1*(1:N)';                                           % (rad/sec) Create the frequency data (vector)
lambda(N_halves-650:end) = 0;                             % Neglect the data after (N_halves-650)*w1
mag_weighting = 20;                                       % Magnitude error weighting
phase_weighting = 1;                                      % Phase error weighting
w2 = w(1:N_halves);                                       % (rad/sec) Frequency grid
t  = (0:N-1)'*dt;                                         % (sec) Time grid
n_exact = length( den_exact )-1;                          % The order of the denominator polynomial
m_exact = length( num_exact )-1;                          % The order of the numerator polynomial
sys_exact = tf( num_exact, den_exact )
num_zeros = 0;                                            % Number of the zero(s) at the origin of the exact system
den_zeros = 0;                                            % Number of the pole(s) at the origin of the exact system

%--- Call Generate_Freq_Domain_Data to generate the frequency domain data of the exact system
[mag_exp, phase_exp] = Generate_Freq_Domain_Data( sys_exact, w, t, Amp, sigma );
db_exp = mag2db(mag_exp);                                 % Convert magnitude to decibels (dB)

%--- Simplex Method -----------------------------------------------------------------------
poly_den = poly( [-10+10j,-10-10j,-10,-10] );             % Denominator polynomial of the transfer function
poly_num = 56*poly_den(end);                              % Numerator polynomial of the transfer function
X0 = [ poly_den(2:end-den_zeros), poly_num(1:end-num_zeros)]

max_iter = 500;                                           % Maximum Iteration for the simplex algorithm
options = optimset('MaxIter', max_iter);                  % Create an options structure for fminsearch
X = fminsearch(@Obj_Function, X0, options);               % fminsearch is a MATLAB® optimization function
      % which can find minimum of unconstrained function using derivative-free (simplex) method

num_simplex = [X(1, n_exact-den_zeros + 1:end), zeros(1, num_zeros)];
den_simplex = [1, X(1, 1:n_exact-den_zeros), zeros(1, den_zeros)];
sys_simplex = tf(num_simplex, den_simplex)                % Create a continuous-time transfer function with
                          % numerator(s) and denominator(s) specified by num_simplex and den_simplex

%% === The function to generate the frequency domain data of the exact system ======================
function [mag_exp, phase_exp] = Generate_Freq_Domain_Data( sys_exact, w, t, Amp, sigma )
N = length(w);
N_halves = N/2;
mag_exp       = zeros(N_halves,1);
phase_exp     = zeros(N_halves,1);
a_u           = zeros(N_halves,1);
b_u           = zeros(N_halves,1);
a_y           = zeros(N_halves,1);
b_y           = zeros(N_halves,1);
% Inject a sinusoidal exciting signal, Amp*sin(w(k)*t), to the exact system, and measure the steady
```

```
% output signal, ak*cos(w(k)*t)+bk*sin(w(k)*t), in which ak and bk are estimated by fft.
for k = 1:N_halves
    u_w0 = Amp*sin(w(k)*t);                    % Real input signal without noise
    noise = sigma*randn(N, 1);                 % Random noise with standard deviation, sigma
    u_w = u_w0;
    y_exact_w = lsim(sys_exact, u_w, t);       % Exact output signal without noise
    y_exp_w = y_exact_w + noise;               % Real output signal with measurement noise
    fft_u = fft(u_w);
    a_u(k) = real(fft_u(k+1)*2/N);
    b_u(k) = imag(fft_u(k+1)*2/N);
    fft_y = fft(y_exp_w);
    a_y(k) = real(fft_y(k+1)*2/N);
    b_y(k) = imag(fft_y(k+1)*2/N);
    mag_exp(k) = fft_y(k+1)/fft_u(k+1);
    phase_exp(k) = atan2(b_y(k), a_y(k)) - atan2(b_u(k), a_u(k));
end
phase_exp = rad2deg(unwrap(phase_exp ) );
mag_exp = abs(mag_exp);

%% === The objective function applied in the simplex method ===================================
function index = Obj_Function(p)
global N_halves
global n_exact
global w2
global lambda
global db_exp phase_exp
global mag_weighting phase_weighting
global num_zeros den_zeros
den = [1, p(1:n_exact-den_zeros), zeros(1,den_zeros)];
num = [p(n_exact-den_zeros+1:end), zeros(1,num_zeros)];
[mag_simplex, phase_simplex] = bode(num, den, w2);
db_simplex = mag2db( mag_simplex );
index_db = norm(lambda.*(db_exp(1:N_halves)-db_simplex))^2;
index_phase = norm(lambda.*(phase_exp(1:N_halves)-phase_simplex))^2;
index = mag_weighting*index_db + phase_weighting*index_phase;
```