Real-time estimations of multi-modal frequencies for smart structures

Keun-Ho Rew[1], Sunmin Kim[2], In Lee[1] and Youngjin Park[2]

[1] Division of Aerospace Engineering, Department of Mechanical Engineering, Korea Advanced Institute of Science and Technology, Kusong-dong, Yusong-gu, Taejeon 305-701, Korea

[2] Division of Mechanical Engineering, Department of Mechanical Engineering, Korea Advanced Institute of Science and Technology, Kusong-dong, Yusong-gu, Taejeon 305-701, Korea

E-mail: inlee@asdl.kaist.ac.kr

Received 21 August 2000, in final form 14 May 2001
Published 6 February 2002
Online at stacks.iop.org/SMS/11/36

Abstract
In this paper, various methods for the real-time estimation of multi-modal frequencies are realized in real time and compared through numerical and experimental tests. These parameter-based frequency estimation methods can be applied to various engineering fields such as communications, radar and adaptive vibration and noise control. Well-known frequency estimation methods are introduced and explained. The Bairstow method is introduced to find the roots of a characteristic equation for estimations of multi-modal frequencies, and the computational efficiency of the Bairstow method is shown quantitatively. For a simple numerical test, we consider two sinusoids of the same amplitudes mixed with various amounts of white noise. The test results show that the auto regressive (AR) and auto regressive and moving average (ARMA) methods are unsuitable in noisy environments. The other methods apart from the AR method have fast tracking capability. From the point of view of computational efficiency, the results reveal that the ARMA method is inefficient, while the cascade notch filter method is very effective. The linearized adaptive notch filter and recursive maximum likelihood methods have average performances. Experimental tests are devised to confirm the feasibility of real-time computations and to impose the severe conditions of drastically different amplitudes and of considerable changes of natural frequencies. We have performed experiments to extract the natural frequencies from the vibration signal of wing-like composite plates in real time. The natural frequencies of the specimen are changed by added masses. Especially, the AR method exhibits a remarkable performance in spite of the severe conditions. This study will be helpful to anyone who needs a frequency estimation algorithm for real-time applications.

Nomenclature

ANF	adaptive notch filter
AR	auto regressive
ARMA	auto regressive and moving average
CNF	cascade notch filter
EachCNF	each error updated CNF
Fabc	case with three added masses
FFT	fast Fourier transform
FLOP	floating-point operations
Fx	case without added mass
LastCNF	last error updated CNF
LinANF	linearized adaptive notch filter
MA	moving average
PLL	phased locked loops
RLS	recursive least squares
RML	recursive maximum likelihood
SNR	signal-to-noise ratio
σ_N	variances of noise

0964-1726/02/010036+12$30.00 © 2002 IOP Publishing Ltd Printed in the UK

$\xi(n)$	white noise
ε	error between the measured and estimated signals
λ	forgetting factor
λ_i	initial value of forgetting factor
λ_f	final value of forgetting factor
ρ	pole contraction factor
ρ_i	initial value of pole contraction factor
ρ_f	final value of pole contraction factor
$\phi(k)$	measured signal vector
$\theta(k)$	unknown parameter vector
σ_S	variances of signal
$A(z^{-1})$	AR part, denominator of any transfer function, i.e. characteristic equation
$B(z^{-1})$	MA part, numerator of transfer function
Err	discrepancy between the true and estimated frequencies
$H(z^{-1})$	arbitrary transfer function
k	time index denoting the current state
$N(z^{-1})$	transfer function of adaptive notch filter
p	vibration mode number
$\mathbf{P}(k)$	covariance matrix
P_i	ith pole
T_s	sampling time
z^{-1}	shifting operator
Z_i	ith zero

1. Introduction

The frequency estimation problem has been an interesting issue since Fourier developed his transformation method in 1822. Since the 1970s, many algorithms have been developed for the on-line estimation of multi-modal frequencies in the fields of radar, communications and adaptive signal processing [1–4]. The on-line information of natural frequencies can be applied to adaptive noise controllers [5, 6], or to the adaptive vibration controller of time-varying structures [7], such as deployable space structures and the robot manipulator dealing with time-varying payloads. Because the local failure and delamination of composite structures tend to reduce natural frequencies, the on-line detection of the natural frequencies is also important in health monitoring. In addition, real-time frequency estimation for smart materials and structures with changing natural frequencies makes them smarter in view of both control and monitoring.

In real-time frequency estimation, important topics are computational burden, accuracy, frequency resolution and robustness to random noise. A heavy computational burden interrupts the finding of the root of a characteristic equation or solving an eigenvalue problem in the subspace method. Even though the digital signal processing technology has improved the computational speed, the frequency estimation within each time step is not so manageable. Therefore, fast and reliable algorithms have been investigated for real-time frequency estimations [5].

There are two main approaches for the frequency estimation. Firstly, non-parametric methods are considered to be more accurate, but these cause a heavy computational burden and leakage, since the algorithms are based on the fast Fourier transform. Secondly, parametric methods use predetermined structures, then estimate the model parameters of the system. These parametric methods are relatively fast and have high resolution despite a short data record [2, 3]. Therefore, parametric methods have been investigated actively for the fast system identification.

Chaplin and Smith [8] patented an electronic circuit containing phased locked loops (PLL) for the frequency estimation. For parametric methods, Nehorai [9] has suggested and shown the advantages of the constrained-pole-zero structure and used the recursive maximum likelihood (RML) method for parameter estimations. Tranvassor-Romano and Bellanger [10] developed the linearized adaptive notch filter (LinANF) method. Kim and Park [5] have developed an on-line multi-tonal noise estimation method applied for an active noise control, called the cascade notch filer (CNF) method, which is used for generating a reference signal. Rew *et al* [7] have applied the auto regressive (AR) method to the adaptive positive position feedback (PPF) vibration control. Rhim and Book [12] have developed an adaptive input shaping technique for robotics using the on-line frequency estimation of a single mode, and the other methods [2, 3, 13] are now being developed. Kay has compared several frequency estimation methods well in his textbook [16]. We survey parameter-based frequency estimation methods, through simulation and experiments, which are useful for the on-line problem due to a lower computational burden.

We adopt the Bairstow method to find the root, which enhance the computational efficiency drastically, and we compare several frequency estimation methods through simple numerical and severe experimental tests. Various methods are investigated as follows: AR, auto regressive and moving average (ARMA), RML, LinANF, each-error-updated CNF (EachCNF), and last-error-updated CNF (LastCNF). The above methods are named after the predetermined structures or parameter estimation method and the abbreviations are used for the sake of simplicity.

2. Background to frequency estimation methods

2.1. Problem definition

Frequency estimation methods are based on the adaptive signal processing theory [2, 3]. In general, the estimation of multi-modal frequencies is defined as the problem of finding frequencies, $f_i(n)$, of sinusoids buried in white noises, $\xi(n)$ as the following equation (1):

$$x(n) = \sum_{i=1}^{p} U_i \sin(2\pi f_i(n) n T_s + \phi_i) + \xi(n). \qquad (1)$$

Six frequency estimation methods are compared from the point of view of computational burden, tracking capability and robustness to environmental noise.

Table 1. Computational burden of root-finding methods.

Order of characteristic equation	Bairstow method (FLOP)[a]	Eigenvalue method (FLOP)	Bairstow/eigenvalue (%)
Fourth order (two modes)	516	1998	25.8
Sixth order (three modes)	2146	6863	31.3
Eighth order (four modes)	2930	14 848	19.7
Tenth order (five modes)	3170	24 273	13.1

[a] FLOP: Floating-point operations.

2.2. AR and ARMA methods

These methods are composed of two steps. Firstly, we estimate the coefficients of a linear transfer function without inversing matrices by the recursive least squares (RLS) method, which gives good convergence from arbitrary initial estimations. Secondly, we find the roots of the characteristic equation by decoupling the characteristic equation into each mode using the Bairstow method.

If a system is modeled to have p modes, the characteristic equation of the system can be described as $2p$ order. A transfer function of an arbitrary structure can be described in the even-order discrete form as follows:

$$H(z^{-1}) = \frac{B(z^{-1})}{A(z^{-1})} = \frac{b_1z^{-1} + b_2z^{-2} + \cdots + b_{2p}z^{-2p}}{1 + a_1z^{-1} + a_2z^{-2} + \cdots + a_{2p}z^{-2p}} \tag{2}$$

where p is the vibration mode number, z^{-1} is a shifting operator, and $A(z^{-1})$ and $B(z^{-1})$ are the AR and MA parts, respectively. The roots of $A(z^{-1})$ and $B(z^{-1})$ are poles and zeros of the system. The AR and ARMA methods are named after the structure of each model.

We are interested in the denominator, $A(z^{-1})$, since the natural frequencies are only related to $A(z^{-1})$. Therefore, the AR method estimates only AR coefficients, neglecting MA. On the other hand, the ARMA method estimates both MA and AR coefficients. Though the MA part is not used in frequency detection, the ARMA method estimates frequencies with robustness to noise, as is shown later.

The RLS method is used to obtain the coefficients of the AR and ARMA models with the weighted cost function to error squares as follows:

$$V = \sum_{j=0}^{k} \lambda^{k-j}\varepsilon^2(j) \tag{3}$$

where k is the current time index, ε is an error between the measured and estimated signals and λ is a forgetting factor to weigh relative importance between the past and present error squares. Details of the procedure of the RLS method are given in appendix A.

In order to obtain natural frequencies from the coupled characteristic equation, $A(z^{-1})$, the equation must be factorized into second-order equations. Abel and Galois have proven that algebraic polynomial equations higher than the fourth order cannot have a closed-form solution, in general. Therefore, only iterative algorithms can factorize the characteristic equation that has more than two modes.

The Bairstow method [14] is a root-finding algorithm for the even-order algebraic equation developed by Bairstow and Hitchcock. With the Bairstow method, we can factorize the AR part as the products of second-order equations as follows:

$$A(z^{-1}) = z^{-2p}\prod_{i=1}^{p}\left(z^2 + p_iz^1 + q_i\right) = 0. \tag{4}$$

Then, the natural frequencies can be obtained from the following equation, which is the result of z-transformation

$$\begin{aligned}\hat{f}_i &= \frac{1}{2\pi T_s}\text{angle}\left(Z_i|A(z^{-1}) = 0\right)\\ &= \frac{1}{2\pi T_s}\left[\cos^{-1}(-p_i/2\sqrt{q_i})\right]\end{aligned} \tag{5}$$

where T_s is the sampling time.

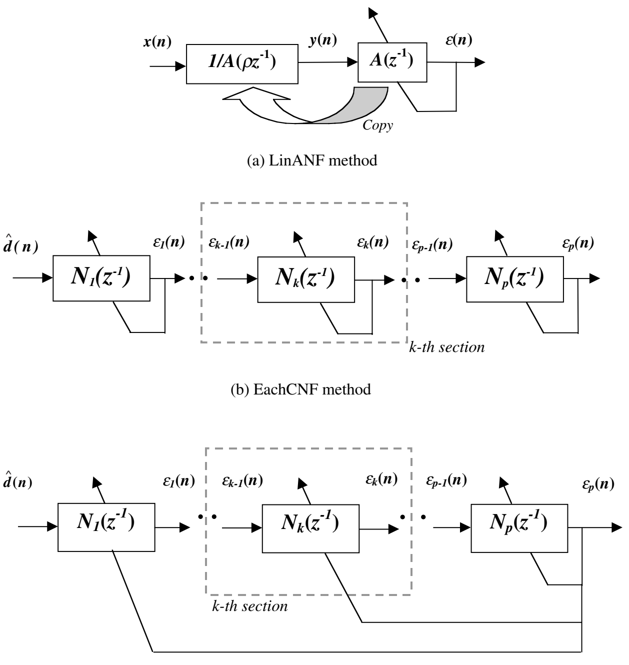
Finally, the obtained natural frequencies are sorted by magnitudes using a simple straight insertion algorithm [14]. The detailed derivation of the Bairstow method is referred to in [14, 15], and the procedure of the Bairstow algorithm is explained in appendix B.

We adopted this Bairstow method in [7] to decouple the characteristic equation in frequency estimation problems. The eigenvalue method contained in the 'roots' function of MATLAB® has often been used in previous approaches. Table 1 shows that the computational burden of the Bairstow method is much less than that of the eigenvalue method. It is neither feasible nor practical to count all computational operations, but most of the important operations can be estimated through the floating-point operation (FLOP). In table 1, FLOP is closely proportional to the computational time. But, when the computational burdens are compared in the FLOP dimension, the computational burden of exponential, sin and other transcendental functions is considered equally as the burden of simple multiplications. Since the Bairstow method uses only plus, minus, multiplication and division operations, this method is relatively much faster than the eigenvalue method, as shown in table 1. Furthermore, it uses only real variables to save memory and gives good convergence from arbitrary initial parameters [14]. In the parametric methods, the adoption of the Bairstow method has made a breakthrough regarding computational efficiency.

2.3. RML method

The RML method using the constrained poles and zeros structure has been suggested by Nehorai [9]. The structure can be written as follows:

$$\begin{aligned}N(z^{-1}) &= \frac{A(z^{-1})}{A(z^{-1})} = \frac{1 + a_1z^{-1} + a_2z^{-2} + \cdots + a_{2p}z^{-2p}}{1 + a_1z^{-1} + {}^2a_2z^{-2} + \cdots + {}^{2p}a_{2p}z^{-2p}}\\ a_{2p} &= 1 \qquad a_i = a_{2p-i} \quad i = 1, \ldots, p\end{aligned} \tag{6}$$



(a) LinANF method

(b) EachCNF method

(c) LastCNF method

Figure 1. Block diagram of each estimation method: (a) LinANF method; (b) EachCNF method; (c) LastCNF method.

where ρ, called a pole contraction factor, is a positive real number close to but smaller than 1.0 and is related to the bandwidth of the notch. Note that the locations of zeros and poles are directly related as follows:

$$P_i = \rho^{-1} Z_i \qquad i = 1, 2, \ldots, p \tag{7}$$

where P_i is the ith pole and Z_i is the ith zero of $N(z)$.

Then, the RML method uses the RLS scheme to estimate the parameters. Nehorai has shown that the performance of the RML method is better than the ARMA method for pure sinusoidal signals. In addition, this RML method is mathematically well established. The procedure for this method is summarized in appendix C.

2.4. LinANF method

The LinANF method is famous for computational efficiency, fast tracking capability and low threshold signal-to-noise ration (SNR). Kim and Park [5] have compared the RML and LinANF methods with respect to convergence speed and noise robustness aspect, and concluded that the LinANF method is more effective.

The LinANF method also adopts the structure of constrained poles and zeros, but the estimation procedure is slightly different. The basic concept of this algorithm is to separate the numerator and the denominator of the adaptive notch filter (ANF) and update only the numerator, then to copy the coefficients to the denominator, as seen in figure 1(a). The computational procedure of this algorithm is reduced by $8p + 7$ additions and $21p + 10$ multiplications, compared to the RML method.

The detailed procedure for this method has been presented in [5], and is given in appendix D.

2.5. CNF method

The CNF method for multi-tonal noise control has been proposed in [5, 11, 17] to reduce the computational burden significantly. This method originates from the following simple idea. The second-order notch filter notches 1 sinusoid. We connect the p second-order notch filters in series to notch p sinusoids. If the parameters of notch filters are adapted properly, then the notched signal will be nearly zero; otherwise they are adapted iteratively until the notch filter deducts the appropriate sinusoid. This repetition of adaptation makes the parameters converge to true ones. Neither a matrix calculation nor a root-finding algorithm is needed. This feature reduces the computational burden significantly. The CNF method has a similar structure to the LinANF method.

The CNF method is classified as the EachCNF and LastCNF methods [11], according to error updating methods, as illustrated in figures 1(b) and (c). The EachCNF method has p sections that notch each tonal signal with each error as equation (E.5a) in appendix E, though all the sections of the LastCNF method are updated with the last error as equation (E.5b). This last error update allows the LastCNF method to avoid frequency conflicts. However, the computational burden is slightly increased. The detailed procedure is presented in appendix E.

2.6. Common techniques

The coefficients of $A(z^{-1})$ amounting to $2p$ can be reduced to p ones with mirror symmetry characteristics as follows:

$$a_{2p} = 1 \qquad a_i = a_{2p-i} \quad i = 1, \ldots, p. \tag{8}$$

This symmetric structure assumes that the signal is composed of pure sinusoids, and this property also accelerates the convergence. Aside from the AR and ARMA methods, this mirror symmetry is applied to all methods.

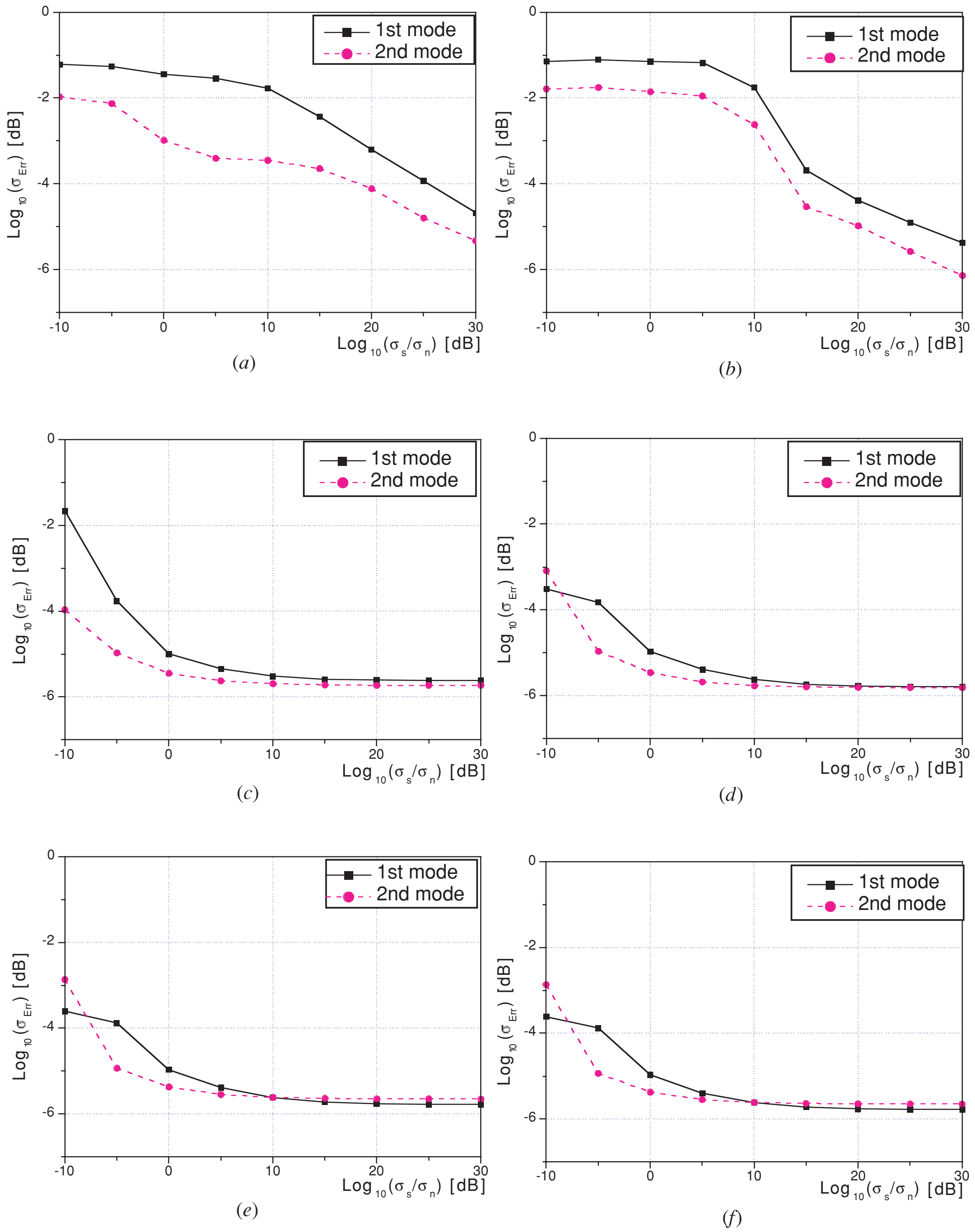
For all the methods considered above, the smaller the forgetting factor λ, the more adaptable the RLS method becomes to the changes of system dynamics; but it also becomes more sensitive to noise. As the pole contraction factor ρ increases, the estimation does not fail but tends to be accurate. Therefore, λ and ρ are exponentially increased for a fast convergence and an accurate estimation from arbitrary initial parameters is as follows:

$$\rho(n+1) = \rho_f \rho(n) + (1 - \rho_f) \qquad \rho(0) = \rho_i \tag{9a}$$
$$\lambda(n+1) = \lambda_f \lambda(n) + (1 - \lambda_f) \qquad \lambda(0) = \lambda_i \tag{9b}$$

where ρ_i and λ_i are the initial values, and ρ_f and λ_f are the final values, respectively. This increasing scheme is applied to all algorithms where these values are used.

2.7. Other methods

There are several methods that have not been mentioned above, such as the eigenspace method, the Pisarenko method, the multiple signal classification (MUSIC) method, the estimation of signal parameters via rotational invariance technique (ESPRIT) method [2, 3], etc. Comparisons with other methods are left for future study.



(*a*)

(*b*)

(*c*)

(*d*)

(*e*)

(*f*)

Figure 2. Variance of estimation error with respect to the SNR: (*a*) AR method; (*b*) ARMA method; (*c*) RML method; (*d*) LinANF method; (*e*) EachCNF method; (*f*) Last CNF method.

3. Numerical comparison of performances

We have compared the frequency estimation methods mentioned above for multiple sinusoids buried in white noise. A simple test is performed to compare various performance indices. This case study reveals the performances of each method.

3.1. Definitions of performance indices

A jargon of the SNR is expressed as follows:

$$\mathrm{SNR} \triangleq 10\mathrm{Log}_{10}(\sigma_S/\sigma_N) \tag{10}$$

where σ_S and σ_N are the variances of signal and noise, respectively. As the SNR increases linearly, the signal with respect to noise increases exponentially. Err is defined as a discrepancy between the real frequency and the estimated frequency

$$\mathrm{Err} \triangleq f_i - \hat{f}_i. \tag{11}$$

The estimation of errors with respect to the SNR are given in figure 2. As the noise increases, Err increases, and the graph tends to move from the lower right to the upper left. At this time, Err increases drastically at a certain point. The x-axis value at this point is referred to as the threshold SNR.

Table 2. Parameters for numerical test.

		Frequency[a] (Hz)	Amplitude
Mode	First mode	200 → 300	1.0 → 1.0
	Second mode	300 → 400	1.0 → 1.0
		Forgetting factor $\lambda_i \rightarrow \lambda_f$	Pole contraction factor $\rho_i \rightarrow \rho_f$
Estimation method	AR	0.98 → 0.99	None
	ARMA	0.98 → 0.99	None
	RML	0.98 → 0.99	0.85 → 0.999
	LinANF	0.98 → 0.99	0.85 → 0.999
	EachCNF	0.85 → 0.98	0.85 → 0.98
	LastCNF	0.85 → 0.98	0.85 → 0.98

[a] Sampling frequency: 1000 Hz.

Table 3. Computational burden and threshold SNR for each method.

Estimation method	Computational burden (FLOP)	Threshold SNR (dB)	Root finding method
AR	516	15	Bairstow
ARMA	2313	15	Bairstow
RML	778	5	Bairstow
LinANF	657	0	Bairstow
EachCNF	78	0	None
LastCNF	101	0	None

The threshold SNR can be the performance index for frequency estimation with noise. The small threshold SNR implies that the application of some algorithms in a noisy environment is valid.

3.2. Simple numerical test

For a typical performance test, six methods were simulated for two sinusoids buried in a noise having equal amplitudes and time-varying frequencies. The first and second frequencies vary from 200, 300 Hz to 300, 400 Hz, respectively. The forgetting factors and the pole contraction parameters increase from the initial values λ_i, ρ_i to the final values λ_f, ρ_f, respectively, as shown in equation (9). The frequencies and the other parameters are introduced in table 2.

The SNR versus logarithmic variance of Err is illustrated in figure 2 and the threshold SNR is given in table 3. The noise robustness of the AR and ARMA methods is not as good as that of the other methods, as shown in figure 2 and table 3. This result implies that the AR and ARMA methods are not suitable for noisy environments.

Figures 3–8 present frequency estimation results for changing frequencies when the SNR is 15 dB. Figures 3–8 show that the Err variance of the first mode is greater than that of the second mode. This is due to the fact that the second sinusoid has a tendency to converge faster, and this results from the fact that the higher frequency signal has more information in the same length of record. The tracking capability is determined based on the time domain results in figures 3–8. The tracking capability is dependent on the tuning parameters, λ_i, ρ_i, λ_f and ρ_f. Five algorithms have fast tracking capability except for the AR method. Therefore the AR method is not selectable for the fast changing system.

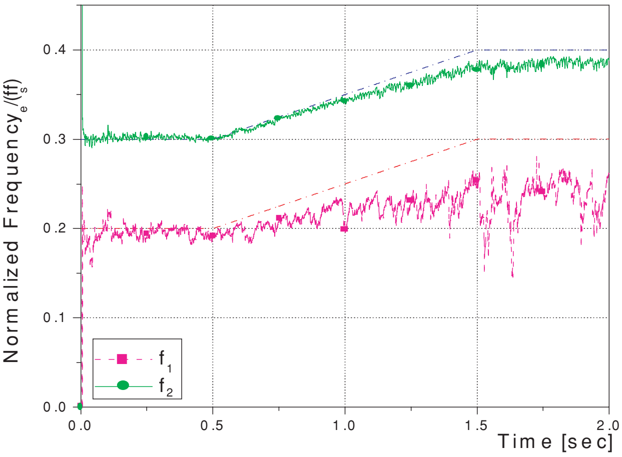


Figure 3. Frequency estimation using the AR method with a SNR of 15 dB.

The computational burden is compared in table 3. When the computational burden is critical, the ARMA method is unsuitable, and the CNF methods are strongly recommended. Note that the AR method needs approximately a quarter of the burden of the ARMA method, because the number of parameters needed for estimation is halved.

Other cases, such as two sinusoids of crossing frequencies, were simulated but the results are omitted here. The omitted results do not affect the tendency of performances.

4. Experimental comparison of performances

The frequency estimation algorithms must be applied not only to a simple case as previous studies, but also to a severe case, which is more realistic. Experimental tests are devised to confirm the feasibility of real-time computations, and to impose strict conditions considering drastically different amplitudes and considerable changes of natural frequencies. These conditions have not been considered seriously in the previous researches.

4.1. Experimental procedure

A wing-like composite plate is manufactured, as shown in figure 9. The stacking sequence of the specimen is $[0_2/90_3]_s$,

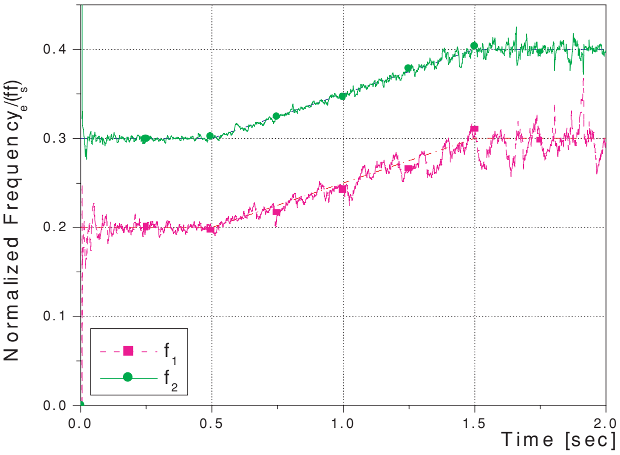


Figure 4. Frequency estimation using the ARMA method with a SNR of 15 dB.

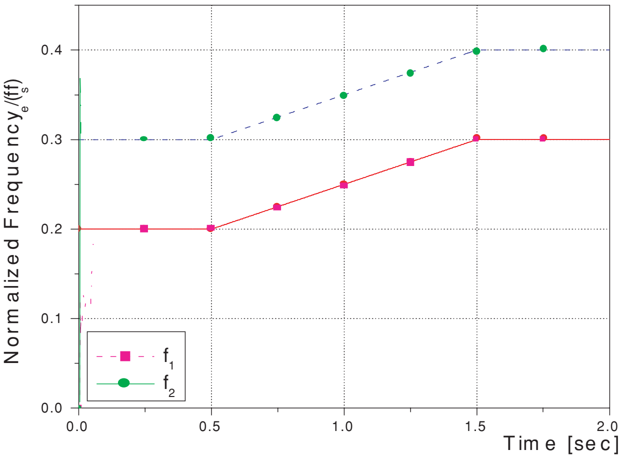


Figure 5. Frequency estimation using the RML method with a SNR of 15 dB.

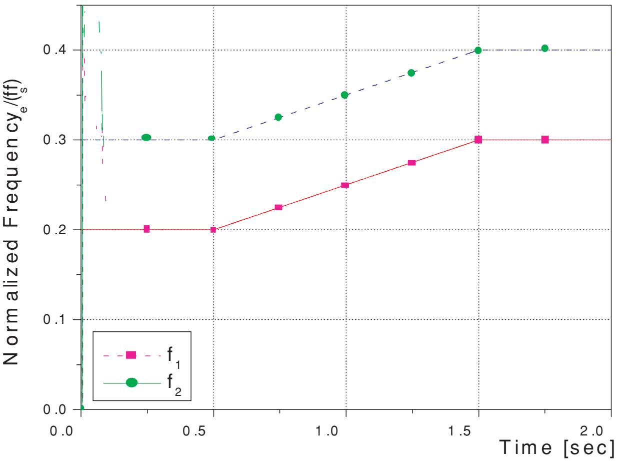


Figure 6. Frequency estimation using the LinANF method with a SNR of 15 dB.

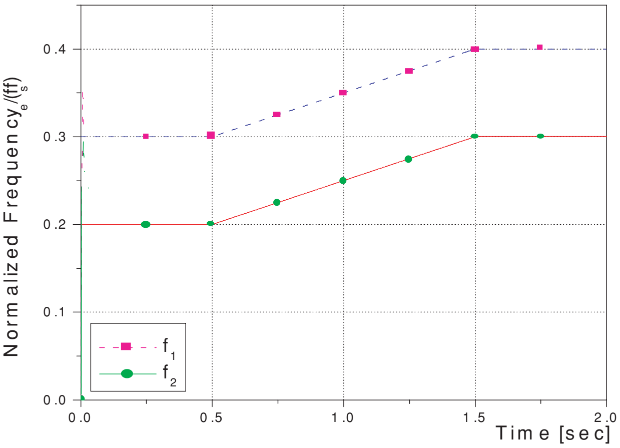


Figure 7. Frequency estimation using the EachCNF method with a SNR of 15 dB.

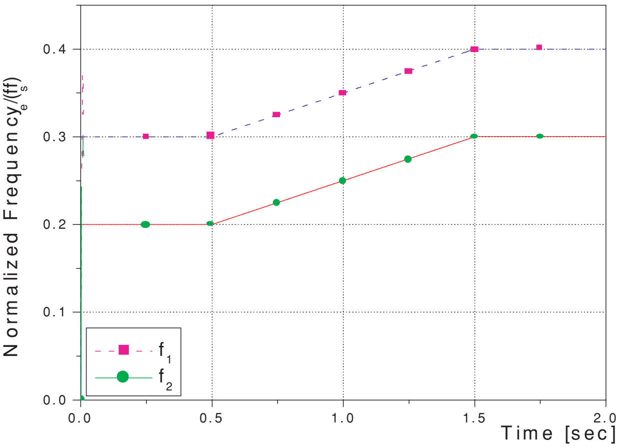


Figure 8. Frequency estimation using the LastCNF method with a SNR of 15 dB.

and the sweep back angle is 9.36°. PZT and PVDF denote the piezo-ceramic actuator and piezo-film sensor, respectively. The piezoelectric sensor and actuator are bonded near the root of the specimen, where the strain is relatively large. Magnetic masses are attached to the wing-like specimen at the leading edge, as shown in figure 9. The added masses can be considered as wing stores such as a fuel tank, a missile and a sidewinder. Each of the added masses amounts to 1.62 g, and the three added masses weigh approximately 12.0% of the specimen. This experiment is devised to show the adaptation ability with respect to the variations of natural frequencies. For simplicity, the cases Fx and Fabc are named after the position of the added masses where Fx represents the case without added mass and Fabc represents the case with added masses at Fa, Fb and Fc. Off-line frequency responses using a fast Fourier transform (FFT) analyser, as seen in figure 10, show that the amplitudes of the first three modes have the amplitude ratio of 6.7:1.1:1.9 in the Fx case. The added masses make the first three natural frequencies decrease by as much as 7.8, 13.5 and 10.1%, respectively.

The experimental setup has been prepared as shown in figure 11. Each estimation algorithm is implemented in the digital signal processing (DSP) board (dSPACE® DS1102), in which the DSP chip (Texas Instrument® TMS320-C31) is used as the main processor. A charge amplifier converts the generated charge signal due to the vibration in the piezoelectric film sensor to the electric voltage signal, and the converted signal is fed into the DSP controller. Then, the estimation

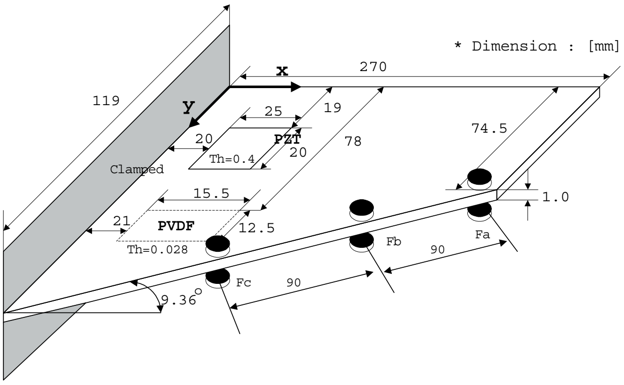


Figure 9. Configuration of specimen.

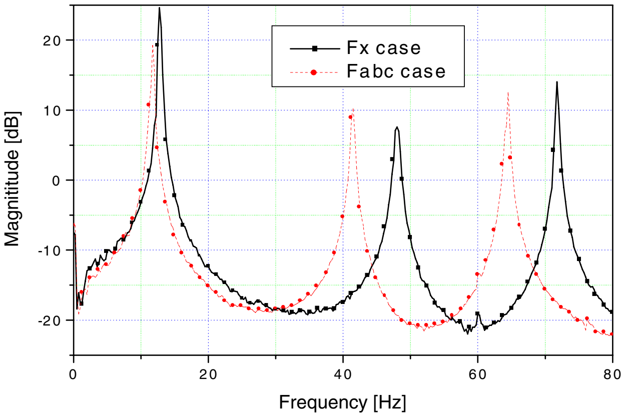


Figure 10. Change of frequency responses by added masses.

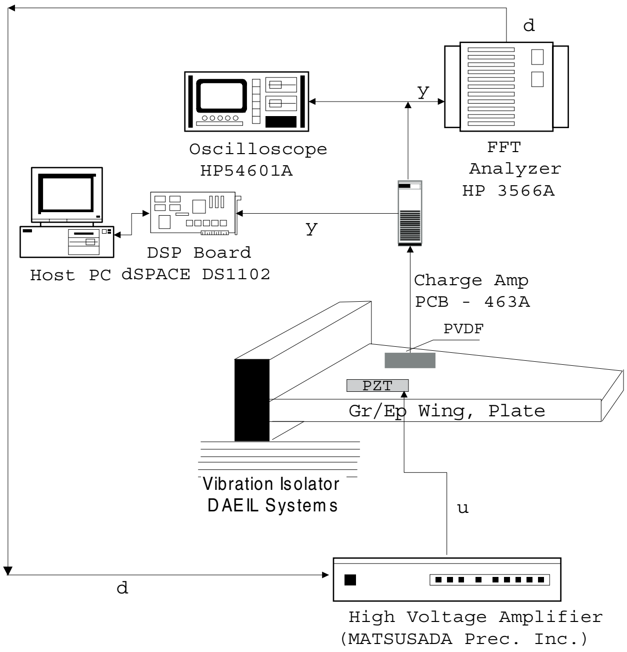


Figure 11. Overall experimental setup.

signal is calculated using the above methods. The source module of the FFT analyser generates an external disturbance that is a banded white signal. This disturbance signal is applied to the piezoelectric actuator after 500 times voltage amplification. The sensor signal is captured in the DSP board and monitored from the host PC during the experiments.

Since the frequency estimation algorithms used in this paper extract modal information from the highest mode they can catch, which is the third mode in this study, the shortest sampling time is not always the best solution. As a result of rough compromise between an aliasing effect and an estimation of unintended higher mode, the time step of frequency estimation, T_s, is selected as 5.0 ms. The frequencies and the other parameters for the experimental test are shown in table 4, in which the notation is the same as table 2.

4.2. Estimation results of multi-modal frequencies

Because the estimated signal depends on the three-mode model, higher-order modes can spoil the estimations. Nonetheless, the frequency estimation performance has a reasonable performance, as shown in figure 12.

The estimation of the first and third modes is relatively precise, but the second mode frequency is somewhat incorrect because the second mode contributes a small amount in the total vibration, as in figure 10. As noted in figure 12, the third mode frequency converges quickly, whereas the first mode converges slowly. This originates from the fact that the vibration signal of a faster mode contains much information in the same number of data. Added masses make the natural frequencies decrease as mentioned above. Those characteristics can be seen in figure 12(*b*). It can be observed that the changes of the frequencies do not deteriorate the estimation performance.

In figure 13, we observe that the ARMA method has a relatively slow convergence compared to the AR method, and this ARMA method exhibits a moderate performance in spite of the frequency changing.

Regalia [3] pointed out through a numerical example that the frequency estimation of three modes is much more difficult than that of two modes. The estimated frequencies can easily converge into one frequency repeatedly, or can fluctuate. Mostly, it is more difficult to detect natural frequencies from the vibration signal of structures having quite different amplitudes, than to detect the frequency of pure two sinusoids with similar amplitudes.

The other methods do not perform well under severe experimental tests, as shown in figure 14. This is due to the fact that the other methods assume pure sinusoids with similar amplitudes as mentioned previously. Although the other methods are faster, they can be misled in the severe case. The frequency estimation results showing divergence are omitted here.

5. Conclusions

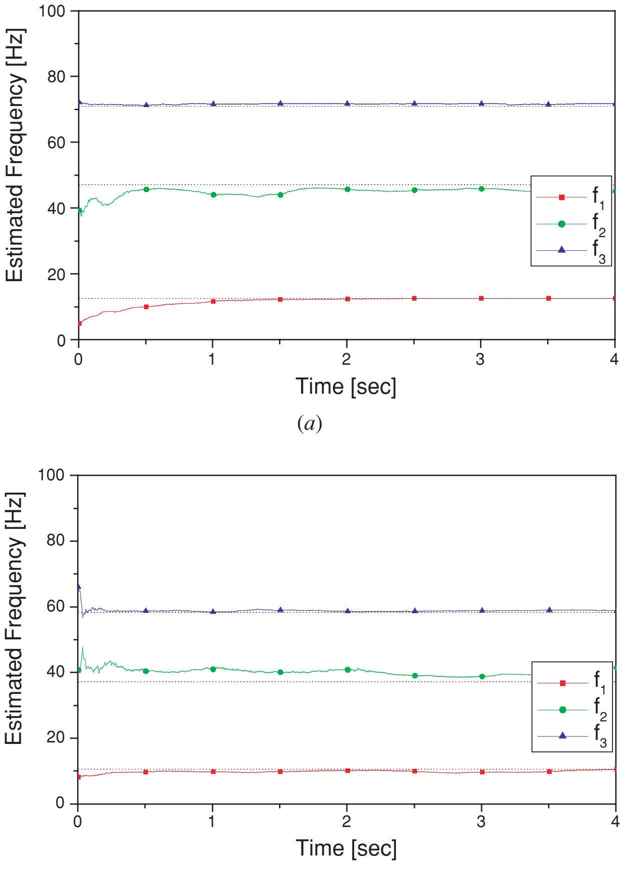
In this paper, various real-time estimation methods of multi-modal frequencies are realized and compared through numerical and experimental tests. This paper will be helpful to anyone who needs a frequency estimation algorithm to implement in real time.

The following methods have been introduced and examined: AR, ARMA, RML, LinANF, EachCNF and LastCNF. We have introduced the Bairstow method to

Table 4. Parameters for experimental test.

		Frequency[a](Hz) (Fx → Fabc)	Amplitude (Fx → Fabc)
Mode	First mode	12.75 → 11.75	3.42 → 3.62
	Second mode	48.0 → 41.5	1.46 → 1.67
	Third mode	71.75 → 64.5	2.02 → 1.87
		Forgetting factor $\lambda_i \rightarrow \lambda_f$	Pole contraction factor $\rho_i \rightarrow \rho_f$
Estimation method	AR	0.95 → 0.995	None
	ARMA	0.95 → 0.995	None
	RML	0.98 → 0.99	0.85 → 0.999
	LinANF	0.98 → 0.99	0.85 → 0.999
	EachCNF	0.85 → 0.98	0.85 → 0.98
	LastCNF	0.85 → 0.98	0.85 → 0.98

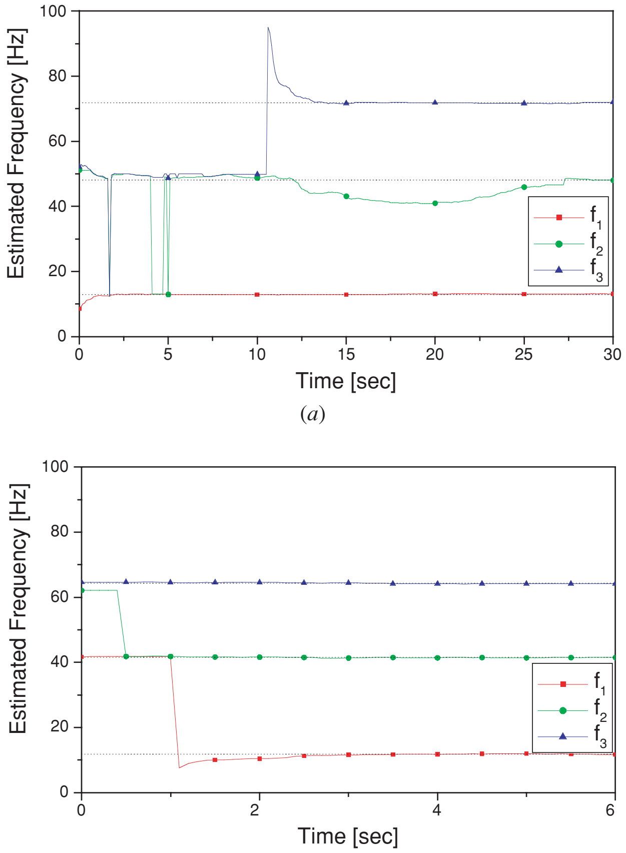
[a] Sampling frequency: 200 Hz.



(*a*)

(*b*)

Figure 12. Frequency estimation using the AR method for each case: (*a*) Fx case; (*b*) Fabc case.



(*a*)

(*b*)

Figure 13. Frequency estimation using the ARMA method for each case: (*a*) Fx case; (*b*) Fabc case.

decouple the characteristic equation during the frequency estimation procedure and have revealed the computational efficiency quantitatively.

A signal of two sinusoids mixed with white noise is considered numerically on the basis of threshold SNRs, tracking capability, and computational burden with FLOP dimension. We concluded from the test results that the AR and ARMA methods are not recommended in noisy environments. Except for the AR method, the five methods have fast tracking capability. Our results reveal that the ARMA method is inefficient with regards to computational efficiency; however, the EachCNF and LastCNF methods are fairly effective. The LinANF and RML methods have average performances.

We have performed experiments to extract natural frequencies from the vibration signal of a wing-like composite plate with a piezoelectric sensor and an actuator. The six methods mentioned above are realized in real time using

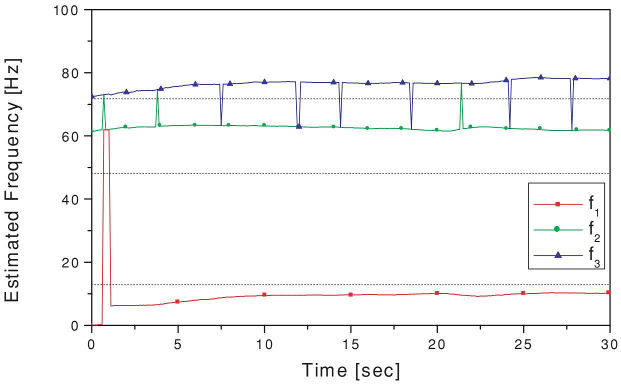


Figure 14. Frequency estimation using the RML method in the Fx case.

the DSP board. By adding masses, we changed the natural frequencies of the specimens. Experimental tests have been devised to confirm the feasibility of real-time computations, and to impose the severe conditions of drastically different amplitudes and of considerable changes of natural frequencies for three modes. The frequency estimation performance of the AR method is better than that of the other methods in spite of the frequency changes during the estimation of frequencies.

From these results, we recommend using the CNF methods when the computational efficiency is critical and the estimation condition is simple, whereas the AR method can be used with the Bairstow method if the test environments are not noisy and there are more than two modes with different amplitudes.

We leave the comparison with other methods for a future study. We expect that the frequency estimation methods will be applied to the health monitoring of structures and adaptive vibration controls in diverse engineering problems.

Acknowledgments

This study has been supported by the Ministry of Science and Technology through National Research Laboratory Programme. The authors gratefully acknowledge the support from the National Research Laboratory Programme.

Appendix A. RLS method

The procedure of the RLS method is summarized as follows.
Step 1: Let us define the measured vector as $\phi(k)$ and the unknown vector as $\theta(k)$.

$$\phi(k) \triangleq [-y(k-1) \quad -y(k-2) \quad \cdots \quad -y(k-2p)]^{\mathrm{T}}$$

$$\phi(k) \triangleq [-y(k-1) \quad -y(k-2) \quad \cdots \quad -y(k-n), \tag{A.1a}$$

$$u(k-1) \quad u(k-2) \quad \cdots \quad (k-2p)]^{\mathrm{T}} \tag{A.1b}$$

$$\theta(k) \triangleq [a_1(k) \quad a_2(k) \quad \cdots \quad a_{2p}(k)]^{\mathrm{T}} \tag{A.2a}$$

$$\theta(k) \triangleq [a_1(k) \quad a_2(k) \quad \cdots \quad a_{2p}(k), b_1(k) \quad b_2(k) \quad \cdots \quad b_{2p}(k)]^{\mathrm{T}} \tag{A.2b}$$

where k is an index denoting the current state, and the equations numbered (a) and (b) denote the AR and ARMA methods, respectively.

Step 2: Then, the measured signal can be described as follows:

$$y(k) = -\sum_{j=1}^{2p} a_j(k-j)y(k-j) + \varepsilon(k) = \theta(k)^{\mathrm{T}}\phi(k) + \varepsilon(k) \tag{A.3a}$$

$$y(k) = -\sum_{j=1}^{2p} a_j(k-j)y(k-j) + \sum_{j=1}^{2p} b_j(k-j)u(k-j) + \varepsilon(k) = \theta(k)^{\mathrm{T}}\phi(k) + \varepsilon(k) \tag{A.3b}$$

where ε is an error between the measured and estimated signals. The estimation of the characteristic equation means that the square of error in equation (3) is to be minimized with respect to the objective function using the RLS method

$$\mathbf{P}(0) = \delta^{-1}\mathbf{I} \qquad \theta(0) = [\text{arbitary vector}] \tag{A.4}$$

where $\mathbf{P}(k)$ is referred to as a covariance matrix of the input vector. The initial estimation of $\mathbf{P}(k)$, denoted by $\mathbf{P}(0)$, is a large diagonal matrix, $10^8\mathbf{I}$, in this study. Similarly, the initial value of $\theta(k)$ is set to be $[1, \ldots, 1]$, which is an arbitrary initial guess.
Step 3: We repeat the following calculation to estimate the parameter vector, $\theta(k)$

$$\theta(k) = \theta(k-1) + \frac{\mathbf{P}(k-2)\phi(k-1)\{y(k) - \phi(k-1)^{\mathrm{T}}\theta(k-1)\}}{\lambda(t)\{\lambda(t) + \phi(k-1)^{\mathrm{T}}\mathbf{P}(k-2)\phi(k-1)\}} \tag{A.5}$$

$$\mathbf{P}(k-1) = \mathbf{P}(k-2) + \frac{\mathbf{P}(k-2)\phi(k-1)\phi(k-1)^{\mathrm{T}}\mathbf{P}(k-2)}{\lambda(k)\{\lambda(k) + \phi(k-1)^{\mathrm{T}}\mathbf{P}(k-2)\phi(k-1)\}}. \tag{A.6}$$

Appendix B. Bairstow method

The procedure of the Bairstow method is summarized as follows.
Step 1: We calculate $\bar{b}_0, \bar{b}_1, \ldots, \bar{b}_k$ using the coefficients of equation (A.2) and estimate the initial values of p_i and q_i as follows:

$$\begin{gathered} \bar{b}_0 = a_0 \qquad \bar{b}_1 = a_1 - p_i\bar{b}_0 \\ \bar{b}_k = a_k - p_i\bar{b}_{k-1} - q_i\bar{b}_{k-2} \quad k = 2, 3, 4, \ldots, n. \end{gathered} \tag{B.1}$$

Step 2: We calculate $c_0, c_1, \ldots, c_k$ using obtained $\bar{b}_k$, p_i and q_i as follows:

$$\begin{gathered} c_0 = \bar{b}_0 \qquad c_1 = \bar{b}_1 - p_i c_0 \\ c_k = \bar{b}_k - p_i c_{k-1} - q_i c_{k-2} \quad k = 2, 3, 4, \ldots, n-1. \end{gathered} \tag{B.2}$$

Step 3: We construct Δp and Δq using b_n, $b_{n-1}, c_{n-1}, c_{n-2}$ and c_{n-3} obtained from steps 1 and 2

$$\begin{aligned} \Delta p &= \begin{vmatrix} b_{n-1} & c_{n-3} \\ b_n & c_{n-2} \end{vmatrix} \Bigg/ \begin{vmatrix} c_{n-2} & c_{n-3} \\ c_{n-1} - b_{n-1} & c_{n-2} \end{vmatrix} \\ \Delta q &= \begin{vmatrix} c_{n-2} & b_{n-1} \\ c_{n-1} - b_{n-1} & b_n \end{vmatrix} \Bigg/ \begin{vmatrix} c_{n-2} & c_{n-3} \\ c_{n-1} - b_{n-1} & c_{n-2} \end{vmatrix}. \end{aligned} \tag{B.3}$$

Step 4: p_i and q_i are updated as follows:

$$p_{i+1} = p_i + \Delta p \qquad q_{i+1} = q_i + \Delta q. \tag{B.4}$$

We check the convergence of p_{i+1} and q_{i+1} using the following criterion

$$\frac{|p_{i+1} - p_i|}{|p_i|} < \varepsilon \quad \text{and} \quad \frac{|q_{i+1} - q_i|}{|q_i|} < \varepsilon. \tag{B.5}$$

We repeat processes 1–4 until p_{i+1} and q_{i+1} are converged.

Step 5: When p_{i+1} and q_{i+1} are converged, equation (3) can be divided by a quadratic factor, $x^2 + p_{i+1}x + q_{i+1}$. The remaining product is a $2(p-1)$th-order polynomial, of which the coefficients are $\bar{b}_k$. We iterate the above procedure for further factorization after substitution as follows

$$a_k \to b_k \qquad n \to n-2 \quad K = 0, 1, 2, \ldots, n-2$$

$$\hat{f}_i = \frac{1}{2\pi T_s} \text{ angle } \left(Z_i | A(z^{-1}) = 0\right) \tag{B.6}$$

$$= \frac{1}{2\pi T_s} \left[\cos^{-1}(-p_i/2\sqrt{q_i})\right] \quad i = 1, 2, \ldots, p \tag{B.7}$$

where T_s is a sampling time. The updated $\hat{f}_i$ is discarded if the value of $(-p_i/2\sqrt{q_i})$ is out of the range $[-1, 1]$ during the real-time process. The obtained natural frequencies are sorted by magnitudes using a simple straight insertion algorithm [14].

Appendix C. RML method

The RML algorithm is summarized as follows.
Step 1: We initialize the algorithm by setting

$$\mathbf{P}(0) = \delta^{-1}\mathbf{I} \qquad \hat{\boldsymbol{\theta}} = [a_1 \quad \cdots \quad a_p]^{\mathrm{T}} \qquad \hat{\boldsymbol{\theta}}(0) = 0. \tag{C.1}$$

Step 2: We formulate the error, the covariance matrix and the parameter vector as follows

$$\varepsilon(t) = y(t) + y(t-2) - \rho^{2p}\varepsilon(t-2p) - \varphi^{\mathrm{T}}(t)\hat{\theta}(t-1) \tag{C.2}$$

$$\mathbf{P}(t) = \mathbf{P}(t-1) - \frac{\mathbf{P}(t-1)\psi(t)\psi(t-1)^{\mathrm{T}}\mathbf{P}(t-1)}{\lambda(t)\{\lambda(t) + \varphi(t)^{\mathrm{T}}\mathbf{P}(t-1)\varphi(t)\}} \tag{C.3}$$

$$\hat{\theta}(t) = \hat{\theta}(t-1) + \mathbf{P}(t)\psi(t)\varepsilon(t). \tag{C.4}$$

Step 3: We calculate the error, the filtered error, the filtered signal, etc as follows

$$\bar{\varepsilon}(t) = y(t) + y(t-2) - \rho^{2p}\bar{\varepsilon}(t-2p) - \phi^{\mathrm{T}}(t)\hat{\theta}(t) \tag{C.5}$$

$$\begin{aligned}\bar{\varepsilon}_F(t) = {} & \bar{\varepsilon}(t) - \rho^{2n}\bar{\varepsilon}_F(t-2p) - \rho^p(t)\bar{\varepsilon}_F(t-p)\hat{a}_p(t) \\ & - \sum_{i=1}^{n-1}\left[\rho^i(t)\bar{\varepsilon}_F(t-i) + \rho^{2p-i}(t)\bar{\varepsilon}_F(t-2p+i)\right]\hat{a}_i(t)\end{aligned} \tag{C.6}$$

$$\begin{aligned}y_F(t) = {} & y(t) - \rho^{2p}y_F(t-2p) - \rho^p(t)\bar{\varepsilon}_F(t-p)\hat{a}_p(t) \\ & - \sum_{i=1}^{p-1}\left[\rho^i(t)y_F(t-i) + \rho^{2p-i}(t)\bar{\varepsilon}_F(t-2p+i)\right]\hat{a}_i(t)\end{aligned} \tag{C.7}$$

$$\varphi_i(t) = \begin{cases} -y(t-i) - y(t-2p+i) + \rho^i(t)\bar{\varepsilon}(t-i) \\ \quad + \rho^{2p-i}(t)\bar{\varepsilon}(t-2p+i) \qquad 1 \leqslant i \leqslant p-1 \\ -y(t-p) - \rho^p(t)\bar{\varepsilon}(t-p) \qquad i = p \end{cases} \tag{C.8}$$

$$\psi_i(t) = \begin{cases} -y_F(t-i) - y_F(t-2p+i) + \rho^i(t)\bar{\varepsilon}_F(t-i) \\ \quad + \rho^{2p-i}(t)\bar{\varepsilon}_F(t-2p+i) \qquad 1 \leqslant i \leqslant p-1 \\ -y_F(t-p) - \rho^p(t)\bar{\varepsilon}_F(t-p) \qquad i = p \end{cases} \tag{C.9}$$

Step 4: We update the parameter vector and the direction vector as follows

$$\varphi(t+1) = [\varphi_1(t+1) \quad \ldots \quad \varphi_p(t+1)]^{\mathrm{T}} \tag{C.10}$$

$$\psi(t+1) = [\psi_1(t+1) \quad \cdots \quad \psi_p(t+1)]^{\mathrm{T}} \tag{C.11}$$

$$\hat{f}_i = \frac{1}{2\pi T_s} \text{ angle } \left(Z_i | A(z^{-1}) = 0\right) \quad i = 1, 2, \ldots, p. \tag{C.12}$$

Step 5: We repeat steps 2–4 at every time step.

Appendix D. LinANF method

The LinANF method is summarized as follows.
Step 1: We initialize the algorithm by setting

$$\mathbf{P}(0) = \delta^{-1}\mathbf{I} \qquad \boldsymbol{\theta} = [a_1 \quad \cdots \quad a_p]^{\mathrm{T}} \qquad \boldsymbol{\theta}(\mathbf{0}) = 0 \tag{D.1}$$

where δ is a small positive constant.
Step 2: We formulate the error, the covariance matrix and the parameter vector as follows

$$y(n) = \frac{1}{A(\rho q^{-1})}x(n). \tag{D.2}$$

Step 3: We compute for each instant of time

$$\psi(n-1) = [\psi_1(n-1) \quad \cdots \quad \psi_p(n-1)]^{\mathrm{T}} \tag{D.3}$$

$$\psi_i(n-1) = \begin{cases} y(n-i) + y(n-2p+i) \quad 1 \leqslant i \leqslant p-1 \\ y(n-p) \quad i = p \end{cases} \tag{D.4}$$

$$\varepsilon(n) = y(n) + y(n-2p) + \boldsymbol{\psi}^{\mathrm{T}}(n-1)\boldsymbol{\theta}(n-1). \tag{D.5}$$

Step 4: We update the parameter vector and the direction vector as follows

$$g(n) = \frac{\mathbf{P}(n-1)\boldsymbol{\psi}(n-1)}{\lambda + \boldsymbol{\psi}(n-1)^{\mathrm{T}}\mathbf{P}(n-1)\boldsymbol{\psi}(n-1)} \tag{D.6}$$

$$\mathbf{P}(n) = \mathbf{P}(n-1)/\lambda - g(n)\boldsymbol{\psi}(n-1)^{\mathrm{T}}\mathbf{P}(n-1)/\lambda \tag{D.7}$$

$$\boldsymbol{\theta}(n) = \boldsymbol{\theta}(n-1) - g(n)\varepsilon(n) \tag{D.8}$$

$$\hat{f}_i = \frac{1}{2\pi T_s} \text{ angle } \left(Z_i | A(z^{-1}) = 0\right) \qquad i = 1, 2, \ldots, p. \tag{D.9}$$

Step 5: We iterate steps 3 and 4 at every time step.

Appendix E. CNF methods

The EachCNF and LastCNF algorithms are summarized as follows.
Step 1: We initialize the algorithm by setting

$$\boldsymbol{\theta} = [a_1 \quad \cdots \quad a_p]^{\mathrm{T}} \qquad \boldsymbol{\theta}(0) = \mathbf{0} \qquad k = 1, 2, \ldots, p. \tag{E.1}$$

Step 2: For each instant of time we compute

$$\begin{aligned}\varepsilon_k(n) &= N_k(q^{-1})\varepsilon_{k-1}(n) = \frac{A_k(q^{-1})}{A_k(\rho q^{-1})}\varepsilon_{k-1}(n) \\ &= \frac{1 + a_k(n)q^{-1} + q^{-2}}{1 + \rho a_k(n)q^{-1} + \rho^2 q^{-2}}\varepsilon_{k-1}(n)\end{aligned} \tag{E.2}$$

$$\varepsilon_0(n) = \hat{d}(n) \qquad k = 1, \ldots, p$$

$$\begin{aligned}\tilde{\varepsilon}_k(n) &= \frac{1}{A_k(\rho q^{-1})}\varepsilon_{k-1}(n) = \varepsilon_{k-1}(n) - \rho a_k(n-1)\tilde{\varepsilon}_k(n-1) \\ &\quad -\rho^2\tilde{\varepsilon}_k(n-2)\end{aligned} \tag{E.3}$$

$$\Phi_k(n) = \lambda\Phi_k(n-1) + \tilde{\varepsilon}_k(n-1)^2. \tag{E.4}$$

Step 3: For each time step we compute

$$E_k(n) = \tilde{\varepsilon}_k(n) + \tilde{\varepsilon}_k(n-2) \qquad \text{(E.5a)}$$

$$\begin{aligned} E_k(n) = \sum_{1=k}^{p-1} \big\{ & a_{l+1}(n-1)\varepsilon_l(n-1) + \varepsilon_l(n-2) \\ & - \rho a_{l+1}(n-1)\varepsilon_{l+1}(n-1) - \rho^2 \varepsilon_{l+1}(n-2) \big\} \\ & + \tilde{\varepsilon}_k(n) + \tilde{\varepsilon}_k(n-2) \qquad \text{(E.5b)} \end{aligned}$$

where equations numbered (a) and (b) are applied to the EachCNF and LastCNF methods, respectively.

Step 4: For each step we compute

$$z_k(n) = \lambda z_k(n-1) + \tilde{\varepsilon}_k(n-1) E_k(n) \qquad \text{(E.6)}$$

$$a_k(n) = -\Phi_k(n)^1 z_k(n) \qquad \text{(E.7)}$$

$$\hat{f}_i = \frac{1}{2\pi T_s} \cos^{-1}\left(-\frac{a_k(n)}{2}\right) \qquad i = 1, 2, \ldots, p. \qquad \text{(E.8)}$$

Step 5 : We repeat steps 2– 4 at every time step.

References

[1] Goodwin G C and Sin K S 1984 *Adaptive Filtering Prediction and Control* 2nd edn (Englewood Cliffs, NJ: Prentice-Hall)

[2] Proakis J G and Manolakis D G 1996 *Digital Signal Processing* 3rd edn (Englewood Cliffs, NJ: Prentice-Hall)

[3] Regalia P A 1995 *Adaptive IIR Filtering in Signal Processing and Control* (New York: Dekker) chapter 10

[4] Hansen C H and Snyder S D 1997 *Active Control of Noise and Vibration* (E & FN SPON) pp 374–543

[5] Kim S and Park Y 1999 Active control of multi-tonal noise with reference generator based on on-line frequency estimation *J. Sound Vib.* **227** 647

[6] Kuo S M and Vijayan D 1994 Adaptive algorithms and experimental verification of feedback active noise control systems *Noise Control Eng.* **42** 37

[7] Rew K H, Han J H and Lee I 2000 Adaptive multi-modal vibration control of wing-like composite structure using adaptive positive position feedback AIAA 2000-1422: *Proc. 41st Conf. on AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Material* (Atlanta, 2000)

[8] Chaplin G R B and Smith R A 1986 Method of and apparatus for canceling vibrations from a source of repetitive vibrations *US Patent 4566118*

[9] Nehorai A 1985 A minimal parameter adaptive notch filter with constrained poles and zeros *IEEE Trans. Acoust. Speech Signal Process.* **33** 983

[10] Tranvassor-Romano J M and Bellanger M 1988 Fast least squares adaptive notch filtering *IEEE Trans. Acoust. Speech Signal Process.* **36** 1536

[11] Kim S 1998 Active noise control algorithm based on noise frequency estimation *MS Thesis* Korea Advanced Institute of Science and Technology

[12] Rhim S and Book W J 1999 Adaptive command shaping using adaptive filter approach in time domain *Proc. American Control Conf.* 81

[13] Varkonyi-Koczy A R *et al* 1998 A fast filter-bank for adaptive fourier analysis *Proc. conf. on IEEE Instrumentation and Measurement Technology* vol 2 pp 915–8

[14] Press W H *et al* 1988 *Numerical Recipes in C* 2nd edn (Cambridge: Cambridge University Press) chapter 9

[15] Chapra S C and Canale R P 1988 *Numerical Methods for Engineers* 2nd edn (New York: McGraw-Hill) chapter 21

[16] Kay S M 1988 *Modern Spectral Estimation: Theory and Application* (Englewood Cliffs, NJ: Prentice-Hall)

[17] Transvassos-Romano J M and Bellinger M 1988 Fast adaptive notch filter in cascade form *EUSIPCO-88: Signal Processing-IV (Grenoble, France,* Sep. 1988*)* pp 567–70