IEMS Vol. 1, No. 1, pp. 79-86, December 2002.

Robust System Identification Algorithm Using Cross Correlation Function

Kazuhiro Takeyasu†
Research and Consulting Division
The Japan Research Institute, Ltd., 16 Ichiban-cho, Chiyoda-ku, Tokyo, 102-0082, JAPAN
Tel: +81-3-3288-4171, Fax: +81-3-3288-4339, E-mail: takeyasu.kazuhiro@jri.co.jp

Takashi Amemiya
Tokyo Metropolitan Institute of Technology, 6-6 Asahigaoka, Hino, Tokyo, 191-0065, JAPAN
Tel & Fax: +81-42-585-8666, E-mail: tamemiya@tmit.ac.jp

Hiroyuki Goto
Research and Consulting Division
The Japan Research Institute, Ltd., 16 Ichiban-cho, Chiyoda-ku, Tokyo, 102-0082, JAPAN
Tel: +81-3-3288-4173, Fax: +81-3-3288-4339, E-mail: goto.hiroyuki@jri.co.jp

Shiro Masuda
Tokyo Metropolitan Institute of Technology, 6-6 Asahigaoka, Hino, Tokyo, 191-0065, JAPAN
Tel & Fax: +81-42-585-8631, E-mail: smasuda@cc.tmit.ac.jp

Abstract. This paper proposes a new algorithm for estimating ARMA model parameters. In estimating ARMA model parameters, several methods such as generalized least square method, instrumental variable method have been developed. Among these methods, the utilization of a bootstrap type algorithm is known as one of the effective approach for the estimation, but there are cases that it does not converge. Hence, in this paper, making use of a cross correlation function and utilizing the relation of structural a priori knowledge, a new bootstrap algorithm is developed. By introducing theoretical relations, it became possible to remove terms, which is liable to include much noise. Therefore, this leads to robust parameter estimation. It is shown by numerical examples that using this algorithm, all simulation cases converge while only half cases succeeded with the previous one. As for the calculation time, judging from the fact that we got converged solutions, our proposed method is said to be superior as a whole.

Keywords: time series analysis, ARMA model, autocorrelation function, cross correlation function, estimation algorithm

1. INTRODUCTION

In the system identification method, using Auto-regressive Moving Average (ARMA) models or Auto-regressive Moving Average with exogenous input (ARMAX) models, system parameters are estimated on the basis of the past time series data. Several algorithms for obtaining unbiased estimate have been developed.

In this paper, we first present the mathematical model, make clear the discussing points and then propose our improvements.

A (p, q) order ARMA model is stated as

$$x_n + \sum_{i=1}^{p} a_i x_{n-i} = e_n + \sum_{j=1}^{q} \bar{b}_j e_{n-j} \tag{1}$$

where

$\{x_n\}$: Sample process of stationary ergodic Gaussian process $x(t)$ $(n = 1, 2, \cdots, N, \cdots)$

$\{e_n\}$: Gaussian white noises with mean 0, variance σ_e^2

†: Corresponding Author

$$A(z) = 1 + a_1 z^{-1} + \cdots + a_p z^{-P}$$

$$B(z) = 1 + b_1 z^{-1} + \cdots + b_q z^{-q}$$

Assume that $A(z)$, $B(z)$ are irreducible and satisfy stationary, invertibility, strictly positive realness conditions.

Eq. (1) may be said a model which does not have a control input in the general ARMAX model such as:

$$\begin{aligned} x_n + a_1 x_{n-1} + \cdots + a_p x_{n-p} \\ = c_1 u_{n-1} + \cdots + c_m u_{n-m} \\ + e_n + b_1 e_{n-1} + \cdots + b_q e_{n-q} \end{aligned} \quad (2)$$

In estimating parameters $\{a_i\}$, $\{b_i\}$ of Eq. (1), the right hand side of the equation is itself a colored noise. Therefore, it should be noted that a biased estimation is obtained by using the least square method. In order to obtain an unbiased estimate, extended least square method, generalized least square method, sequential maximum likelihood method, instrumental variable method, and pseudo linear regression method have been developed (Tokumaru *et al.*, 1982; Katayama, 1994).

In pseudo linear regression method, it utilizes bootstrap method and make iterative calculation. However, it has a difficulty that there are many cases that these iteration do not converge (Sagara *et al.*, 1994; Katayama, 1994).

In this paper, we propose an improving method for that problem. By utilizing this method, the total calculation time can be reduced. Introducing the cross correlation function of output and noise, and also using the theoretical relation, the above improvement is achieved.

As for the shortening the calculation time, Nakamura and Oishi (1984) proposed a method to use the generalized least square method. They considered an input-output system with noise. Generally, we often have time series data of output and do not have control input. Therefore, in this paper, we consider the system with the following viewpoints. Using cross correlation function and utilizing a priori knowledge of the structural character of the model, we can get robust parameter estimation. Convergence probability is higher than that of the previous method. Therefore, we can achieve a reduction of total time to get the solution that converges.

Thus, there have been papers corresponding to this theme but there are no papers as we propose.

In section 2, we propose a robust system identification algorithm of bootstrap type utilizing cross correlation functions. In section 3, we state the relation between our method and the maximum likelihood method. In section 4, we show a pseudo linear regression method for the comparison with our new method. Numerical examples are given in section 5.

2. PARAMETER ESTIMATION OF ARMA MODEL USING CROSS CORRELATION FUNCTION

By definition, the auto correlation function of $\{x_i\}$ is stated as

$$R_k = E[x_n x_{n+k}] \quad (3)$$

$$R_{-k} = R_k \quad (4)$$

Because $\{e_j\}$ is Gaussian white noise, the cross correlation function of $\{x_i\}$ and $\{e_j\}$ are

$$\left.\begin{aligned} T_{ex}(l) &= E[e_n x_{n+l}] &= T_l \\ T_{xe}(l) &= E[x_n e_{n+l}] &= T_{-l} \\ T_{ex}(-l) &= E[e_n x_{n-l}] &= T_{-l} \\ T_{xe}(l) &= E[x_n e_{n-l}] &= T_l \end{aligned}\right\} \quad (5)$$

As for $\{e_k\}$, $\{e_l\}$,

$$E[e_k e_{k+l}] = E[e_{k+l} e_k] = S_l \quad (6)$$

$$E[e_k^2] = \sigma_e^2 = S_0 \quad (7)$$

T_{-l}, S_l $(l > 0)$ ought to be 0 theoretically. However, if they are estimated under finite number of data, they usually have non-zero value.

It is well known that p and q of Eq. (1) satisfy $p \leq q$ when ARMA process is expressed in state space expression under modern control theory (Tokumaru and Takeyasu, 1977). For simplicity, let $p = q$ hereafter. Then Eq. (1) is expressed as

$$x_n = -\sum_{i=1}^{p} a_i x_{n-i} + \sum_{j=1}^{p} b_j e_{n-j} + e_n \quad (8)$$

Define Z_n, θ_n as

$$\begin{aligned} \boldsymbol{Z}_n &= [-x_{n-1}, \ldots, -x_{n-p}, e_{n-1}, \ldots, e_{n-p}]^T \\ &= [-\boldsymbol{x}_n{}^T, \boldsymbol{e}_n{}^T]^T \end{aligned} \quad (9)$$

$$\begin{aligned} \boldsymbol{\theta}_n &= [a_1, \ldots, a_p, b_1, \ldots, b_p]^T \\ &= [\boldsymbol{a}^T, \boldsymbol{b}^T]^T \end{aligned} \quad (10)$$

Then Eq. (8) is expressed as

$$x_n = \boldsymbol{\theta}_n{}^T \boldsymbol{Z}_n + e_n \quad (11)$$

Let $\hat{\theta}_N$ be the θ which minimizes

$$I_N = \sum_{n=1}^{N} \left[x_n - \theta^T Z_n\right]^2 \tag{12}$$

then $\hat{\theta}_N$ is given by

$$\hat{\theta} = \left[\sum_{n=1}^{N} Z_n Z_n^T\right]^{-1} \sum_{n=1}^{N} Z_n x_n \tag{13}$$

Let $N \to \infty$, then the right hand side of Eq. (13) is expressed, with the help of the autocorrelation functions of $\{x_i\}$ and cross correlation functions $\{x_i\}$ and $\{e_j\}$ as follows

$$\begin{bmatrix} a \\ b \end{bmatrix} = \begin{bmatrix} R & -T \\ -T^T & \sigma_e^2 I \end{bmatrix}^{-1} \begin{bmatrix} -r \\ t \end{bmatrix} \tag{14}$$

where

$$R = \begin{bmatrix} R_0 & R_1 & \cdots & R_{p-1} \\ R_1 & R_0 & \cdots & R_{p-2} \\ \vdots & \vdots & \ddots & \vdots \\ R_{p-1} & R_{p-2} & \cdots & R_0 \end{bmatrix}$$

$$T = \begin{bmatrix} T_0 & T_1 & \cdots & T_{p-1} \\ 0 & T_0 & \cdots & T_{p-2} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & T_0 \end{bmatrix}$$

$$r = [R_1, R_2, \cdots, R_p]^T, \quad t = [T_1, T_2, \cdots, T_p]^T$$

From Eq. (14), the following equations are obtained.

$$Ra - Tb = -r \tag{15}$$

$$-T^T a + \sigma_e^2 b = t \tag{16}$$

From Eq. (16), we get

$$b = \frac{1}{\sigma_e^2}\left(t + T^T a\right) \tag{17}$$

Putting this into Eq. (15), we get

$$a = \left(R - \frac{1}{\sigma_e^2} T T^T\right)^{-1} \left(\frac{1}{\sigma_e^2} T t - r\right) \tag{18}$$

We can get b by substituting this to Eq. (17).

Entire parameter estimation algorithm can be expressed as shown in Table 1.

As T is an upper triangular matrix, the calculation time can be shortened by using its characteristics. In the previous estimation method, parameter estimation is done by using the full matrix of T. Here we utilize a priori knowledge of the structural character of the model, then

- Robust solution
- Shortening calculation time

can be expected.

Table 1. Parameter estimation algorithm

Step1 :	Calculate auto correlation function $\{\hat{R}_k\}$ by observed data $\{x_n\}$
Step2 :	Set initial values for $\{e_n\}$
Step3 :	Calculate cross correlation function $\{\hat{T}_i\}$
Step4 :	Estimate $\hat{a}^{(l)}$, $\hat{b}^{(l)}$ from Eq. (17), (18)
Step5 :	Estimate $\{e_n\}$ from Eq. (8)
Step6 :	Iterate step 3, 4, 5 until $\hat{a}^{(l)}$ and $\hat{b}^{(l)}$ converge

When we use Eqs. (5)-(7), the corresponding equation to Eq. (14) is

$$\begin{bmatrix} a \\ b \end{bmatrix} = \begin{bmatrix} R & -\tilde{T} \\ -\tilde{T}^T & S \end{bmatrix}^{-1} \begin{bmatrix} -r \\ t \end{bmatrix} \tag{19}$$

where

$$R = \begin{bmatrix} R_0 & R_1 & \cdots & R_{p-1} \\ R_1 & R_0 & \cdots & R_{p-2} \\ \vdots & \vdots & \ddots & \vdots \\ R_{p-1} & R_{p-2} & \cdots & R_0 \end{bmatrix}$$

$$\tilde{T} = \begin{bmatrix} T_0 & T_1 & \cdots & T_{p-1} \\ T_{-1} & T_0 & \cdots & T_{p-2} \\ \vdots & \vdots & \ddots & \vdots \\ T_{-(p-1)} & T_{-(p-2)} & \cdots & T_0 \end{bmatrix}$$

$$S = \begin{bmatrix} S_0 & S_1 & \cdots & S_{p-1} \\ S_{-1} & S_0 & \cdots & S_{p-2} \\ \vdots & \vdots & \ddots & \vdots \\ S_{(p-1)} & S_{(p-2)} & \cdots & S_0 \end{bmatrix}$$

$$r = [R_1, R_2, \ldots, R_p]^T, \quad t = [T_1, T_2, \cdots, T_p]^T$$

Re-writing them, the following equations are derived.

$$Ra - \tilde{T}b = -r \tag{20}$$

$$-\tilde{T}^T a + Sb = t \tag{21}$$

By Eq. (21), it follows

$$b = S^{-1}\left(t + \tilde{T}^T a\right) \tag{22}$$

Substituting this to Eq. (20), it follows

$$a = \left(R - \tilde{T}S^{-1}\tilde{T}^T\right)^{-1}\left(\tilde{T}S^{-1}t - r\right) \tag{23}$$

Substituting this to Eq. (22), we can get $\boldsymbol{b}$.

3. RELATIONSHIP WITH MAXIMUM LIKELIHOOD ESTIMATION

Maximum Likelihood Estimation Algorithm using Gauss-Newton method is as follows.

$$S(\theta) = \sum_{n=1}^{N} e_n^2 = \sum_{n=1}^{N}\left(x_n - \theta^T Z_n\right)^2 \tag{24}$$

Make iteration by

$$\begin{aligned}\theta(i+1) &= \theta(i) - \left[\frac{\partial^2 S}{\partial\theta\partial\theta^T}\right]^{-1}_{\theta(i)} \left.\frac{\partial S}{\partial\theta}\right|_{\theta(i)} \\ &= \theta(i) - \left[\sum_{n=1}^{N}\frac{\partial e_n}{\partial\theta}\frac{\partial e_n}{\partial\theta^T}\right]^{-1}_{\theta(i)} \sum_{n=1}^{N} e_n \left.\frac{\partial e_n}{\partial\theta}\right|_{\theta(i)}\end{aligned} \tag{25}$$

Minimum least square estimate of Eq. (24) is given by

$$\hat{\theta}_N = \left[\sum_{n=1}^{N} Z_n Z_n^T\right]^{-1} \sum_{n=1}^{N} Z_n x_n \tag{26}$$

The following recursive estimation algorithm using the above relation gives an approximate maximum likelihood estimation (Tokumaru *et al.*, 1982).

$$\begin{cases}\hat{\theta}_N = \hat{\theta}_{N-1} + \hat{k}_N\left[x_n - \hat{z}_N^T\hat{\theta}_{N-1}\right] \\ \hat{k}_N = p_{N-1}\hat{z}_N\left[1 + \hat{z}_N^T p_{N-1}\hat{z}_N\right]^{-1} \\ p_N = \left[I - \hat{k}_N Z_N^T\right]p_{N-1}\end{cases}$$

When $N \to \infty$, Eq. (24) is expressed as

$$\lim_{N\to\infty} S(\theta) = R_0 - 2\theta^T\left[\frac{-r}{t}\right] + \theta^T\begin{bmatrix} R & -T \\ -T^T & \sigma_e^2 I\end{bmatrix}\theta \tag{27}$$

Eq. (26) is expressed as

$$\lim_{N\to\infty}\hat{\theta}^N = \begin{bmatrix} R & -T \\ -T^T & \sigma_e^2 I\end{bmatrix}\left[\frac{-r}{t}\right] \tag{28}$$

while

$$\frac{\partial S}{\partial\theta} = -2\left[\frac{-r}{t}\right] + 2\begin{bmatrix} R & -T \\ -T^T & \sigma_e^2 I\end{bmatrix}\theta \tag{29}$$

$$\frac{\partial^2 S}{\partial\theta\partial\theta^T} = 2\begin{bmatrix} R & -T \\ -T^T & \sigma_e^2 I\end{bmatrix} \tag{30}$$

Eq. (25) becomes

$$\theta(i+1) = \theta(i) - \begin{bmatrix} R & -T \\ -T^T & \sigma_e^2 I\end{bmatrix}^{-1}\left\{-\left[\frac{-r}{t}\right] + \begin{bmatrix} R & -T \\ -T^T & \sigma_e^2 I\end{bmatrix}\theta\right\} \tag{31}$$

Therefore, utilizing the relation of Eq. (28), Eq. (31) becomes

$$\theta(i+1) \to \theta(i) \tag{32}$$

This means that it is because Eq. (32) has reached the equilibrium point in $N \to \infty$.

The new method stated in the previous section is said that it has utilized the relation of the equilibrium point in $N \to \infty$. That is to say that the new method uses the following relation.

T is a triangular matrix ($T_{-l} = 0$) in Eq. (30), and $\boldsymbol{S}$ (defined below Eq. (19)) is a diagonal matrix ($S_0 = \sigma_e^2$, $S_l = 0$ ($l > 0$)) and that leads to Eq. (14) which is the same form as Eq. (28).

Using finite data, we calculate $\{\hat{R}_i\}$, $\{\hat{T}_i\}$ which is not real R_i, T_j. So the iterative calculation becomes meaningful.

4. PSEUDO LINEAR REGRESSION METHOD

Here, we show the pseudo linear regression method for a comparison sake. Set,

$$\varphi(n,\theta)^T = \left(x_{n-1}, \cdots, x_{n-p}, -e_{n-1}, \cdots, -e_{n-p}\right) \tag{33}$$

$$\theta(n,\theta)^T = \left(a_1, \cdots, a_p, b_1, \cdots, b_p\right) \tag{34}$$

We get

$$\frac{1}{N}\sum_{n=1}^{N}\varphi(n,\theta)\left[x_n - \varphi^T(n,\theta)\theta\right] = 0 \tag{35}$$

$$\hat{\theta}^{(i+1)} = \left(\frac{1}{N}\sum_{n=1}^{N}\varphi(n,\hat{\theta}^{(i)})\varphi^T(n,\hat{\theta}^{(i)})\right)^{-1}\frac{1}{N}\sum_{n=1}^{N}\varphi(t,\hat{\theta}^{(i)})x_n \tag{36}$$

Using Eq. (36), we calculate the pseudo linear regression vector. We repeat this which is known as a bootstrap method (Katayama, 1994). Eq. (36) corresponds to Eq. (19).

5. NUMERICAL EXAMPLES

5.1 Numerical Calculation

We use a (2, 2) order ARMA model in Eq. (1) and examine in three types (Table 2).

Conditions to be satisfied are as follows, which are stated in section 1.

Table 2. Numerical models

	Parameters			
Model	a_1	a_2	b_1	b_2
Model 1	− 1.5	0.7	− 0.6	0.3
Model 2	− 1.4	0.6	− 0.5	0.2
Model 3	− 1.2	0.4	− 0.4	0.2

- Stationarity condition; root of $A(z)$ exist inside the unit circle
- Invertibility condition; root of $B(z)$ exist inside the unit circle
- Strictly Positive Realness condition; $|b_1| + |b_2| < 1$

Model 1 has a root close to the boundary, which makes the system identification rather hard. We suppose that model 2, 3 are much easier in system identification.

Table 3 shows the result of 100 times simulations with different initial state of $\{e_n\}$ on the data N=1000, 3000, 5000, 10000 for each.

In generating time series data, the first 100 data are ignored and we use MATLAB for calculation. The proposed method is the one which uses Eqs. (14)-(18) and the previous method is the one stated in section 4.

Hereafter, we examine;

1. Convergence ratio
2. Parameter estimation accuracy
3. Calculation time

We first compare the convergence ratio (Table 3). Each case contains 100 times simulations, so the number in each column of the Table 3 is given in percent.

In the previous method, all cases diverge on model 1 which is hard to identify. Even in model 2, it converges in the case N = 1000. But when the number of data increase, only 5% for N = 3000 and 0% for the case more than N = 5000 converge.

In model 3, the convergence ratio is nearly 100%. Thus, the convergence ratios by the previous method differ remarkably by the model or the number of data.

On the other hand, the convergence ratio of the proposed method is 27.25%, 26.25%, 36.75% for each model 1, 2, 3, and they are all near around 30%.

In all cases, the proposed method converges within

Table 3. Comparison of convergence ratio

Model	N				Av.
	1000	3000	5000	10000	
Model 1					
Proposed	33	19	27	30	27.25
Previous	0	0	0	0	0
Model 2					
Proposed	31	23	26	25	26.25
Previous	100	5	0	0	26.25
Model 3					
Proposed	47	35	30	35	36.75
Previous	100	100	100	99	99.75

the range of nearly 20% to 50%.

It suggests that even if there is a case which does not converge, varying initial data, the converged solution can be obtained within 3 or 4 trials. Therefore, it can be said that the proposed method is a quite robust system identification algorithm compared with the previous method.

The convergence ratio is very low for every case of model 1 and for each case of model 2 with data more than N = 3000 in the previous method of Table 3. This is because that in the non-convergence case, the estimation accuracy of $\hat{\boldsymbol{a}}^{(1)}$ is not good and $\hat{\boldsymbol{b}}^{(1)}$ which is estimated next does not satisfy strictly positive realness condition. Generally, when $\hat{\boldsymbol{a}}^{(l)}$ and $\boldsymbol{b}^{(l)}$ are estimated in the state like above, further iterative estimation diverse. The reason of this is as follows.

For simplicity, set initial value of e_1, e_2 to be zero. Denote left hand side of Eq. (1) in (2, 2) order ARMA model as

$$X_n = x_n + \sum_{i=1}^{2} a_i x_{n-i} \tag{37}$$

then it follows

$$\begin{aligned} e_3 &= X_3 \\ e_4 &= X_4 - b_1 X_3 \\ e_5 &= X_5 - b_1 X_4 + (b_1^2 - b_2) X_3 \\ e_6 &= X_6 - b_1 X_5 + (b_1^2 - b_2) X_4 - b_1 (b_1^2 - 2b_2) X_3 \\ &\vdots \end{aligned} \tag{38}$$

Thus, terms are attached to the equation accumulatively as i for e_i increases. In this case, strictly positive realness condition is

$$|b_1| + |b_2| < 1 \tag{39}$$

Watching Eq. (38), for the case that does not satisfy strictly positive realness condition such as $|b_1|>1$, $|b_2|>1$, $\{e_i\}$ diverse. We also confirmed above relation in the numerical experiments.

There are the following dual phase characteristics in the data amount of N and parameter estimation. As N grows big, parameter estimation accuracy becomes good, which is stated later. On the other hand, when N grows big, divergence possibility of $\{e_n\}$ grows big because accumulative terms are attached as we have examined before. This becomes one of the reasons for the decrease of convergence ratio. Therefore, it can be safely said that when N grows big, the convergence ratio is not necessarily improved. In the proposed method of Table 3, even if N grows big, convergence ratio is not necessarily improved.

Another simulation case shows that the convergence ratio is 20% under N=1000, 30% for N=3000, 20% for N=10000. After watching many simulation cases, we cannot find certain common characteristics. Their convergence ratio varies from the range of 20% to 50%. We must have much more investigation to understand characteristics.

In Table 3, the proposed method exhibits a stable convergence ratio of nearly 30%, but the previous method's convergence ratio varies extremely. This is because initial estimation of $\hat{a}^{(1)}$ is nearly the same in the previous method even when we make 100 times simulation. Therefore, if the initial estimation of $\hat{a}^{(1)}$ is appropriate, it converges. Other simulation is similar. So, convergence ratio goes to 100%. If not, it goes to 0. On the other hand, initial estimation of $\hat{a}^{(1)}$ is properly scattered in proposed method. So, convergence ratio does not go extremely as the previous method. One reason for this is, some estimation may become worse by setting T_{-l} to 0 in Eq. (5) among many simulation cases. By improving this, convergence ratio may be improved.

At this stage, proposed method shows good improvement that nearly 30% converge for all cases. We make great progress for practical use. Improvement of convergence ratio is our further issue to be investigated.

Next, we compare the parameter estimation accuracy. Table 4shows the average of the following parameter estimate;

$$J_a = |\hat{a}_1 - a_1| + |\hat{a}_2 - a_2|$$

$$J_b = |\hat{b}_1 - b_1| + |\hat{b}_2 - b_2|$$

$$J = J_a + J_b$$

$\bar{J}_a$, $\bar{J}_b$ and $\bar{J}$ are the averages.

Estimation accuracy becomes good as the number of data grows large. With same number of data, both methods do not have much difference.

Finally, we make a comparison between the calculation times of both methods. In Table 5, each column holds the average calculation time of the converged cases.

As a whole, calculation times decrease from model 1

Table 4. Comparison of parameter estimation accuracy

		N			
		1000	3000	5000	10000
Model 1					
Proposed	$\bar{J}_a$	0.0343	0.0367	0.0054	0.0054
	$\bar{J}_b$	0.0544	0.0447	0.0254	0.0147
	$\bar{J}$	0.0887	0.0814	0.0308	0.0201
Previous	$\bar{J}_a$	–	–	–	–
	$\bar{J}_b$	–	–	–	–
	$\bar{J}$	–	–	–	–
Model 2					
Proposed	$\bar{J}_a$	0.0442	0.0435	0.0068	0.0309
	$\bar{J}_b$	0.0398	0.0376	0.0386	0.0211
	$\bar{J}$	0.0840	0.0811	0.0454	0.0520
Previous	$\bar{J}_a$	0.0817	0.0355	–	–
	$\bar{J}_b$	0.0511	0.0339	–	–
	$\bar{J}$	0.1328	0.0694	–	–
Model 3					
Proposed	$\bar{J}_a$	0.0947	0.1432	0.1546	0.0408
	$\bar{J}_b$	0.1133	0.0926	0.1050	0.0496
	$\bar{J}$	0.2080	0.2358	0.2596	0.0904
Previous	$\bar{J}_a$	0.1025	0.1382	0.1515	0.0356
	$\bar{J}_b$	0.1176	0.0901	0.1033	0.0468
	$\bar{J}$	0.2201	0.2283	0.2548	0.0824

– : cases which does not converge

Table 5. Comparison of calculation time

	N			
Model	1000	3000	5000	10000
Model 1				
Proposed	19.9132	36.7894	27.4806	38.0886
Previous	–	–	–	–
Model 2				
Proposed	2.0088	12.7160	12.0292	37.7286
Previous	1.9532	8.4582	–	–
Model 3				
Proposed	2.0509	14.2467	19.6144	22.9011
Previous	1.5437	8.4854	12.1460	19.9935

– : cases which does not converge

to 3, accordingly.

As the number of data grows large, the calculation time grows big which is a matter of course. Comparing the proposed method to the previous method, calculation time of previous method is slightly short in the converged cases. But converged cases and non-converged cases are mixed, so simple comparison is meaningless.

Though there are cases which converge and cases which do not converge, convergence cases are said to be better even if they take rather long calculation time.

Our aim is to get a converged solution, so comparing the time by which to grasp convergence case firstly is reasonable. In that case, as the previous method has many cases which do not converge, proposed method has shorter calculation time.

5.2 Remarks

As is shown in Table 3, every cases of the proposed method converge and each case, nearly 30% converge under different initial values. On the other hand, the previous methods have only half cases of convergence.

By introducing theoretical relations, we can neglect noisy terms which otherwise exist under finite number of data. So, this leads to robust parameter estimation. As for calculation time, considering the time by which we get converged solution, this proposed method, which has converged in all cases, is said to be a superior solution.

In the algorithm of Table 1, $\boldsymbol{a}$, $\boldsymbol{b}$ are estimated after estimating cross correlation function $\{T_i\}$. When $\{T_i\}$ are constant, the proposed algorithm converges as is proved in Appendix. In this paper, this is not the case, but is attached for reference.

6. CONCLUSION

In estimating ARMA model parameters, several bootstrap methods have been developed. However, there are many cases that they do not converge. This paper presents an improving method for this problem. Using cross correlation functions and utilizing an a priori knowledge of the structural characteristics of the model, we get robust parameter estimation. We can achieve the reduction of total time to get the solution which converges. We suppose that further extensions of this method to such as ARIMA model would be made from now on.

ACKNOWLEDGEMENTS

This research was partially supported by special research fund of Tokyo Metropolitan government.

APPENDIX

Here, we examine the characteristics of algorithms of bootstrap type in estimating $[\boldsymbol{a}^T\ \boldsymbol{b}^T]^T$.

For simplicity, we remove ^ on the estimated parameter.

In the case $\{b_i\} = 0$, Eq. (1) becomes an AR model and $\boldsymbol{a}$ is expressed as

$$\boldsymbol{a} = -\boldsymbol{R}^{-1}\boldsymbol{r} \tag{40}$$

So, we use this as an initial value of $\boldsymbol{a}$. From Eq. (16), we get

$$\boldsymbol{b} = \frac{1}{\sigma_e^2}\left(\boldsymbol{t} - \boldsymbol{T}^T\boldsymbol{R}^{-1}\boldsymbol{r}\right) \tag{41}$$

Substituting Eqs. (40), (41) to Eqs. (15), (16), we get

$$\boldsymbol{a} = -\boldsymbol{R}^{-1}\boldsymbol{r} + \frac{1}{\sigma_e^2}\boldsymbol{R}^{-1}\boldsymbol{T}\left(\boldsymbol{t} - \boldsymbol{T}^T\boldsymbol{R}^{-1}\boldsymbol{r}\right) \tag{42}$$

$$\boldsymbol{b} = \frac{1}{\sigma_e^2}\left\{\boldsymbol{t} - \boldsymbol{T}^T\boldsymbol{R}^{-1}\boldsymbol{r} + \frac{1}{\sigma_e^2}\boldsymbol{T}^T\boldsymbol{R}^{-1}\boldsymbol{T}\left(\boldsymbol{t} - \boldsymbol{T}^T\boldsymbol{R}^{-1}\boldsymbol{r}\right)\right\} \tag{43}$$

Repeating this similarly, we get

$$\begin{aligned}\boldsymbol{a} = &-\boldsymbol{R}^{-1}\boldsymbol{r} + \frac{1}{\sigma_e^2}\boldsymbol{R}^{-1}\boldsymbol{T}\left(\boldsymbol{t} - \boldsymbol{T}^T\boldsymbol{R}^{-1}\boldsymbol{r}\right) \\ &+ \frac{1}{\sigma_e^4}\boldsymbol{R}^{-1}\boldsymbol{T}\boldsymbol{T}^T\boldsymbol{R}^{-1}\boldsymbol{T}\left(\boldsymbol{t} - \boldsymbol{T}^T\boldsymbol{R}^{-1}\boldsymbol{r}\right)\end{aligned} \tag{44}$$

$$\boldsymbol{b} = \frac{1}{\sigma_e^2}\left\{\begin{aligned}&\boldsymbol{t} - \boldsymbol{T}^T\boldsymbol{R}^{-1}\boldsymbol{r} \\ &+ \frac{1}{\sigma_e^2}\boldsymbol{T}^T\boldsymbol{R}^{-1}\boldsymbol{T}\left(\boldsymbol{t} - \boldsymbol{T}^T\boldsymbol{R}^{-1}\boldsymbol{r}\right) \\ &+ \frac{1}{\sigma_e^4}\left(\boldsymbol{T}^T\boldsymbol{R}^{-1}\boldsymbol{T}\right)^2\left(\boldsymbol{t} - \boldsymbol{T}^T\boldsymbol{R}^{-1}\boldsymbol{r}\right)\end{aligned}\right\} \tag{45}$$

Eqs. (42), (43) have extra terms attached to Eqs. (40), (41). Also Eqs. (44), (45) to Eqs. (42), (43).

From Eq. (14), we get

$$\boldsymbol{P} = \begin{bmatrix}\boldsymbol{R} & -\boldsymbol{T} \\ -\boldsymbol{T}^T & \sigma_e^2\boldsymbol{I}\end{bmatrix}^{-1} = \begin{bmatrix}\boldsymbol{P}_{11} & \boldsymbol{P}_{12} \\ \boldsymbol{P}_{21} & \boldsymbol{P}_{22}\end{bmatrix} \tag{46}$$

where

$$\begin{aligned}\boldsymbol{P}_{11} &= \boldsymbol{R}^{-1} + \boldsymbol{R}^{-1}\boldsymbol{T}\left[\sigma_e^2\boldsymbol{I} - \boldsymbol{T}^T\boldsymbol{R}^{-1}\boldsymbol{T}\right]^{-1}\boldsymbol{T}^T\boldsymbol{R}^{-1} \\ &= \boldsymbol{R}^{-1} + \boldsymbol{R}^{-1}\boldsymbol{T} \\ &\quad\left[\frac{1}{\sigma_e^2}\sum_{n=0}^{\infty}\left(\frac{1}{\sigma_e^2}\boldsymbol{T}^T\boldsymbol{R}^{-1}\boldsymbol{T}\right)^n\boldsymbol{T}^T\boldsymbol{R}^{-1}\right]\end{aligned} \tag{47}$$

$$P_{12} = R^{-1}T\left[\sigma_e^2 I - T^T R^{-1} T\right]^{-1}$$
$$= R^{-1}T \frac{1}{\sigma_e^2} \sum_{n=0}^{\infty} \left(\frac{1}{\sigma_e^2} T^T R^{-1} T \right)^n \tag{48}$$

$$P_{21} = \left[\sigma_e^2 I - T^T R^{-1} T\right]^{-1} T^T R^{-1}$$
$$= \frac{1}{\sigma_e^2} \sum_{n=0}^{\infty} \left(\frac{1}{\sigma_e^2} T^T R^{-1} T \right)^n T^T R^{-1} \tag{49}$$

$$P_{22} = \left[\sigma_e^2 I - T^T R^{-1} T\right]^{-1}$$
$$= \frac{1}{\sigma_e^2} \sum_{n=0}^{\infty} \left(\frac{1}{\sigma_e^2} T^T R^{-1} T \right)^n \tag{50}$$

Using these equations, we get

$$a = -R^{-1}r + \frac{1}{\sigma_e^2} R^{-1}T\left(t - T^T R^{-1} r\right)$$
$$\sum_{n=0}^{\infty} \left(\frac{1}{\sigma_e^2} T^T R^{-1} T \right)^n \left(t - T^T R^{-1} r\right) \tag{51}$$

$$b = \frac{1}{\sigma_e^2} \begin{pmatrix} t - T^T R^{-1} r \\ + \frac{1}{\sigma_e^2} T^T R^{-1} T \\ \cdot \sum_{n=0}^{\infty} \left(\frac{1}{\sigma_e^2} T^T R^{-1} T \right)^n \left(t - T^T R^{-1} r\right) \end{pmatrix} \tag{52}$$

Excluding $\Sigma(*)$ in Eqs. (51), (52), they become same with Eqs. (40), (41).

And it is pointed out that

- When $n = 0$, Eqs. (51), (52) are equal to Eqs. (42), (43), respectively
- When n=1, Eqs. (51), (52) are equal to Eqs. (44), (45), respectively

Iterating the estimation of bootstrap type as of Eqs (40)-(45), finally we get Eqs. (51), (52).

As P is a positive definite matrix, Eq. (14) has a unique solution.

As is stated above, the algorithm of bootstrap type of Eqs. (40)-(45) becomes same with Eqs. (51), (52) under infinite iteration.

As Eqs. (51), (52) is equivalent to Eq. (14), the above algorithm of bootstrap type converges to the solution of Eq. (14).

REFERENCES

Katayama, T. (1994) *System Identification* (In Japanese), Asakura-Shoten Publishing.

Nakamura, M. and Oishi, Y. (1984) Generalized least square estimation algorithm designed to reduce a calculation time (In Japanese), *SICE Trans.* **20**, 471-478.

Sagara, S., Akitsuki, K., Nakamizo, T. and Katayama, T. (1994) *System Identification* (In Japanese), The Society of Instrument and Control Engineers.

Tokumaru, H., Soeda, T., Nakamizo, T. and Akitsuki, K. (1982) *Measurement and Analysis-Theory and Application of Random Data Handling* (In Japanese), Baifukan Publishnig.

Tokumaru, H. and Takeyasu, K. (1977) A new method for estimating the power spectral density function of stationary AR processes (In Japanese), *SICE Trans.* **13**, 148-153.