<table>
<thead>
<tr>
<th>Title</th>
<th>ARMA Digital Lattice Filter Based on New Criterion</th>
</tr>
</thead>
<tbody>
<tr>
<td>Author(s)</td>
<td>Miyanaga, Yoshikazu; Nagai, Nobuo; Miki, Nobuhiro</td>
</tr>
<tr>
<td>Citation</td>
<td>IEEE Transactions on Circuits and Systems, 34(6): 617-628</td>
</tr>
<tr>
<td>Issue Date</td>
<td>1987-06</td>
</tr>
<tr>
<td>Doc URL</td>
<td><a href="http://hdl.handle.net/2115/6047">http://hdl.handle.net/2115/6047</a></td>
</tr>
<tr>
<td>Rights</td>
<td>© 1987 IEEE. Personal use of this material is permitted. However, permission to reprint/republish this material for advertising or promotional purposes or for creating new collective works for resale or redistribution to servers or lists, or to reuse any copyrighted component of this work in other works must be obtained from the IEEE.</td>
</tr>
<tr>
<td>Type</td>
<td>article</td>
</tr>
<tr>
<td>File Information</td>
<td>ITCS34_6.pdf</td>
</tr>
</tbody>
</table>

Hokkaido University Collection of Scholarly and Academic Papers: HUSCAP
ARMA Digital Lattice Filter Based on New Criterion

YOSHIKAZU MIYANAGA, MEMBER, IEEE, NOBUO NAGAI, MEMBER, IEEE, AND NOBUHIRO MIKI, MEMBER, IEEE

Abstract—In this paper, a new ARMA digital lattice filter is proposed. First, a new criterion is defined for designing the ARMA digital lattice filter from given stochastic data. Based on this criterion, an ARMA parameter estimation algorithm is developed. This algorithm can estimate the ARMA parameters with little calculation cost. Then, from the fast algorithm of the ARMA parameter estimation, two elementary sections of the ARMA digital lattice filter are invented. Any ARMA model with an arbitrary AR order and an arbitrary MA order can be realized by using these elementary sections. In this paper, the proposed ARMA lattice filter is compared with other fast algorithms of ARMA parameter estimation and other ARMA lattice filters. Experimental results obtained from applying this filter to model reduction problems are shown. From these results, it is shown that this ARMA lattice filter is quite useful for ARMA digital signal processing.

I. INTRODUCTION

VARIOUS AUTOREGRESSIVE and moving-average (ARMA) digital filters have been introduced. It is shown in some papers [1]–[6] that an ARMA digital filter is quite useful for signal processing. The ARMA digital filter can represent both the concentration and the dispersion of power in the spectral domain, while an AR digital filter can represent only its concentration. By using this characteristic, we can analyze certain kinds of stochastic time series very accurately. For example, ARMA digital filters are recognized as superb filters in speech analysis [3], [4].

As one of these ARMA digital filters, an ARMA lattice filter has been designed. The construction of this filter is basically a cascaded-type digital filter. Owing to the cascaded type, this filter is precise for spectrum matching and analyzing a stochastic time series by adding a new section at the final stage without changing the filter form already obtained. The original characteristics of the ARMA lattice filter have been introduced [4]–[6]. In addition, various filter forms have been developed as ARMA lattice filters. These lattice filters are mainly derived from an AR lattice filter. Thus, these include the characteristics of the AR order of the ARMA lattice filter implicitly even if these characteristics are not desired.

This paper proposes a new ARMA lattice filter. This filter is based on a new criterion. The criterion is newly introduced to identify an ARMA model. Thus, the ARMA digital lattice filter derived from this criterion is not based on the design method of the conventional AR lattice filter. In other words, there are some original characteristics. We can easily design the ARMA digital lattice filter with an arbitrary AR order and an arbitrary MA order. In addition, every estimation error obtained in each cascaded section satisfies orthogonal conditions.

The new ARMA digital lattice filter is directly designed by an ARMA parameter estimation algorithm. This algorithm can estimate ARMA parameters with little calculation cost. The estimation parameters optimally makes the proposed criterion minimal. The proposed algorithm consists of two recursive formulas. One is an AR-type recursive formula. It can estimate ARMA parameters as the AR order of an ARMA estimation model increases by one. The other is an MA-type recursive formula. It can estimate ARMA parameters as the MA order of an ARMA estimation model increases by one. From these recursive formulas, we can develop two types of elementary sections used in an ARMA digital lattice filter (i.e., an AR-type elementary section and an MA-type elementary section associated with, respectively, the AR-type recursive formula and the MA-type recursive formula). The particularity of this filter is the design of an ARMA model with an arbitrary AR order and an arbitrary MA order. In other words, there is no restriction for each order, and no dependency of an AR model on an MA model or vice versa.

Section II introduces a criterion. Then, in Section III, four estimation errors are defined to derive a fast recursive method. Section IV develops the fast recursive method and Section V shows the elementary sections of an ARMA lattice filter associated with the fast recursive method. The proposed ARMA lattice filter has several unique characteristics because of the criterion and the fast recursive method. The differences between it and existing methods are shown in Section VI.

The final part of this paper shows some experimental results. It shows that the proposed ARMA lattice filter is quite suitable for a model reduction algorithm and that several ARMA lattice filters can be obtained from the observed data owing to the new criterion.

II. CRITERION

In this section, the criterion for an ARMA parameter estimation is defined. The criterion is based on linear prediction theory. All of ARMA parameters are calculated by minimizing the proposed criterion. In order to define
functions
functions:

tables
the criterion, let us assume the following two transfer functions:
\[ H_x(z^{-1}) = Z[h_x(k)] \]
\[ y(k) = h_x(k) * u(k) \]

\[ H_y(z^{-1}) = Z[h_y(k)] \]
\[ x(k) = h_y(k) * u(k) \]

where \(*\) denotes a convolution, and \(Z[\ ]\) denotes \(z\)-
transformation. The stochastic variable \(u(k)\) is a zero
mean white Gaussian process with a variance \(\sigma_u^2\). The
functions \(H_x(z^{-1})\) and \(H_y(z^{-1})\) are stable transfer functions.
According to (1), it turns out that stochastic variables
\(x(k)\) and \(y(k)\) are represented with \(u(i)\) \((i = 1, 2, \cdots, k)\).
Let us call \(H_x(z^{-1})\) and \(H_y(z^{-1})\) reference models.

For the reference models, define the following estimation
models:
\[ \hat{A}(z^{-1}) = 1 + \hat{a}_1 z^{-1} + \cdots + \hat{a}_s z^{-s} \]
\[ \hat{B}(z^{-1}) = \hat{b}_0 + \hat{b}_1 z^{-1} + \cdots + \hat{b}_t z^{-t} \]

(2)

where \(\hat{a}_i\) and \(\hat{b}_j\) \((i = 1, 2, \cdots, s\) and \(j = 0, 1, \cdots, t)\) are,
respectively, AR parameters and MA parameters (explained
later). From (1) and (2), define a new criterion as
\[ V_{s,t} = \frac{1}{2\pi \sigma_u^2} \int_{|z| = 1} |H_x(z^{-1})\hat{A}(z^{-1})| \]
\[ - H_x(z^{-1})\hat{B}(z^{-1})|^2 dz/z. \]

(3)

Minimizing this criterion corresponds to calculating
the coefficients of \(\hat{A}(z^{-1})\) and \(\hat{B}(z^{-1})\) which minimize the
mean value of \(\nu(k)^2\), shown in Fig. 1. In other words, by
using the Parseval equality, (3) is rewritten as
\[ V_{s,t} = E[(\hat{A}(z^{-1})y(k) - \hat{B}(z^{-1})x(k))^2] \]
\[ = E[(y(k) + \hat{a}_1 y(k-1) + \cdots + \hat{a}_s y(k-s) \]
\[ - \hat{b}_0 x(k) - \hat{b}_1 x(k-1) - \cdots - \hat{b}_t x(k-t)]^2. \]

(4)

The criterion of (3) is the generalized form of the
criterion proposed by Levinson and Mullis-Roberts. If the
reference model \(H_x(z^{-1})\) is zero, we get Levinson's criterion
from (3) [2]. This criterion has been used for AR parameter estimation
based on a least mean square problem.

If \(H_x(z^{-1}) = 1\), the Mullis-Roberts criterion is
obtained from (3) [7]. The Mullis-Roberts criterion has been
given as a modified least mean square problem for ARMA
parameter estimation.

The coefficients \(\hat{a}_i\) \((i = 1, \cdots, s)\) and \(\hat{b}_j\) \((j = 0, \cdots, t)\)
which minimize \(V_{s,t}\) of (4) are calculated as follows [2], [8],
[12]:
\[ \begin{bmatrix} \hat{a}_{s,t} & \hat{b}_0 & \hat{b}_{s,t} \end{bmatrix} R_{s,t} = \begin{bmatrix} \min V_{s,t} & 0 & 0 \end{bmatrix} \]

(5)

where \(\min V_{s,t}\) is the minimal value of \(V_{s,t}\), and each
vector and matrix are defined as
\[ \hat{a}_{s,t} = [\hat{a}_1, \cdots, \hat{a}_s] \]
\[ \hat{b}_{s,t} = [\hat{b}_0, \cdots, \hat{b}_t] \]
\[ R_{s,t} = E[\begin{bmatrix} h_x(k) \end{bmatrix}^T \begin{bmatrix} h_x(k) \end{bmatrix}] \]
\[ h_x(k) = [y(k) \cdots y(k-s)]^T \]

(6)

In the above equation, the superscript \(T\) denotes transpose.
In this paper, all vectors are defined as row vectors. Thus,
the column vector has the transpose \(T\) at its superscript.

Equation (5) is the normal equation for ARMA parameter
estimation. This is quite similar to the normal equation for
AR parameter estimation.

If \(x(k) (k = 1, 2, \cdots)\) are input signals and \(y(k) (k = 1, 2, \cdots)\) are output signals, the whole reference model
\(H(z^{-1})\) is given as
\[ H(z^{-1}) = H_x(z^{-1})/H_y(z^{-1}). \]

(7)

Equation (7) is derived from
\[ y(k) = H(z^{-1})x(k) \quad y(k) = H_y(z^{-1})u(k) \]
\[ x(k) = H_x(z^{-1})u(k) \]

(8)

According to (3), if \(V_{s,t} = 0\), we get
\[ H(z^{-1}) = H_y(z^{-1})/H_x(z^{-1}) \]
\[ = \hat{B}(z^{-1})/\hat{A}(z^{-1}). \]

(9)

Thus, \(\hat{A}(z^{-1})\) and \(\hat{B}(z^{-1})\) are called, respectively, the AR
part and the MA part of an estimated ARMA model.

Under ordinary circumstances, it is hard to find the ARMA
parameters which make \(V_{s,t} = 0\). If \(V_{s,t} \neq 0\) because of certain errors (for example, the disturbance for
the signal \(y(k)\) and \(x(k)\), and the calculation errors), the
ARMA parameters are calculated as the best approximation
on \(V_{s,t}\) of (3).

III. Four Prediction Errors
This section introduces the four estimation errors which
are employed to solve (5) with a small amount of calculation
cost. As defined in the previous section, \(x(k)\) and
\(y(k)\) are an input signal and an output signal, respectively.

For these input and output signals, let us define the
following four estimation models:
\[ \hat{x}_j(k) = - \sum_{j=1}^{t} \hat{b}_j x(k-i) + \sum_{j=1}^{s} \hat{a}_j y(k-j) \]

(10)

where \(\hat{x}_j(k)\) is a forward estimation signal for \(x(k)\); \(j = 0, 1, \cdots, t\)
\[ \hat{y}_j(k) = - \sum_{j=1}^{s} \hat{a}_j y(k-j) + \sum_{j=1}^{t} \hat{b}_j x(k-i) \]

(11)

where \(\hat{y}_j(k)\) is a forward estimation signal for \(y(k)\); \(j = 0, 1, \cdots, s\)
\[ \hat{x}_b(k-t) = - \sum_{j=1}^{t} \hat{b}_j x(k-i) + \sum_{j=1}^{s} \hat{a}_j y(k-j) \]

(12)

where \(\hat{x}_b(k-t)\) is a backward estimation signal for \(x(k-t)\).
\[ \hat{y}_b(k-s) = - \sum_{j=0}^{s-1} \hat{a}_j^r y(k-j) + \sum_{i=0}^{t-1} \hat{b}_i^r x(k-i) \quad (13) \]

where \( \hat{y}_b(k-s) \) is a backward estimation signal for \( y(k-s) \). From (10)–(13), the following estimation errors are defined at the time \( k \):

\[ \begin{align*}
\gamma_{s,t}(k) &= [\gamma_{s,t}^r(k) \, \gamma_{s,t}^i(k)] \\
\xi_{s,t}(k) &= [\xi_{s,t}^r(k) \, \xi_{s,t}^i(k)]
\end{align*} \]

where \( \gamma_{s,t}^r, \gamma_{s,t}^i, \xi_{s,t}^r, \) and \( \xi_{s,t}^i \) are defined as

\[ \begin{align*}
\theta_{s,t} &= [0 \, \delta_{s,t}^r \, \cdots \, \delta_{s,t}^r \, 1 \, \delta_{s,t}^i \, \cdots \, \delta_{s,t}^i] \\
\gamma_{s,t} &= [\delta_{s,t}^r \, \cdots \, \delta_{s,t}^r_1 \, 0 \, \delta_{s,t}^i \, \cdots \, \delta_{s,t}^i_1 \, 1] \\
\xi_{s,t} &= [\delta_{s,t}^r \, \cdots \, \delta_{s,t}^r_1 \, 1 \, \delta_{s,t}^i \, \cdots \, \delta_{s,t}^i_1 \, 0].
\end{align*} \]

The two forward estimation errors are represented in a vector (i.e., \( \gamma_{s,t}(k) \)). However, the two backward estimation errors are defined as \( \gamma_{s,t}(k) \) and \( \xi_{s,t}(k) \) independently. In the following section, these notations play important roles in ARMA parameter calculation and ARMA lattice filter design. The purpose in representing the backward errors independently, in contrast to the errors defined in other lattice filters [1], [5], [13], is the realization of the new lattice filter proposed in this paper (i.e., a filter based on an independent order update of the AR part and MA part).

The above estimation errors are determined as they satisfy the following orthogonal conditions:

\[ \begin{align*}
E[\gamma_{s,t}(k) \, y(k-j)] &= 0 \quad (0 \, 0) \\
E[\gamma_{s,t}(k) \, x(k-i)] &= 0 \quad (0 \, 0) \\
E[\gamma_{s,t}(k-1) \, y(k-j)] &= 0 \\
E[\gamma_{s,t}(k-1) \, x(k-i)] &= 0 \\
E[\xi_{s,t}(k-1) \, y(k-j)] &= 0 \\
E[\xi_{s,t}(k-1) \, x(k-i)] &= 0 \\
\end{align*} \]

From (16)–(18), we get the following equations:

\[ \begin{align*}
[\gamma_{s,t}, R_{s,t}] &= \begin{bmatrix} V_{s,t}^x \gamma_{s,t}^x & V_{s,t}^x \gamma_{s,t}^y \\ V_{s,t}^y \gamma_{s,t}^x & V_{s,t}^y \gamma_{s,t}^y \end{bmatrix} \\
[\xi_{s,t}, R_{s,t}] &= \begin{bmatrix} 0 & V_{s,t}^x \xi_{s,t}^y \\ 0 & V_{s,t}^y \xi_{s,t}^y \end{bmatrix}
\end{align*} \]

where

\[ \begin{align*}
V_{s,t}^x &= E\left[\left(\gamma_{s,t}(k-1)\right)^2\right] \\
V_{s,t}^y &= E\left[\left(\gamma_{s,t}(k-1)\right)^2\right] \\
V_{s,t}^{xy} &= E\left[\left(\gamma_{s,t}(k-1)\xi_{s,t}(k-1)\right)^2\right]
\end{align*} \]

From (19), each value in (5) is given as

\[ \begin{align*}
\hat{a}_i &= \hat{a}_i^r - \hat{a}_i^r V_{s,t}^{xy} / V_{s,t}^x \\
\hat{b}_0 &= - V_{s,t}^{xy} / V_{s,t}^x \\
\hat{b}_j &= \hat{b}_j^r - \hat{b}_j^r V_{s,t}^{xy} / V_{s,t}^x \\
\end{align*} \]

Equation (23) is derived as follows: In (5), the first element of the estimation parameter vector is 1. All elements except the first element in the right-hand side of (5) are zero. Thus, if the \((s+2)\)th element of the second row in the right-hand side of (19) (i.e., \( V_{s,t}^{xy} \)) becomes zero by using the first row in the right-hand side of (19), then the estimation parameter vector associated with this second row satisfies (5). In other words, (23) is satisfied.

If we solve the equations in (19) and (20), all of the values in (5) can be easily obtained from (23).

### IV. Recursive Formulas to Solve the Normal Equation for ARMA Parameters

In the previous section, we introduced four estimation models and four estimation errors. The estimation parameter matrix and vectors (i.e., \( \theta_{s,t}, \gamma_{s,t}, \xi_{s,t}, \)) satisfy (19) and (20). In this section, let us show how to calculate the matrix and vectors with little calculation cost. If we calculate an inverse matrix to solve the given matrix equation in (5), we need the calculation cost in proportion to \((s + t + 1)^3\). But if the recursive formulas are used as we propose later, we need just the calculation cost in proportion to \((s + t + 1)^2\).

There are two types of recursive formulas. One is the AR-type recursive formula, which calculates ARMA parameters as the AR order of an estimation model increases by one. The other is the MA-type recursive formula, which calculates ARMA parameters as the MA order of an estimation model increases by one. Before deriving these formulas, let us define the four matrices \( I_1, I_2, I_3, \) and \( I_4 \) with \((s + t + 2 \times s + t + 3)\) dimension:

\[ \begin{align*}
I_1 &= [w_1 \cdots w_{s+1} \mid w_{s+2} \cdots w_{s+t+2}]
\end{align*} \]

\[ \begin{align*}
I_2 &= [w_1 \cdots w_{s+1} \mid w_{s+2} \cdots w_{s+t+2}]
\end{align*} \]

\[ \begin{align*}
I_3 &= [w_1 \cdots w_{s+1} \mid w_{s+2} \cdots w_{s+t+2}]
\end{align*} \]

\[ \begin{align*}
I_4 &= [w_1 \cdots w_{s+1} \mid w_{s+2} \cdots w_{s+t+2}]
\end{align*} \]

The above matrix is easily obtained by a \((s + t + 2)\)-dimensional vector \( e \), where all elements of \( e_i \) are zero except the \( i \)th element, which is 1. For example, \( I_1 \) is given as

\[ I_1 = \begin{bmatrix} e_1^T & e_2^T & \cdots & e_{s+1}^T & 0^T & e_{s+2}^T & \cdots & e_{s+t+2}^T \end{bmatrix}. \]
These matrices are used in the recursive formulas introduced later.

### A. AR-Type Recursive Formula

Let us show the AR-type recursive formula. This recursive formula estimates the parameters of an ARMA estimation model as the AR order of this model increases by one. In other words, the AR-type recursive formula assumes that the parameters of the ARMA estimation model with AR order and MA order, abbreviated as the (s, t) ARMA parameters, have already been calculated, and then calculates the (s + 1, t) ARMA parameters. Since the AR order increases by one after this calculation, this formula is regarded as the AR order update algorithm which allows ARMA parameter estimation.

By using (24), we get

\[ \begin{bmatrix} \theta_{s+1,t} \end{bmatrix} = \begin{bmatrix} \theta_{s,t} \\ \xi_{s,t} \end{bmatrix} R_{s+1,t} = \begin{bmatrix} V_{s+1,s}^y & 0 \\ V_{s+1,s}^x \\ V_{s+1,t} \\ 0 \\ 0 \\ 0 \\ V_{s+1,t} \\ V_{s+1,t} \\ V_{s+1,t} \\ 0 \\ 0 \\ 0 \\ \end{bmatrix} \]

(27)

\[ \begin{bmatrix} \gamma_{s+1,t} \\ \xi_{s+1,t} \end{bmatrix} R_{s+1,t} = \begin{bmatrix} \gamma_{s,t} \\ \xi_{s,t} \end{bmatrix} = \begin{bmatrix} \gamma_{s,t} \\ \xi_{s,t} \end{bmatrix} + \begin{bmatrix} \mu_1 \xi_{s,t} \\ \mu_2 \xi_{s,t} \\ \mu_3 \xi_{s,t} \end{bmatrix} \]

(28)

(29)

where

\[ \begin{bmatrix} \tau_1 \\ \tau_2 \end{bmatrix} = E\left[\xi_{s,t},(k-1)v_{s,t}(k)\right] \]

\[ \tau_3 = -E\left[y(k-s-1)\gamma_{s,t}(k)\right] \]

\[ \tau_4 = E\left[y(k-s-1)\xi_{s,t}(k)\right]. \]

Equations (27)–(29) are derived in Appendix II. As already defined, the AR-type recursive formula calculates the parameter matrix and vectors \( \theta_{s+1,t}, \gamma_{s+1,t}, \) and \( \xi_{s+1,t} \) by using (27)–(29). From (19) and (20), \( \theta_{s+1,t}, \gamma_{s+1,t}, \) and \( \xi_{s+1,t} \) satisfy

\[ \begin{bmatrix} \theta_{s+1,t} \\ \gamma_{s+1,t} \\ \xi_{s+1,t} \end{bmatrix} R_{s+1,t} = \begin{bmatrix} \theta_{s,t} \\ \gamma_{s,t} \\ \xi_{s,t} \end{bmatrix} + \begin{bmatrix} \mu_1 \xi_{s,t} \\ \mu_2 \xi_{s,t} \\ \mu_3 \xi_{s,t} \end{bmatrix} \]

(30)

Comparing the right-hand sides of (27)–(29) with those of (30), we can obtain (30) if the (s + 2)th elements in the first and second rows of (27) become zero and the (s + 1)th elements in the first and second rows of (28) become zero. In order to make these elements zero, this recursive formula employs (29). The AR-type recursive formula is given as

\[ \begin{bmatrix} \theta_{s+1,t} \\ \gamma_{s+1,t} \\ \xi_{s+1,t} \end{bmatrix} = \begin{bmatrix} \theta_{s,t} \\ \gamma_{s,t} \\ \xi_{s,t} \end{bmatrix} + \begin{bmatrix} \mu_1 \xi_{s,t} \\ \mu_2 \xi_{s,t} \\ \mu_3 \xi_{s,t} \end{bmatrix} \]

(31)

The above equations are derived in Appendix III.

### B. MA-Type Recursive Formula

Next let us show the recursive formula which calculates ARMA parameters as the MA order of an estimation model increases by one. This formula assumes that the \( (s, t) \) ARMA parameters have already been calculated, and then calculates the \( (s, t+1) \) ARMA parameters. Since the MA order increases by one after this calculation, this formula is regarded as the MA order update algorithm which allows ARMA parameter estimation.

According to (15)–(20), the MA-type recursive formula is equal to the AR-type recursive formula if the variables associated with the output signal \( y(k) \) and the variables associated with the input signal \( x(k) \) are interchanged. Thus, if we replace \( y(k), \gamma_{s,t}, \) and \( -\gamma_{s,t}(k) \) with \( -x(k), \xi_{s,t}, \) and \( \xi_{s,t}(k), \) and vice versa, then the MA-type recursive formula is derived. The MA-type recursive formula is given as

\[ \begin{bmatrix} \theta_{s,t+1} \\ \gamma_{s,t+1} \\ \xi_{s,t+1} \end{bmatrix} = \begin{bmatrix} \theta_{s,t} \\ \gamma_{s,t} \\ \xi_{s,t} \end{bmatrix} + \begin{bmatrix} \eta_1 \xi_{s,t} \\ \eta_2 \gamma_{s,t} \\ \eta_3 \gamma_{s,t} \end{bmatrix} \]

(34)

where

\[ \begin{bmatrix} \tau_1 \\ \tau_2 \end{bmatrix} = E\left[\gamma_{s,t},(k-1)v_{s,t}(k)\right] \]

\[ \tau_3 = -E\left[y(k-s-1)\gamma_{s,t}(k)\right] \]

\[ \tau_4 = E\left[y(k-s-1)\xi_{s,t}(k)\right]. \]

Equations (32)–(33) are derived in Appendix IV. As already defined, the MA-type recursive formula calculates the parameter matrix and vectors \( \theta_{s,t+1}, \gamma_{s,t+1}, \) and \( \xi_{s,t+1} \) by using (32)–(33). From (19) and (20), \( \theta_{s,t+1}, \gamma_{s,t+1}, \) and \( \xi_{s,t+1} \) satisfy

\[ \begin{bmatrix} \theta_{s,t+1} \\ \gamma_{s,t+1} \\ \xi_{s,t+1} \end{bmatrix} R_{s,t+1} = \begin{bmatrix} \theta_{s,t} \\ \gamma_{s,t} \\ \xi_{s,t} \end{bmatrix} + \begin{bmatrix} \eta_1 \xi_{s,t} \\ \eta_2 \gamma_{s,t} \\ \eta_3 \gamma_{s,t} \end{bmatrix} \]

(35)

Comparing the right-hand sides of (32)–(33) with those of (35), we can obtain (35) if the (s + 2)th elements in the first and second rows of (32) become zero and the (s + 1)th elements in the first and second rows of (33) become zero. In order to make these elements zero, this recursive formula employs (33). The MA-type recursive formula is given as
inverse matrix algorithm which estimates ARMA parameters in (5) by using an inverse matrix. Thus, let us call these formulas the fast recursive method, abbreviated FRM.

If the two proposed recursive algorithms join together, we get the ARMA-type recursive formula which calculates ARMA parameters as the AR and MA orders of an estimation model increase by one simultaneously. This formula is not essential but is useful in some cases. This ARMA-type recursive formula has already been proposed in [5] and [8].

V. ELEMENTARY SECTIONS USED IN ARMA DIGITAL LATTICE FILTER

Levinson's algorithm is the recursive formula which solves the normal equation for the AR model with the lowest calculation cost. The AR digital lattice inverse filter can be designed by using the prediction errors defined in this recursive formula. The term "inverse filter" has been coined to describe the whitening filter. The observed stochastic signals are inputted to the whitening filter and the whitened stochastic signals are obtained from this filter. In addition, an AR lattice filter can be designed from the obtained AR lattice inverse filter easily. This filter synthesizes the observed stochastic signals with white Gaussian input signals.

A similar procedure is applied to the recursive formulas given in the previous section. In other words, from the proposed formulas, we can easily design an ARMA digital lattice inverse filter. Then, the ARMA digital lattice filter is also designed. Associated with the recursive formulas (i.e., the AR-type recursive formula and the MA-type recursive formula), we can design two elementary sections used in an ARMA digital lattice filter. Let us call the two sections the AR-type elementary lattice inverse section and the MA-type elementary lattice inverse section which correspond to, respectively, the AR-type recursive formula and the MA-type recursive formula.

Let us design the AR-type elementary lattice inverse section first. From (6) and (24), the following equations are satisfied:

\[
\begin{align*}
\hat{h}_{s,t+1}(k) I_t^T &= [y(k) \cdots y(k-s) ; \ldots -x(k) \cdots -x(k-t+1) - x(k-t)] \\
\hat{h}_{s,t+1}(k) I_t^T &= [y(k-1) \cdots y(k-s-1) ; \ldots -x(k-1) \cdots -x(k-t)].
\end{align*}
\]

(37)

Thus, the AR-type error recursions are derived from (31) as follows:

\[
\begin{align*}
\nu_{s,t+1}(k) &= \nu_{s,t}(k) + [\mu_1^T \mu_2^T] \xi_{s,t}(k-1) \\
\gamma_{s,t+1}(k) &= \gamma_{s,t}(k) - \mu_3 \xi_{s,t}(k) \\
\xi_{s,t+1}(k) &= \xi_{s,t}(k-1) + [\mu_3^T \mu_4^T] \nu_{s,t}(k)^T - \mu_4 \gamma_{s,t+1}(k)
\end{align*}
\]

(38)

where \( \mu_1 = [\mu_1^T \mu_2^T] \) and \( \mu_3 = [\mu_3^T \mu_4^T] \). Equations (37) and (38) are derived in Appendix IV. From (38), the AR-type elementary lattice inverse section is designed in Fig. 2. In this figure, two forward estimation errors (i.e., \( \nu_{s,t}(k) \) and \( \nu_{s,t}(k) \)) are used instead of \( \nu_{s,t}(k) \).

The error recursions for the MA-type elementary lattice inverse section are derived from (34) as follows:

\[
\begin{align*}
\nu_{s,t+1}(k) &= \nu_{s,t}(k) - [\eta_1^T \eta_2^T] \gamma_{s,t}(k-1) \\
\xi_{s,t+1}(k) &= \xi_{s,t}(k) - \eta_2 \gamma_{s,t}(k) \\
\gamma_{s,t+1}(k) &= \gamma_{s,t}(k-1) - [\eta_3^T \eta_4^T] \nu_{s,t}(k)^T - \eta_4 \xi_{s,t+1}(k)
\end{align*}
\]

(39)

where \( \eta_1 = [\eta_1^T \eta_2^T] \) and \( \eta_3 = [\eta_3^T \eta_4^T] \). The above recursions are obtained as the complementary form of the AR-type error recursions.
The MA-type elementary lattice inverse section is designed in Fig. 3. From the lattice inverse sections designed above, we can design the two lattice sections (i.e., the AR-type lattice elementary section and the MA-type lattice elementary section) shown in Figs. 4 and 5, respectively. In addition, an ARMA digital lattice filter is constructed by these sections, as shown in Fig. 6.

VI. COMPARISONS BETWEEN THE PROPOSED ARMA LATTICE FILTER AND THE OTHER ARMA LATTICE FILTERS

This paper discusses the new FRA and its ARMA lattice filter realization. This method assumes that the correlation data of the output and the input sequences have already been given. In other words, \( R_{x,t} \) of (5) is first given and then the \( (s, t) \) ARMA parameters are calculated by FRA. The values \( \tau_1, \tau_4, \tau_4' \) used in FRA are easily calculated from the elements of \( R_{x,t} \). Thus, this method is regarded as batch processing.

Some other fast algorithms and ARMA lattice filters have been developed. This section compares these algorithms and ARMA lattice filters with our proposed FRA and its lattice filter. Although the algorithms already proposed are essentially batch processing, the adaptive approaches are also derived in the advanced forms. However, the advanced forms slightly change the subject in this paper. In the following discussion, we therefore consider the batch-processing methods. In addition, every method is compared at the orthogonal conditions of estimation errors and at the possibility of an ARMA order update. The conventional fast algorithms and the ARMA lattice filters are put in the following groups:

1) **Lattice joint process** [4];
2) **ARMA lattice filter** by Lee–Morf–Friedlander [1], [5];
3) **Lattice joint process** by Benveniste–Chaure [11];
4) **FRA by Monden–Yamada–Arimoto** [12].

The lattice joint process estimator (i.e., (VI-1)) developed in [4] is depicted in Fig. 7. In this figure, the orthogonal conditions of \( \xi_s(k) \) and \( \xi_s(k) \) are the same as (16) and (18). However, there is no orthogonal condition among \( e_s \). In other words, this filter guarantees the orthogonal conditions on the output \( y(k) \) but does not satisfy the orthogonal conditions on the input \( x(k) \). Each \( e_s(k) \) is represented with \( \xi_s(k) \) and \( x(k) \) is represented with \( e_s(k) \). Thus, if the stochastic property of \( x(k) \) is different from that of \( y(k) \), this filter may require many elementary sections. Our ARMA lattice filter can realize all the orthogonal conditions on \( y(k) \) and \( x(k) \). Thus, this compensates for the above disadvantage.

The ARMA lattice filter of (VI-2) also satisfies the orthogonal conditions on \( y(k) \) and \( x(k) \). The elementary section of this filter is shown in Fig. 8. This section is easily designed by cascading our two elementary sections (i.e., the AR-type elementary lattice inverse section and the MA-type elementary lattice inverse section). This filter is realized as the ARMA-type recursive formula which calculates the \( (s + 1, s + 1) \) ARMA parameters using the assumption that the \( (s, s) \) ARMA parameters have already been given. Note that the recursive algorithms of (VI-2) are the simultaneous AR and MA order update. Thus, it cannot increase an AR order or an MA order with an arbitrary arrangement, whereas the new algorithm can. This means that the (VI-2) realization of the ARMA lattice filter is restricted with \( s = t \).

The FRA of (VI-3) requires the same restriction (i.e., \( s = t \)). In addition, this algorithm solves the mean square problem of (3) where \( H_{F}(z^{-1}) = 1 \). In other words, this algorithm assumes that the input \( x(k) \) is a white Gaussian process (i.e., \( x(k) = u(k) \)). Thus, it is not applied to the
reference model with colored input signals. However, the
calculation cost of this algorithm is the smallest among
ARMA-type FRA's including our algorithm.

The algorithm of (VI-4) can realize the ARMA model
with arbitrary AR and MA orders (i.e., $s \neq t$). In addition,
$\delta$ can be applied to the model with colored input signals.
However, it needs weak restriction. In other words, this
algorithm consists of

Type 1: Evaluation of the $(s+1, t+1)$ error set as a
function of the $(s, t)$ error set.

Type 2: Evaluation of the $(s+1, 0)$ error set as a
function of the $(s, 0)$ error set.

Type 3: Evaluation of the $(0, t+1)$ error set as a
function of the $(0, t)$ error set.

Here, the error set denotes the four estimation errors
defined in (14). Thus, the (VI-4) algorithm is an expanded
form of (VI-1) and (VI-2). In other words, the recursion of
the forward estimation errors in (VI-4) is the same as that
of the (VI-1) algorithm. If $s = t$, the (VI-4) algorithm
corresponds to the (VI-2) algorithm. Although this
can realize the ARMA model with arbitrary
ARMA orders, the arrangement for each filter section is
restricted. According to Type 2 and Type 3, this algorithm
has to realize either the AR part or the MA part at the first
time if $s \neq t$. In other words, it cannot realize the ARMA
model with arbitrary arrangement. For example, this
algorithm cannot realize the ARMA model with the
arrangement of $(0, 1), (1, 1), (2, 1)$, and $(3, 1)$ ARMA order. In this
example, it has to realize this model with only the arrangement
of $(1, 0), (2, 0)$, and $(3, 1)$. The new algorithm
proposed in this paper can realize this model without any
restrictions for the arrangement. If the orders of the
estimation model are already known, it is enough to employ
(VI-4). However, if any information of its orders has not
been given, the new algorithm may be better than (VI-4).

The algorithm (VI-5) can realize any ARMA model with
any arrangement for the orders of the estimation model.
However, the errors given in this algorithm, if they can be
calculated, do not satisfy orthogonal conditions even for the
forward estimation model.

The proposed algorithm compensates for the disad-
antage of the above algorithms. It has been shown that
each of the algorithms above shows excellent ability for a
certain kind of signal processing. However, the proposed
FRA together with its lattice filter also presents the same
ability since it is an expanded form of the others.

VII. EXPERIMENTAL RESULTS

We carried out some experiments for the proposed
ARMA digital lattice filter. In this section, let us show the
experimental results. The experiments were performed to
solve a model reduction problem. The model reduction
problem has been established in system identification [9],
[10]. This is used to yield a reduced model from a given
reference model with either high order or infinite order. In
this section, we show three different ways of approaching
the reduction problems by using the proposed ARMA
digital lattice filter.

Assume we have the reference model whose spectrum
and pole-zero locations are shown in Fig. 9. This model is
the ARMA model with 14 AR order and 10 MA order.
According to the spectrum, the reference model has four
resonances with narrow bandwidth, i.e., 1 kHz, 1.25 kHz, 3
kHz, and 4 kHz (resonance frequency), and two antireso-
nances with wide bandwidth, i.e., 2.5 kHz and 4.5 kHz
(antiresonance frequency).

First, let us consider the following criterion:

$$V_{r, i} = \sigma_i^2 / (2\pi j) \int_{|z|=1} |B(z^{-1})\hat{A}(z^{-1})|^2 dz/z$$

where the reference model $H(z^{-1})$ is given as
$B(z^{-1})/A(z^{-1})$. In the above criterion, $H_p(z^{-1})$ and
$H_x(z^{-1})$ of (3) are $A(z^{-1})$ and $B(z^{-1})$, respectively. Thus,
the estimated ARMA model is given by
$\hat{H}(z^{-1}) = \hat{B}(z^{-1})/\hat{A}(z^{-1})$. Since $B(z^{-1})$ and $A(z^{-1})$
had been already given, we could design the ARMA lattice filter
which minimized $V_{r, i}$ by using the proposed method. The
designed ARMA lattice filter is shown in Fig. 10. The left
four sections of this filter were designed with the ARMA-
type lattice section. These sections were designed by the
two proposed elementary sections. The right four sections
Fig. 11. (a) Spectrum and (b) pole-zero locations of the designed ARMA lattice filter based on $\nu_{s,t}$ of (40).

were the AR-type lattice elementary sections. In this filter, the four ARMA-type lattice sections were designed first. As discussed in the previous section, any other algorithms cannot design this filter with this arrangement. The AR and the MA order were 8 and 4. The spectrum and the pole-zero locations of this filter are shown in Fig. 11. This filter is unstable according to the pole-zero locations.

Next, let us consider the following criterion:

$$V'_{s,t} = \sigma_u^2/(2\pi j) \int (B(z^{-1})/A(z^{-1}))$$

$$A(z^{-1}) - \hat{B}(z^{-1})|^2 dz/z \quad |z|=1. \quad (41)$$

This criterion is the same as the modified least square criterion introduced by Mullis and Roberts. To calculate the parameters which minimize $V'_{s,t}$, we needed the second-order information and the first-order information (i.e., autocorrelation data and impulse responses of the reference model). In this experiment, after we calculated the 256 points of the impulse response of $B(z^{-1})/A(z^{-1})$, both data were calculated. The estimated ARMA lattice filter was the same as that in Fig. 10. Although the criterion is the same as that of Mullis and Roberts, we could design the ARMA lattice filter with a different ARMA order (i.e., $s=8$ and $t=4$). The spectrum and the pole-zero locations of the designed ARMA lattice filter are shown in Fig. 12. Comparing this filter with the reference model, the spectrum envelope of this estimated filter is quite similar to that of the reference model. In addition, all the poles and zeros lie inside the unit circle (i.e., this filter is stable). Thus, this filter is considered as a fine reduced ARMA model.

Let us show the other example. The criterion is defined as

$$V''_{s,t} = \sigma_u^2/(2\pi j) \int (\hat{A}(z^{-1})/A(z^{-1}))$$

$$- \hat{B}(z^{-1})/B(z^{-1})|^2 dz/z \quad |z|=1. \quad (42)$$

We needed both the impulse response of $1/A(z^{-1})$ and the impulse response of $1/B(z^{-1})$ to calculate the estimation ARMA parameters. We calculated the 256 points of these impulse responses. The designed ARMA lattice filter was the same as in Fig. 10. The spectrum and the pole-zero locations of this filter are shown in Fig. 13. This spectrum and the pole-zero locations are similar to those of the previous lattice filter. Thus, this filter is also the fine reduced ARMA lattice filter.

Now, let us consider why the estimated lattice filter has different characteristics for different criteria. The ARMA parameters of each filter are estimated to minimize each criterion. The estimation model $\hat{A}(z^{-1})$ and $\hat{B}(z^{-1})$ have to identify the resonances and the antiresonances of the reference model. Thus, in $V_{s,t}$ of (40), $\hat{A}(z^{-1})$ and $\hat{B}(z^{-1})$ have to identify the zero points of $A(z^{-1})$ and $B(z^{-1})$ in the unit circle, respectively. However, even if an estimated zero point is slightly shifted from the given zero point, the value of $V_{s,t}$ is not greatly changed. In other words, the sensitivity of the estimation is poor because the estimation error around the resonances and the antiresonances does not influence the criterion greatly. On the other hand, for $V''_{s,t}$, $\hat{A}(z^{-1})$ and $\hat{B}(z^{-1})$ have to identify the pole points of $A(z^{-1})$ and the zero points of $B(z^{-1})$ in the unit circle. Thus, if an estimated pole point of $\hat{A}(z^{-1})$ is slightly shifted from the given pole point, the value of $V''_{s,t}$ is
Recently changed. It shows that the sensitivity of the estimation for $A(z^{-1})$ is quite good. In addition, it is shown that the sensitivity of the estimation for $A(z^{-1})$ and $B(z^{-1})$ is also good concerning $V_{z}'$. Thus, we can get the accurate reduced model for $V_{z}'$ and $V_{z''}'$.


delimiters
Thus, we get (20) from (A6) - (A8).

Thus, using (17) and (18) again, \( \beta_1, \beta_2, \beta_3, \) and \( \beta_4 \) in (A6) are given as
\[
\begin{align*}
\beta_1 &= -E[\gamma_{s,t}(k-1)\xi_{s,t}(k-1)] \\
&= -E[\gamma_{s,t}(k-1)\xi_{s,t}(k-1)] = V_{\gamma_{s,t}} \\
\beta_2 &= E[\gamma_{s,t}(k-1)x(k-s-1)] \\
&= E[(\gamma_{s,t}(k-1))^2] = V_{\gamma_{s,t}} \\
\beta_3 &= E[\xi_{s,t}(k-1)y(k-s-1)] \\
&= E[(\xi_{s,t}(k-1))^2] = V_{\xi_{s,t}} \\
\beta_4 &= -E[\gamma_{s,t}(k-1)\xi_{s,t}(k-1)] = V_{\gamma_{s,t}}.
\end{align*}
\]
In addition, using (A1) and (A5), the left-hand side of (A6) is rewritten as
\[
E\left[ \begin{bmatrix} -\gamma_{s,t}(k-1) \\ \xi_{s,t}(k-1) \end{bmatrix} h_{s,t}(k-1) \right] = E\left[ \begin{bmatrix} \gamma_{s,t} \\ \xi_{s,t} \end{bmatrix} h_{s,t}(k-1) \right] = \begin{bmatrix} \gamma_{s,t} \\ \xi_{s,t} \end{bmatrix} R_{s,t}.
\]
Thus, we get (20) from (A6) - (A8).

**Appendix II**

**Derivation of (27) - (29)**

Let us derive (27). According to (24), \( \theta_{s,t} I_1 \) is given as
\[
\begin{bmatrix}
0 \\
1
\end{bmatrix} \begin{bmatrix}
\delta_{s} \\
\beta_{s}
\end{bmatrix} = \begin{bmatrix}
\delta_{s} \\
\beta_{s}
\end{bmatrix}.
\]

Thus, using \( R_{s,t} = E[h_{s,t}(k)^T h_{s,t}(k)] \), we get
\[
\begin{bmatrix}
\gamma_{s,t} \\
\xi_{s,t}
\end{bmatrix} = \begin{bmatrix}
V_{\gamma_{s,t}} \\
V_{\xi_{s,t}}
\end{bmatrix} \begin{bmatrix}
\tau_1 \\
\tau_2
\end{bmatrix} = \begin{bmatrix}
\gamma_{s,t} \\
\xi_{s,t}
\end{bmatrix} R_{s,t}.
\]

Where (14) and (A3) are used in (A10), and \( \tau_1 = -E[v_{s,t}(k)\gamma_{s,t}(k-1)] \) and \( \tau_2 = E[v_{s,t}(k)\gamma_{s,t}(k-1)] \).

By using (16), the values \( \tau_1 \) and \( \tau_2 \) are rewritten as
\[
\begin{bmatrix}
\tau_1 \\
\tau_2
\end{bmatrix} = \begin{bmatrix}
\gamma_{s,t}(k-1) \\
\xi_{s,t}(k-1)
\end{bmatrix} v_{s,t}(k).
\]

Thus, from (A10) and (A11) we get (27).

Next, let us derive (28). Using (24), we get
\[
\begin{bmatrix}
\gamma_{s,t} \\
\xi_{s,t}
\end{bmatrix} = \begin{bmatrix}
\delta_{s} \\
\beta_{s}
\end{bmatrix} = \begin{bmatrix}
\delta_{s} \\
\beta_{s}
\end{bmatrix}.
\]

Equation (29) is derived as follows. Using (24), we get
\[
\begin{bmatrix}
\gamma_{s,t} \\
\xi_{s,t}
\end{bmatrix} = \begin{bmatrix}
\delta_{s} \\
\beta_{s}
\end{bmatrix} = \begin{bmatrix}
\gamma_{s,t} \\
\xi_{s,t}
\end{bmatrix} R_{s,t}.
\]

Note that the (1,1)th, (1, s + 3)th, (2,1)th and (2, s + 3)th elements of the right-hand side in (A18) must be 0, 1, 1, and 0, respectively, since the elements of \( \theta_{s,t} I_1 \), associated with their positions are 0, 1, 1, and 0, respectively.
Next, consider the following equation:
\[
\begin{bmatrix} V_{s,1}^\xi & \mu_2 V_{s,1}^\eta \end{bmatrix} \cdot I_1 \cdot \begin{bmatrix} R_{s+1,1}^\xi \\
\end{bmatrix} = \begin{bmatrix} 0 & V_{s,1}^\gamma \end{bmatrix} \cdot \begin{bmatrix} \gamma_{s,1}^\xi + \mu_2 \tau_4 \end{bmatrix}.
\]
Equation (A20) is derived from (28). Thus, if \( \mu_2 = -V_{s,1}^\gamma / V_{s,1}^\xi \), then the \((s+1)\)th element of the right-hand side becomes zero. From (A20) and (30), we get
\[
\gamma_{s+1,1}^\xi = [\gamma_{s,1}^\xi + \mu_2 \xi_{s,1}] I_1
\]
\[
\gamma_{s+1,1}^\eta = -V_{s,1}^\gamma / V_{s,1}^\xi
\]
\[
\gamma_{s+1,1}^\tau = \tau_3 + \mu_2 \tau_4
\]
\[
\gamma_{s+1,1}^\gamma = V_{s,1}^\gamma + \mu_2 V_{s,1}^\eta.
\]
Note that the \((s+2)\)th and \((s+t+3)\)th elements of the right-hand side in (A21) must be 0 and 1 since the elements \(\gamma_{s+1,1}^\xi\) associated with their positions are 0 and 1, respectively.

Finally, let us derive the recursion of \(\xi_{s+1,1}^\tau\). We have to notice that the \((s+2)\)th and \((s+t+3)\)th elements of \(\xi_{s+1,1}^\tau\) are 1 and 0, respectively. First, consider the following equation:
\[
\mu_3 \theta_{s,1} R_{s+1,1} = -[\tau_1 \tau_2] \cdot \begin{bmatrix} V_{s,1}^\gamma & 0 \tau_1 & \tau_2 | V_{s,1}^\gamma & 0 \end{bmatrix} - \begin{bmatrix} -\tau_1 \tau_2 \lambda & -\tau_1 0 \end{bmatrix}
\]
where (27) is used in (A23) and
\[
\mu_3 = -[\tau_1 \tau_2] \cdot \mu_2^2.
\]
\[
\mu_3 = -[\tau_1 \tau_2] \cdot \mu_2^2
\]
\[
\lambda = [\tau_1 \tau_2] \cdot \mu_3^2.
\]

Thus, from (29) and (A23), we get
\[
(\xi_{s+1,1}, I_2 + \mu_3 \theta_{s,1}, I_1) \cdot R_{s+1,1} = \begin{bmatrix} 0 & V_{s,1}^\gamma & \lambda & 0 & 0 \end{bmatrix}.
\]

However, from (A9) and (A14), the vector \(\xi_{s,1}, I_2 + \mu_3 \theta_{s,1}, I_1\) is
\[
[0 \ a_0 \cdots a_{s-1} 1 0 b_0 \cdots b_{s-1}] + \mu_3 [0 \ a_0 \cdots a_1 0 1 b_0 \cdots b_{s-1}]
\]
\[
+ \mu_3 [0 \ a_0 \cdots a_2 0 0 b_0 \cdots b_{s-1}].
\]
In (A26), the \((s+t+3)\)th element (i.e., the rightmost element) does not become zero. Thus, we have to make it zero in order to obtain \(\xi_{s+1,1}\). Let us consider the following equation:
\[
(\xi_{s+1,1} - \mu_4 \gamma_{s+1,1}) R_{s+1,1} = \begin{bmatrix} 0 & V_{s,1}^\gamma & -\mu_4 V_{s,1}^\gamma & 0 & 0 \end{bmatrix}
\]
Equation (A27) is derived from (30). Thus, comparing (A27) with (A25), we get
\[
\xi_{s+1,1} - \mu_4 \gamma_{s+1,1} = \xi_{s,1} I_2 + \mu_3 \theta_{s,1}, I_1.
\]
From (A21), (A24), (A27), and (A28), we finally obtain
\[
\xi_{s+1,1} = \xi_{s,1} I_2 + [\mu_3 \theta_{s,1} + \mu_4 \gamma_{s,1} + \mu_5 \xi_{s,1}] I_1
\]
\[
\mu_3 = -[\tau_1 \tau_2] V_{s,1}^\gamma
\]
\[
\mu_4 = V_{s,1}^\gamma / V_{s,1}^\gamma = \tau_3 + \mu_2 \tau_4 / (V_{s,1}^\gamma + \mu_2 V_{s,1}^\eta)
\]
\[
= \left( V_{s,1}^\gamma V_{s,1}^\gamma - V_{s,1}^\gamma V_{s,1}^\eta \right) / \left( V_{s,1}^\gamma V_{s,1}^\gamma + \mu_2 V_{s,1}^\eta \right)
\]
\[
\mu_5 = \mu_4 \mu_2
\]
\[
\mu_5 = [\tau_1 \tau_2] V_{s,1}^\gamma + \mu_4 V_{s,1}^\eta
\]
\[
= V_{s,1}^\gamma + \mu_4 V_{s,1}^\gamma
\]
\[
= V_{s,1}^\gamma + \mu_4 V_{s,1}^\gamma
\]
From (A18), (A21), and (A29), the AR-type recursive formula is derived.

### APPENDIX IV
**DERIVATION OF (37) AND (38)**

First, let us derive (37). From (24), \(I_1\) and \(I_2\) are rewritten by a new vector \(\vec{e}\) as
\[
I_1^T = [\vec{e}_1^T \cdots \vec{e}_{s+1}^T \cdots \vec{e}_{s+1+t+3}^T]
\]
\[
I_2^T = [\vec{e}_1^T \cdots \vec{e}_{s+1}^T \cdots \vec{e}_{s+1+t+3}^T \ 0^T]
\]
where \(\vec{e}_i\) is the \((s+t+3)\) dimensional row vector which has 1 at the \(i\)th element and zero at the others. Thus, we get
\[
h_{s+1,1}(k) I_1^T = \begin{bmatrix} y(k) & \cdots & y(k-s-1) & x(k) & \cdots & x(k-t) \end{bmatrix} I_1^T
\]
\[
h_{s+1,1}(k) I_2^T = \begin{bmatrix} y(k) & \cdots & y(k-s-1) & x(k) & \cdots & x(k-t) \end{bmatrix} I_2^T
\]
\[
= \begin{bmatrix} y(k-1) & \cdots & y(k-s-1) & x(k-1) & \cdots & x(k-t) & 0 \end{bmatrix}.
\]
From (A33) and (A34), (37) is obtained.

Let us derive (38) next. From (14) and (31), we get
\[
v_{s+1,1}(k) = h_{s+1,1}(k) \vec{e}_1^T
\]
\[
= h_{s+1,1}(k) \begin{bmatrix} I_1^T \vec{e}_1^T + I_2^T \vec{e}_1 \end{bmatrix}
\]
\[
= v_{s,1}(k) + \xi_{s,1}(k-1) \vec{e}_1^T.
\]
From (14) and (31), the other errors \(\gamma_{s+1,1}(k)\) and \(\xi_{s+1,1}(k)\) are given as follows:
\[
\gamma_{s+1,1}(k) = -h_{s+1,1}(k) \gamma_{s+1,1}
\]
\[
= h_{s+1,1}(k) I_1^T \left[ \gamma_{s+1,1} + \mu_2 \xi_{s+1,1} \right]
\]
\[
= \gamma_{s,1}(k) - \mu_2 \xi_{s,1}(k)
\]
\[
\xi_{s+1,1}(k) = h_{s+1,1}(k) \vec{e}_2^T
\]
\[
= h_{s+1,1}(k) I_2^T \vec{e}_1^T + h_{s+1,1}(k) I_1^T
\]
\[
= \xi_{s,1}(k-1) + v_{s,1}(k) \mu_2 \xi_{s,1}(k) + \mu_3 \xi_{s,1}(k)
\]
\[
= \xi_{s,1}(k-1) + v_{s,1}(k) \mu_2 \xi_{s,1}(k) + \mu_3 \xi_{s,1}(k)
\]
where (A37) uses (A36) and $\mu_3 = \mu_4 \mu_2$. From (A35)-(A37), (38) is obtained.

ACKNOWLEDGMENT

The authors express their appreciation to Prof. K. Hatori of Hokkaido University, Sapporo, for his useful advice.

REFERENCES