US 20040059551A1

(19) **United States**

(12) **Patent Application Publication** (10) **Pub. No.: US 2004/0059551 A1**

Nishiyama (43) **Pub. Date: Mar. 25, 2004**

(54) **SYSTEM IDENTIFYING METHOD**

(76) Inventor: **Kiyoshi Nishiyama**, Iwate (JP)

Correspondence Address:
Thomas W Tolpin
Welsh & Katz
22nd Floor
120 S Riverside Plaza
Chicago, IL 60606 (US)

(21) Appl. No.: **10/415,674**

(22) PCT Filed: **Oct. 16, 2001**

(86) PCT No.: **PCT/JP01/09082**

(30) **Foreign Application Priority Data**

Oct. 24, 2000 (JP) 2000-323958

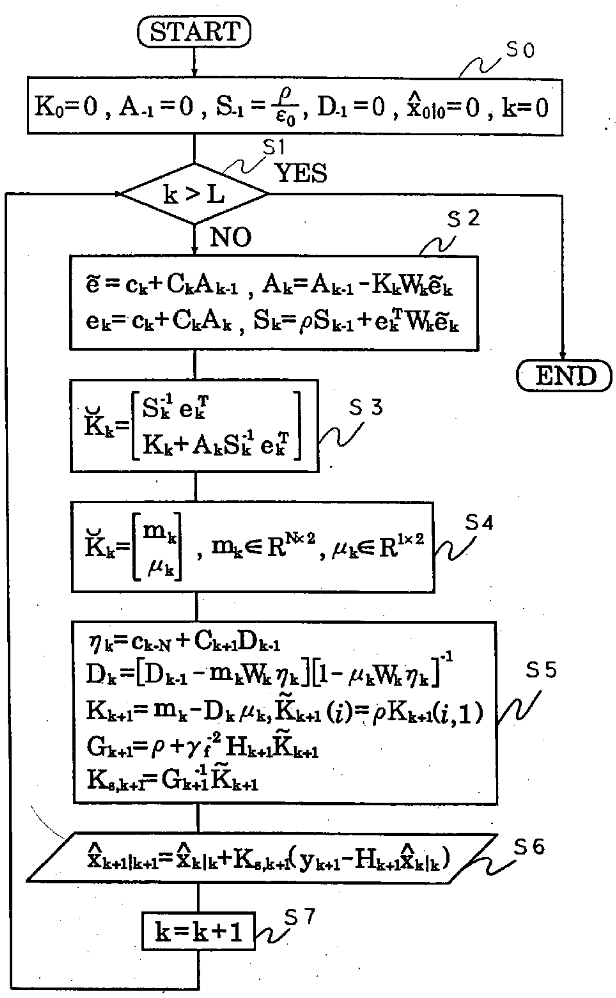
Publication Classification

(51) **Int. Cl.**[7] **G06F 17/10**; G06F 7/60

(52) **U.S. Cl.** **703/2**

(57) **ABSTRACT**

Quick real-time identification and estimation of a time-non-varying or time-varying system. A new H_∞ evaluation criterion is determined, a fast algorithm for a modified H_∞ filter based on the criterion is developed, and a quick time-varying system identifying method according to the fast H_∞ filtering algorithm is provided. By the fast H_∞ filtering algorithm, a time-varying system sharply varying can be traced with an amount of calculation O(N) per unit time step. The algorithm completely agrees with a fast Kalman filtering algorithm at the extreme of the upper limit value. If the estimate of impulse response is determined, a pseudo-echo is sequentially determined from the estimate and subtracted from the real echo to cancel the echo. Thus an echo chancellor is realized.



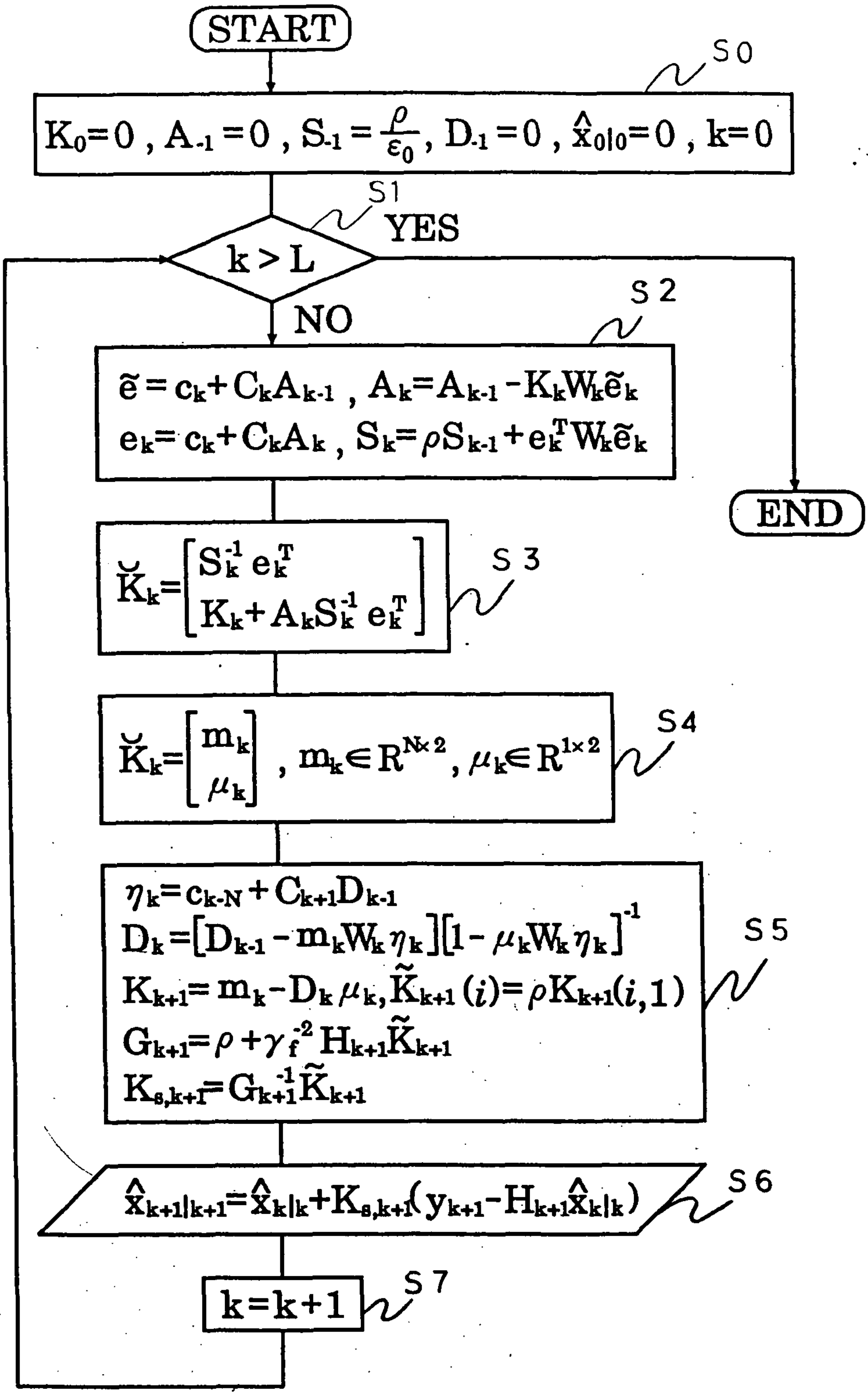
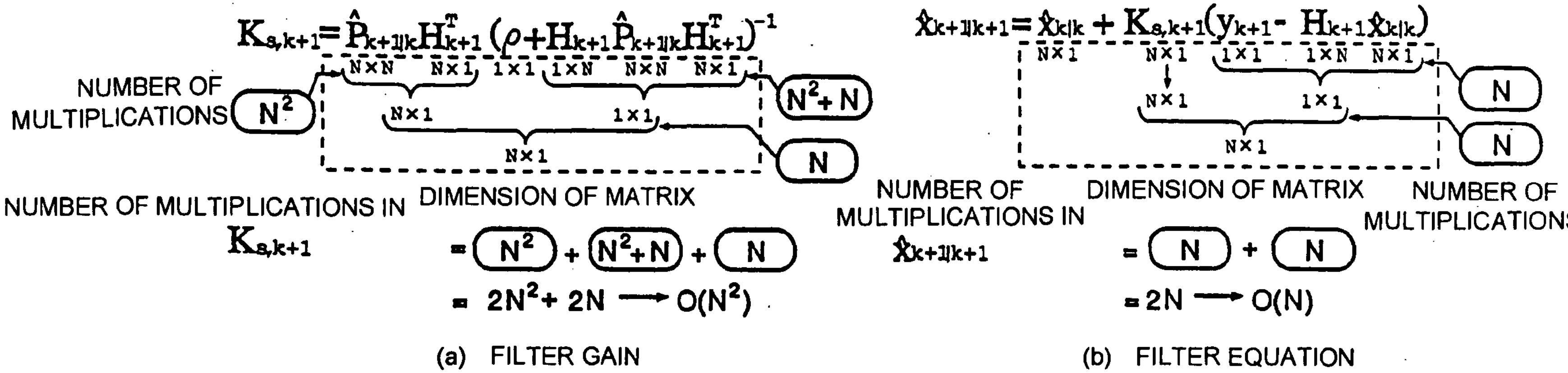
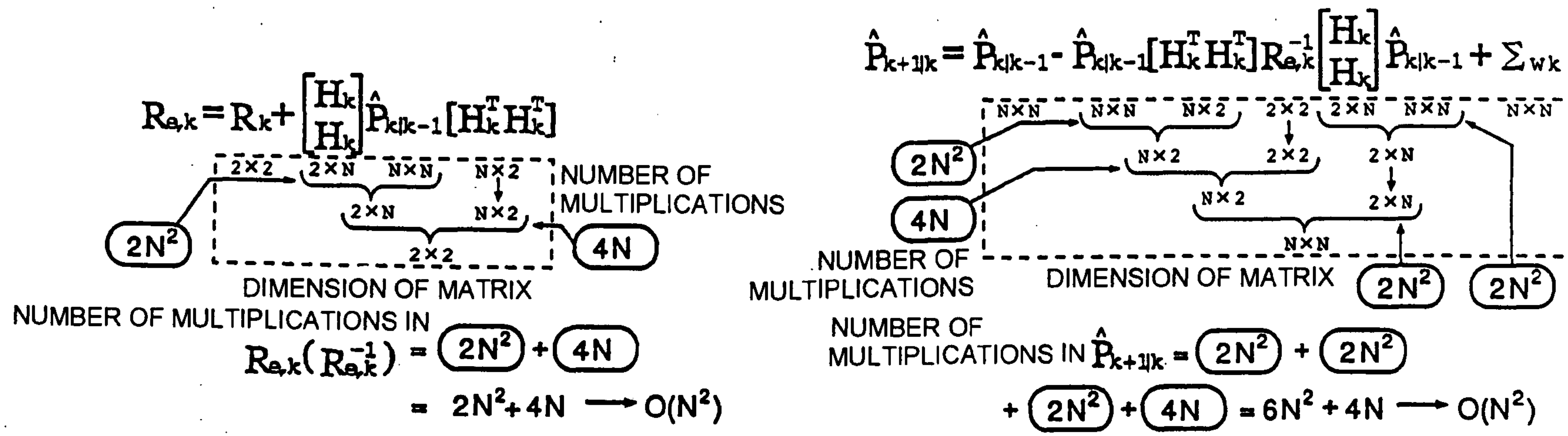

START
S0
K0=0 , A-1=0 , S-1=ρ/ε0 , D-1=0 , x̂0|0=0 , k=0
S1
k > L
YES
NO
END
S2
ẽ=ck+CkAk-1 , Ak=Ak-1-KkWkẽk
ek=ck+CkAk , Sk=ρSk-1+ekTWkẽk
S3
S4
S5
S6
x̂k+1|k+1=x̂k|k+Ks,k+1(yk+1-Hk+1x̂k|k)
S7
k=k+1

FIG. 1



$K_{s,k+1} = \hat{P}_{k+1|k} H^T_{k+1} (\rho + H_{k+1} \hat{P}_{k+1|k} H^T_{k+1})^{-1}$
N×N N×1 1×1 1×N N×N N×1
N×1 1×1
N×1
NUMBER OF MULTIPLICATIONS
N2
N2+ N
N
NUMBER OF MULTIPLICATIONS IN $K_{s,k+1}$
DIMENSION OF MATRIX
= N2 + N2+N + N
= 2N2+ 2N → O(N2)
(a) FILTER GAIN
$\hat{x}_{k+1|k+1} = \hat{x}_{k|k} + K_{s,k+1}(y_{k+1} - H_{k+1}\hat{x}_{k|k})$
N×1 N×1 1×1 1×N N×1
N×1 1×1
N×1
N
N
NUMBER OF MULTIPLICATIONS IN $\hat{x}_{k+1|k+1}$
DIMENSION OF MATRIX
NUMBER OF MULTIPLICATIONS
= N + N
= 2N → O(N)
(b) FILTER EQUATION

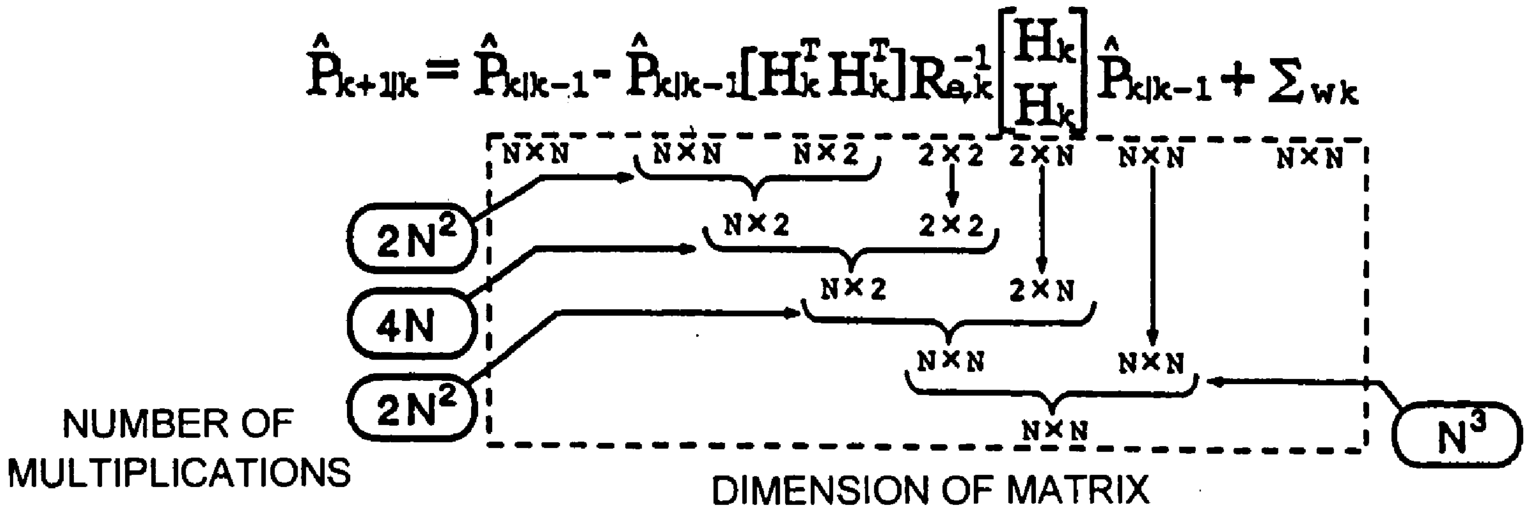
FIG. 2



(a) EQUATION OF $R_{e,k}$

(b) RICCATI EQUATION

FIG.3

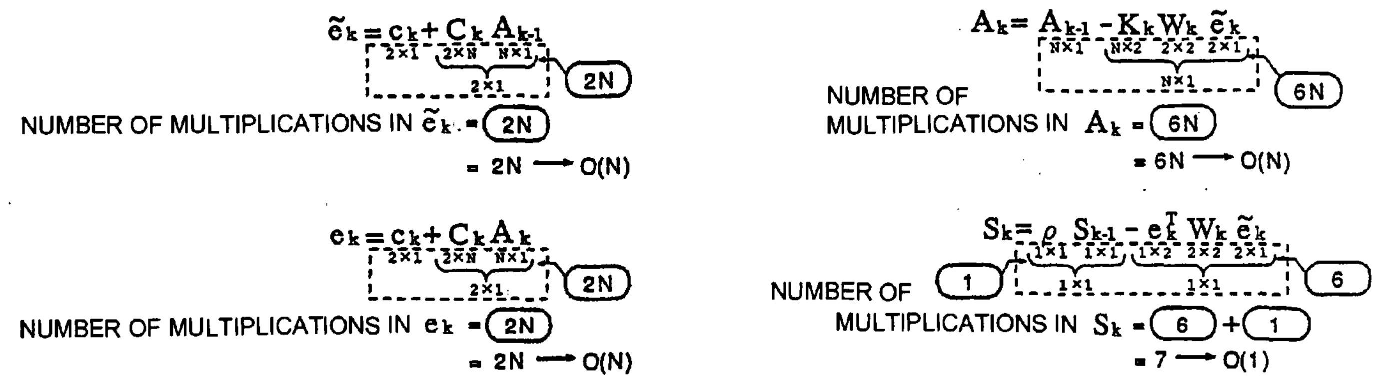


NUMBER OF MULTIPLICATIONS IN

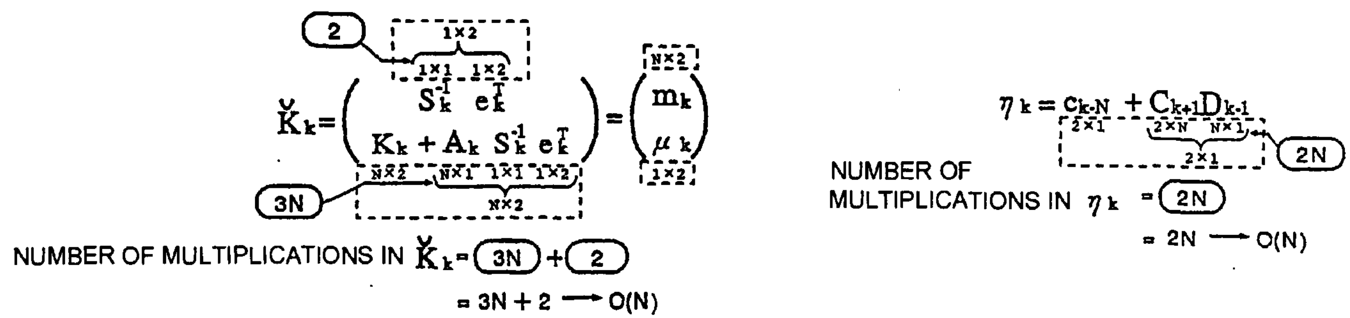
$$\hat{P}_{k+1|k} = (2N^2) + (4N) + (2N^2) + (N^3)$$

$$= N^3 + 4N^2 + 4N \longrightarrow O(N^3)$$

FIG.4

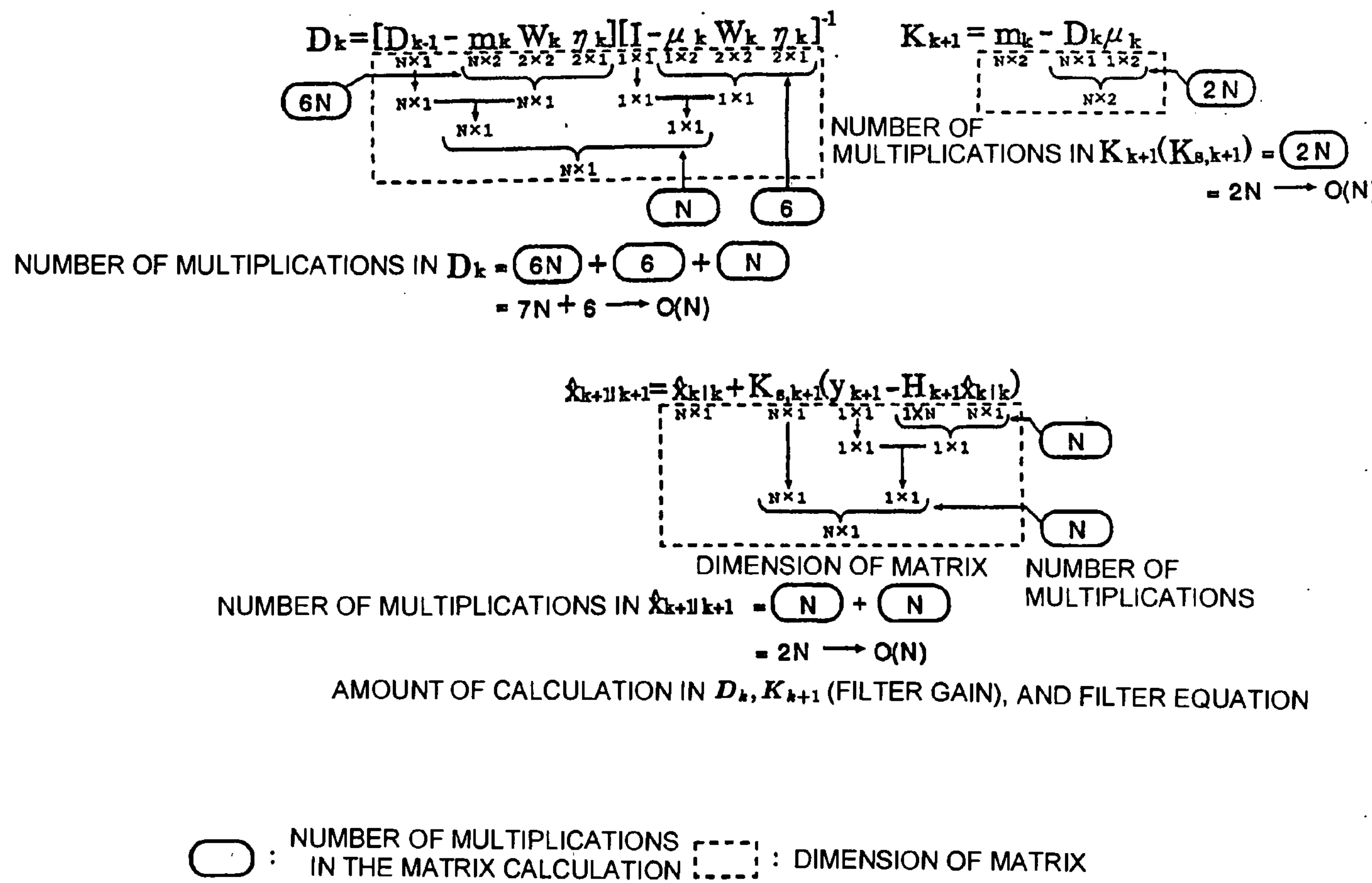


(a) AMOUNT OF CALCULATION IN $\tilde{e}_k$, A_k, e_k, AND S_k



(b) AMOUNT OF CALCULATION IN $\check{K}_k$, AND η_k

FIG.5



6N
N
6
2N
NUMBER OF
MULTIPLICATIONS IN $K_{k+1}(K_{s,k+1})$ = 2N
= 2N → O(N)
NUMBER OF MULTIPLICATIONS IN D_k = 6N + 6 + N
= 7N + 6 → O(N)
N
N
DIMENSION OF MATRIX
NUMBER OF
MULTIPLICATIONS
NUMBER OF MULTIPLICATIONS IN $\hat{x}_{k+1|k+1}$ = N + N
= 2N → O(N)
AMOUNT OF CALCULATION IN D_k, K_{k+1} (FILTER GAIN), AND FILTER EQUATION
: NUMBER OF MULTIPLICATIONS
IN THE MATRIX CALCULATION
: DIMENSION OF MATRIX

FIG.6

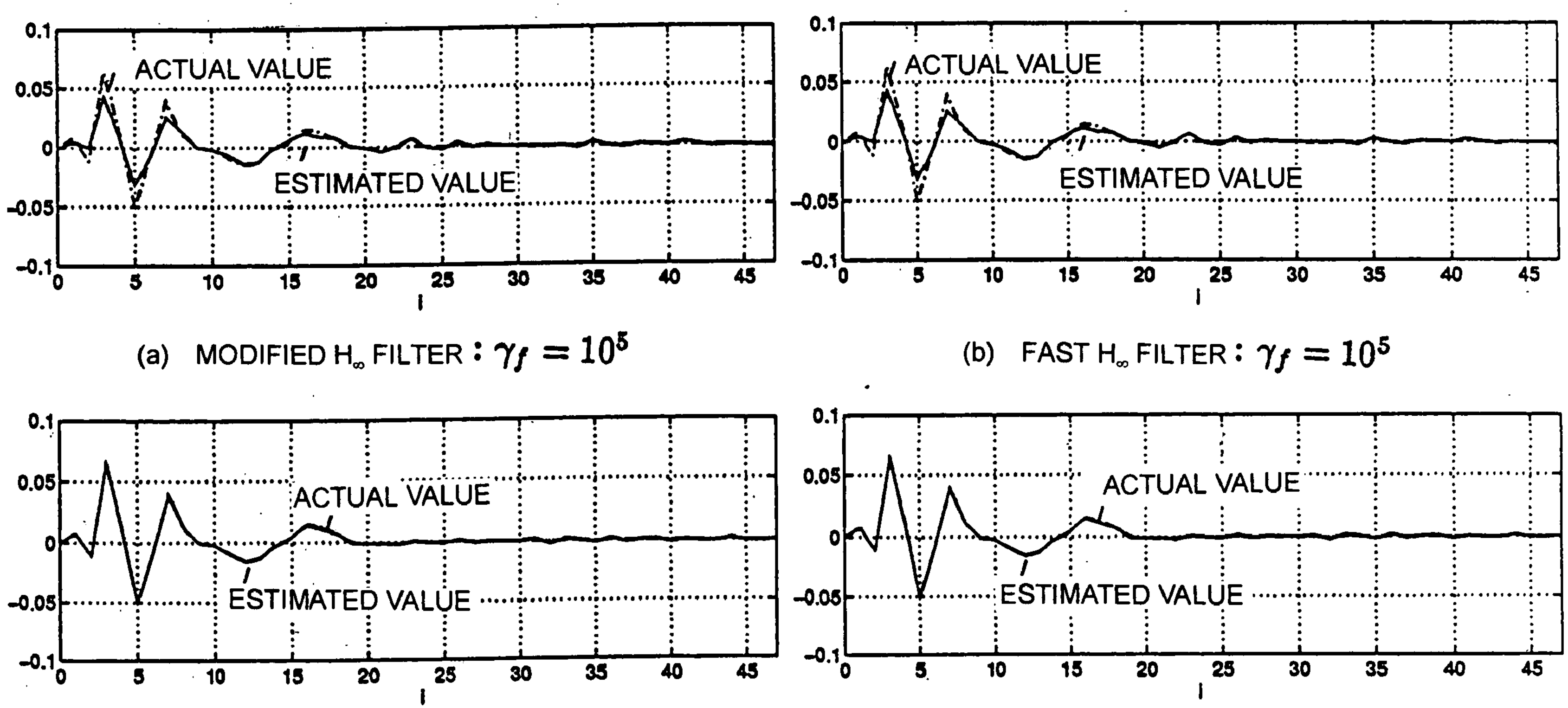
h_0	h_1	h_2	h_3	h_4	h_5
0.0	0.008	-0.012	0.064	0.013	-0.052

h_6	h_7	h_8	h_9	h_{10}	h_{11}
-0.007	0.039	0.011	0.0	-0.002	-0.009

h_{12}	h_{13}	h_{14}	h_{15}	h_{16}	h_{17}
-0.016	-0.013	-0.001	0.004	0.015	0.013

h_{18}	h_{19}	h_{20}	h_{21}	h_{22}	h_{23}
0.007	0.0	-0.001	-0.002	-0.001	0.0

FIG.7



(a) MODIFIED H_∞ FILTER : $\gamma_f = 10^5$

(b) FAST H_∞ FILTER : $\gamma_f = 10^5$

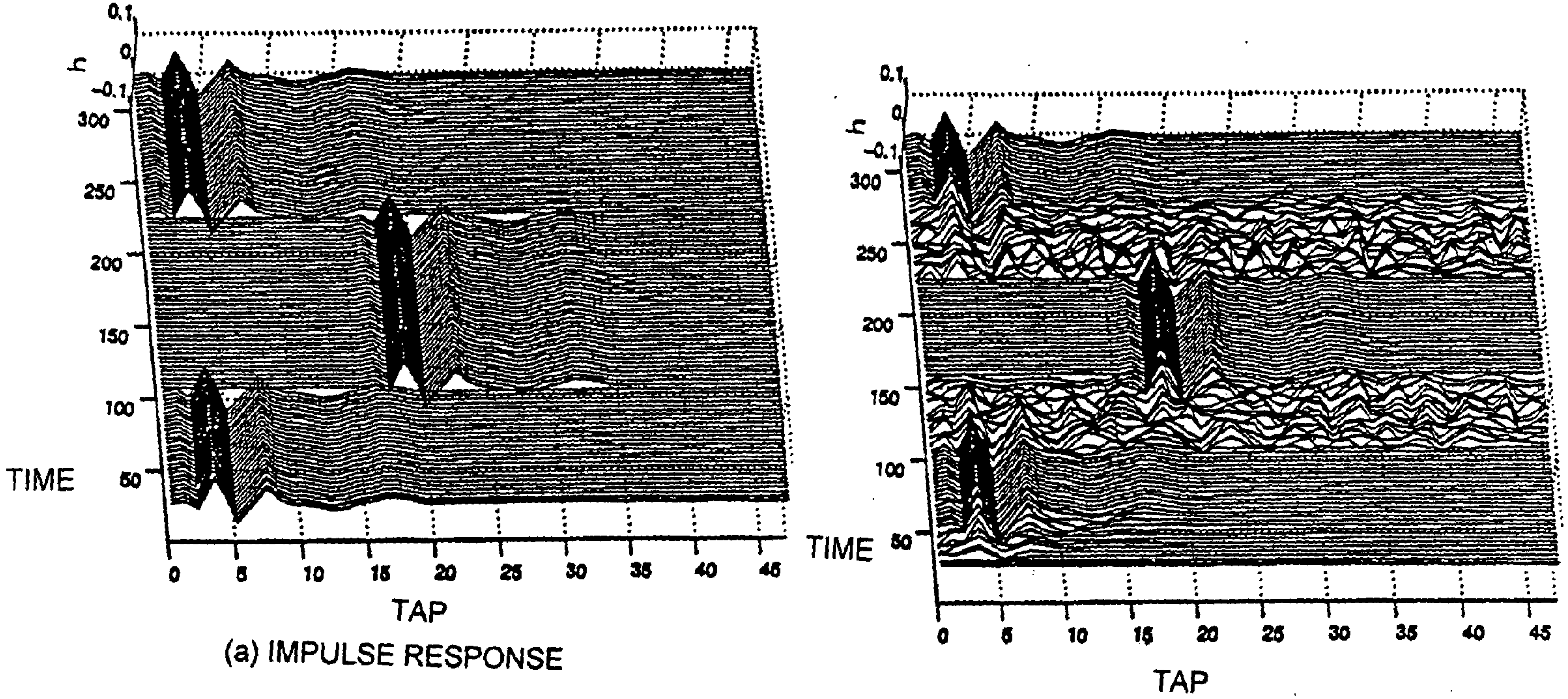
(c) MODIFIED H_∞ FILTER : $\gamma_f = 2.0$

(d) FAST H_∞ FILTER : $\gamma_f = 2.0$

FIG.8

TAP NUMBER	MODIFIED H_∞ FILTER (1)	MODIFIED H_∞ FILTER (2)	FAST H_∞ FILTER
24	1.76	1.37	1.95
48	6.66	2.77	2.92
96	49.9	8.56	4.76
192	419.1	32.5	8.61
384	3.41×10^3	126.6	16.3

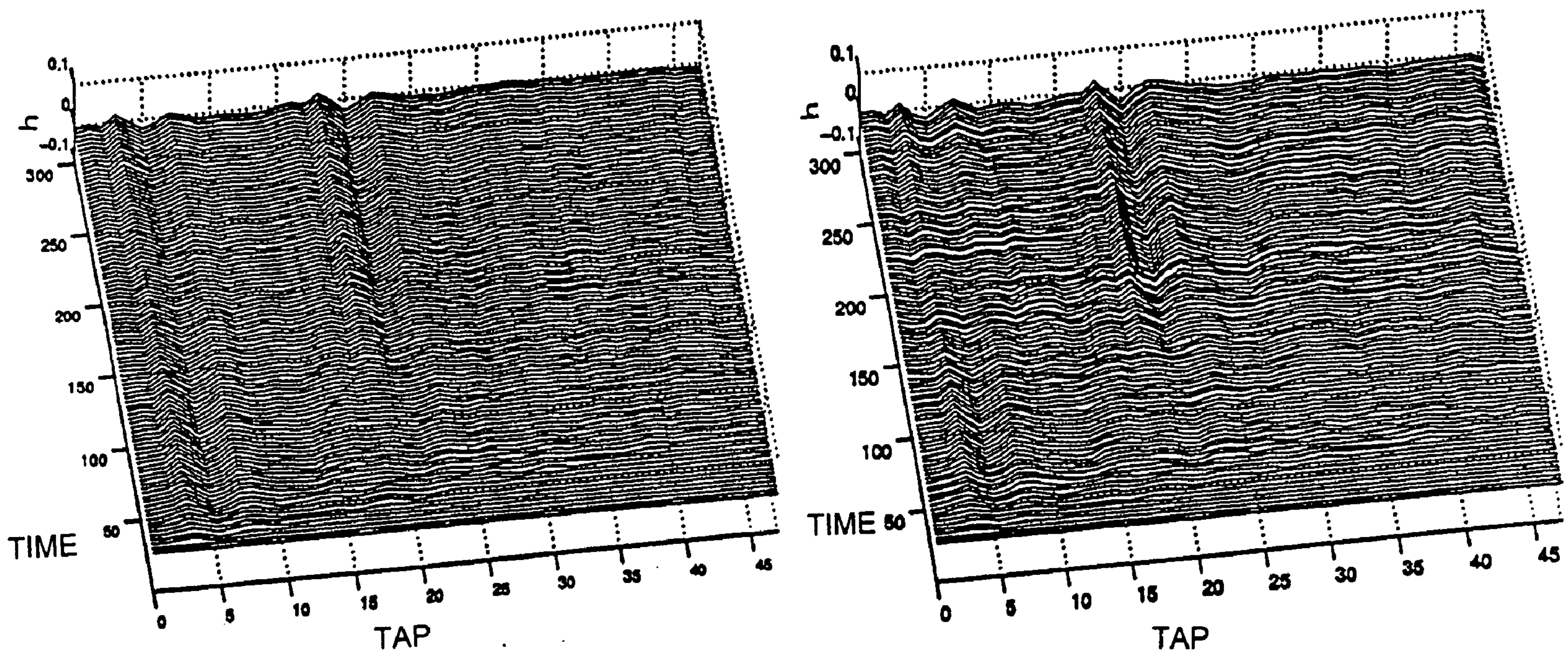
FIG.9



(a) IMPULSE RESPONSE

(b) ESTIMATE BY FAST HF : $\gamma_f = 2.0$

FIG.10



(a) ESTIMATE BY FAST KF

(b) ESTIMATE BY LMS : $\mu = 5.0$

FIG.11

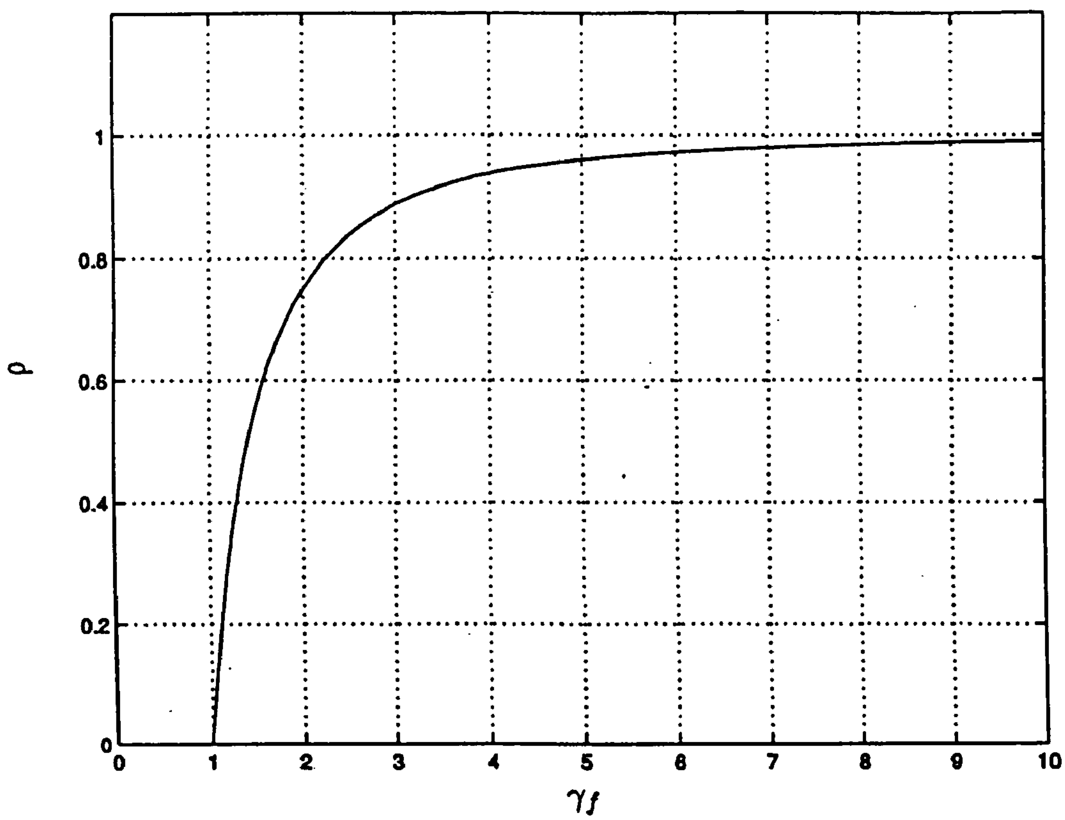
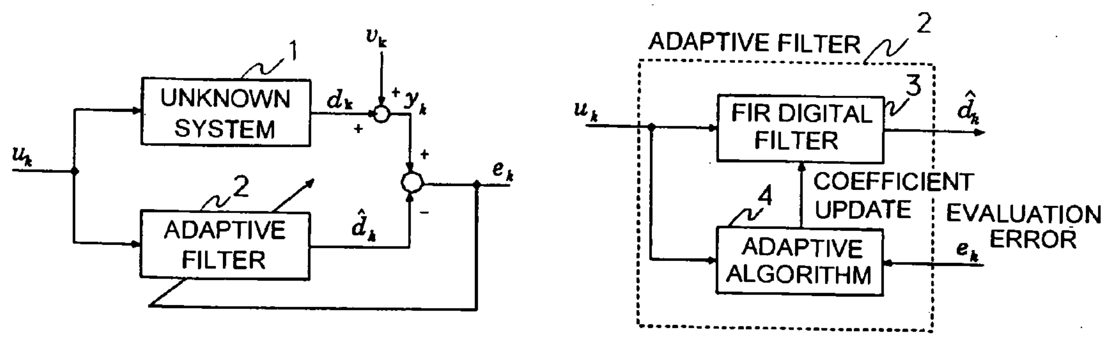


FIG.12

TAP NUMBER	FAST HF [s]	FAST KF [s]	LMS [s]
48	8.82	6.00	2.18
96	14.9	10.3	3.75
192	27.3	19.2	6.96
384	51.7	35.8	13.4

FIG.13



(a) OUTPUT ERROR METHOD

(b) STRUCTURE OF ADAPTIVE FILTER

FIG.14

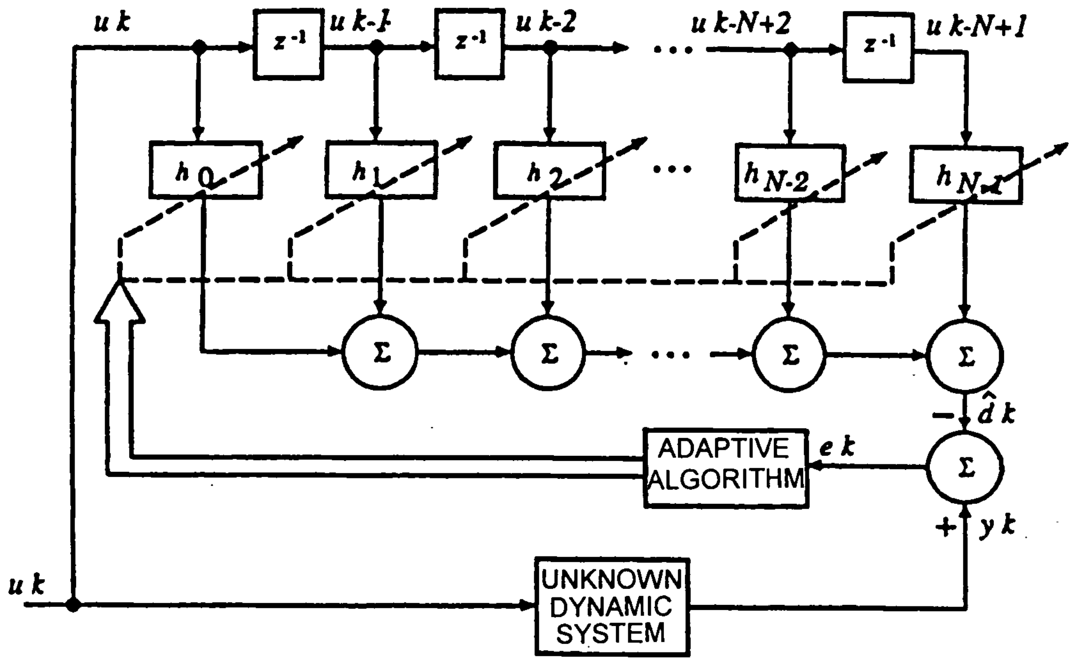

u k
z⁻¹
u k-1
z⁻¹
u k-2
u k-N+2
z⁻¹
u k-N+1
h 0
h 1
h 2
h N-2
h N-1
Σ
Σ
Σ
Σ
−
d̂ k
ADAPTIVE ALGORITHM
e k
Σ
+
y k
u k
UNKNOWN DYNAMIC SYSTEM

FIG.15

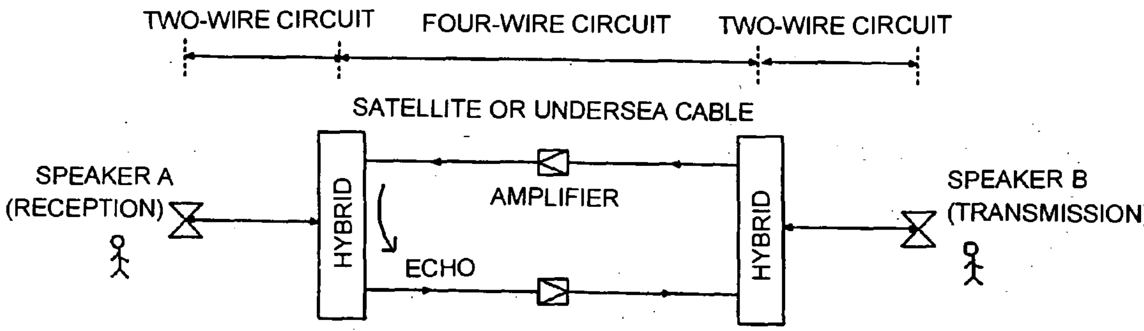

TWO-WIRE CIRCUIT
FOUR-WIRE CIRCUIT
TWO-WIRE CIRCUIT
SATELLITE OR UNDERSEA CABLE
SPEAKER A
(RECEPTION)
HYBRID
AMPLIFIER
ECHO
HYBRID
SPEAKER B
(TRANSMISSION)

FIG.16

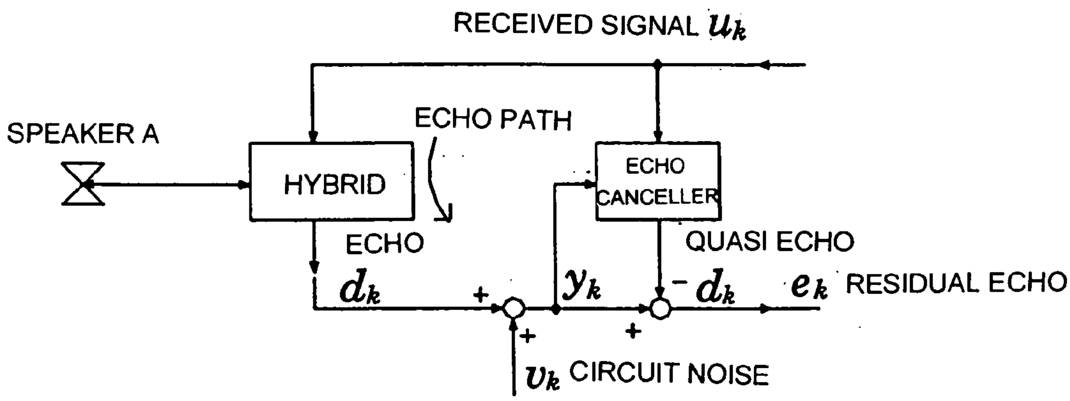

RECEIVED SIGNAL u_k
SPEAKER A
ECHO PATH
HYBRID
ECHO
CANCELLER
ECHO
QUASI ECHO
d_k
+
y_k
$-\hat{d}_k$
e_k RESIDUAL ECHO
+
+
v_k CIRCUIT NOISE

FIG.17

SYSTEM IDENTIFYING METHOD

BACKGROUND OF THE INVENTION

[0001] The present invention relates to system identification methods, and more particularly, to a time-varying system identification method at a high speed in real time by using a fast algorithm for modified H_∞ filters developed based on a new H_∞ evaluation criterion.

[0002] In general, system identification is to estimate a mathematical model (a transfer function, an impulse response, or the like) of a system input-and-output relationship according to the input-and-output data. Typical applications thereof include echo cancellers in international communications, automatic equalizers in data communications, echo cancellers in acoustic systems, sound-field reproduction, and active noise control in vehicles. Details are written in "Digital Signal Processing Handbook", the Institute of Electronics, Information and Communication Engineers, 1993, or the like.

[0003] (Basic Principle)

[0004] **FIG. 14** shows a configuration for system identification. This system includes an unknown system **1** and an adaptive filter **2**. The adaptive filter **2** has an FIR digital filter **3** and an adaptive algorithm **4**.

[0005] One case which uses an output error method to identify the unknown system **1** will be described below. Here, u_k indicates an input to the unknown system **1**, d_k indicates the output of the system, which is a signal to be obtained, and $d\hat{}_k$ indicates the output of the filter. (A mark "ˆ" means an estimated value and should be placed above characters, but it is placed at the upper right of the characters for input convenience. This notation may be used through the present specification.)

[0006] Since an impulse response is generally used as a parameter of an unknown system, the adaptive filter adjusts a coefficient of the FIR digital filter **3** by the adaptive algorithm such that an evaluation error $e_k=d_k-d\hat{}_k$ in the figure is minimized.

[0007] **FIG. 15** shows the structure of an impulse-response adjustment mechanism.

[0008] Here, as an example of adaptive algorithm, the following LMS algorithm (least mean square algorithm) is widely used because of its computational simplicity.

[0009] [LMS Algorithm]

$$\hat{h}_{k+1}=\hat{h}_k+\mu u_k(y_k-u_k^T\hat{h}_k) \quad (1)$$

[0010] where,

$$\hat{h}_k=[\hat{h}_0[k], \ldots, \hat{h}_{N-1}[k]]^T,\ u_k=[u_k, \ldots, u_{k-N+1}]^T,\ \mu>0 \quad (2)$$

[0011] Generally, Kalman filters, which converges quickly, are suitable for identifying a time-varying system.

[0012] [Kalman Filter]

$$\hat{x}_{k|k}=\hat{x}_{k|k-1}+K_k(y_k-H_k\hat{x}_{k|k-1})$$

$$\hat{x}_{k+1|k}=\hat{x}_{k|k} \quad (3)$$

$$K_k=\hat{P}_{k|k-1}H_k^T(1H_k\hat{P}_{k|k-1}H_k^T)^{-1} \quad (4)$$

$$\hat{P}_{k|k}=\hat{P}_{k|k-1}-K_kH_k\hat{P}_{k|k-1}$$

[0013]

$$\hat{P}_{k+1/k}=\hat{P}_{k/k}+\frac{\sigma_w^2}{\sigma_v^2}I \quad (5)$$

[0014] where,

$$\hat{x}_{k|k}=[\hat{h}_0[k], \ldots, \hat{h}_{N-1}[k]]^T,\ H_k=[u_{k-1}, \ldots, u_{k-N}]$$

$$\hat{x}_{0|1}=0,\ \hat{P}_{0|-1}=\epsilon_0 I,\ \epsilon_0>0 \quad (6)$$

[0015] Here, the impulse response $\{h_i\}$ of unknown system is obtained as a state estimate $x\hat{}_{k|k}$, and an input $\{u_k\}$ to the system is used as an element of an observation matrix H_k.

[0016] Fast Kalman filtering algorithm is also known, which reduces the amount of calculation per unit time step to the number of times calculations are performed proportional to N, that is, O(N) by applying the shift property ($H_{k+1}(i+1)=H_k(i)$) of the observation matrix H_k to a Kalman filter obtained when $\sigma^2_w=0$. Details are written in "Digital Signal Processing Handbook", the Institute of Electronics, Information and Communication Engineers, 1993, or the like.

[0017] (Applications to Echo Cancellers)

[0018] Four-wire circuits are used for long distance telephone lines such as for international calls for a reason of signal amplification and others. On the other hand, two-wire circuits are used for subscriber lines because they are relatively short.

[0019] **FIG. 16** is an explanation view of a communication system and an echo. Impedance matching is performed by disposing hybrid transformers at connection points between two-wire circuits and a four-wire circuit, as shown in the figure. If the impedance matching is perfect, a signal (voice) from a speaker B reaches only a speaker A. However, it is difficult to make the matching perfect in general. A phenomenon occurs in which a part of a received signal leaks to the four-wire circuit, is amplified, and returns to the receiver (speaker A). This is an echo. As the transmission distance gets longer (delay time gets longer), the effect of the echo gets larger, and the quality of telephone speech significantly deteriorates (in pulse transmission, an echo influences significantly even in a short distance, and the quality of telephone speech deteriorates).

[0020] **FIG. 17** shows a basic principle of an echo canceller.

[0021] As shown in the figure, the echo canceller is introduced to successively estimate the impulse response of an echo path by using a received signal and an echo directly observable, and to subtract a quasi echo obtained by using the estimate from an actual echo to cancel and remove the echo.

[0022] The impulse response of the echo path is estimated such that the mean square error of a residual echo e_k is minimized. Elements which disturb the estimation of the echo path are line noise and a signal (voice) from the speaker A. When two speakers start talking at the same time (double talking), the estimation of the impulse response is generally halted. Since the impulse response of the hybrid transformers has a length of approximately 50 [ms], if the sampling period is set to 125 [μs], the order of the impulse response of the echo path actually becomes approximately 400.

SUMMARY OF THE INVENTION

[0023] In conventional arts, the LMS algorithm (least-mean-square algorithm) has been widely used as adaptive algorithm due to its simplicity, but it is impossible to closely track a time-varying system which varies suddenly due to its very slow convergence.

[0024] A Kalman filter, which has an excellent tracking performance, requires the amount of calculation proportional to $O(N^2)$ or $O(N^3)$. Since the amount of calculation rapidly increases with the tap number N, it is difficult to process actual issues requiring a high tap number N in real time. As a countermeasure therefor, a fast Kalman filter with the computational complexity of O(N) per unit time step for a tap number N has been proposed, but it is impossible for the filter to identify a time-varying system because of its stationary characteristic (incapable of taking system noise into account).

[0025] In view of the above-described points, it is an object of the present invention to implement a fast real-time identification of time-varying and time-invariant systems by using a fast algorithm for modified H_∞ filters developed based on a new H_∞ evaluation criterion. It is another object of the present invention to include, as a particular case of the present algorithm, a fast Kalman filtering algorithm, and to determine theoretically the covariance of system noise which is dominant in the tracking performance of time-varying systems. It is still another object of the present invention to provide a fast time-varying system identification method which is substantially effective even when an input signal is discontinuously varied, such as in an echo canceller for a time-varying system which varies extremely as sudden line switching. It is a further object of the present invention to provide a system identification method which is applicable to echo cancellers in communication systems and acoustic systems, sound-field reproduction, and noise control.

[0026] In order to solve the problems described above, a new H_∞ evaluation criterion is introduced, a fast algorithm for the modified H_∞ filters is developed according to the criterion, and a fast time-varying system identification method based on the fast algorithm is proposed in the present invention. The fast algorithm according to the present invention is capable of tracking a time-varying system which varies rapidly, with the complexity of O(N) per unit time step. Further, it has a convenient property that it completely matches the fast Kalman filtering algorithm at a limit of $\gamma_f=\infty$.

BRIEF DESCRIPTION OF THE DRAWINGS

[0027] **FIG. 1** is a flowchart of a fast algorithm.

[0028] **FIG. 2** is an explanation view (1) of the amount of calculation in each part of a modified H_∞ filtering algorithm.

[0029] **FIG. 3** is an explanation view (2) of the amount of calculation in each part of the modified H_∞ filtering algorithm.

[0030] **FIG. 4** is an explanation view of the amount of calculation when the order of matrix calculation is changed.

[0031] **FIG. 5** is an explanation view (1) of the amount of calculation in a fast H_∞ filtering algorithm.

[0032] **FIG. 6** is an explanation view (2) of the amount of calculation in the fast H_∞ filtering algorithm.

[0033] **FIG. 7** is a view showing values of an impulse response $\{h_i\}$ to be estimated.

[0034] **FIG. 8** is a comparative explanation view of estimated results of impulse responses obtained by the modified H_∞ filter and the fast H_∞ filter.

[0035] **FIG. 9** is a view of measurement results of computation time.

[0036] **FIG. 10** is a view (1) of simulation results of each algorithm.

[0037] **FIG. 11** is a view (2) of simulation results of each algorithm.

[0038] **FIG. 12** is a view showing the relationship between γ, and ρ.

[0039] **FIG. 13** is a view showing the relationship among tap numbers and computation times of impulse responses obtained by the fast H_∞ filter, fast Kalman filter, and LMS algorithm.

[0040] **FIG. 14** is a view showing a configuration for system identification.

[0041] **FIG. 15** is a view showing the configuration of an impulse-response adjustment mechanism.

[0042] **FIG. 16** is an explanation view of a communication system and an echo.

[0043] **FIG. 17** is a view showing the principle of an echo canceller.

DESCRIPTION OF THE PREFERRED EMBODIMENTS

[0044] Embodiments of the present invention will be described hereinafter. Details are shown, for example, in "Derivation of A Fast Algorithm of Modified H_∞ Filters", K. Nishiyama, IEEE international Conference on Industrial Electronics, Control and Instrumentation, RBC-II, pp.462-467, October, 2000.

[0045] 1. Description of Symbols

[0046] First, main symbols used in the embodiments of the present invention and whether they are known or unknown will be described.

[0047] x_k: State vector or just state, unknown and to be estimated.

[0048] x_0: Initial state, unknown.

[0049] w_k: System noise, unknown.

[0050] v_k: Observation noise, unknown.

[0051] y_k: Observation signal, known and input to a filter.

[0052] z_k: Output signal, unknown.

[0053] H_k: Observation matrix, known.

[0054] L_k: Output matrix, known.

[0055] $\hat{x}_{k|k}$: State value of the state x_k at time k, estimated by using observation signals y_0-y_k. Given by a filter equation.

[0056] $\hat{x}_{0|0}$: Initial estimate of a state, essentially unknown but set to 0 for convenience.

[0057] $K_{s,\,k+1}$: Filter gain, obtained by matrix $\hat{P}_{k|k-1}$.

[0058] Σ_{wk}: Corresponds to the covariance matrix of the system noise, known in theory but unknown in practice.

[0059] $\hat{P}_{k|k-1}$: Corresponds to the covariance matrix of the error of $\hat{x}_{k|k-1}$, given by a Riccati equation.

[0060] $\hat{P}_{1|0}$: Corresponds to the covariance matrix of an error in the initial state, essentially unknown but set to $\epsilon_0 I$ for convenience.

[0061] σ^2_v: Variance of the observation noise, treated as known in theory but unknown in practice.

[0062] σ^2_w: Variance of the system noise, treated as known in theory but unknown in practice.

[0063] A mark "ˆ" placed above a symbol indicates an estimated value, a mark "U" indicates that the matrix is extended by one row, and a mark "" is added for convenience. These marks are placed at the upper right of characters for input convenience, but, as shown in expressions, they are identical with those placed above characters. "L", "H", "P", and "K" indicate matrixes. Some of them are written in bold face as in expressions, but they are usually written in lightface for convenience.

[0064] 2. Modified H_∞ Filter

[0065] Next, a state-space model as in the following equations (7) to (9) is discussed.

$$x_{k+1}=x_k+w_k,\ w_k, x_k \epsilon R^N \tag{7}$$

$$y_k=H_k x_k+v_k,\ y_k,\ v_k \epsilon R \tag{8}$$

$$z_k=H_k x_k,\ z_k \epsilon R,\ H_k \epsilon R^{1\times N} \tag{9}$$

[0066] where, $L_k=H_k$ ($H_k=[u_k\ u_{k-}2 \ldots u_{k-N+1}]$) assuming an echo canceller or the like. For such a state-space model, an H_∞ evaluation criterion (γ_f is newly placed in the left-hand side) such as that shown by expression (10) is proposed.

$$\sup_{x_0,\{w_i\},\{v_i\}} \frac{\sum_{i=0}^{k} \|e_{f,i}\|^2/\rho}{\|x_0-\check{x}_{0|-1}\|^2_{\Sigma_0^{-1}} + \sum_{i=0}^{k} \|w_i\|^2_{\Sigma_{w_k}} + \sum_{i=0}^{k} \|v_i\|^2/\rho} < \gamma_f^2 \tag{10}$$

[0067] When it is assumed that ρ or Σ_{wk} does not depend on γ_f, a modified H_∞ filter of level γ_f satisfying the evaluation criterion can be given by the following equations (11) to (14) by applying a standard H_∞ estimation scheme known in the system identification field. This scheme is shown, for example, in "Linear Estimation in Krein Spaces—Part I: Theory," B. Hassibi, A. H. Sayed, and T. Kailath, IEEE Trans. Automatic Control, 41, 1, pp.18-33, 1996., and "Linear Estimation in Krein Spaces—Part II: Applications," B. Hassibi, A. H. Sayed, and T. Kailath, IEEE Trans. Automatic Control, 41, 1, pp.34-49, 1996.

$$\bar{z}_{k|k} = H_k x_{k|k} \tag{11}$$

$$\hat{x}_{k+1|k+1} = \hat{x}_{k|k} + K_{a,k+1}(y_{k+1} - H_{k+1}\hat{x}_{k|k}) \qquad \text{Filter equation} \tag{12}$$

$$K_{a,k+1} = \hat{P}_{k+1|k} H_{k+1}^T \left(H_{k+1}\hat{P}_{k+1|k} H_{k+1}^T + \rho\right)^{-1} \qquad \text{Filter gain} \tag{13}$$

$$\hat{P}_{k+1|k} = \hat{P}_{k|k-1} - \hat{P}_{k|k-1}[H_k^T H_k^T] R_{e,k}^{-1} \begin{bmatrix} H_k \\ H_k \end{bmatrix} \hat{P}_{k|k-1} + \sum\nolimits_{w_k} \qquad \textit{Riccati equation} \tag{14}$$

where,

$$e_{j,i} = z_{i|i} - H_i x_i \tag{15}$$

$$R_{e,k} = R_k + \begin{bmatrix} H_k \\ H_k \end{bmatrix} P_{k|k-1}[H_k^T H_k^T]$$

$$R_k = \begin{bmatrix} \rho & 0 \\ 0 & -\rho\gamma_j^2 \end{bmatrix},\ \Sigma_{w_k} = \gamma_f^{-2} P_{k+1|k}$$

$$P_{k|k-1}^{-1} + H_k^T H_k < 0,\ P_{1|0} = \varepsilon_0 I,\ \varepsilon_0 > 0$$

$$0 < \rho = 1 - \gamma_f^{-2} \le 1,\ \gamma_f > 1$$

[0068] Since a weight ρ in the evaluation criterion depends on an upper limit γ_f determined in advance, the above algorithm is essentially different from that applied to normal H_∞ filters. The present algorithm controls a maximum energy gain from disturbances (having the initial state x_0, the system noise $\{w_i\}$, and the observation noise $\{v_i\}$) weighted by ρ to a filter error $\{e_{f,i}\}$ so as to be smaller than γ_f^2. Therefore, the present algorithm is a robust filtering algorithm against the disturbances. This property is reflected by the tracking characteristic of a time-varying system. When $\gamma_f \rightarrow \infty$ is satisfied, ρ=1 and Σ_{wk}=0. In this time, the modified H_∞ filter becomes a normal H_∞ filter.

[0069] The main load for calculating the modified H_∞ filter rises during the update of $\hat{P}_{k+1|k} \epsilon R^{N\times N}$, which requires the amount of calculation in proportion to N^2 or N^3. That is, an arithmetic operation of $O(N^2)$ per unit time step is required. Here, a tap number N matches the dimension of the state vector x_k. Therefore, as the dimension of x_k increases, the computation time required to perform the modified H_∞ filter increases rapidly. In order to overcome the drawback, the introduction of a fast algorithm of the modified H_∞ filter is needed.

[0070] 3. Fast H_∞ Filtering Algorithm

[0071] The calculation of the Riccati equation (covariance equation of a state estimation error) shown in equation (14) is dominant in the computational complexity of the modified H_∞ filter. Therefore, to process the modified H_∞ filter at a high speed, if the filter gain of equation (13) is directly determined without using the Riccati equation, the computational burden can be significantly reduced.

[0072] Since it is difficult to derive a fast algorithm for directly obtaining a filter gain $K_{s,k} \epsilon R^{N\times1}$, however, evolving an algorithm for fast calculating a gain matrix defined as follows is examined.

$$K_k = P_k C_k^T \epsilon R^{N\times2} \tag{16}$$

[0073] where,

$$P_k = [\mathcal{O}_k^T \Omega_k \mathcal{O}_k]^{-1} = \left[\sum_{i=1}^{k} \rho^{k-i} C_i^T W_i C_i\right]^{-1} \tag{17}$$

$$\Omega_k = \left[\begin{array}{c|c} \rho\Omega_{k-1} & 0 \\ \hline 0 & W_k \end{array}\right], \Omega_1 = W_1, W_i = \rho R_i^{-1} = \begin{bmatrix} 1 & 0 \\ 0 & -\gamma_f^{-2} \end{bmatrix} \in \mathcal{R}^{2\times2}$$

$$\mathcal{O}_k = \begin{bmatrix} C_1 \\ \vdots \\ C_k \end{bmatrix}, C_i = \begin{bmatrix} H_i \\ H_i \end{bmatrix} \in \mathcal{R}^{2\times N}.$$

[0074] Here, the following lemmas are formed.

[0075] Lemma 1

[0076] A matrix P_k satisfies the Riccati equation of (14). Therefore, when a gain matrix K_k is obtained, the filter gain $K_{s,\,k}$ is immediately obtained from the following lemma.

[0077] Lemma 2

[0078] The filter gain $K_{s,\,k}$ of the modified H_∞ filter is obtained by using the gain matrix K_k as shown below. In practice, the gain matrix K_k can be fast calculated by the recursive method in Lemma 3.

$$K_{s,k} = G_k^{-1}\bar{K}_k,\ G_k = \rho + \gamma_f^{-2} H_k \bar{K}_k \epsilon R \tag{18}$$

[0079] where,

$$\bar{K}_k(i) = \rho K_k(i, 1), i = 1, 2, \ldots, N \tag{19}$$

[0080] Lemma 3

[0081] The gain matrix K_k is updated as follows.

$$K_{k+1} = m_k - B_k F_k^{-1} \mu_k \epsilon R^{N\times2} \tag{20}$$

[0082] Here, $m_k \epsilon R^{N\times2}$ and $\mu_k\ \epsilon R^{1\times2}$ are obtained by dividing a matrix of $K^U_k = Q^U{}_k^{-1} C^U{}_k$ as shown below.

$$\begin{bmatrix} m_k \\ \mu_k \end{bmatrix} = \begin{bmatrix} 0 \\ K_k \end{bmatrix} + \begin{bmatrix} S_k^{-1} \\ A_k S_k^{-1} \end{bmatrix} [c_k^T + A_k^T C_k^T] \tag{21}$$

[0083] Auxiliary variables $A_k \epsilon R^{N\times1}$, $S_k \epsilon R$, and $B^k F^-{}_{1k} \epsilon R^{N\times1}$ are obtained as well.

[0084] In conclusion, the fast H_∞ filtering algorithm can be summarized as below.

[0085] **FIG. 1** shows a flowchart of the fast algorithm, where L indicates a maximum data length.

[0086] [Step 0] Set initial conditions of a recursive expression as follows, where ϵ_0 is a substantially large positive constant.

$$K_0 = 0, A_{-1} = 0, S_{-1} = \frac{\rho}{\varepsilon_0}, D_{-1} = 0, \hat{x}_{0|0} = 0$$

[0087] [Step 1] Compare time k with the maximum data length L. When the time k is larger than the maximum data length, terminate the processing. When the time k is equal to or smaller than the maximum data length, the processing proceeds to the next step (a conditional statement can be removed, if unnecessary).

[0088] [Step 2] Determine A_k and S_k recursively as follows.

$$\tilde{e}_k = c_k + C_k A_{k-1}\ \epsilon R^{2\times1}$$

$$A_k = A_{k-1} - K_k W_k \tilde{e}_k\ \epsilon R^{N\times1}$$

$$e_k = c_k + C_k A_k\ \epsilon R^{2\times1}$$

$$S_k = \rho S_{k-1} + e_k^T W_k \tilde{e}_k\ \epsilon R$$

[0089] [Step 3] Calculate K^U_k as follows.

$$\breve{K}_k = \begin{bmatrix} S_k^{-1} e_k^T \\ K_k + A_k S_k^{-1} e_k^T \end{bmatrix} \in \mathcal{R}^{(N+1)\times2}$$

[0090] [Step 4] Divide K^U_k as follows.

$$\breve{K}_k = \begin{bmatrix} m_k \\ \mu_k \end{bmatrix} m_k \in \mathcal{R}^{N\times2}, \mu_k \in \mathcal{R}^{1\times2}$$

[0091] [Step 5] Determine D_k, and obtain a gain matrix $K_{s,\,k+1}$ from K_{k+1} as follows:

$$\eta_k = c_{k-N} C_{k+1} D_{k-1}$$

$$D_k = [D_{k-1} - m_k W_k \eta_k][1 - \mu_k W_k \eta_k]^{-1}$$

$$K_{k+1} = m_k - D_k \mu_k$$

$$\bar{K}_{k+1}(i) = \rho K_{k+1}(i, 1), i = 1, \ldots, N$$

$$K_{s,k+1} = G_{k+1}^{-1} \bar{K}_{k+1},\ G_{k+1} = \rho + \gamma_f^{-2} H_{k+1} \bar{K}_{k+1}$$

[0092] where, $\eta_k \epsilon R^{2\times1}$, $D_k \epsilon R^{N\times1}$, $K_{k+1} \epsilon R^{N\times2}$, $K_{s,\,k+1} \epsilon R^{N\times1}$, $0<\rho=1-\gamma_f^{-2} \leqq 1$, $\gamma_f > 1$.

[0093] [Step 6] Update the filter equation of the H_∞ filter as follows.

$$\hat{x}_{k+1|k-1} = \hat{x}_{k|k} + K_{a,\,k+1}(y_{k+1} - H_{k+1}\hat{x}_{k|k})$$

[0094] [Step7] Put the time k forward (k=k+1). The processing returns to Step 2, and the subsequent processes are repeated as long as the data exists.

[0095] Lemma 6 (Existence Condition Suitable for Fast Processing)

[0096] The existence of the fast H_∞ filter can be checked with the computational complexity of O(N) by using the following existence condition.

[0097] [Existence Condition]

$$-e\hat{\Xi}_i+\rho\gamma_f^2>0,\ i=0,\ \ldots,\ k \tag{22}$$

[0098] where,

$$\varrho = 1-\gamma_f^2,\ \hat{\Xi}_i\frac{H_i\tilde{K}_i}{1-H_i\tilde{K}_i} \tag{23}$$

[0099] 4. Computational Complexity of the Present Fast Algorithm

[0100] Next, how the computational complexity of the fast H_∞ filtering algorithm decreases, as compared with the computational requirement of the modified H, filtering algorithm, will be discussed. Only the number of multiplications is used for evaluating the amount of calculation of an equation, and is calculated by the following method.

[0101] Number of multiplications when a J-by-K matrix is multiplied by a K-by-L matrix is J×K×L (times).

[0102] Here, when three or more matrixes or vectors are multiplied, they are calculated from the left unless a direction is specially shown in the figure.

[0103] (Computational Complexity of the Modified H_∞ Filtering Algorithm)

[0104] **FIGS. 2 and 3** are views showing of the amount of calculation of each part of the modified H_∞ filtering algorithm, where N indicates a tap number. In **FIG. 3**(*a*), a calculation for obtaining $R_{e,\ k}^{-1}$ from $R_{e,\ k}$ is ignored. Similarly, in **FIG. 2**(*a*), a calculation for obtaining $(H_{k+1}\hat{P}_{k+1|k}H^T_{k+1}+1)^{-1}$ from $(H_{k+1}\hat{P}_{k+1|k}H^T_{k+1}+1)$ is also ignored.

[0105] As shown in FIGS. **2**(*a*), **3**(*a*), and **3**(*b*), the amount of calculation of each of $K_{s|k+1}$, $R_{e,\ k}$, and $\hat{P}_{k+1|k}$ is in proportion to the square of the tap number. Therefore, the amount of calculation of the entire modified H_∞ filtering algorithm is $O(N^2)$ per unit time step.

[0106] **FIG. 4** is a view showing the amount of calculation required when the order of matrix calculations is changed. More specifically, **FIG. 4** shows the amount of calculation required when the order of matrix calculations in the following part is changed in the Riccati equation, compared with **FIG. 3**(*b*).

[0107] Since the amount of calculation of the above-described part is proportional to the cube of the tap number, the amount of calculation of $\hat{P}_{k+1|k}$ is also in proportion to the cube of the tap number. Accordingly, the amount of calculation of the entire H_∞ filter increases from the square to the cube of the tap number.

[0108] Since either algorithm requires the amount of calculation proportional to the square or cube of the tap number, however, the computational burden for carrying out the filter increases significantly as the tap number increases. In fact, since a tap number used in the field of communication engineering, for example, is approximately 400, the practical use of the algorithm becomes very difficult.

[0109] (Computational Complexity of the Fast H_∞ Filtering Algorithm)

[0110] **FIGS. 5 and 6** are views showing the amount of calculation in the fast H_∞ filtering algorithm. In the expression of K^u_k in **FIG. 5**(*b*), S_k^{-1} is obtained from S_k, but the calculation thereof is ignored. Similarly, in the expression of D_k in **FIG. 6**(*b*), a calculation for obtaining $[1-\mu_k W_k \eta_k]^{-1}$ from $[1-\mu_k W_k \eta_k]$ is also ignored.

[0111] The amount of calculation in the entire present fast algorithm is O(N) per unit time step according to **FIGS. 5 and 6**. Therefore, the amount of calculation in the fast H_∞ filtering algorithm is in proportion to the tap number. In this case, the amount of calculation (the number of multiplications) for performing the fast H_∞ filter once is 28N+16 per unit step, and is approximately double the amount (multiplication frequency) of calculation required for a fast Kalman filter, that is 12N+3.

[0112] As described above, although the computational complexity proportional to the square or cube of the tap number is required for the modified H_∞ filtering algorithm, the computational complexity of the present fast algorithm is smaller and proportional to the tap number.

[0113] 5. Echo Canceller

[0114] The advantage of the present invention will be examined, with an echo canceller being taken as an example.

[0115] An observation value $\{y_k\}$ of an echo $\{d_k\}$ is expressed in the following expression by an (time-varying) impulse response ($\{h_i[k]\}$ of an echo path, where it is considered that a received signal $\{u_k\}$ is an input signal to the echo path:

$$y_k = d_k + v_k = \sum_{i=0}^{N-1} h_i[k]u_{k-i} + v_k,\quad k=0,1,2,\ldots \tag{24}$$

[0116] where, u_k and y_k indicate, respectively, the received signal and the echo at time t_k (=kT, T is a sampling period); v_k indicates circuit noise having zero mean at time t_k; and $h_i[k]$ (i=0, . . . , and N−1) is a time-varying impulse responses assuming a gradual change, and the tap number N thereof is known. Once estimated values $\{\hat{h}_i[k]\}$ of the impulse response are obtained, a quasi echo is generated as follows by using the estimated values.

$$\hat{d}_k = \sum_{i=0}^{N-1} \hat{h}_i[k]u_{k-i},\quad k=0,1,2,\ldots \tag{25}$$

[0117] Subtracting this from the echo ($y_k-\hat{d}_k\approx 0$) , the echo is cancelled, where $u_{k-1}=0$ when k−i<0.

[0118] From the above description, the echo canceller problem is equivalent to successively estimating the impulse response $\{h_i[k]\}$ of the echo path from the received signal $\{u_k\}$ and echo $\{y_k\}$, both of which are directly observable.

[0119] In general, when the H_∞ filter is applied to an echo canceller, equation (24) has to be expressed by a state-space model formed of a state equation and an observation equation. In this case, since the state vector to be obtained is the impulse response $\{h_i[k]\}$, allowing a state vector x_k to fluctuate with w_k, the following state-space model can be constructed for the echo path.

$$\hat{x}_{k+1}=\hat{x}_{k+}w_k,\ \hat{x}_{wk}\in R^N \tag{26}$$

$$y_k=H_kx_k+v_k,\ y_k,v_k\in R \tag{27}$$

$$z_k=H_kx_k,\ z_k\in R,\ H_k\in R^{1\times N} \tag{28}$$

[0120] where,

$$x_k=[h_0[k], \ldots, h_{N-1}[k]]^T,\ w_k=[w_k(1), \ldots, w_k(N)]^T$$

$$H_k=[u_k, \ldots, u_{k-1}],\ L_k=H_k$$

[0121] Modified H_∞ filtering algorithm and fast H_∞ filtering algorithm for such a state-space model are the same as those described above. While the impulse response is estimated, if the occurrence of a transmission signal is detected, the estimation is generally stopped in the meanwhile.

[0122] Thus, when an estimate $\{\hat{h}_i[k]\}$ of the impulse response is obtained, the quasi echo is successively obtained therefrom as follows.

$$\hat{d}_k=H_k\hat{x}_{k|k}=\sum_{i=0}^{N-1}\hat{h}_i[k]u_{k-i} \tag{29}$$

[0123] Therefore, subtracting this from an actual echo to cancel the echo, an echo canceller is implemented. Here, an estimate error, $e_k=y_k-\hat{d}_k$, is called a residual echo.

[0124] 6. Evaluation for Time-Invariant Impulse Response

[0125] (Evaluation of Estimation Accuracy)

[0126] A modified H_∞ filter and a fast H_∞ filter are evaluated by simulation in a case in which the impulse response of an echo path is time-invariant ($h_i[k]=h_i$) and the tap number N thereof is 24.

$$y_k=\sum_{i=0}^{23}h_iu_{k-i}+v_k \tag{30}$$

[0127] **FIG. 7** is a view showing values of the impulse response $\{h_i\}$ in this case. v_k is stationary Gaussian white noise having zero mean and variance σ_v^2 of 1.0×10^{-6}, and a sampling period T is set to 1.0 for convenience.

[0128] The received signal $\{u_k\}$ is approximated by a quadratic AR model as shown below.

$$u_k=\alpha_1u_{k-1}+\alpha_2u_{k-2}+w_{k'} \tag{31}$$

[0129] where, $\alpha_1=0.7$, $\alpha_2=0.1+$, and w_k' is stationary Gaussian white noise having zero mean and variance $\sigma_{w'}^2$ of 0.04.

[0130] The modified H_∞ filter and the fast H_∞ filter will be compared.

[0131] **FIG. 8** includes views showing estimated results of the impulse responses of the modified H_∞ filter and the fast H_∞ filter (initial value $\hat{x}_{0|0}=0$, estimated value $\hat{x}_{100|100}$ at 100th step, $\epsilon_o=20$). FIGS. **8**(*a*) and (*b*) show estimated results of both filters when $\gamma_f=10^5$, and FIGS. **8**(*c*) and (*d*) show estimated results thereof when $\gamma_f=2.0$. From the figures, performance on the estimation accuracy of both filters is equal. In other words, speeding-up does not reduce the estimation accuracy. Note that, if γ_f is too small, the existence condition of the filters is not satisfied. When $\gamma_f=1.0\times10^5$, the results are substantially equal to that of a fast Kalman filter. Therefore, it is found that the fast H_∞ filtering algorithm includes the fast Kalman filtering algorithm and its convergence rate can be accelerated by adjusting γ_f.

[0132] (Evaluation of Computation Time)

[0133] Next, the computation time required for the modified H_∞ filter and that for the fast H_∞ filter are evaluated under conditions where the impulse response of the echo path is time-invariant and the tap number is increased to 24, 48, 96, 192, and 384. Since dispersion may occur in one measurement, the average of four measurements was used. The values shown in **FIG. 7** are used as impulse responses $\{h_i\}$ in simulation, and impulse responses $\{h_i\}$ thereafter ($24\leqq k<N$) are set to 0. The filter calculation is performed up to step **100**. The computation time was measured by a command "etime" of MATLAB on a workstation (sparc, 60 MHz, 32 MB).

[0134] **FIG. 9** is a view showing measurement results of the computation time. In the Riccati equation, matrix calculation is performed for a modified H_∞ filter (2) such that the amount of calculation is in proportion to the square of the tap number, and matrix calculation is performed for a modified H_∞ filter (1) such that the amount of calculation is in proportion to the cube of the tap number (see **FIG. 3**(*b*) and **FIG. 4**). In modified H_∞ filters, since the computational complexity is in proportion to the square or cube of the tap number depending on the order of matrix calculation as described above, they are not practical.

[0135] 7. Evaluation for Time-Varying Impulse Response

[0136] (Evaluation of Tracking Performance)

[0137] The tracking performance of each algorithm will be evaluated by using the echo canceller in a case in which the system (impulse response) is varied with time. It is assumed that the tap number of the impulse response is 48, and $\{h_i\}$ is varied with time, as shown in **FIG. 10**(*a*), based on the values shown in **FIG. 7**. It is also assumed that v_k is stationary Gaussian white noise having zero mean and variance σ_v^2 of 1.0×10^{-6}, and the sampling period T is for convenience. The received signal $\{u_k\}$ is approximated by a quadratic AR model as follows.

$$u_k=\alpha_1u_{k-1}+\alpha_2u_{k-2}+w_k' \tag{32}$$

[0138] Here, $\alpha_1=0.7$, $\alpha_2=0.1$, and w_k' is stationary Gaussian white noise having zero mean and variance $\sigma^2_{w'}$ of 0.04.

[0139] **FIGS. 10 and 11** are views showing the simulation result of each algorithm. These views show the tracking performance of time-varying systems which employ a fast H_∞ filter (fast HF), a fast Kalman filter (fast KF), and LMS algorithm (LMS). **FIG. 10**(*b*) shows the estimates obtained with the fast H_∞ filter when $\gamma_f=2.0$. **FIG. 11**(*a*) shows the estimates obtained with the fast Kalman filter. The initial value of the fast H_∞ filter is set such that $\hat{x}_{0|0}=0$ and so $\epsilon_o=20$, and the initial value of the fast Kalman filter is set in the same way. **FIG. 11**(*b*) shows the estimates obtained by the LMS algorithm, wherein the initial value is set such that $\hat{h}_o=0$, and the step size is set such that $\mu=0.5$ so as to give a stable and rapid convergence. It is found that the tracking performance of the fast H_∞ filter is extremely excellent, and the estimates become stable in about thirty steps after the impulse response is varied. On the other hand, the fast Kalman filter and the LMS algorithm cannot track the impulse response at all.

[0140] Generally, the tracking performance of H_∞ filters having no system noise drops with time since the filter gain becomes smaller due to a decay in the diagonal component of $\hat{P}_{k|k-1}$ and the amount of update of the estimates decreases. In other words, as the number of steps increases, the estimates are updated little. Therefore, in order to improve the tracking performance of Kalman filters and H_∞ filters, an appropriate value needs to be externally added to the diagonal component of the matrix $\hat{P}_{k|k-1}$. If it is directly added, however, a fast algorithm which uses the shift property of an observation matrix H_k cannot be implemented. It is one of significant features of the present invention to solve this problem theoretically by applying a weight ρ of $1-\gamma_f^{-2}$ to the H_∞ evaluation criterion. The weight ρ appears in an update equation of S_k of the fast H_∞ filtering algorithm, as follows.

[0141] (Update of Auxiliary Variable S_k of the Fast H_∞ Filter)

[0142] An auxiliary variable S_k of the fast H_∞ filter is indicated by the following expression.

$$S_k=\rho S_{k-1}+e^T_k W_k e_k,\ 0<\rho=1-\gamma_f^{-2}\leqq 1$$

[0143] In the fast H_∞ filtering algorithm, S_k is used as S_k^{-1} in the equation of K^u_k. In order to largely update the filter equation, S_k^{-1} must be larger. In other word, S_k needs to be kept small to make the large update. The existence of ρ prevents S_k from increasing rapidly, which is resultantly equivalent to adding system noise, and thereby the tracking performance is improved. Since the weight ρ is defined as $1-\gamma^{f-2}$ the tracking performance can be varied by adjusting γ_f as confirmed in the simulation.

[0144] **FIG. 12** is a view showing the relationship between γ_f and ρ. According to the figure, when $\gamma_f=3.0$, $\rho=0.8889$, which means that 89% of $S_{k-}1$ is transmitted to S_k. Note that, if γ_f is set very small, however, the effect of $S_{k-}1$ is significantly reduced and the existence condition of the filter is not satisfied. When γ_f is large, γ=1. An increase in S_k is not suppressed at all, and therefore, the tracking performance drops. When $\gamma_f=\infty$, in particular, the present fast algorithm completely matches the fast Kalman filtering algorithm.

[0145] (Evaluation of Computation Time)

[0146] **FIG. 13** is a view showing the relationship among the tap number and the computation time for the fast H_∞ filter, the fast Kalman filter, and the LMS algorithm, where the number of time steps executed for the filters is 300and $\gamma_f=3.0$. The computation time was measured for the fast H_∞ filter, the fast Kalman filtering algorithm, and the LMS algorithm when the tap number was increased to 48, 96, 192, and 384 in the cases shown in **FIGS. 10 and 11**. Because dispersion may occur in one measurement result, the average of four measurement results, for example, was used.

[0147] It can be confirmed that, in any algorithm, the amount of calculation is in proportion to the tap number. It is also found that when the tap number is large, the computation time for the fast H_∞ filtering algorithm is about a little less than twice the computation time for the fast Kalman filtering algorithm, and is approximately four times longer than that for the LMS algorithm, which is practical. Considering the tracking performance, it can be said that the fast H_∞ filtering algorithm is sufficiently effective.

[0148] 8. Demonstration of Lemmas

[0149] Now, the above-described lemmas will be demonstrated.

[0150] (Demonstration of Lemma 1)

[0151] The inverse matrix of P_k will be indicated by equation (33). Further, a recursive equation for the matrix P_k can be obtained, as shown in equation (34), by using the matrix inversion lemma.

$$\begin{aligned} P_k^{-1}&=\rho O_{k-1}^T\Omega_{k-1}O_{k-1}+C_k^T W_k C_k\\ &=\rho P_{k-1}^{-1}+C_k^T W_k C_k \end{aligned} \tag{33}$$

[0152]

$$\begin{aligned} P_k &= \left[\rho P_{k-1}^{-1} + [H_k^T H_k^T] W_k \begin{bmatrix} H_k \\ H_k \end{bmatrix}\right]^{-1} \\ &= \rho^{-1}P_{k-1} - \rho^{-1}P_{k-1}[\,H_k^T \quad H_k^T\,]\cdot \\ &\quad \left(W_k^{-1} + \begin{bmatrix} H_k \\ H_k \end{bmatrix}\rho^{-1}P_{k-1}[\,H_k^T \quad H_k^T\,]\right)^{-1}\cdot\begin{bmatrix} H_k \\ H_k \end{bmatrix}\rho^{-1}P_{k-1}, \\ \rho P_k &= P_{k-1} - P_{k-1}[\,H_k^T \quad H_k^T\,]\cdot \\ &\quad \left(R_h + \begin{bmatrix} H_k \\ H_k \end{bmatrix}P_{k-1}[\,H_k^T \quad H_k^T\,]\right)^{-1}\cdot\begin{bmatrix} H_k \\ H_k \end{bmatrix}P_{k-1}, \\ P_k &= P_{k-1} - P_{k-1}[\,H_k^T \quad H_k^T\,]\cdot \\ &\quad \left(R_h + \begin{bmatrix} H_k \\ H_k \end{bmatrix}P_{k-1}[\,H_k^T \quad H_k^T\,]\right)^{-1}\cdot\begin{bmatrix} H_k \\ H_k \end{bmatrix}P_{k-1} + \gamma_f^{-2}P_k. \end{aligned} \tag{34}$$

[0153] It is understood, when P_k is replaced with $\hat{P}_{k+1|k}$, that the above equation satisfies the Riccati equation of (13).

[0154] (Demonstration of Lemma 2)

[0155] The gain matrix K_k can be expressed as follows.

$$\begin{aligned} K_k &= P_k C_k^T = [\rho P_{k-1}^{-1} + C_k^T W_k C_k]^{-1} C_k^T \\ &= \rho^{-1}P_{k-1}C_k^T - \rho^{-1}P_{k-1}C_k^T\cdot \\ &\quad [W_k^{-1} + C_k\rho^{-1}P_{k-1}C_k^T]^{-1}C_k\rho^{-1}P_{k-1}C_k^T \\ &= \rho^{-1}P_{k-1}C_k^T - \\ &\quad \rho^{-1}P_{k-1}C_k^T[W_k^{-1} + C_k\rho^{-1}P_{k-1}C_k^T]^{-1}\cdot \\ &\quad [(W_k^{-1} + C_k\rho^{-1}P_{k-1}C_k^T) - W_k^{-1}] \\ &= \rho^{-1}P_{k-1}C_k^T[I + W_k C_k \rho^{-1}P_{k-1}C_k^T]^{-1} \\ &= \rho^{-1}P_{k-1}C_k^T W_k\cdot[W_k + \rho^{-1}W_k C_k P_{k-1}C_k^T W_k]^{-1} \\ &= \rho^{-1}P_{k-1}[H_k^T - \gamma_f^{-2}H_k^T]\left[\begin{bmatrix} 1 & 0 \\ 0 & -\gamma_f^{-2} \end{bmatrix} + \right. \end{aligned} \tag{35}$$

-continued

$$\rho^{-1}\begin{bmatrix} H_k \\ -\gamma_f^{-2}H_k \end{bmatrix}P_{k-1}[\,H_k^T \quad -\gamma_f^{-2}H_k^T\,]\Bigg]^{-1} = \rho^{-1}P_{k-1}[\,H_k^T \quad H_k^T\,](1+H_kP_{k-1}H_k^T)^{-1}$$

[0156] Further, the filter gain can be obtained from the first block column of the gain matrix K_k, as shown in equation (18), by using $G_k=(p+H_kP_{k-1}H^T_k)/(1+H_kP_{k-1}H^T_k)$ and $H_kK_k=H_kP_{k-1}H^T_k/(1+H_kP_{k-1}H^T_k)$.

[0157] (Demonstration of Lemma 3)

[0158] Assuming that the gain matrix K_i (i=0, . . . , and k) is given, the following matrix, K_{k+1}, will be calculated.

$$Q_{k+1}K_{k+1}=C_{k+1}^T \tag{36}$$

[0159] First, equations (37) and (38) are newly introduced to utilize the shift property of C_k. Q^U_k is expressed recursively as shown in equation (39), and is divided as in the following equation (40).

$$\check{C}_k^T = \begin{bmatrix} c_k^T \\ C_k^T \end{bmatrix} = \begin{bmatrix} C_{k+1}^T \\ c_{k-N}^T \end{bmatrix} \in \mathcal{R}^{(N+1)\times 2} \tag{37}$$

$$\check{Q}_k = \sum_{i=1}^{k}\rho^{k-i}\check{C}_i^T W_i \check{C}_i \in \mathcal{R}^{(N+1)\times(N+1)} \tag{38}$$

$$\check{Q}_k=\rho\check{Q}_{k-1}\check{C}_k^TW_k\check{C}_k \tag{39}$$

[0160]

$$\check{Q}_k = \begin{bmatrix} M_k & T_k^T \\ T_k & Q_k \end{bmatrix} = \begin{bmatrix} Q_{k+1} & \underline{T}_k^T \\ \underline{T}_k & \underline{M}_k \end{bmatrix}. \tag{40}$$

[0161] Using this notation, equation (36) of the time steps k and k+1 is included in the following equation.

$$\check{Q}_k\begin{bmatrix} 0 \\ K_k \end{bmatrix} = \begin{bmatrix} \alpha_k^T \\ C_k^T \end{bmatrix} = \check{C}_h^T + \begin{bmatrix} \alpha_k^T - c_k^T \\ 0 \end{bmatrix} \tag{41}$$

$$\check{Q}_k\begin{bmatrix} K_{k+1} \\ 0 \end{bmatrix} = \begin{bmatrix} C_{k-1}^T \\ \beta_k^T \end{bmatrix} = \check{C}_k^T + \begin{bmatrix} 0 \\ \beta_k^T - c_{k-N}^T \end{bmatrix} \tag{42}$$

[0162] where,

$\alpha_k^T=T_k^TK_k\in R^{1\times 2}$, $\beta_k^T=\underline{T}_kK_{k+1}\in R^{1\times 2}$.

[0163] Based on the notation, it is more convenient to obtain $K^U_k\in R^{(N+1)\times 2}$, which satisfies the following equation, than to obtain K_k directly.

$$\check{Q}_k\check{K}_k=\check{C}_k^T \tag{43}$$

[0164] where,

$$\check{K}_k=[k_{k+1}^T\ K_k^T]^T=[K_{k+1}^T\ k_{k-N}^T]^T \tag{44}$$

[0165] To this end, $K^U_k\in R^{(N+1)\times 2}$ can be expressed as shown in equation (46) by using equation (45), obtained from equation (41).

$$\check{C}_k^T = \check{Q}_k\begin{bmatrix} 0 \\ K_k \end{bmatrix} - \begin{bmatrix} \alpha_k^T - c_k^T \\ 0 \end{bmatrix} \tag{45}$$

$$\check{K}_k = \begin{bmatrix} m_k \\ \mu_k \end{bmatrix} = \check{Q}_k^{-1}\check{C}_k^T = \begin{bmatrix} 0 \\ K_k \end{bmatrix} - \check{Q}_k^{-1}\begin{bmatrix} \alpha_k^T - c_k^T \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ K_k \end{bmatrix} - \begin{bmatrix} S_k^{-1} \\ A_kS_k^{-1} \end{bmatrix}[\alpha_k^T - c_k^T] \tag{46}$$

[0166] Here, K^U_k is divided into $m_k\in R^{N\times 2}$ and $\mu_k\in R^{1\times 2}$. Also note that $\alpha^T_k-C^T_k=-(C_k^T+A_k^TC_k^T)$. Further, assuming that Q^U_k has an inverse matrix, auxiliary variables $A_k\in R^{N\times 1}$ and $S_k\in R$ satisfy the following equation.

$$\check{Q}_k\begin{bmatrix} 1 \\ A_k \end{bmatrix} = \begin{bmatrix} S_k \\ 0 \end{bmatrix}\left(\begin{bmatrix} 1 \\ A_k \end{bmatrix}S_k^{-1} = \check{Q}_k^{-1}\begin{bmatrix} 1 \\ 0 \end{bmatrix}\right) \tag{47}$$

[0167] where, the bottom block of the above equation means $T_k+Q_kA_k=0$ or $T_kT=-A_k^TQ_k^T$.

[0168] Next, auxiliary variables $B_k\in R^{N\times 1}$ and $F_k\in R$ such as those shown in the following equation (48) are introduced to delete μ_k in equation (46) without affecting the top block of C^T_k. Further, subtracting $B^U_kF_k^{-1}\mu_k$ from K^U_k in equation (46) provides equation (49).

$$\check{Q}_k\check{B}_k = \check{Q}_k\begin{bmatrix} B_k \\ F_k \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}\left(\check{B}_k = \begin{bmatrix} B_k \\ F_k \end{bmatrix}\right) \tag{48}$$

$$\check{K}_k - \check{B}_kF_k^{-1}\mu_k = \begin{bmatrix} m_k \\ \mu_k \end{bmatrix} - \begin{bmatrix} B_kF_k^{-1} \\ 1 \end{bmatrix}\mu_k = \begin{bmatrix} m_k - B_kF_k^{-1}\mu_k \\ 0 \end{bmatrix} \tag{49}$$

[0169] Then, the left-hand side of equation (49) is multiplied by Q^U_k from the left to obtain the following equation.

$$\check{Q}_k(\check{K}_k - \check{B}_kF_k^{-1}\mu_k) = \check{Q}_k\check{K}_k - \check{Q}_k\check{B}_kF_k^{-1}\mu_k = \check{C}_k^T - \begin{bmatrix} 0 \\ 1 \end{bmatrix}F_k^{-1}\mu_k = \check{C}_k^T - \begin{bmatrix} 0 \\ F_k^{-1}\mu_k \end{bmatrix} \tag{50}$$

[0170] Equation (49) is substituted for the left-hand side of the above equation. Then, equation (43) is expressed as follows:

$$\check{Q}_k(\check{K}_k - \check{B}_kF_k^{-1}\mu_k) = \check{C}_k^T - \begin{bmatrix} 0 \\ F_k^{-1}\mu_k \end{bmatrix}, \tag{51}$$

-continued

$$\begin{bmatrix} Q_{k+1} & \underline{T}_k^T \\ \underline{T}_k & M_k \end{bmatrix}\begin{bmatrix} m_k - B_k F_k^{-1}\mu_k \\ 0 \end{bmatrix} = \begin{bmatrix} C_{k+1}^T \\ c_{k-N}^T \end{bmatrix} + \begin{bmatrix} 0 \\ -F_k^{-1}\mu_k \end{bmatrix}$$

[0171] This is the same form as equation (42). The following equation (52) can be obtained from the top block of equation (51).

$$Q_{k+1}(m_k - B_k F_k^{-1}\mu_k) = C_{k+1}^T \tag{52}$$

[0172] Equations (36) and (52) are compared to obtain the update equation of the gain matrix K_k.

[0173] (Lemma 4)

[0174] The auxiliary variables A_k and S_k can be obtained as follows:

$$A_k = A_{k-1} - K_k W_k [c_k + C_k A_{k-1}] \in R^{N\times 1} \tag{53}$$

$$S_k = \rho S_{k-1} + [c_k^T + A_k^T C_k^T] W_k [c_k + C_k A_{k-1}] \in R \tag{54}$$

[0175] where, $A_{-1}=0$, and $S_{-1}=1/\epsilon_0$.

[0176] (Demonstration) By using the following equation (55) of A_k and S_k and equation (39), equation (56) is obtained.

$$\breve{Q}_{k-1}\begin{bmatrix} 1 \\ A_{k-1} \end{bmatrix} = \begin{bmatrix} S_{k-1} \\ 0 \end{bmatrix} \tag{55}$$

$$\begin{aligned} \breve{Q}_k\begin{bmatrix} 1 \\ A_{k-1} \end{bmatrix} &= \rho\breve{Q}_{k-1}\begin{bmatrix} 1 \\ A_{k-1} \end{bmatrix} + \breve{C}_k^T W_k [c_k + C_k A_{k-1}] \\ &= \begin{bmatrix} \rho S_{k-1} \\ 0 \end{bmatrix} + \begin{bmatrix} c_k^T \\ C_k^T \end{bmatrix} W_k [c_k + C_k A_{k-1}] \end{aligned} \tag{56}$$

[0177] On the other hand, the following equation is obtained by multiplying both sides of equation (41) by $W_k[C_k + C_k A_{k-1}]$.

$$\breve{Q}_k\begin{bmatrix} 0 \\ K_k \end{bmatrix} W_k [c_k + C_k A_{k-1}] = \begin{bmatrix} \alpha_k \\ C_k^T \end{bmatrix} W_k [c_k + C_k A_{k-1}]. \tag{57}$$

[0178] By subtracting equation (57) from equation (56), the following equation (58) is formed.

$$\begin{aligned} \breve{Q}_k\left[\begin{bmatrix} 1 \\ A_{k-1} \end{bmatrix} - \begin{bmatrix} 0 \\ K_k \end{bmatrix} W_k [c_k + C_k A_{k-1}]\right] &= \begin{bmatrix} \rho S_{k-1} \\ 0 \end{bmatrix} + \begin{bmatrix} c_k^T \\ C_k^T \end{bmatrix} \\ W_k [c_k + C_k A_{k-1}] &- \begin{bmatrix} \alpha_k^T \\ C_k^T \end{bmatrix} W_k [c_k + C_k A_{k-1}], \\ \breve{Q}_k\begin{bmatrix} 1 \\ A_{k-1} - K_k W_k [c_k + C_k A_{k-1}] \end{bmatrix} &= \\ &\begin{bmatrix} \rho S_{k-1} + [c_k^T - \alpha_k^T] W_k [c_k + C_k A_{k-1}] \\ 0 \end{bmatrix} \end{aligned} \tag{58}$$

[0179] This equation is compared with equation (47). Since $\alpha_k^T = T_k^T K_k = -A_k^T C_k^T$, equations (53) and (54) are obtained.

[0180] (Lemma 5)

[0181] The auxiliary variable $D_k = B_k F_k^{-1}$ is obtained by the following equation (59). F_k is updated by the following equation (60).

$$D_k = [D_{k-1} - m_k W_k \eta_k][1 - \mu_k W_k \eta_k]^{-1} \in R^{N\times 1} \tag{59}$$

$$F_k = F_{k-1}[1 - \mu_k W_k \eta_k]/\rho \in R \tag{60}$$

[0182] where, $\eta_k = C^U{}_k D^U{}_{k-1} = c_{k-N} + C_{k+1} D_{k-1}$, $D_{-1}=0$, and $F_{-1}=0$.

[0183] (Demonstration) In order to update B_k and F_k, equation (62) is formed by using equation (61).

$$\breve{Q}_{k-1}\breve{B}_{k-1} = \breve{Q}_{k-1}\begin{bmatrix} B_{k-1} \\ F_{k-1} \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \tag{61}$$

$$\begin{aligned} \breve{Q}_k\breve{B}_{k-1} &= \rho\breve{Q}_{k-1}\breve{B}_{k-1} + \breve{C}_k^T W_k \breve{C}_k \breve{B}_{k-1} \\ &= \rho\begin{bmatrix} 0 \\ 1 \end{bmatrix} + \breve{C}_k^T W_k \breve{C}_k \breve{B}_{k-1} \end{aligned} \tag{62}$$

[0184] In order to modify the above equation so as to have the same form as equation (61), $C^U{}_k{}^T W_k C^U{}_k B_{k-1}{}^U$ is subtracted from equation (62) to obtain the following equation.

$$\breve{Q}_k\breve{B}_{k-1} - \breve{C}_k^T W_k \breve{C}_k \breve{B}_{k-1} = \breve{Q}_k\breve{B}_{k-1} - \breve{Q}_k\breve{K}_k W_k \breve{C}_k \breve{B}_{k-1} = \rho\begin{bmatrix} 0 \\ 1 \end{bmatrix}, \tag{63}$$

$$\breve{Q}_k\left[\breve{B}_{k-1}\breve{K} W_k \breve{C}_k \breve{B}_{k-1}\right] = \rho\begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

[0185] Comparing the above last equation with equation (48) yields a recursive equation for $B^U{}_k$.

$$\breve{B}_k = (\breve{B}_{k-1} - K_k W_k \breve{B}_{k-1})/\rho \tag{64}$$

[0186]

$$D_k = B_k F_k^{-1},\ \breve{D}_k = \breve{B}_k F_k^{-1} = \begin{bmatrix} D_k \\ 1 \end{bmatrix} \tag{65}$$

[0187] B_k and F_k are updated by this equation.

[0188] Since they appear only for $B^U{}_k$ and $D_k = B_k F_k^{-1} \in R^{N\times 1}$, however, it is more convenient to express equations (48) and (64) to the following equation (65). The matrix D_k satisfies the following equation (66).

$$\breve{Q}_k\breve{D}_k = \breve{Q}_k\breve{B}_k F_k^{-1} = \begin{bmatrix} 0 \\ 1 \end{bmatrix} F_k^{-1},\ \breve{Q}_k\begin{bmatrix} D_k \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ F_k^{-1} \end{bmatrix} \tag{66}$$

$$\begin{aligned} \breve{Q}_k\left[\breve{B}_{k-1}F_{k-1}^{-1} - \breve{K}_k W_k \breve{C}_k \breve{B}_{k-1}F_{k-1}^{-1}\right] &= \breve{Q}_k\left[\breve{D}_{k-1} - \breve{K}_k W_k \breve{C}_k \breve{D}_{k-1}\right] \\ &= \begin{bmatrix} 0 \\ \rho F_{k-1}^{-1} \end{bmatrix} \end{aligned} \tag{67}$$

[0189] Next, equation (63) is multiplied by F_{k-1}^{-1} to obtain the following equation (67), and is further expressed by the following equation (68) when $D^U_{k-1}=B^U_{k-1}F_{k-1}^{-1}$ is used.

$$\check{Q}_k\left[\check{D}_{k-1}-\begin{bmatrix}m_k\\ \mu_k\end{bmatrix}W_k\check{C}_k\check{D}_{k-1}\right]=\begin{bmatrix}0\\ \rho F_{k-1}^{-1}\end{bmatrix},\tag{68}$$

$$\check{Q}_k\begin{bmatrix}D_{k-1}-m_kW_k\check{C}_k\check{D}_{k-1}\\ 1-\mu_kW_k\check{C}_k\check{D}_{k-1}\end{bmatrix}=\begin{bmatrix}0\\ \rho F_{k-1}^{-1}\end{bmatrix}$$

[0190] Therefore, the following equation is obtained when equation (68) is multiplied by $[1-\mu_kW_kC^U_kD^U_{k-1}]^{-1}$.

$$\check{Q}_k\begin{bmatrix}\left[D_{k-1}-m_kW_k\check{C}_k\check{D}_{k-1}\right]\left[1-\mu_kW_k\check{C}_k\check{D}_{k-1}\right]^{-1}\\ 1\end{bmatrix}=\begin{bmatrix}0\\ \rho F_{k-1}^{-1}\left[1-\mu_kW_k\check{C}_k\check{D}_{k-1}\right]^{-1}\end{bmatrix}$$

[0191] By comparing this equation with equation (66), an update equation for D_k and F_k is finally obtained.

[0192] (Demonstration of Lemma 6 (Existence Condition Suitable for Fast Processing))

[0193] As described above, the existence of the fast H_∞ filter can be checked with the computational complexity of O(N) by using the existence condition of equations (22) and (23). A demonstration thereof will be shown below.

[0194] When the characteristic equation of a 2×2 matrix $R_{e,k}$ shown in the following equation (69) is solved, a eigenvalue λ_i of $R_{e,k}$ is obtained by the following equation (70).

$$|\lambda I-R_{e,k}|=\begin{vmatrix}\lambda-\left(\rho+H_k\hat{\sum}_{k|k-1}H_k^T\right) & -H_k\hat{\sum}_{k|k-1}H_k^T\\ -H_k\hat{\sum}_{k|k-1}H_k^T & \lambda-\left(-\rho\gamma_f^2+H_k\hat{\sum}_{k|k-1}H_k^T\right)\end{vmatrix}$$
$$=\lambda^2-\left(2H_k\hat{\sum}_{k|k-1}H_k^T+\rho\varrho\right)\lambda-\rho^2\gamma_f^2+\rho\varrho H_k\hat{\sum}_{k|k-1}H_k^T=0\tag{69}$$

$$\lambda_i=\frac{\Phi\pm\sqrt{\Phi^2-4\rho\varrho H_k\hat{\Sigma}_{k|k-1}H_k^T+4\rho^2\gamma_f^2}}{2}\tag{70}$$

[0195] where, $\Phi=2H_k\hat{\Sigma}_{k|k-1}H_k^T+\rho e$, $e=1-\gamma_f^2$

[0196] If the following expression (71) is satisfied, one of the two eigenvalues of the matrix $R_{e,k}$ is positive and the other is negative, and the matrixes R_k and $R_{e,k}$ have the same inertia. Therefore, the existence condition of equation (22) is obtained by using the following equation (72). Here, the calculation of H_kK_k requires the same number of multiplications as O(N).

$$-4\rho eH_k\hat{\Sigma}_{k|k-1}H_k^T+4\rho^2\gamma_f^2>0\tag{71}$$

[0197]

$$H_k\hat{\sum}_{k|k-1}H_k^T=\frac{H_k\tilde{K}_b}{1-H_kK_b}\tag{72}$$

[0198] Industrial Applicability

[0199] According to the present invention, as described above, the fast real-time identification of time-invariant and time-variant systems can be implemented by using the fast algorithm (fast H_∞ filtering algorithm) for the modified H_∞ filters developed based on the new H_∞ evaluation criterion. In addition, according to the present invention, the present algorithm includes, as a particular case, the fast Kalman filtering algorithm, and a term corresponding to the covariance of system noise which is dominant in the tracking performance of a time-varying system can be theoretically determined. Further, according to the present invention, a fast time-varying system identification method can be provided, which is very effective particularly when a system (impulse response) is varied discontinuously with time, such as an echo canceller for a time-varying system which varies extremely as sudden line switching. Furthermore, according to the present invention, a system identification method can be provided, which is applicable to echo cancellers in communication systems and acoustic systems, sound-field reproduction, and noise control.

1. A system identification method for performing a fast real-time identification of a time-invariant or time-variant system, wherein an H_∞ filter equation expressed by the following equation is used,

$$\hat{x}_{k+1|k+1}=x_{k|k}+K_{s,k+1}(\gamma_{k+1}H_{k+1}x_{k|k})$$

where,

$\hat{x}_{k|k}$: The estimate of state x_k at time k, obtained by using observation signals y_o to y_k

y_k: Observation signal

$K_{s,k+1}$: Filter gain

H_k: Observation matrix

and a filtering algorithm robust against disturbance is formed by setting, as an H_∞ evaluation criterion, a maximum energy gain from disturbance weighted by a weight (ρ) of the evaluation function to a filter error to be smaller than a predetermined upper limit (γ_f^2)

2. A system identification method according to claim 1, wherein, as the H_∞ evaluation criterion, the maximum value of {(value indicating the filter error/weight (ρ) of the evaluation function)/[(value indicating an initial state)+(value indicating system noise)+(value indicating observation noise/weight (ρ) of the evaluation function)]} is set to be smaller than the predetermined upper limit (γ_f^2).

3. A system identification method according to claim 1 or **2**, wherein an output signal is further obtained from the state estimate $\hat{x}_{k|k}$ at time k by the following equation.

$$\hat{z}_{k|k}=H_k\hat{x}_{k|k}$$

z_k: Output signal

4. A system identification method according to any one of claims 1 to 3, wherein the following equation shows the relationship between the weight (ρ) of the evaluation function and the predetermined upper limit (γ_f^2)

$$0<\rho=1-\gamma_f^{-2}\leqq 1$$

$$\gamma_f>1$$

5. A system identification method according to any one of claims 1 to 4, wherein the filter gain $K_{s,k}$ is given by the following relational equation by using a gain matrix K_k.

$$K_{s,k}=G_k^{-1}\tilde{K}_k,\ G_k=\rho+\gamma_f^{-2}H_k\tilde{K}_k\in R$$

where,

$$\tilde{K}_k(i)=\rho K_k(i,1),\ i=1,2,\ldots,N.$$

6. A system identification method according to any one of claims 1 to 5, comprising steps of;

setting initial conditions of a recursive equation of the gain matrix K_k, the auxiliary variables, and the state estimate $\hat{x}_{k|k}$,

recursively determining the auxiliary variables at time k and obtaining a second gain matrix in which a row including the auxiliary variables is added to the gain matrix K_k,

dividing the second gain matrix and obtaining first and second divisional gain matrixes,

obtaining a gain matrix K_{k+1} at time k+1 from an equation including the first and second divisional gain matrixes, and obtaining a filter gain $K_{s,k+1}$ at time k+1 from the relational equation of the gain matrix K_k and the filter gain $K_{s,k}$,

updating the H_∞ filter equation according to the obtained filter gain $K_{s,k+1}$, and

repeating each of the above steps with the time being put forward.

7. A system identification method according to any one of claims 1 to 6, wherein the existence of the fast H_∞ filter is checked by using the following equation as an existence condition suitable for fast processing, with the computational complexity of O(N).

$$-\varrho\hat{\Xi}_i+\rho\gamma_f^2>0,\ i=0,\ldots,k$$

where,

$$\varrho=1-\gamma_f^2,\ \hat{\Xi}_i=\frac{H_i\tilde{K}_i}{1-H_i\tilde{K}_i}$$

8. A system identification method according to any one of claims 1 to 7, wherein an echo canceller is implemented by

applying the H_∞ filter equation to obtain the state estimate $\hat{x}_{k|k}$,

producing a quasi echo as in the following equation, and

canceling an actual echo by the obtained quasi echo.

$$\hat{d}_k=H_k\hat{x}_{k|k}=\sum_{i=0}^{N-1}\hat{h}_i[k]u_{k-i}$$

$$H_k=[u_k,\ldots,u_{k-N+1}]$$

where,

$\hat{d}_k$ Quasi-echo

u_k Received signal

N Tap number

$\hat{h}_i[k]$ Estimate of impulse response of echo path

* * * * *