Recursive Relaxation Identification of Linear Multivariable Systems with its Parallel Algorithm

MI LU

and

KUNLIN LIU

Department of Electrical Engineering, Texas A&M University, College Station, TX 77843-3128

ABSTRACT

In this paper, we propose a new method for the identification of discrete-time linear multivariable systems. This method comprises three recursive computation schemes including two parameter estimate schemes for both state equation and observation equation, respectively, and the state estimate scheme via Kalman filtering. We call this method the recursive relaxation identification, abbreviated as RRI. Compared with the existing methods, the RRI algorithm has advantages of simplification, fast computation, and high efficiency, and is suitable for fast real-time on-line adaptive identification of linear multivariable systems.

1. INTRODUCTION

Without losing generality, we consider the linear multivariable system described by the following state-space description:

\[
x(k + 1) = Ax(k) + Bu(k) + w(k),
\]

\[
y(k) = Cx(k) + v(k),
\]

where \( x(k) \in \mathbb{R}^n \), \( u(k) \in \mathbb{R}^r \) is a deterministic input vector, \( y(k) \in \mathbb{R}^m \), \( r, m \leq n \), \( \{w(k)\} \) and \( \{v(k)\} \) are uncorrelated zero mean Gaussian white noise sequences.
noise sequences, assumed that

\[ E\{w(k)w^T(j)\} = w\delta(k-j), \quad w \geq 0, \quad (3) \]

\[ E\{v(k)v^T(j)\} = v\delta(k-j), \quad v > 0, \quad (4) \]

and the initial state \( x(0) \) holding

\[ Ex(0) = \dot{x}(0), \quad (5) \]

\[ E\{w(k)x^T(0)\} = 0, \quad \forall k, \quad (6) \]

\[ E\{v(k)x^T(0)\} = 0, \quad \forall k, \quad (7) \]

\[ E\{x(0)x^T(0)\} = P(0) \geq 0, \quad (8) \]

where \( \dot{x}(0), \rho(0), w, \) and \( v \) are considered given by some prior knowledge about the system.

The unknown parameter matrices \( A \in R^{n \times n}, B \in R^{n \times r}, \) and \( C \in R^{m \times n} \) are expected to be identified by using the observation data from \( \{y(k)\} \) and \( \{u(k)\} \).

As is well known, there are two existing methods to deal with the problem under consideration. The first one, as a classical method, can be summarized as follows [1-4].

Substitute the solution of (1) into (2), yielding a so-called input-output model corresponding to the considered state-space model. Then, solve the parameter identification from the obtained input-output model. Based on the result of identification, according to some canonical form determined in advance, construct the minimum-realization of the multivariable system under consideration.

It is apparent that we can only obtain the algebraically equivalent one of the wanted real system. Moreover, solving the minimum-realization problem of a multivariable system from its transform function is almost always highly complex.

The second method is the so-called extended Kalman filtering, or EKF in brief [5]. The EKF used for the system identification overcame some drawbacks of the first method and resulted in a substantial improvement in the applications of identifying multivariable systems. However, the problems incurred by the EKF are that an extended state vector leads to the increased dimension of state vector, and that applications always involve dealing with some nonlinear problems which means that two nonlinear
functions requiring some linear Taylor approximation always follow the linear system identification problem. The modified extended Kalman filtering (MEKF) and its parallel algorithm [6, 7] have enhanced the computation efficiency and stability of the EKF, but still have to deal with the nonlinear and algorithmic complex problems.

The RRI with its parallel algorithm to be discussed in the next two sections can overcome the drawbacks of the previous methods and are completely constructed by dealing with linear problems.

2. THE RECURSIVE RELAXATION IDENTIFICATION METHOD

In this section, we construct the recursive relaxation identification method for the linear multivariable system described in expressions (1) and (2).

Assume that the linear multivariable system under consideration is observable and controllable, and set

\[ \hat{x}(k|k) = E\{x(k) | y(t); t = 0, 1, \ldots, k\}, \]

\[ \hat{x}(k|k-1) = E\{x(k) | y(t); t = 0, 1, \ldots, k-1\}, \]

\[ P(k) = P(k,k) = E\{(x(k) - \hat{x}(k)) (x(k) - \hat{x}(k))^T\}, \]

\[ P(k,k-1) = E\{(x(k) - \hat{x}(k|k-1)) (x(k) - \hat{x}(k|k-1))^T\}. \]

First, suppose that the estimators of parameter matrices A, B, and C are already obtained at the kth instant, noted as \( \hat{A}_k \), \( \hat{B}_k \), and \( \hat{C}_k \); then the Kalman filtering estimator and predictor at the \((k+1)\)th instant, based on the obtained \( \hat{A}_k \), \( \hat{B}_k \), and \( \hat{C}_k \), can be given as follows [8]:

\[ P(k+1,k) = \hat{A}_k P(k) \hat{A}_k^T + w, \]

\[ \hat{x}(k+1|k) = \hat{A}_k \hat{x}(k) + \hat{B}_k u(k), \]

\[ G(k+1) = P(k+1,K) \hat{C}_k \left[ \hat{C}_k P(k+1,k) \hat{C}_k^T + v \right]^{-1}, \]

\[ \hat{x}(k+1) = \hat{x}(k+1|k) + G(k+1) \left[ y(k+1) - \hat{C}_k \hat{x}(k+1|k) \right]. \]
\[ P(k+1) = P(k+1,k) - G(k+1) \hat{C}_k P(k+1,k), \quad (17) \]

To ensure the symmetry of matrix \( P(k) \) in our computation process, (17) can be rewritten as

\[ P(k+1) = P(k+1,k) - G(k+1) \left[ \hat{C}_k P(k+1,k) \hat{C}_k^T + v \right] G^T(k+1). \quad (18) \]

Second, using the data from \( \{\hat{x}(k)\} \) and \( \{u(k)\} \), according to the state equation, we construct the least squares scheme to solve the estimators of the parameter matrices \( A \) and \( B \).

Assume that \( k > n + r \) and introduce notations

\[ \varphi_k^T = \begin{bmatrix} \hat{x}(k) \\ u(k) \end{bmatrix} \in \mathbb{R}^{(n+r) \times 1}, \quad (19) \]
\[ \Phi_k = \begin{bmatrix} \varphi_0^T \\ \varphi_1^T \\ \vdots \\ \varphi_{k-1}^T \end{bmatrix} \in \mathbb{R}^{(n+r) \times k}, \quad (20) \]
\[ \Theta = \begin{bmatrix} A & B \end{bmatrix} \in \mathbb{R}^{n \times (n+r)}, \quad (21) \]
\[ X_k = \begin{bmatrix} \hat{x}(1) & \hat{x}(2) & \cdots & \hat{x}(k) \end{bmatrix} \in \mathbb{R}^n \times k. \quad (22) \]

If \( \text{rank} \, \Phi_k = n + r \), then the least squares estimator of \( \Theta \), at the \( k \)th instant, should be

\[ \hat{\Theta}_k = X_k \Phi_k^T (\Phi_k \Phi_k^T)^{-1}, \quad (23) \]

based on the data sets \( \{\hat{x}(t)\}_{t=0}^{k} \) and \( \{u(t)\}_{t=0}^{k-1} \) [9, 10]. Setting

\[ P_k = \left( \sum_{i=0}^{k-1} \varphi_i^T \varphi_i \right)^{-1} = (\Phi_k \Phi_k^T)^{-1} \in \mathbb{R}^{(n+r) \times (n+r)}, \quad (24) \]

we can see that

\[ P_{k+1}^{-1} = P_k^{-1} + \varphi_k^T \varphi_k. \quad (25) \]
Considering the estimator of $\Theta$, at the $(k + 1)$th instant, we have

$$\hat{\Theta}_{k+1} = X_{k+1} \Phi^T_{k+1} P_{k+1}$$

$$= \left( \sum_{t=0}^{k-1} \hat{x}(t+1) \varphi_t + \hat{x}(k+1) \varphi_k \right) P_{k+1}. \quad (26)$$

Noticing that

$$\sum_{t=0}^{k-1} \hat{x}(t+1) \varphi_t = \hat{\Theta}_k P_k^{-1}$$

we can rewrite (26) as

$$\hat{\Theta}_{k+1} + \hat{\Theta}_k + \left[ \hat{x}(k+1) - \hat{\Theta}_k \varphi_k^T \varphi_k \right] \varphi_k P_{k+1}. \quad (27)$$


$$P_{k+1} = P_k - P_k \varphi_k^T R_{k+1}, \quad (29)$$

$$R_{k+1} = \frac{\varphi_k P_k}{1 + \varphi_k P_k \varphi_k^T}. \quad (30)$$

In fact, it can be obtained that

$$\varphi_k P_{k+1} = R_{k+1}. \quad (31)$$

For the consideration of ensuring the symmetry of matrix $P_k$, which supplies numerical stability for the computation, we rewrite (29) and obtain the final recursive schemes for computing $\hat{\Theta}_{k+1}$ as follows:

$$P_{k+1} = P_k - \left( 1 + \varphi_k P_k \varphi_k^T \right) R_{k+1} R_{k+1}, \quad (32)$$

$$\hat{\Theta}_{k+1} = \hat{\Theta}_k + \left[ \hat{x}(k+1) - \hat{\Theta}_k \varphi_k^T \right] R_{k+1}. \quad (33)$$
Third, using the data sets \( \{ \hat{x}(t) | t = 0, 1, \ldots, k + 1 \} \) and \( \{ y(t) | t = 0, 1, \ldots, k + 1 \} \), consider the least squares scheme corresponding to the observation (2) and set

\[
Q_k = \left( \sum_{t=0}^{k} \hat{x}(t) \hat{x}^T(t) \right)^{-1} \in R^{n \times n},
\]

(34)

with the assumption \( \text{rank}[\hat{x}(0), \hat{x}(1), \ldots, \hat{x}(k)] = n \). Then the recursive estimator of matrix \( C \) can be similarly formulated as

\[
G_{k+1} = G_k + (y(k + 1) - \hat{C}_k \hat{x}(k + 1)) S_{k+1},
\]

(37)

It is clear that a set of completely recursive relaxation algorithm schemes can be established under some initialization of \( \hat{x}(0), P(0), \) and \( P_0, \hat{\Theta}_0 \) as well as \( Q_0 \) and \( \hat{C}_0 \). The recursive relaxation algorithm schemes are designed to implement formulations (13)–(16), formulations (18), (30), (32), (33), as well as formulations (35)–(37). Such a set of schemes make it possible to design a parallel computation pattern which has nicer parallel and pipelining characteristics.

3. PARALLEL ALGORITHM OF THE RRI

The parallel algorithm of the RRI is designed to comprise three main paths or pipelines for recursively computing \( \hat{x}(k), \hat{\Theta}_k \), and \( \hat{C}_k \), respectively, and the corresponding algorithm schemes are summarized as follows.

Path I

1) \( \hat{x}(0) = E x(0), P(0) = E \{ x(0) \hat{x}^T(0) \} \), set \( k = 0 \)
2) picking up \( \hat{\Theta}_k \) and \( \hat{C}_k \) from Path II and Path III
3) \( P(k + 1, k), \) for \( k = 0, 1, \ldots, \) via (13)
4) \( T^{-1}_{k+1} = (\hat{C}_k P(k + 1, k) \hat{C}_k^T + \nu)^{-1}, \) for \( k = 0, 1, \ldots \)
5) \( G(k + 1), \) for \( k = 0, 1, \ldots, \) via (15)
6) \( \hat{x}(k + 1), \) for \( k = 0, 1, \ldots, \) via (16), being transmitted to Path II and Path III
7) $P(k + 1)$, for $k = 0, 1, \ldots$, via (18)
8) $k := k + 1$, return to 2)

Path II

1) initial guess of $\hat{\Theta}_0$, $P_0 = \alpha^2 I$ ($10 \leq \alpha \leq 100$), set $k = 0$
2) picking up $\hat{x}(k + 1)$ from Path I
3) $R_{k+1}$, for $k = 0, 1, \ldots$, via (30)
4) $P_{k+1}$, for $k = 0, 1, \ldots$, via (32)
5) $\hat{\Theta}_{k+1}$, for $k = 0, 1, \ldots$, via (33), being transmitted to Path I
6) $k := k + 1$, return to 2)

Path III

1) initial guess of $\hat{C}_0$, $Q_0 = \beta^2 I$ ($10 \leq \beta \leq 100$), set $k = 0$
2) picking up $\hat{x}(k + 1)$ from Path I
3) $S_{k+1}$, for $k = 0, 1, \ldots$, via (35)
4) $Q_{k+1}$, for $k = 0, 1, \ldots$, via (36)
5) $\hat{C}_{k+1}$, for $k = 0, 1, \ldots$, via (37), being transmitted to Path I
6) $k := k + 1$, return to 2)

Each computation path employs a group of processors to execute the partial parallel computation with communications to others. An illustrated pattern for the overall algorithm structure is given in Fig. 1.
For solving $\hat{\Theta}_{k+1}$ and $\hat{C}_{k+1}$, each iteration of the RRI consists of two stages, which are described below.

STAGE I: Complete the state filtering estimate.

1) Compute $P(k + 1, k)$ with the available $\hat{A}_k$ and $P(k)$ as well as $w$, in time $O(\log n)$ using $O(n^2 m)$ processors.

2) Compute $\hat{T}_{k+1}$ and $\hat{C}_k P(k + 1, k) \hat{C}_k^T + \nu$ simultaneously, in time $O(\log n)$ using $O(n^2 m)$ processors. Store $\hat{T}_{k+1}$ for the use in Stage II 6), with $m^2$ processors.

3) Evaluate $\hat{T}_{k+1}$ using the fast iterative matrix inversion [12], in time $O(\log^2 m)$ employing $O(m^3)$ processors.

4) Compute $G(k + 1)$ in time $O(\log n + \log m)$ using $O(nm^2)$ processors in time $O(\log m)$.

5) Compute $\hat{x}(k + 1)$ in time $O(\log n + \log m)$ using $O(nm^2)$ processors.

STAGE II: Compute the variance matrix of the state filtering estimate and complete the computation of parameters identification.

6) Simultaneously compute $P(k + 1)$ via (18) and (38),

$$P(k + 1) = P(k + 1, k) - G(k + 1) \hat{T}_{k+1} G^T(k + 1),$$

$R(k + 1)$ via (30) and $S(k + 1)$ via (35). $nm^2$ processors are used for evaluating $P(k + 1)$ in time $O(\log m)$, $(n + r)^2$ processors for $R(k + 1)$ in time $O(\log(n + r))$, and $n^2$ processors for $S(k + 1)$ in $O(\log n)$ time steps. At the same time, save the following two scalars:

$$r_{k+1} = 1 + \varphi_k P_k \varphi_k^T,$$

$$(40)$$

$$h_{k+1} = 1 + \hat{x}^T(k + 1) Q_k \hat{x}(k + 1)$$

$$(41)$$
on two single processors for the next step. Notice that $\max(m, r) \leq n$, and the execution of this step can be completed by employing $O(n^2 m)$ processors in time $O(\log(n + r))$, excluding the communication costs.

7) Compute $\hat{\Theta}_{k+1}$ and $\hat{C}_{k+1}$ simultaneously; the employed processor numbers are $O(n^2)$ and $O(mn)$, and the execution time $O(\log(n + r))$ and $O(\log n)$, respectively. The number of total employed processors for this step is $O(n^2 + nm)$ and the execution can be completed in time $O(\log(n + r))$.

8) Evaluate $P_{k+1}$ and $Q_{k+1}$ simultaneously using $O((n + r)^2 + n^2)$ processors in time $O(1)$.

9) Return to 1) by setting $k := k + 1$. 


Overall, for one iteration step in the RRI, the number of employed processors is \( O(n^2m) \) and the required execution time is \( O(\log^2 m + \alpha \log(n+r)) \), with \( \alpha \) being a small positive integer.

To compare the computation complexity of our algorithm with that of the previously existing ones, we simply recall the time performance of the EKF and the MEKF as follows.

The computation time of the EKF is \((\log N + 3)N^3 + (\log n + 3)n^3 + (n^2 + N^2)(r + m + 4)\), where \( N = n + \text{Dim}(\theta_k) \) and \( \theta_k \) together with \( x(k) \) form the extended state vector [6].

For the MEKF, the computation in one iteration step takes \((\log N + 8)N + 3m + r \) time using
\[ (1.5((n + m)^2 + N^2) + 2m^2 + m(2n + m) + 0.5r(N + n + m) + (N + n + 2m)) \]
processors [6].

To compare the numbers of processors employed by the RRI and by the MEKF, it can be seen that \((n + m)^2 + N^2\) processors are required for the MEKF given \( N > n \geq m \), while \( n^2m \) processors are sufficient for the RRI.

When \( n \) is not extremely large, which is usually the case, we can see that the processors employed by the RRI are less than by the MEKF. As for the time complexity, we can find that the time performance of the RRI is better than that of the MEKF.

4. STABILITY, CONVERGENCE, AND INITIAL VALUES

The convergence analysis of the algorithm proposed in this paper is based upon the following facts.

4.1 STABILITY OF THE KALMAN FILTERING ALGORITHM

Let the linear system described by (1) and (2) be both completely controllable and observable. Then, the Kalman filtering algorithm formed by (13)–(18) is asymptotically stable, i.e., for any initial state \( x(0) \) assuring
\[ P(0, -1) = P(0) = \text{Var} x(0) \]
nonnegative definite and symmetric, \( P(k, k - 1) \to P \) and \( G(k) \to G \) as \( k \to 0 \), where \( P > 0 \) is symmetric and \( G \) constant [8]. Furthermore,
\[
\lim_{k \to x} \| \hat{x}(k) - \tilde{x}(k) \|^2 = 0.
\]

\[
\lim_{k \to x} \| x(k) - \hat{x} \|^2 = \lim_{k \to x} \| x(k) - \tilde{x}(k) \|^2 = (P^{-1} + C^TC)^{-1} > 0,
\]
where
\[ \hat{x}(k+1) = \hat{x}(k+1|k) + G \left[ y(k+1) - \hat{C}_k \hat{x}(k+1|k) \right]. \]

4.2 CONVERGENCE OF PARAMETER IDENTIFICATION

The parameter identification algorithms for A, B, and C are convergent for any initial values, which can be proved according to the ODE method [11]. Here, we give the following explanation in brief.

From (29) and (30), considering that \( P_k \) is symmetric and positive definite, we have
\[ P_k - P_{k+1} = \frac{P_k \varphi_k^T \varphi_k P_k}{1 + \varphi_k P_k \varphi_k^T} \geq 0. \]

This means that the matrix sequence \( \{P_k\} \) is monotonously decreasing and bounded below. As a matter of fact, under some additional mild assumption, it can be proved that
\[ \lim_{k \to \infty} P_k = 0, \]
which leads to
\[ \lim_{k \to \infty} \left( \hat{\theta}_{k+1} - \hat{\theta}_k \right) = 0, \]
from (28), i.e., \( \hat{\theta}_k \) is convergent. In the same way, we can show the convergence of \( \{C_k\} \) in (37).

4.3 A GLOBAL CONDITION

Moreover, for the global stability and convergence of the relaxation algorithm in our paper, the continuity condition of the solution with respect to the parameters in state space equation (1) is needed. That is, if the change of parameter matrices A or B is small, then the change of \( x(k) \), said the solution of (1) is small too. Clearly, such a mild condition is always held by the linear difference equation like state space equation (1).

4.4 INITIAL VALUE PROBLEM

By using the first \((n+r)\) sampled data, we compute the initial values \( \hat{A}_0, \hat{B}_0, \) and \( \hat{C}_0 \) according to the first method (a classical method) men-
tioned in Section 1. In general, \((n + r)\) is not a very large number, which means the computation task for solving the initial values should be bearable.

We give the major points below [4].

(a) Identifying the input-output model corresponding to the considered system described by (1) and (2) gives transfer function matrix

\[
H(z) = \{h_{ij}(z)\}_{m \times r} = C[zI - A]^{-1}B.
\]

(b) Expand \(h_{ij}(z)\) into Markov parameter sequence

\[
h_{ij} = h_{ij}(1)z^{-1} + h_{ij}(2)z^{-2} + \cdots + h_{ij}(l)z^{-l} + \cdots,
\]

\(i = 1, 2, \ldots, m; \quad j = 1, 2, \ldots, r.\)

(c) Let \(\alpha_i\) be the degree of the minimum common denominator of the \(i\)th row and \(\beta_j\) the one of the \(j\)th column in matrix \(H(z)\), and set

\[
P_i = \alpha_i + 1, \quad q_j = \beta_j.
\]

Then, construct the Hankel matrix as follows:

\[
H = \begin{bmatrix}
H_{11}(p_1, q_1) & H_{12}(p_1, q_2) & \cdots & H_{1r}(p_1, q_r) \\
H_{21}(p_2, q_1) & H_{22}(p_2, q_2) & \cdots & H_{2r}(p_2, q_r) \\
\vdots & \vdots & \ddots & \vdots \\
H_{m1}(p_m, q_1) & H_{m2}(p_m, q_2) & \cdots & H_{mr}(p_m, q_r)
\end{bmatrix},
\]

where

\[
H_{ij}(p_i, q_j) = \begin{bmatrix}
h_{ij}(1) & h_{ij}(2) & \cdots & h_{ij}(q_j) \\
h_{ij}(2) & h_{ij}(3) & \cdots & h_{ij}(q_j + 1) \\
\vdots & \vdots & \ddots & \vdots \\
h_{ij}(p_i) & h_{ij}(p_i + 1) & \cdots & h_{ij}(p_i + q_j - 1)
\end{bmatrix}.
\]

(d) By means of elementary transformation, transform the Hankel matrix \(H\) into a Hermite canonical form with unit row vectors and take
away the possibly appeared zero-column(s) and the linear dependent row(s) (only one row among the linear dependent rows should be left) from the transformed matrix. The last obtained matrix is denoted as $\mathbf{S}$, which has the following form:

$$
\mathbf{S} = \begin{bmatrix}
S_{11} & \cdots & S_{1r} \\
S_{21} & \cdots & S_{2r} \\
\vdots & \ddots & \vdots \\
S_{m1} & \cdots & S_{mr}
\end{bmatrix},
$$

where

$$
S_{ii} = \begin{bmatrix}
* & * & \cdots & * \\
I_{a_i} & * & \cdots & * \\
\end{bmatrix}, \quad S_{ij} = \begin{bmatrix}
* & * & 0 & \cdots & * \\
\end{bmatrix} \text{ for } i \neq j.
$$

(e) Taking out the first row vector of every rowblock in matrix $\mathbf{S}$, in order, form initial guess $\hat{\mathbf{C}}_0$, and let the remainder form $\hat{\mathbf{A}}_0$. As for parameter $\mathbf{B}$, we take out the first $a_i$ entries of the first column vector of every block $H_{ij}(p_i, q_j)$ in matrix $\mathbf{H}$, in regular sequence, to construct the initial guess $\hat{\mathbf{B}}_0$.

Let $\mathbf{P}_0 = \alpha^2 \mathbf{I}$ with $10 < \alpha < 10^2$, and start the RRI algorithm from $k = 1$ patterned in Fig. 1.

On the other hand, considering the stability of Kalman filtering, the initial values $\hat{\theta}_0$ (i.e., $\hat{\mathbf{A}}_0$ and $\hat{\mathbf{B}}_0$) and $\hat{\mathbf{C}}_0$ are needed to ensure the controllability and observability of the considered linear system. That is,

$$
\mathbf{Q}_{CA} = \begin{bmatrix}
\hat{\mathbf{C}}_0 \\
\hat{\mathbf{C}}_0 \hat{\mathbf{A}}_0 \\
\vdots \\
\hat{\mathbf{C}}_0 \hat{\mathbf{A}}_0^{n-1}
\end{bmatrix} \quad \text{and} \quad \mathbf{Q}_{AB} = \begin{bmatrix}
\hat{\mathbf{B}}_0 \\
\hat{\mathbf{A}}_0 \hat{\mathbf{B}}_0 \\
\vdots \\
\hat{\mathbf{A}}_0^{n-1} \hat{\mathbf{B}}_0
\end{bmatrix}
$$

should be of full rank. Therefore, some modification may be needed for the initial values if the obtained initial estimates do not hold the previous condition.

5. CONCLUSION AND REMARKS

A new identification method for discrete-time multivariable systems, called recursive relaxation identification, is proposed, and a set of parallel
algorithm schemes are given and discussed. The proposed RRI with its parallel algorithm only bases upon the linear system model described by (1) and (2). No nonlinear problem is necessary to deal with, which is desirable in applications. Furthermore, the RRI with its parallel algorithms can be extended to dealing with the time-variant linear multivariable systems, especially to dealing with the real-time on-line identification and adaptive prediction for this kind of systems.

REFERENCES


Received 1 March 1994; revised 1 July 1995