Circuits Syst Signal Process (2016) 35:3323–3338
DOI 10.1007/s00034-015-0190-6

SHORT PAPER

Recursive Least Squares Parameter Estimation for a Class of Output Nonlinear Systems Based on the Model Decomposition

Feng Ding[1] · Xuehai Wang[1] · Qijia Chen[1] · Yongsong Xiao[1]

Received: 25 June 2014 / Revised: 23 October 2015 / Accepted: 24 October 2015 /
Published online: 17 November 2015

Abstract In this paper, we study the parameter estimation problem of a class of output nonlinear systems and propose a recursive least squares (RLS) algorithm for estimating the parameters of the nonlinear systems based on the model decomposition. The proposed algorithm has lower computational cost than the existing over-parameterization model-based RLS algorithm. The simulation results indicate that the proposed algorithm can effectively estimate the parameters of the nonlinear systems.

Keywords Parameter estimation · Recursive identification · Least squares · Hierarchical identification principle · Nonlinear system

1 Introduction

Iterative methods and recursive methods have widely been used in system identification [7,8,15], system control [31,33], signal processing [30] and multivariate pseudo-linear regressive analysis [37] and for solving matrix equations [6]. The parameter estimation methods have received much attention in system identification [28,38,39]. For

✉ Feng Ding
fding@jiangnan.edu.cn

Xuehai Wang
xuehaiwang735@163.com

Qijia Chen
qjchen11@163.com

Yongsong Xiao
ys_xiao82@163.com

[1] Key Laboratory of Advanced Process Control for Light Industry (Ministry of Education),
Jiangnan University, Wuxi 214122, People's Republic of China

example, Wang [35] gave a least squares-based recursive estimation algorithm and an iterative estimation algorithm for output error moving average systems using the data filtering technique. Dehghan et al. [5] studied the fourth-order variants of the Newton's method without the second-order derivatives for solving nonlinear equations. Shi et al. [32] studied the output feedback stabilization of networked control systems with random delays modeled by the Markov chains. Li [25] developed a maximum likelihood estimation algorithm for Hammerstein CARARMA systems based on the Newton iteration. Wang and Zhang [43] proposed an improved least squares identification algorithm for multivariable Hammerstein systems.

The typical nonlinear systems include the Wiener systems [14], the Hammerstein systems [18], the Hammerstein–Wiener systems [3] and feedback nonlinear systems [1,20]. A Wiener nonlinear model is composed of a linear dynamic block followed by a static nonlinear function and a Hammerstein model puts a nonlinear function before a linear dynamic block [2,44]. Vörös [34] proposed the key term separation technique for identifying Hammerstein systems with multi-segment piecewise-linear characteristics. Recently, a decomposition-based Newton iterative identification method was proposed for a Hammerstein nonlinear FIR system with ARMA noise [12].

This paper considers the parameter identification problem of a special class of output nonlinear systems, whose output is the nonlinear function of the past outputs [7,8]. For this class of nonlinear systems with colored noise, Wang et al. [36] gave a least squares-based and a gradient-based iterative identification algorithms; Hu et al. [21] proposed a recursive extended least squares parameter estimation algorithm using the over-parameterization model and a multi-innovation generalized extended stochastic gradient algorithm for nonlinear autoregressive moving average systems [22]; Bai presented an optimal two-stage identification algorithm for Hammerstein–Wiener nonlinear systems [4]. The least squares algorithms play a key role in the parameter estimation of linear systems [17,19,29]. This paper derives a new recursive least squares algorithm for output nonlinear systems using the hierarchical identification principle. The proposed method has less computational load and can be extended to study parameter estimation of dual-rate/multi-rate sampled systems [9,10,26].

This paper is organized as follows: Section 2 gives the representation of a class of nonlinear systems. Sections 3 and 4 derive a least squares algorithm and a model decomposition-based recursive least squares algorithm. Section 5 gives the computational efficiency of the proposed algorithm and the recursive extended least squares algorithm. Section 6 provides a numerical example to show the effectiveness of the proposed algorithm. Finally, some concluding remarks are offered in Sect. 7.

2 The System Description and its Identification Model

Let us define some notation. "$A =: X$"or "$X := A$" stands for "A is defined as X"; $\mathbf{1}_n$ denotes an n-dimensional column vector whose elements are all 1; $\boldsymbol{I}$ $(\boldsymbol{I}_n)$ represents an identity matrix of appropriate sizes $(n \times n)$; z denotes a unit forward shift operator with $zx(t) = x(t+1)$ and $z^{-1}x(t) = x(t-1)$. Define the polynomials in the unit backward shift operator z^{-1}:

$$\begin{aligned} A'(z) &:= a'_1 z^{-1} + a'_2 z^{-2} + \cdots + a'_{n_a} z^{-n_a}, \\ A(z) &:= a_1 z^{-1} + a_2 z^{-2} + \cdots + a_{n_a} z^{-n_a}, \\ B(z) &:= b_1 z^{-1} + b_2 z^{-2} + \cdots + b_{n_b} z^{-n_b}, \\ D(z) &:= 1 + d_1 z^{-1} + d_2 z^{-2} + \cdots + d_{n_d} z^{-n_d}, \end{aligned}$$

and the parameter vectors:

$$\begin{aligned} \boldsymbol{a} &:= [a_1, a_2, \ldots, a_{n_a}]^{\mathrm{T}} \in \mathbb{R}^{n_a}, \\ \boldsymbol{b} &:= [b_1, b_2, \ldots, b_{n_b}]^{\mathrm{T}} \in \mathbb{R}^{n_b}, \\ \boldsymbol{c} &:= [c_1, c_2, \ldots, c_{n_c}]^{\mathrm{T}} \in \mathbb{R}^{n_c}, \\ \boldsymbol{d} &:= [d_1, d_2, \ldots, d_{n_d}]^{\mathrm{T}} \in \mathbb{R}^{n_d}. \end{aligned}$$

A Hammerstein system (i.e., an input nonlinear system) can be expressed as [11]

$$y(t) = A'(z)y(t) + B(z)f(u(t)) + D(z)v(t),$$

by extending the input nonlinearity to the output nonlinearity, we can obtain a special class of nonlinear systems [21,36]:

$$y(t) = A(z)f(y(t)) + B(z)u(t) + D(z)v(t), \tag{1}$$

where $u(t)$ and $y(t)$ are the input and output of the system, respectively, and $v(t)$ is white noise with zero mean and variance σ^2.

For simplicity, assume that the nonlinear part is a linear combination of a known basis $\boldsymbol{f} := (f_1, f_2, \ldots, f_{n_c})$ with coefficients $(c_1, c_2, \ldots, c_{n_c})$:

$$\bar{y}(t) := f(y(t)) = c_1 f_1(y(t)) + c_2 f_2(y(t)) + \cdots + c_{n_c} f_{n_c}(y(t)) = \boldsymbol{f}(y(t))\boldsymbol{c}.$$

For the parameter identifiability, we must fix one of the coefficients c_i's, or $\|\boldsymbol{c}\| = 1$ with $c_1 > 0$ [11].

Equation (1) can be rewritten as

$$\begin{aligned} y(t) &= \sum_{i=1}^{n_a} a_i z^{-i} f(y(t)) + \sum_{i=1}^{n_b} b_i z^{-i} u(t) + \sum_{i=1}^{n_d} d_i z^{-i} v(t) + v(t) \\ &= a_1 f(y(t-1)) + a_2 f(y(t-2)) + \cdots + a_{n_a} f(y(t-n_a)) \\ &\quad + b_1 u(t-1) + b_2 u(t-2) + \cdots + b_{n_b} u(t-n_b) \\ &\quad + d_1 v(t-1) + d_2 v(t-2) + \cdots + d_{n_d} v(t-n_d) + v(t). \end{aligned} \tag{2}$$

Define the information matrix $\boldsymbol{F}(t)$, the input information vector $\boldsymbol{\varphi}(t)$ and the noise information vector $\boldsymbol{\psi}(t)$ as

$$\boldsymbol{F}(t) := \begin{bmatrix} f_1(y(t-1)) & f_2(y(t-1)) & \dots & f_{n_c}(y(t-1)) \\ f_1(y(t-2)) & f_2(y(t-2)) & \dots & f_{n_c}(y(t-2)) \\ \vdots & \vdots & & \vdots \\ f_1(y(t-n_a)) & f_2(y(t-n_a)) & \dots & f_{n_c}(y(t-n_a)) \end{bmatrix} \in \mathbb{R}^{n_a \times n_c}, \tag{3}$$

$$\boldsymbol{\varphi}(t) := [u(t-1), u(t-2), \dots, u(t-n_b)]^{\mathrm{T}} \in \mathbb{R}^{n_b}, \tag{4}$$

$$\boldsymbol{\psi}(t) := [v(t-1), v(t-2), \dots, v(t-n_d)]^{\mathrm{T}} \in \mathbb{R}^{n_d}. \tag{5}$$

Then, Eq. (2) can be written as

$$y(t) = \boldsymbol{a}^{\mathrm{T}}\boldsymbol{F}(t)\boldsymbol{c} + \boldsymbol{\varphi}^{\mathrm{T}}(t)\boldsymbol{b} + \boldsymbol{\psi}^{\mathrm{T}}(t)\boldsymbol{d} + v(t). \tag{6}$$

The objective of identification is to present new methods for estimating the unknown parameter vector $\boldsymbol{c}$ for the nonlinear part and the unknown parameter vectors $\boldsymbol{a}$, $\boldsymbol{b}$ and $\boldsymbol{d}$ for the linear subsystems from the measurement data $\{u(t), y(t) : t = 1, 2, 3, \dots\}$.

3 The Least Squares Algorithm Based on the Model Decomposition

Let $\hat{\boldsymbol{\theta}}(t) := \begin{bmatrix} \hat{\boldsymbol{a}}(t) \\ \hat{\boldsymbol{d}}(t) \end{bmatrix}$ and $\hat{\boldsymbol{\vartheta}}(t) := \begin{bmatrix} \hat{\boldsymbol{b}}(t) \\ \hat{\boldsymbol{c}}(t) \end{bmatrix}$ denote the estimates of $\boldsymbol{\theta} := \begin{bmatrix} \boldsymbol{a} \\ \boldsymbol{d} \end{bmatrix}$ and $\boldsymbol{\vartheta} := \begin{bmatrix} \boldsymbol{b} \\ \boldsymbol{c} \end{bmatrix}$ at time t, respectively, and $\boldsymbol{\Theta} := \begin{bmatrix} \boldsymbol{\theta} \\ \boldsymbol{\vartheta} \end{bmatrix}$. For the identification model in (6), using the hierarchical identification principle (the decomposition technique), define the quadratic cost functions:

$$\begin{aligned} J_1(\boldsymbol{\theta}) &:= J(\boldsymbol{a}, \hat{\boldsymbol{b}}(t-1), \hat{\boldsymbol{c}}(t-1), \boldsymbol{d}) \\ &= \sum_{j=1}^{t} \left[y(j) - \boldsymbol{\varphi}^{\mathrm{T}}(j)\hat{\boldsymbol{b}}(t-1) - [\hat{\boldsymbol{c}}^{\mathrm{T}}(t-1)\boldsymbol{F}^{\mathrm{T}}(j), \boldsymbol{\psi}^{\mathrm{T}}(j)]\boldsymbol{\theta} \right]^2, \\ J_2(\boldsymbol{\vartheta}) &:= J(\hat{\boldsymbol{a}}(t), \boldsymbol{b}, \boldsymbol{c}, \hat{\boldsymbol{d}}(t)) \\ &= \sum_{j=1}^{t} \left[y(j) - \boldsymbol{\psi}^{\mathrm{T}}(j)\hat{\boldsymbol{d}}(t) - [\boldsymbol{\varphi}^{\mathrm{T}}(j), \hat{\boldsymbol{a}}^{\mathrm{T}}(t)\boldsymbol{F}(j)]\boldsymbol{\vartheta} \right]^2. \end{aligned}$$

Define the output vector $\boldsymbol{Y}_t$ and the information matrices $\boldsymbol{\Phi}_t$, $\boldsymbol{\Psi}_t$, $\boldsymbol{\Omega}_t$ and $\boldsymbol{\Xi}_t$ as

$$\boldsymbol{Y}_t := [y(1), y(2), \dots, y(t)]^{\mathrm{T}} \in \mathbb{R}^{t}, \tag{7}$$

$$\boldsymbol{\Phi}_t := [\boldsymbol{\varphi}(1), \boldsymbol{\varphi}(2), \dots, \boldsymbol{\varphi}(t)]^{\mathrm{T}} \in \mathbb{R}^{t \times n_b}, \tag{8}$$

$$\boldsymbol{\Psi}_t := [\boldsymbol{\psi}(1), \boldsymbol{\psi}(2), \dots, \boldsymbol{\psi}(t)]^{\mathrm{T}} \in \mathbb{R}^{t \times n_d}, \tag{9}$$

$$\hat{\boldsymbol{\Omega}}_t := \begin{bmatrix} \boldsymbol{F}(1)\hat{\boldsymbol{c}}(t-1) & \boldsymbol{F}(2)\hat{\boldsymbol{c}}(t-1) & \dots & \boldsymbol{F}(t)\hat{\boldsymbol{c}}(t-1) \\ \hat{\boldsymbol{\psi}}(1) & \hat{\boldsymbol{\psi}}(2) & \dots & \hat{\boldsymbol{\psi}}(t) \end{bmatrix}^{\mathrm{T}} \in \mathbb{R}^{t \times (n_a + n_d)}, \tag{10}$$

$$\hat{\boldsymbol{\Xi}}_t := \begin{bmatrix} \boldsymbol{\varphi}(1) & \boldsymbol{\varphi}(2) & \ldots & \boldsymbol{\varphi}(t) \\ \boldsymbol{F}^{\mathrm{T}}(1)\hat{\boldsymbol{a}}(t) & \boldsymbol{F}^{\mathrm{T}}(2)\hat{\boldsymbol{a}}(t) & \ldots & \boldsymbol{F}^{\mathrm{T}}(t)\hat{\boldsymbol{a}}(t) \end{bmatrix}^{\mathrm{T}} \in \mathbb{R}^{t\times(n_b+n_c)}. \tag{11}$$

Then, $J_1(\boldsymbol{\theta})$ and $J_2(\boldsymbol{\vartheta})$ can be equivalently written as

$$J_1(\boldsymbol{\theta}) = \|\boldsymbol{Y}_t - \boldsymbol{\Phi}_t\hat{\boldsymbol{b}}(t-1) - \hat{\boldsymbol{\Omega}}_t\boldsymbol{\theta}\|^2,$$
$$J_2(\boldsymbol{\vartheta}) = \|\boldsymbol{Y}_t - \boldsymbol{\Psi}_t\hat{\boldsymbol{d}}(t) - \hat{\boldsymbol{\Xi}}_t\boldsymbol{\vartheta}\|^2.$$

For the two optimization problems, letting the partial derivatives of $J_1(\boldsymbol{\theta})$ and $J_2(\boldsymbol{\vartheta})$ with respect to $\boldsymbol{\theta}$ and $\boldsymbol{\vartheta}$ be zero gives

$$\left.\frac{\partial J_1(\boldsymbol{\theta})}{\partial\boldsymbol{\theta}}\right|_{\boldsymbol{\theta}=\hat{\boldsymbol{\theta}}(t)} = -2\hat{\boldsymbol{\Omega}}_t^{\mathrm{T}}[\boldsymbol{Y}_t - \boldsymbol{\Phi}_t\hat{\boldsymbol{b}}(t-1) - \hat{\boldsymbol{\Omega}}_t\hat{\boldsymbol{\theta}}(t)] = \boldsymbol{0}, \tag{12}$$

$$\left.\frac{\partial J_2(\boldsymbol{\vartheta})}{\partial\boldsymbol{\vartheta}}\right|_{\boldsymbol{\vartheta}=\hat{\boldsymbol{\vartheta}}(t)} = -2\hat{\boldsymbol{\Xi}}_t^{\mathrm{T}}[\boldsymbol{Y}_t - \boldsymbol{\Psi}_t\hat{\boldsymbol{d}}(t) - \hat{\boldsymbol{\Xi}}_t\hat{\boldsymbol{\vartheta}}(t)] = \boldsymbol{0}. \tag{13}$$

or

$$\hat{\boldsymbol{\Omega}}_t^{\mathrm{T}}\hat{\boldsymbol{\Omega}}_t\hat{\boldsymbol{\theta}}(t) = \hat{\boldsymbol{\Omega}}_t^{\mathrm{T}}[\boldsymbol{Y}_t - \boldsymbol{\Phi}_t\hat{\boldsymbol{b}}(t-1)],$$
$$\hat{\boldsymbol{\Xi}}_t^{\mathrm{T}}\hat{\boldsymbol{\Xi}}_t\hat{\boldsymbol{\vartheta}}(t) = \hat{\boldsymbol{\Xi}}_t^{\mathrm{T}}[\boldsymbol{Y}_t - \boldsymbol{\Psi}_t\hat{\boldsymbol{d}}(t)].$$

In order to ensure that the inverses of the matrices $\hat{\boldsymbol{\Omega}}_t^{\mathrm{T}}\hat{\boldsymbol{\Omega}}_t$ and $\hat{\boldsymbol{\Xi}}_t^{\mathrm{T}}\hat{\boldsymbol{\Xi}}_t$ exist, we suppose that the information matrices $\hat{\boldsymbol{\Omega}}_t$ and $\hat{\boldsymbol{\Xi}}_t$ are persistently exciting. Let $\hat{\boldsymbol{\Psi}}_t$, $\hat{\boldsymbol{\psi}}(t)$ $\hat{v}(t)$ be the estimates of $\boldsymbol{\Psi}_t$, $\boldsymbol{\psi}(t)$ and $v(t)$ at time t.

Replacing the unknown $\boldsymbol{\Psi}_t$ in (12)–(13) with $\hat{\boldsymbol{\Psi}}_t$, we have the following least squares algorithm for estimating the parameter vectors $\boldsymbol{\theta}$ and $\boldsymbol{\vartheta}$:

$$\hat{\boldsymbol{\theta}}(t) = [\hat{\boldsymbol{\Omega}}_t^{\mathrm{T}}\hat{\boldsymbol{\Omega}}_t]^{-1}\hat{\boldsymbol{\Omega}}_t^{\mathrm{T}}[\boldsymbol{Y}_t - \boldsymbol{\Phi}_t\hat{\boldsymbol{b}}(t-1)], \tag{14}$$

$$\hat{\boldsymbol{\vartheta}}(t) = [\hat{\boldsymbol{\Xi}}_t^{\mathrm{T}}\hat{\boldsymbol{\Xi}}_t]^{-1}\hat{\boldsymbol{\Xi}}_t[\boldsymbol{Y}_t - \hat{\boldsymbol{\Psi}}_t\hat{\boldsymbol{d}}(t)], \tag{15}$$

$$\hat{\boldsymbol{\Psi}}_t = [\hat{\boldsymbol{\psi}}(1), \hat{\boldsymbol{\psi}}(2), \ldots, \hat{\boldsymbol{\psi}}(t)]^{\mathrm{T}}, \tag{16}$$

$$\hat{\boldsymbol{\psi}}(t) = [\hat{v}(t-1), \hat{v}(t-2), \ldots, \hat{v}(t-n_d)]^{\mathrm{T}}, \tag{17}$$

$$\hat{v}(t) = y(t) - \hat{\boldsymbol{a}}^{\mathrm{T}}(t)\boldsymbol{F}(t)\hat{\boldsymbol{c}}(t) - \hat{\boldsymbol{\varphi}}^{\mathrm{T}}(t)\hat{\boldsymbol{b}}(t) - \hat{\boldsymbol{\psi}}^{\mathrm{T}}(t)\hat{\boldsymbol{d}}(t). \tag{18}$$

The procedures of computing the parameter estimates $\hat{\boldsymbol{\theta}}(t)$ and $\hat{\boldsymbol{\vartheta}}(t)$ are listed in the following.

1. To initialize: let $t = p$, collect the input–output data $\{u(i), y(i) : i = 0, 1, 2, \ldots, p-1\}$ $(p \gg n_a + n_b + n_c + n_d)$ and set the initial values $\hat{\boldsymbol{b}}(p-1) = \boldsymbol{1}_{n_b}/p_0$, $\hat{\boldsymbol{c}}(p-1) =$ a random vector with $\|\hat{\boldsymbol{c}}(p-1)\| = 1$, $\hat{v}(i) =$ a random number. p_0 is normally a large positive number (e.g., $p_0 = 10^6$), give the basis function $f_j(*)$.
2. Collect the input–output data $u(t)$ and $y(t)$, form $\boldsymbol{Y}_t$ using (7), $\boldsymbol{F}(t)$ using (3), $\boldsymbol{\varphi}(t)$ using (4), $\boldsymbol{\Phi}_t$ using (8) and $\hat{\boldsymbol{\psi}}(t)$ using (17).

3. Compute $\hat{\boldsymbol{\Omega}}_t$ using (10).
4. Update the parameter estimate $\hat{\boldsymbol{\theta}}(t)$ using (14) and read $\hat{\boldsymbol{a}}(t)$ and $\hat{\boldsymbol{d}}(t)$ from $\hat{\boldsymbol{\theta}}(t) = \begin{bmatrix} \hat{\boldsymbol{a}}(t) \\ \hat{\boldsymbol{d}}(t) \end{bmatrix}$.
5. Compute $\hat{\boldsymbol{\Xi}}_t$ using (11) and form $\hat{\boldsymbol{\Psi}}_t$ using (16).
6. Update the parameter estimate $\hat{\boldsymbol{\vartheta}}(t)$ using (15), read $\hat{\boldsymbol{b}}(t)$ from $\hat{\boldsymbol{\vartheta}}(t) = \begin{bmatrix} \hat{\boldsymbol{b}}(t) \\ \hat{\boldsymbol{c}}(t) \end{bmatrix}$ and normalize $\hat{\boldsymbol{c}}(t)$ using

$$\hat{\boldsymbol{c}}(t) = \mathrm{sgn}\{[\hat{\boldsymbol{\vartheta}}(t)](n_b+1)\}\frac{[\hat{\boldsymbol{\vartheta}}(t)](n_b+1:n_b+n_c)}{\|[\hat{\boldsymbol{\vartheta}}(t)](n_b+1:n_b+n_c)\|}.$$

7. Compute $\hat{v}(t)$ using (18).
8. Increase t by 1, go to Step 2 and continue calculation.

4 The Recursive Least Squares Algorithm Based on the Model Decomposition

For the identification model in (6), define the quadratic cost functions:

$$J_3(\boldsymbol{\theta}) := \sum_{j=1}^{t}\left[y(j) - \boldsymbol{\varphi}^{\mathrm{T}}(j)\boldsymbol{b} - [\boldsymbol{c}^{\mathrm{T}}\boldsymbol{F}^{\mathrm{T}}(j), \boldsymbol{\psi}^{\mathrm{T}}(j)]\boldsymbol{\theta}\right]^2,$$

$$J_4(\boldsymbol{\vartheta}) := \sum_{j=1}^{t}\left[y(j) - \boldsymbol{\psi}^{\mathrm{T}}(j)\boldsymbol{d} - [\boldsymbol{\varphi}^{\mathrm{T}}(j), \boldsymbol{a}^{\mathrm{T}}\boldsymbol{F}(j)]\boldsymbol{\vartheta}\right]^2.$$

Define the information matrices $\boldsymbol{\varphi}_1(t)$, $\boldsymbol{\Omega}_t$ and $\boldsymbol{\Xi}_t$ as

$$\boldsymbol{\varphi}_1(t) := [\boldsymbol{c}^{\mathrm{T}}\boldsymbol{F}^{\mathrm{T}}(t), \boldsymbol{\psi}^{\mathrm{T}}(t)]^{\mathrm{T}} \in \mathbb{R}^{n_a+n_d}, \tag{19}$$

$$\boldsymbol{\Omega}_t := \begin{bmatrix} \boldsymbol{F}(1)\boldsymbol{c} & \boldsymbol{F}(2)\boldsymbol{c} & \ldots & \boldsymbol{F}(t)\boldsymbol{c} \\ \boldsymbol{\psi}(1) & \boldsymbol{\psi}(2) & \ldots & \boldsymbol{\psi}(t) \end{bmatrix}^{\mathrm{T}} \in \mathbb{R}^{t\times(n_a+n_d)}, \tag{20}$$

$$\boldsymbol{\Xi}_t := \begin{bmatrix} \boldsymbol{\varphi}(1) & \boldsymbol{\varphi}(2) & \ldots & \boldsymbol{\varphi}(t) \\ \boldsymbol{F}^{\mathrm{T}}(1)\boldsymbol{a} & \boldsymbol{F}^{\mathrm{T}}(2)\boldsymbol{a} & \ldots & \boldsymbol{F}^{\mathrm{T}}(t)\boldsymbol{a} \end{bmatrix}^{\mathrm{T}} \in \mathbb{R}^{t\times(n_b+n_c)}. \tag{21}$$

Then, $J_3(\boldsymbol{\theta})$ and $J_4(\boldsymbol{\vartheta})$ can be equivalently rewritten as

$$J_3(\boldsymbol{\theta}) = \|\boldsymbol{Y}_t - \boldsymbol{\Phi}_t\boldsymbol{b} - \boldsymbol{\Omega}_t\boldsymbol{\theta}\|^2,$$
$$J_4(\boldsymbol{\vartheta}) = \|\boldsymbol{Y}_t - \boldsymbol{\Psi}_t\boldsymbol{d} - \boldsymbol{\Xi}_t\boldsymbol{\vartheta}\|^2.$$

Similarly, minimizing $J_3(\boldsymbol{\theta})$ and $J_4(\boldsymbol{\vartheta})$, we can obtain the least squares estimates:

$$\hat{\boldsymbol{\theta}}(t) = [\boldsymbol{\Omega}_t^{\mathrm{T}}\boldsymbol{\Omega}_t]^{-1}\boldsymbol{\Omega}_t^{\mathrm{T}}[\boldsymbol{Y}_t - \boldsymbol{\Phi}_t\boldsymbol{b}], \tag{22}$$

$$\hat{\boldsymbol{\vartheta}}(t) = [\boldsymbol{\Xi}_t^{\mathrm{T}}\boldsymbol{\Xi}_t]^{-1}\boldsymbol{\Xi}_t^{\mathrm{T}}[\boldsymbol{Y}_t - \boldsymbol{\Psi}_t\boldsymbol{d}]. \tag{23}$$

Define the covariance matrix,

$$\begin{aligned}\boldsymbol{P}_1^{-1}(t) &: = \boldsymbol{\Omega}_t^{\mathrm{T}}\boldsymbol{\Omega}_t = \sum_{j=1}^{t}\boldsymbol{\varphi}_1(j)\boldsymbol{\varphi}_1^{\mathrm{T}}(j)\\ &= \boldsymbol{P}_1^{-1}(t-1) + \boldsymbol{\varphi}_1(t)\boldsymbol{\varphi}_1^{\mathrm{T}}(t) \in \mathbb{R}^{(n_a+n_d)\times(n_a+n_d)}, \quad \boldsymbol{P}(0) = \boldsymbol{I}_{n_a+n_d}. \end{aligned} \tag{24}$$

Hence, Eq. (22) can be rewritten as

$$\begin{aligned}\hat{\boldsymbol{\theta}}(t) &= \boldsymbol{P}_1(t)\boldsymbol{\Omega}_t^{\mathrm{T}}[\boldsymbol{Y}_t - \boldsymbol{\Phi}_t\boldsymbol{b}]\\ &= \boldsymbol{P}_1(t)[\boldsymbol{\Omega}_{t-1}^{\mathrm{T}}, \boldsymbol{\varphi}_1(t)]\begin{bmatrix}\boldsymbol{Y}_{t-1} - \boldsymbol{\Phi}_{t-1}\boldsymbol{b}\\ y(t) - \boldsymbol{\varphi}^{\mathrm{T}}(t)\boldsymbol{b}\end{bmatrix}\\ &= \boldsymbol{P}_1(t)\boldsymbol{P}_1^{-1}(t-1)\boldsymbol{P}_1(t-1)\left\{\boldsymbol{\Omega}_{t-1}^{\mathrm{T}}[\boldsymbol{Y}_{t-1} - \boldsymbol{\Phi}_{t-1}\boldsymbol{b}] + \boldsymbol{\varphi}_1(t)[y(t) - \boldsymbol{\varphi}^{\mathrm{T}}(t)\boldsymbol{b}]\right\}\\ &= \boldsymbol{P}_1(t)\boldsymbol{P}_1^{-1}(t-1)\hat{\boldsymbol{\theta}}(t-1) + \boldsymbol{P}_1(t)\boldsymbol{\varphi}_1(t)[y(t) - \boldsymbol{\varphi}^{\mathrm{T}}(t)\boldsymbol{b}]. \end{aligned} \tag{25}$$

Applying the matrix inversion lemma [7,27]

$$(\boldsymbol{A} + \boldsymbol{B}\boldsymbol{C})^{-1} = \boldsymbol{A}^{-1} - \boldsymbol{A}^{-1}\boldsymbol{B}(\boldsymbol{I} + \boldsymbol{C}\boldsymbol{A}^{-1}\boldsymbol{B})^{-1}\boldsymbol{C}\boldsymbol{A}^{-1}.$$

to (24) gives

$$\boldsymbol{P}_1(t) = \boldsymbol{P}_1(t-1) - \frac{\boldsymbol{P}_1(t-1)\boldsymbol{\varphi}_1(t)\boldsymbol{\varphi}_1^{\mathrm{T}}(t)\boldsymbol{P}_1(t-1)}{1 + \boldsymbol{\varphi}_1^{\mathrm{T}}(t)\boldsymbol{P}_1(t-1)\boldsymbol{\varphi}_1(t)}. \tag{26}$$

Pre-multiplying both sides of (24) by $\boldsymbol{P}_1(t)$ gives

$$\boldsymbol{I} = \boldsymbol{P}_1(t)\boldsymbol{P}_1^{-1}(t-1) + \boldsymbol{P}_1(t)\boldsymbol{\varphi}_1(t)\boldsymbol{\varphi}_1^{\mathrm{T}}(t). \tag{27}$$

Substituting (27) into (25) gives the recursive estimate of the parameter vector $\boldsymbol{\theta}$:

$$\begin{aligned}\hat{\boldsymbol{\theta}}(t) &= [\boldsymbol{I} - \boldsymbol{P}_1(t)\boldsymbol{\varphi}_1(t)\boldsymbol{\varphi}_1^{\mathrm{T}}(t)]\hat{\boldsymbol{\theta}}(t-1) + \boldsymbol{P}_1(t)\boldsymbol{\varphi}_1(t)[y(t) - \boldsymbol{\varphi}^{\mathrm{T}}(t)\boldsymbol{b}]\\ &= \hat{\boldsymbol{\theta}}(t-1) + \boldsymbol{P}_1(t)\boldsymbol{\varphi}_1(t)[y(t) - \boldsymbol{\varphi}^{\mathrm{T}}(t)\boldsymbol{b} - \boldsymbol{\varphi}_1^{\mathrm{T}}(t)\hat{\boldsymbol{\theta}}(t-1)]. \end{aligned} \tag{28}$$

Define the gain vector $\boldsymbol{L}_1(t) := \boldsymbol{P}_1(t)\boldsymbol{\varphi}_1(t) \in \mathbb{R}^{n_a+n_d}$. Using (26), it follows that

$$\boldsymbol{L}_1(t) = \frac{\boldsymbol{P}_1(t-1)\boldsymbol{\varphi}_1(t)}{1 + \boldsymbol{\varphi}_1^{\mathrm{T}}(t)\boldsymbol{P}_1(t-1)\boldsymbol{\varphi}_1(t)}. \tag{29}$$

Using (29), Eq. (26) can be rewritten as

$$\boldsymbol{P}_1(t) = \boldsymbol{P}_1(t-1) - \boldsymbol{L}_1(t)\boldsymbol{\varphi}_1^{\mathrm{T}}(t)\boldsymbol{P}_1(t-1) = [\boldsymbol{I} - \boldsymbol{L}_1(t)\boldsymbol{\varphi}_1^{\mathrm{T}}(t)]\boldsymbol{P}_1(t-1), \ \boldsymbol{P}_1(0) = p_0\boldsymbol{I}. \tag{30}$$

Here, we can see that the right-hand sides of (19) and (28) contain the unknown parameter vectors $\boldsymbol{c}$ and $\boldsymbol{b}$, respectively. The solution is to replace the unknown $\boldsymbol{b}$

and $\boldsymbol{\varphi}_1(t)$ in (28) and (30) with their corresponding estimates $\hat{\boldsymbol{b}}(t-1)$ and $\hat{\boldsymbol{\varphi}}_1(t) = [\hat{\boldsymbol{c}}^{\mathrm{T}}(t-1)\boldsymbol{F}^{\mathrm{T}}(t), \hat{\boldsymbol{\psi}}^{\mathrm{T}}(t)]^{\mathrm{T}}$, we have

$$\hat{\boldsymbol{\theta}}(t) = \hat{\boldsymbol{\theta}}(t-1) + \boldsymbol{P}_1(t)\boldsymbol{\varphi}_1(t)[y(t) - \boldsymbol{\varphi}^{\mathrm{T}}(t)\hat{\boldsymbol{b}}(t-1) - \hat{\boldsymbol{\varphi}}_1^{\mathrm{T}}(t)\hat{\boldsymbol{\theta}}(t-1)],$$

$$\boldsymbol{P}_1(t) = [\boldsymbol{I}_{n_a+n_d} - \boldsymbol{L}_1(t)\hat{\boldsymbol{\varphi}}_1^{\mathrm{T}}(t)]\boldsymbol{P}_1(t-1),\ \boldsymbol{P}_1(0) = p_0\boldsymbol{I}_{n_a+n_d}, \tag{31}$$

$$\hat{\boldsymbol{\psi}}(t) = [\hat{v}(t-1), \hat{v}(t-2), \ldots, \hat{v}(t-n_d)]^{\mathrm{T}} \in \mathbb{R}^{n_d}. \tag{32}$$

Define the information vector $\boldsymbol{\varphi}_2(t) := [\boldsymbol{\varphi}^{\mathrm{T}}(t), \boldsymbol{a}^{\mathrm{T}}\boldsymbol{F}(t)]^{\mathrm{T}} \in \mathbb{R}^{n_b+n_c}$ and the covariance matrix $\boldsymbol{P}_2^{-1}(t) := \boldsymbol{\Xi}_t^{\mathrm{T}}\boldsymbol{\Xi}_t \in \mathbb{R}^{(n_b+n_c)\times(n_b+n_c)}$ and the gain vector $\boldsymbol{L}_2(t) := \boldsymbol{P}_2(t)\boldsymbol{\varphi}_2(t) \in \mathbb{R}^{n_b+n_c}$. Similarly, some unknown terms are replaced with their estimates; we can obtain the recursive estimate of the parameter vector $\boldsymbol{\vartheta}$:

$$\hat{\boldsymbol{\vartheta}}(t) = \hat{\boldsymbol{\vartheta}}(t-1) + \boldsymbol{P}_2(t)\hat{\boldsymbol{\varphi}}_2^{\mathrm{T}}(t)[y(t) - \hat{\boldsymbol{\psi}}^{\mathrm{T}}(t)\hat{\boldsymbol{d}}(t) - \hat{\boldsymbol{\varphi}}_2^{\mathrm{T}}(t)\hat{\boldsymbol{\vartheta}}(t-1)], \tag{33}$$

$$\hat{\boldsymbol{\varphi}}_2(t) = [\boldsymbol{\varphi}^{\mathrm{T}}(t), \hat{\boldsymbol{a}}^{\mathrm{T}}(t)\boldsymbol{F}(t)] \in \mathbb{R}^{n_b+n_c}. \tag{34}$$

Thus, we can summarize the recursive least squares algorithm for estimating the parameter vectors $\boldsymbol{\theta}$ and $\boldsymbol{\vartheta}$ of the nonlinear systems based on the model decomposition (the ON-RLS algorithm for short) as follows:

$$\hat{\boldsymbol{\theta}}(t) = \hat{\boldsymbol{\theta}}(t-1) + \boldsymbol{L}_1(t)[y(t) - \boldsymbol{\varphi}^{\mathrm{T}}(t)\hat{\boldsymbol{b}}(t-1) - \hat{\boldsymbol{\varphi}}_1^{\mathrm{T}}(t)\hat{\boldsymbol{\theta}}(t-1)], \tag{35}$$

$$\boldsymbol{L}_1(t) = \boldsymbol{P}_1(t-1)\hat{\boldsymbol{\varphi}}_1(t)[1 + \hat{\boldsymbol{\varphi}}_1^{\mathrm{T}}(t)\boldsymbol{P}_1(t-1)\hat{\boldsymbol{\varphi}}_1(t)]^{-1}, \tag{36}$$

$$\boldsymbol{P}_1(t) = [\boldsymbol{I}_{n_a+n_d} - \boldsymbol{L}_1(t)\hat{\boldsymbol{\varphi}}_1^{\mathrm{T}}(t)]\boldsymbol{P}_1(t-1),\ \boldsymbol{P}_1(0) = p_0\boldsymbol{I}_{n_a+n_d}, \tag{37}$$

$$\hat{\boldsymbol{\vartheta}}(t) = \hat{\boldsymbol{\vartheta}}(t-1) + \boldsymbol{L}_2(t)[y(t) - \hat{\boldsymbol{\psi}}^{\mathrm{T}}(t)\hat{\boldsymbol{d}}(t) - \hat{\boldsymbol{\varphi}}_2^{\mathrm{T}}(t)\hat{\boldsymbol{\vartheta}}(t-1)], \tag{38}$$

$$\boldsymbol{L}_2(t) = \boldsymbol{P}_2(t-1)\hat{\boldsymbol{\varphi}}_2(t)[1 + \hat{\boldsymbol{\varphi}}_2^{\mathrm{T}}(t)\boldsymbol{P}_2(t-1)\hat{\boldsymbol{\varphi}}_2(t)]^{-1}, \tag{39}$$

$$\boldsymbol{P}_2(t) = [\boldsymbol{I}_{n_b+n_c} - \boldsymbol{L}_2(t)\hat{\boldsymbol{\varphi}}_2^{\mathrm{T}}(t)]\boldsymbol{P}_2(t-1),\ \boldsymbol{P}_2(0) = p_0\boldsymbol{I}_{n_b+n_c}, \tag{40}$$

$$\hat{\boldsymbol{\varphi}}_1(t) = [\hat{\boldsymbol{c}}^{\mathrm{T}}(t-1)\boldsymbol{F}^{\mathrm{T}}(t), \boldsymbol{\psi}^{\mathrm{T}}(t)]^{\mathrm{T}}, \tag{41}$$

$$\hat{\boldsymbol{\varphi}}_2(t) = [\boldsymbol{\varphi}^{\mathrm{T}}(t), \hat{\boldsymbol{a}}^{\mathrm{T}}(t)\boldsymbol{F}(t)]^{\mathrm{T}}, \tag{42}$$

$$\boldsymbol{\varphi}(t) = [u(t-1), u(t-2), \ldots, u(t-n_b)]^{\mathrm{T}}, \tag{43}$$

$$\hat{\boldsymbol{\psi}}(t) = [\hat{v}(t-1), \hat{v}(t-2), \ldots, \hat{v}(t-n_d)]^{\mathrm{T}}, \tag{44}$$

$$\hat{v}(t) = y(t) - \hat{\boldsymbol{a}}^{\mathrm{T}}(t)\boldsymbol{F}(t)\hat{\boldsymbol{c}}(t) - \hat{\boldsymbol{\varphi}}^{\mathrm{T}}(t)\hat{\boldsymbol{b}}(t) - \hat{\boldsymbol{\psi}}^{\mathrm{T}}(t)\hat{\boldsymbol{d}}(t), \tag{45}$$

$$\boldsymbol{F}(t) = \begin{bmatrix} f_1(y(t-1)) & f_2(y(t-1)) & \ldots & f_{n_c}(y(t-1)) \\ f_1(y(t-2)) & f_2(y(t-2)) & \ldots & f_{n_c}(y(t-2)) \\ \vdots & \vdots & & \vdots \\ f_1(y(t-n_a)) & f_2(y(t-n_a)) & \ldots & f_{n_c}(y(t-n_a)) \end{bmatrix}, \tag{46}$$

$$\hat{\boldsymbol{\Theta}}(t) = \begin{bmatrix} \hat{\boldsymbol{\theta}}(t) \\ \hat{\boldsymbol{\vartheta}}(t) \end{bmatrix}. \tag{47}$$

The procedures of computing the parameter estimation vectors $\hat{\boldsymbol{\theta}}(t)$ and $\hat{\boldsymbol{\vartheta}}(t)$ in (35)–(47) are listed in the following.

1. To initialize: let $t = 1$, and set the initial values $\boldsymbol{P}_1(0) = p_0\boldsymbol{I}_{n_a+n_d}$, $\hat{\boldsymbol{\theta}}(0) = \mathbf{1}_{n_a+n_d}/p_0$, $\hat{\boldsymbol{b}}(0) = \mathbf{1}_{n_b}/p_0$, $\boldsymbol{P}_2(0) = p_0\boldsymbol{I}_{n_b+n_c}$, $p_0 = 10^6$, $\|\hat{\boldsymbol{c}}(0)\| = 1$, give the basis function $f_j(*)$.
2. Collect the input–output data $u(t)$ and $y(t)$, form $\boldsymbol{F}(t)$ using (46), $\boldsymbol{\varphi}(t)$ using (43), $\hat{\boldsymbol{\psi}}(t)$ using (44) and compute $\hat{\boldsymbol{\varphi}}_1(t)$ in (41).
3. Compute $\boldsymbol{L}_1(t)$ using (36) and $\boldsymbol{P}_1(t)$ using (37).
4. Update the parameter estimate $\hat{\boldsymbol{\theta}}(t)$ using (35) and read $\hat{\boldsymbol{a}}(t)$ and $\hat{\boldsymbol{d}}(t)$ from $\hat{\boldsymbol{\theta}}(t) = \begin{bmatrix} \hat{\boldsymbol{a}}(t) \\ \hat{\boldsymbol{d}}(t) \end{bmatrix}$.
5. Form $\hat{\boldsymbol{\varphi}}_2(t)$ in (42) and compute $\boldsymbol{L}_2(t)$ using (39) and $\boldsymbol{P}_2(t)$ using (40).
6. Update the parameter estimate $\hat{\boldsymbol{\vartheta}}(t)$ using (38) and read $\hat{\boldsymbol{b}}(t)$ from $\hat{\boldsymbol{\vartheta}}(t) = \begin{bmatrix} \hat{\boldsymbol{b}}(t) \\ \hat{\boldsymbol{c}}(t) \end{bmatrix}$ and normalize $\hat{\boldsymbol{c}}(t)$ using

$$\hat{\boldsymbol{c}}(t) = \text{sgn}\{[\hat{\boldsymbol{\vartheta}}(t)\}(n_b+1)\}\frac{[\hat{\boldsymbol{\vartheta}}(t)](n_b+1:n_b+n_c)}{\|[\hat{\boldsymbol{\vartheta}}(t)](n_b+1:n_b+n_c)\|}, \tag{48}$$

   and let $\hat{\boldsymbol{\vartheta}}(t) = \begin{bmatrix} \hat{\boldsymbol{b}}(t) \\ \hat{\boldsymbol{c}}(t) \end{bmatrix}$.
7. Compute $\hat{v}(t)$ using (45).
8. Increase t by 1, go to Step 2 and continue the recursive calculation.

The flowchart for computing the estimates $\hat{\boldsymbol{\theta}}(t)$ and $\hat{\boldsymbol{\vartheta}}(t)$ in (35)–(47) is shown in Fig. 1.

To show the advantages of the proposed ON-RLS algorithm, the following gives the stochastic gradient algorithm with a forgetting factor λ for estimating the parameter vectors $\boldsymbol{\theta}$ and $\boldsymbol{\vartheta}$ of the nonlinear systems (the ON-SG algorithm for short) [11]:

$$\hat{\boldsymbol{\theta}}(t) = \hat{\boldsymbol{\theta}}(t-1) + \frac{\hat{\boldsymbol{\varphi}}_1(t)}{r_1(t)}[y(t) - \boldsymbol{\varphi}^{\mathrm{T}}(t)\hat{\boldsymbol{b}}(t-1) - \hat{\boldsymbol{\varphi}}_1^{\mathrm{T}}(t)\hat{\boldsymbol{\theta}}(t-1)], \tag{49}$$

$$r_1(t) = \lambda r_1(t) + \|\hat{\boldsymbol{\varphi}}_1(t)\|^2,\ 0 \leqslant \lambda \leqslant 1,\ r_1(0) = 1, \tag{50}$$

$$\hat{\boldsymbol{\vartheta}}(t) = \hat{\boldsymbol{\vartheta}}(t-1) + \frac{\hat{\boldsymbol{\varphi}}_2(t)}{r_2(t)}[y(t) - \hat{\boldsymbol{\psi}}^{\mathrm{T}}(t)\hat{\boldsymbol{d}}(t) - \hat{\boldsymbol{\varphi}}_2^{\mathrm{T}}(t)\hat{\boldsymbol{\vartheta}}(t-1)], \tag{51}$$

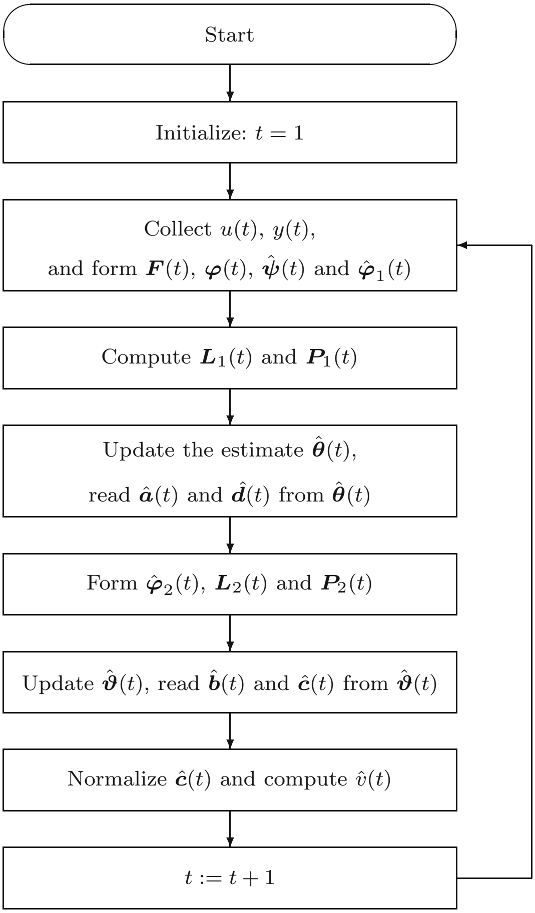
$$r_2(t) = \lambda r_2(t) + \|\hat{\boldsymbol{\varphi}}_2(t)\|^2,\ r_2(0) = 1. \tag{52}$$

Remark 1 The ON-RLS algorithm in (35)–(47) has faster convergence rates than the ON-SG algorithm in (49)–(52)—see the last columns in Tables 3 and 4.

5 The Comparison of the Computational Efficiency

In order to show the advantage of the ON-RLS algorithm, the following gives simply recursive extended least squares algorithm in [21] for comparison.

Fig. 1 The flowchart of computing the parameter estimates $\hat{\boldsymbol{\theta}}(t)$ and $\hat{\boldsymbol{\vartheta}}(t)$


Start
Initialize: $t = 1$
Collect $u(t)$, $y(t)$,
and form $\boldsymbol{F}(t)$, $\boldsymbol{\varphi}(t)$, $\hat{\boldsymbol{\psi}}(t)$ and $\hat{\boldsymbol{\varphi}}_1(t)$
Compute $\boldsymbol{L}_1(t)$ and $\boldsymbol{P}_1(t)$
Update the estimate $\hat{\boldsymbol{\theta}}(t)$,
read $\hat{\boldsymbol{a}}(t)$ and $\hat{\boldsymbol{d}}(t)$ from $\hat{\boldsymbol{\theta}}(t)$
Form $\hat{\boldsymbol{\varphi}}_2(t)$, $\boldsymbol{L}_2(t)$ and $\boldsymbol{P}_2(t)$
Update $\hat{\boldsymbol{\vartheta}}(t)$, read $\hat{\boldsymbol{b}}(t)$ and $\hat{\boldsymbol{c}}(t)$ from $\hat{\boldsymbol{\vartheta}}(t)$
Normalize $\hat{\boldsymbol{c}}(t)$ and compute $\hat{v}(t)$
$t := t + 1$

Define the parameter vectors,

$$\boldsymbol{\vartheta} := \begin{bmatrix} \boldsymbol{a} \otimes \boldsymbol{c} \\ \boldsymbol{b} \\ \boldsymbol{d} \end{bmatrix} \in \mathbb{R}^n, \quad n := n_a n_c + n_b + n_d.$$

Then, we have the following recursive extended least squares (RELS) algorithm [21]:

$$\begin{aligned}
\hat{\boldsymbol{\vartheta}}(t) &= \hat{\boldsymbol{\vartheta}}(t-1) + \boldsymbol{L}(t)[y(t) - \hat{\boldsymbol{\varphi}}^{\mathrm{T}}(t)\hat{\boldsymbol{\vartheta}}(t-1)],\ \hat{\boldsymbol{\vartheta}}(0) = \mathbf{1}_n/p_0, \\
\boldsymbol{L}(t) &= \boldsymbol{P}(t-1)\hat{\boldsymbol{\varphi}}(t)[1 + \hat{\boldsymbol{\varphi}}^{\mathrm{T}}(t)\boldsymbol{P}(t-1)\hat{\boldsymbol{\varphi}}(t)]^{-1}, \\
\boldsymbol{P}(t) &= [\boldsymbol{I}_n - \boldsymbol{L}(t)\hat{\boldsymbol{\varphi}}^{\mathrm{T}}(t)]\boldsymbol{P}(t-1),\ \boldsymbol{P}(0) = p_0\boldsymbol{I}_n, \\
\hat{\boldsymbol{\varphi}}(t) &= [\boldsymbol{h}^{\mathrm{T}}(y(t-1)), \ldots, \boldsymbol{h}^{\mathrm{T}}(y(t-n_a)), \\
&\qquad u(t-1), \ldots, u(t-n_b), \hat{v}(t-1), \ldots, \hat{v}(t-n_d)]^{\mathrm{T}}, \\
\boldsymbol{h}(t) &= [h_1(y(t)), h_2(y(t)), \ldots, h_{n_c}(y(t))]^{\mathrm{T}} \in \mathbb{R}^{n_c},
\end{aligned}$$

$$\hat{v}(t) = y(t) - \hat{\boldsymbol{\varphi}}^{\mathrm{T}}(t)\hat{\boldsymbol{\vartheta}}(t).$$

Remark 2 Compared with the recursive extended least squares algorithm, which involves the covariance matrix $\boldsymbol{P}(t)$ of large size $(n_an_c+n_b+n_d)\times(n_an_c+n_b+n_d)$, the ON-RLS algorithm has less computational load because it involves two covariance matrices $\boldsymbol{P}_1(t)$ and $\boldsymbol{P}_2(t)$ of small sizes $(n_a+n_d)\times(n_a+n_d)$ and $(n_b+n_c)\times(n_b+n_c)$—see the details in Tables 1 and 2. From the simulation example (omitted in the paper), we can see that the parameter estimation errors given by the ON-RLS algorithm are very close to those given by the RELS algorithm.

It has been just pointed out by Golub and Van Loan [16] that the flop (floating point operation) counting is a necessarily crude approach to the measuring of program efficiency since it ignores subscripting, memory traffic, and the countless other overheads associated with program execution, the flop counting is just a “quick and dirty” accounting method that captures only one of the several dimensions of the efficiency issue although multiplication/division and addition/subtraction with different lengths are different. The flop numbers of the ON-RLS and RELS algorithms at each recursion are given in Tables 1 and 2. Their total flops are, respectively, given by

$$\begin{aligned}N_1 &:= 4(n_a+n_d)^2+4(n_b+n_c)^2+4n_an_c+5n_a+10n_b+7n_c+10n_d,\\ N_2 &:= 4(n_an_c+n_b+n_d)^2+6(n_an_c+n_b+n_d).\end{aligned}$$

In order to compare the computational efficiency of these two algorithms, we count the difference between the amount of calculation of these two algorithms. When $n_a > 2$ and $n_b > 2$, $n_an_b > n_a+n_b$, $N_2 > 4(n_a+n_b+n_c+n_d)^2+6(n_a+n_b+n_c+n_d)$. Then, we have

$$\begin{aligned}N_2-N_1 &> 4(n_a+n_c+n_b+n_d)^2+6(n_a+n_c+n_b+n_d)-4(n_a+n_d)^2\\ &\quad -4(n_b+n_c)^2-4n_an_c-5n_a-10n_b-7n_c-10n_d\\ &= 10n_an_c+(8n_b-5)n_a+(8n_c-4)n_b+(8n_b-4)n_d > 0.\end{aligned}$$

It is clear that the ON-RLS algorithm requires less computational load than the RELS algorithm. For example, when $n_a = n_b = n_c = n_d = 5$, we have $N_2 - N_1 = 5110 - 1060 = 4050$ flops.

6 Example

Consider the following nonlinear system:

$$\begin{aligned}y(t) &= A(z)f(y(t))+B(z)u(t)+D(z)v(t),\\ A(z) &= a_1z^{-1}+a_2z^{-2} = -0.75z^{-1}-0.61z^{-2},\\ B(z) &= b_1z^{-1} = 0.96z^{-1},\\ D(z) &= 1+d_1z^{-1} = 1+0.4z^{-1},\\ f(y(t)) &= c_1y(t)+c_2\sin(y(t)) = 0.61y(t)+0.79\sin(y(t)),\end{aligned}$$

Table 1 The computational efficiency of the ON-RLS algorithm

Variables	Numbers of multiplications	Numbers of additions
$\hat{\boldsymbol{\theta}}(t) = \hat{\boldsymbol{\theta}}(t-1) + \boldsymbol{L}_1(t)e_1(t)$	$n_a + n_d$	$n_a + n_d$
$e_1(t) := y(t) - \boldsymbol{\varphi}^{\mathrm{T}}(t)\hat{\boldsymbol{b}}(t-1) - \hat{\boldsymbol{\varphi}}_1^{\mathrm{T}}(t)\hat{\boldsymbol{\theta}}(t-1)$	$n_a + n_b + n_d$	$n_a + n_b + n_d$
$\boldsymbol{\zeta}_1(t) := \boldsymbol{P}_1(t-1)\hat{\boldsymbol{\varphi}}_1(t)$	$(n_a + n_d)^2$	$(n_a + n_d)^2 - (n_a + n_d)$
$\boldsymbol{L}_1(t) = \boldsymbol{\zeta}_1(t)/(1 + \hat{\boldsymbol{\varphi}}_1^{\mathrm{T}}(t)\boldsymbol{\zeta}_1(t))$	$2(n_a + n_d)$	$n_a + n_d$
$\boldsymbol{P}_1(t) = \boldsymbol{P}_1(t-1) - \boldsymbol{L}_1(t)\hat{\boldsymbol{\zeta}}_1^{\mathrm{T}}(t)$	$(n_a + n_d)^2$	$(n_a + n_d)^2$
$\hat{\boldsymbol{\varphi}}_1(t) = [\hat{\boldsymbol{c}}^{\mathrm{T}}(t-1)\boldsymbol{F}^{\mathrm{T}}(t), \boldsymbol{\psi}^{\mathrm{T}}(t)]^{\mathrm{T}}$	$n_a n_c$	$n_a n_c - n_a$
$\hat{\boldsymbol{\vartheta}}(t) = \hat{\boldsymbol{\vartheta}}(t-1) + \boldsymbol{L}_1(t)e_2(t)$	$n_b + n_c$	$n_b + n_c$
$e_2(t) := y(t) - \hat{\boldsymbol{\psi}}^{\mathrm{T}}(t)\hat{\boldsymbol{d}}(t) - \hat{\boldsymbol{\varphi}}_2^{\mathrm{T}}(t)\hat{\boldsymbol{\vartheta}}(t-1)$	$n_b + n_c + n_d$	$n_b + n_c + n_d$
$\boldsymbol{\zeta}_2(t) := \boldsymbol{P}_2(t-1)\hat{\boldsymbol{\varphi}}_2(t)$	$(n_b + n_c)^2$	$(n_b + n_c)^2 - (n_b + n_c)$
$\boldsymbol{L}_2(t) = \boldsymbol{\zeta}_2(t)/(1 + \hat{\boldsymbol{\varphi}}_2^{\mathrm{T}}(t)\boldsymbol{\zeta}_2(t))$	$2(n_b + n_c)$	$n_b + n_c$
$\boldsymbol{P}_2(t) = \boldsymbol{P}_2(t-1) - \boldsymbol{L}_2(t)\hat{\boldsymbol{\zeta}}_2^{\mathrm{T}}(t)$	$(n_b + n_c)^2$	$(n_b + n_c)^2$
$\hat{\boldsymbol{\zeta}}_3(t) := \hat{\boldsymbol{a}}^{\mathrm{T}}(t)\boldsymbol{F}(t)$	$n_a n_c$	$n_a n_c - n_c$
$e(t) = y(t) - \hat{\boldsymbol{\zeta}}_3(t)\hat{\boldsymbol{c}}(t) - \hat{\boldsymbol{\varphi}}^{\mathrm{T}}(t)\hat{\boldsymbol{b}}(t) - \hat{\boldsymbol{\psi}}^{\mathrm{T}}(t)\hat{\boldsymbol{d}}(t)$	$n_b + n_c + n_d$	$n_b + n_c + n_d$
Sum	$2(n_a + n_d)^2 + 2(n_b + n_c)^2$ $+ 2n_a n_c + 4n_a + 6n_b + 5n_c + 6n_d$	$2(n_a + n_d)^2 + 2(n_b + n_c)^2$ $+ 2n_a n_c + n_a + 4n_b + 2n_c + 4n_d$
Total flops	$N_1 := 4(n_a + n_d)^2 + 4(n_b + n_c)^2 + 4n_a n_c + 5n_a + 10n_b + 7n_c + 10n_d$	

Table 2 The computational efficiency of the RELS algorithm

Variables	Numbers of multiplications	Numbers of additions
$\hat{\boldsymbol{\vartheta}}(t) = \hat{\boldsymbol{\vartheta}}(t-1) + \boldsymbol{L}(t)e(t)$	$n_a n_c + n_b + n_d$	$n_a n_c + n_b + n_d$
$e(t) = y(t) - \hat{\boldsymbol{\varphi}}^{\mathrm{T}}(t)\hat{\boldsymbol{\vartheta}}(t-1)$	$n_a n_c + n_b + n_d$	$n_a n_c + n_b + n_d$
$\boldsymbol{\zeta}(t) := \boldsymbol{P}(t-1)\hat{\boldsymbol{\varphi}}(t)$	$(n_a n_c + n_b + n_d)^2$	$(n_a n_c + n_b + n_d)^2 - (n_a n_c + n_b + n_d)$
$\boldsymbol{L}(t) = \boldsymbol{\zeta}(t)/(1 + \hat{\boldsymbol{\varphi}}^{\mathrm{T}}(t)\boldsymbol{\zeta}(t))$	$2(n_a n_c + n_b + n_d)$	$n_a n_c + n_b + n_d$
$\boldsymbol{P}(t) = \boldsymbol{P}(t-1) - \boldsymbol{L}(t)\hat{\boldsymbol{\zeta}}^{\mathrm{T}}(t)$	$(n_a n_c + n_b + n_d)^2$	$(n_a n_c + n_b + n_d)^2$
Sum	$2(n_a n_c + n_b + n_d)^2 + 4(n_a n_c + n_b + n_d)$	$2(n_a n_c + n_b + n_d)^2 + 2(n_a n_c + n_b + n_d)$
Total flops	$N_2 := 4(n_a n_c + n_b + n_d)^2 + 6(n_a n_c + n_b + n_d)$	

Table 3 The ON-RLS parameter estimates and errors

t	a_1	a_2	d_1	b_1	c_1	c_2	δ (%)
100	−0.72330	−0.58823	0.18282	0.88621	0.64330	0.71342	14.64393
200	−0.74339	−0.62911	0.30539	0.95622	0.57017	0.86840	7.76977
500	−0.73155	−0.59822	0.28568	0.96286	0.56532	0.84086	8.00060
1000	−0.76569	−0.60078	0.37239	0.98029	0.59412	0.81967	3.05103
2000	−0.75253	−0.60414	0.36940	0.96736	0.58600	0.81925	2.94798
3000	−0.75807	−0.61033	0.39273	0.96469	0.59701	0.80581	1.40439
True values	−0.75000	−0.61000	0.40000	0.96000	0.61000	0.79000	

$$\begin{aligned}
\boldsymbol{\theta} &= [a_1, a_2, d_1]^{\mathrm{T}} \\
&= [-0.75, -0.61, 0.4]^{\mathrm{T}}, \\
\boldsymbol{\vartheta} &= [b_1, c_1, c_2]^{\mathrm{T}} = [0.96, 0.61, 0.79]^{\mathrm{T}}, \\
\boldsymbol{\Theta} &= [a_1, a_2, d_1, b_1, c_1, c_2]^{\mathrm{T}} = [-0.75, -0.61, 0.4, 0.96, 0.61, 0.79]^{\mathrm{T}}.
\end{aligned}$$

In simulation, the input $\{u(t)\}$ is taken as a persistent excitation signal sequence with zero mean and unit variance, and $\{v(t)\}$ as a white noise sequence with zero mean and variance $\sigma^2 = 0.50^2$. Applying the ON-RLS algorithm and the ON-SG algorithm with $\lambda = 0.99$ to estimate the parameters of this system, the parameter estimates and errors are given in Tables 3 and 4, and the parameter estimation errors $\delta := \|\hat{\boldsymbol{\Theta}}(t) - \boldsymbol{\Theta}\|/\|\boldsymbol{\Theta}\|$ versus t are shown in Figs. 2 and 3.

From Tables 3, 4 and Figs. 2, 3, we can draw the following conclusions.

- It is clear that the parameter estimation errors given by two algorithms become smaller with the data length increasing.
- The parameter estimation accuracy of the ON-RLS algorithm is higher than that of the ON-SG algorithm.
- The parameter estimates given by the ON-RLS algorithm converge faster to their true values compared with the ON-SG algorithm for appropriate forgetting factors.

Table 4 The ON-SG parameter estimates and errors

t	a_1	a_2	d_1	b_1	c_1	c_2	δ (%)
100	−0.87270	−0.62619	0.45483	0.74665	0.56561	0.90142	16.61447
200	−0.83230	−0.66142	0.47472	0.79375	0.56805	0.88884	13.82537
500	−0.79048	−0.66185	0.48719	0.90499	0.51606	0.87919	10.58202
1000	−0.77568	−0.62755	0.46471	0.95122	0.51390	0.84625	7.88796
2000	−0.76758	−0.63322	0.45726	0.96158	0.55884	0.83236	5.48969
3000	−0.77247	−0.59245	0.44199	0.97625	0.59996	0.79394	3.22994
True values	−0.75000	−0.61000	0.40000	0.96000	0.61000	0.79000	

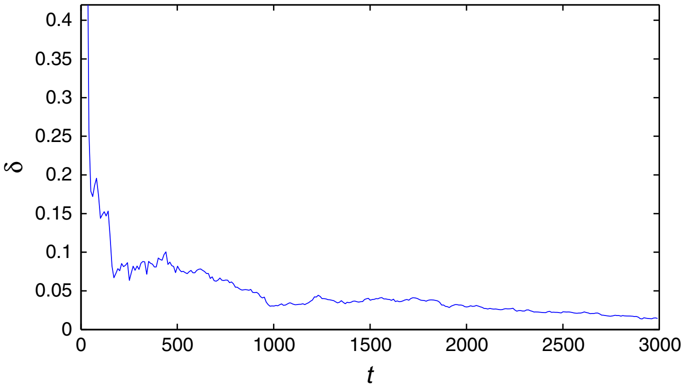


Fig. 2 The ON-RLS estimation errors versus t ($\sigma^2 = 0.50^2$)

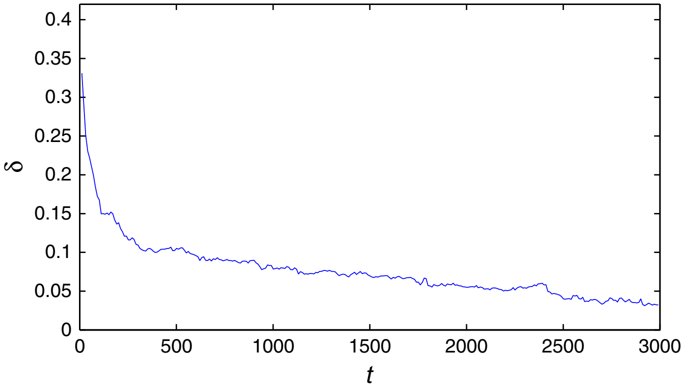


Fig. 3 The ON-SG estimation errors versus t ($\sigma^2 = 0.50^2$)

7 Conclusions

Using the hierarchical identification principle, a recursive least squares algorithm is derived for a special class of output nonlinear systems by transforming a nonlinear system into two identification models. The proposed algorithm can give a satisfactory identification accuracy and has higher computational efficiencies compared with the recursive extended least squares parameter estimation algorithm in [21]. The proposed algorithm can be extended to study identification problems of multivariable systems [13], linear-in-parameters systems [41,42] and impulsive dynamical systems [23,24] and applied to other fields [45–47].

Acknowledgments This work was supported by the National Natural Science Foundation of China (No. 61203111) and the PAPD of Jiangsu Higher Education Institutions.

References

1. J.C. Agüero, G.C. Goodwin, P.M.J. Van den Hof, A virtual closed loop method for closed loop identification. Automatica **47**(8), 1626–1637 (2011)
2. M. Ahmadi, H. Mojallali, Identification of multiple-input single-output Hammerstein models using Bezier curves and Bernstein polynomials. Appl. Math. Model. **35**(4), 1969–1982 (2011)
3. E.W. Bai, A blind approach to the Hammerstein–Wiener model identification. Automatica **38**(6), 967–979 (2002)
4. E.W. Bai, An optimal two-stage identification algorithm for Hammerstein–Wiener nonlinear systems. Automatica **34**(3), 333–338 (1998)
5. M. Dehghan, M. Hajarian, Fourth-order variants of Newton's method without second derivatives for solving non-linear equations. Eng. Comput. **29**(4), 356–365 (2012)
6. M. Dehghan, M. Hajarian, Analysis of an iterative algorithm to solve the generalized coupled Sylvester matrix equations. Appl. Math. Model. **35**(7), 3285–3300 (2011)
7. F. Ding, *System Identification-New Theory and Methods* (Science Press, Beijing, 2013)
8. F. Ding, *System Identification-Performances Analysis for Identification Methods* (Science Press, Beijing, 2014)
9. J. Ding, C.X. Fan, J.X. Lin, Auxiliary model based parameter estimation for dual-rate output error systems with colored noise. Appl. Math. Model. **37**(6), 4051–4058 (2013)
10. J. Ding, J.X. Lin, Modified subspace identification for periodically non-uniformly sampled systems by using the lifting technique. Circuits Syst. Signal Process. **33**(5), 1439–1449 (2014)
11. F. Ding, X.P. Liu, G. Liu, Identification methods for Hammerstein nonlinear systems. Digit. Signal Process. **21**(2), 215–238 (2011)
12. F. Ding, K.P. Deng, X.M. Liu, Decomposition based Newton iterative identification method for a Hammerstein nonlinear FIR system with ARMA noise. Circuits Syst. Signal Process. **33**(9), 2881–2893 (2014)
13. F. Ding, Y.J. Wang, J. Ding, Recursive least squares parameter estimation algorithms for systems with colored noise using the filtering technique and the auxiliary model. Digit. Signal Process. **37**, 100–108 (2015)
14. D. Fan, K. Lo, Identification for disturbed MIMO Wiener systems. Nonlinear Dyn. **55**(1–2), 31–42 (2009)
15. G.C. Goodwin, K.S. Sin, *Adaptive Filtering Prediction and Control* (Prentice Hall, Englewood Cliffs, 1984)
16. G.H. Golub, C.F. Van Loan, *Matrix Computations*, 3rd edn. (Johns Hopkins University Press, Baltimore, 1996)
17. Y. Gu, F. Ding, J.H. Li, States based iterative parameter estimation for a state space model with multi-state delays using decomposition. Signal Process. **106**, 294–230 (2015)
18. A. Hagenblad, L. Ljung, A. Wills, Maximum likelihood identification of Wiener models. Automatica **44**(11), 2697–2705 (2008)
19. Y.B. Hu, Iterative and recursive least squares estimation algorithms for moving average systems. Simul. Model. Pract. Theory **34**, 12–19 (2013)
20. P.P. Hu, F. Ding, Multistage least squares based iterative estimation for feedback nonlinear systems with moving average noises using the hierarchical identification principle. Nonlinear Dyn. **73**(1–2), 583–592 (2013)

21. Y.B. Hu, B.L. Liu, Q. Zhou, C. Yang, Recursive extended least squares parameter estimation for Wiener nonlinear systems with moving average noises. Circuits Syst. Signal Process. **33**(2), 655–664 (2014)
22. Y.B. Hu, B.L. Liu, Q. Zhou, A multi-innovation generalized extended stochastic gradient algorithm for output nonlinear autoregressive moving average systems. Appl. Math. Comput. **247**, 218–224 (2014)
23. Y. Ji, X.M. Liu, Unified synchronization criteria for hybrid switching-impulsive dynamical networks. Circuits Syst. Signal Process. **34**(5), 1499–1517 (2015)
24. Y. Ji, X.M. Liu et al., New criteria for the robust impulsive synchronization of uncertain chaotic delayed nonlinear systems. Nonlinear Dyn. **79**(1), 1–9 (2015)
25. J.H. Li, Parameter estimation for Hammerstein CARARMA systems based on the Newton iteration. Appl. Math. Lett. **26**(1), 91–96 (2013)
26. X.G. Liu, J. Lu, Least squares based iterative identification for a class of multirate systems. Automatica **46**(3), 549–554 (2010)
27. L. Ljung, *System Identification: Theory for the User*, 2nd edn. (Prentice Hall, Englewood Cliffs, 1999)
28. Y.W. Mao, F. Ding, A novel data filtering based multi-innovation stochastic gradient algorithm for Hammerstein nonlinear systems. Digit. Signal Process. **46**, 215–225 (2015)
29. P. Qin, R. Nishii, Z.J. Yang, Selection of NARX models estimated using weighted least squares method via GIC-based method and l1-norm regularization methods. Nonlinear Dyn. **70**(3), 1831–1846 (2012)
30. Y. Shi, T. Chen, Optimal design of multi-channel transmultiplexers with stopband energy and passband magnitude constraints. IEEE Trans. Circuits Syst. II Analog Digit. Signal Process. **50**(9), 659–662 (2003)
31. Y. Shi, B. Yu, Robust mixed H-2/H-infinity control of networked control systems with random time delays in both forward and backward communication links. Automatica **47**(4), 754–760 (2011)
32. Y. Shi, B. Yu, Output feedback stabilization of networked control systems with random delays modeled by Markov chains. IEEE Trans. Autom. Control **54**(7), 1668–1674 (2009)
33. J. Vörös, Modelling and identification of Wiener systems with two-segment nonlinearities. IEEE Trans. Control Syst. Technol. **11**(2), 253–265 (2003)
34. J. Vörös, Modeling and parameter identification of systems with multi-segment piecewise-linear characteristics. IEEE Trans. Autom. Control **47**(1), 184–188 (2002)
35. D.Q. Wang, Least squares-based recursive and iterative estimation for output error moving average systems using data filtering. IET Control Theory Appl. **5**(14), 1648–1657 (2011)
36. D.Q. Wang, F. Ding, Least squares based and gradient based iterative identification for Wiener nonlinear systems. Signal Process. **91**(5), 1182–1189 (2011)
37. X.H. Wang, F. Ding, Convergence of the recursive identification algorithms for multivariate pseudo-linear regressive systems. Int. J. Adapt. Control Signal Process. **30** (2016). doi:10.1002/acs.2642
38. X.H. Wang, F. Ding, Recursive parameter and state estimation for an input nonlinear state space system using the hierarchical identification principle. Signal Process. **117**, 208–218 (2015)
39. D.Q. Wang, Y.P. Gao, Recursive maximum likelihood identification method for a multivariable controlled autoregressive moving average system. IMA J. Math. Control Inf. (2015). doi:10.1093/imamci/dnv021
40. D.Q. Wang, H.B. Liu, F. Ding, Highly efficient identification methods for dual-rate Hammerstein systems. IEEE Trans. Control Syst. Technol. **23**(5), 1952–1960 (2015)
41. C. Wang, T. Tang, Recursive least squares estimation algorithm applied to a class of linear-in-parameters output error moving average systems. Appl. Math. Lett. **29**, 36–41 (2014)
42. C. Wang, T. Tang, Several gradient-based iterative estimation algorithms for a class of nonlinear systems using the filtering technique. Nonlinear Dyn. **77**(3), 769–780 (2014)
43. D.Q. Wang, W. Zhang, Improved least squares identification algorithm for multivariable Hammerstein systems. J. Frankl. Inst. Eng. Appl. Math. **352**(11), 5292–5370 (2015)
44. A. Wills, T.B. Schön, L. Ljung, B. Ninness, Identification of Hammerstein–Wiener models. Automatica **49**(1), 70–81 (2013)
45. D.Q. Zhu, X. Hua, B. Sun, A neurodynamics control strategy for real-time tracking control of autonomous underwater vehicles. J. Navig. **67**(1), 113–127 (2014)
46. D.Q. Zhu, H. Huang, S.X. Yang, Dynamic task assignment and path planning of multi-AUV system based on an improved self-organizing map and velocity synthesis method in 3D underwater workspace. IEEE Trans. Cybern. **43**(2), 504–514 (2013)
47. D.Q. Zhu, Q. Liu, Z. Hu, Fault-tolerant control algorithm of the manned submarine with multi-thruster based on quantum behaved particle swarm optimization. Int. J. Control **84**(11), 1817–1829 (2012)