IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received February 10, 2021, accepted March 1, 2021, date of publication March 9, 2021, date of current version March 19, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3064869

Adaptive Gradient Descent Algorithm for Networked Control Systems Using Redundant Rule

LIXIN LV AND JIAN ZHANG
Wuxi Professional College of Science and Technology, Wuxi 214028, China
Corresponding author: Lixin Lv (lv_wuxi@163.com)

This work was supported by the Natural Science Foundation for Colleges and Universities in Jiangsu Province under Grant 18KJB120009.

ABSTRACT This study proposes an adaptive gradient descent algorithm for networked control systems, where the systems encounter a time-delay. To estimate the parameters and time-delay simultaneously, a redundant rule based method is introduced to transform the system into an augmented model. Then, the parameters are estimated by using an adaptive gradient descent algorithm. Finally, the true parameters and time-delay can be obtained based on the parameter estimates of the augmented model. A simulation example is proposed to validate the effectiveness of the proposed algorithm.

INDEX TERMS Networked control system, gradient descent, redundant rule method, adaptive method, time-delayed model.

I. INTRODUCTION

Networked control systems (NCSs) widely exist in modern society owing to the development of the sensor and communication network technologies [1]–[3]. In many engineering practices, the work conditions are not ideal for on-site process data collection. Researchers usually use some sensors or robots to collect the data and then transmit these data to the control center for analysis [4], [5]. With the help of the data, the researchers can establish the process model and make robust controller. Therefore, system identification plays an important role in controlling the networked control system (NCS) [6], [7].

The least squares (LS) algorithm and gradient descent (GD) algorithm are two main kinds of estimation algorithms [8]–[11]. The LS algorithm is a form of mathematical regression analysis used to determine the line of best fit for a set of collected input and output data, providing a visual demonstration of the relationship between data points. Each data point represents the relationship between a known independent variable and an unknown dependent variable. The LS algorithm can obtain the parameter estimates in only one iteration by computing a derivative function, thus its convergence rate is fast [12], [13]. However, it needs to calculate a matrix inverse. When the order of the matrix is high, the computational efforts of the LS algorithm are heavy [14], [15].

The GD algorithm can be regarded as a worthy addition to the LS algorithm, for the reason that it does not require matrix inverse. The basic idea of the GD algorithm is to generate a sequential estimate by using an iterative function which consists of three parts: the initial estimate, the direction and corresponding step-size [16]–[19]. It is noted that the direction and its corresponding step-size are two major factors in the GD algorithm designing. Once the direction is determined, one should compute its corresponding optimal step-size to make the results quickly converge to the true values. Since the eigenvalue calculation is involved in the step-size design, it is difficult/impossible to compute the eigenvalues of a matrix with high-order [20], [21]. The focus of this paper is on proposing an adaptive GD algorithm which can avoid the eigenvalue calculation.

NCSs usually suffer from time-delay, packet loss or nonlinearity [22]–[24], which makes the identification of NCSs challenging. Therefore, a reliable estimation for NCSs can help the researchers make correct decision. Recently, many identification methods have been developed for NCSs. For example, Chen et al provided a VB approach for NCSs with missing outputs and varying time-delays, where the outputs and time-delays are estimated in the VB-E step, and the parameters are updated in the VB-M step [25]. Gopaluni proposed a particle filter based EM algorithm to a nonlinear

The associate editor coordinating the review of this manuscript and approving it for publication was Bing Li.

NCS with missing outputs, while the missing outputs are obtained in the EM-E step using the particle filter, and the parameters are estimated in the EM-M step [26]. Both algorithms are off-line algorithms, which update the parameters using matrix inverse, thus they employ heavy computational effort especially for large-scale systems.

In this study, an adaptive GD algorithm is proposed for an NCS with time-delay. By using the redundant rule method, the time-delayed NCS is transformed into an augmented model. To avoid the eigenvalue calculation, an adaptive GD algorithm which updates each element in the parameter vector is derived. The redundant rule based adaptive GD (RR-AGD) algorithm can estimate the parameters and time-delay quickly and simultaneously. The contributions of the RR-AGD algorithm are summarized as follows:

(1) Unlike the LS algorithm, the RR-AGD algorithm has no matrix inversion calculation, thus can be extended to large-scale systems;

(2) Different from the GD algorithm, the RR-AGD algorithm does not require eigenvalue calculation, therefore involves less computational effort;

(3) The RR-AGD algorithm utilizes different step-sizes and directions for each element in the parameter vector, thus has faster convergence rate than the GD algorithm.

Briefly, the paper is listed as follows. Section II introduces the networked control system. Section III develops the redundant rule based gradient descent algorithm. Section IV develops the redundant rule based adaptive gradient descent algorithm. Section V provides a simulation example. Finally, Section VI gives a conclusion of this paper.

II. NETWORKED CONTROL SYSTEM

The structure of the networked control system is shown in Figure 1, where $u(s)$ and $y(s)$ are the input and output data of the continuous-time process, $u(t)$ and $y(t)$ are the sampled data from the continuous $u(s)$ and $y(s)$ at each sampling instant h. When $y(t)$ is transmitted over the wireless network, it encounters a time-delay τ. Therefore, the process model is represented by

$$y(t) = \boldsymbol{\alpha}(z)y(t-\tau) + \boldsymbol{\beta}(z)u(t) + v(t), \tag{1}$$

where $v(t)$ is a Gaussian white noise. The parameters $\boldsymbol{\alpha}(z)$ and $\boldsymbol{\beta}(z)$ are written as

$$\begin{aligned}\boldsymbol{\alpha}(z) &= \alpha_1 z^{-1} + \alpha_2 z^{-2} + \cdots + \alpha_n z^{-n},\\ \boldsymbol{\beta}(z) &= \beta_1 z^{-1} + \beta_2 z^{-2} + \cdots + \beta_m z^{-m},\end{aligned}$$

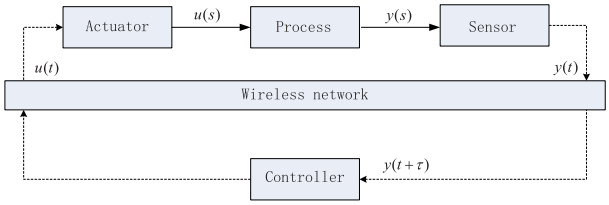

FIGURE 1. The three-tank system.

and $z^{-i}u(t) = u(t-i)$.

Define

$$\begin{aligned}\boldsymbol{\kappa} &= [\alpha_1, \cdots, \alpha_n, \beta_1, \cdots, \beta_n]^{\mathrm{T}} \in R^{m+n},\\ \boldsymbol{\pi}(t) &= [y(t-\tau-1), \cdots, y(t-\tau-n), u(t-1), \cdots,\\ &\quad u(t-m)]^{\mathrm{T}} \in R^{m+n}.\end{aligned}$$

Then the process model can be simplified as

$$y(t) = \boldsymbol{\pi}^{\mathrm{T}}(t)\boldsymbol{\kappa} + v(t). \tag{2}$$

Generate L input and output data and let

$$\begin{aligned}Y(L) &= [y(L), y(L-1), \cdots, y(1)]^{\mathrm{T}} \in R^{L},\\ W(L) &= [\boldsymbol{\pi}(L), \boldsymbol{\pi}(L-1), \cdots, \boldsymbol{\pi}(1)]^{\mathrm{T}} \in R^{L\times(m+n)},\\ V(L) &= [v(L), v(L-1), \cdots, v(1)]^{\mathrm{T}} \in R^{L}.\end{aligned}$$

Define the cost function

$$J(\boldsymbol{\kappa}, \tau) = \frac{1}{2}[Y(L) - W(L)\boldsymbol{\kappa}]^{\mathrm{T}}[Y(L) - W(L)\boldsymbol{\kappa}].$$

We aim to find the optimal parameter estimate $\boldsymbol{\kappa}$ and time-delay estimate τ to keep the cost function $J(\boldsymbol{\kappa}, \tau)$ reaching the minimum.

III. REDUNDANT RULE BASED GRADIENT DESCENT ALGORITHM

Since the information matrix $W(L)$ contains the unknown time-delay, there exist two ways to obtain both estimates: (1) use the iterative algorithm to obtain these two kinds estimates separately; (2) regard the time-delay as an augment parameter, and then estimate these two kinds of estimates simultaneously.

A. METHOD 1

In this case, we should assign prior knowledge of the time-delay first. Usually, assume that the time-delay lays in the interval $[0, M]$. Since the time-delay is an integer, the unknown τ may be equal to $0, 1, \cdots$ or M. Let $\tau = i$, $i = 0, 1, \cdots, M$. In iteration k, $M+1$ cost functions are performed as

$$J(\boldsymbol{\kappa}, \tau = i) = \frac{1}{2}[Y(L) - W_i(L)\boldsymbol{\kappa}_{k-1}]^{\mathrm{T}}[Y(L) - W_i(L)\boldsymbol{\kappa}_{k-1}],$$

where $\boldsymbol{\kappa}_{k-1}$ means the parameter estimates in iteration $k-1$, and the optimal time-delay estimate can be computed by

$$\tau_k = \arg\min_{\tau=i}\{J(\boldsymbol{\kappa}, \tau = i), i = 0, 1, \cdots, M\}. \tag{3}$$

Once the time-delay is obtained, the cost function can be written as

$$J(\boldsymbol{\kappa}, \tau = k) = \frac{1}{2}[Y(L) - W_k(L)\boldsymbol{\kappa}]^{\mathrm{T}}[Y(L) - W_k(L)\boldsymbol{\kappa}].$$

Using the gradient descent algorithm to update the parameters yields

$$\boldsymbol{\kappa}_k = \boldsymbol{\kappa}_{k-1} + \gamma_{k-1}W_k^T(L)[Y(L) - W_k(L)\boldsymbol{\kappa}_{k-1}], \tag{4}$$

where $W_k^T(L)[Y(L) - W_k(L)\boldsymbol{\kappa}_{k-1}]$ is the negative gradient direction and γ_{k-1} is its corresponding step-size, which should satisfy

$$0 < \gamma_{k-1} < \frac{2}{\lambda_{max}[W_k^T(L)W_k(L)]}, \tag{5}$$

in which $\lambda_{max}[W_k^T(L)W_k(L)]$ means the maximum eigenvalue of matrix $[W_k^T(L)W_k(L)]$.

Then the gradient descent iterative (GDI) algorithm is summarized as follows:

$$\boldsymbol{\kappa}_k = \boldsymbol{\kappa}_{k-1} + \gamma_{k-1}W_k^T(L)[Y(L) - W_k(L)\boldsymbol{\kappa}_{k-1}], \tag{6}$$

$$\tau_k = \arg\min_{\tau=i}\{J(\boldsymbol{\kappa}, \tau = i), i = 0, 1, \cdots, M\}, \tag{7}$$

$$0 < \gamma_{k-1} < \frac{2}{\lambda_{max}[W_k^T(L)W_k(L)]}, \tag{8}$$

and the steps of the GDI algorithm are listed in the following:

1) Initialization: Let $u(-t) = 0, v(-t) = 0, y(-t) = 0$, and give a small positive number ε and a positive integer M.
2) Let $\boldsymbol{\kappa}_0 = [0, \cdots, 0]^{\mathrm{T}} \in R^{m+n}$ and $k = 0$.
3) Collect the input and output data $u(t), y(t), t = 1, 2, \cdots, L$.
4) Compute the cost functions $J(\boldsymbol{\kappa}, \tau = i), i = 0, 1, \cdots, M$.
5) Estimate τ_k by (7).
6) Compute γ_{k-1} by (8).
7) Compute $\boldsymbol{\kappa}_k$ based on (6).
8) Compare $\boldsymbol{\kappa}_k$ and $\boldsymbol{\kappa}_{k-1}$: if $\|\boldsymbol{\kappa}_k - \boldsymbol{\kappa}_{k-1}\| \leqslant \varepsilon$, then obtain the estimates $\boldsymbol{\kappa}_k$ and τ_k; otherwise, increase k by 1 and go to step 4.

Remark 1: The GDI algorithm estimates the parameters and time-delay iteratively, the parameter estimates are dependent on the time-delay estimate, and vice versa. If one kind estimate is not accurate, the other one also has poor performance.

Remark 2: The GDI algorithm has heavy computational efforts because of the two main calculations: one is the M cost functions computation; the other is the step-size calculation where one should compute the eigenvalue of a changing matrix $W_k^T(L)W_k(L)$ in each iteration.

B. METHOD 2

The other way to estimate the time-delay networked model is to update the parameters and time-delay simultaneously. Let the time-delay be an augment parameter, and define a new parameter vector as

$$\bar{\boldsymbol{\kappa}} = [\alpha_1, \cdots, \alpha_n, \beta_1, \cdots, \beta_n, \tau]^{\mathrm{T}} \in R^{m+n+1},$$

and the cost function can be written by

$$J(\bar{\boldsymbol{\kappa}}) = \frac{1}{2}[Y(L) - W(L, \tau)\boldsymbol{\kappa}]^{\mathrm{T}}[Y(L) - W(L, \tau)\boldsymbol{\kappa}].$$

Clearly, one cannot extract $\bar{\boldsymbol{\kappa}}$ from the cost function, thus obtaining the parameter estimates $\boldsymbol{\kappa}$ is infeasible.

The redundant rule method proposed in [27], [28] is an effective algorithm which can estimate these two kinds estimates simultaneously. Its basic idea is to use the redundant rule to turn the time-delayed model into an augmented model which contains two parts: one is the original model part, and the other is the redundant part. By analysing the parameter estimates of the augmented model, the time-delay and the original parameters can be simultaneously obtained.

Assume that the upper bound of the time-delay is M, and define the augmented parameter vector and the information vector as

$$\begin{aligned}\boldsymbol{\kappa}_{au} &= [\bar{\alpha}_1, \cdots, \bar{\alpha}_{M+n}, \beta_1, \cdots, \beta_n]^{\mathrm{T}} \in R^{m+n+M},\\ \boldsymbol{\pi}_{au}(t) &= [y(t-1), y(t-2), \cdots, y(t-\tau-n), \cdots,\\ &\quad y(t-M-n), u(t-1), \cdots, u(t-m)]^{\mathrm{T}} \in R^{m+n+M}.\end{aligned}$$

Actually, the augmented parameter vector $\boldsymbol{\kappa}_{au}$ consists of four parts, that is

$$\begin{aligned}\boldsymbol{\kappa}_{au} = [&\overbrace{\bar{\alpha}_1, \cdots, \bar{\alpha}_\tau}^{zero\ vector}, \underbrace{\bar{\alpha}_{\tau+1}, \cdots, \bar{\alpha}_{\tau+n}}_{true\ vector}, \overbrace{\bar{\alpha}_{\tau+n+1}, \bar{\alpha}_{M+n}}^{zero\ vector},\\ &\underbrace{\beta_1, \cdots, \beta_m}_{true\ vector}]^{\mathrm{T}} \in R^{m+n+M},\end{aligned}$$

and the augmented information vector also consists of four parts,

$$\begin{aligned}\boldsymbol{\pi}_{au}(t) = [&\overbrace{y(t-1), \cdots, y(t-\tau)}^{redundant\ vector},\\ &\underbrace{y(t-\tau-1), \cdots, y(t-\tau-n)}_{true\ vector},\\ &\overbrace{y(t-\tau-n-1), \cdots, y(t-M-n)}^{redundant\ vector},\\ &\underbrace{u(t-1), \cdots, u(t-m)}_{true\ vector}]^{\mathrm{T}} \in R^{m+n+M}.\end{aligned}$$

Since the redundant vectors in the information vector play no role in the output $y(t)$, their corresponding parameter vectors are equal to zero vectors.

Define

$$W_{au}(L) = [\boldsymbol{\pi}_{au}(L), \boldsymbol{\pi}_{au}(L-1), \cdots, \boldsymbol{\pi}_{au}(1)]^{\mathrm{T}} \in R^{L\times(m+n+M)}.$$

The augmented model can be expressed as

$$Y(L) = W_{au}(L)\boldsymbol{\kappa}_{au} + V(L). \tag{9}$$

It follows that the redundant rule based GDI (RR-GDI) algorithm for the augmented model is

$$\begin{aligned}\boldsymbol{\kappa}_{au}^k &= \boldsymbol{\kappa}_{au}^{k-1} + \gamma^{k-1}W_{au}^{\mathrm{T}}(L)[Y(L) - W_{au}(L)\boldsymbol{\kappa}_{au}^{k-1}],\\ 0 &< \gamma^{k-1} < \frac{2}{\lambda_{max}[W_{au}^{\mathrm{T}}(L)W_{au}(L)]}.\end{aligned}$$

Once the parameter estimates are obtained, each element in the parameter estimate vector should be compared with a threshold. If the absolute value of the element is smaller than the threshold, this element can be regarded as a zero element; otherwise, it is the true element.

Remark 3: The RR-GDI algorithm can estimate the parameters and time-delay simultaneously. However, to choose an optimal threshold is challenging. A small one may lead the redundant elements to be true elements, while a large one may mistaken some true elements for zero elements.

Remark 4: The information matrix $W_{au}^{\mathrm{T}}(L)W_{au}(L)$ is unchanged in each iteration, one does not need to compute the eigenvalues of the matrix $W_{au}^{\mathrm{T}}(L)W_{au}(L)$ in each iteration. Therefore, the RR-GDI algorithm has less computational efforts.

IV. REDUNDANT RULE BASED ADAPTIVE GRADIENT DESCENT ALGORITHM

In the RR-GDI algorithm, one should compute the eigenvalues of the information matrix $W_{au}^{\mathrm{T}}(L)W_{au}(L)$. Sometimes, we have no confidence of the time-delay τ, thus a larger upper bound M should be given. If the dimension $M+m+n$ of the matrix is large, obtaining the eigenvalues is challenging/impossible. To overcome this difficulty, a redundant rule based adaptive gradient descent iterative (RR-AGD) algorithm is proposed.

A. RR-AGD ALGORITHM

Rewrite the information matrix as

$$\begin{aligned} W_{au}(L) &= [\boldsymbol{\pi}_{au}(L), \boldsymbol{\pi}_{au}(L-1), \cdots, \boldsymbol{\pi}_{au}(1)]^{\mathrm{T}} \\ &= [\omega_{au}(1), \omega_{au}(2), \cdots, \\ &\quad \omega_{au}(m+n+M)] \in R^{L\times(m+n+M)}. \end{aligned} \tag{10}$$

The augmented model can be written by

$$\begin{aligned} Y(L) &= \omega_{au}(1)\bar{\alpha}_1 + \omega_{au}(2)\bar{\alpha}_2 + \cdots \\ &\quad + \omega_{au}(M+n)\bar{\alpha}_{M+n} + \omega_{au}(M+n+1)\beta_1 + \cdots \\ &\quad + \omega_{au}(M+n+m)\beta_m + V(L). \end{aligned}$$

Define the cost function

$$\begin{aligned} J(\boldsymbol{\kappa}_{au}) = \frac{1}{2}\|&Y(L) - \omega_{au}(1)\bar{\alpha}_1 - \omega_{au}(2)\bar{\alpha}_2 - \cdots \\ &- \omega_{au}(M+n)\bar{\alpha}_{M+n} - \omega_{au}(M+n+1)\beta_1 - \cdots \\ &- \omega_{au}(M+n+m)\beta_m\|. \end{aligned} \tag{11}$$

In order to avoid calculating eigenvalues, the AGD method updates each element in the parameter vector separately. Assume that the parameter vector estimate in iteration $k-1$ is $\boldsymbol{\kappa}_{au}^{k-1}$, we then update the parameter $\bar{\alpha}_1$ first. The corresponding cost function of $\bar{\alpha}_1$ is

$$\begin{aligned} &J(\bar{\alpha}_1, \bar{\alpha}_2^{k-1}, \cdots, \bar{\alpha}_{M+n+m}^{k-1}) \\ &= \frac{1}{2}\|Y(L) - \omega_{au}(1)\bar{\alpha}_1 - \omega_{au}(2)\bar{\alpha}_2^{k-1} - \cdots \\ &\quad - \omega_{au}(M+n)\bar{\alpha}_{M+n}^{k-1} - \omega_{au}(M+n+1)\beta_1^{k-1} - \cdots \\ &\quad - \omega_{au}(M+n+m)\beta_m^{k-1}\|. \end{aligned} \tag{12}$$

Using the GD algorithm to update the parameter $\bar{\alpha}_1$ gives

$$\begin{aligned} \bar{\alpha}_1^k &= \bar{\alpha}_1^{k-1} + \gamma_1^{k-1}\omega_{au}^{\mathrm{T}}(1)[Y(L) - \omega_{au}(1)\bar{\alpha}_1^{k-1} \\ &\quad - \omega_{au}(2)\bar{\alpha}_2^{k-1} - \cdots \\ &\quad - \omega_{au}(M+n)\bar{\alpha}_{M+n}^{k-1} - \omega_{au}(M+n+1)\beta_1^{k-1} - \cdots \\ &\quad - \omega_{au}(M+n+m)\beta_m^{k-1}] \\ &= \bar{\alpha}_1^{k-1} + \gamma_1^{k-1}\omega_{au}^{\mathrm{T}}(1)[Y(L) - W_{au}(L)\boldsymbol{\kappa}_{au}^{k-1}]. \end{aligned}$$

Substituting the above equation into Equation (12) and taking the derivative with respect to γ_1^{k-1} yield

$$\gamma_1^{k-1} = \frac{1}{\omega_{au}^{\mathrm{T}}(1)\omega_{au}(1)}.$$

Then the RR-AGD algorithm for the augmented model is summarized as follows

$$\begin{aligned} \bar{\alpha}_i^k &= \bar{\alpha}_i^{k-1} + \gamma_i^{k-1}\omega_{au}^{\mathrm{T}}(i)[Y(L) - W_{au}(L)\boldsymbol{\kappa}_{au}^{k-1}], \\ &\quad i = 1, \cdots, M+n+m, \beta_j^k = \bar{\alpha}_{M+n+j}^k, \\ &\quad j = 1, \cdots, m, \end{aligned} \tag{13}$$

$$\gamma_i^{k-1} = \frac{1}{\omega_{au}^{\mathrm{T}}(i)\omega_{au}(i)}. \tag{14}$$

The steps of the RR-AGD algorithm are listed in the following:

1) Initialization: Let $u(-t) = 0, v(-t) = 0, y(-t) = 0$, and give a small positive number ε and a positive integer M.
2) Let $\boldsymbol{\kappa}_{au}^0 = [0, \cdots, 0]^{\mathrm{T}} \in R^{m+n+M}$ and $k = 0$.
3) Collect the input and output data $u(t), y(t), t = 1, 2, \cdots, L$.
4) Compute $\gamma_i^{k-1}, i = 1, 2, \cdots, m+n+M$ according to (14).
5) Update $\bar{\alpha}_i^k$ based on (13).
6) Compare $\boldsymbol{\kappa}_{au}^k$ and $\boldsymbol{\kappa}_{au}^{k-1}$: if $\|\boldsymbol{\kappa}_{au}^k - \boldsymbol{\kappa}_{au}^{k-1}\| \leqslant \varepsilon$, then obtain the estimates $\boldsymbol{\kappa}_{au}^k$ and go to next step; otherwise, increase k by 1 and go to step 4.
7) Compare each element $\bar{\alpha}_i^k, i = 1, 2, \cdots, m+n+M$ in $\boldsymbol{\kappa}_{au}^k$ with the threshold ς, if $|\bar{\alpha}_i^k| \leqslant \varsigma$, let $\bar{\alpha}_i^k = 0$; otherwise, keep it unchanged.
8) Obtain the final parameter estimates $\boldsymbol{\kappa}_{au}^k$.
9) Compute the time-delay τ according to the four parts in $\boldsymbol{\kappa}_{au}^k$.

Remark 5: The RR-AGD algorithm updates the elements in the parameter vector one by one, and assigns different step-sizes for each element, thus we termed it as adaptive gradient descent iterative algorithm. In addition, this algorithm is more effective when the elements in the parameter vector have very large scalar difference among them.

Remark 6: The RR-AGD algorithm uses vector multiplications instead of the eigenvalue calculation, thus it is easier than the RR-GDI algorithm.

B. CONVERGENCE ANALYSIS OF THE RR-AGD ALGORITHM

In this subsection, the convergence analysis of the RR-AGD algorithm is given which can guarantee the effectiveness of the RR-AGD algorithm.

Theorem 1: For the augmented model proposed in (9), the parameter estimates using the RR-AGD algorithm are given by (13) and (14). Then, the cost function $J(\boldsymbol{\kappa}_{au})$ is monotonically decreasing.

Proof: Assume the parameter vector estimate in iteration $k-1$ is $\boldsymbol{\kappa}_{au}^{k-1}$. Then the cost function for this estimate is

$$\begin{aligned} J(\boldsymbol{\kappa}_{au}^{k-1}) = \frac{1}{2}\|Y(L) &- \omega_{au}(1)\bar{\alpha}_1^{k-1} - \omega_{au}(2)\bar{\alpha}_2^{k-1} - \cdots \\ &- \omega_{au}(M+n)\bar{\alpha}_{M+n}^{k-1} \\ &- \omega_{au}(M+n+1)\beta_1^{k-1} - \cdots \\ &- \omega_{au}(M+n+m)\beta_m^{k-1}\|. \end{aligned}$$

Since the estimate $\bar{\alpha}_1^k$ is updated by

$$\begin{aligned} \bar{\alpha}_1^k &= \bar{\alpha}_1^{k-1} - \gamma_1^{k-1}\nabla J(\boldsymbol{\kappa}_{au})_{\boldsymbol{\kappa}_{au}^{k-1}} \\ &= \bar{\alpha}_1^{k-1} + \gamma_1^{k-1}\omega_{au}^{\mathrm{T}}(1)[Y(L) - \omega_{au}(1)\bar{\alpha}_1^{k-1} \\ &\quad - \omega_{au}(2)\bar{\alpha}_2^{k-1} - \cdots \\ &\quad - \omega_{au}(M+n)\bar{\alpha}_{M+n}^{k-1} - \omega_{au}(M+n+1)\beta_1^{k-1} - \cdots \\ &\quad - \omega_{au}(M+n+m)\beta_m^{k-1}], \end{aligned}$$

where $-\nabla J(\boldsymbol{\kappa}_{au})_{\boldsymbol{\kappa}_{au}^{k-1}}$ is the negative gradient direction, which can ensure the estimates converge to the true values. The steps-size $\gamma_1^{k-1} = \frac{1}{\omega_{au}^{\mathrm{T}}(1)\omega_{au}(1)}$ can guarantee

$$J(\bar{\alpha}_1^k, \bar{\alpha}_2^{k-1}, \bar{\alpha}_3^{k-1}, \cdots, \bar{\alpha}_{m+n+M}^{k-1}) \leqslant J(\boldsymbol{\kappa}_{au}^{k-1}).$$

In the same way, we can get the following inequalities by Equations (13) and (14)

$$\begin{aligned} J(\boldsymbol{\kappa}_{au}^k) &= J(\bar{\alpha}_1^k, \bar{\alpha}_2^k, \bar{\alpha}_3^k, \cdots, \bar{\alpha}_{m+n+M}^k) \leqslant \cdots \leqslant \cdots \\ &\leqslant J(\bar{\alpha}_1^k, \bar{\alpha}_2^k, \bar{\alpha}_3^{k-1}, \cdots, \bar{\alpha}_{m+n+M}^{k-1}) \\ &\leqslant J(\bar{\alpha}_1^k, \bar{\alpha}_2^{k-1}, \bar{\alpha}_3^{k-1}, \cdots, \bar{\alpha}_{m+n+M}^{k-1}) \leqslant J(\boldsymbol{\kappa}_{au}^{k-1}). \end{aligned} \tag{15}$$

The above inequalities demonstrate that the cost function is monotonically decreasing. ■

Remark 7: Since the cost function $J(\boldsymbol{\kappa}_{au})$ is monotonically decreasing, the RR-AGD algorithm is convergent. In addition, if the noise $v(t)$ is a Gaussian white noise, the sequence $\{\boldsymbol{\kappa}_{au}^k\}$ converges to true values.

V. EXAMPLE

Consider the networked control system proposed in [25], where $u(t)$, $x(t)$ are the inlet water valve of Tank1 and the liquid level of Tank2, respectively, $y(t)$ is the computer output which is equal to $y(t) = x(t) + v(t)$, $v(t)$ is a Gaussian white noise, see Figure 2. The relationship between $y(t)$ and $u(t)$ can be modeled by

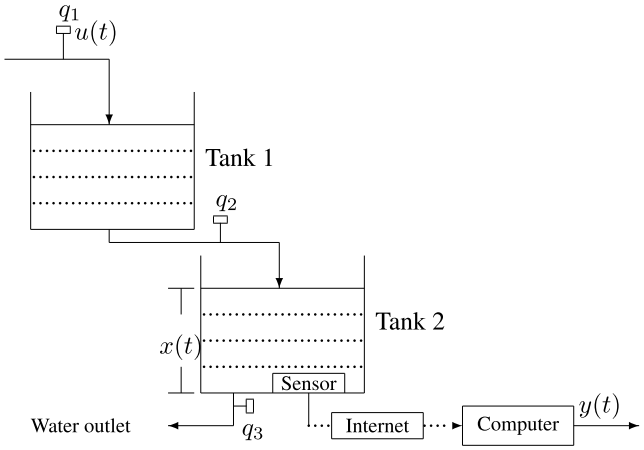

FIGURE 2. A water tank system.

$$\begin{aligned} y(t) &= \alpha_1 y(t-1) + \alpha_2 y(t-2) + \beta_1 u(t-1) \\ &\quad + \beta_2 u(t-2) + v(t) \\ &= 0.22y(t-1) - 0.7y(t-2) + 0.36u(t-1) \\ &\quad + 1.1u(t-2) + v(t). \end{aligned}$$

The input $\{u(t)\}$ is a filtered random binary signal sequence and updated at every $h = 10sec$. We impose a time-delay of $20sec$ for the output, which means that the true time-delay is $\tau = 2$, that is the time-delayed system is written by

$$\begin{aligned} y(t) = 0.22y(t-3) - 0.7y(t-4) + 0.36u(t-1) \\ + 1.1u(t-2) + v(t). \end{aligned}$$

Assume that the upper bound of the time-delay τ is $M = 4$, then the augmented model can be written as

$$\begin{aligned} y(t) &= \bar{\alpha}_1 y(t-1) + \bar{\alpha}_2 y(t-2) + \bar{\alpha}_3 y(t-3) + \bar{\alpha}_4 y(t-4) \\ &\quad + \bar{\alpha}_5 y(t-5) + \bar{\alpha}_6 y(t-6) + \beta_1 u(t-2) + \beta_2 u(t-2) + v(t), \\ \boldsymbol{\kappa}_{au} &= [\bar{\alpha}_1, \bar{\alpha}_2, \bar{\alpha}_3, \bar{\alpha}_4, \bar{\alpha}_5, \bar{\alpha}_6, \beta_1, \beta_2]^{\mathrm{T}} \\ &= [0, 0, 0.22, -0.7, 0, 0, 0.36, 1.1]^{\mathrm{T}}. \end{aligned}$$

Apply the RR-GDI and RR-AGD algorithms to estimate the parameters and time-delay. The parameter estimates and their estimation errors $\delta := \|\boldsymbol{\kappa}_{au}^k - \boldsymbol{\kappa}_{au}\|/\|\boldsymbol{\kappa}_{au}\|$ are shown in Table 1 and Figure 3.

Assign the threshold $\varsigma = 0.1$, then the estimates calculated by using the RR-GDI algorithm are

$$\boldsymbol{\kappa}_{au}^{100} = [0, 0, 0.2081, -0.70813, 0, 0, 0.35756, 1.06787]^{\mathrm{T}},$$

while the estimates using the RR-AGD algorithm are

$$\boldsymbol{\kappa}_{au}^{100} = [0, 0, 0.20804, -0.70085, 0, 0, 0.36258, 1.09412]^{\mathrm{T}}.$$

Therefore, we can get that the time-delay is $\tau = 2$.

Using the Monte Carlo method for this system with 100 set of different noises (RR-AGD), the errors δ are shown in Figure 4.

TABLE 1. The parameter estimates and errors.

Algorithms	k	$\bar{\alpha}_1$	$\bar{\alpha}_2$	$\bar{\alpha}_3$	$\bar{\alpha}_4$	$\bar{\alpha}_5$	$\bar{\alpha}_6$	β_1	β_2	δ (%)
RR-GDI	2	0.07053	0.11586	0.19512	-0.07285	0.15577	0.12784	0.13676	0.20350	83.40728
	20	0.09088	0.08841	0.25297	-0.62229	0.12103	0.08344	0.26433	0.68709	34.57193
	40	0.07946	0.04119	0.21622	-0.70167	0.07666	0.03572	0.31839	0.90544	17.07687
	60	0.05352	0.01532	0.20800	-0.71011	0.05152	0.01169	0.34119	1.00055	9.33960
	80	0.03427	0.00444	0.20748	-0.70813	0.03510	0.00049	0.35209	1.04547	5.50236
	100	0.02183	0.00072	0.20801	-0.70558	0.02441	-0.00398	0.35756	1.06787	3.50105
RR-AGD	2	0.17008	0.15658	0.36211	-0.51587	0.30903	0.17774	0.18377	0.36251	65.65750
	20	0.00133	0.00296	0.20973	-0.69888	0.00709	-0.00294	0.35992	1.09181	1.13812
	40	0.00055	0.00390	0.20804	-0.70086	0.00467	-0.00258	0.36258	1.09409	1.10471
	60	0.00051	0.00393	0.20804	-0.70085	0.00462	-0.00256	0.36258	1.09412	1.10332
	80	0.00051	0.00393	0.20804	-0.70085	0.00462	-0.00256	0.36258	1.09412	1.10328
	100	0.00051	0.00393	0.20804	-0.70085	0.00462	-0.00256	0.36258	1.09412	1.10328
	True Values	0.00000	0.00000	0.22000	-0.70000	0.00000	0.00000	0.36000	1.10000	0.00000

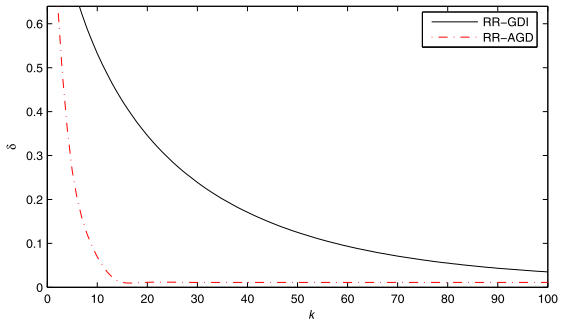

FIGURE 3. The parameter estimation errors δ versus k.

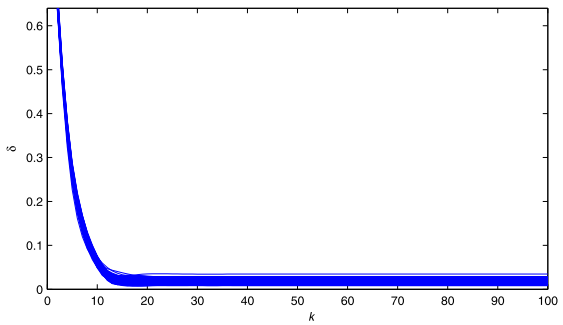

FIGURE 4. The parameter estimation errors δ versus k using Monte Carlo method (100 set of different noises, RR-AGD).

From this simulation example, we can conclude:

(1) The parameter and time-delay estimates of the RR-GDI and RR-AGD algorithms can asymptotically converge to the true values with the increasing of k, as shown in Figure 3;

(2) The RR-AGD algorithm has a quicker convergence rate than that of the RR-GDI algorithm, which is shown in Table 1.

(3) The RR-AGD algorithm is robust to noises, which can be shown in Figure 4.

VI. CONCLUSION

This paper presents a redundant rule based adaptive gradient descent algorithm for a NCS with time-delay. In order to simultaneously estimate the parameters and time-delay, a redundant rule method is introduced to transform the NCS into an augmented model. Then an adaptive gradient descent algorithm is developed to update each element in the parameter vector one by one. Compared with the traditional RR-GDI algorithm, the proposed algorithm involves less computational efforts and faster convergence rates, thus can be widely used in engineering practice.

REFERENCES

[1] J. Baillieul and P. J. Antsaklis, "Control and communication challenges in networked real-time systems," *Proc. IEEE*, vol. 95, no. 1, pp. 9–28, Jan. 2007.

[2] W. Chen, D. Ding, X. Ge, Q. L. Han, and G. Wei, "$\mathcal{H}_\infty$ containment control of multiagent systems under event-triggered communication scheduling: The finite-horizon case," *IEEE Trans. Cybern.*, vol. 50, no. 4, pp. 1372–1382, Apr. 2020.

[3] H. Li and Y. Shi, "Distributed receding horizon control of large-scale nonlinear systems: Handling communication delays and disturbances," *Automatica*, vol. 50, no. 4, pp. 1264–1271, Apr. 2014.

[4] H. Sun, J. Sun, and J. Chen, "Analysis and synthesis of networked control systems with random network-induced delays and sampling intervals," *Automatica*, vol. 125, Mar. 2021, Art. no. 109385, doi: 10.1016/j.automatica.2020.109385.

[5] J. Cao, D. Ding, J. Liu, E. Tian, S. Hu, and X. Xie, "Hybrid-triggered-based security controller design for networked control system under multiple cyber attacks," *Inf. Sci.*, vol. 548, pp. 69–84, Feb. 2021.

[6] M. Jalanko, Y. Sanchez, V. Mahalec, and P. Mhaskar, "Adaptive system identification of industrial ethylene splitter: A comparison of subspace identification and artificial neural networks," *Comput. Chem. Eng.*, vol. 147, Apr. 2021, Art. no. 107240, doi: 10.1016/j.compchemeng.2021.107240.

[7] M. Fei, D. Du, and K. Li, "A fast model identification method for networked control system," *Appl. Math. Comput.*, vol. 205, no. 2, pp. 658–667, Nov. 2008.

[8] H. Xu, F. Ding, and E. Yang, "Three-stage multi-innovation parameter estimation for an exponential autoregressive time-series model with moving average noise by using the data filtering technique," *Int. J. Robust Nonlinear Control*, vol. 31, no. 1, pp. 166–184, Jan. 2021.

[9] L. Xu, F. Ding, and E. Yang, "Auxiliary model multiinnovation stochastic gradient parameter estimation methods for nonlinear sandwich systems," *Int. J. Robust Nonlinear Control*, vol. 31, no. 1, pp. 148–165, Jan. 2021.

[10] D. Wang, S. Zhang, M. Gan, and J. Qiu, "A novel EM identification method for Hammerstein systems with missing output data," *IEEE Trans. Ind. Informat.*, vol. 16, no. 4, pp. 2500–2508, Apr. 2020.

[11] G.-Y. Chen, M. Gan, C. L. P. Chen, and H.-X. Li, "Basis function matrix-based flexible coefficient autoregressive models: A framework for time series and nonlinear system modeling," *IEEE Trans. Cybern.*, vol. 51, no. 2, pp. 614–623, Feb. 2021, doi: 10.1109/TCYB.2019.2900469.

[12] D. Wang, Y. Yan, Y. Liu, and J. Ding, "Model recovery for Hammerstein systems using the hierarchical orthogonal matching pursuit method," *J. Comput. Appl. Math.*, vol. 345, pp. 135–145, Jan. 2019.
[13] M. Gan, G. Y. Chen, L. Chen, and C. L. P. Chen, "Term selection for a class of separable nonlinear models," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 31, no. 2, pp. 445–451, Feb. 2020, doi: 10.1109/TNNLS.2019.2904952.
[14] F. Ding, X. Liu, and T. Hayat, "Hierarchical least squares identification for feedback nonlinear equation-error systems," *J. Franklin Inst.*, vol. 357, no. 5, pp. 2958–2977, Mar. 2020.
[15] F. Ding, X. Zhang, X. Lu, X.-S. Zhan, A. Alsaedi, and T. Hayat, "Hierarchical extended least squares estimation approaches for a multi-input multi-output stochastic system with colored noise from observation data," *J. Franklin Inst.*, vol. 357, no. 15, pp. 11094–11110, Oct. 2020.
[16] J. Chen, F. Ding, Q. Zhu, and Y. Liu, "Interval error correction auxiliary model based gradient iterative algorithms for multirate ARX models," *IEEE Trans. Autom. Control*, vol. 65, no. 10, pp. 4385–4392, Oct. 2020.
[17] J. Chen, Q. Zhu, and Y. Liu, "Modified Kalman filtering based multi-step-length gradient iterative algorithm for ARX models with random missing outputs," *Automatica*, vol. 118, Aug. 2020, Art. no. 109034.
[18] D. Wang, L. Mao, and F. Ding, "Recasted models-based hierarchical extended stochastic gradient method for MIMO nonlinear systems," *IET Control Theory Appl.*, vol. 11, no. 4, pp. 476–485, Feb. 2017.
[19] F. Giri, Y. Rochdi, and F. Z. Chaoui, "Hammerstein systems identification in presence of hard nonlinearities of preload and dead-zone type," *IEEE Trans. Autom. Control*, vol. 54, no. 9, pp. 2174–2178, Sep. 2009.
[20] Y. Saad, *Iterative Methods for Sparse Linear Systems*. Philadelphia, PA, USA: Society for Industrial and Applied Mathematics, 2003.
[21] K. Chakrabarti, N. Gupta, and N. Chopra, "Iterative pre-conditioning to expedite the gradient-descent method," in *Proc. Amer. Control Conf. (ACC)*, Denver, CO, USA, Jul. 2020, pp. 3977–3982.
[22] H. Li, B. Jin, and W. Yan, "Distributed model predictive control for linear systems under communication noise: Algorithm, theory and implementation," *Automatica*, vol. 125, Mar. 2021, Art. no. 109422, doi: 10.1016/j.automatica.2020.109422.
[23] L. Chunmao and X. Jian, "Adaptive delay estimation and control of networked control systems," in *Proc. Int. Symp. Commun. Inf. Technol.*, Bangkok, Thailand, Oct. 2006, pp. 707–710.
[24] A. S. Matveev and A. V. Savkin, *Estimation and Control Over Communication Networks*. Basel, Switzerland: Birkhäuser, 2008.
[25] J. Chen, B. Huang, F. Ding, and Y. Gu, "Variational Bayesian approach for ARX systems with missing observations and varying time-delays," *Automatica*, vol. 94, pp. 194–204, Aug. 2018.
[26] R. B. Gopaluni, "A particle filter approach to identification of nonlinear processes under missing observations," *Can. J. Chem. Eng.*, vol. 86, no. 6, pp. 1081–1092, 2008.
[27] X. Zhang, Q. Liu, F. Ding, A. Alsaedi, and T. Hayat, "Recursive identification of bilinear time-delay systems through the redundant rule," *J. Franklin Inst.*, vol. 357, no. 1, pp. 726–747, Jan. 2020.
[28] J. Chen, J. Ma, Y. Liu, and F. Ding, "Identification methods for time-delay systems based on the redundant rules," *Signal Process.*, vol. 137, pp. 192–198, Aug. 2017.

LIXIN LV received the B.Sc. and M.Sc. degrees from the Xi'an University of Science and Technology, Xi'an, China, in 1988 and 1990, respectively. He is currently an Associate Professor with the School of Internet of Things Engineering, Wuxi Professional College of Science and Technology, Wuxi, China. His current research interests include machine learning, adaptive control, and artificial intelligence.

JIAN ZHANG received the Ph.D. degree in measuring and testing technologies and instruments from the Harbin University of Science and Technology, Harbin, China, in 2008. He is currently an Associate Professor with the Wuxi Professional College of Science and Technology, Wuxi, China. His current research interests include image processing, encryption, and blockchain.