

Gradient based particle filter algorithm for an ARX model with nonlinear communication output

Jing Chen, Yanjun Liu, Feng Ding, Quanmin Zhu

Abstract—A stochastic gradient based particle filter algorithm is developed for an ARX model with nonlinear communication output in this paper. This non-standard ARX model consists of two submodels, one is a linear ARX model and the other is a nonlinear output model. The process outputs (outputs of the linear submodel) transmitted over a communication channel are unmeasurable, while the communication outputs (outputs of the nonlinear submodel) are available, and both of the two-type outputs are contaminated by white noises. Based on the rich input data and the available communication output data, a stochastic gradient based particle filter algorithm is proposed to estimate the unknown process outputs and parameters of the ARX model. Furthermore, a direct weight optimization method and the Epanechnikov kernel method are extended to modify the particle filter when the measurement noise is a Gaussian noise with unknown variance and the measurement noise distribution is unknown. The simulation results demonstrate that the stochastic gradient based particle filter algorithm is effective.

Index Terms—Parameter estimation, stochastic gradient, particle filter, auxiliary model, ARX model.

I. INTRODUCTION

The non-standard ARX model which is a standard ARX model mixed with a nonlinear communication output model, can be seen as a networked control system [1], [2]. Its basic structure is shown in Fig. 1, which consists of two submodels, one is an ARX model given by

$$A(d)x(t) = B(d)u(t) + v(t),$$

and the other is a nonlinear communication output model written as

$$y(t) = f(x(t)) + e(t),$$

where $u(t)$ is the input, $x(t)$ is the output of the ARX model and influenced by the process noise $v(t)$, $y(t)$ is the output of the communication network and influenced by the measurement noise $e(t)$, $u(t)$ and $y(t)$ are measurable, while $x(t)$ is unmeasurable, $f(\cdot)$ is a known continuous nonlinear function, $v(t)$ and $e(t)$ are independent and identically distributed Gaussian noises with variances ζ^2 and σ^2 , respectively, and $A(d)$ and $B(d)$ are polynomials which can be expressed as

$$\begin{aligned} A(d) &:= 1 + a_1d^{-1} + \dots + a_nd^{-n}, \\ B(d) &:= b_1d^{-1} + b_2d^{-2} + \dots + b_md^{-m}, \end{aligned}$$

where $d^{-1}x(t) = x(t-1)$. Different from the traditional non-

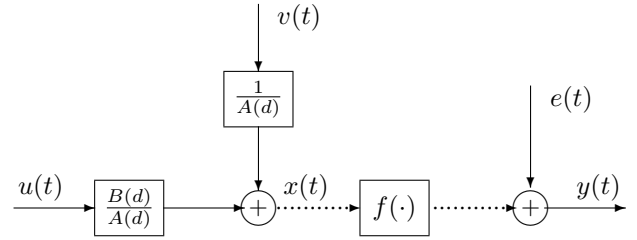


Fig. 1. The ARX model with nonlinear communication output

linear models, such as the Hammerstein model, the Wiener model or the Hammerstein-Wiener-Hammerstein model, the non-standard ARX model has two submodels and has two kinds noises, thus the identification for this model is more difficult. Recently, this type of model is usually considered in process control and estimation literature [3], [4]. Under such a scenario, the process outputs are often sampled by some sensors and are typically sent to a laboratory for analysis. Due to the complexities of communication network and laboratory analysis process, these outputs as the laboratory analysis outcomes can have nonlinear properties as well as their own errors from the laboratory analysis. The focus of this paper is to develop an on-line identification algorithm for the non-standard ARX model whose measurement noise has different properties.

The ARX model identification has been extensively studied in theory [5], [6]. Many identification methods, such as the recursive least squares (RLS) algorithms [7], the hierarchical algorithms [8], the stochastic gradient (SG) algorithms [9] and the iterative algorithms [10], have been proposed for ARX models. The hierarchical algorithms decompose the complex model into several submodels with small dimensions and then coordinate the associate terms between submodels [11]. The iterative algorithms are off-line algorithms which have heavy computational efforts and cannot be used to estimate the parameters recursively by new data, while the RLS and SG algorithms can be used as on-line algorithms and can update the parameters based on new data. Furthermore, these on-line algorithms have less computational efforts, thus widely used in engineering application.

The on-line algorithms often have the assumption that all the system data are available. However, in engineering practice, some data of the systems may be missing or unmeasurable. The auxiliary model is an important tool which is usually applied for systems with missing data [12]. Its basic idea is to replace the missing data with the outputs of an auxiliary model. For example, Wang and Ding studied an auxiliary model based RLS algorithm for multivariable systems [13]. Jin et al investigated an auxiliary model based identification algorithm for multivariable OE-like systems with missing outputs [12]. Such methods only use an auxiliary model to predict the missing outputs, while the measurable outputs are not applied to improve the predicted outputs, which makes the auxiliary model have slow convergence rates.

With the development of communication networks, control systems often use network to transmit the sampled outputs via a communication channel [14]–[16]. However, the complex network and laboratory

J. Chen is with School of Science, Jiangnan University, Wuxi 214122, PR China (chenjing1981929@126.com)

Y.J. Liu and F. Ding are with Key Laboratory of Advanced Process Control for Light Industry (Ministry of Education), Jiangnan University, Wuxi 214122, PR China (yanjunliu_1983@126.com, fding@jiangnan.edu.cn)

Q.M. Zhu is with Department of Engineering Design and Mathematics, University of the West of England, Bristol BS16 1QY, UK (quan.zhu@uwe.ac.uk)

This work is supported by the National Natural Science Foundation of China (No. 61403165) and the Natural Science Foundation for Colleges and Universities in Jiangsu Province (No. 16KJB120006)

analysis process also bring some identification constraints, such as the packet dropouts, the communication time-delays, the nonlinear signals and the unmeasurable sampled outputs [17]–[19]. There exist many identification methods for systems with such constraints. For example, Guo et al proposed an identification algorithm for FIR systems, in which the outputs are constrained by both the binary-valued quantization and the communication unreliability [20]. Xiong et al investigated an EM algorithm for nonlinear systems with unmeasurable outputs, in which those unmeasurable outputs were estimated by an auxiliary model [21]. Xie et al developed an EM algorithm for multi-rate systems with random time-delays, where the output model is linear and the process model is not influenced by process noise, by using an auxiliary model and an off-line algorithm, the missing outputs and the parameters were estimated simultaneously [22].

The Kalman filter and the particle filter are two effective tools which are often applied for state-space systems [23]–[25]. In [26], a Kalman filter based off-line algorithm is derived for state-space systems, and in order to enhance the computational efficiency, a model decomposition based off-line algorithm is also provided. In [3], an EM based particle filter algorithm is proposed for nonlinear parameter varying state-space systems, where the proposed EM algorithm is an off-line algorithm. Unlike the auxiliary model, the Kalman filter and the particle filter apply the measurable data to improve the priori estimates, which makes them be more accurate. However, these two filters are often used for state-space systems.

This paper takes the above described literature into study and develops an SG based particle filter algorithm for an ARX model with nonlinear communication output. The considered system in this paper is a non-state-space system with different kinds of measurement noises, and the proposed algorithm is an on-line algorithm. A particle filter is derived to estimate the process outputs by the communication outputs. Then based on the estimated process output data and the rich input data, the unknown parameters can be estimated by an SG algorithm. The problem discussed in this paper is more complicated and challenging than those in [3], [4], [22] and the main contributions are listed below.

- 1) Study an on-line identification method for a non-standard ARX model which consists of two submodels, one is the ARX model to reflect process dynamics, and the other is an equation reflecting nonlinear output.
- 2) Propose a particle filter instead of an auxiliary model for nonlinear non-state-space systems, which can estimate the unknown variables and can increase the estimation accuracy.
- 3) Apply the direct weight optimization method and the Epanechnikov kernel method to modify the particle filter for the situations that the measurement noise with unknown variance and unknown distribution.

Briefly, the rest of this paper is organized as follows. Section II presents an SG based particle filter algorithm. Section III develops two modified particle filters. In Section IV, three illustrative examples are provided. Section V concludes this paper and gives future directions.

II. THE SG BASED PARTICLE FILTER ALGORITHM

For the model depicted in Fig. 1, define the information vector $\varphi(t)$ and the parameter vector θ as

$$\begin{aligned} \varphi(t) &:= [-x(t-1), \dots, -x(t-n), u(t-1), \dots, \\ &\quad u(t-m)]^T \in \mathbb{R}^{n+m}, \\ \theta &:= [a_1, \dots, a_n, b_1, \dots, b_m]^T \in \mathbb{R}^{n+m}, \end{aligned}$$

then the non-standard ARX model can be rewritten as follows,

$$x(t) = \varphi^T(t)\theta + v(t), \quad (1)$$

$$y(t) = f(x(t)) + e(t). \quad (2)$$

In Equation (1), the information vector contains unknown process outputs $x(i)$, $i = 1, 2, \dots$, thus the classical SG algorithm cannot be

applied to (1) directly. The auxiliary model based algorithms are often used to identify systems with unmeasurable variables [27], [28]. For example, the corresponding auxiliary model based SG (AM-SG) algorithm for the system model in (1) and (2) is below,

$$\hat{\theta}(t) = \hat{\theta}(t-1) + \frac{\hat{\varphi}(t)}{\nu(t)} \epsilon(t), \quad (3)$$

$$\hat{\varphi}(t-i) = [-\hat{x}(t-i-1), \dots, -\hat{x}(t-i-n), u(t-i-1), \dots, \\ u(t-i-m)]^T, \quad (4)$$

$$\hat{x}(t-i) = \hat{\varphi}^T(t-i)\hat{\theta}(t-i-1), \quad (5)$$

$$\epsilon(t) = x(t) - \hat{\varphi}^T(t)\hat{\theta}(t-1), \quad (6)$$

$$\nu(t) = \nu(t-1) + \|\hat{\varphi}(t)\|^2, \quad \nu(0) = 1, \quad (7)$$

in which $\hat{\theta}(t-i-1)$ is the estimate of θ at $t-i-1$. However, here the SG-AM algorithm cannot be applied to deal with this non-standard model.

Remark 1: Unlike the models in [27], [28], all the process outputs in this non-standard ARX model are unknown, and if we use $\hat{x}(t) = \hat{\varphi}^T(t)\hat{\theta}(t-1)$ to replace $x(t)$, then $\epsilon(t)$ in Equation (6) becomes zero, thus the SG-AM algorithm proposed in Equations (3)-(7) is invalid for this non-standard ARX model.

The key characteristic of our treatment is to develop a particle filter based on-line algorithm for this non-standard ARX model.

A. The particle filter

The particle filter is a tool for nonlinear state-space systems with unmeasurable states. In what follows we will extend this tool to nonlinear non-state-space systems.

The basic idea of the particle filter is to use a series of particles with associated weights to approximate the posterior density function. Based on the particle filter method in [3], [29], the posterior density function $p(x(t-i)|y(t-i), \dots, y(1), x(t-i-1), \dots, x(1), u(t-i-1), \dots, u(1))$ can be approximately as

$$\begin{aligned} &p(x(t-i)|y(t-i), \dots, y(1), x(t-i-1), \dots, x(1), \\ &\quad u(t-i-1), \dots, u(1), \hat{\theta}(t-i-1)) \\ &\approx \sum_{j=1}^N \omega_j \delta(x(t-i) - \hat{x}_j(t-i)), \end{aligned} \quad (8)$$

where ω_j is the normalized weight associated with the j th particle and $\delta(\cdot)$ denotes the Dirac delta function. Once the particles and their weights have been obtained, we can compute any desired statistical measure of the posterior density function. For example, the mean of the posterior density function can be computed as

$$\hat{x}(t-i) = \sum_{j=1}^N \bar{\omega}_j \hat{x}_j(t-i), \quad (9)$$

in which $\bar{\omega}_j = \frac{\omega_j}{\sum_{j=1}^N \omega_j}$ is the weight associated with the j th particle

and $\sum_{j=1}^N \bar{\omega}_j = 1$, $\hat{x}_j(t-i)$ is the j th particle drawn from the posterior density function $p(x(t-i)|y(t-i), \dots, y(1), x(t-i-1), \dots, x(1), u(t-i-1), \dots, u(1), \theta)$. However, it is difficult to draw particles from this posterior density function directly. In order to get the new particles, an importance density $q(\cdot)$ is given by,

$$\begin{aligned} &q(x(t-i)|y(t-i), \dots, y(1), x(t-i-1), \dots, x(1), \\ &\quad u(t-i-1), \dots, u(1), \hat{\theta}(t-i-1)) \\ &= p(x(t-i)|x(t-i-1), \dots, x(1), \\ &\quad u(t-i-1), \dots, u(1), \hat{\theta}(t-i-1)). \end{aligned} \quad (10)$$

From Equations (1) and (10), one gets

$$q(x(t-i)|y(t-i), \dots, y(1), x(t-i-1), \dots, x(1),$$

$$u(t-i-1), \dots, u(1), \hat{\boldsymbol{\theta}}(t-i-1)) \\ = \frac{1}{\sqrt{2\pi\varsigma}} \exp \left[-\frac{(x(t-i) - \boldsymbol{\varphi}^T(t-i)\hat{\boldsymbol{\theta}}(t-i-1))^2}{2\varsigma^2} \right]. \quad (11)$$

Remark 2: If the variance ς is unknown, then the density function in Equation (11) would not be computed, and the particles could not be updated. In this scenario, the auxiliary model can be applied to draw the particles, e.g.,

$$\hat{x}_j(t-i) = \hat{\boldsymbol{\varphi}}_j^T(t-i)\hat{\boldsymbol{\theta}}(t-i-1), \\ \hat{\boldsymbol{\varphi}}_j(t-i) = [-\hat{x}_j(t-i-1), \dots, -\hat{x}_j(t-i-n), \\ u(t-i-1), \dots, u(t-i-m)]^T,$$

where $\hat{\boldsymbol{\theta}}(t-i-1)$ is the estimate of $\boldsymbol{\theta}$ at time $t-i-1$.

From Equation (11), we can get the new particles. The next is to update the weight of each new particle. Based on [3], the weight $\omega_j(t)$ can be updated as follow,

$$\omega_j(t) = p(y(t)|\hat{x}_j(t))\omega_j(t-1). \quad (12)$$

According to Equation (2), the density function $p(y(t)|\hat{x}_j(t))$ can be computed by

$$p(y(t)|\hat{x}_j(t)) = \frac{1}{\sqrt{2\pi\sigma}} \exp \left[-\frac{(y(t) - f(\hat{x}_j(t)))^2}{2\sigma^2} \right]. \quad (17)$$

Substituting (17) into (16) gets

$$\omega_j(t) = \frac{1}{\sqrt{2\pi\sigma}} \exp \left[-\frac{(y(t) - f(\hat{x}_j(t)))^2}{2\sigma^2} \right] \omega_j(t-1). \quad (18)$$

Normalizing $\omega_j(t)$ yields

$$\bar{\omega}_j(t) = \frac{\omega_j(t)}{\sum_{j=1}^N \omega_j(t)}. \quad (19)$$

Then we have

$$\hat{x}(t-i) = \sum_{j=1}^N \bar{\omega}_j(t-i)\hat{x}_j(t-i). \quad (20)$$

When apply the particle filter to estimate the unknown variables, the degeneracy phenomenon is an inevitable problem [30]. After a few iterations, the weights of some particles become small, which means that heavy computational efforts are utilized to update those particles whose contributions are negligible. In order to avoid this phenomenon, a sample size N_{eff} is defined as,

$$N_{eff} := \left[\sum_{j=1}^N \bar{\omega}_j(t)^2 \right]^{-1}. \quad (21)$$

Given a threshold N_{hold} in prior, $N_{eff} < N_{hold}$ means severe degeneracy. Thus we should use resampling, and then the weight of each particle is assigned as $\bar{\omega}_j = \frac{1}{N}$.

The particle filter proceeds by performing the following steps.

- 1) Initialization: At time t , collect the input-output data $\{u(1), y(1), \dots, u(t), y(t)\}$. Draw N initial particles $\{\hat{x}_j(0)\}_{j=1}^N$ from $p(x(0)|\hat{\boldsymbol{\theta}}(t-1))$ and set $\{\omega_j(0)\}_{j=1}^N = \frac{1}{N}$, $\hat{\boldsymbol{\theta}}(t-1)$ is the estimate of $\boldsymbol{\theta}$ at time $t-1$.
- 2) Let the process noise variance be ς^2 and the measurement noise variance be σ^2 , and assign a positive number N_{hold} .
- 3) Let $i = t-1$.
- 4) Compute $\{\hat{x}_j(t-i)\}_{j=1}^N$ from Equation (11).
- 5) Compute $\omega_j(t-i)$ by Equation (18).
- 6) Normalize $\bar{\omega}_j(t-i)$ from Equation (19).
- 7) Compute the missing output $\hat{x}(t-i)$ by Equation (20).
- 8) Compute N_{eff} by Equation (21) and compare N_{eff} with N_{hold} , if $N_{eff} < N_{hold}$, reset the weight of each particle with $\bar{\omega}_j(t-i) = \frac{1}{N}$ and go to next step; otherwise, go to next step.
- 9) Let $i = i-1$, if $i \geq 0$ go to step 4; otherwise terminate the procedure.

B. The identification algorithm

In order to estimate $x(t)$, two methods are discussed here. One is to keep the estimates of the unknown process outputs at time $t-i-2$ unchanged, while we only use $\hat{\boldsymbol{\theta}}(t-i-1)$ to estimate $x(t-i)$.

Draw N particles from the following importance density function,

$$q(x(t-i)|y(t-i), \dots, y(1), \hat{x}(t-i-1), \dots, \hat{x}(1), \\ u(t-i-1), \dots, u(1), \hat{\boldsymbol{\theta}}(t-i-1)), \quad (22)$$

in which

$$\hat{x}(t-i-1) = [x(t-i-1)]|_{\hat{\boldsymbol{\theta}}(t-i-2)}, \\ \vdots \\ \hat{x}(1) = [x(1)]|_{\hat{\boldsymbol{\theta}}(0)},$$

where $[x(t-i-j)]|_{\hat{\boldsymbol{\theta}}(t-i-j-1)}$, $j = 1, \dots, i-1$ means the estimate of $x(t-i-j)$ by using $\hat{\boldsymbol{\theta}}(t-i-j-1)$. Based on Equation (20), we can get the estimate $\hat{x}(t-i)$. Then with the estimates $\{\hat{x}(t-i), \hat{x}(t-i-1), \dots, \hat{x}(1)\}$ and $\hat{\boldsymbol{\theta}}(t-i-1)$, one can get the parameter estimate $\hat{\boldsymbol{\theta}}(t-i)$.

From (22), we can see that only one process output data $x(t-i)$ is required to be estimated at each sampling time $t-i-1$.

Unlike the first method, in order to get the parameter estimates at $t-i$, the second method is to estimate all the unknown process outputs up to and including time $t-i$ by using the parameter estimate $\hat{\boldsymbol{\theta}}(t-i-1)$.

Draw N particles from

$$q(x(t-i)|y(t-i), \dots, y(1), \hat{x}(t-i-1), \dots, \hat{x}(1), \\ u(t-i-1), \dots, u(1), \hat{\boldsymbol{\theta}}(t-i-1)), \quad (23)$$

in which

$$\hat{x}(t-i-1) = [x(t-i-1)]|_{\hat{\boldsymbol{\theta}}(t-i-1)}, \\ \vdots \\ \hat{x}(1) = [x(1)]|_{\hat{\boldsymbol{\theta}}(t-i-1)}.$$

From Equation (20), we can obtain the estimates $\hat{x}(t-i)$. Then according to the estimates $\{\hat{x}(t-i), \hat{x}(t-i-1), \dots, \hat{x}(1)\}$ and $\hat{\boldsymbol{\theta}}(t-i-1)$, we can get the updated parameter vector $\hat{\boldsymbol{\theta}}(t-i)$.

The density function in (23) declares that one should estimate $t-i$ process outputs at each sampling time $t-i-1$.

Remark 3: The estimates of the process output data by using the second method are more accurate than those by using the first method. However, compared with the first method, the second method has heavier computational efforts.

In order to estimate the unmeasurable data more accurately, we use the second method in this paper. Then the SG based particle filter (SG-PF) algorithm is given by,

$$\hat{\boldsymbol{\theta}}(t) = \hat{\boldsymbol{\theta}}(t-1) + \frac{\hat{\boldsymbol{\varphi}}(t)}{r(t)} [\hat{x}(t) - \hat{\boldsymbol{\varphi}}^T(t)\hat{\boldsymbol{\theta}}(t-1)], \quad (24)$$

$$\hat{\boldsymbol{\varphi}}(t) = [-\hat{x}(t-1), \dots, -\hat{x}(t-n), u(t-1), \dots, \\ u(t-m)]^T, \quad (25)$$

$$\hat{x}(t-i) = \sum_{j=1}^N \bar{\omega}_j(t-i)\hat{x}_j(t-i), \quad (26)$$

$$r(t) = r(t-1) + \|\hat{\boldsymbol{\varphi}}(t)\|^2, \quad r(0) = 1. \quad (27)$$

The SG-PF algorithm proceeds by performing the following steps.

- 1) Let $u(t) = 0, \hat{x}(t) = 0, y(t) = 0, t \leq 0$, and give a small positive number ϵ .
- 2) Let $t = 1, \hat{\boldsymbol{\theta}}(0) = [1/p_0, \dots, 1/p_0]^T \in \mathbb{R}^{n+m}$, where $p_0 = 10^6$.
- 3) Collect the input-output data $\{u(t), y(t)\}$.

$$\begin{aligned}
\omega_j(t) &= \frac{p(\hat{X}_j(t), Y(t), \hat{\theta}(t-1))}{q(\hat{X}_j(t)|Y(t), \hat{\theta}(t-1))} \\
&= \frac{p(\hat{x}_j(t), y(t), \hat{X}_j(t-1), Y(t-1), \hat{\theta}(t-1))}{q(\hat{x}_j(t)|\hat{X}_j(t-1), Y(t), \hat{\theta}(t-1))q(\hat{X}_j(t-1)|Y(t), \hat{\theta}(t-1))} \\
&= \frac{p(y(t)|\hat{x}_j(t), \hat{X}_j(t-1), Y(t-1), \hat{\theta}(t-1))p(\hat{x}_j(t)|\hat{X}_j(t-1), Y(t-1), \hat{\theta}(t-1))p(\hat{X}_j(t-1), Y(t-1), \hat{\theta}(t-1))}{q(\hat{x}_j(t)|\hat{X}_j(t-1), Y(t), \hat{\theta}(t-1))q(\hat{X}_j(t-1)|Y(t-1), \hat{\theta}(t-1))} \\
&= \frac{p(y(t)|\hat{x}_j(t), \hat{\theta}(t-1))p(x_j(t)|\hat{x}_j(t-1), \dots, \hat{x}_j(1), u(t-1), \dots, u(1), \hat{\theta}(t-1))}{q(x_j(t)|y(t), \dots, y(1), \hat{x}_j(t-1), \dots, \hat{x}_j(1), u(1), u(2), \dots, u(t-1), \hat{\theta}(t-1))} \omega_j(t-1) \\
&= p(y(t)|\hat{x}_j(t))\omega_j(t-1).
\end{aligned} \tag{16}$$

- 4) Update $\bar{\omega}_j(t-i)$ and $\hat{x}_j(t-i)$, $j = 1, \dots, N$ by using the particle filter.
- 5) Compute $\hat{x}(t-i)$, $i = t-1, t-2, \dots, 0$ by Equation (26).
- 6) Form $\hat{\varphi}(t)$ by Equation (25).
- 7) Compute $r(t)$ by Equation (27).
- 8) Estimate $\hat{\theta}(t)$ by Equation (24).
- 9) Let $\tau = \|\hat{\theta}(t) - \hat{\theta}(t-1)\|/\|\hat{\theta}(t)\|$, if $\tau \leq \varepsilon$, then obtain the estimated parameter vector $\hat{\theta}(t)$; otherwise, let $t = t+1$ and go back to step 3.

Remark 4: From the SG-PF algorithm in (24)-(27), we can see that the unknown process output is estimated by Equation (26), obviously, $\hat{x}(t) - \hat{\varphi}^T(t)\hat{\theta}(t-1) \neq 0$. Thus the SG-PF algorithm can be utilized to estimate the unknown parameters.

III. TWO MODIFIED PARTICLE FILTERS

In the above section, we assume that the measurement noise is a Gaussian noise with known variance. However, in engineering practice, the variance of the Gaussian noise is often unknown. Moreover, the distribution of the measurement noise may be unknown. In this section, we consider the cases that the measurement noise is a Gaussian noise with unknown variance and the measurement noise distribution is unknown, and develop two modified particle filters to estimate the process outputs. .

A. The Gaussian measurement noise with unknown variance

Since the variance of the measurement noise is unknown, the weight $w_j(t)$ in Equation (18) cannot be computed directly. In order to circumvent this difficulty, we introduce a direct weight optimization method to update the weight of each particle.

Draw N particles $\hat{x}_j(t)$, $j = 1, \dots, N$ from (23) and define

$$p(y(t)|\hat{x}_j(t)) := \lambda_j(t).$$

The next is to find the weights $\lambda_j(t)$, $j = 1, \dots, N$ so that the nonlinear function $f(x(t))$ can be approximated by

$$f(x(t)) \approx \sum_{j=1}^N \lambda_j(t)(f(\hat{x}_j(t))),$$

in which $\sum_{j=1}^N \lambda_j(t) = 1$, $\lambda_j(t) \geq 0$. Based on the direct weight optimization method in [31], the weights $\lambda_j(t)$, $j = 1, \dots, N$ can be obtained by minimizing the following function

$$\left[\sum_{j=1}^N \lambda_j(t)(f(\hat{x}_j(t)) + e(t)) - y(t) \right]^2$$

$$= \left(\sum_{j=1}^N \lambda_j(t)f(\hat{x}_j(t)) + \sum_{j=1}^N \lambda_j(t)e(t) - \sum_{j=1}^N \lambda_j(t)y(t) \right)^2. \tag{28}$$

Define

$$|y(t) - f(\hat{x}_j(t))| := \gamma_j(t).$$

Taking the conditional expectation on (28) yields

$$\begin{aligned}
&E \left[\sum_{j=1}^N \lambda_j(t)(f(\hat{x}_j(t)) + e(t)) - y(t) \right]^2 \\
&= \left(\sum_{j=1}^N \lambda_j(t)\gamma_j(t) \right)^2 + \sigma^2 \sum_{j=1}^N \lambda_j^2(t).
\end{aligned} \tag{29}$$

Let

$$z(t) = \sum_{j=1}^N \lambda_j(t)\gamma_j(t) + \sum_{j=1}^N |\lambda_j(t)e(t)|.$$

According to the direct weight optimization method proposed in [31], we can find the weights to make sure that $Prob(z \geq \gamma(t))$ is minimized. In this paper, we choose $\gamma(t)$ as

$$\gamma(t) = \max\{\gamma_j(t), j = 1, \dots, N\} + 1.$$

The choice of $\gamma(t)$ can keep the number of particles unchanged and can prevent sample impoverishment. Since the measurement noise $e(t)$ is a Gaussian noise, the density function of z can be expressed as

$$p(z) = \begin{cases} \frac{2}{\sqrt{(2\pi)\zeta^2}} \exp\left(-\frac{(z-z_0)^2}{2\zeta^2}\right), & z \geq z_0, \\ 0, & \end{cases} \tag{30}$$

where

$$z_0 = \sum_{j=1}^N \lambda_j(t)\gamma_j(t), \quad \zeta = \sigma \sqrt{\sum_{j=1}^N \lambda_j^2(t)}.$$

Minimizing $Prob(z \geq \gamma(t))$ is equivalent to

$$\text{Max}_{\sum_{j=1}^N \lambda_j(t)=1, \lambda_j(t)} \frac{\gamma(t) - \sum_{j=1}^N \lambda_j(t)\gamma_j(t)}{\sum_{j=1}^N \lambda_j^2(t)}.$$

Then the weight can be computed by

$$\lambda_j(t) = \frac{\gamma(t) - \gamma_j(t)}{N\gamma(t) - \sum_{j=1}^N \gamma_j(t)}. \tag{31}$$

Substituting Equation (31) into Equation (16) gets

$$\omega_j(t) = \lambda_j(t)\omega_j(t-1),$$

and the normalized weight can be expressed as

$$\bar{\omega}_j(t) = \frac{\omega_j(t)}{\sum_{k=1}^N \omega_k(t)}. \quad (32)$$

Remark 5: Equations (31) and (32) imply that the weight of each particle can be obtained without the knowledge of the variance of the measurement noise.

B. The measurement noise with unknown distribution

When the distribution of the measurement noise is unknown, both the density function in Equations (17) and (30) cannot be obtained. In order to solve this problem, we use the Kernel density estimation method to update the weight of each particle.

Rewrite

$$\begin{aligned} |y(t) - f(\hat{x}_j(t))| &= \gamma_j(t), \\ \gamma(t) &= \max\{\gamma_j(t), j = 1, \dots, N\} + 1. \end{aligned}$$

Since $f(\cdot)$ is a continuous function, we can conclude that the smaller the $\gamma_j(t)$ is, the more important role of the j th particle at time t in estimating the true output $x(t)$ plays. The Epanechnikov function is a significant type of kernel function and is optimal in a mean square error sense. Here, the Epanechnikov function is introduced to estimate $p(y(t)|\hat{x}_j(t))$ [32].

Since

$$\frac{\gamma_j(t)}{\gamma(t)} < 1, \quad j = 1, \dots, N,$$

then the density function is given by

$$p(y(t)|\hat{x}_j(t)) = \frac{3}{4} \left[1 - \left(\frac{\gamma_j(t)}{\gamma(t)} \right)^2 \right].$$

By normalizing the density function, one has

$$p(y(t)|\hat{x}_j(t)) = \frac{\gamma^2(t) - \gamma_j^2(t)}{N\gamma^2(t) - \sum_{j=1}^N \gamma_j^2(t)}. \quad (33)$$

Substituting Equation (33) into Equation (16) gets

$$\omega_j(t) = \frac{\gamma^2(t) - \gamma_j^2(t)}{N\gamma^2(t) - \sum_{j=1}^N \gamma_j^2(t)} \omega_j(t-1).$$

Normalizing $\omega_j(t)$ yields

$$\bar{\omega}_j(t) = \frac{\omega_j(t)}{\sum_{j=1}^N \omega_j(t)}.$$

Remark 6: Assume $\gamma_s(t) < \gamma_l(t)$, $1 \leq s \leq N$, $1 \leq l \leq N$ and $s \neq l$. It follows that

$$\frac{\gamma^2(t) - \gamma_s^2(t)}{N\gamma^2(t) - \sum_{j=1}^N \gamma_j^2(t)} > \frac{\gamma^2(t) - \gamma_l^2(t)}{N\gamma^2(t) - \sum_{j=1}^N \gamma_j^2(t)}.$$

It means that the s th particle plays a more important role in estimating the estimate $\hat{x}(t)$ than the l th particle.

Theorem 1: Assume that $p(y(t)|\hat{x}_j(t))$ is expressed by Equation (33) and the distribution of the measurement noise is unknown, then the following inequality holds

$$\sum_{j=1}^N p(y(t)|\hat{x}_j(t))(y(t) - f(\hat{x}_j(t)))^2 \leq \sum_{j=1}^N \frac{1}{N} (y(t) - f(\hat{x}_j(t)))^2.$$

The detailed derivation is given in Appendix A.

Theorem 1 shows that the estimation accuracy can be improved by using the updated weights.

Remark 7: When the measurement noise is a Gaussian noise with unknown variance, the Kernel density estimation method also can be applied to update the weight of each particle. On the other hand, the direct weight optimization method cannot be utilized to update the weight of each particle when the distribution of the measurement noise is unknown.

Theorem 2: Let $\lambda_j^d(t)$ be the weight updated by Equation (31), $\lambda_j^k(t)$ be the weight updated by Equation (33). Assume that the measurement noise is a Gaussian noise with unknown variance, then

$$\sum_{j=1}^N \lambda_j^d(t)(y(t) - f(\hat{x}_j(t)))^2 \leq \sum_{j=1}^N \lambda_j^k(t)(y(t) - f(\hat{x}_j(t)))^2.$$

The proof is given in Appendix B.

Theorem 2 means that the direct weight optimization method is the optimal method when the measurement noise is a Gaussian noise with unknown variance.

IV. EXAMPLES

Example 1: Consider an ARX model with unknown outputs,

$$\begin{aligned} A(d)x(t) &= B(d)u(t) + v(t), \\ A(d) &= 1 + a_1d^{-1} = 1 + 0.49d^{-1}, \\ B(d) &= b_1d^{-1} + b_2d^{-2} = -0.83d^{-1} + 0.17d^{-2}, \end{aligned}$$

and the output model is

$$\begin{aligned} y(t) &= \exp(x(t)) + e(t), \\ u &\sim N(0, 1), \quad v \sim N(0, \zeta^2), \quad e \sim N(0, \sigma^2), \end{aligned}$$

in which $x(t)$ is unmeasurable, $y(t)$ is measurable, the variances of $\{v(t)\}$ and $\{e(t)\}$ are both 0.10^2 . In simulation, we choose 100 particles.

Apply the SG-PF algorithm to estimate the parameters of the ARX model. The simulation data of the inputs, process outputs and nonlinear communication outputs are shown in Fig. 2 (samples 1-200). The parameter estimation errors $\tau := \|\hat{\theta} - \theta\|/\|\theta\|$ versus t are depicted in Fig. 3, and the parameter estimates and their errors are presented in Table I. The estimated process outputs, the true process outputs and their errors are shown in Fig. 4 (samples 800-900).

TABLE I
THE SG-PF ALGORITHM ESTIMATES AND ERRORS IN EXAMPLE 1

t	a_1	b_1	b_2	τ (%)
100	0.30121	-0.82158	0.03847	23.52512
200	0.37672	-0.82428	0.09128	14.10643
400	0.46706	-0.82416	0.12997	4.75154
600	0.48365	-0.82306	0.15922	1.46191
800	0.48532	-0.84526	0.17290	1.65736
1000	0.48776	-0.82236	0.16352	1.04817
True Values	0.49000	-0.83000	0.17000	

Example 2: Consider the non-standard ARX model proposed in Example 1.

$$\begin{aligned} A(d)x(t) &= B(d)u(t) + v(t), \\ y(t) &= \exp(x(t)) + e(t), \\ u &\sim N(0, 1), \quad v \sim N(0, \zeta^2). \end{aligned}$$

The variance of $\{v(t)\}$ is 0.10^2 , while the distribution of $\{e(t)\}$ is unknown. In simulation, we also choose 100 particles.

Apply the Epanechnikov function to improve the weights of the 100 particles and the SG-PF algorithm to identify the ARX model.

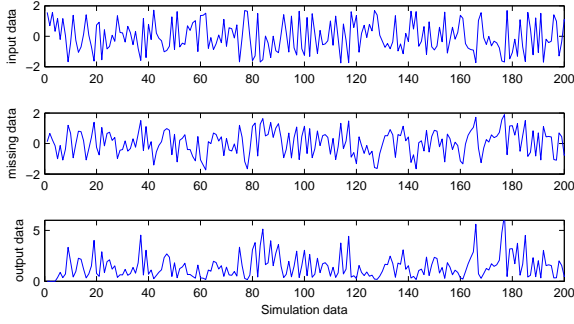


Fig. 2. The simulation data in Example 1

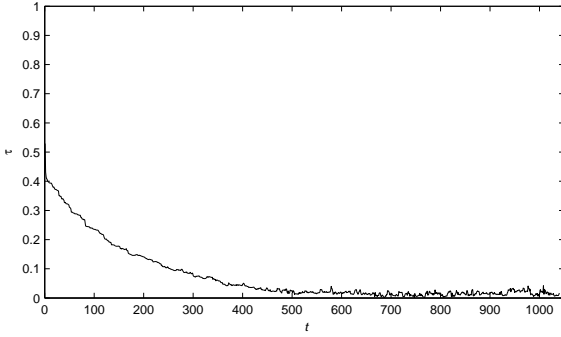


Fig. 3. The parameter estimation errors τ versus t in Example 1

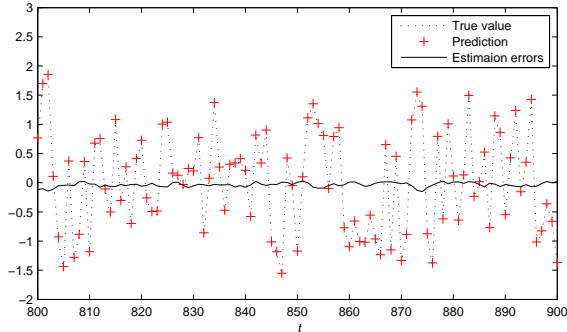


Fig. 4. Estimated model outputs, true outputs and estimation errors in Example 1

The estimation errors τ versus t are depicted in Fig. 5, and the parameter estimates and their errors are presented in Table II. The estimated process outputs, the true process outputs and their errors are shown in Fig. 6 (samples 800-900).

From Examples 1 and 2, we can get the following finds.

- 1) Figs. 3 and 5 show that the parameter estimation errors of these two methods decay and finally vanish when t is increased.
- 2) Figs. 4 and 6 declare that the estimated outputs by the SG-PF algorithm can track the true missing outputs.
- 3) Values in Tables I and II witness that the parameter estimation errors through the SG-PF algorithm with known measurement noise are smaller than those through the SG-PF algorithm with unknown measurement noise.

Example 3: Consider an experiment setup of a water tank system in Fig. 7, where $u(t)$ is the valve opening, and $x(t)$ is the liquid level. There is a pressure sensor at the bottom of Tank 2 which can transmit the liquid level of Tank 2 over a communication network.

TABLE II
THE SG-PF ALGORITHM ESTIMATES AND ERRORS IN EXAMPLE 2

t	a_1	b_1	b_2	τ (%)
100	0.23461	-0.82651	-0.01625	32.29839
200	0.29956	-0.82358	0.03349	23.94949
400	0.39711	-0.82158	0.09146	12.45844
600	0.44787	-0.82502	0.13561	5.57991
800	0.46981	-0.83960	0.15788	2.59833
1000	0.47811	-0.82710	0.15714	1.81366
True Values	0.49000	-0.83000	0.17000	

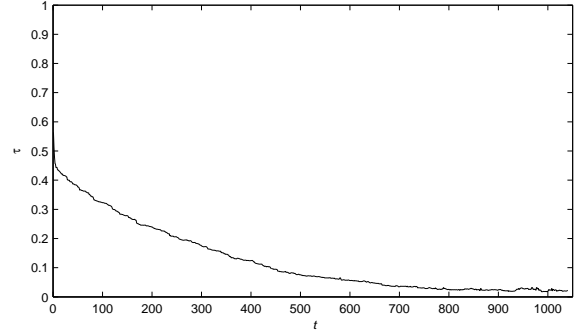


Fig. 5. The parameter estimation errors τ versus t in Example 2

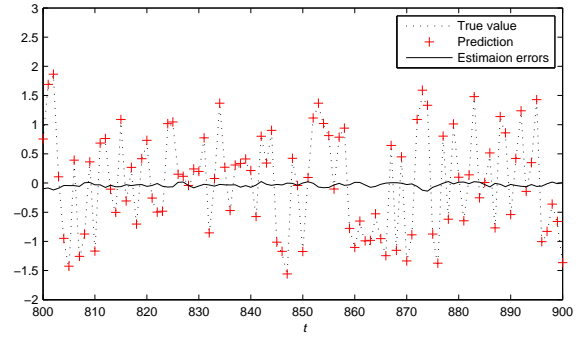


Fig. 6. Estimated model outputs, true outputs and estimation errors in Example 2

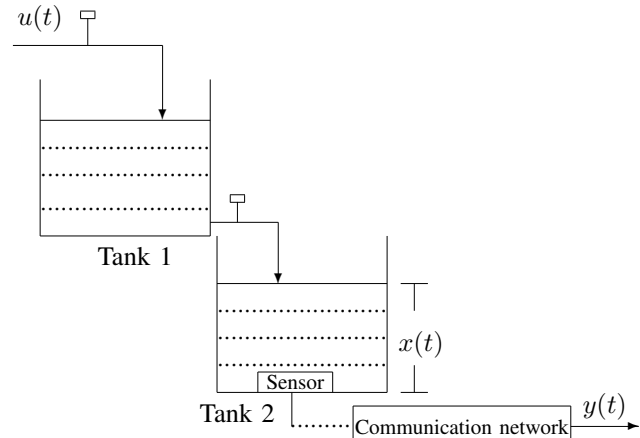


Fig. 7. A water tank system

The measurable output is $y(t)$. The system can be expressed by the

following ARX model,

$$\begin{aligned} A(d)x(t) &= B(d)u(t) + v(t), \\ A(d) &= 1 + a_1d^{-1} + a_2d^{-2} = 1 + 0.47d^{-1} + 0.18d^{-2}, \\ B(d) &= b_1d^{-1} + b_2d^{-2} = -0.82d^{-1} + 0.31d^{-2}, \end{aligned}$$

and we manually amplify the output as

$$y(t) = x(t) + 0.2x^2(t) + e(t).$$

This kind of nonlinear function can be seen in [21], [33]. In simulation, the input $\{u(t)\}$ is a filtered random binary signal sequence, $\{v(t)\}$ and $\{e(t)\}$ are Gaussian white noise sequences and both of their variances are 0.10^2 . In simulation, we use 50 particles.

The simulation data of the inputs, process outputs and nonlinear outputs are depicted in Fig. 8 (samples 1-200). The estimation errors τ versus t are depicted in Fig. 9, and the parameter estimates and their estimation errors are presented in Table III. The estimated process outputs, the true process outputs and their errors are shown in Fig. 10 (samples 800-900).

TABLE III
THE SG-PF ALGORITHM ESTIMATES AND ERRORS IN EXAMPLE 3

t	a_1	a_2	b_1	b_2	τ (%)
100	0.24267	0.20352	-0.83175	0.11461	29.76843
200	0.31686	0.22772	-0.80939	0.18038	20.42832
400	0.38488	0.18285	-0.82841	0.24033	10.91766
600	0.39947	0.17449	-0.80752	0.28280	7.59923
800	0.45080	0.18725	-0.82558	0.27935	3.69062
1000	0.46375	0.18250	-0.82056	0.29438	1.68361
True Values	0.47000	0.18000	-0.82000	0.31000	

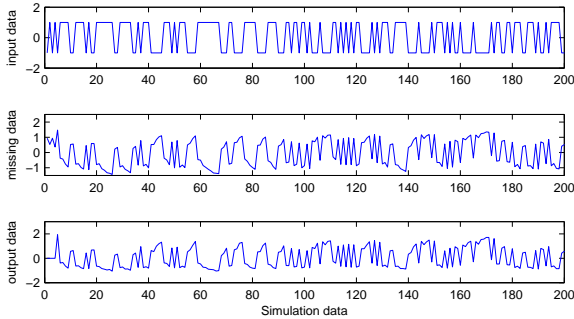


Fig. 8. The simulation data in Example 3

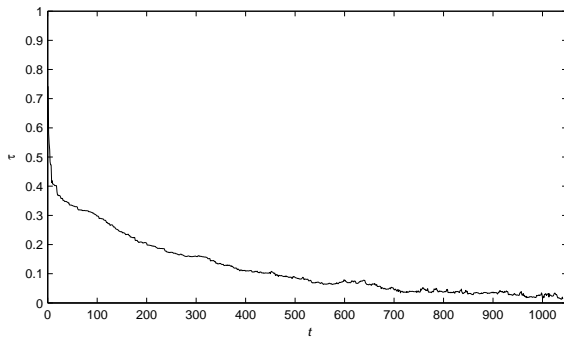


Fig. 9. The parameter estimation errors τ versus t in Example 3

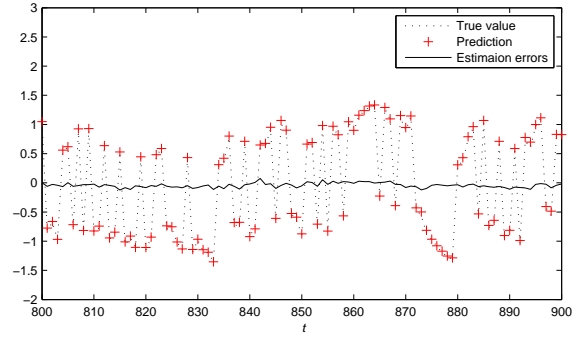


Fig. 10. Estimated model outputs, true outputs and estimation errors in Example 3

From Fig. 9 and Table III, we can conclude that as t increases, the parameter estimation errors approach zero. From Fig. 10, we find that the estimated process outputs can track the true unmeasurable outputs well.

V. CONCLUSIONS

An SG-PF algorithm is proposed for a non-standard ARX model in this paper. This model consists of two submodels, one is an ARX model and the other is a nonlinear communication output model. Since the outputs of the ARX model are unmeasurable, a particle filter is utilized to estimate these unknown outputs. Then based on the measurable input data and the estimated process output data, the unknown parameters of the ARX model can be estimated by the SG algorithm. Furthermore, two modified particle filters are developed for the non-standard ARX model with the assumptions that the measurement noise is a Gaussian noise with unknown variance and the measurement noise distribution is unknown.

The purpose of this paper is to develop an on-line identification method for systems with nonlinear communication output. There are still some interesting topics not discussed in this paper. For example, if the structure of the nonlinear function is unknown, how to adjust the weights of the particles? Another topic is how to prove the convergence properties of the SG-PF algorithm. These topics will remain as open issues in future.

APPENDIX A PROOF OF THEOREM 1

Define

$$\begin{aligned} |y(t) - f(\hat{x}_j(t))| &:= \gamma_j(t), \quad j = 1, 2, \dots, N, \\ \gamma(t) &:= \max\{\gamma_j(t), j = 1, \dots, N\} + 1. \end{aligned}$$

According to Equation (33), one can get

$$\begin{aligned} &\sum_{j=1}^N p(y(t)|\hat{x}_j(t))(y(t) - f(\hat{x}_j(t)))^2 - \sum_{j=1}^N \frac{1}{N} (y(t) - f(\hat{x}_j(t)))^2 \\ &= \sum_{j=1}^N \left[\frac{\gamma^2(t) - \gamma_j^2(t)}{N\gamma^2(t) - \sum_{k=1}^N \gamma_k^2(t)} - \frac{1}{N} \right] \gamma_j^2(t) \\ &= \sum_{j=1}^N \frac{-(N-1)\gamma_j^2(t) + \sum_{k=1, k \neq j}^N \gamma_k^2(t)}{N(N\gamma^2(t) - \sum_{k=1}^N \gamma_k^2(t))} \gamma_j^2(t). \end{aligned}$$

Since the denominator of the last term of the above equation is greater than zero, one only need to consider the following equation

$$\begin{aligned}
& \sum_{j=1}^N \left[-(N-1)\gamma_j^2(t) + \sum_{k=1, k \neq j}^N \gamma_k^2(t) \right] \gamma_j^2(t) \\
&= \left[(1-N)\gamma_1^4(t) + \gamma_1^2(t) \sum_{k=2}^N \gamma_k^2(t) \right] + \dots \\
& \left[(1-N)\gamma_j^4(t) + \gamma_j^2(t) \sum_{k=1, k \neq j}^N \gamma_k^2(t) \right] + \dots \\
& \left[(1-N)\gamma_N^4(t) + \gamma_N^2(t) \sum_{k=1}^{N-1} \gamma_k^2(t) \right] \\
&= -\sum_{k=2}^N (\gamma_1^2(t) - \gamma_k^2(t))^2 - \sum_{k=3}^N (\gamma_2^2(t) - \gamma_k^2(t))^2 - \dots \\
& - \sum_{k=j+1}^N (\gamma_j^2(t) - \gamma_k^2(t))^2 - \dots - (\gamma_{N-1}^2(t) - \gamma_N^2(t))^2 \leq 0.
\end{aligned}$$

Clearly, the following inequality holds

$$\sum_{j=1}^N p(y(t)|\hat{x}_j(t))(y(t) - f(\hat{x}_j(t)))^2 \leq \sum_{j=1}^N \frac{1}{N} (y(t) - f(\hat{x}_j(t)))^2.$$

APPENDIX B PROOF OF THEOREM 2

The weight which updated by the direct weight optimization method is expressed as

$$\lambda_j^d(t) = \frac{\gamma(t) - \gamma_j(t)}{N\gamma(t) - \sum_{j=1}^N \gamma_j(t)}, \quad (34)$$

while the weight updated by the Kernel density estimation method is written by

$$\lambda_j^k(t) = \frac{\gamma^2(t) - \gamma_j^2(t)}{N\gamma^2(t) - \sum_{j=1}^N \gamma_j^2(t)}. \quad (35)$$

Then one can get

$$\begin{aligned}
& \sum_{j=1}^N \lambda_j^d(t)(y(t) - f(\hat{x}_j(t)))^2 - \sum_{j=1}^N \lambda_j^k(t)(y(t) - f(\hat{x}_j(t)))^2 \\
&= \sum_{j=1}^N \lambda_j^d(t)\gamma_j^2(t) - \sum_{j=1}^N \lambda_j^k(t)\gamma_j^2(t) \\
&= \frac{\sum_{k=1}^N \gamma_k^2(t) \left[\sum_{j=1}^N \gamma_j^2(t)(\gamma_j(t) - \gamma(t)) \right]}{\left(N\gamma(t) - \sum_{j=1}^N \gamma_j(t) \right) \left(N\gamma^2(t) - \sum_{j=1}^N \gamma_j^2(t) \right)} + \\
& \frac{\sum_{k=1}^N \gamma_k(t) \left[\sum_{j=1}^N \gamma_j^2(t)(\gamma_j(t) + \gamma(t))(\gamma_j(t) - \gamma(t)) \right]}{\left(N\gamma(t) - \sum_{j=1}^N \gamma_j(t) \right) \left(N\gamma^2(t) - \sum_{j=1}^N \gamma_j^2(t) \right)} + \\
& \frac{N\gamma(t) \left[\sum_{j=1}^N \gamma_j^3(t)(\gamma_j(t) - \gamma(t)) \right]}{\left(N\gamma(t) - \sum_{j=1}^N \gamma_j(t) \right) \left(N\gamma^2(t) - \sum_{j=1}^N \gamma_j^2(t) \right)}. \quad (36)
\end{aligned}$$

Since the denominator of Equation (36) is greater than zero, one only need to consider the numerator of Equation (36),

$$\begin{aligned}
& \sum_{k=1}^N \gamma_k^2(t) \left[\sum_{j=1}^N \gamma_j^2(t)(\gamma_j(t) - \gamma(t)) \right] + \\
& \sum_{k=1}^N \gamma_k(t) \left[\sum_{j=1}^N \gamma_j^2(t)(\gamma_j(t) + \gamma(t))(\gamma_j(t) - \gamma(t)) \right] + \\
& N\gamma(t) \left[\sum_{j=1}^N \gamma_j^3(t)(\gamma_j(t) - \gamma(t)) \right] \\
&= -(\gamma_1(t) - \gamma(t))(\gamma_2(t) - \gamma(t)) \times \\
& \quad (\gamma_1(t) + \gamma_2(t))(\gamma_1(t) - \gamma_2(t))^2 - \dots \\
& -(\gamma_1(t) - \gamma(t))(\gamma_N(t) - \gamma(t)) \times \\
& \quad (\gamma_1(t) + \gamma_N(t))(\gamma_1(t) - \gamma_N(t))^2 - \dots \\
& -(\gamma_j(t) - \gamma(t))(\gamma_{j+1}(t) - \gamma(t)) \times \\
& \quad (\gamma_j(t) + \gamma_{j+1}(t))(\gamma_j(t) - \gamma_{j+1}(t))^2 - \dots \\
& -(\gamma_j(t) - \gamma(t))(\gamma_N(t) - \gamma(t)) \times \\
& \quad (\gamma_j(t) + \gamma_N(t))(\gamma_j(t) - \gamma_N(t))^2 - \dots \\
& -(\gamma_{N-1}(t) - \gamma(t))(\gamma_N(t) - \gamma(t)) \times \\
& \quad (\gamma_{N-1}(t) + \gamma_N(t))(\gamma_{N-1}(t) - \gamma_N(t))^2, \quad (37)
\end{aligned}$$

in which

$$\begin{aligned}
& \gamma_j(t) - \gamma(t) \leq 0, \quad \gamma_{j+1}(t) - \gamma(t) \leq 0, \quad \gamma_j(t) + \gamma_{j+1}(t) > 0, \\
& (\gamma_j(t) - \gamma_{j+1}(t))^2 \geq 0, \quad j = 1, \dots, N-1.
\end{aligned}$$

It follows that each term of Equation (37) is

$$-(\gamma_j(t) - \gamma(t))(\gamma_{j+1}(t) - \gamma(t))(\gamma_j(t) + \gamma_{j+1}(t))(\gamma_j(t) - \gamma_{j+1}(t))^2 \leq 0.$$

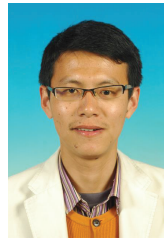
Thus we can get

$$\sum_{j=1}^N \lambda_j^d(t)(y(t) - f(\hat{x}_j(t)))^2 \leq \sum_{j=1}^N \lambda_j^k(t)(y(t) - f(\hat{x}_j(t)))^2.$$

REFERENCES

- [1] D. Yue, Q.L. Han, and C. Peng, "State feedback controller design of networked control systems," *IEEE Trans. Circuits Syst.(II)*, vol. 51, no. 11, pp. 640-644, Nov. 2004.
- [2] Z. Wang, B. Shen, and X. Liu, " H_∞ filtering with randomly occurring sensor saturations and missing measurements," *Automatica*, vol. 48, pp. 556-562, Mar. 2012.
- [3] J. Deng and B. Huang, "Identification of nonlinear parameter varying systems with missing output data," *AICHE J.*, vol. 58, pp. 3454-3467, Mar. 2012.
- [4] R.B. Gopaluni, "A particle filter approach to identification of nonlinear process under missing observations," *Can. J. Chem. Eng.*, vol. 86, pp. 1081-1092, Dec. 2008.
- [5] Y.J. Lu, B. Huang, and S. Khatibisepehr, "A variational Bayesian approach to robust identification of switched ARX models," *IEEE Trans. Cybern.*, vol. 46, no. 12, pp. 3195-3208, Dec. 2016.
- [6] M. Galrinho, N. Everitt, and H. Hjalmarsson, "ARX modeling of unstable linear systems," *Automatica*, vol. 75, pp. 167-171, Jan. 2017.
- [7] K. George and P. Mutalik, "A multiple model approach to time-series prediction using an online sequential learning algorithm," *IEEE Trans. Syst. Man, Cybern., Syst.*, vol. 99, 2017. DOI: 10.1109/TSMC.2017.2712184
- [8] L.Y. Wang, F. Ding, and X.P. Liu, "Convergence of HLS estimation algorithms for multivariable ARX-like systems," *Appl. Math. Comput.*, vol. 190, no. 2, pp. 1081-1093, Jul. 2007.
- [9] J.Y. Ha, D.H. Kang, and F.C. Park, "A stochastic global optimization algorithm for the two-frame sensor calibration problem," *IEEE Trans. Ind. Electron.*, vol. 63, no. 4, pp. 2434-2446, Apr. 2016.
- [10] W. Nam and S.H. Kong, "Least-squares-based iterative multipath super-resolution technique," *IEEE Trans. Signal Process.*, vol. 61, no. 3, pp. 519-529, Feb. 2013.

- [11] X.H. Wang, F. Ding, F.E. Alsaadi, and T. Hayat, "Convergence analysis of the hierarchical least squares algorithm for bilinear-in-parameter systems," *Circuits Syst. Signal Process.*, vol. 35, no. 12, pp. 4307-4330, Feb. 2016.
- [12] Q.B. Jin, Z. Wang, and X.P. Liu, "Auxiliary model-based interval-varying multi-innovation least squares identification for multivariable OE-like systems with scarce measurements," *J. Process Control*, vol. 35, no. 11, pp. 154-168, Sept. 2015.
- [13] Y.J. Wang and F. Ding, "Novel data filtering based parameter identification for multiple-input multiple-output systems using the auxiliary model," *Automatica*, vol. 71, pp. 308-313, Sept. 2016.
- [14] H. Li, Y. Gao, P. Shi, and H.K. Lam, "Observer-based fault detection for nonlinear systems with sensor fault and limited communication capacity," *IEEE Trans. Autom. Control*, vol. 61, no. 9, pp. 2745-2751, Sept. 2016.
- [15] H. Li and Y. Shi, "Distributed model predictive control of constrained nonlinear systems with communication delays," *Syst. Control Lett.*, vol. 62, no. 10, pp. 819-826, Oct. 2013.
- [16] X. Yin, L. Zhang, Y. Zhu, C.H. Wang, and Z.J. Li, "Robust control of networked systems with variable communication capabilities and application to a semi-active suspension system," *IEEE/ASME Trans. Mech.*, vol. 21, no. 4, pp. 2097-2107, Aug. 2016.
- [17] F. Guo, O.Y. Wu, H. Kodamana, Y.S. Ding, and B. Huang, "An augmented model approach for identification of nonlinear errors-in-variables systems using the EM algorithm," *IEEE Trans. Syst. Man, Cybern., Syst.*, vol. 99, 2017. DOI: 10.1109/TSMC.2017.2692273.
- [18] H. Li and Y. Shi, "Distributed receding horizon control of large-scale nonlinear systems: Handling communication delays and disturbances," *Automatica*, vol. 50, pp. 1264-1271, Apr. 2014.
- [19] X.Q. Yang, S. Yin, and O. Kaynak, "Robust identification of LPV time-delay system with randomly missing measurements," *IEEE Trans. Syst. Man, Cybern., Syst.*, vol. 99, 2017. DOI: 10.1109/TSMC.2017.2689920
- [20] J. Guo, Y.L. Zhao, C.Y. Sun, and Y. Yao, "Recursive identification of FIR systems with binary-valued outputs and communication channels," *Automatica*, vol. 60, pp. 165-172, Oct. 2015.
- [21] W.L. Xiong, X.Q. Yang, K. Liang, and B.G. Xu, "EM algorithm-based identification of a class of nonlinear Wiener systems with missing output data," *Nonlinear Dyn.*, vol. 80, no. 1, pp. 329-339, Apr. 2015.
- [22] L. Xie, H.Z. Yang, and B. Huang, "FIR model identification of multirate processes with random delays using EM algorithm," *AIChE J.* vol. 59, pp. 4124-4132, June. 2013.
- [23] D. Simon and Y.S. Shmaliy, "Unified forms for Kalman and finite impulse response filtering and smoothing," *Automatica*, vol. 49, no. 6, pp. 1892-1899, Jun. 2013.
- [24] S.Y. Zhao, Y.S. Shmaliy, and F. Liu, "Fast Kalman-like optimal unbiased FIR filtering with applications," *IEEE Trans. Signal Process.*, vol. 64, no. 9, pp. 2284-2296, May 2016.
- [25] S.Y. Zhao, B. Huang, and F. Liu, "Linear optimal unbiased filter for time-variant systems without apriori information on initial conditions," *IEEE Trans. Autom. Control*, vol. 62, no. 2, pp. 882-887, Feb. 2017.
- [26] F. Ding, X.M. Liu, and X.Y. Ma, "Kalman state filtering based least squares iterative parameter estimation for observer canonical state space systems using decomposition," *J. Comput. Appl. Math.*, vol. 301, no. 135-143, Aug. 2016.
- [27] F. Ding and Y. Gu, "Performance analysis of the auxiliary model-based stochastic gradient parameter estimation algorithm for state space systems with one-step state delay," *Circuits Syst. Signal Process.*, vol. 32, no. 2, pp. 585-599, Apr. 2013.
- [28] F. Ding, X.M. Liu, and Y. Gu, "An auxiliary model based least squares algorithm for a dual-rate state space system with time-delay using the data filtering," *J. Frankl. Inst.*, vol. 353, no. 2, pp. 398-408, Jan. 2016.
- [29] D. Simon, *Optimal State Estimation: Kalman, H_∞ , and Nonlinear Approaches*, John Wiley & Sons, Inc., New Jersey, 2006.
- [30] M.S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, "Tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking," *IEEE Trans. Signal Process.*, vol. 50, no. 2, pp. 174-188, Feb. 2002.
- [31] E.W. Bai and Y. Liu, "A minimal probability approach in nonparametric nonlinear system identification," *IEEE Trans. Autom. Control*, vol. 52, no. 7, pp. 1218-1231, Jul. 2007.
- [32] J. Chen, J. Li, and Y.J. Liu, "Gradient iterative algorithm for dual-rate nonlinear systems based on a novel particle filter," *J. Frankl. Inst.*, vol. 354, no. 11, pp. 4425-4437, Jul. 2017.
- [33] J. Chen, Y. Zhang, and R. Ding, "Auxiliary model based multi-innovation algorithms for multivariable nonlinear systems," *Math. Comput. Model.*, vol. 52, no. 9-10, pp. 1428-1434, Nov. 2010.



Jing Chen received his B.Sc. degree in School of Mathematical Science and M.Sc. degree in School of Information Engineering from Yangzhou University (Yangzhou, China) in 2003 and 2006, respectively, and received his Ph.D. degree in the School of Internet of Things Engineering, Jiangnan University (Wuxi, China) in 2013. He is currently an associate professor in College of Science, Jiangnan University (Wuxi, China). His research interests include Processing Control and system identification.

PLACE
PHOTO
HERE

Yanjun Liu received his B.Sc. degree in School of Mathematical Science and M.Sc. degree in School of Information Engineering from Yangzhou University (Yangzhou, China) in 2003 and 2006, respectively, and received his Ph.D. degree in the School of Internet of Things Engineering, Jiangnan University (Wuxi, China) in 2013. He is currently an associate professor in College of Science, Jiangnan University (Wuxi, China). His research interests include Processing Control and system identification.

PLACE
PHOTO
HERE

Feng Ding received his B.Sc. degree in School of Mathematical Science and M.Sc. degree in School of Information Engineering from Yangzhou University (Yangzhou, China) in 2003 and 2006, respectively, and received his Ph.D. degree in the School of Internet of Things Engineering, Jiangnan University (Wuxi, China) in 2013. He is currently an associate professor in College of Science, Jiangnan University (Wuxi, China). His research interests include Processing Control and system identification.

PLACE
PHOTO
HERE

Quanmin Zhu is Professor in control systems at the Department of Engineering Design and Mathematics, University of the West of England, Bristol, UK. He obtained his M.Sc. in Harbin Institute of Technology, China in 1983 and Ph.D. in Faculty of Engineering, University of Warwick, UK in 1989. His main research interest is in the area of nonlinear system modelling, identification, and control. Currently Professor Zhu is acting as Editor (and Founder) of International Journal of Modelling, Identification and Control (<http://www.inderscience.com/ijmic>), Editor of International Journal of Computer Applications in Technology (<http://www.inderscience.com/ijcat>), and president (and founder) of series annual International Conference on Modelling, Identification and Control (<http://icmic.org.uk/>).