Gradient-Based Particle Filter Algorithm for an ARX Model With Nonlinear Communication Output

Jing Chen, Yanjun Liu, Feng Ding, and Quanmin Zhu

Abstract—A stochastic gradient (SG)-based particle filter (SG-PF) algorithm is developed for an ARX model with nonlinear communication output in this paper. This 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 SG-PF 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 SG-PF algorithm is effective.

Index Terms—ARX model, auxiliary model, parameter estimation, particle filter, stochastic gradient (SG).

I. INTRODUCTION

This paper considers an ARX model mixed with a nonlinear communication output model, which 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

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

Fig. 1. ARX model with nonlinear communication output.

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 $\varsigma^2$ and $\sigma^2$, respectively, and $A(d)$ and $B(d)$ are polynomials which can be expressed as

$$A(d) := 1 + a_1 d^{-1} + \cdots + a_n d^{-n}$$
$$B(d) := b_1 d^{-1} + b_2 d^{-2} + \cdots + b_m d^{-m}$$

Manuscript received December 5, 2017; accepted February 14, 2018. Date of publication March 7, 2018; date of current version May 15, 2020. This work was supported in part by the National Natural Science Foundation of China under Grant 61403165, and in part by the Natural Science Foundation for Colleges and Universities in Jiangsu Province under Grant 16KJB120006. This paper was recommended by Associate Editor J.-H. Chou. (Corresponding author: Jing Chen.)

J. Chen is with the School of Science, Jiangnan University, Wuxi 214122, China (e-mail: chenjing1981929@126.com).

Y. Liu and F. Ding are with the School of Internet of Things Engineering, Jiangnan University, Wuxi 214122, China (e-mail: yanjunliu_1983@126.com; fding@jiangnan.edu.cn).

Q. Zhu is with the Department of Engineering Design and Mathematics, University of the West of England, Bristol BS16 1QY, U.K. (e-mail: quan.zhu@uwe.ac.uk).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TSMC.2018.2810277
computational efforts and cannot be used to estimate the parameters recursively by new data, while the RLS and SG algorithms can be used as online algorithms and can update the parameters based on new data [12]. Furthermore, these online algorithms have less computational efforts, thus widely used in engineering application.

The online 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 [13]. Its basic idea is to replace the missing data with the outputs of an auxiliary model. For example, Wang and Ding [14] studied an auxiliary model-based RLS algorithm for multivariable systems. Jin et al. [13] investigated an auxiliary model-based identification algorithm for multivariable OE-like systems with missing outputs. 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 [15]–[17]. However, the complex network and laboratory analysis process also bring some identification constraints, such as the packet dropouts, the communication time-delays, the nonlinear signals and the unmeasurable sampled outputs [18]–[20]. There exist many identification methods for systems with such constraints. For example, Guo et al. [21] proposed an identification algorithm for FIR systems, in which the outputs are constrained by both binary-valued quantization and communication unreliability. Xiong et al. [22] investigated an EM algorithm for nonlinear systems with unmeasurable outputs, in which those unmeasurable outputs were estimated by an auxiliary model. Xie et al. [23] developed an EM algorithm for multivariate 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 offline algorithm, the missing outputs and the parameters were estimated simultaneously.

The Kalman filter and the particle filter are two effective tools which are often applied for state-space systems [24]–[26]. In [27], a multi-innovation gradient algorithm is developed for a linear state-space system with time-delay, in order to improve the convergence rate, a Kalman filtering-based multi-innovation gradient algorithm is proposed. In [3], an EM-based particle filter algorithm is proposed for nonlinear parameter varying state-space systems, where the proposed EM algorithm is an offline 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 (SG-PF) algorithm for an ARX model with nonlinear communication output. The considered system in this paper is a nonstate-space system with different kinds of measurement noises, and the proposed algorithm is an online 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], and [23] and the main contributions are listed below.

1) Study an online identification method for an ARX model which consists of two submodels, one is an equation-error 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 nonstate-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-PF 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. SG-BASED PARTICLE FILTER ALGORITHM

For the model depicted in Fig. 1, define the information vector $\vect{\phi}(t)$ and the parameter vector $\vect{\theta}$ as

$$\begin{align*}
\vect{\phi}(t) &:= \left[ -x(t-1), \ldots, -x(t-n), u(t-1), \ldots, u(t-m) \right]^T \in \mathbb{R}^{n+m} \\
\vect{\theta} &:= [a_1, \ldots, a_n, b_1, \ldots, b_m]^T \in \mathbb{R}^{n+m}
\end{align*}$$

then the ARX model can be rewritten as follows:

$$\begin{align*}
x(t) &= \vect{\phi}^T(t)\vect{\theta} + \nu(t) \\
y(t) &= f(x(t)) + e(t)
\end{align*}$$

(1) (2)

In (1), the information vector contains unknown process outputs $x(i), i = 1, 2, \ldots, \text{thus the classical SG algorithm cannot be applied to (1) directly. The auxiliary model-based algorithms are often used to identify systems with unmeasurable variables [13], [14], [28].}$ For example, the corresponding auxiliary model-based SG algorithm for the system in (1) and (2) is as follows:

$$\hat{\vect{\theta}}(t) = \vect{\hat{\theta}}(t-1) + \frac{\hat{\vect{\phi}}(t)}{v(t)} e(t)$$

(3)

$$\hat{\vect{\phi}}(t-i) = \left[ -\hat{x}(t-i-1), \ldots, -\hat{x}(t-i-n), u(t-i-1), \ldots, u(t-i-m) \right]^T$$

(4)

$$\hat{x}(t-i) = \hat{\vect{\phi}}^T(t-i) \hat{\vect{\theta}}(t-i-1)$$

(5)

$$e(t) = x(t) - \hat{\vect{\phi}}^T(t) \hat{\vect{\theta}}(t-1)$$

(6)

$$v(t) = v(t-1) + \| \hat{\vect{\phi}}(t) \|^2, \quad v(0) = 1$$

(7)

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

Remark 1: Unlike the models in [13], [14], and [28], all the process outputs in this ARX model are unknown, and if
we use $\hat{x}(t) = \hat{\phi}^T(t)\hat{\theta}(t - 1)$ to replace $x(t)$, then $\epsilon(t)$ in (6) becomes zero, thus the SG-AM algorithm proposed in (3)–(7) is invalid for this ARX model.

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

### A. 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 nonstate-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] and [29], the posterior density function $p(x(t - i)|y(t - i), \ldots, y(1), x(t - i - 1), \ldots, x(1), u(t - i - 1), \ldots, u(1))$ can be approximately by

$$p\left(x(t - i)|y(t - i), \ldots, y(1), x(t - i - 1), \ldots, x(1)\right) \times u(t - i - 1), \ldots, u(1), \hat{\theta}(t - 1)) \approx \sum_{j=1}^{N} \omega_j \delta(x(t - i) - \hat{x}_j(t - i))$$

(8)

where $\omega_j$ 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

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

(9)

in which $\bar{\omega}_j = [(\omega_j)/\sum_{j=1}^{N} \omega_j]$ 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 $p(x(t - i)|y(t - i), \ldots, y(1), x(t - i - 1), \ldots, x(1), u(t - i - 1), \ldots, u(1), \theta)$. 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

$$q\left(x(t - i)|y(t - i), \ldots, y(1), x(t - i - 1), \ldots, x(1)\right)$$

$$u(t - i - 1), \ldots, u(1), \hat{\theta}(t - 1)) = p\left(x(t - i)|x(t - i - 1), \ldots, x(1)\right)$$

$$u(t - i - 1), \ldots, u(1), \hat{\theta}(t - 1))$$

(10)

From (1) and (10), one gets

$$q\left(x(t - i)|y(t - i), \ldots, y(1), x(t - i - 1), \ldots, x(1)\right)$$

$$u(t - i - 1), \ldots, u(1), \hat{\theta}(t - 1)) = \frac{1}{\sqrt{2\pi\varsigma}} \exp\left[-\frac{(x(t - i) - \phi^T(t - i)\hat{\theta}(t - i - 1))^2}{2\varsigma^2}\right]$$

(11)

Remark 2: If the variance $\varsigma$ is unknown, then the density function in (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.,

$$\hat{x}_j(t - i) = \hat{\phi}^T(t - i)\hat{\theta}(t - i - 1)$$

$$\phi_j(t - i) = [-\hat{x}_j(t - i - 1), \ldots, -\hat{x}_j(t - i - n)]^T$$

$$u(t - i - 1), \ldots, u(t - i - m)$$

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

From (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 follows:

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

(12)

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

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

(13)

Substituting (13) into (12) gets

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

(14)

Normalizing $\omega_j(t)$ yields

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

(15)

Then we have

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

(16)

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 $N_{\text{eff}}$ is defined as

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

(17)

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

The particle filter proceeds by performing the following steps.

1) **Initialization**: At time $t$, collect the input–output data $\{u(t), y(t), \ldots, u(1), y(1)\}$. Draw $N$ initial particles $\left[\tilde{x}_j(0)\right]_{j=1}^{N}$ from $p(x(0)|\hat{\theta}(t - 1))$ and set $\{\omega_j(0)\}_{j=1}^{N} = (1/N)$, $\hat{\theta}(t - 1)$ is the estimate of $\theta$ at time $t - 1$.

2) Let the process noise variance be $\varsigma^2$ and the measurement noise variance be $\sigma^2$, and assign a positive number $N_{\text{hold}}$.
3) Let \(i = t - 1\).
4) Compute \(\{\hat{x}(t-\bar{j})\}_{\bar{j}=1}^{N}\) from (11).
5) Compute \(\hat{\omega}_j(t-\bar{i})\) by (14).
6) Normalize \(\hat{\omega}_j(t-\bar{i})\) from (15).
7) Compute the missing output \(\hat{x}(t-\bar{i})\) by (16).
8) Compute \(N_{\text{eff}}\) by (17) and compare \(N_{\text{eff}}\) with \(N_{\text{hold}},\) if \(N_{\text{eff}} < N_{\text{hold}},\) reset the weight of each particle with \(\hat{\omega}_j(t-\bar{i}) = (1/N)\) 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. 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 \(\hat{\theta}(t-i-1)\) to estimate \(x(t)\).

Draw \(N\) particles from the following importance density function:

\[
q\left(x(t-i)|y(t-i), \ldots, y(1), \hat{x}(t-i-1), \ldots, \hat{x}(1)\right)
\]

\[
u(t-i-1), \ldots, u(1), \hat{\theta}(t-i-1)\)

in which

\[
\hat{x}(t-i-1) = [x(t-i-1)]|\hat{\theta}(t-i-2)
\]

\[
\vdots
\]

\[
\hat{x}(1) = [x(1)]|\hat{\theta}(0)
\]

where \([x(t-i-j)]|\hat{\theta}(t-i-j-1)\), \(j = 1, \ldots, i-1\) means the estimate of \(x(t-i-j)\) by using \(\hat{\theta}(t-i-j-1)\). Based on (16), we can get the estimate \(\hat{x}(t-i)\). Then with the estimates \(\hat{x}(t-i), \hat{x}(t-i-1), \ldots, \hat{x}(1)\) and \(\hat{\theta}(t-i-1)\), one can get the parameter estimate \(\hat{\theta}(t-i)\).

From (18), 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 \(\hat{\theta}(t-i-1)\).

Draw \(N\) particles from

\[
q\left(x(t-i)|y(t-i), \ldots, y(1), \hat{x}(t-i-1), \ldots, \hat{x}(1)\right)
\]

\[
u(t-i-1), \ldots, u(1), \hat{\theta}(t-i-1)\)

in which

\[
\hat{x}(t-i-1) = [x(t-i-1)]|\hat{\theta}(t-i-1)
\]

\[
\vdots
\]

\[
\hat{x}(1) = [x(1)]|\hat{\theta}(t-i-1)
\]

From (16), we can obtain the estimates \(\hat{x}(t-i)\). Then according to the estimates \(\hat{x}(t-i), \hat{x}(t-i-1), \ldots, \hat{x}(1)\) and \(\hat{\theta}(t-i-1)\), we can get the updated parameter vector \(\hat{\theta}(t-i)\).

The density function in (19) 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-PF algorithm is given by

\[
\hat{\theta}(t) = \hat{\theta}(t-1) + \frac{\hat{\phi}(t)}{r(t)} \left[\hat{x}(t) - \phi^T(t)\hat{\theta}(t-1)\right]
\]

\[
\phi(t) = [\hat{x}(t-1), \ldots, \hat{x}(t-n), u(t-1), \ldots, u(t-m)]^T
\]

\[
\hat{x}(t-i) = \sum_{j=1}^{N} \hat{\omega}_j(t-i) \hat{x}_j(t-i)
\]

\[
r(t) = r(t-1) + \|\hat{\phi}(t)\|^2, \quad r(0) = 1.
\]

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 \(\varepsilon\).
2) Let \(t = 1, \hat{\theta}(0) = [1/p_0, \ldots, 1/p_0]^T \in \mathbb{R}^{n+m}\), where \(p_0 = 10^0\).
3) Collect the input–output data \(\{u(t), y(t)\}\).
4) Update \(\hat{\omega}_j(t-i)\) and \(\hat{x}_j(t-i), j = 1, \ldots, N\) by using the particle filter.
5) Compute \(\hat{x}(t-i), i = t - 1, t - 2, \ldots, 0\) by (22).
6) Form \(\phi(t)\) by (21).
7) Compute \(r(t)\) by (23).
8) Estimate \(\hat{\theta}(t)\) by (20).
9) Compare \(\hat{\theta}(t)\) and \(\hat{\theta}(t-1)\), if \(\|\hat{\theta}(t) - \hat{\theta}(t-1)\| / \|\hat{\theta}(t)\| \leq \varepsilon\), 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 (20)–(23), we can see that the unknown process output is estimated by (22), obviously, \(\hat{x}(t) - \phi^T(t)\hat{\theta}(t-1) \neq 0\). 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. Gaussian Measurement Noise With Unknown Variance**

Since the variance of the measurement noise is unknown, the weight \(w_j(t)\) in (14) 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 $\hat{x}_j(t), j = 1, \ldots, N$ from (19) and define

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

The next is to find the weights $\lambda_j(t), j = 1, \ldots, N$ 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 $\lambda_j(t), j = 1, \ldots, N$ can be obtained by minimizing the following function:

$$\left[ \sum_{j=1}^{N} \lambda_j(t)(f(\hat{x}_j(t)) + e(t)) - y(t) \right]^2 = \left( \sum_{j=1}^{N} \lambda_j(t)f(\hat{x}_j(t)) + \sum_{j=1}^{N} \lambda_j(t)e(t) - \sum_{j=1}^{N} \lambda_j(t)y(t) \right)^2.$$  \hspace{1cm} (24)

Define

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

Taking the conditional expectation on (24) yields

$$E \left[ \sum_{j=1}^{N} \lambda_j(t)(f(\hat{x}_j(t)) + e(t)) - y(t) \right]^2 = \left( \sum_{j=1}^{N} \lambda_j(t)\gamma_j(t) \right)^2 + \sigma^2 \sum_{j=1}^{N} \lambda_j^2(t).$$ \hspace{1cm} (25)

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 $\text{Prob}(z \geq \gamma(t))$ is minimized. In this paper, we choose $\gamma(t)$ as

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

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

$$p(z) = \left\{ \begin{array}{ll} \frac{2}{\sqrt{2\pi}\zeta^2} \exp \left(-\frac{(z-z_0)^2}{2\zeta^2}\right), & z \geq z_0 \\ 0 & \end{array} \right.$$ \hspace{1cm} (26)

where

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

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

$$\text{Max}_{\sum_{j=1}^{N} \lambda_j(t) = 1} \sum_{j=1}^{N} \lambda_j(t)\gamma_j(t).$$

Then the weight can be computed by

$$\lambda_j(t) = \frac{\gamma(t) - \gamma_j(t)}{N\gamma(t) - \sum_{j=1}^{N} \gamma_j(t)}.$$ \hspace{1cm} (27)

Substituting (27) into (12) gets

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

and the normalized weight can be expressed as

$$\tilde{\omega}_j(t) = \frac{\omega_j(t)}{\sum_{k=1}^{N} \omega_k(t)}. \hspace{1cm} (28)$$

Remark 5: Equations (27) and (28) imply that the weight of each particle can be obtained without the knowledge of the variance of the measurement noise.

B. Measurement Noise With Unknown Distribution

When the distribution of the measurement noise is unknown, both density function in (13) and (26) cannot be obtained. In order to solve this problem, we use the Kernel density estimation method to update the weight of each particle.

Rewrite

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

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

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

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

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

$$p(y(t)|\hat{x}_j(t)) = \frac{y^2(t) - \gamma_j^2(t)}{N\gamma^2(t) - \sum_{j=1}^{N} \gamma_j^2(t)}.$$ \hspace{1cm} (29)

Substituting (29) into (12) gets

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

Normalizing \( \omega_j(t) \) yields
\[
\tilde{\omega}_j(t) = \frac{\omega_j(t)}{\sum_{j=1}^{N} \omega_j(t)}.
\]

**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:
\[
\frac{\gamma^2(t) - \gamma^2_l(t)}{N\gamma^2(t) - \sum_{j=1}^{N} \gamma^2_j(t)} > \frac{\gamma^2(t) - \gamma^2_l(t)}{N\gamma^2(t) - \sum_{j=1}^{N} \gamma^2_j(t)}.
\]

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 (29) 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 can also 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 \( \lambda^d_j(t) \) be the weight updated by (27), \( \lambda^k_j(t) \) be the weight updated by (29). Assume that the measurement noise is a Gaussian noise with unknown variance, then
\[
\sum_{j=1}^{N} \lambda^d_j(t)(y(t) - f(\hat{x}_j(t)))^2 \leq \sum_{j=1}^{N} \lambda^k_j(t)(y(t) - f(\hat{x}_j(t)))^2.
\]

The proof is given in Appendix B.

Theorem 2 means that the direct weight optimization method is better than the Kernel density estimation method when the measurement noise is a Gaussian noise with unknown variance.

**IV. EXAMPLES**

**Example 1:** Consider the following ARX model:
\[
A(d)x(t) = B(d)u(t) + v(t)
\]
\[
A(d) = 1 + a_1 d^{-1} = 1 + 0.49 d^{-1}
\]
\[
B(d) = b_1 d^{-1} + b_2 d^{-2} = -0.83 d^{-1} + 0.17 d^{-2}
\]
\[
y(t) = \exp(x(t)) + e(t)
\]
\[
u(t) \sim N(0, \tau^2), \quad v \sim N(0, \varsigma^2)
\]
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).

**Example 2:** Consider the ARX model in Example 1
\[
A(d)x(t) = B(d)u(t) + v(t)
\]
\[
y(t) = \exp(x(t)) + e(t)
\]
\[
u(t) \sim N(0, \tau^2), \quad v \sim N(0, \varsigma^2)
\]

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. The estimation errors \( \tau \) 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).

| \( \tau \) | \( \alpha_1 \) | \( b_1 \) | \( b_2 \) | \( \varsigma \) |
|---|---|---|---|---|
| 100 | 0.30121 | -0.81258 | 0.03847 | 23.52512 |
| 200 | 0.37672 | -0.82428 | 0.09128 | 14.10643 |
| 400 | 0.46706 | -0.82416 | 0.12997 | 75.154 |
| 600 | 0.48365 | -0.82306 | 0.15922 | 4.16911 |
| 800 | 0.48532 | -0.84526 | 0.17290 | 1.65736 |
| 1000 | 0.48776 | -0.82236 | 0.16352 | 1.04817 |

**TABLE I**

![Fig. 2. Simulation data in Example 1.](image1)

![Fig. 3. Parameter estimation errors \( \tau \) versus \( t \) in Example 1.](image2)

![Fig. 4. Parameter estimation errors \( \tau \) versus \( t \) in Example 2.](image3)

![Fig. 5. Parameter estimation errors \( \tau \) versus \( t \) in Example 2.](image4)

![Fig. 6. Parameter estimation errors \( \tau \) versus \( t \) in Example 2.](image5)
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. The measurable output is $y(t)$.

This kind of nonlinear function $f(x(t)) = x(t) + 0.2x^2(t) + e(t)$.

and we manually amplify the output as

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

and the simulation data of the inputs, process outputs and nonlinear outputs are depicted in Fig. 8 (samples 1–200). The estimation errors $\tau$ 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).
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. CONCLUSION

An SG-PF algorithm is proposed for an ARX model mixed with a nonlinear communication output 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 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 online 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

\[
|y(t) - f(\hat{x}_j(t))| := \gamma_j(t), \quad j = 1, 2, \ldots, N
\]

\[
\gamma(t) := \max\{\gamma_j(t), j = 1, \ldots, N\} + 1.
\]

According to (29), one can get

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

\[
= \sum_{j=1}^{N} \left[ \frac{\gamma^2(t) - \gamma_j^2(t)}{N\gamma^2(t) - \sum_{k=1}^{N} \gamma_k^2(t)} - \frac{1}{N} \right] \gamma_j^2(t)
\]

\[
= \sum_{j=1}^{N} \frac{-(N-1)\gamma_j^2(t) + \sum_{k=1, k \neq j}^{N} \gamma_k^2(t)}{N(N\gamma^2(t) - \sum_{k=1}^{N} \gamma_k^2(t))} \gamma_j^2(t).
\]

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

\[
\sum_{j=1}^{N} \left[ -(N-1)\gamma_j^2(t) + \sum_{k=1, k \neq j}^{N} \gamma_k^2(t) \right] \gamma_j^2(t)
\]

\[
= \left[ (1-N)\gamma_1^2(t) + \gamma_2^2(t) \sum_{k=2}^{N} \gamma_k^2(t) \right] + \ldots + \left[ (1-N)\gamma_N^2(t) + \gamma_1^2(t) \sum_{k=1, k \neq j}^{N} \gamma_k^2(t) \right]
\]

\[
= \ldots + \left[ (1-N)\gamma_N^2(t) + \gamma_1^2(t) \sum_{k=1}^{N-1} \gamma_k^2(t) \right]
\]

\[
= -\sum_{k=2}^{N} \left( \gamma_1^2(t) - \gamma_k^2(t) \right)^2 - \sum_{k=3}^{N} \left( \gamma_2^2(t) - \gamma_k^2(t) \right)^2 - \ldots
\]

\[
-\sum_{k=j+1}^{N} \left( \gamma_j^2(t) - \gamma_k^2(t) \right)^2 - \ldots - \left( \gamma_N^2(t) - \gamma_N^2(t) \right)^2 \leq 0.
\]

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

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

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

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

Then one can get

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

Thus we can get

$$\sum_{j=1}^{N} \lambda_j^d(t)(\gamma(t) - f(\hat{x}_j(t)))^2 \leq \sum_{j=1}^{N} \lambda_j^k(t)(\gamma(t) - f(\hat{x}_j(t)))^2.$$
[15] 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. Auton. Control, vol. 61, no. 9, pp. 2745–2751, Sep. 2016.

[16] 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.

[17] 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. Mechatronics, vol. 21, no. 4, pp. 2097–2107, Aug. 2016.

[18] 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., to be published, doi: 10.1109/TSMC.2017.2692273.

[19] 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.

[20] 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., to be published, doi: 10.1109/TSMC.2017.2689920.

[21] 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.

[22] 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,” Nonlin. Dyn., vol. 80, nos. 1–2, pp. 329–339, Apr. 2015.

[23] L. Xie, H. Z. Yang, and B. Huang, “FIR model identification of multirate processes with random delays using EM algorithm,” AICHE J. vol. 59, no. 11, pp. 4124–4132, Jun. 2013.

[24] 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.

[25] 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.

[26] S. Y. Zhao, B. Huang, and F. Liu, “Linear optimal unbiased filter for time-variant systems without apriori information on initial conditions,” IEEE Trans. Auton. Control, vol. 62, no. 2, pp. 882–887, Feb. 2017.

[27] F. Ding, X. H. Wang, L. Mao, and L. Xu, “Joint state and multi-innovation parameter estimation for time-delay linear systems and its convergence based on the Kalman filtering,” Digit. Signal Process., vol. 62, pp. 211–223, Mar. 2017.

[28] F. Ding, F. F. Wang, L. Xu, T. Hayat, and A. Alsaedi, “Parameter estimation for pseudo-linear systems using the auxiliary model and the decomposition technique,” IET Control Theory Appl., vol. 11, no. 3, pp. 390–400, Jan. 2017.

[29] D. Simon, Optimal State Estimation: Kalman, H∞, and Nonlinear Approaches. Hoboken, NJ, USA: Wiley, 2006.

[30] M. S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, “A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking,” IEEE Trans. Signal Process., vol. 50, no. 2, pp. 174–187, Feb. 2002.

[31] F. Ding, X. H. Wang, L. Mao, and L. Xu, “Joint state and multi-innovation parameter estimation for time-delay linear systems and its convergence based on the Kalman filtering,” Digit. Signal Process., vol. 62, pp. 211–223, Mar. 2017.

[32] F. Ding, F. F. Wang, L. Xu, T. Hayat, and A. Alsaedi, “Parameter estimation for pseudo-linear systems using the auxiliary model and the decomposition technique,” IET Control Theory Appl., vol. 11, no. 3, pp. 390–400, Jan. 2017.

[33] D. Simon, Optimal State Estimation: Kalman, H∞, and Nonlinear Approaches. Hoboken, NJ, USA: Wiley, 2006.

[34] F. Ding, X. H. Wang, L. Mao, and L. Xu, “Joint state and multi-innovation parameter estimation for time-delay linear systems and its convergence based on the Kalman filtering,” Digit. Signal Process., vol. 62, pp. 211–223, Mar. 2017.

[35] F. Ding, F. F. Wang, L. Xu, T. Hayat, and A. Alsaedi, “Parameter estimation for pseudo-linear systems using the auxiliary model and the decomposition technique,” IET Control Theory Appl., vol. 11, no. 3, pp. 390–400, Jan. 2017.

[36] D. Simon, Optimal State Estimation: Kalman, H∞, and Nonlinear Approaches. Hoboken, NJ, USA: Wiley, 2006.

[37] F. Ding, X. H. Wang, L. Mao, and L. Xu, “Joint state and multi-innovation parameter estimation for time-delay linear systems and its convergence based on the Kalman filtering,” Digit. Signal Process., vol. 62, pp. 211–223, Mar. 2017.

[38] F. Ding, F. F. Wang, L. Xu, T. Hayat, and A. Alsaedi, “Parameter estimation for pseudo-linear systems using the auxiliary model and the decomposition technique,” IET Control Theory Appl., vol. 11, no. 3, pp. 390–400, Jan. 2017.

[39] D. Simon, Optimal State Estimation: Kalman, H∞, and Nonlinear Approaches. Hoboken, NJ, USA: Wiley, 2006.

[40] F. Ding, X. H. Wang, L. Mao, and L. Xu, “Joint state and multi-innovation parameter estimation for time-delay linear systems and its convergence based on the Kalman filtering,” Digit. Signal Process., vol. 62, pp. 211–223, Mar. 2017.

[41] F. Ding, F. F. Wang, L. Xu, T. Hayat, and A. Alsaedi, “Parameter estimation for pseudo-linear systems using the auxiliary model and the decomposition technique,” IET Control Theory Appl., vol. 11, no. 3, pp. 390–400, Jan. 2017.

[42] D. Simon, Optimal State Estimation: Kalman, H∞, and Nonlinear Approaches. Hoboken, NJ, USA: Wiley, 2006.

[43] F. Ding, X. H. Wang, L. Mao, and L. Xu, “Joint state and multi-innovation parameter estimation for time-delay linear systems and its convergence based on the Kalman filtering,” Digit. Signal Process., vol. 62, pp. 211–223, Mar. 2017.

[44] F. Ding, F. F. Wang, L. Xu, T. Hayat, and A. Alsaedi, “Parameter estimation for pseudo-linear systems using the auxiliary model and the decomposition technique,” IET Control Theory Appl., vol. 11, no. 3, pp. 390–400, Jan. 2017.

[45] D. Simon, Optimal State Estimation: Kalman, H∞, and Nonlinear Approaches. Hoboken, NJ, USA: Wiley, 2006.