Neural Networks 67 (2015) 131–139

Contents lists available at ScienceDirect

Neural Networks

journal homepage: www.elsevier.com/locate/neunet

Low-dimensional recurrent neural network-based Kalman filter for speech enhancement☆

Youshen Xia [a,*], Jun Wang [b]

[a] College of Mathematics and Computer Science, Fuzhou University, China

[b] Department of Mechanical and Automation Engineering, The Chinese University of Hong Kong, Hong Kong

ARTICLE INFO

Article history:
Received 25 May 2014
Received in revised form 1 March 2015
Accepted 19 March 2015
Available online 7 April 2015

Keywords:
Recurrent neural network
Speech enhancement
Non-Gaussian noise
Noise-constrained estimation

ABSTRACT

This paper proposes a new recurrent neural network-based Kalman filter for speech enhancement, based on a noise-constrained least squares estimate. The parameters of speech signal modeled as autoregressive process are first estimated by using the proposed recurrent neural network and the speech signal is then recovered from Kalman filtering. The proposed recurrent neural network is globally asymptomatically stable to the noise-constrained estimate. Because the noise-constrained estimate has a robust performance against non-Gaussian noise, the proposed recurrent neural network-based speech enhancement algorithm can minimize the estimation error of Kalman filter parameters in non-Gaussian noise. Furthermore, having a low-dimensional model feature, the proposed neural network-based speech enhancement algorithm has a much faster speed than two existing recurrent neural networks-based speech enhancement algorithms. Simulation results show that the proposed recurrent neural network-based speech enhancement algorithm can produce a good performance with fast computation and noise reduction.

© 2015 Elsevier Ltd. All rights reserved.

1. Introduction

Speech enhancement techniques have been successfully used in many areas such as mobile communication systems, speech recognition systems, and hearing aid devices, where received speech signals are corrupted by white or colored noise (Kay, 1993; Loizou, 2007). The main objective of speech enhancement is to improve the performance of speech communication in noise environments. Over the past decades, much research has focused on this area. Speech enhancement techniques may be divided into single-channel speech enhancement and multi-channel speech enhancement (Bobillet et al., 2007; Boll, 1979; Doclo & Moonen, 2002, 2005; Ephraim & Malah, 1984; Epharim & Van Trees, 1995a; Ephraim & Van Trees, 1995b; Gabrea, 2005; Gabrea, Grivel, & Najim, 1999; Gannot, Burshtein, & Weinstein, 1998; Gerkmann & Hendriks, 2012; Gibson, Koo, & Gray, 1991; Kay, 1993; Labarre, Grivel, Najim, & Todini, 2004; Lee & Jung, 2000; Ning, Bouchard, & Goubran, 2006; Roberto & Guidorzi, 2007; Wang, Li, & Dong, 2010; Xia & Yu, 2010; Xia, 2012; Xia & Wang, 2013). In this paper we focus on the single-channel speech enhancement.

There are mainly three types of single-channel speech enhancement algorithms. The first type is called the frequency domain method, including the Wiener filter algorithm and the MMSE amplitude spectrum estimation algorithm (Doclo & Moonen, 2005; Ephraim & Malah, 1984; Gerkmann & Hendriks, 2012; Wang et al., 2010). The Wiener algorithm requires estimating the power spectra of speech and noise and its performance depends on the estimation of the speech and noise spectra. The Wiener algorithm has a good noise reduction effect but could muffle speech. The MMSE amplitude spectrum estimation algorithm consists of two parts: a priori SNR estimate and an MMSE spectral amplitude estimate. This algorithm has a better performance than the conventional spectral subtraction algorithm (Boll, 1979), however, it needs an assumption that an estimate of the speech spectrum is available and white noise is Gaussian. The second type is the subspace method. Signal enhancement is to remove the noise subspace and to estimate the clean speech signal from the noisy speech subspace. Traditional subspace methods are suitable for white noise environments (Epharim & Van Trees, 1995a; Ephraim & Van Trees, 1995b). Several improved subspace methods were presented to deal with

☆ This work is supported by the National Natural Science Foundation of China under Grant No. 61179037 and 61473330, and in part from the Research Grants Council of the Hong Kong Special Administrative Region, China (Project no. CUHK416811E).

* Corresponding author.
E-mail addresses: ysxia@fzu.edu.cn (Y. Xia), jwang@mae.cuhk.edu.hk (J. Wang).

http://dx.doi.org/10.1016/j.neunet.2015.03.008
0893-6080/© 2015 Elsevier Ltd. All rights reserved.

colored noise by adding the computational task for eigendecomposition of a non-symmetric matrix (Doclo & Moonen, 2002; Wei & Xia, 2013). The third type is called the parameter estimation-based Kalman filtering method in which the speech signal is modeled as autoregressive process and the speech signal is then recovered from Kalman filtering (Bobillet et al., 2007; Gabrea, 2005; Gabrea et al., 1999; Gannot et al., 1998; Gibson et al., 1991; Labarre et al., 2004; Lee & Jung, 2000; Ning et al., 2006; Park & Choi, 2008). Compared with other two type methods, the Kalman filtering method has no assumption of stationary speech signals.

Traditional parameter estimation-based Kalman filtering algorithms differ only by the choice of the algorithm used to estimate model parameters and the choice of the models adopted for the speech signal and additive noise. For example, Gibson et al. (1991) proposed a method that provides a sub-optimal solution, using the estimate-maximize algorithm based on the maximum likelihood argument. Gannot et al. (1998) proposed the use of the EM algorithm to iteratively estimate the spectral parameters of speech and noise parameters. Lee and Jung (2000) have developed a time-domain approach, without a priory information, to enhance speech signals. Gabrea presented (Gabrea, 2005) an adaptive parameter estimation method. Bobillet et al. presented (Labarre et al., 2004) an optimal smoothing and parameter identification algorithm. These parameter identification algorithms have a standard Gaussian noise assumption (Alimorad & Mahmood, 2011). To deal with the situation in non-Gaussian white noise environments, the Bayesian estimation-based methods were developed (Alliney & Ruzinsky, 1994; Christmas & Everson, 2011; Giannakis & Mendel, 1990; Smidl & Quinn, 2005). Park and Choi presented (Park & Choi, 2008) a neural network method for speech enhancement. Among these methods, the noise statistical distribution is required to be known and there is also a slow speed for parameter learning. Recently, to avoid the requirement of a priori statistical information, two noise constrained estimation-based methods for robust parameter identification were presented by minimizing a generalized least absolute deviation cost function and a quadratic cost function (Xia & Kamel, 2008; Xia, Kamel, & Henry, 2010), respectively. For their implementation, two recurrent neural networks were presented in Xia (2012) and Xia and Yu (2010), respectively. Because the two neural network methods have the total number of neurons which is larger than the sample length of the speech signal, their order of complexity is usually depends on the sample length of the speech signal. So, resulting neural network-based speech enhancement algorithms have a very slower speed.

To increase computational efficiency, we propose a low-dimensional recurrent neural network for fast speech enhancement by using a noise-constrained least squares estimate for Kalman filter parameters. It is shown that the proposed neural network is globally asymptomatically stable to the optimal solution of a noise constrained estimation problem. Because the noise-constrained estimate has a robust performance against non-Gaussian noise, the proposed recurrent neural network-based speech enhancement algorithm can minimize the estimation error of Kalman filtering parameters in non-Gaussian noise. Furthermore, having the low order of complexity, the proposed neural network-based speech enhancement algorithm has a much faster speed than two existing recurrent neural networks-based speech enhancement algorithms. Simulation results show that the proposed recurrent neural network-based speech enhancement algorithm produces a good performance in fast computation and noise reduction.

The paper is organized as follows. In Section 2, autoregressive (AR) model and its noise-constrained estimation are introduced. In Section 3, two existing recurrent neural networks for estimating the AR model parameter are discussed, and a new recurrent neural network with global convergence is proposed. In Section 4, the speech model and Kalman filter are described, and a new recurrent neural network-based speech enhancement algorithm is presented. In Section 5, computed examples are reported. Section 6 gives the concluding remarks of this paper.

2. AR model and estimation

Consider the following pth-order AR signal system:

$$x(t) = \sum_{i=1}^{p} a_i^* x(t-i) + v(t), \tag{1}$$

where p is the known order of the system, $\mathbf{a}^* = [a_1^*, \ldots, a_p^*]^T$ is the unknown AR parameter vector, $v(t)$ is the driving noise, $x(t)$ is an AR signal process with $x(t) = 0$ for $t \leq 0$, and $x(t)$ is observed in additive measurement noise $w(t)$:

$$y(t) = x(t) + w(t), \tag{2}$$

and $w(t)$ is assumed to be uncorrelated with $v(t)$. For simplicity, we denote the noisy signal vector by $\mathbf{y}_t = [y(t-1), \ldots, y(t-p)]^T$, and the measurement noise vector by $\mathbf{w}_t = [w(t-1), \ldots, w(t-p)]^T$. Then the AR signal observation model can be written as

$$y(t) = \mathbf{y}_t{}^T \mathbf{a}^* - n(t), \tag{3}$$

where $n(t) = \mathbf{w}_t{}^T \mathbf{a}^* - w(t) - v(t)$. The problem under study is to estimate AR parameter vector $\mathbf{a}^*$ from noisy observations $\{y(t)\}_1^N$ where N is the number of observations. The most basic approach to estimate the AR parameter vector is the least square (LS) method. The LS estimation minimizers

$$E(\mathbf{a}) = \frac{1}{N} \sum_{t=1}^{N} (y(t) - \mathbf{y}_t{}^T \mathbf{a})^2$$

and is given by

$$\mathbf{a}_{LS} = \left(\frac{1}{N} \sum_{t=1}^{N} \mathbf{y}_t \mathbf{y}_t{}^T \right)^{-1} \left(\frac{1}{N} \sum_{t=1}^{N} \mathbf{y}_t y(t) \right),$$

where $\mathbf{a} = [a_1, \ldots, a_p]^T$. In addition, there is an error between the LS estimate $\mathbf{a}_{LS}$ and the true AR parameter vector $\mathbf{a}^*$:

$$\mathbf{a}_{LS} \approx \mathbf{a}^* - \sigma^2 \hat{R}^{-1} \mathbf{a}^*,$$

where $\hat{R} = \frac{1}{N} \sum_{t=1}^{N} \mathbf{y}_t \mathbf{y}_t{}^T$ and σ^2 is the variance of the measurement noise.

Many AR parameter estimation methods have been developed to improve the LS estimation. For example, one is called the instrumental variable (IV) method (Bobillet et al., 2007; Labarre et al., 2004). Most of the IV algorithms is used for solving a set of high-order Yule–Walker equations. Another is called the bias correction method (Alimorad & Mahmood, 2011) where the AR model parameters, the observation noise variance, and the driving noise variance are estimated in an alternating iteration. The main feature among them is that the estimate of the AR model parameters is usually dependent on the estimate of observation noise variance with an assumption of Gaussian white noise. In practice, the noise corrupted in noisy speech is usually non-Gaussian and colored. Although the Bayesian estimation method developed can handle non-Gaussian noise cases (Christmas & Everson, 2011; Smidl & Quinn, 2005), it requires a priori statistical information.

Recently, to avoid a priori statistical information, two noise-constrained estimation methods were developed in Xia and

Kamel (2008) and Xia et al. (2010), respectively. Let $\mathbf{y} = [y(1), \ldots, y(N)]^T$, $\mathbf{n} = [n(1), \ldots, n(N)]^T$, and

$$B = \begin{pmatrix} y(0) & y(-1) & \ldots & y(1-p) \\ y(1) & y(0) & \ldots & y(2-p) \\ \vdots & \vdots & \ddots & \vdots \\ y(N-1) & y(N-2) & \ldots & y(N-p) \end{pmatrix}. \tag{4}$$

Then the AR model defined in (1) can be further written as a system of linear regression equations in a matrix and vector form:

$$B\mathbf{a}^* - \mathbf{y} - \mathbf{n} = \mathbf{0}.$$

A noise-constrained l_1 estimation method for AR parameters was proposed by solving the following quadratic convex optimization problem (Xia & Kamel, 2008):

$$\begin{array}{ll} \min & f_1(\mathbf{a}, \mathbf{z}) = \|B\mathbf{a} - \mathbf{y} - \mathbf{z}\|_1 \\ \text{s.t} & \mathbf{a} \in R^p,\ \mathbf{z} \in \Omega_\gamma, \end{array} \tag{5}$$

where $(\mathbf{a}, \mathbf{z}) \in R^{p+N}$ is the regression vector, $\|\cdot\|_1$ denotes l_1 norm, $\Omega_\gamma = \{\mathbf{z} \in R^N \mid \mathbf{l} \le \mathbf{z} \le \mathbf{h}\}$, $\mathbf{l} = \gamma_1 E[y(t)]\mathbf{e}$, $\mathbf{h} = \gamma_2 E[y(t)]\mathbf{e}$, $\mathbf{e} = [1, \ldots, 1]^T \in R^N$, and γ_1 and γ_2 are design parameters. Let $(\mathbf{a}_1^*, \mathbf{z}_1^*)$ be an optimal solution of (5). The AR parameter estimate is then given by $\mathbf{a}_1^*$. Furthermore, to avoid the disadvantage of non-smooth cost function in (5), a noise-constrained l_2 estimation method was presented by solving the following quadratic convex optimization problem (Xia et al., 2010):

$$\begin{array}{ll} \min & f_2(\mathbf{a}, \mathbf{z}) = \|B\mathbf{a} - \mathbf{y} - \mathbf{z}\|_2^2 \\ \text{s.t} & \mathbf{a} \in R^p,\ \mathbf{z} \in \Omega_\gamma, \end{array} \tag{6}$$

where $(\mathbf{a}, \mathbf{z}) \in R^{p+N}$ is the regression vector, $\|\cdot\|_2$ denotes l_2 norm. Let $(\mathbf{a}_2^*, \mathbf{z}_2^*)$ be an optimal solution of (6). The AR parameter estimate is then given by $\mathbf{a}_2^*$. The two noise-constrained estimation methods are shown to have a robust performance with a small mean square error and there is no requirement on a priori statistical information, compared with the Bayesian estimation method.

3. Recurrent neural networks

There are many numerical optimization algorithms for solving (5) or (6), but they have at least a computational complexity $O(N(N+p))$ per iteration. In recent decades, in view of the inherent nature of parallel and distributed information processing in neural networks, neural networks are promising computational models for real-time applications. Feedforward neural networks and recurrent neural networks are two major classes of neural network models. Feedforward neural networks are mainly used for approximation and prediction (Mandic & Chambers, 2001). In contrast, recurrent neural networks, as dynamical systems, are usually used as computational models for solving optimization problems (Cichocki & Amari, 2002; Xia, 2004; Xia, Gang, & Wang, 2008). Mathematically, the recurrent neural network approach converts an optimization problem into a neural dynamical system so that whose state output will give the optimal solution of the optimization problem and then the optimal solution can be obtained by tracking the state trajectory of the designed dynamical system based on hardware and software implementation. Compared with the conventional numerical optimization method, the recurrent neural network approach has a potential capability for solving the optimization problems due to having the low computational complexity and inherent dynamical nature.

3.1. Related recurrent neural networks

For computing the noise-constrained l_1 estimate and noise-constrained l_2 estimate, two recurrent neural networks for solving (5) and (6) were presented, respectively. One continuous-time recurrent neural network for solving (5) was proposed in Xia and Kamel (2008):

State equation

$$\begin{cases} \dfrac{d\mathbf{x}}{dt} = -\mu B^T g^0(\mathbf{c} + B\mathbf{x} - \mathbf{y} - \mathbf{z}) \\ \dfrac{d\mathbf{c}}{dt} = -\mu\{\mathbf{w}_z - g^0(\mathbf{c} + B\mathbf{x} - \mathbf{y} - \mathbf{z}) + g^1(\mathbf{z} + \mathbf{c})\} \\ \dfrac{d\mathbf{z}}{dt} = -\mu\{\mathbf{z} - g^1(\mathbf{z} + \mathbf{c}) + \mathbf{e} + g^0(\mathbf{c} + B\mathbf{x} - \mathbf{y} - \mathbf{z})\}. \end{cases} \tag{7}$$

Output equation

$$\mathbf{a}(t) = \mathbf{x}(t) \tag{8}$$

where $(\mathbf{x}(t), \mathbf{c}(t), \mathbf{z}(t)) \in R^p \times R^N \times R^N$ is the state trajectory vector, $\mathbf{a}(t) \in R^p$ is the output trajectory vector, $\mu > 0$ is a design constant, $\mathbf{w}_z = \mathbf{c} + BB^T\mathbf{c} - \mathbf{z}$, B is defined in (4), $g^0(\mathbf{z})$ is the projection on the set Ω_γ defined in (5), and $g^1(\mathbf{z})$ is the projection on the set $X_1 = \{\mathbf{z} \in R^N \mid max_j|z_j| \le 1\}$ where $g^1(\mathbf{z}) = [g^1(z_1), \ldots, g^1(z_N)]^T$, $g^0(\mathbf{z}) = [g^0(z_1), \ldots, g^0(z_N)]^T$, and for $i = 1, \ldots, N$

$$\begin{aligned} g^1(z_i) &= \begin{cases} -1 & z_i < -1 \\ z_i & -1 \le z_i \le 1 \\ 1 & z_i > 1 \end{cases}, \\ g^0(z_i) &= \begin{cases} l_1 & z_i < l_1 \\ z_i & l_i \le z_i \le h_i \\ h_i & z_i > h_i. \end{cases} \end{aligned} \tag{9}$$

Another discrete-time recurrent neural network for solving (6) was proposed in Xia et al. (2010):

State equation

$$\begin{cases} \mathbf{x}(k+1) = (I - \beta\hat{B}^T\hat{B})\mathbf{x}(k) + \beta\hat{B}^T\mathbf{z}(k) + q \\ \mathbf{z}(k+1) = (1-\beta)\mathbf{z}(k) + \beta\hat{g}^0(\hat{B}\mathbf{x}(k) - \hat{\mathbf{y}}). \end{cases} \tag{10}$$

Output equation

$$\mathbf{a}(k+1) = \mathbf{x}(k+1) \tag{11}$$

where $(\mathbf{x}(k), \mathbf{z}(k)) \in R^p \times R^N$ is the state trajectory vector, $\mathbf{a}(k) \in R^p$ is the output trajectory vector, $I \in R^{p\times p}$ is a unit matrix, $\hat{B} = B/\alpha$, $\hat{\mathbf{y}} = \mathbf{y}/\alpha$, $\alpha = \|B\|_2^2$, $q = \beta\hat{B}^T\mathbf{y}$, $\beta > 0$ is a given step length, and $\hat{g}^0(\mathbf{z})$ is the projection on the set $\hat{\Omega}_\gamma = \{\mathbf{z} \in R^N \mid \hat{\mathbf{l}} \le \mathbf{z} \le \hat{\mathbf{h}}\}$, $\hat{\mathbf{l}} = \mathbf{l}/\alpha$, $\hat{\mathbf{h}} = \mathbf{h}/\alpha$, $\hat{g}^0(\mathbf{z}) = [\hat{g}^0(z_1), \ldots, \hat{g}^0(z_N)]^T$ and

$$\hat{g}^0(y_i) = \begin{cases} \hat{l}_1 & z_i < \hat{l}_1 \\ z_i & \hat{l}_i \le z_i \le \hat{h}_i \\ \hat{h}_i & z_i > \hat{h}_i. \end{cases} \quad (i = 1, \ldots, N) \tag{12}$$

The discrete-time recurrent neural network in (10) and (11) may be viewed as a discrete version of the following continuous-time recurrent neural network, called as the extended projection neural network in Xia (2004):

State equation

$$\begin{cases} \dfrac{d\mathbf{x}(t)}{dt} = -\mu\{\hat{B}^T\mathbf{z}(t) + \hat{B}^T\mathbf{y} - \hat{B}^T\hat{B}\mathbf{x}(t)\} \\ \dfrac{d\mathbf{z}(t)}{dt} = -\mu\{g^0(\hat{B}\mathbf{x}(t) - \hat{\mathbf{y}}) - \mathbf{z}(t)\}. \end{cases} \tag{13}$$

Output equation

$$\mathbf{a}(t) = \mathbf{x}(t) \tag{14}$$

where $(\mathbf{x}(t), \mathbf{z}(t)) \in R^p \times R^N$ is the state trajectory vector, $\mathbf{a}(t) \in R^p$ is the output trajectory vector, and $\mu > 0$ is a design constant. It was reported in Xia (2004) and Xia and Kamel (2008) that the output trajectory of two recurrent neural networks defined in (7) and (10) or (13) is globally convergent to the optimal solution of (5) and (6), respectively. On the other side, the two recurrent neural networks have the total number of neurons being $p + 2N$ and $p + N$, respectively. Thus they have a model complexity being $O(N)$ at least. As a result, the speed of the two recurrent neural networks will be slow when N becomes large. Also, to determine the optimal error set defined in (5) and (6), a sequential cross-validation method was employed in Xia and Kamel (2008) and Xia et al. (2010). It needs an additional computation cost.

3.2. Proposed recurrent neural network

For fast AR parameter estimation, we first introduce the following suboptimal noise error set in (6):

$$\Omega_{\gamma^*} = \{\mathbf{z} \in R^N \mid \mathbf{l}^* \le \mathbf{z} \le \mathbf{h}^*\} \tag{15}$$

where $\mathbf{h}^* = max\{B\mathbf{a}_{LS} - \mathbf{y}, \mathbf{m}\}$, $\mathbf{l}^* = min\{B\mathbf{a}_{LS} - \mathbf{y}, \mathbf{m}\}$, $\mathbf{a}_{LS}$ is a least-square estimate, $\mathbf{m} = E[y(t)]\mathbf{e}$, and $\mathbf{e} = [1, \ldots, 1]^T \in R^N$. That is, (6) becomes the following quadratic optimization problem:

$$\begin{array}{ll} \min & f_2(\mathbf{x}, \mathbf{z}) = \|B\mathbf{x} - \mathbf{y} - \mathbf{z}\|_2^2 \\ \text{s.t} & \mathbf{x} \in R^p,\ \mathbf{z} \in \Omega_{\gamma^*}. \end{array} \tag{16}$$

Let $(\hat{\mathbf{x}}^*, \hat{\mathbf{z}}^*)$ be the optimal solution of (16). The noise-constrained estimate of the AR parameter is given by $\hat{\mathbf{x}}^*$. Second, we propose a new recurrent neural network for solving (16) as follows:

State equation

$$\frac{d\mathbf{x}(t)}{dt} = -\mu\{B^T g(B\mathbf{x} - \mathbf{y}) + B^T\mathbf{y} - B^T B\mathbf{x}\}. \tag{17}$$

Output equation

$$\mathbf{a}(t) = \mathbf{x}(t) \tag{18}$$

where $\mathbf{x} \in R^p$ is the state trajectory vector, $\mathbf{a}(t) \in R^p$ is the output trajectory vector, B and $\mathbf{y}$ are defined in (6), $g(\mathbf{z})$ is the projection on the set Ω_{γ^*} given by $g(\mathbf{z}) = [g(z_1), \ldots, g(z_N)]^T$, and

$$g(y_i) = \begin{cases} l_1^* & z_i < l_1^* \\ z_i & l_i^* \le z_i \le h_i^* \\ h_i^* & z_i > h_i^*. \end{cases} \quad (i = 1, \ldots, N). \tag{19}$$

It is seen that the proposed recurrent neural network in (17) has p neurons. By contrast, the recurrent neural network defined in (7) has $2N + p$ neurons being and the recurrent neural network defined in (13) has $N + p$ neurons. Because p is the speech model order and N is the number of observations, $p \ll N$. As a result, the proposed recurrent neural network has a very smaller model size than both (7) and (13). Table 1 lists model complexity and computational complexity of three recurrent neural networks. From Table 1 we see that the proposed recurrent neural network has lower model complexity than the other two ones in terms of order of complexity. Furthermore, the proposed recurrent neural network is guaranteed to be global asymptotically stable at its equilibrium point (this proof is given in the Appendix) and thus its output trajectory can converge globally to the noise-constrained estimation, that is, the optimal solution of (6).

Remark 1. It should be noted that the proposed recurrent neural network is of continuous-time. In software simulation, it is often taken as the discrete-time form by using numerical ODE techniques. For example, applying the well-known Euler technique to (17), we have the following discrete-time recurrent neural network (Xia, Lin, & Zheng, 2014):

$$\mathbf{x}(k+1) = \mathbf{x}(k) + h_k(B^T g(B\mathbf{x}(k) - \mathbf{y}) + B^T\mathbf{y} - B^T B\mathbf{x}(k))$$

where $h_k > 0$ is a step length.

Table 1
Complexity of three recurrent neural network models.

Model	Computational complexity	Neurons
New model	$2Np + p^2$	p
Existing model (12)	$2N^2 + N(3p + 11) + p^2$	$2N+p$
Existing model (14)	$2Np + N + p^2$	$N + p$

4. Proposed neural network-based speech enhancement

4.1. Speech model and Kalman filtering

Consider noisy speech model:

$$y(n) = s(n) + w(n) \tag{20}$$

where $y(n)$ is the nth sample of the noisy speech observation and $w(n)$ is additive observed noise with covariance matrix R_w, which is assumed to be uncorrelated with the drive noise sequence $v(n)$. In a special case that the observation noise is a white noise, R_w is a diagonal matrix and its diagonal elements represent the noise variances. The purpose of speech enhancement is to estimate the clean speech $s(n)$ from received noisy speech observation $y(n)$.

Let the nth sample of speech signal $s(n)$ be modeled as:

$$s(n) = \sum_{i=1}^{p} \hat{a}_i^* s(n-i) + v(n) \tag{21}$$

where $\{\hat{a}_i^*\}$ are AR parameters of the speech signal, $v(n)$ is the nth sample of the driving white noise with variance σ_v^2, and p is the speech model order, which is usually taken as 8 or so.

Define a p-dimensional speech state vector, measured noisy speech vector, measured noise vector, and deriving noise vector as $\mathbf{u}(n) = [s(n-p+1), \ldots, s(n-1), s(n)]^T$, $\mathbf{y}(n) = [y(n-p+1), \ldots, y(n-1), y(n)]^T$, $\mathbf{w}(n) = [w(n-p+1), \ldots, w(n-1), w(n)]^T$, $\mathbf{v}(n) = [v(n-p+1), \ldots, v(n-1), v(n)]^T$, and the transition matrix (called companion matrix) as

$$F = \begin{pmatrix} 0 & 1 & 0 & \ldots & 0 & 0 \\ 0 & 0 & 1 & \ldots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \ldots & 0 & 1 \\ \hat{a}_p^* & \hat{a}_{p-1}^* & \hat{a}_{p-2}^* & \ldots & \hat{a}_2^* & \hat{a}_1^* \end{pmatrix} \tag{22}$$

respectively. Together with the noisy speech model, the state-space model of the measured speech signal is expressed as

$$\begin{cases} \mathbf{u}(n) = F\mathbf{u}(n-1) + G\mathbf{v}(n) \\ \mathbf{y}(n) = H\mathbf{u}(n) + \mathbf{w}(n) \end{cases} \tag{23}$$

where H is a p-th order identity matrix and $G = [0, \ldots, 0, 1]^T \in R^p$. Then the standard Kalman filter estimation and updating equations for speech enhancement are as follows:

$$\begin{cases} \mathbf{e}(n) = \mathbf{y}(n) - \hat{\mathbf{u}}(n|n-1) \\ \mathbf{K}(n) = P(n|n-1)(R_w + P(n|n-1))^{-1} \\ \hat{\mathbf{u}}(n|n) = \hat{\mathbf{u}}(n|n-1) + \mathbf{K}(n)\mathbf{e}(n) \\ P(n|n) = (I - \mathbf{K}(n))P(n|n-1) \\ \hat{\mathbf{u}}(n+1|n) = F\hat{\mathbf{u}}(n|n) \\ P(n+1|n) = FP(n|n)F^T + \sigma_v^2 GG^T \end{cases} \tag{24}$$

where $\mathbf{e}(n)$ is the innovation vector, $K(n)$ is the Kalman gain matrix, $\hat{\mathbf{u}}(n|n)$ represents the filtered estimate of state vector $\mathbf{u}(n)$, $\hat{\mathbf{u}}(n|n-1)$ is the minimum mean square estimate of the state vector $\mathbf{u}(n)$ given the past observation $y(1), \ldots, y(n-1)$, $P(n|n)$ is the filtered state error covariance matrix, and $P(n|n-1)$ is predicted state error correlation matrix. The speech estimate at n is then given by $\hat{s}(n) = G^T\hat{\mathbf{u}}(n|n)$.

It is seen that the Kalman filter parameters include the speech signal AR parameters $\{a_i\}$ in the transition matrix F, the deriving noise variance σ_v^2, and the covariance matrix of the measured noise w. There exist conventional algorithms for these parameter estimation, but they have a requirement of a priori statistical information and their computation speed is dependent on sample length N. For fast speech enhancement, in next section we will propose a new recurrent neural network-based speech enhancement algorithm.

Remark 2. Based on the standard Kalman filtering theory, the optimal minimum mean-square linear state estimate is obtained by the Kalman filtering when noise is white. When noise is colored, the state estimate may not optimal. To detail with this case, one possible approach is the pre-whiten technique (Wei & Xia, 2013).

Remark 3. If the transition matrix is taken as:

$$\hat{F} = \begin{pmatrix} \hat{a}_1^* & \hat{a}_2^* & \hat{a}_3^* & \dots & \hat{a}_{p-1}^* & \hat{a}_p^* \\ 1 & 0 & 0 & \dots & 0 & 0 \\ 0 & 1 & 0 & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & 1 & 0 \end{pmatrix}, \tag{25}$$

the state-space model of the measured speech signal is expressed as

$$\begin{cases} \mathbf{u}(n) = \hat{F}\mathbf{u}(n-1) + \hat{G}\mathbf{v}(n) \\ \mathbf{y}(n) = H\mathbf{u}(n) + \mathbf{w}(n) \end{cases} \tag{26}$$

where $\hat{G} = [1, 0, \dots, 0]^T$. The speech estimate at n is then given by the first element of the filtered estimate of the state vector.

4.2. Proposed speech enhancement algorithm

To introduce a new recurrent neural network-based speech enhancement algorithm, the Kalman filtering parameters need being estimated previously.

First, to estimate covariance matrix of the measured noise v, we use the noisy speech observation and estimated AR model parameters. In the case that measured noise is white, we use the well-known formulation:

$$(R_y - \sigma_w^2 I)\mathbf{a} = r_y \tag{27}$$

where $r_y = E[\mathbf{y}(t)y(t)]$ and $R_y = E[\mathbf{y}(t)\mathbf{y}^T(t)]$. Let $\hat{\mathbf{a}}$ be obtained by the proposed recurrent neural network. The variance of the measured noise is thus estimated by

$$\hat{\sigma}_w^2 = \frac{\hat{\mathbf{a}}^T R_y \hat{\mathbf{a}} - r_y^T \hat{\mathbf{a}}}{\|\hat{\mathbf{a}}\|_2^2}. \tag{28}$$

Then the covariance matrix of the measured noise is given by $R_w = \sigma_w^2 I$ where I is the unit matrix. In the case that the measured noise is colored, the covariance matrix of the measured noise is approximately estimated by $R_w = E[\mathbf{w}(t)\mathbf{w}^T(t)]$ during the speech-absent frame. Finally, the variance estimation of driving noise v can be used by

$$\hat{\sigma}_v^2 = E[y^2] - r_y^T \hat{\mathbf{a}} - \hat{\sigma}_w^2. \tag{29}$$

The proposed recurrent neural network-based speech enhancement algorithm is described as follows:

Let N be the sample length of each speech frame. Let $\mathbf{u}(0|0) = 0$ and let $P(0|0)$ be a unit matrix. Given AR order p.

Step 1. Compute matrix B defined in (4) based on the noisy speech signal $\{y(n)\}$, compute the AR parameter estimate by using the proposed recurrent neural network, and compute the state transition matrix F by using the estimated vector $\hat{\mathbf{a}}^*$.

Step 2. Compute the covariance matrix of the measured noise by using (28) or by $R_v = E[\mathbf{w}(t)\mathbf{w}^T(t)]$ during the speech-absent frame, and compute the deriving noise variance by using (29).

Step 3. Perform the Kalman filtering defined in (24) to obtain $\hat{\mathbf{u}}(n|n)$.

Step 4. Compute speech signal estimate: $z(n) = G^T\hat{\mathbf{u}}(n|n)$.

The novelty of the proposed recurrent neural network-based speech enhancement algorithm is twofold. First, for learning Kalman filtering parameters, the conventional Kalman filtering methods for speech enhancement (Bobillet et al., 2007; Gabrea, 2005; Labarre et al., 2004; Ning et al., 2006; Roberto & Guidorzi, 2007) used the recursive least square algorithm with variable forgetting factor under an assumption of Gaussian noise. By contrast, because the noise-constrained estimation approach has a robust performance against non-Gaussian noise, the proposed neural network algorithm can minimize the estimation error of the Kalman filtering parameters and has no tuning forgetting factor. Second, the existing recurrent neural networks (Xia, 2004; Xia & Kamel, 2008; Xia et al., 2010) have a model complexity being $O(N)$, while the proposed recurrent neural network has a model complexity being $O(1)$ only. Because N is a large sample length of the speech signal, the proposed recurrent neural network-based speech enhancement algorithm can have a much faster speed than two existing neural networks-based speech enhancement algorithms (Xia, 2012; Xia & Yu, 2010). Simulation results in Section 5 confirm this good performance.

Remark 4. It should be noted that the proposed recurrent neural network is used for learning Kalman filtering parameters including the speech signal AR parameters, the deriving noise variance, and the measured noise variance. As a result, it is different from training neural networks with the Kalman filter in literatures (Goh & Mandic, 2007; Simon, 2002).

Remark 5. It should be noted that the AR model parameters $\{\hat{a}_i^*\}$ are usually time dependent on different speech frame but time independent on each speech frame given. As for estimation and stability of time-variant AR parameters, reader may refer to literatures Mandic and Chambers (2000a) and Mandic and Chambers (2000b).

5. Computational examples

In this section, we give illustrative examples to demonstrate the effectiveness of the proposed recurrent neural network-based speech enhancement algorithm. We evaluate the algorithm performance by using the signal to noise radio (SNR), the spectral distortion (SD), and the quality of enhanced speech components. The SNR is defined by

$$SNR = 10\log\frac{\sum_{n=1}^{N_1} x(n)^2}{\sum_{n=1}^{N_1}[x(n) - \hat{x}(n)]^2}$$

where $\hat{x}(n)$ is the estimated speech signal and N_1 is the total sample length. SD is given in Wang et al. (2010):

$$SD = \frac{1}{4N_1}\sum_{i=1}^{N_1/64}\sum_{k=0}^{255} 20\left|\log_{10}|S(k)| - \log_{10}\left|\hat{S}(k)\right|\right| \tag{30}$$

where $S(k)$ and $\hat{S}(k)$ are the spectra of the clean and enhanced speech after normalizing $s(k)$ and $\hat{s}(k)$ to be zero mean and unit variance, respectively. In our experiments, two clean speech data, called "sp01" and "sp04", are collected from speech corpus (NOIZEUS). The frame size was 256 samples with 50% overlap. The speech AR order is taken as $p = 10$. The simulation is conducted in MATLAB.

Table 2
SNR results of three methods in Example 1.

Speech	Algorithm	AR colored noise (0 dB)	AR colored noise (5 dB)	CUP (s)
sp_01	Proposed algorithm	8.11	11.11	6.58
	(7)-based algorithm	8.01	10.98	9.389
	(13)-based algorithm	8.09	11.12	1615.36
sp_04	Proposed algorithm	8.21	10.55	4.85
	(7)-based algorithm	8.16	10.48	8.36
	(13)-based algorithm	8.20	10.51	1216.58

Table 3
SD results of three methods in Example 1.

Speech	Algorithm	AR colored noise (0 dB)	AR colored noise (5 dB)	CUP (s)
sp_01	Proposed algorithm	6.03	5.76	6.58
	(7)-based algorithm	6.12	5.80	9.389
	(13)-based algorithm	6.08	5.79	1615.36
sp_04	Proposed algorithm	6.36	5.87	4.85
	(7)-based algorithm	6.39	5.97	8.36
	(13)-based algorithm	6.37	5.89	1216.58

Table 4
SNR results of three methods in Example 2.

Speech	Algorithm	Airport noise (dB)		Babble noise (dB)		Train noise (dB)		CUP (s)
		0	5	0	5	0	5	
sp_01	Proposed algorithm	4.38	7.82	4.48	6.28	5.02	8.45	5.07
	Adaptive algorithm	−1.71	5.99	1.22	6.23	2.97	6.20	18.9
	Subspace algorithm	3.35	7.64	4.6	6.25	3.82	8.36	12.59
sp_04	Proposed algorithm	4.70	8.18	4.20	8.51	4.74	8.58	4.26
	Adaptive algorithm	2.85	5.09	2.91	5.55	3.46	6.56	10.78
	Subspace algorithm	3.96	8.08	4.20	7.45	5.11	8.41	9.71

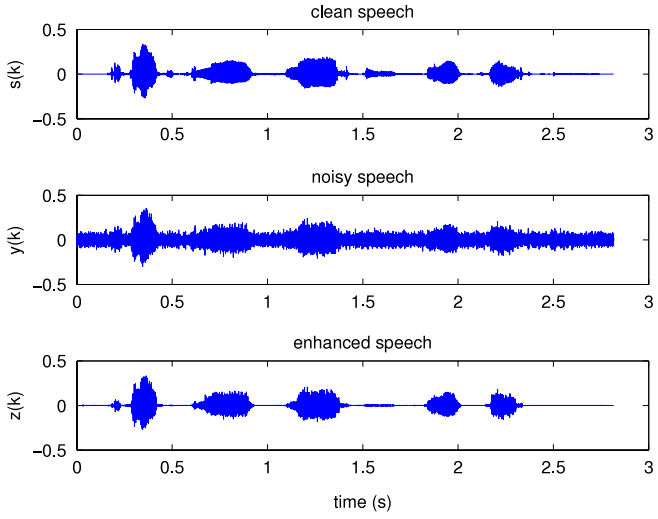

Fig. 1. Waveform results of the clean, noisy speech, and enhanced speech by the proposed algorithm in AR Colored noisy speech sp01 (0 dB).

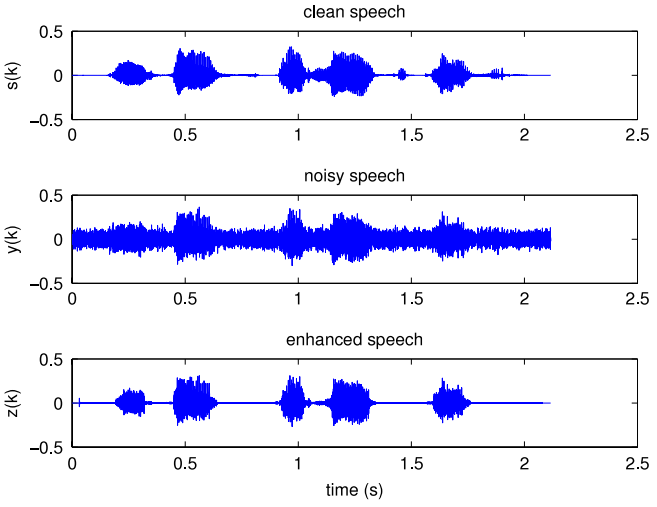

Fig. 2. Waveform results of the clean, noisy speech, and enhanced speech by the proposed algorithm in AR Colored noisy speech sp04 (0 dB).

Example 1. Consider the two clean speech signals corrupted by colored observation noise modeled as

$$v(k) = 1.2v(k-1) - 0.9559v(k-2) + 0.6727v(k-3) + u(k)$$

where $u(k)$ is white Gaussian noise. We study two different noise levels with variance 0.018 such that input SNR is 0 dB. The noisy speech signal has the sampling frequency of 8000 Hz. 256 samples are used for each frame. The waveform results of the clean speech (sp01), noisy speech, and restored speech by the proposed algorithm are depicted in Fig. 1. The waveform results of the clean speech (sp04), noisy speech, and restored speech by the proposed algorithm are depicted in Fig. 2. For a comparison, we also perform the neural network in (7)-based algorithm and the neural network in (13)-based algorithm, respectively. The obtained output SNRs and SDs by three algorithms and their computation time are summarized in Tables 2 and 3, respectively at noise level 0 dB and 5 dB with input SD value being 10.26 and 9.01, respectively. We see that the output SNRs given by the three algorithms are all improved and the output SDs are all reduced greatly. Furthermore, the proposed algorithm has a much faster speed than other two algorithms.

Example 2. Consider the two clean speech signals corrupted by three real colored noise (Airport noise, Babble noise, and Train noise) at levels 0 dB and 5 dB, respectively. For a comparison, we

Table 5
SD results of four methods in Example 2.

Speech	Algorithm	Airport noise (dB)		Babble noise (dB)		Train noise (dB)		CUP (s)
		0	5	0	5	0	5	
sp_01	Proposed algorithm	5.74	5.54	6.02	5.45	5.95	5.39	5.07
	Adaptive algorithm	10.07	6.53	7.67	6.16	8.90	7.57	18.9
	Subspace algorithm	6.74	5.58	6.18	5.48	8.04	6.39	12.59
sp_04	Proposed algorithm	5.85	5.69	6.27	5.50	6.21	5.39	4.26
	Adaptive algorithm	8.38	7.35	8.51	7.26	9.04	7.64	10.78
	Subspace algorithm	7.11	5.71	7.34	6.16	6.41	5.60	9.71

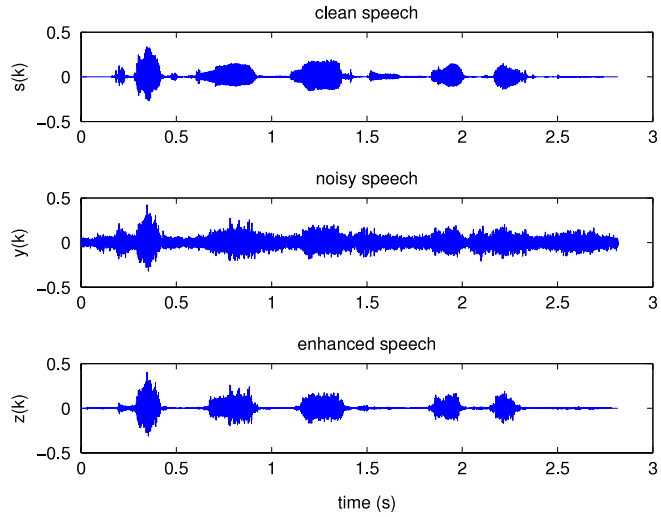

Fig. 3. Waveform results of the clean, noisy speech, and enhanced speech by the proposed algorithm in train noisy speech sp01 (0 dB).

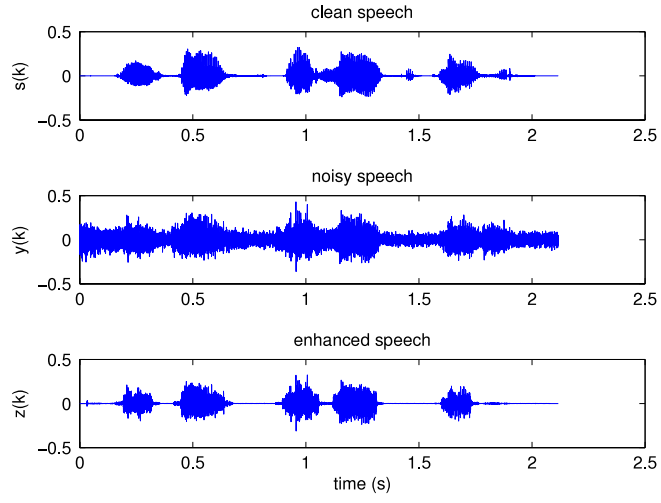

Fig. 4. Waveform results of the clean, noisy speech, and enhanced speech by the proposed algorithm in train noisy speech sp04 (0 dB).

perform the proposed algorithm, the adaptive estimation-based Kalman filter algorithm (Gabrea, 2005), and the subspace algorithm (Ephraim & Van Trees, 1995b), respectively. The obtained output SNRs and SDs by three algorithms and their computation time are summarized in Tables 4 and 5 under three different noise. We see that the output SNRs by the proposed algorithm are greatly enhanced and are totally higher than other two algorithms. Moreover, the output SDs are all reduced greatly at different noise levels. It shows that the proposed method has less spectral distortion than the other two methods. Furthermore, the proposed algorithm has a faster speed than other two algorithms also. Finally,

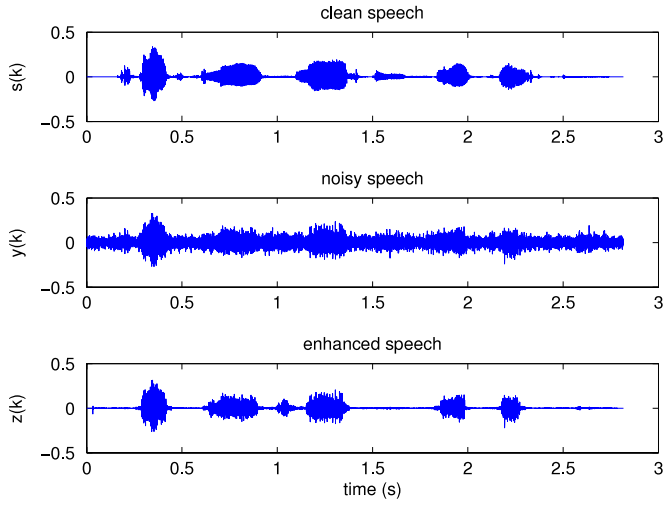

Fig. 5. Waveform results of the clean, noisy speech, and enhanced speech by the proposed algorithm in airport noisy speech sp01 (0 dB).

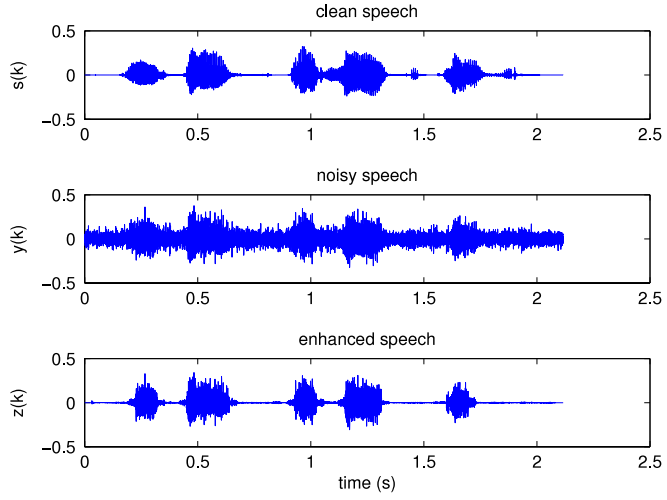

Fig. 6. Waveform results of the clean, noisy speech, and enhanced speech by the proposed algorithm in airport noisy speech sp04 (0 dB).

Figs. 3–8 list the waveform results of the clean speech (sp01, sp04), noisy speech, and enhanced speech of the proposed algorithm. Clearly, the waveform results by the proposed algorithm improve greatly the input waveform results.

6. Conclusion

This paper has developed a new recurrent neural network-based Kalman filter for speech enhancement based on a noise-constrained estimate. The proposed recurrent neural network is shown to be globally asymptomatically stable to the noise-constrained estimate. Because the proposed neural network is of a

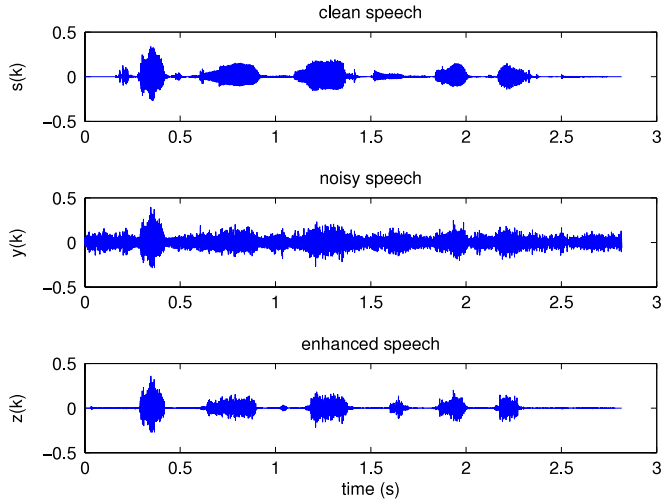

Fig. 7. Waveform results of the clean, noisy speech, and enhanced speech by the proposed algorithm in babble noisy speech sp01 (0 dB).

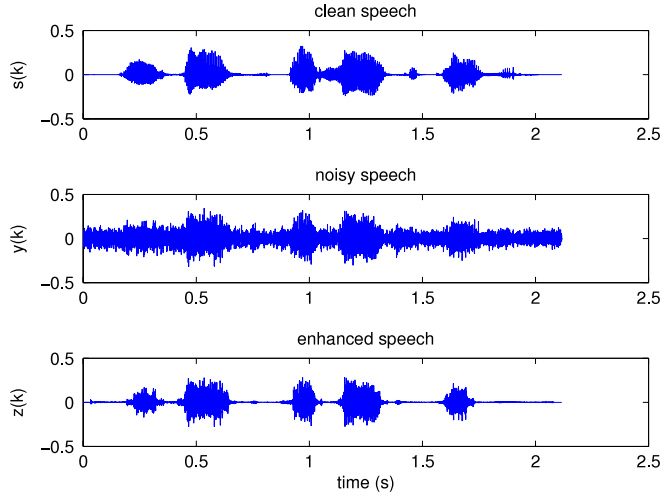

Fig. 8. Waveform results of the clean, noisy speech, and enhanced speech by the proposed algorithm in babble noisy speech sp04 (0 dB).

low-dimensional model, the resulting speech enhancement algorithm has a much faster speed than two existing neural networks-based speech enhancement algorithms. Compared with other Kalman filter-based speech enhancement algorithms, because the noise-constrained estimate has a robust performance against non-Gaussian noise, the proposed neural network-based speech enhancement algorithm can minimize the estimation error of the Kalman filtering parameters and has no tuning forgetting factor. Computed results show that the proposed neural network-based speech enhancement algorithm has a good performance in both fast computation and noise reduction.

Acknowledgments

The authors thank the associate editor and reviewers for their encouragement and valued comments, which helped in improving the quality of the paper.

Appendix. Asymptotic stability of the proposed neural network in (17)

Proof. First, we show that the equilibrium point of the proposed neural network equals the noise-constrained estimate defined in (16). By the optimality condition (Boyd & Vandenberghe, 2006) we see that $(\hat{\mathbf{x}}^*, \hat{\mathbf{z}}^*)$ is an optimal solution of (16) if and only if for any $\mathbf{x} \in R^p$, $\forall \mathbf{z} \in \Omega_{\gamma^*}$

$$(\mathbf{x}-\mathbf{x}^*)^T \frac{\partial f_2(\mathbf{x},\mathbf{z})}{\partial \mathbf{x}}|_{(\mathbf{x}^*,\mathbf{z}^*)} + (\mathbf{z}-\mathbf{z}^*)^T \frac{\partial f_2(\mathbf{x},\mathbf{z})}{\partial \mathbf{z}}|_{(\mathbf{x}^*,\mathbf{z}^*)} \geq 0 \qquad (31)$$

where

$$\begin{cases} \dfrac{\partial f_2(\mathbf{x},\mathbf{z})}{\partial \mathbf{x}} = B^T B\mathbf{x} - B^T\mathbf{z} - B^T\mathbf{y} \\ \dfrac{\partial f_2(\mathbf{x},\mathbf{z})}{\partial \mathbf{z}} = \mathbf{z} - B\mathbf{x} + \mathbf{y}. \end{cases}$$

Furthermore, $(\mathbf{x}^*, \mathbf{z}^*)$ satisfies (31) if and only if $(\mathbf{x}^*, \mathbf{z}^*)$ satisfies the following

$$\left.\frac{\partial f_2(\mathbf{x},\mathbf{z})}{\partial \mathbf{x}}\right|_{(\mathbf{x}^*,\mathbf{z}^*)} = 0$$

and

$$(\mathbf{z}-\mathbf{z}^*)^T \frac{\partial f_2(\mathbf{x},\mathbf{z})}{\partial \mathbf{z}}|_{(\mathbf{x}^*,\mathbf{z}^*)} \geq 0, \quad \forall \mathbf{z} \in \Omega_{\gamma^*}. \qquad (32)$$

From the projection theorem (Boyd & Vandenberghe, 2006) it follows that $(\mathbf{x}^*, \mathbf{z}^*)$ satisfies (32) if and only if $(\mathbf{x}^*, \mathbf{z}^*)$ satisfies equation $\mathbf{z} = g(B\mathbf{x} - \mathbf{y})$ where projection function g is defined in (19). It follows that $(\mathbf{x}^*, \mathbf{z}^*)$ is an optimal solution of (16) if and only if $(\mathbf{x}^*, \mathbf{z}^*)$ satisfies the following system of two equations:

$$\begin{cases} B^T B\mathbf{x} - B^T\mathbf{z} - B^T\mathbf{y} = 0, \\ g(B\mathbf{x} - \mathbf{y}) - \mathbf{z} = 0. \end{cases} \qquad (33)$$

If $\mathbf{x}_0^*$ is an equilibrium point of the proposed neural network in (17). Then

$$B^T B\mathbf{x}_0^* - B^T g(B\mathbf{x}_0^* - \mathbf{y}) + B^T\mathbf{y} = 0.$$

Let $\mathbf{z}_0^* = g(B\mathbf{x}_0^* - \mathbf{y})$. Then

$$B^T B\mathbf{x}_0^* - B^T\mathbf{z}_0^* - B^T\mathbf{y} = 0.$$

It implies that $(\mathbf{x}_0^*, \mathbf{z}_0^*)$ satisfies (33). Thus $(\mathbf{x}_0^*, \mathbf{z}_0^*)$ is an optimal solution of (16) where $\mathbf{x}_0^*$ is a noise-constrained estimate defined in (16).

Now, we show the global asymptotic stability of the proposed neural network in (17). Let the mapping

$$F(\mathbf{x}) = B^T B\mathbf{x} - B^T g(B\mathbf{x} - \mathbf{y}) + B^T\mathbf{y}$$

and let $\mathbf{x}_0^*$ be one equilibrium point of the proposed neural network in (17). Then $F(\mathbf{x}_0^*) = 0$. Assume that there exists another equilibrium point $\hat{\mathbf{x}}$ such that $F(\hat{\mathbf{x}}) = 0$. Then

$$\begin{aligned}(\hat{\mathbf{x}}-\mathbf{x}_0^*)^T(F(\hat{\mathbf{x}}) - F(\mathbf{x}_0^*)) &= (\hat{\mathbf{x}}-\mathbf{x}_0^*)^T B^T B(\hat{\mathbf{x}}-\mathbf{x}_0^*) \\ &\quad - (\hat{\mathbf{x}}-\mathbf{x}_0^*)^T B^T(g(B\hat{\mathbf{x}}-\mathbf{y}) - g(B\mathbf{x}_0^*-\mathbf{y})) \\ &= \|B(\hat{\mathbf{x}}-\mathbf{x}_0^*)\|_2^2 - ((B\hat{\mathbf{x}}-\mathbf{y}) - (B\mathbf{x}_0^*-\mathbf{y}))^T(g(B\hat{\mathbf{x}}-\mathbf{y}) \\ &\quad - g(B\mathbf{x}_0^*-\mathbf{y})).\end{aligned}$$

Because

$$\begin{aligned}&((B\hat{\mathbf{x}}-\mathbf{y}) - (B\mathbf{x}_0^*-\mathbf{y}))^T(g(B\hat{\mathbf{x}}-\mathbf{y}) \\ &\quad - g(B\mathbf{x}_0^*-\mathbf{y})) \leq \|(B\hat{\mathbf{x}}-\mathbf{y}) - (B\mathbf{x}_0^*-\mathbf{y})\|_2^2\end{aligned}$$

and the equality above holds only when $B\hat{\mathbf{x}} - \mathbf{y} = B\mathbf{x}_0^* - \mathbf{y}$, we have

$$(\hat{\mathbf{x}}-\mathbf{x}_0^*)^T(F(\hat{\mathbf{x}}) - F(\mathbf{x}_0^*)) \geq 0$$

and the equality above holds only when $B\hat{\mathbf{x}} - \mathbf{y} = B\mathbf{x}_0^* - \mathbf{y}$. That is, $B\hat{\mathbf{x}} = B\mathbf{x}_0^*$. Note that $rank(B) = p$. It implies that $\hat{\mathbf{x}} = \mathbf{x}_0^*$. So, $\mathbf{x}_0^*$ is only one equilibrium point of (17).

Finally, we define a Lyapunov function:

$$V(\mathbf{x}) = \frac{1}{2}\|\mathbf{x} - \mathbf{x}_0^*\|_2^2, \quad \mathbf{x} \in R^p.$$

Then

$$\frac{dV(\mathbf{x}(t))}{dt} = (\mathbf{x} - \mathbf{x}_0^*)^T \frac{d\mathbf{x}}{dt} = -\mu(\mathbf{x} - \mathbf{x}_0^*)^T (F(\mathbf{x}) - F(\mathbf{x}_0^*)).$$

Because

$$(\mathbf{x} - \mathbf{x}_0^*)^T (F(\mathbf{x}) - F(\mathbf{x}_0^*)) = \|B(\mathbf{x} - \mathbf{x}_0^*)\|_2^2 - ((B\mathbf{x} - \mathbf{y}) - (B\mathbf{x}_0^* - \mathbf{y}))^T (g(B\mathbf{x} - \mathbf{y}) - g(B\mathbf{x}_0^* - \mathbf{y})) \geq 0,$$

we have

$$\frac{dV(\mathbf{x}(t))}{dt} = (\mathbf{x} - \mathbf{x}_0^*)^T \frac{d\mathbf{x}}{dt} = -\mu(\mathbf{x} - \mathbf{x}_0^*)^T (F(\mathbf{x}) - F(\mathbf{x}_0^*)) \leq 0.$$

Moreover, we know that

$$(F(\mathbf{x}) - F(\mathbf{x}_0^*))^T (\mathbf{x} - \mathbf{x}_0^*) = 0$$

if only and if $\mathbf{x} = \mathbf{x}_0^*$, we have $dV/dt < 0$ for any $\mathbf{x} \neq \mathbf{x}_0^*$. It follows that the proposed neural network is globally asymptotically stable.

References

Alimorad, M., & Mahmood, K. (2011). Inverse filtering based method for estimation of noisy autoregressive signals. *Signal Processing*, *91*, 1659–1664.

Alliney, S., & Ruzinsky, S. A. (1994). An algorithm for the minimization of mixed L_1 and L_2 norms with application to Bayesian-estimation. *IEEE Transactions on Signal Processing*, *42*, 618–627.

Bobillet, W., Diversi, R., Grivel, E., et al. (2007). Speech enhancement combining optimal smoothing and errors-in-variables identification of noisy AR processes. *IEEE Transaction on Signal Processing*, *55*, 5564–5578.

Boll, S. F. (1979). Suppression of acoustic noise in speech using spectral subtraction. *IEEE Transations on Acoustics, Speech and Signal processing*, *27*, 113–120.

Boyd, S., & Vandenberghe, L. (2006). *Convex optimization*. Cambridge University Press.

Christmas, J., & Everson, R. (2011). Robust autoregression: Student-t innovations using variational bayes. *IEEE Transactions on Signal Processing*, *59*, 48–57.

Cichocki, A., & Amari, S. (2002). *Adaptive blind signal and image processing:learning algorithms and applications*. John Wiley and Sons, Ltd,.

Doclo, S., & Moonen, Marc (2002). GSVD-based optimal filtering for signal and multi-microphone speech enhancement. *IEEE Transactions on Signal Processing*, *50*, 2230–2244.

Doclo, S., & Moonen, M. (2005). On the output SNR of the speech-distortion weighted multichannel Wiener filter. *IEEE Signal Processing Letters*, *12*(12), 809–811.

Ephraim, Y., & Malah, D. (1984). Speech enhancement and using minimum mean square error short-time spectral amplitude estimator. *IEEE Transaction on acoustics Speech and Signal Processing*, *32*, 1109–1121.

Epharim, Y., & Van Trees, H.L. (1995a). A spectrally-based signal subspace approach for speech enhancement. In *IEEE international conference on acoustics, speech, and signal processing*. Vol. 1, (pp. 804 807).

Ephraim, Y., & Van Trees, H. L. (1995b). A signal subspace approach for speech Enhancement. *IEEE Transactions on Speech Audio Processing*, *3*, 251. 166.

Gabrea, M. (2005). An adaptive Kalman filter for the speech enhancement. In *IEEE workshop on application of signal processing to audio and acoustics*.

Gabrea, M., Grivel, E., & Najim, A. (1999). A single microphone Kalman filter-based noise canceller. *IEEE Signal Processing Letters*, *6*, 55–59.

Gannot, S., Burshtein, D., & Weinstein, E. (1998). Iterative and sequentiel Kalman filter-based speech enhancement algorithms. *IEEE Transactions on Speech and Audio Processing*, *6*, 373.

Gerkmann, T., & Hendriks, R. C. (2012). Unbiased MMSE-based noise power estimation with low complexity and low tracking delay. *IEEE Transactions on Acoustics, Speech and Signal Processing*, *20*, 1383. 1393.

Giannakis, G. B., & Mendel, J. M. (1990). Cumulant-based order determination of non-Gaussian ARMA models. *IEEE Transactions on Acoustics Speech And Signal Processing*, *38*, 1411–1423.

Gibson, J. D., Koo, Boneung, & Gray, Steven D. (1991). Filtering of colored noise for speech enhancement and coding. *IEEE Transactions on Signal Processing*, *39*, 1732–1742.

Goh, S. L., & Mandic, D. P. (2007). An augmented extended kalman filter algorithm for complex-valued recurrent neural networks. *Neural Computation*, *19*(4), 1–17.

Kay, S. M. (1993). *Fundamentals of statistical signal processing: Estimation theory*. Englewood Cliffs, NJ: Prentice-Hall.

Labarre, D., Grivel, E., Najim, M. H., & Todini, E. (2004). Two-Kalman filters based instrumental variable techniques for speech enhancement. *IEEE Transactions on Signal Processing*,.

Lee, K. Y., & Jung, S. (2000). Time-domain approach using multiple Kalman filters and em algorithm to speech enhancement with nonstationary noise. *IEEE Transactions on Speech and Audio Processing*, *8*, 282–291.

Loizou, Philipos C. (2007). *Speech enhancement theory and practice*. Canada: CRC Press.

Mandic, D. P., & Chambers, J. A. (2000a). On stability of relaxive systems described by polynomials with time-variant coefficients. *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, *47*(10), 1534–1537.

Mandic, D. P., & Chambers, J. A. (2000b). On robust stability of time-variant discrete-time nonlinear systems with bounded parameter perturbations. *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, *47*(2), 185–188.

Mandic, D. P., & Chambers, Jonathon A. (2001). Recurrent neural networks for prediction: Learning algorithms, architectures and stability. In *Adaptive and learning systems for signal processing, communications and control*. Wiley.

Ning, M., Bouchard, Martin, & Goubran, Rafik A. (2006). Speech enhancement using a masking threshold constrained Kalman filter and its heuristic implementations. *IEEE Transactions on Audio, Speech, and Language Processing*, *14*, 19. 32.

Park, S., & Choi, S. (2008). A constrained sequential EM algorithm for speech enhancement. *Neural Networks*, *21*, 1401–1409.

Roberto, D., & Guidorzi, R. (2007). Fast filtering of noisy autoregressive signals. *Signal Processing*, *27*, 2843–2849.

Simon, D. (2002). Training radial basis neural networks with the extended Kalman filter. *Neurocomputing*, *48*, 445–475.

Smidl, V., & Quinn, A. (2005). Mixture-based extension of the AR model and its recursive bayesian identification. *IEEE Transactions on Signal Processing*, *53*, 3530–3542.

Wang, G., Li, C., & Dong, L. (2010). Noise estimation using mean square cross prediction error for speech enhancement. *IEEE Transactions on Circuits and Systems I: Regular Papers*, *57*, 1489. 1499.

Wei, Q., & Xia, Y.S. (2013). A novel prewhitening subspace method for enhancing speech corrupted by colored noise. In *The 6th international congress on image and signal processing*.

Xia, Y. S. (2004). An extended projection neural network for constrained optimization. *Neural Computation*, *16*, 863–883.

Xia, Y.S. (2012). Speech enhancement using a novel noise constrained least square estimation. In *International conference on audio, language and image processing* (pp. 980–985).

Xia, Y. S., Gang, F., & Wang, J. (2008). A novel recurrent neural network for solving nonlinear optimization problems with inequality constraints. *IEEE Transactions on Neural Networks*, *19*, 1340–1353.

Xia, Y. S., & Kamel, M. S. (2008). A generalized least absolute deviation method for parameter estimation of autoregressive signals. *IEEE Transactions on Neural Networks*, *19*(1), 107–118.

Xia, Y. S., Kamel, M. S., & Henry, L. (2010). A fast algorithm for AR parameter estimation using a novel noise-constrained least squares method. *Neural Networks*, *33*, 396–405.

Xia, Y.S., & Wang, Penyu (2013). Speech enhancement in presence of colored noise using an improved least square estimation. In *Proceedings of the 3rd international conference on multimedia technology* (pp. 779–786).

Xia, Y.S., & Yu, Y. (2010). Speech enhancement using generalized least absolute deviation estimation. In *International conference on audio, language and image processing*.

Xia, Y.S., Lin, G., & Zheng, W.X. (2014). A Fast Discrete-time Learning Algorithm for Speech Enhancement Using Noise Constrained Parameter Estimation. In International Joint Conference on Neural Networks (pp. 3149–3154).