Kalman-Filtering Speech Enhancement Method Based on a Voiced-Unvoiced Speech Model

Zenton Goh, Kah-Chye Tan, *Senior Member, IEEE,* and B. T. G. Tan

Abstract—**In this work, we are concerned with optimal estimation of clean speech from its noisy version based on a speech model we propose. We first propose a (single) speech model which satisfactorily describes voiced and unvoiced speech and silence (i.e., pauses between speech utterances), and also allows for exploitation of the long term characteristics of noise. We then reformulate the model equations so as to facilitate subsequent application of the well-established Kalman filter for computing the optimal estimate of the clean speech in the minimum-mean-square-error sense. Since the standard algorithm for Kalman filtering involves multiplications of very large matrices and thus demands high computational cost, we devise a mathematically equivalent algorithm which is computationally much more efficient, by exploiting the sparsity of the matrices concerned. Next, we present the methods we use for estimating the model parameters and give a complete description of the enhancement process. Performance assessment based on spectrogram plots, objective measures and informal subjective listening tests all indicate that our method gives consistently good results. As far as signal-to-noise ratio is concerned, the improvements over existing methods can be as high as 4 dB.**

Index Terms—**Kalman filter, noise reduction, speech enhancement, speech model, speech processing.**

I. Introduction

SPEECH enhancement is a subject of both theoretical interest and practical importance. As a matter of fact, the presence of noise can result in appreciable degradation in the quality and intelligibility of recorded speech. Consequently, not only can it cause difficulty in interpreting and understanding the speech message, but it can also lead to unsatisfactory results on subjecting the noisy recorded speech to speech coding, speech recognition, or speaker identification.

One key to speech enhancement is satisfactorily modeling the human speech production process. Such modeling is difficult because speech signals are, in general, highly nonstationary. Classical speech enhancement methods [1]–[5] consider only models for short-time speech segment, and this overcomes the difficulty to a certain extent since speech signals are often quite stationary during a short period. However, such short-time models preclude exploitation of the long-term characteristics of noise. On the other hand, such long-term characteristics are naturally taken care of in the autoregressive (AR) approach [6]–[9] as speech signals are not modeled on a short-time basis but as a whole. The AR model is also known to be good for representing unvoiced speech. However, it is not quite appropriate for voiced speech since voiced speech is often quite periodic in nature. This has motivated us to look into speech models which can satisfactorily describe both voiced and unvoiced speech, and allow for exploitation of the long-term characteristics of noise.

In this work, we first propose a (single) speech model which can satisfactorily describe both voiced and unvoiced speech, as well as silence. Since it originates from AR modeling, the long-term characteristics of noise are naturally taken care of. Coupling the proposed speech model with the popular additive white-Gaussian-noise model, we are able to treat the enhancement problem quite realistically on a theoretical basis. Our objective is to obtain an optimal estimate of the clean speech in the minimum-mean-square-error (MMSE) sense, using the abovementioned models. To achieve this, we first reformulate the model equations so as to facilitate a subsequent application of the well-established Kalman filter for computing the desired estimate.

Since the standard algorithm for Kalman filtering involves multiplications of very large matrices and thus demands high computational cost, we improve the efficiency quite significantly by exploiting the sparsity of the matrices concerned. In this connection, we propose a mathematically equivalent algorithm whose computational cost (in terms of additions and multiplications) is only 1/900th of that of the standard algorithm. As the proposed algorithm requires *a priori* knowledge about the model parameters, we estimate them from the noisy speech using an iterative procedure which can be viewed as a form of expectation-maximization (EM).

Since the proposed algorithm is developed based on white-Gaussian-noise assumption, it is expected that its performance in colored noise will degrade. In this connection, we propose a practical and effective scheme for enhancing the applicability of the proposed algorithm in colored-noise scenarios. Moreover, the algorithm can be implemented online.

Performance assessment based on spectrogram plots, objective measures and informal subjective listening tests show that our enhancement method gives consistently good results. In particular, it gives better performance than the classical spectral subtraction method [1] and the AR-based method [6]–[7] (which is separately referred to as the Kalman-filtering method in [6] and the scalar-Kalman-filter method in [7]).

Manuscript received August 4, 1997; revised September 4, 1998.

Z. Goh and K.-C. Tan are with the Centre for Signal Processing, School of Electrical and Electronic Engineering, Nanyang Technological University, Nanyang Avenue, Singapore 639798, Republic of Singapore (e-mail: ezgoh@ntu.edu.sg).

B. T. G. Tan is with the Faculty of Science, National University of Singapore, Singapore 119260, Republic of Singapore.

Publisher Item Identifier S 1063-6676(99)06569-4.

II. Problem Statement

Consider the following model for noisy speech:

$$y(n) = s(n) + w(n) \tag{1}$$

where $n = 1, 2, \cdots$, and $y(n), s(n)$, and $w(n)$ denote discrete-time samples of noisy speech, clean speech and noise, respectively. Basically, our objective is to devise a method for obtaining an optimal (in the MMSE sense) estimate for each sample of the clean speech, based on the past and current samples, as well as future samples in a neighborhood of the noisy speech. In other words, we want to develop an algorithm for computing $\hat{s}(n)$, the MMSE estimate of $s(n)$, which can be expressed as

$$\hat{s}(n) \triangleq E(s(n)|y(n+\tau), \cdots, y(n), \cdots, y(1)) \tag{2}$$

for $n = 1, 2, \cdots$, where τ denotes the number of future samples of the noisy speech to be used, and $E(\cdot)$ denotes the expectation operator.

To achieve the objective, one has to first specify the statistical models for $w(n)$, the noise, and $s(n)$, the clean speech. In this connection, our model assumptions on $w(n)$ are the usual ones as follows:

1) it is generated by a stationary zero-mean white Gaussian process with variance σ_w^2;
2) it is independent of $s(n)$.

Our assumptions on $s(n)$ are based on the speech model that we shall propose in the next section.

III. The Proposed Speech Model

Before introducing the proposed speech model, it is worthwhile mentioning the speech model employed in [6] and [7], which has influenced our work. In [6] and [7], speech is assumed to be generated by AR process:

$$s(n) = \left(\sum_{k=1}^{q} a(n,k)s(n-k)\right) + e(n) \tag{3}$$

where $e(n)$, the excitation signal, is generated by a zero-mean white Gaussian process with variance $\sigma_{e(n)}^2, a(n,k)$'s are the adaptive filter coefficients, q is the filter order, and $s(n)$ is the output (clean) speech. Such an AR model is quite appropriate for describing unvoiced speech. However, it is not appropriate for describing voiced speech, since the excitation signal for voiced speech is often quite periodic and not as random as white Gaussian noise.

Our aim is to propose a single model to describe both voiced and unvoiced speech as well as the silence. Since both voiced and unvoiced speech are characterized by their excitation signals, our strategy is to appropriately model the excitation signals to accommodate both voiced and unvoiced speech. In this connection, we propose the following model for the excitation signals [in conjunction with the speech model given by (3)]:

$$e(n) = b(n,p_n)e(n-p_n) + d(n). \tag{4}$$

where $d(n)$ is generated by a zero-mean white Gaussian process with variance $\sigma_{d(n)}^2, p_n$ is the instantaneous pitch period and $b(n,p_n)$ is a measure of the instantaneous degree of voicing (or "periodicity"). For the next few paragraphs, we shall discuss how our proposed model caters for both voiced and unvoiced speech and silence as well.

To represent unvoiced speech which is by nature quite random, $b(n,p_n)$ is set to zero so that $e(n) = d(n)$ and thus the excitation signal $e(n)$ is white Gaussian noise (with variance $\sigma_{d(n)}^2$). (Note that p_n does not have any effect here and one can arbitrarily set it to any value, say 0.) Since the excitation signal for unvoiced speech can be well represented by white Gaussian noise, our proposed model is quite appropriate for unvoiced speech.

On the other hand, to represent voiced speech which is quite periodic, we set p_n to be the pitch period of the voiced speech, $b(n,p_n)$ close to one and $\sigma_{d(n)}^2$ close to zero, so that $e(n) \approx e(n-p_n)$ and thus the excitation signal is quite periodic. If the voiced speech is relatively less periodic, $b(n,p_n)$ will be assigned a value closer to zero, and $\sigma_{d(n)}^2$ will be assigned a value significantly larger than zero. Consequently, the periodicity will be weakened.

To represent silence, both $b(n,p_n)$ and $\sigma_{d(n)}^2$ are set to zero so that $e(n) = 0$, and thus the excitation signal is a zero signal. (Note that p_n does not have any effect and can be set to zero.) Consequently, the speech signal $s(n)$ will eventually decay to zero.

In summary, we have proposed a single speech model, as described by (3) and (4), which can appropriately describe the three different states of a speech signal, namely voiced speech, unvoiced speech, and silence. In Section VI, we shall present the methods for estimating the values of the parameters $a(n,k), p_n, b(n,p_n)$, and $\sigma_{d(n)}^2$.

IV. Optimal Estimation of Clean Speech

Considering the speech model given by (3) and (4) and the additive noise model given by (1), our objective is to obtain an optimal estimate (in the MMSE sense) of the clean speech as expressed in (2). Our approach is to utilize the well-established Kalman filter [10] to obtain our desired estimate. (Kalman filter is capable of providing the optimal estimate for a specific set of linear equations [10]–[11].) In this connection, we first reformulate the model equations (1), (3), and (4) to a specific form required by the Kalman filter.

A. Reformulation of Model Equations

First, it can be easily shown that (3) is equivalent to the following state-space equation:

$$\boldsymbol{s}_n = \boldsymbol{A}_n \boldsymbol{s}_{n-1} + \Gamma_1 e_n \tag{5}$$

where $\boldsymbol{s}_n = (s(n), \cdots, s(n-r+1))^T, r = \max(q, \tau+1), \Gamma_1$ is an $(r \times 1)$ vector given by $(1, 0, \cdots, 0)^T, e_n = e(n)$, and

$\boldsymbol{A}_n$ is an $(r \times r)$ matrix given by

$$\boldsymbol{A}_n = \begin{pmatrix} a(n,1) & \cdots & a(n,q) & 0 & \cdots & 0 \\ 1 & 0 & \cdots & 0 & \cdots & 0 \\ 0 & 1 & \ddots & & & \vdots \\ \vdots & \ddots & \ddots & \ddots & & \vdots \\ \vdots & & \ddots & \ddots & \ddots & \vdots \\ 0 & \cdots & \cdots & 0 & 1 & 0 \end{pmatrix}. \tag{6}$$

Second, to reformulate (4) into state-space form, we first note that (4) can be written as

$$e(n) = \sum_{l=1}^{p} b(n,l)e(n-l) + d(n) \tag{7}$$

where p is taken to be a constant equal to the maximum possible pitch period of human speech, and $b(n,l) = 0$ for all $l \neq p_n$, where p_n is the instantaneous pitch period. Subsequently, it can be easily shown that (7), as thus also (4), is equivalent to the following state-space equation:

$$\boldsymbol{e}_n = \boldsymbol{B}_n \boldsymbol{e}_{n-1} + \Gamma_2 \, d_n \tag{8}$$

where $\boldsymbol{e}_n = (e(n), \cdots, e(n-p+1))^T, \Gamma_2$ is a $(p \times 1)$ vector given by $(1, 0, \cdots, 0)^T, d_n = d(n)$, and $\boldsymbol{B}_n$ is a $(p \times p)$ matrix given by

$$\boldsymbol{B}_n = \begin{pmatrix} b(n,1) & b(n,2) & \cdots & \cdots & b(n,p) \\ 1 & 0 & \cdots & 0 & 0 \\ 0 & \ddots & \ddots & \vdots & \vdots \\ \vdots & \ddots & \ddots & 0 & \vdots \\ 0 & \cdots & 0 & 1 & 0 \end{pmatrix}. \tag{9}$$

Third, it can be shown that (5) and (8) can be combined into a single state-space equation as follows:

$$\boldsymbol{x}_{n+1} = \boldsymbol{F}_n \boldsymbol{x}_n + \Gamma_3 \, d_{n+1} \tag{10}$$

where $\boldsymbol{x}_n = \binom{\boldsymbol{s}_{n-1}}{\boldsymbol{e}_n}, \Gamma_3$ is an $((r+p) \times 1)$ vector given by $(\underbrace{0, \cdots, 0}_{r}, 1, 0, \cdots, 0)^T$, and $\boldsymbol{F}_n$ is an $(r+p) \times (r+p)$ matrix given by

$$\boldsymbol{F}_n = \begin{pmatrix} \boldsymbol{A}_n & \Gamma_1 \Gamma_2^T \\ \boldsymbol{O} & \boldsymbol{B}_{n+1} \end{pmatrix}. \tag{11}$$

Fourth, it can be easily shown that (1) is equivalent to the following state-space equation:

$$y(n) = \Gamma_4^T \boldsymbol{x}_{n+1} + w(n) \tag{12}$$

where Γ_4 is an $((r+p) \times 1)$ vector given by $(1, 0, \cdots, 0)^T$.

In summary, we have reformulated the model equations given by (1), (3), and (4) into the equivalent state-space equations given by (10) and (12).

B. Desired Optimal Estimate Obtained with the Kalman Filter

Now with the state-space equations given by (10) and (12) [which are equivalent to (1), (3), and (4)], we are ready to apply the Kalman filter [10], i.e., using the following algorithm for computing the output $\hat{s}(n)$, our desired optimal estimate (in the MMSE sense) of clean speech.

1) Initialization:

$$\begin{aligned} a(0,1) = \cdots = a(0,q) = s(0) = \cdots = s(1-q) = e(0) \\ = \cdots = e(-p) = y(0) = \sigma_{w(0)}^2 = 0, \end{aligned} \tag{13}$$

$$\boldsymbol{P}_0 = \boldsymbol{O}_{(r+p)\times(r+p)}, \quad \hat{\boldsymbol{x}}_0 = \boldsymbol{O}_{(r+p)\times 1}. \tag{14}$$

2) Recursion: For $n = 1, 2, \cdots,$

$$\boldsymbol{Q}_n = \boldsymbol{F}_{n-1}\boldsymbol{P}_{n-1}\boldsymbol{F}_{n-1}^T + \sigma_{d(n)}^2 \Gamma_3 \Gamma_3^T, \tag{15}$$

$$\boldsymbol{G}_n = \boldsymbol{Q}_n \Gamma_4 (\Gamma_4^T \boldsymbol{Q}_n \Gamma_4 + \sigma_{w(n-1)}^2)^{-1}, \tag{16}$$

$$\boldsymbol{P}_n = (\boldsymbol{I} - \boldsymbol{G}_n \Gamma_4^T)\boldsymbol{Q}_n, \tag{17}$$

$$\hat{\boldsymbol{x}}_n = \boldsymbol{F}_{n-1}\hat{\boldsymbol{x}}_{n-1} + \boldsymbol{G}_n(y(n-1) - \Gamma_4^T \boldsymbol{F}_{n-1}\hat{\boldsymbol{x}}_{n-1}). \tag{18}$$

3) Output: For $n = 1, 2, \cdots,$

$$\hat{s}(n) = (\underbrace{0, \cdots, 0}_{\tau}, 1, 0, \cdots 0)\hat{\boldsymbol{x}}_{n+\tau+1}. \tag{19}$$

For convenience, we shall hereafter call the above algorithm Algorithm 1.

V. Computation Reduction

We have obtained an algorithm, i.e., Algorithm 1 as discussed in the preceding section, for computing the optimal estimate (in the MMSE sense) of clean speech. Unfortunately, the computational cost is very high. One main reason for this is as follows. The maximum pitch period of human speech can be as high as 20 ms, which translates to 160 samples for an 8-kHz sampling rate. By definition of the constant p [see the sentence after (7)], it is then assigned the value of 160. Consequently, the sizes of the matrices $\boldsymbol{F}_n$ and $\boldsymbol{P}_n$ which appear in (15) will be at least 160 × 160. Therefore, Algorithm 1 involves many multiplications of very large matrices [see, e.g., (15)] and this leads to high computational overhead.

In this section, we shall reduce the computational cost of Algorithm 1 by exploiting the sparsity of the matrix $\boldsymbol{F}_n$. The reduced-computation algorithm, which we shall call Algorithm 2, will be shown to be equivalent to Algorithm 1 in Theorem 1. Although our presentation of Algorithm 2 will appear lengthy and complicated, the computational cost involved is only 1/900 of that required by Algorithm 1 (this will be elaborated later). At this juncture, it is worthwhile noting that in relevant studies such as [6]–[9], the computational cost required is relatively much less since the algorithms employed in these studies do not involve multiplications of very large matrices due to incorporation of (8).

We will first introduce some notations for ease of discussion and then state a crucial theorem. Subsequently, we shall state and compare the amount of computations required by Algorithms 1 and 2.

Notations: Let $\boldsymbol{H}$ be an $(m \times n)$ matrix and m_1, m_2, n_1, and n_2 be integers such that $1 \le m_1 \le m_2 \le m$ and $1 \le n_1 \le n_2 \le n$. Then $\boldsymbol{H}[m_1 : m_2, n_1 : n_2]$ shall denote the $(m_2 - m_1 + 1) \times (n_2 - n_1 + 1)$ submatrix of $\boldsymbol{H}$ formed by Row m_1 to Row m_2 and Column n_1 to Column n_2 of $\boldsymbol{H}$. Moreover, the notation $\boldsymbol{H}[m_1 : m_2, n_1 : n_2]$ will be simplified for some special cases:

1) if $m_1 = m_2$, then we simply write $\boldsymbol{H}[m_1 : m_1, n_1 : n_2]$ as $\boldsymbol{H}[m_1, n_1 : n_2]$, and $\boldsymbol{H}[m_1 : m_2, n_1 : n_1]$ as $\boldsymbol{H}[m_1 : m_2, n_1]$ for $n_1 = n_2$;
2) if $m_1 = 1$ and $m_2 = m$, then we simply write $\boldsymbol{H}[1 : m, n_1 : n_2]$ as $\boldsymbol{H}[:, n_1 : n_2]$, and $\boldsymbol{H}[m_1 : m_2, 1 : n]$ as $\boldsymbol{H}[m_1 : m_2, :]$ for $n_1 = 1$ and $n_2 = n$;
3) if $m_1 = m_2 = m = 1$, then $\boldsymbol{H}$ is a row vector and we simply write $\boldsymbol{H}[1 : 1, n_1 : n_2]$ as $\boldsymbol{H}[n_1 : n_2]$, and $\boldsymbol{H}[m_1 : m_2, 1 : 1]$ as $\boldsymbol{H}[m_1 : m_2]$ for $n_1 = n_2 = n = 1$;
4) if $m_1 = m_2 = m = 1$ and $n_1 = n_2$, then we simply write $\boldsymbol{H}[1 : 1, n_1 : n_1]$ as $\boldsymbol{H}[n_1]$, and $\boldsymbol{H}[m_1 : m_1, 1 : 1]$ as $\boldsymbol{H}[m_1]$ for $n_1 = n_2 = n = 1$ and $m_1 = m_2$.

Note that the $((r + p) \times (r + p))$ matrices $\boldsymbol{P}_{n-1}$ and $\boldsymbol{Q}_n$ appearing in (15) are symmetric and can be written as

$$\boldsymbol{P}_{n-1} = \begin{pmatrix} \boldsymbol{P}_{n-1,1} & \boldsymbol{P}_{n-1,2} \\ \boldsymbol{P}_{n-1,2}^T & \boldsymbol{P}_{n-1,3} \end{pmatrix}, \quad \boldsymbol{Q} = \begin{pmatrix} \boldsymbol{Q}_{n,1} & \boldsymbol{Q}_{n,2} \\ \boldsymbol{Q}_{n,2}^T & \boldsymbol{Q}_{n,3} \end{pmatrix} \tag{20}$$

where $\boldsymbol{P}_{n-1,1}$ and $\boldsymbol{Q}_{n,1}$ are symmetric $(r \times r)$ matrices, $\boldsymbol{P}_{n-1,2}$ and $\boldsymbol{Q}_{n,2}$ are $(r \times p)$ matrices, and $\boldsymbol{P}_{n-1,3}$ and $\boldsymbol{Q}_{n,3}$ are symmetric $(p \times p)$ matrices. We are now ready to state a crucial theorem.

Theorem 1: The following algorithm [stated in (21) to (37)] is equivalent to Algorithm 1 [stated in (13) to (19)].

1) Initialization:

$$\begin{aligned} a(0,1) = \cdots = a(0,q) = s(0) = \cdots = s(1-q) = e(0) \\ = \cdots = e(-p) = y(0) = \sigma_{w(0)}^2 = 0, \end{aligned} \tag{21}$$

$$\begin{aligned} \boldsymbol{P}_{0,1} &= \boldsymbol{O}_{r\times r}, \quad \boldsymbol{P}_{0,2} = \boldsymbol{O}_{r\times p}, \\ \boldsymbol{P}_{0,3} &= \boldsymbol{O}_{p\times p}, \quad \hat{\boldsymbol{x}}_0 = \boldsymbol{O}_{(r+p)\times 1}. \end{aligned} \tag{22}$$

2) Recursion: For $n = 1, 2, \cdots$,

$$\boldsymbol{a}_{n-1} = (a(n-1,1), \cdots, a(n-1,q))^T, \tag{23}$$

$$\boldsymbol{f}_{n-1} = \boldsymbol{P}_{n-1,1}[:, 1 : q]\boldsymbol{a}_{n-1} + \boldsymbol{P}_{n-1,2}[:, 1], \tag{24}$$

$$\boldsymbol{g}_n = \boldsymbol{P}_{n-1,3}[:, p_n]b(n, p_n), \tag{25}$$

$$\begin{aligned} q_{n,1} = {} & \boldsymbol{a}_{n-1}^T(\boldsymbol{f}_{n-1}[1 : q] + \boldsymbol{P}_{n-1,2}[1 : q, 1]) \\ & + \boldsymbol{P}_{n-1,3}[1, 1], \end{aligned} \tag{26}$$

$$\begin{aligned} q_{n,2} = {} & (\boldsymbol{a}_{n-1}^T\boldsymbol{P}_{n-1,2}[1 : q, p_n] \\ & + \boldsymbol{P}_{n-1,3}[1, p_n])b(n, p_n), \end{aligned} \tag{27}$$

$$q_{n,3} = \boldsymbol{g}_n[p_n]b(n, p_n) + \sigma_{d(n)}^2, \tag{28}$$

$$\begin{aligned} \boldsymbol{q}_{n,4} = {} & (\boldsymbol{a}_{n-1}^T\boldsymbol{P}_{n-1,2}[1 : q, 1 : p-1] \\ & + \boldsymbol{P}_{n-1,3}[1, 1 : p-1])^T, \end{aligned} \tag{29}$$

$$\boldsymbol{q}_{n,5} = \boldsymbol{P}_{n-1,2}[1 : r-1, p_n]b(n, p_n), \tag{30}$$

$$\boldsymbol{G}_n = \begin{pmatrix} q_{n,1} \\ \boldsymbol{f}_{n-1}[1 : r-1] \\ q_{n,2} \\ \boldsymbol{q}_{n,4} \end{pmatrix} (q_{n,1} + \sigma_w^2)^{-1}, \tag{31}$$

$$\begin{aligned} \boldsymbol{P}_{n,1} = {} & \begin{pmatrix} q_{n,1} & \boldsymbol{f}_{n-1}[1 : r-1]^T \\ \boldsymbol{f}_{n-1}[1 : r-1] & \boldsymbol{P}_{n-1,1}[1 : r-1, 1 : r-1] \end{pmatrix} \\ & - \boldsymbol{G}_n[1 : r](q_{n,1}, \boldsymbol{f}_{n-1}[1 : r-1]^T), \end{aligned} \tag{32}$$

$$\begin{aligned} \boldsymbol{P}_{n,2} = {} & \begin{pmatrix} q_{n,2} & \boldsymbol{q}_{n,4}^T \\ \boldsymbol{q}_{n,5} & \boldsymbol{P}_{n-1,2}[1 : r-1, 1 : p-1] \end{pmatrix} \\ & - \boldsymbol{G}_n[1 : r](q_{n,2}, \boldsymbol{q}_{n,4}^T), \end{aligned} \tag{33}$$

$$\begin{aligned} \boldsymbol{P}_{n,3} = {} & \begin{pmatrix} q_{n,3} & \boldsymbol{g}_n[1 : p-1]^T \\ \boldsymbol{g}_n[1 : p-1] & \boldsymbol{P}_{n-1,3}[1 : p-1, 1 : p-1] \end{pmatrix} \\ & - \boldsymbol{G}_n[r+1 : r+p](q_{n,2}, \boldsymbol{q}_{n,4}^T), \end{aligned} \tag{34}$$

$$\boldsymbol{h}_{n-1} = \boldsymbol{a}_{n-1}^T\hat{\boldsymbol{x}}_{n-1}[1 : q] + \hat{\boldsymbol{x}}_{n-1}[r+1], \tag{35}$$

$$\begin{aligned} \hat{\boldsymbol{x}}_n = {} & \begin{pmatrix} \boldsymbol{h}_{n-1} \\ \hat{\boldsymbol{x}}_{n-1}[1 : r-1] \\ \hat{\boldsymbol{x}}_{n-1}[r+p_n]b(n, p_n) \\ \hat{\boldsymbol{x}}_{n-1}[r+1 : r+p-1] \end{pmatrix} \\ & + (y(n-1) - \boldsymbol{h}_{n-1})\boldsymbol{G}_n. \end{aligned} \tag{36}$$

3) Output: For $n = 1, 2, \cdots$,

$$\hat{s}(n) = \hat{\boldsymbol{x}}_{n+\tau+1}[\tau + 1]. \tag{37}$$

Remarks:

1) For convenience, we shall call the algorithm stated in (21)–(37) Algorithm 2. Clearly, Algorithm 2 is also one that produces the output $\hat{s}(n)$, our desired optimal estimate (in the MMSE sense) of clean speech as expressed in (2), based on the model assumptions stated in (1), (3), and (4).
2) In addition to establishing Theorem 1 via mathematical proof, our extensive experiments show that the outputs of Algorithms 1 and 2 are numerically identical.

Proof: See Appendix A. ■

Amount of Computation Reduction: First, note that both Algorithms 1 and 2 involve only additions and multiplications (and 1 division), but not specific functions such as trigonometric or exponential. Therefore, we shall compare the computational costs of the two algorithms only in terms of the number of additions and the number of multiplications required for each iteration.

It can be verified that each iteration of Algorithm 1 as specified by (15)–(18) and each iteration of Algorithm 2 as specified by (23)–(36) require the following amount of computations:

$$Mult_1 = 3(r+p)^3 + 4(r+p)^2 + 4(r+p), \tag{38}$$

$$Add_1 = 3(r+p)^3 + (r+p)^2 + (r+p), \tag{39}$$

$$\begin{aligned} Mult_2 = {} & r^2 + p^2 + rp + (r+p+2)(q+1) \\ & + 2(r+p), \end{aligned} \tag{40}$$

$$Add_2 = r^2 + p^2 + rp + (r+p+3)(q+1) \tag{41}$$

where $Mult_k$ and Add_k denote, respectively, the number of multiplications and additions required by each iteration of Algorithm k for $k = 1$ and 2. At this juncture, it is worthwhile recalling that p is the maximum possible pitch period of human speech, r is the number of columns of the matrix $\boldsymbol{A}_n$ defined in (6), and q is the total number of filter coefficients used in the model given by (3).

Reasonable choices of the values for p, q, and r are $p = 160, q = 10$, and $r = 101$ (which are indeed adopted in our implementation of Algorithm 2, and the details will be discussed in the next section). For these values, $Mult_1 = 53,612,271, Add_1 = 53,407,125$, $Mult_2 = 55,376$, and $Add_2 = 54,865$, and thus $Mult_1/Mult_2 \approx 968$ and $Add_1/Add_2 \approx 973$. Clearly, the amount of computation reduction is quite significant, as both the number of multiplications and the number of additions are reduced by more than 900 times.

Note that the computational requirement for the AR-based method (which is separately referred to as the Kalman-filtering method in [6] and the scalar-Kalman-filter method in [7]) is about 3440 multiplications and 3110 additions, both about 1/16th those of Algorithm 2. In comparison, the computations are about 1/15 000th those of Algorithm 1, which is a primitive version of Algorithm 2 without computation reduction. Note that it is not entirely unexpected that the computational requirement of Algorithm 2 is higher than that of the AR-based method since Algorithm 2 is based on a more sophisticated speech model. However, as we shall see in Section VIII, Algorithm 2 will give appreciably better performance, in terms of quality improvement, than the AR-based method.

VI. Some Other Crucial Issues and Summary of Our Enhancement Method

In the preceding section, we have proposed an algorithm for computing the optimal estimate of the clean speech. The algorithm requires knowledge about the parameters of the additive noise model given by (1) and those of the speech model given by (3) and (4). For the additive noise model, the only parameter is σ_w^2, the variance of the stationary noise. A commonly accepted estimate of σ_w^2 is the variance of those segments of the noisy speech signals that contain only the noise. In this connection, one may use a voice activity detector [4] to identify the noise-only segments. However, for simplicity and consistency, we simply take the beginning 100 ms of the speech signals as the noise-only segment in this study (the results obtained with such a simple approach are quite reasonable). For the speech model, there are four time-varying and three constant parameters. The time-varying parameters are

1) $a(n,k)$'s, the adaptive filter coefficients;
2) p_n's, the instantaneous pitch periods;
3) $b(n,p_n)$'s, the instantaneous periodicities;
4) $\sigma_{d(n)}^2$'s, the (instantaneous) variances of the signal $d(n)$ appearing in (4).

The constant parameters are

1) τ, the number of future samples of the noisy speech to be used in the formulation of the MMSE estimate given in (2);
2) q, the total number of the filter coefficients $a(n,k)$'s for each n;
3) p, the maximum possible pitch period of human speech.

For the rest of the section, we first discuss how the time-varying parameters are estimated using an iterative procedure (note that the clean speech will also be estimated in the process), and then mention the choice of the constant parameters. Subsequently, we provide a summary of our enhancement method.

A. Estimation of Clean Speech and Time-Varying Parameters

For this purpose, we have identified a fairly effective procedure similar to those mentioned in [3] and [7]. The procedure, which can be considered as a version of the EM algorithm (as noted in [7]), involves alternately estimating the parameters based on the last version of the estimate for the clean speech and estimating the clean speech based on the last version of the estimates for the parameters, until a stage where the quality/intelligibility of the estimate of the clean speech has reached a desired level. The details are as follows.

For the first iteration of the procedure, the time-varying parameters $a(n,k), p_n, b(n,p_n)$, and $\sigma_{d(n)}^2$ are estimated based on $y(n)$, the noisy speech, in the following way. For each n, the estimates of $a(n,k)$'s for $k = 1, \cdots, q$, are obtained with the well-established Durbin–Levinson algorithm [12], using a "smoothed version" of $y(n)$ as input. (If they were to be obtained using the noisy speech $y(n)$ directly without "smoothing," the estimates for the $a(n,k)$'s, which are closely related to the spectral envelopes, would vary so drastically that undesirable "musical" noise will become apparent. By using a "smoothed version" of $y(n)$, we found that such "musical" noise will be significantly weakened.) Indeed, we first compute the short-time magnitude spectrum and phase spectrum of the segment in a neighborhood (32 ms) of the sample $y(n)$. Note that the spectra are obtained through first multiplying the segment by a 256-point Hamming window followed by performing fast Fourier transform (FFT) analysis. Moreover, such operations are carried out for each sample. Second, we compute the short-time magnitude spectra of four neighboring segments which overlap with the original segment by 12 or 24 ms (i.e., two of the neighboring segments are obtained by shifting the analysis window 8 ms back and forth into the past and into the future, and two others 20 ms into the past and into the future). Third, we generate a "smooth" magnitude spectrum by taking the minimum of the five magnitude spectra for each frequency bin. Before we proceed, it is worthwhile explaining the choice of the word "smooth." Indeed, we have observed through our experiments that taking minimum in each frequency bin could effectively reduce undesirable "spikes" which appear in the frequency domain, resulting in a relatively "smoother" spectrum. Fourth, we obtain a time-domain signal by taking the inverse FFT of the "smooth" magnitude spectrum combined with the phase spectrum obtained in the first step. Finally, we estimate the $a(n,k)$'s from the signal so obtained using the Durbin–Levinson algorithm. Note that the number of samples we use for calculating the $a(n,k)$'s is 256.

The other three parameters (i.e., $p_n, b(n,p_n)$ and $\sigma_{d(n)}^2$) are estimated as follows. First, each estimate of p_n, the instantaneous pitch period, is obtained using $R(t)$, the autocorrelation function of the speech segment in a neighborhood (32 ms) of the sample $y(n)$, with 40% center clipping [12]. Second, each $b(n,p_n)$, the instantaneous periodicity, is estimated using the ratio $R(p_n)/R(0)$. If the ratio is more than 0.5, the speech segment is considered periodic and $b(n,p_n)$ is set to $R(p_n)/R(0)$. Otherwise, $b(n,p_n)$ is set to zero. Third, each $\sigma^2_{d(n)}$, the (instantaneous) variance of $d(n)$, is estimated in the following manner. We first compute $d(n)$ based on (3) and (4), and then estimate each $\sigma^2_{d(n)}$ by the variance of the segment in a small 8 ms neighborhood centered at $d(n)$. Note that at the beginning (end) of the speech signal, such neighborhood is taken to be only 4 ms into the future (4 ms into the past) related to n.

With these estimates of the time-varying parameters, we use them as input parameters to the algorithm stated in (21)–(37). Note that the input signal to the algorithm is the noisy speech, and the output signal is the first version of the estimate for the clean speech, which shall be denoted as $\hat{s}_1(n)$. This completes the first iteration of the procedure.

For the second iteration, we first reestimate the four parameters (i.e., $a(n,k), p_n, b(n,p_n)$, and $\sigma^2_{d(n)}$) using the above methods, except that here $\hat{s}_0(n)$ is used, instead of $y(n)$, as input to those methods. These reestimated parameters are then used as input parameters to the algorithm stated in (21)–(37). Note that the input signal to the algorithm is also the noisy speech (and would always be the noisy speech), and the output signal is the second version of the estimate for the clean speech, which shall be denoted as $\hat{s}_2(n)$.

Subsequent iterations are similar, and we stop the iterative procedure when the quality/intelligibility of the latest version of the estimate of the clean speech has reached a desired level. This completes the description of the iterative procedure.

B. Choice of the Constant Parameters

First, we address τ, the number of future samples of the noisy speech to be used in the estimation process. On one hand, a large τ means more noisy speech samples are used and thus more information is exploited. Consequently, the estimation error may be expected to be relatively smaller. On the other hand, a very large value of τ would lead to very high computational cost. In this connection, $\tau = 100$ gives a good compromise for the scenarios under consideration.

For the value of q, which is the total number of the filter coefficients, our consideration is that there should be a large enough number of filter coefficients so that the spectral envelop of speech is adequately represented. Hence we choose q to be ten, which is a figure commonly accepted by the speech processing community. Note that r, the number of columns of the matrix $\boldsymbol{A}_n$ as given by (6), will then be equal to 101 since $r = \max(q, \tau + 1)$.

Since the pitch period of human speech rarely exceeds 20 ms (which translates to 160 samples for a sampling rate of 8 kHz), we set p to 160.

C. Summary of Our Enhancement Method

Now we shall present a summary of our enhancement method, for which a block diagram is shown in Fig. 1. Given a noisy speech sampled at 8 kHz, we first estimate the parameter of the additive noise model according to the method mentioned in the first paragraph of Section VI. Next, the constant parameters of the speech model are chosen based on the justifications provided in Section VI-B. Subsequently, we use the iterative procedure mentioned in Section VI-A to obtain estimates for the time-varying parameters of the speech model and estimates for the clean speech. Based on the experiments we conducted, we found that at the third or fourth iteration, the quality/intelligibility of the enhanced speech (i.e., the estimate of the clean speech) usually reaches an acceptable level.

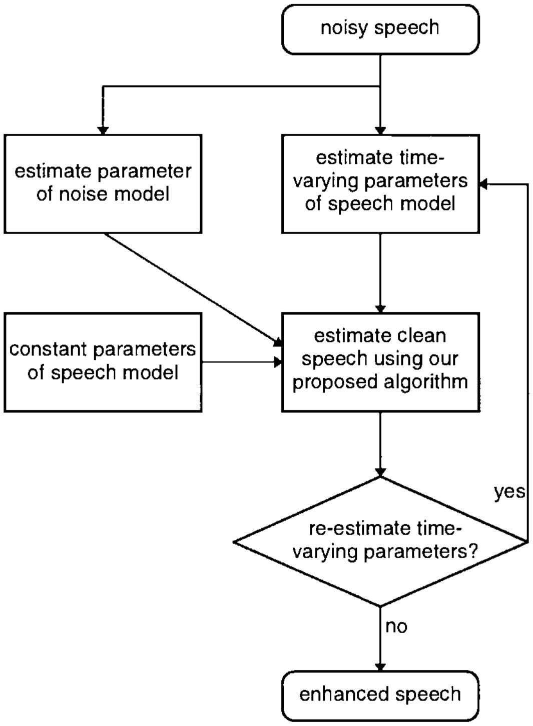

noisy speech
estimate parameter of noise model
estimate time-varying parameters of speech model
constant parameters of speech model
estimate clean speech using our proposed algorithm
re-estimate time-varying parameters?
yes
no
enhanced speech

Fig. 1. Block diagram of our proposed speech enhancement method (based on white-Gaussian-noise assumption).

D. Online Processing

One straightforward way to carry out the iterative procedure is to perform each iteration of parameter and speech estimation based on the entire speech utterance. However, we would like to highlight that the iterative procedure can also be carried out on *each speech sample*, and so online processing with some delay is possible. Indeed, processing of each speech sample requires up to $421m$ future samples (as have been worked out in Appendix C), where m is the number of iterations. Consequently, the time delay will be $52.625m$ ms (for the sampling rate of 8 kHz). At this juncture, it is worthwhile recalling that a typical value of m is three (see Section VI-C), resulting in a total of about 158 ms time delay.

Now we shall discuss our proposed procedure for online processing. Our approach is to describe how to obtain the first sample, the second sample, and so on, for the m-iteration, where m is a positive integer.

First, we shall describe how to obtain $\hat{s}_m(1)$, the first sample of the m-iteration enhanced speech. For convenience, let δ denote the delay per iteration, which is 421 samples, $\delta_1 = 133, \delta_2 = 101$, and $\delta_3 = 576$. In order to obtain $\hat{s}_m(1)$, one has to go through the first, second, $\cdots, (m-1)$th iterations involving the noisy speech samples $y(1), \cdots, y(1+m\delta)$. Indeed, with those $(1+m\delta)$ noisy speech samples, one can compute altogether $(1+(m-1)\delta)$ first-iteration enhanced speech samples, namely $\hat{s}_1(1), \cdots, \hat{s}_1(1+(m-1)\delta)$, using the algorithm stated in (21)–(37). Next, with $\hat{s}_1(1), \cdots, \hat{s}_1(1+(m-1)\delta)$, one can compute altogether $(1+(m-2)\delta)$ second-iteration enhanced speech samples, namely $\hat{s}_2(1), \cdots, \hat{s}_2(1+(m-2)\delta)$ (note that one would obtain δ samples fewer for a particular iteration as compared to the previous one). Eventually, one would obtain $\hat{s}_m(1)$ at the end of the mth iteration.

Now we shall discuss the parameters that have to be computed in the process of obtaining $\hat{s}_m(1)$. In fact, to obtain $\hat{s}_1(1), \cdots, \hat{s}_1(1+(m-1)\delta)$ from $y(1), \cdots, y(1+m\delta)$, one has to first compute the parameters $p_{n,1}, b_1(n, p_{n,1})$ and $a_1(n,k)$ for $n = 1, \cdots, (1+(m-1)\delta+\delta_1)$ and $k = 1, \cdots, q$, and $\sigma^2_{d(n),1}$ for $n = 1, \cdots, (1+(m-1)\delta+\delta_2)$, based on the parameter estimation methods outlined in Section VI-A. Note that we have appended the subscript 1 to all the parameters to indicate that these parameters would be obtained in the process of carrying out the steps required for the first-iteration. The values of these parameters will have to be recomputed in all subsequent iterations and we shall make similar indications by using appropriate subscripts. Subsequently, in a similar way, one has to first compute the parameters $p_{n,i}, b_i(n, p_{n,1})$, and $a_i(n,k)$ for $n = 1, \cdots, (1+(m-i)\delta+\delta_1)$ and $k = 1, \cdots, q$, and $\sigma^2_{d(n),i}$ for $n = 1, \cdots, (1+(m-i)\delta+\delta_2)$, in order to obtain $\hat{s}_i(1), \cdots, \hat{s}_i(1+(m-i)\delta)$ from $\hat{s}_{i-1}(1), \cdots, \hat{s}_{i-1}(1+(m-(i-1))\delta)$ for $i \geq 2$.

Next, we shall describe how to obtain $\hat{s}_m(2)$, the second sample of the m-iteration enhanced speech. In order to obtain $\hat{s}_m(2)$, one has to go through the first, second, $\cdots, (m-1)$th iterations involving noisy speech samples $y(j), \cdots, y(2+m\delta)$, where $j = \max(1, 2+m\delta(-\delta_3+1)$. Indeed, with the noisy speech samples, one can compute the first-iteration enhanced speech sample $\hat{s}_1(2+(m-1)\delta)$, using the algorithm stated in (23)–(37). Next, with $\hat{s}_1(j), \cdots, \hat{s}_1(2+(m-1)\delta)$, where $j = \max(1, 2+(m-1)\delta-\delta_3+1)$, one can compute the second-iteration enhanced speech sample $\hat{s}_2(2+(m-2)\delta)$ [note that the first-iteration enhanced speech samples $\hat{s}_1(j), \cdots, \hat{s}_1(1+(m-1)\delta)$ have been computed previously in the process of computing $\hat{s}_m(1)$]. Eventually, one would obtain $\hat{s}_m(2)$ at the end of the mth iteration.

Now we shall mention the parameters that have to be computed. Indeed, to obtain $\hat{s}_1(2+(m-1)\delta)$ from $y(j), \cdots, y(2+m\delta)$ where $j = \max(1, 2+m\delta-\delta_3+1)$, one has to first compute the parameters $p_{n,1}, b_1(n, p_{n,1})$, and $a_1(n,k)$ for $n = (2+(m-1)\delta+\delta_1)$ and $k = 1, \cdots, q$, and $\sigma^2_{d(n),1}$ for $n = (2+(m-1)\delta+\delta_2)$, based on the parameter estimation methods outlined in Section VI-A. Similarly, to obtain $\hat{s}_i(2+(m-i)\delta)$ from $\hat{s}_{i-1}(j), \cdots, \hat{s}_{i-1}(2+(m-(i-1))\delta)$ where $j = \max(1, 2+(m-(i-1))\delta-\delta_3+1)$, and for $i \geq 2$, one has to first compute the parameters $p_{n,i}, b_i(n, p_{n,1})$, and $a_i(n,k)$ for $n = (2+(m-i)\delta+\delta_1)$ and $k = 1, \cdots, q$, and $\sigma^2_{d(n),i}$ for $n = (2+(m-i)\delta+\delta_2)$.

The steps for obtaining $\hat{s}_m(3), \hat{s}_m(4), \cdots$, are similar to that for obtaining $\hat{s}_m(2)$, and thus we shall not elaborate it further and hereby end the discussion of on-line processing.

To achieve real-time processing, we need a computer which can perform about 0.3 million[1] floating-point operations per speech sample. This results in computational requirement of about 2.4 giga-floating-point operations per second (GFLOPS). In this connection, real-time processing is possible with today's high-end computers [13]–[15]. Moreover, there will be relatively much cheaper digital signal processors (DSP's) capable of yielding 3 GFLOPS performance in the near future (one example is Texas Instruments TMS320C67x DSP [16]).

VII. Dealing with Colored Noise

In the previous sections, we have developed a speech enhancement method based on white-Gaussian-noise assumption. When colored noise is encountered, it is expected that the enhancement method, as it is, will not perform as optimally as in the case of white Gaussian noise. Here, we shall propose a scheme that enhances the performance of our method in the presence of colored noise.

A. Overcoming Colored Noise

Our strategy is to "whiten" the noise before applying our enhancement method described in Section VI-C, and "undo the whitening" after the enhancement process. The details are as follows. To begin, let us consider the additive noise model given by (1) where $w(n)$ now denotes *colored noise* instead of white Gaussian noise. The objective is to estimate $s(n)$ from $y(n)$. We first carry out AR modeling of $w(n)$

$$w(n) = \sum_{i=1}^{u} c(n,i)w(n-i) + v(n) \tag{42}$$

where $v(n)$ is the output of a white Gaussian process, $c(n,i)$'s are the filter coefficients and u is the filter order. Note that such a modeling is commonly employed [7]–[9] and is appropriate so long as the filter order is large enough. We then filter $y(n)$ using the $c(n,i)$'s in the following way:

$$y_f(n) = y(n) - \sum_{i=1}^{u} c(n,i)y(n-i) \tag{43}$$

where $y_f(n)$ is the filtered signal. It can be easily shown that $y_f(n)$ can be expressed as follows:

$$y_f(n) = s_f(n) + v(n) \tag{44}$$

where $s_f(n) = s(n) - \Sigma_{i=1}^{u} c(n,i)s(n-i)$ denotes a filtered speech signal, and $v(n)$ is the white Gaussian noise. Note that $s_f(n)$ itself fits well into our proposed speech model presented in Section III.

Now the problem of estimating $s(n)$ from $y(n)$ is translated into the problem of estimating $s_f(n)$ from $y_f(n)$. The main difference is that we are now dealing with white Gaussian

[1] This figure originates from the fact that $(Mult_2 + Add_2)^*3 = (55,376 + 54865)^*3 \approx 0.3$ million, where $Mult_2$ and Add_2 are the number of multiplications and additions required by each iteration of Algorithm 2, and that three iterations are often enough to achieve enhanced speech with reasonably good quality.

noise instead of colored noise. Consequently, the enhancement method we have developed on the basis of white Gaussian noise can be used to obtain $\hat{s}_f(n)$, the MMSE estimate of $s_f(n)$.

With $\hat{s}_f(n)$, we can obtain an estimate of $s(n)$ by "inverse-filtering" using the following recipe:

$$\tilde{s}(n) = \sum_{i=1}^{u} c(n,i)\tilde{s}(n-i) + \hat{s}_f(n) \tag{45}$$

where $\tilde{s}(n)$ is the desired estimate, which is also the enhanced speech signal.

In a subsequent section (Section VIII-B), we shall demonstrate that the proposed scheme work well with real-life noise which is colored in nature.

B. Complete Colored-Noise Enhancement Procedure

Now we shall present the complete colored-noise enhancement procedure, for which a block diagram is shown in Fig. 2. Given a noisy speech (with colored noise), one has to first identify the frames which contain only noise. It is followed by a computation of the estimates of $c(n,i)$'s, the AR filter coefficients of the colored noise based on those frames using the Durbin–Levinson algorithm [12]. In this connection, the filter order we recommend is 16. On identification of segments containing only noise, one may use a voice activity detector [4]. However, for simplicity and consistency, we simply take the beginning 100 ms of the speech signals as the noise-only segment in this study (the results obtained with such a simple approach are quite reasonable). Next, the AR coefficients estimated are used to compute $y_f(n)$, the filtered noisy speech according to (43). Subsequently, we apply the enhancement method mentioned in Section VI-C to the filtered noisy speech and obtain $\hat{s}_f(n)$, the MMSE estimate of the filtered speech signal. Finally, the desired enhanced speech signal is obtained from $\hat{s}_f(n)$ according to (45).

VIII. Performance Assessments

We shall now address the performance assessment of our enhancement method. The test signals we use are 20 phonetically balanced speech sentences, of which ten are produced by male speakers and ten by female, taken from the TIMIT speech database [17]. The signals, which are originally sampled at 16 kHz, are downsampled to 8 kHz. Also, white Gaussian noise as well as automobile noise amounting to various values (namely −5, 0, 5, and 10 dB) of signal-to-noise ratio (SNR) are considered. Note that the automobile noise is recorded inside a slowly moving car (Hyundai Excel) with the air-conditioner being switched on. For performance assessment, we rely on objective measures, such as SNR and segmental SNR (SEGSNR), spectrogram plots, and informal subjective listening tests. The objective measures will first be computed based on the speech signals as a whole, and then based on only the voiced part of the speech signals (the motivation is that our proposed methods, unlike the existing methods [6]–[9], takes specific considerations of the voiced speech in addition to the unvoiced). Note that the voiced part of all the speech signals are marked out through manual observation and listening effort. Note also that we have removed the silent intervals in the speech signals before computing SEGSNR since they could drastically affect its value.

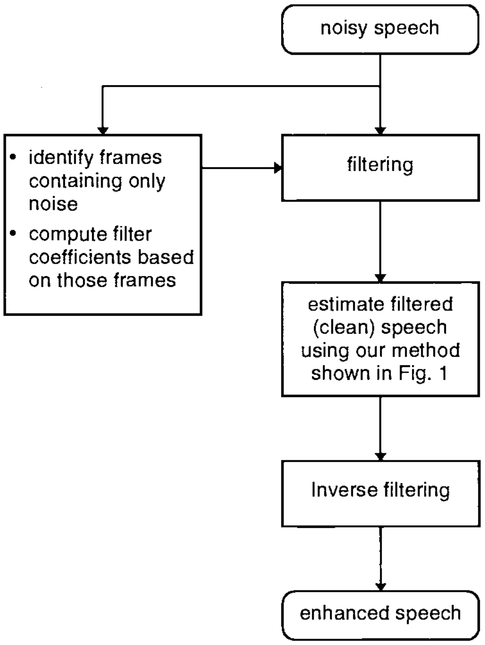

noisy speech
• identify frames containing only noise
• compute filter coefficients based on those frames
filtering
estimate filtered (clean) speech using our method shown in Fig. 1
Inverse filtering
enhanced speech

Fig. 2. Block diagram of the enhancement procedure we propose for tackling colored noise.

A. Case of White Gaussian Noise

Here we are concerned with white Gaussian noise and since our proposed model is developed on the basis of such noise, there is no necessity of noise whitening. But in the next subsection, whitening is necessary since it involves colored noise.

For the sake of predicting the best and worst performance of our proposed method, we first consider two extreme cases. One extreme case is that the time-varying parameters of the speech model are estimated using the clean speech, which we shall refer to as the "*ideal*" case[2]. Note that the methods for estimating the parameters are basically the same as those methods mentioned in Section VI-A, except that 1) the clean speech, instead of the noisy one, is used as input, and 2) the estimation of the $a(n,k)$'s are carried out without "smoothing" the clean speech. (In contrast, we have recommended working on the "smoothed version" of the noisy speech in Section VI-A). The other extreme case is that the parameters are estimated using the noisy speech with only one iteration, which we shall

[2] The "ideal" case serves as a possible "upperbound" for the performance of our proposed enhancement method. Indeed, our enhancement method contains a parameter-estimation submodule. We would not want to assume that the parameter-estimation methods we employ could yield the best possible estimates of the parameters concerned. On the other hand, we would expect better estimates if more effective methods could be devised and employed, which in turn leads to better enhancement results. While we do not attempt to devise such better methods here, we assume that if the methods we employ were applied on clean speech instead of its noisy version, the estimates would be better and this would result in higher SNR/SEGSNR figures. Consequently, we would like to take such figures as "upperbounds" for the performance of our proposed enhancement method.

TABLE I
SNR’s and SEGSNR’s for the Enhanced Speech Obtained with the AR-Based Method and Our Proposed Method for Two Extreme Cases (See Section VIII for Elaboration), Based on All 20 Sentences. Note That White Gaussian Noise Is Considered Here

	SNR (dB)				SEGSNR (dB)			
Noisy speech	–5	0	5	10	–11.72	–6.73	–1.71	3.29
Enhanced speech; AR-based method; baseline case	–0.27	4.01	8.19	12.45	–6.98	–2.70	1.48	5.68
Enhanced speech; our proposed method; baseline case	–0.10	4.38	8.76	13.09	–6.82	–2.39	1.96	6.25
Enhanced speech; AR-based method; ideal case	4.92	7.34	10.36	13.87	2.21	3.75	5.89	8.63
Enhanced speech; our proposed method; ideal case	7.51	9.89	12.60	15.73	3.51	5.31	7.48	10.18

TABLE II
Same as Table I Except That Only the Voiced Part of All 20 Sentences Is Considered. Note That White Gaussian Noise is Considered Here

	SNR (dB)				SEGSNR (dB)			
Noisy speech	–2.08	2.92	7.93	12.94	–4.88	0.14	5.14	10.15
Enhanced speech; AR-based method; baseline case	1.93	5.93	9.93	14.13	–0.78	3.24	7.24	11.41
Enhanced speech; our proposed method; baseline case	2.25	6.52	10.79	15.07	–0.49	3.82	8.07	12.35
Enhanced speech; AR-based method; ideal case	5.20	7.75	10.98	14.71	3.57	5.76	8.68	12.20
Enhanced speech; our proposed method; ideal case	8.16	10.72	13.59	16.90	6.09	8.48	11.26	14.46

call the “baseline” case. For comparison, we also consider the same two extreme cases for the AR-based method (which is referred to as the Kalman-filtering method in [6] and the scalar-Kalman-filter method in [7]). Table I tabulates the SNR’s and SEGSNR’s for the enhanced speech obtained with both methods for the two extreme cases, based on the entire speech signals (i.e., with voiced, unvoiced and silence) of all twenty sentences. Table II is similar to Table I except that the objective measures are computed based on only the voiced part of all twenty sentences. Both tables indicate that our proposed method is consistently superior to the AR-based method for the two extreme cases.

Second, we compare at various iterations the enhanced speech obtained using the AR-based method with that obtained using our proposed method. The comparison of our proposed method with the AR-based method can be done with Table III. The table tabulates the SNR’s and SEGSNR’s for the enhanced speech obtained with both methods at the first to sixth iterations, based on the entire speech signals of all 20 sentences. (Note that the first iteration is identical to the “baseline” case mentioned earlier.) Table IV is similar to Table III except that the objective measures are computed based on only the voiced part of all twenty sentences. Both tables indicate that our proposed method is consistently superior to the AR-based method. Moreover, it is evident from Table IV that for voiced speech, our proposed method gives much better performance than the AR-based method. On a separate note, observe that the best results are obtained with around three iterations, and SNR/SEGSNR decreases thereafter. Such phenomenon that SNR/SEGSNR decreases after a fixed number (in this case three) of iterations is also observed by Lim and Oppenheim [3] and Gibson *et al.* [7].

Third, we make a comparison among the enhanced speech obtained with spectral subtraction [1], the AR-based method at the third iteration, and our proposed method at the third iteration (we choose the third iteration since it usually gives best results). Note that the formula for spectral subtraction is given by

$$|\hat{S}_r[k]| = \begin{cases} (|Y_r[k]|^2 - |\hat{D}_r[k]|^2)^{1/2}, & \text{if } |Y_r[k]|^2 > |\hat{D}_r[k]|^2 \\ 0, & \text{otherwise} \end{cases} \tag{46}$$

where $|\hat{S}_r[k]|, |Y_r[k]|$, are $|\hat{D}_r[k]|$ the rth-frame magnitude spectra of enhanced speech, noisy speech and noise, respectively. Note also that the noise spectra is estimated using those segments of the noisy speech that contain only the

TABLE III
SNR's AND SEGSNR's FOR THE ENHANCED SPEECH OBTAINED WITH THE AR-BASED METHOD AND OUR PROPOSED METHOD AT THE FIRST TO SIXTH ITERATIONS, BASED ON ALL 20 SENTENCES. NOTE THAT WHITE GAUSSIAN NOISE IS CONSIDERED HERE

	Iteration no.	SNR (dB)				SEGSNR (dB)			
Noisy speech		–5	0	5	10	–11.72	–6.73	–1.71	3.29
Enhanced speech; AR-based method	1	–0.27	4.01	8.19	12.45	–6.98	–2.70	1.48	5.68
	2	2.95	6.06	9.44	13.23	–2.97	0.14	3.41	6.93
	3	***3.71***	***6.33***	***9.53***	***13.26***	–0.33	1.71	4.34	7.50
	4	3.31	5.96	9.24	13.07	0.56	***2.18***	***4.52***	***7.56***
	5	2.92	5.61	8.95	12.87	***0.78***	2.18	4.40	7.44
	6	2.63	5.34	8.71	12.68	0.66	1.96	4.18	7.21
Enhanced speech; our proposed method	1	–0.10	4.38	8.76	13.09	–6.82	–2.39	1.96	6.25
	2	3.55	7.07	10.62	14.29	–2.67	0.79	4.24	7.81
	3	***4.94***	***7.86***	***11.01***	***14.45***	0.12	2.60	5.39	8.50
	4	4.81	7.64	10.74	14.20	***1.05***	***3.09***	***5.58***	***8.51***
	5	4.42	7.23	10.36	13.89	0.83	2.70	5.19	8.15
	6	4.04	6.85	10.07	13.64	0.37	2.18	4.71	7.83

TABLE IV
SAME AS TABLE III EXCEPT THAT ONLY THE VOICED PART OF ALL 20 SENTENCES IS CONSIDERED. NOTE THAT WHITE GAUSSIAN NOISE IS CONSIDERED HERE

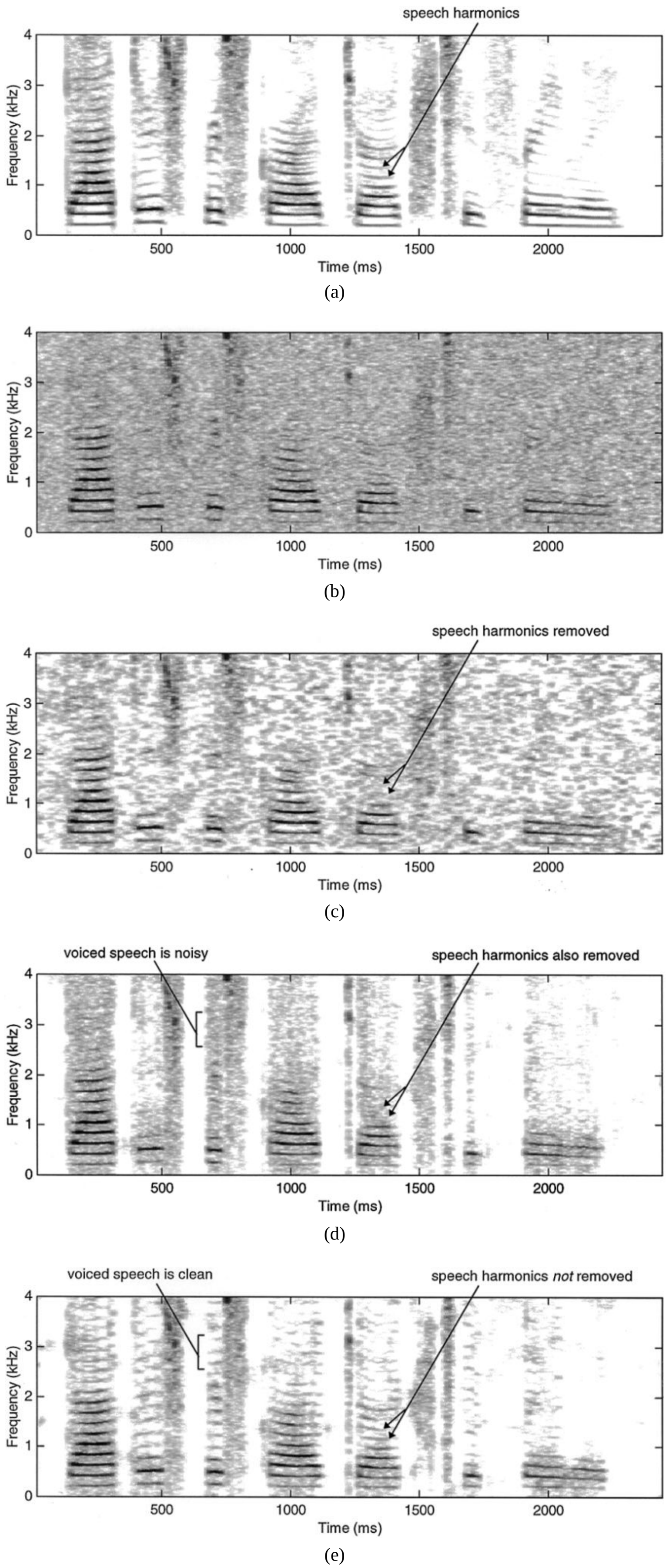
	Iteration no.	SNR (dB)				SEGSNR (dB)			
Noisy speech		–2.08	2.92	7.93	12.94	–4.88	0.14	5.14	10.15
Enhanced speech; AR-based method	1	1.93	5.93	9.93	14.13	–0.78	3.24	7.24	11.41
	2	3.92	***6.90***	***10.35***	***14.30***	1.75	4.58	***7.86***	***11.68***
	3	***4.04***	6.80	10.22	14.23	***2.39***	***4.68***	7.80	11.63
	4	3.54	6.40	9.95	14.08	2.01	4.33	7.53	11.48
	5	3.12	6.05	9.67	13.92	1.69	4.01	7.28	11.31
	6	2.81	5.76	9.42	13.74	1.45	3.74	7.05	11.14
Enhanced speech; our proposed method	1	2.25	6.52	10.79	15.07	–0.49	3.82	8.07	12.35
	2	4.80	8.23	11.86	15.67	2.42	5.79	9.34	13.09
	3	***5.50***	***8.64***	***12.04***	***15.71***	***3.48***	***6.40***	***9.70***	***13.26***
	4	5.27	8.41	11.80	15.54	3.34	6.27	9.59	13.18
	5	4.87	8.04	11.48	15.31	2.98	5.96	9.35	13.02
	6	4.48	7.64	11.19	15.08	2.69	5.64	9.11	12.84

noise. Table V, which tabulates the SNR's and SEGSNR's for the enhanced speech obtained with the three methods, shows that our proposed method is consistently superior to the other two, and the SNR improvements over the AR-based method, spectral subtraction and the original noisy speech are 1.2–1.5 dB, 1.7–4.1 dB, and 4.5–9.9 dB, respectively. (As shown in Table IV, for the voiced part alone, the SNR improvement upon the AR-based method can attain up to 2 dB.) Informal subjective listening tests which we have conducted also yield similar findings. In particular, undesirable "musical" noise can be heard in the enhanced speech obtained with spectral subtraction, but not that obtained with our proposed method. Moreover, the enhanced speech obtained with our proposed method demonstrates clarity and naturalness whereas that obtained with the AR-based method sounds somewhat distorted and occasionally muffled, especially for voiced speech.

Next, we compare the spectrograms of the enhanced speech obtained with the three methods. Fig. 3 shows the spectrograms of: the (original) clean speech, noisy speech, enhanced speech obtained with spectral subtraction, enhanced speech obtained with the AR-based method, and enhanced speech obtained with our proposed method. First, note that both Fig. 3(e) (our proposed method) and Fig. 3(d) (the AR-based method) appear much "cleaner" and more similar to Fig. 3(a) (the clean speech) than Fig. 3(c) (spectral subtraction). This indicates that our proposed method and the AR-based method are superior to spectral subtraction. Second, the voiced part of speech in Fig. 3(e) appears "cleaner" than that in Fig. 3(d). Third, some weak harmonics, which appear as parallel "stripes" in the clean speech [see Fig. 3(a)], are removed in the enhanced speech obtained with spectral subtraction and the AR-based method [see Fig. 3(c) and (d)]. On the other hand, many of these weak harmonics are still present in the enhanced speech obtained with our proposed method [see Fig. 3(e)].

B. Case of Colored Noise

Similar to the case of white Gaussian noise, we compare the various iterations of the enhanced speech obtained using



(a)

(b)

(c)

(d)

(e)

Fig. 3. Spectrograms of (a) (original) clean speech, (b) noisy speech, (c) enhanced speech obtained with spectral subtraction, (d) enhanced speech obtained with the AR-based method, and (e) enhanced speech obtained with our proposed method.

TABLE V
SNR's AND SEGSNR's FOR THE ENHANCED SPEECH OBTAINED WITH (a) SPECTRAL SUBTRACTION, (b) THE AR-BASED METHOD AT THE THIRD ITERATION, AND (c) OUR PROPOSED METHOD AT THE THIRD ITERATION, BASED ON ALL 20 SENTENCES. NOTE THAT WHITE GAUSSIAN NOISE IS CONSIDERED HERE

	SNR (dB)				SEGSNR (dB)			
Noisy speech	–5	0	5	10	–11.72	–6.73	–1.71	3.29
Enhanced speech; spectral subtraction	–0.86	3.76	8.27	12.72	–7.62	–3.02	1.47	5.88
Enhanced speech; AR-based method	3.71	6.33	9.53	13.26	–0.33	1.71	4.34	7.50
Enhanced speech; our proposed method	4.94	7.86	11.01	14.45	0.12	2.60	5.39	8.50

TABLE VI
SNR's AND SEGSNR's FOR THE ENHANCED SPEECH OBTAINED WITH THE AR-BASED METHOD AND OUR PROPOSED METHOD (BOTH WITH COLORED NOISE CONSIDERATION) AT THE FIRST TO SIXTH ITERATIONS AND FOR THE IDEAL CASE, BASED ON ALL 20 SENTENCES. NOTE THAT COLORED (AUTOMOBILE) NOISE IS CONSIDERED HERE

	Iteration no.	SNR (dB)				SEGSNR (dB)			
Noisy speech		–5	0	5	10	–11.67	–6.67	–1.67	3.34
Enhanced speech; AR-based method	1 (baseline)	–1.13	3.07	7.34	11.76	–7.78	–3.60	0.62	4.97
	2	1.12	4.46	8.20	12.30	–4.80	–1.56	1.95	5.83
	3	2.01	4.93	8.45	12.46	–2.70	–0.24	2.78	6.36
	4	2.28	5.07	8.53	12.49	–1.41	0.56	3.25	6.63
	5	2.40	5.17	8.58	12.51	–0.76	0.96	3.46	6.73
	6	2.48	5.22	8.60	12.52	–0.56	1.08	3.53	6.76
	Ideal	3.49	6.06	9.25	12.98	0.40	1.99	4.30	7.33
Enhanced speech; our proposed method	1 (baseline)	–0.29	4.11	8.33	12.53	–7.03	–2.65	1.58	5.79
	2	2.93	6.33	9.70	13.35	–3.39	0.02	3.39	6.99
	3	4.29	7.07	10.09	13.56	–0.90	1.67	4.46	7.68
	4	4.57	7.18	10.12	13.56	0.33	2.43	4.91	7.97
	5	4.47	7.05	10.00	13.48	0.63	2.56	4.99	8.01
	6	4.35	6.90	9.86	13.37	0.70	2.65	5.01	7.94
	Ideal	8.34	10.49	12.85	15.60	4.65	6.32	8.28	10.59

the AR-based method with that obtained using our proposed method, both with colored-noise consideration. The whitening preprocessing scheme we propose in Section VII will be applied to both our proposed method and the AR-based method. In addition, both the two extreme cases (the "baseline" case and the "ideal" case as mentioned in the preceding subsection) will be considered. Table VI tabulates the SNR's and SEGSNR's for the enhanced speech obtained with both methods at the 1st to 6th iterations and also for the "baseline" and "ideal" cases (note that the results of the 1st iteration are identical to those of the "baseline" case). Note that the results are computed using all twenty sentences mentioned in the first paragraph of Section VIII. We also compare our method with spectral subtraction. Indeed, Table VII is similar to Table V except here the colored noise, which is recorded in close proximity of an automobile with both engine and air-conditioner being turned on, is considered instead of white Gaussian noise. Both Tables VI and VII indicate that our proposed method is consistently superior to the AR-based method and spectral subtraction. Moreover, Table VII shows that our method has led to SNR improvements over the AR-based method, spectral subtraction and the original noisy speech amounting to 1.1–2.3 dB, 0–4.1 dB, and 3.6–9.3 dB, respectively.

We have also conducted informal subjective listening tests for the colored noise case. Indeed, we have invited ten listeners with speech processing experience to assess the quality of the enhanced speech obtained with our proposed method, and that with the AR-based method as well as that with spectral subtraction. All listeners found the enhanced speech obtained with spectral subtraction quite annoying (we think it is due to "musical" noise), but not those obtained with the AR-based method and our proposed method. As to the comparison between the AR-based method and our method, seven out of the ten listeners prefer the enhanced speech obtained with the latter. Our own assessment is that it is mainly due to the relative clarity of the voiced speech obtained with our method.

Before we end this section, we shall demonstrate that our proposed scheme for tackling colored noise is indeed effective. Table VIII shows the results of our method based on white-Gaussian-noise assumption and our method with colored-noise consideration (both at the third iteration). It is obvious that the latter perform significantly better.

TABLE VII
SNR'S AND SEGSNR'S FOR THE ENHANCED SPEECH OBTAINED WITH (a) SPECTRAL SUBTRACTION, (b) THE AR-BASED METHOD WITH COLORED-NOISE CONSIDERATION AT THE THIRD ITERATION, AND (c) OUR PROPOSED METHOD WITH COLORED-NOISE CONSIDERATION AT THE THIRD ITERATION, BASED ON ALL 20 SENTENCES. NOTE THAT COLORED (AUTOMOBILE) NOISE IS CONSIDERED HERE

	SNR (dB)				SEGSNR (dB)			
Noisy speech	–5	0	5	10	–11.67	–6.67	–1.67	3.34
Enhanced speech; spectral subtraction	0.20	4.83	9.36	13.53	–6.30	–1.72	2.76	7.23
Enhanced speech; AR-based method	2.01	4.93	8.45	12.46	–2.70	–0.24	2.78	6.36
Enhanced speech; our proposed method	4.29	7.07	10.09	13.56	–0.90	1.67	4.46	7.68

TABLE VIII
SNR'S AND SEGSNR'S FOR THE ENHANCED SPEECH OBTAINED WITH (a) OUR PROPOSED METHOD BASED ON WHITE-GAUSSIAN-NOISE ASSUMPTION AT THE THIRD ITERATION, AND (b) OUR PROPOSED METHOD WITH COLORED-NOISE CONSIDERATION AT THE THIRD ITERATION, BASED ON ALL 20 SENTENCES. NOTE THAT COLORED (AUTOMOBILE) NOISE IS CONSIDERED HERE

	SNR (dB)				SEGSNR (dB)			
Noisy speech	–5	0	5	10	–11.67	–6.67	–1.67	3.34
Enhanced speech; our proposed method based on white-Gaussian noise assumption	–3.32	0.99	5.40	10.03	–9.35	–5.10	–0.72	3.88
Enhanced speech; our proposed method with colored-noise consideration	4.29	7.07	10.09	13.56	–0.90	1.67	4.46	7.68

IX. CONCLUSION

We have developed an effective speech enhancement method based on a speech model that satisfactorily describes voiced and unvoiced speech and silence. We have also reformulated the model equations to facilitate the application of the well-established Kalman filter. In addition, we have addressed the computation issue and obtained an efficient algorithm for computing the optimal estimate of the clean speech in the MMSE sense. We went on to present the methods we use for estimating the model parameters and the scheme we propose for tackling colored noise, and conduct performance assessments of our enhancement method. Although the performance assessments conducted indicated that our enhancement method yielded consistently good performance, further scrutinization using additional real data as well as formal listening or intelligibility tests will be necessary before making a concrete conclusion.

For further research work, we suggest investigating two issues, namely 1) using more sophisticated methods for parameter estimation, and 2) further reducing the computational cost. We shall elaborate these issues in the next few paragraphs.

First, since estimation of model parameters from noisy (speech) signal is not the main issue of this paper, we have relied on some reasonable (existing) methods for parameter estimation. Better enhancement results could be expected if one fine-tunes the parameter estimation methods that we have employed for this work, or uses later/better parameter estimation methods such as those proposed by Sörquist *et al.* [8] and Gannot *et al.* [9]. On a separate note, the model parameter $a(n,k)$'s, which denote the adaptive filter coefficients, have a great influence on the quality/intelligibility of the enhanced speech. Therefore, methods capable of yielding good estimates of $a(n,k)$'s, such as that proposed by Hansen and Clements [18], can potentially lead to better enhancement results.

Second, further reduction in the computational cost of the proposed method can be achieved by exploiting the fact that human speech is often quite stationary in a reasonably short period. As a matter of fact, during such a short period, the model parameters are practically constant and so some computationally efficient methods specially developed for Kalman filter with constant parameters (interested readers please refer to [11] for details) can be considered. Furthermore, the model parameters need to be estimated only once (in contrast, they are estimated as many times as the number of samples during the short period under consideration). As a result, the computational cost is relatively lower.

APPENDIX A
PROOF OF THEOREM 1

Our strategy is to exploit the sparsity of some matrices that appear in Algorithm 1. Basically, the approach we adopt is to analyze each step of Algorithm 1, identify redundant computations/assignments (which originate from the sparsity of matrices), remove these redundancies and then show that the remaining (i.e., those that are not redundant) computations/assignments are exactly identical to those of Algorithm 2. Note that the proof of this theorem requires two lemmas and they are stated in Appendix B.

The first two steps [i.e., (13) and (14)] of Algorithm 1 are clearly equivalent to the first two steps [i.e., (21) and (22)] of Algorithm 2. Next, (15) involves computation of the matrix $\boldsymbol{Q}_n$. By Lemma 1 (see Appendix B), computing $\boldsymbol{Q}_n$ is equivalent to computing its submatrices $\boldsymbol{Q}_{n,1}, \boldsymbol{Q}_{n,2}$, and $\boldsymbol{Q}_{n,3}$. Since the formulae [(47)–(49)] for computing $\boldsymbol{Q}_{n,1}, \boldsymbol{Q}_{n,2}$, and $\boldsymbol{Q}_{n,3}$ involve the variables $\boldsymbol{a}_{n-1}, \boldsymbol{f}_{n-1}$, and $\boldsymbol{g}_n$, these variables have to be computed first, and it is done in (23)–(25). Now since $\boldsymbol{Q}_{n,1}[2:r,1] = \boldsymbol{f}_{n-1}[1:r-1]$ and $\boldsymbol{f}_{n-1}$ has already been computed in (24), it is redundant to compute $\boldsymbol{Q}_{n,1}[2:r,1]$. Similarly, it is redundant to compute $\boldsymbol{Q}_{n,1}[1,2:r]$. Next since $\boldsymbol{Q}_{n,1}[2:r,2:r] = \boldsymbol{P}_{n-1,1}[1:r-1,1:r-1]$, it is redundant to compute $\boldsymbol{Q}_{n,1}[2:r,2:r]$ since $\boldsymbol{P}_{n-1,1}[1:r-1,1:r-1]$ will have been computed in the previous (i.e., $(n-1)$th) iteration of the Algorithm 2. Consequently, it is not necessary to compute the whole matrix $\boldsymbol{Q}_{n,1}$—only $\boldsymbol{Q}_{n,1}[1,1]$ needs to be computed. Note that the computation of $\boldsymbol{Q}_{n,1}[1,1]$ is done in (26) where $q_{n,1} = \boldsymbol{Q}_{n,1}[1,1]$.

Next, since $\boldsymbol{Q}_{n,2}[2:r,2:p] = \boldsymbol{P}_{n-1,2}[1:r-1,1:p-1]$, it is redundant to compute $\boldsymbol{Q}_{n,2}[2:r,2:p]$ since $\boldsymbol{P}_{n-1,2}[1:r-1,1:p-1]$ will have been computed in the previous iteration of Algorithm 2. Consequently, it is not necessary to compute the whole matrix $\boldsymbol{Q}_{n,2}$—only $\boldsymbol{Q}_{n,2}[1,1], \boldsymbol{Q}_{n,2}[1,2:p]$, and $\boldsymbol{Q}_{n,2}[2:r,1]$ need to be computed. Note that the computation of these three terms are done in (27), (29), and (30), where $q_{n,2} = \boldsymbol{Q}_{n,2}[1,1], \boldsymbol{q}_{n,4} = \boldsymbol{Q}_{n,2}[1,2:p]^T$, and $\boldsymbol{q}_{n,5} = \boldsymbol{Q}_{n,2}[2:r,1]$.

For $\boldsymbol{Q}_{n,3}$, since $\boldsymbol{Q}_{n,3}[2:p,1] = \boldsymbol{g}_n[1:p-1]$ and $\boldsymbol{g}_n$ has already been computed in (25), it is redundant to compute $\boldsymbol{Q}_{n,3}[2:p,1]$. Similarly, it is redundant to compute $\boldsymbol{Q}_{n,3}[1,2:p]$. Next, since $\boldsymbol{Q}_{n,3}[2:p,2:p] = \boldsymbol{P}_{n-1,3}[1:p-1,1:p-1]$, it is redundant to compute $\boldsymbol{Q}_{n,3}[2:p,2:p]$ since $\boldsymbol{P}_{n-1,3}[1:p-1,1:p-1]$ will have been computed in the previous iteration of Algorithm 2. Consequently, it is not necessary to compute the whole matrix $\boldsymbol{Q}_{n,3}$—only $\boldsymbol{Q}_{n,3}[1,1]$ needs to be computed. Note that the computation of $\boldsymbol{Q}_{n,3}[1,1]$ is done in (28) where $q_{n,3} = \boldsymbol{Q}_{n,3}[1,1]$.

The next two steps [i.e., (16) and (17)] of Algorithm 1 involve the computation of $\boldsymbol{G}_n$ and $\boldsymbol{P}_n$. By Lemma 2 (see Appendix B), computing these 2 terms is equivalent to computing $\boldsymbol{G}_n, \boldsymbol{P}_{n,1}, \boldsymbol{P}_{n,2}$, and $\boldsymbol{P}_{n,3}$, using (50)–(53). Subsequently, it can be easily shown that (50)–(53) are equivalent to (31)–(34), by using the relationships among $q_{n,1}, q_{n,2}, q_{n,3}, \boldsymbol{q}_{n,4}, \boldsymbol{q}_{n,5}, \boldsymbol{Q}_{n,1}, \boldsymbol{Q}_{n,2}$, and $\boldsymbol{Q}_{n,3}$. Finally, it can be easily shown that the last 2 steps [i.e., (18) and (19)] of Algorithm 1 are equivalent to the last three steps [i.e., (35)–(37)] of Algorithm 2. This completes the proof. ■

APPENDIX B

We shall now state two lemmas, the proofs of which are quite straightforward and thus omitted.

Lemma 1: Equation (15) is equivalent to the three equations, (47)–(49), shown at the bottom of the page, where $\boldsymbol{a}_{n-1} = (a(n-1,1), \cdots, a(n-1,q))^T, \boldsymbol{f}_{n-1} = \boldsymbol{P}_{n-1,1}[:,1:q]\boldsymbol{a}_{n-1} + \boldsymbol{P}_{n-1,2}[:,1]$, and $\boldsymbol{g}_n = \boldsymbol{P}_{n-1,3}[:,p_n]b(n,p_n)$.

Lemma 2: Consider (16) and (17). First, (16) is equivalent to the following equation:

$$\boldsymbol{G}_n = \begin{pmatrix} \boldsymbol{Q}_{n,1}[:,1] \\ \boldsymbol{Q}_{n,2}[1,:]^T \end{pmatrix} (\boldsymbol{Q}_{n,1}[1,1] + \sigma_w^2)^{-1}. \tag{50}$$

Second, (17) is equivalent to the three equations that follow:

$$\boldsymbol{P}_{n,1} = \boldsymbol{Q}_{n,1} - \boldsymbol{G}_n[1:r]\boldsymbol{Q}_{n,1}[:,1]^T, \tag{51}$$

$$\boldsymbol{P}_{n,2} = \boldsymbol{Q}_{n,2} - \boldsymbol{G}_n[1:r]\boldsymbol{Q}_{n,2}[1,:], \tag{52}$$

$$\boldsymbol{P}_{n,3} = \boldsymbol{Q}_{n,3} - \boldsymbol{G}_n[r+1:r+p]\boldsymbol{Q}_{n,2}[1,:]. \tag{53}$$

APPENDIX C

The objective of this appendix is to derive the minimum number of future speech samples that one will make use of in order to obtain the enhanced speech with our method. Let us consider a particular speech sample $\hat{s}_1(n)$, which is the nth sample of the first-iteration enhanced speech. Next, let us assume that to compute $\hat{s}_1(n)$, the noisy speech samples involved are $y(1), \cdots, y(n+\delta)$, where δ is a positive integer we shall determine. Now recall that the computation of $\hat{s}_1(n)$ is done using (37), which requires the knowledge of the vector $\hat{\boldsymbol{x}}_{n+\tau+1}$. To compute $\hat{\boldsymbol{x}}_{n+\tau+1}$, we need to carry out computations based on (23)–(36). In this connection, it can be verified that such computations require the knowledge of $y(n+\tau), p_{(n+\tau+1)}, b(n+\tau+1, p_{(n+\tau+1)}), a(n+\tau,k)$ for $k = 1, \cdots, q$, and $\sigma^2_{d(n+\tau+1)}$. As a matter of fact, it is the computations of the parameters $p_{(n+\tau+1)}, b(n+\tau+1, p_{(n+\tau+1)}), a(n+\tau,k)$, and $\sigma^2_{d(n+\tau+1)}$ that involve future noisy speech samples.

(In order to follow the subsequent arguments, it is beneficial to be familiar with the methods for computing the parameters are described in Section VI-A.) For the first and second

$$\boldsymbol{Q}_{n,1} = \begin{pmatrix} \boldsymbol{a}_{n-1}^T(\boldsymbol{f}_{n-1}[1:q] + \boldsymbol{P}_{n-1,2}[1:q,1]) + \boldsymbol{P}_{n-1,3}[1,1] & \boldsymbol{f}_{n-1}[1:r-1]^T \\ \boldsymbol{f}_{n-1}[1:r-1] & \boldsymbol{P}_{n-1,1}[1:r-1,1:r-1] \end{pmatrix}, \tag{47}$$

$$\boldsymbol{Q}_{n,2} = \begin{pmatrix} (\boldsymbol{a}_{n-1}^T\boldsymbol{P}_{n-1,2}[1:q,p_n] + \boldsymbol{P}_{n-1,3}[1,p_n])b(n,p_n) & \boldsymbol{a}_{n-1}^T\boldsymbol{P}_{n-1,2}[1:q,1:p-1] + \boldsymbol{P}_{n-1,3}[1,1:p-1] \\ \boldsymbol{P}_{n-1,2}[1:r-1,p_n]b(n,p_n) & \boldsymbol{P}_{n-1,2}[1:r-1,1:p-1] \end{pmatrix} \tag{48}$$

$$\boldsymbol{Q}_{n,3} = \begin{pmatrix} \boldsymbol{g}_n[p_n]b(n,p_n) + \sigma^2_{d(n)} & \boldsymbol{g}_n(1:p-1]^T \\ \boldsymbol{g}_n[1:p-1] & \boldsymbol{P}_{n-1,3}[1:p-1,1:p-1] \end{pmatrix} \tag{49}$$

parameters $p_{(n+\tau+1)}$ and $b(n+\tau+1, p_{(n+\tau+1)})$, it is quite clear that the computations require future samples up to $y(n+\tau+1+128)$, assuming 8-kHz sampling rate. For the third parameter $a(n+\tau,k)$, it can be verified that the computation requires future samples up to $y(n+\tau+128+160)$ (it involves more future samples mainly because of the "smoothing process" as discussed in Section VI-A). For the fourth parameter $\sigma^2_{d(n+\tau+1)}$, it can be verified that the computation requires the knowledge about $y(n+\tau+1+32), p_{(n+\tau+1+32)}, b(n+\tau+1+32, p_{(n+\tau+1+32)})$, and $a(n+\tau+1+32,k)$ for $k=1,\cdots,q$, and it is the computation of $a(n+\tau+1+32,k)$ that requires a sample that is most into the future, i.e., $y(n+\tau+1+32+128+160)$.

Comparing the requirement of future samples for the above computations—computations of $p_{(n+\tau+1)}$ and $b(n+\tau+1, p_{(n+\tau+1)})$ involve up to $y(n+\tau+1+128)$; computation of $a(n+\tau,k)$ involves up to $y(n+\tau+128+160)$; computation of $\sigma^2_{d(n+\tau+1)}$ involves up to $y(n+\tau+1+32+128+160)$—it is clear that the maximum delay is incurred on computation of $\sigma^2_{d(n+\tau+1)}$ which amounts to 421 samples (recall that in Section VI-B, we have chosen $\tau = 100$). Therefore, $\tau = 421$, and this translates to 52.625 ms in time.

The above delay that we have worked out is for the first iteration. For the second iteration, it can be shown, using similar arguments, that it involves the first-iteration enhanced speech up to the same number of future samples. So the total delay for computing the first and second iterations is 52.625 × 2 ms. Based on the similar arguments, one can arrive at $52.625m$ ms total delay for m iterations.

Acknowledgment

We would like to thank the reviewers for their many insightful comments and useful suggestions.

References

[1] J. S. Lim and A. V. Oppenheim, "Enhancement and bandwidth compression of noisy speech," *Proc. IEEE*, vol. 67, pp. 1586–1604, Dec. 1979.

[2] S. F. Boll, "Suppression of acoustic noise in speech using spectral subtraction," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-27, pp. 113–120, Apr. 1979.

[3] J. S. Lim and A. V. Oppenheim, "All-pole modeling of degraded speech," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-26, pp. 197–210, Jun. 1978.

[4] R. J. McAulay and M. L. Malpass, "Speech enhancement using a soft-decision noise suppression filter," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-28, pp. 137–145, Apr. 1980.

[5] Y. Ephraim and D. Malah, "Speech enhancement using a minimum mean-square error short-time spectral amplitude estimator," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-32, pp. 1109–1121, Dec. 1984.

[6] K. K. Paliwal and A. Basu, "A speech enhancement method based on Kalman filtering," in *Proc. ICASSP'87*, pp. 177–180.

[7] J. D. Gibson, B. Koo, and S. D. Gray, "Filtering of colored noise for speech enhancement and coding," *IEEE Trans. Signal Processing*, vol. 39, pp. 1732–1742, Aug. 1991.

[8] P. Sörquist, P. Händel, and B. Ottersten, "Kalman filtering for low distortion speech enhancement in mobile communication," in *Proc. ICASSP'97*, pp. 1219–1222.

[9] S. Gannot, D. Burshtein, and E. Weinstein, "Iterative-batch and sequential algorithms for single microphone speech enhancement," *ICASSP'97*, pp. 1215–1218.

[10] R. E. Kalman, "A new approach to linear filtering and prediction problems," *J. Basic Eng., Trans. ASME*, Ser. D, vol. 82, pp. 35–45, Mar. 1960.

[11] B. D. O. Anderson and J. B. Moore, *Optimal Filtering*. Englewood Cliffs, NJ: Prentice-Hall, 1979.

[12] L. R. Rabiner and R. W. Schafer, *Digital Processing of Speech Signals*. Englewood Cliffs, NJ: Prentice-Hall, 1978.

[13] http://www.hpc.comp.nec.co.jp/sx-e/Products/sx-4b.html.

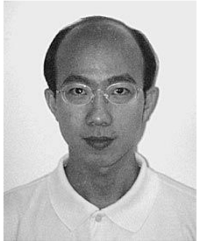
[14] http://www.digital.com/info/hpc/systems/systems.html.

[15] http://www.hp.com/hpwebcat/ch_qsumm.html.

[16] http://www.ti.com/sc/docs/dsps/products/c67x/index.htm.

[17] National Institute of Standards and Technology (NIST), *DARPA TIMIT Acoustics-Phonetic Continuous Speech Corpus,* NIST Speech Disc 1-1.1, Oct. 1990.

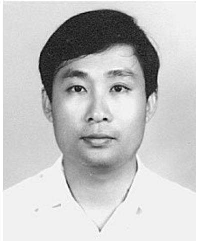
[18] J. H. L. Hansen and M. A. Clements, "Constrained iterative speech enhancement with application to speech recognition," *IEEE Trans. Signal Processing*, vol. 39, pp. 795–805, Apr. 1991.



Zenton Goh was born in Singapore. He received the B.Sc. degree (with first class honors) in mathematics from the National University of Singapore in 1992.

From 1993 to 1996, he was with DSO National Laboratories, Singapore. Since August 1996, he has been with the Center for Signal Processing (CSP), Nanyang Technological University, Singapore. He is currently a program manager for the voice based biometrics program of CSP. His research interests are in speech processing, especially speech enhancement, and array signal processing.

Mr. Goh is the recipient of the Singapore National Academy of Science Award in 1991, Tan Siak Kew Gold Medal in 1990 and 1991, Lijen Industrial Develepment Medal in 1992, Toh Chin Chye Book Prize in 1989, Sugar Industry of Singapore Book Prize in 1989, and Lim Soo Peng Book Prize in 1990.



Kah-Chye Tan (SM'98) was born in Singapore in 1961. He received the B.Sc (Hons) and Ph.D degrees in physics, both from the National University of Singapore, in 1985 and 1992, respectively.

From 1985 to 1996, he worked at the Defence Science Organization of Ministry of Defence, Singapore, as a Principal Analyst. In August 1996, he joined Center for Signal Processing, Nanyang Technological University, Singapore, as Research and Development Manager, and was appointed as Deputy Director in June 1998. His research interests are in array signal processing, speech/audio processing, image/video processing, and digital communications.

Dr. Tan is the recipient of the Defence Technology Prize (individual) in 1993 and the Institute of Physics Medal in 1985.

B. T. G. Tan received the B.Sc. (hons.) degree in physics from the University of Singapore in 1965, and the D.Phil. degree from Oxford University, Oxford, U.K., in 1968.

Since 1968, he has taught at the National University of Singapore, where he is now an Associate Professor in physics. His research interests include the microwave properties of semiconductors and dielectrics, surface physics, image processing, and digital sound synthesis.

Dr. Tan is a chartered engineer and a member of the IEE.