ADAPTIVE FREQUENCY ESTIMATION BASED ON NORMAL REALIZATIONS AND ITS APPLICATION IN SPEECH PROCESSING

J. Zhou and G. Li

School of EEE
Nanyang Technological University
Singapore 639798

ABSTRACT

In this paper, we investigate the problem of direct frequency estimation. A new adaptive algorithm is proposed based on the work in [12], where the constrained pole-zero notch filter was cascaded and each stage was implemented with the controllable realization. It is well known that the performance of adaptive filters is strongly related to how the filters are parametrized and implemented. The normal based realizations have some nice numerical properties. In the proposed algorithm, each subfilter is parametrized with its notching frequency and implemented with a normal realization. Compared with the one in [12], this structure makes the algorithm more robust such that the stability can be ensured automatically no matter the algorithm is implemented with or without infinite precision. Simulations show that the adaptive algorithm with the proposed structure also has better convergence behavior. Application of this algorithm to speech processing is also discussed.

1. INTRODUCTION

The problem of estimating frequencies from a desired signal of multiple sinusoids buried in an additive noise can be found from many practical situations. For example, the voiced speech signals show a high periodicity. To code such signals efficiently, one needs to detect the period (or equivalently, the fundamental frequency), called pitch in speech processing community. There exist two classes of processing. The first class, called off-line processing, is based spectral estimation techniques which are based on Discrete Fourier Transformation (DFT) and some MUltiple SIgnal Classification (MUSIC) based algorithms (see, e.g., [1]-[2]). These methods usually require a high cost of computation. The second class, called on-line processing, is based on adaptive notch filtering techniques. One of the popularly used models is the constrained notch filter (see, e.g., [3]-[12]), which has some superior properties such as better stability, fast convergence, and computation efficiency to unconstrained adaptive filters.

Classically, the adaptive notch filter is parametrized with the polynomial coefficients of its transfer function. These coefficients are function of the notching frequencies for which the notch filter has or nearly has a zero gain. These techniques, referred as indirect frequency estimation methods, require stability monitoring, that is the model stability has to be checked after each adaptation, which leads to a lot of additional computation. To overcome this weakness, a new adaptive algorithm for direct frequency estimation was developed in [12], where the constrained notch filter is cascaded into a series of second order subfilters and each subfilter is parametrized with its notching frequency.

It should be pointed out that in [12], each subfilter is implemented with the controllable realization which usually has poor numerical properties. It is well known that for a given transfer function, there exist a number of different realizations. These realizations, though equivalent, have different numerical properties such as error propagation and transfer function sensitivity. Normal realizations have a minimal pole sensitivity and are free of overflow oscillations. See, e.g., [7], [8]. The main objective in this paper is to develop an adaptive frequency estimation algorithm using the constrained notch filter which is implemented with a normal realization.

2. ADAPTIVE NOTCH FILTERS

Let $y(n)$ be a measurable signal, which consists of N sinusoids $s(n)$ with an additive broad-band noise $e(n)$:

$$\begin{aligned} y(n) &= \sum_{k=1}^{N} A_k cos(\theta_k^0 n - \phi_k) + e(n) \\ &\triangleq s(n) + e(n) \end{aligned} \quad (1)$$

where $s(n)$ and $e(n)$ are assumed to be independent, and $\{A_k \neq 0, \theta_k^0, \phi_k\}$ are the amplitude, (angular) fre-

quency and phase parameter of the sinusoids, respectively.

To estimate the frequencies $\{\theta_k^0\}$ with the only available signal $y(n)$, the following constrained adaptive notch filter was proposed in [12]:

$$H(z^{-1}) = \prod_{k=1}^{N} \frac{A_0(\theta_k, \beta z^{-1})}{A_0(\theta_k, \alpha z^{-1})} \triangleq \prod_{k=1}^{N} H_0(\theta_k, z^{-1}), \quad (2)$$

where $A_0(\theta_k, \gamma z^{-1}) \triangleq 1 - 2\gamma z^{-1} cos\theta_k + \gamma^2$, $\gamma = \alpha, \beta$ with α, β two positive constant close to 1. Typically, $\alpha = 0.9950$ and $\beta = 0.9999$. This notch filter has its poles at $\{\alpha e^{\pm j\theta_k}\}$, and zeros at $\{\beta e^{\pm j\theta_k}\}$. We note that $H(e^{j\theta_k}) \approx 0, \ \forall k$.

Denoting $\hat{\theta} \triangleq (\theta_1, \ \theta_2, \cdots, \ \theta_N)^{\mathcal{T}}$, we now define a cost function $V(\hat{\theta}, n)$ as

$$V(\hat{\theta}, n) \triangleq \frac{1}{2n} \sum_{j=1}^{n} \hat{e}^2(j). \quad (3)$$

By minimizing this cost function with respect to $\hat{\theta}$, one can obtain the estimate of the frequencies $\{\theta_k^0\}$. The algorithm we are to develop is based on a Gaussian-Newton type recursive prediction error based adaptive algorithm (RPE), that is:

$$\begin{aligned} K(n) &= \frac{P(n-1)\Psi_\theta(n-1)}{\lambda(n) + \Psi_\theta(n-1)P(n-1)\Psi_\theta(n-1)} \\ P(n) &= \lambda^{-1}(n)[P(n-1) - K(n)\Psi_\theta(n-1)P(n-1)] \\ \hat{\theta}(n) &= \hat{\theta}(n-1) + K(n)\hat{e}(n), \end{aligned} \quad (4)$$

where $\Psi_\theta(n-1) \triangleq -\frac{\partial \hat{e}(n)}{\partial \hat{\theta}}$ and

$$\begin{aligned} \alpha(n+1) &= \alpha_\infty - [\alpha_\infty - \alpha(n)]\alpha_0 \\ \lambda(n+1) &= \lambda_\infty - [\lambda_\infty - \lambda(n)]\lambda_0 \end{aligned} \quad (5)$$

with $\alpha_0 = 0.9900, \alpha_\infty = 0.9950$ and $\lambda(n)$ the forgetting factor. For detailed discussion on these parameters, we refer to [12].

To compute $\theta(n)$ with $P(n-1)$ and $\theta(n-1)$, one needs $\Psi_\theta(n-1) \triangleq -\frac{\partial \hat{e}(n)}{\partial \hat{\theta}}$. We note that

$$\hat{e}(n) = \prod_{k=1}^{N} H_0(\theta_k, z^{-1}) y(n).$$

It can be shown with some manipulations that

$$\begin{aligned} \frac{\partial \hat{e}(n)}{\partial \theta_i} &= 2[\frac{\beta z^{-1}}{A_0(\theta_i, \beta z^{-1})}\hat{e}(n) - \frac{\alpha z^{-1}}{A_0(\theta_i, \alpha z^{-1})}\hat{e}(n)] sin\theta_i \\ &\triangleq 2[e_{Fi}(\beta, n) - e_{Fi}(\alpha, n)] sin\theta_i, \end{aligned} \quad (6)$$

where $e_{Fi}(\gamma, n) = \frac{\gamma z^{-1}}{A_0(\theta_i, \gamma z^{-1})}\hat{e}(n), \gamma = \alpha, \ \beta$.

3. IMPLEMENTATIONS WITH NORMAL REALIZATIONS

In the adaptive algorithm by (4), one has to compute $\hat{e}(n)$ and $e_{Fi}(\gamma, t), \forall i$ for $\gamma = \alpha, \beta$. Now, let look at how to compute $\hat{e}(n)$.

Denote

$$x_i(n) \triangleq \prod_{k=1}^{i} H_0(\theta_k, z^{-1}) y(n). \quad (7)$$

$\hat{e}(n)$ can be evaluated the following iterative equations:

$$x_i(n) = H_0(\theta_i(n-1), z^{-1}) x_{i-1}(n) \quad (8)$$

with $x_0(n) = y(n)$ and $x_N(n) = \hat{e}(n)$.

One can compute $x_i(n) = H_0(\theta_i(n-1), z^{-1})x_{i-1}(n)$ using the input-output relationship or any direct form. It was argued in [12] that it is desired to use the state-space equations in order to have a better convergence behavior. In [12], the following state-space equations:

$$\begin{aligned} Z_i(n+1) &= A_c(n-1)Z_i(n) + B_c(n-1)x_{i-1}(n) \\ x_i(n) &= C_c(n-1)Z_i(n) + x_{i-1}(n) \end{aligned} \quad (9)$$

where

$$\begin{aligned} A_c(n) &= \begin{pmatrix} 2\alpha cos\theta_i(n) & -\alpha^2 \\ 1 & 0 \end{pmatrix}, B_c(n) = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \\ C_c(n) &= \begin{pmatrix} 2(\alpha - \beta)cos\theta_i(n) & \beta^2 - \alpha^2 \end{pmatrix}, \end{aligned} \quad (10)$$

which is the controllable realization of $H_0(\theta, z^{-1})$.

Remark 3.1: It is well known that the controllable realization, though simple, has some undesired numerical properties such as large sensitivity with respect to the parameter perturbation and high potential of zero-input overflow oscillation, which are very serious problems when the algorithm is implemented using a digital signal processor which is always of finite word length (FWL). Another important issue is stability. As argued in [12], (9) is always stable no matter what the variable θ_i takes with or without FWL errors. In the actual implementation of (9), the parameters $2\alpha cos\theta_i(n)$ and α^2 have to be truncated or rounded into their FWL version. Therefore, (9) may become unstable in that case.

The realizations of a given transfer function are not unique. Let

$$T = \begin{pmatrix} -\frac{cos\theta_i}{sin\theta_i} & 1 \\ -\frac{1}{\alpha sin\theta_i} & 0 \end{pmatrix}.$$

This is a similarity transformation under which the controllable realization can be transformed into the follow-

ing realization (A, B, C):

$$\begin{aligned} A &= T^{-1}A_cT = \begin{pmatrix} \alpha cos\theta_i & -\alpha sin\theta_i \\ \alpha sin\theta_i & \alpha cos\theta_i \end{pmatrix} \\ B &= T^{-1}B_c = \begin{pmatrix} 0 \\ 1 \end{pmatrix} \\ C &= C_cT = 2(\alpha-\beta)\begin{pmatrix} sin\theta_i & cos\theta_i \end{pmatrix}, \end{aligned} \quad (11)$$

which is a normal realization due to the fact that the matrix A is normal, i.e., $AA^{\mathcal{T}} = A^{\mathcal{T}}A$. Then we can use this realization to implement the adaptive algorithm:

$$\begin{aligned} Z_i(n+1) &= A(n)Z_i(n) + Bx_{i-1}(n) \\ x_i(n) &= C(n)Z_i(n) + x_{i-1}(n), \end{aligned} \quad (12)$$

where $A(n)$ and $C(n)$ are given by (11) with θ_i and α replaced by $\theta_i(n-1)$ and $\alpha(n)$.

Remark 3.2: The normal realizations have very nice numerical properties. In fact, they yield a minimal pole sensitivity [7] and are free of zero-input overflow oscillation [8]. Besides, (12) is always stable no matter the parameters in the A matrix are truncated into FWL numbers or not. All these nice properties are particularly very important for actual implementation.

Similarly, $e_{Fi}(\gamma, n) = \frac{\gamma z^{-1}}{A_0(\theta_i, \gamma z^{-1})}\hat{e}(n)$ can be implemented with a normal realization instead of the controllable realization. With some manipulations, it can be shown that

$$\begin{aligned} e_{Fi}(\gamma, n) &= \gamma \begin{pmatrix} -ctan\theta_i & 1 \end{pmatrix} W_i(\gamma, n) \\ W_i(\gamma, n+1) &= \gamma \begin{pmatrix} cos\theta_i & -sin\theta_i \\ sin\theta_i & cos\theta_i \end{pmatrix} W_i(\gamma, n) \\ &\quad + \begin{pmatrix} 0 \\ 1 \end{pmatrix} \hat{e}(n) \end{aligned} \quad (13)$$

for $\gamma = \alpha, \beta$, where $W_i(\gamma, n)$ is the corresponding state vector.

4. FUNDAMENTAL FREQUENCY ESTIMATION IN SPEECH PROCESSING

It is well known that speech signal is usually processed frame by frame with an interval of 10 - 30 ms. Voiced speech signals can be modeled with (1) and for most of the cases the speech signal shows highly periodicity for such a short time interval. The period, called pitch, is a very important parameter in speech encoder design and there exist several traditionally used algorithms to detect the pitch such as the average magnitude difference function and zero crossing measure (see, e.g., [10], [11]). The problem with such algorithms is that the obtained pitch is always an integer, therefore, there exists an estimation error constantly for most of the cases. Such an error may greatly affect the quality of the synthesized speech signal.

We can model a voiced speech signal using the following:

$$y(n) = \sum_{k=1}^{N} A_k cos(k\theta_0 n - \phi_k) + e(n), \quad (14)$$

where θ_0 is the fundamental (angular) frequency. In this case, we have one frequency, θ_0, to determine instead of N. Our proposed algorithm can be adapted for this situation easily. In fact, with the constraint

$$\hat{\theta}(n) = \begin{pmatrix} 1 & 2 & ... & k & ... & N \end{pmatrix}^{\mathcal{T}} \hat{\theta}_0(n), \quad (15)$$

one has

$$\begin{aligned} \Psi_{\theta_0}(n) &\triangleq -\frac{d\hat{e}(n)}{d\theta_0} = -\sum_{k=1}^{N} \frac{d\hat{e}(n)}{d\theta_k}\frac{d\theta_k}{d\theta_0} \\ &= \begin{pmatrix} 1 & 2 & ... & k & ... & N \end{pmatrix}^{\mathcal{T}} \hat{\Psi}_\theta \end{aligned} \quad (16)$$

where Ψ_θ, as defined before, is the derivative of the prediction error with respect to the frequency vector $\hat{\theta}$. Therefore, the corresponding algorithm to the fundamental frequency estimation is the same as (4) but with the vector $\hat{\theta}(n)$ replaced with the (scalar) fundamental frequency $\hat{\theta}_0(n)$ and hence $K(n)$ and $P(n)$ are scalar instead of matrix of order $N \times N$.

With the obtained fundamental frequency $\hat{\theta}_0$, the corresponding synthesized speech signal $\hat{y}(n)$ can be obtained with

$$\hat{y}(n) = \sum_{k=1}^{N} A_k cos(k\hat{\theta}_0 n - \phi_k), \quad (17)$$

where $\{A_k, \phi_k\}$ can be determined by minimizing

$$\sigma^2 \triangleq \frac{1}{L}\sum_{n=1}^{L}[y(n) - \hat{y}(n)]^2 \quad (18)$$

with respect to these variables, where L is the number of samples in one frame. In fact, denoting $A_k^c \triangleq A_k cos\phi_k$ and $A_k^s \triangleq A_k sin\phi_k$ for all k, one can see that σ^2 is a quadratic function in $\{A_k^c, A_k^s\}$ with $\hat{\theta}_0$ and $y(n)$ given. Therefore, σ^2 can be minimized with respect to $\{A_k^c, A_k^s\}$ easily. With the obtained optimal $\{A_k^c, A_k^s\}$, one can then convert them into $\{A_k, \phi_k\}$ steadily. In the next section, numerical examples will be given.

5. NUMERICAL EXAMPLES AND SIMULATIONS

In this section, we present two numerical examples and the corresponding simulations to examine the performance of our algorithm.

Example I: This example was used in [5], [12]. The signal $y(n)$ is given by

$$y(n) = 2sin(0.5n) + 2sin(1n) + 2sin(2n) + e(n),$$

where $e(n)$ is a white noise with zero-mean and unit variance. Clearly, the Signal-to-Noise Ratio (SNR) is 3dB for each sinusoid. We generate 30 different frames $y(n)$, each of them contains 500 samples. We tested the algorithm in [12], denoted as GL's algorithm, and our proposed one.

In order to have a fair comparison, we set the same initial conditions for the two algorithms. In the 30 trials, GL's algorithm converge to their true parameter vector 22 times after the 400 iterations, while our proposed algorithm, converges 26 times. Due to the limited space, simulations will be presented on the conference.

Example II: In this example, the signal is a frame of voiced speech signal with $L = 500$. This speech signal is sampled with $20kHz$. We estimate the pitch using the classical average magnitude difference function (AMDF) [11] and our proposed algorithm with $N = 16$.

With AMDF, the fundamental frequency is 0.0706. The corresponding error variance, as defined in (18) where $\hat{y}(n)$ is computed with (17) with $N = 16$, is 339.6915. We run the proposed adaptive algorithm with $\hat{\theta}_0(0) = 0.05$ and $P(0) = 20$, and obtain the fundamental frequency $\hat{\theta}_0(500) = 0.0708$. Then the corresponding synthesized speech signal is computed with (17) for $N = 16$. The error variance is 144.5735, much smaller than the one obtained with AMDF. In Fig. 1, the solid line is the original speech signal while the pointed line is the synthesized one computed with the pitch estimated with our proposed algorithm.

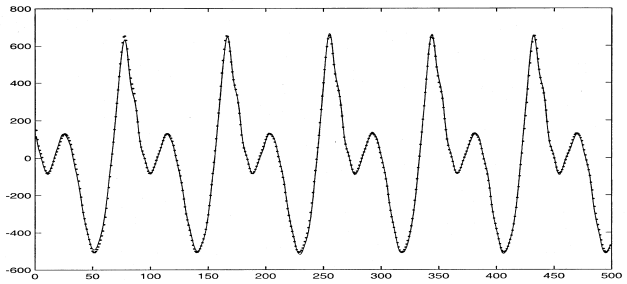

Fig. 1. The original speech signal and the synthesized one

6. REFERENCES

[1] S.M. Kay, Modern Spectral Estimation: Theory and Application, Prentice-Hall, Inc. Englewood Cliffs, N.J., 1988.

[2] J.T. Karhunen and J. Joutsensalo, "Sinusoidal Frequency Estimation by Signal Subspace Approximation," Signal Processing, Vol. 40, No. 12, pp. 2961-2972, 1992.

[3] D. V. Rao and S. Y. Kung, "Adaptive notch filtering for the retrieval of sinusoids in noise," IEEE Trans. on Acoust. Speech Signal Processing, Vol. ASSP-32, pp. 791-802, Aug. 1984.

[4] A. Nehorai, "A minimal parameter adaptive notch filter with constrained poles and zeros," IEEE Trans. on Acoust. Speech Signal Processing, Vol. ASSP-33, pp. 983-996, 1985.

[5] B. S. Chen, T. Y. Yang and B. H. Lin, "Adaptive notch filter by direct frequency estimation," Signal Processing, Vol. 27, pp. 161-176, 1992.

[6] T.S. Ng, "Some Aspects of an Adaptive Digital Notch Filter with Constrained Poles and Zeros," IEEE Trans. on Acoust. Speech Signal Processing, Vol. ASSP-35, pp. 158-161, 1987.

[7] M. Gevers and G. Li, Parametrizations in Control, Estimation and Filtering Problems: Accuracy Aspects, Springer Verlag, London, in Communications and Control Engineering Series, 1993.

[8] R.A. Roberts and C.T. Mullis, Digital Signal Processing, Addison-Wesley, 1987.

[9] L.R. Rabiner and R.W. Schafer, Digital Processing of Speech Signals, Prentice-Hall, Inc. Englewood Cliffs, N.J., 1978.

[10] L.R. Rabiner and R.W. Schafer, Digital Processing of Speech Signals, Prentice-Hall, Inc. Englewood Cliffs, N.J., 1978.

[11] J.R. Deller, J.G. Proakis, and J.H.L. Hansen, Discrete-Time Processing OF SPEECH SIGNALS, Prentice-Hall, Inc., 1993.

[12] G. Li, " A Stable and Efficient Adaptive Notch Filter for Direct Frequency Estimation", IEEE Trans. on Signal Processing, Vol. 45, no. 8, pp. 2001-2009, Aug. 1997.