IMPROVING OF MFCC/LPC USING DIFFERENCE EQUATION SYSTEMS

Gheorghe RADU

The Air Force Academy "Henri Coandă" of Braşov

Abstract: *In this paper is presented a new way for computing cepstral coefficients by using linear prediction, in a different manner than used into scientific papers. With this new approach we avoid truncation errors, at practical level, caused by approximation when we try to calculate MFCC coefficients. We can do this the same time like Fourier analyses but with use of a lower amount of memory because is not necessary to store any reference vector. The equations used by this new approach for calculation of cepstral coefficients (MFCC) can be easily used by a digital signals processor and can work together with Levinson-Durbin algorithm for calculation of linear prediction coefficients. The advantage of computing cepstral coefficients by linear prediction also comes from the fact that the linear prediction equations are differential equations. These equations can also be solved by any classical or operational method.*

Key words*: cepstral coefficients (MFCC), linear prediction coding (LPC).*
2000 Mathematics Subject Classification: *60G25, 60M20, 60G35*

1. PRELIMINARY PROCESSING OF VOICE SIGNALS

The sound recorded by a microphone is sent through an analogue-digital converter in order to obtain a signal that can be used for digital processing. After some preliminary processing phases signal *windows* (frames) result. In each frame is possible to describe everything that happens with a vector of *features*, (*n* numbers). Vectors quantification means to allocate a single label c_i of each frame instead of *n* numbers which result in a low amount of memory used. The labels c_i are in fact *cepstral coefficients*. Inside signal windows a stationary vocal signal [2] was found and it's possible to use Fourier analyses processing. Let $s(n)$ a voice signal and $s_m(n)$ a short time signal assigned to signal analysis window number *m*. The short time voice signal is:

$$s_m(n) = s(n).w_m(n) \tag{1}$$

where $w_m(n)$ is *window* function, which is zero everywhere except a narrow (in time) region.

Although the window function can take different values for different frames, usually the time window is the same for each frame:

$$w_m(n) = w_m(m-n) \tag{2}$$

Using both equation result fast Fourier transform assigned to number *m* windows of voice signal $s(n)$:

$$S_m(e^{j\omega}) = \sum_{n=-\infty}^{\infty} s_m(n).e^{-j\omega n} = \sum_{n=-\infty}^{\infty} s(n)w(m-n).e^{-j\omega n} \tag{3}$$

This poses all properties of Fourier transform known from signals theory. Assume that $s_m(n)$ is periodic signal with period of time T, related to window of analysis number *m*. T is also fundamental period ("pitch") for $s_m(n)$. In this case is known from signal theory (see by example [3]) his spectrum is described by a sum of Dirac pulses:

$$S_m(e^{j\omega}) = \sum_{k=-\infty}^{\infty} S_m(k).\delta\left(m - \frac{2\pi k}{T}\right) \tag{4}$$

Because Fourier transform of window signal $w(n)$ is:

$$W_m(e^{j\omega}) = \sum_{n=-\infty}^{\infty} w(n).e^{-j\omega n} \quad (5)$$

Result for $w(m-n)$, $W(e^{-j\omega}).e^{-j\omega n}$.

As a result with the help of property of convolution for a window number *m* where *m* is known, Fourier transform of $w(m-n).s(n)$ result from frequency convolution as:

$$S_m(e^{j\omega}) =$$

$$= \sum_{k=-\infty}^{\infty} S_m(k).W(e^{j(\omega-2nk/T)}).e^{j(\omega-2nk/T)m} \quad (6)$$

which is balanced sum of $W(e^{j\omega})$ translated on every harmonic with a rectangular window. It is well known that human perception of voice sound frequency does not follow a linear scale. These empirical findings leaded to the idea of defining a fundamental subjective frequency for pure tones (the sounds of the musical scale).

In this way for every real frequency *f* measured in Hz, one related subjective frequency was defined on a nonlinear scale named "Mel" after S. Mermelstein that created it. Relation between real frequency *f* and *subjective Mel frequency* $\hat{f}$ is:

$$\tilde{f} = M(f) = 1125.\ln\left(1 + f/700\right) \quad (7)$$

2. DETERMINATION OF CEPSTRAL COEFFICIENTS MFCC

One of the most prevalent perceptible parameterization is represented by Mel Frequency Cepstral Coefficients (MFCC) M_F^C. M_F^C Coefficients are derivate from Fourier transform like normal cepstral coefficients; the difference consists in nonlinear scale used. If, suppose the voice signal is continuous in time, split up with frequency f_e and noted with k discrete frequency related to real frequency kf_e one structure is necessary which made transformation:

$$\{kf_e\} \mapsto \{M(kf_e)\}, \quad (8)$$

where *M*(°) is transformation from equation (7).

Practically, this structure is usually a filters bed. If noted with $S(k)$ discrete Fourier transform of $s(n)$ signal results:

$$S(k) = \sum_{n=0}^{N-1} s(n)\, e^{-j2\pi nk/N} \quad (9)$$

Let frequencies f_l and f_h minimum and maximum (Hz) covered by filters bed and N, number of points where discrete Fourier transform of s_n is calculated. Central frequency of filters bed is calculated with relation [2]:

$$f(m) = \frac{N}{f_e}.M^{-1}\left(M(f_l) + m.\frac{M(f_h) - M(f_l)}{p+1}\right) \quad (10)$$

where inverse Mel transform can be obtained immediately from equation (7):

$$f = M^{-1}\left(\breve{f}\right) = 700.\left(\exp\left(\frac{\breve{f}}{1125}\right) - 1\right) \quad (11)$$

The next step in determination of M_F^C coefficients is calculation of energy logarithm from output of every filter from the filter bed:

$$E_m^N = ln\left(\sum_{k=0}^{N-1} |S(k)|^2 H_m(k)\right); \quad m = \overline{1,p} \quad (12)$$

Mel frequency cepstrum is inverse discrete cosine transform of output (in energy domain) of *m* filters in bed:

$$M_F^C(n) = \sum_{m=1}^{p} E_m^N.\cos\left(\frac{\pi n(m+1/2)}{p}\right); \quad n = \overline{0,p-1}; \quad (13)$$

Use of bed filters to achieve (11) transform equations is fast and elegant theoretically and practically. But in real time implementation it takes long time to obtain inverse or direct Fourier transform or inverse cosine transform even if rapid Fourier transform is used. Even more, in order to achieve rapid implementation of Fourier transform tables with predefined vectors are made in order to keep in memory the values of used exponentials witch lead in a large amount of memory used. In this situation is preferred other way to obtain M_F^C coefficients.

Starting from an idea showed in [6] to calculate M_F^C coefficients with the help of LPC coefficients we try to develop a new, more efficient algorithm.

In classical literature for determination of cepstral coefficients starting with linear prediction coefficients the following equation is used:

$$\ln\left(\frac{1}{1-\sum_{i=0}^{p} a_i z^{-i}}\right)=\sum_{n=0}^{\infty} c_n z^{-n},$$

where c_n is n^{th} real cepstral coefficient.

If the following transform is used:

$$z^{-1}=\frac{\tilde{z}^{-1}+\beta}{1+\beta\tilde{z}^{-1}} \tag{14}$$

where β is a real constant in sub unitary absolute value, the M_F^C coefficients can be determined from LPC coefficients using equation:

$$\sum_{n=0}^{L} c_n z^{-n}=\sum_{n=0}^{\infty} M_F^C(n)\tilde{z}^{-n}$$

where L is the number of cepstral coefficients determined from the linear prediction ones.

In order to keep errors of approximation used for calculation of $\tilde{c}_n$ coefficients at acceptable level is necessary L >> p, where p is prediction filter order. This constraint can not be used in practical applications. For this reason we propose a *different approach*:

1. It uses transform (14).
2. It determines $\tilde{a}_i$ coefficients of convert linear prediction filter with equation:

$$\frac{1}{1-\sum_{i=1}^{p} a_i z^{-i}}=\frac{1}{1-\sum_{i=1}^{\infty}\tilde{a}_i\tilde{z}^{-i}} \tag{15}$$

3. It determines M_F^C coefficients with equation:

$$\ln\left(\frac{1}{1-\sum_{i=1}^{\infty}\tilde{a}_i\tilde{z}^{-i}}\right)=\sum_{n=0}^{\infty} M_F^C(n)z^{-n}$$

In all these steps the most important thing is the presence of symbol "∞" on the right hand of equation (15) which allows us to obtain a prediction filter transformed by *infinite* order.

This practically translates, on equation (15) from which allows us to choose as big order as necessary, the same on both sides. It is possible to avoid, at practical level, errors caused by approximation in calculation of M_F^C coefficients and to achieve the same result like in Fourier transform way but in short time and with utilization of low memory. In order to obtain filtered Mel coefficients using the way presented above, the flowing equation must be used:

$$\tilde{a}_n^{(i)}\Big|_{i=\overline{-p,0}}=\begin{cases} a_{-i}+\beta\tilde{a}_0^{(i-1)} & ;\ n=0 \\ (1-\beta)^2\tilde{a}_0^{(i-1)}+\beta\tilde{a}_1^{(i-1)} & ;\ n=1 \\ \tilde{a}_{n-1}^{(i-1)}+\beta\left(\tilde{a}_n^{(i-1)}-\tilde{a}_{n-1}^{(i)}\right) & ;\ n=\overline{2,p} \end{cases};$$

$$\tilde{1}=\frac{1}{\tilde{a}_0^{(0)}};\ \tilde{a}_n=\frac{\tilde{a}_n^{(0)}}{\tilde{a}_0^{(0)}};\ n=\overline{1,p}$$

$$M_F^C(n)=\begin{cases} \ln\left(\tilde{1}\right) & ;\ n=0 \\ -\tilde{a}_n-\sum_{j=1}^{n-1}\frac{j}{n}M_F^C(j)\tilde{a}_{n-j} & ;\ n=\overline{1,p} \end{cases} \tag{16}$$

These equations can be easy used by a digital signal processor and can work together with Durbin-Levinson algorithm for determination of linear prediction coefficients. The M_F^C coefficients resulted from equation (16) are different than coefficients resulted from equation (13). It is well known that the spectrum for multiple processes use in practice application *real* cepstral coefficients. Below is presented a way for real cepstral coefficients calculation using LPC linear prediction coefficients. Real cepstral coefficients are defined by an equation like:

$$C_s(\omega)=\ln|S(f)|=\ln|V(f)U(f)|,$$

where $S(f)$ is Fourier transform at f frequency and $U(f)$ and $V(f)$ act at upper and lower level of frequency domain ("quefrency").

For calculation of real cepstral coefficients the following steps are taken:

1. First of all the following equation is used:

$$\ln\left[\frac{1}{1-\sum_{k=1}^{p} a_k z^{-k}}\right]=\sum_{n=1}^{\infty} c_n z^{-n} \tag{17}$$

where c_n is order n real cepstrum coefficient.

2. In order to achieve relation of recurrence between cepstral coefficients and prediction coefficients equation (17) is differentiation with respect of $t=z^{-1}$ and it results:

$$\sum_{k=1}^{p} a_k k t^{k-1} = \sum_{k=1}^{\infty} c_k k t^{k-1} \left(1 - \sum_{k=1}^{p} a_k t^k\right)$$

After identification of t^{n-1} coefficients resulted from above equation the following relation of recurrence is resulting:

$$n a_n = n c_n + n \sum_{k=1}^{n-1} a_k c_{n-k} - \sum_{k=1}^{n-1} k a_k c_{n-k}$$

From result the following difference equations:

$$\begin{cases} c_1 = a_1 \\ c_n = \sum_{k=1}^{n-1} \left(1 - \frac{k}{n}\right) a_k c_{n-k} + a_n \end{cases} \quad (18)$$

where c_n are cepstral coefficients and a_n are linear prediction coefficients. A new way for calculation will be presented in the next chapter. Real cepstral coefficients can be conversed into M_F^C coefficients with (7) equation.

3. LINEAR PREDICTION ANALYSES

3.1. SETTING THE PROBLEM

One of the most popular ways for voice signal analyses is based on linear predictive encoding well known LPC (Linear Predictive Coding).

Into pure acoustic theory about producing voice signal [2] the voice spectrum can be approximated by a filter with infinite impulse response (IIR), only poles, with a big enough number of poles, with transfer function $H(z)$:

$$H(z) = \frac{S(z)}{GU(z)} = \frac{1}{1 - \sum_{k=1}^{p} a_k z^{-k}} \quad (19)$$

or

$$S(z) = \sum_{k=1}^{p} a_k S(z) z^{-k} + GU(z) \quad (20)$$

where p is the order of LPC analyses and a_k are coefficients of linear prediction filter. Reverse filter has the transfer function $A(z)$.

$$A(z) = \frac{GU(z)}{S(z)} = 1 - \sum_{k=1}^{p} a_k z^{-k} \quad (21)$$

By applying inverse "z" transform in equation (20), we obtain the difference equation:

$$s(n) = \sum_{k=1}^{p} a_k s(n-k) + e(n) \quad (22)$$

where $e(n) = Gu(n)$ is error compensation.

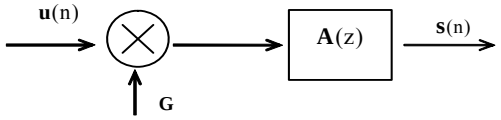

Fig. 1 LP model of speech

In Figure 2 it is shown a model of speech related to LPC analyses. Speech signal $u(n)$ together with G signal is applied at input of digital filter. Voice generator parameters are classified by taking into account the intensity and duration of speaking sound, gain parameters and prediction coefficients $\{a_k\}$.

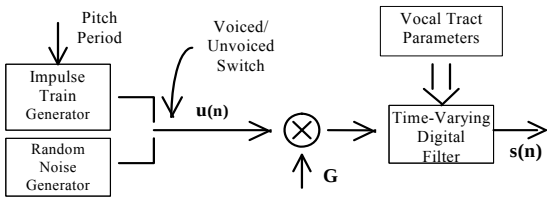

Fig. 2 Voice synthesis using an LPC model

All that parameters are time varying. The equations of LPC analyses will be presented in next chapter.

3.2. EQUATIONS OF LPC ANALYSES

Linear predictive coding method is based on prediction of current signal using a linear combination of p earlier parts of signal; result a signal predicted for time n labeled $\tilde{s}(n)$:

$$\tilde{s}(n) = \sum_{k=1}^{n} a_k s(n-k) \quad (23)$$

The prediction error e(n) in this case is equal to Gu(n):

$$e(n) = s(n) - \tilde{s}(n) = s(n) - \sum_{k=1}^{n} a_k s(n-k) \quad (24)$$

It is possible to use Fourier analyses to estimate $(a_k)_{k=\overline{1,p}}$ coefficients from voice signal fragment. Let $s_m(n)$ a voice signal fragment nearby m specimen: $s_m(n) = s(m+n)$.

This short time error of prediction related to that specimen is: $e_m(n) = e(m+n)$.

The medium root square error of short time prediction is:

$$E_m = \sum_n e_m^2(n) \tag{25}$$

From (23) and (24) result:

$$E_m = \sum_n \left[s_m(n) - \sum_{k=1}^{p} a_k s_m(n-k) \right]^2$$

$$= \sum_n s_m^2(n) - 2\sum_{k=1}^{p} a_k \sum_n s_m(n) s_m(n-k) +$$

$$+ \sum_n \left[\sum_{k=1}^{p} a_k s_m(n-k) \right] \tag{26}$$

Because probability distribution of $(a_k)_{k=\overline{1,p}}$ coefficients is unknown, for estimation of those coefficients is used minimizing of root mean square errors (25). So, for a signal $s_m(n)$ the LPC coefficients are estimated like numbers which minimizes prediction error E_m. If partial derivative of error E_m with respect to a_k is cancelled:

$$\frac{\partial}{\partial} \frac{E_m}{a_k} = 0; \quad k = \overline{1, p} \tag{27}$$

result scalar product of specimen vector of local prediction error and vector formed of specimens of signal fragment is zero. For all coefficients which minimize prediction error, the prediction local error is orthogonal with precedent vectors (Fig.3):

$$\sum_n e_m(n) s_m(n-i) = 0; \; 1 \le i \le p \tag{28}$$

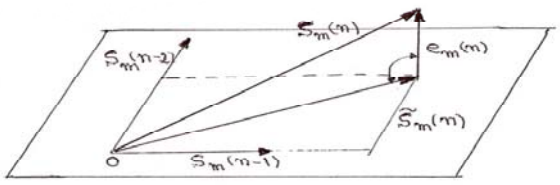

Fig. 3 Orthogonal error prediction

Equation (28) is made by p linear equations

$$\sum_n s_m(n) s_m(n-j) =$$

$$= \sum_{j=1}^{p} a_j \sum_n s_m(n-i) s_m(n-j), \quad 1 \le i \le p \tag{29}$$

Coefficients of covariant are defined by:

$$R_m(i,k) = \sum_n s_m(n-i) s_m(n-j) \tag{30}$$

Combining (29) and (30) equations results *Yule-Walker* equations (see [2]) with a_i variables:

$$\sum_{j=1}^{p} a_j R_m(i,k) = R_m(i,0); \quad i = \overline{1,p} \tag{31}$$

Result prediction total error:

$$\hat{E}_m = \sum_n s_m^2(n) - \sum_{j=1}^{p} a_j \sum_n s_m(n) s_m(n-j)$$

$$= R(0,0) - \sum_{j=1}^{p} a_j R(0,j) \tag{32}$$

In order to solve equation (31) is possible to use different methods like autocorrelation, covariates or any method used for solving differential equations [1].

3.3. AUTOCORRELATION METHOD

With this method is possible to obtain linear prediction coefficients a_i using Levinson –Durbin algorithm [2].

The following conditions are necessary:

$$s_m(n) = \begin{cases} s(m+n).w(n); & 0 \le n \le N\text{-}1 \\ 0 & ; \text{ otherwise.} \end{cases} \tag{33}$$

where w(m) is speaking signal time for $0 \le n \le N\text{-}1$.

Result for $m < 0$ signal error $e_n(m)$ is zero because $s_m(n) = 0$ for any $n < 0$ and there is no prediction error.

The same thing happens for $n > N\text{-}1+p$ because $s_m(n) = 0$ for any $n > N\text{-}1$.

According with equation (31) the new medium value for prediction error minimum is

$$E_m = \sum_{n=0}^{N-1+p} e_m^2(n) \tag{34}$$

And covariant coefficients $R_m(i,k)$ are:

$$R_m(i,k) = \sum_{n=0}^{N-1+p} s_n(n-i) s_m(n-k) =$$

$$= \sum_{n=0}^{N-1-(i-k)} s_m(n) s_m(n+i-k); \tag{35}$$

where: $1 \le i \le p$; $0 \le k \le p$.

Because covariant coefficients $R_m(i,k)$ are dependable only by i and k independent variables result:

$$R_m(i,k) = r_m(i-k) = \sum_{n=0}^{N-1-(i-k)} s_m(n) s_m(n+i-k) \tag{36}$$

And because the autocorrelation function is even, $r_m(-k)=r_m(k)$, the LPC equations are:

$$\sum_{k=1}^{p} r_m(|i-k|)a_k = r_m(i);\ 1\le i\le p \qquad (37)$$

Or in matrix form

$$\Re a = r \qquad (38)$$

where:

$$\Re=\begin{bmatrix} \mathbf{r}_m(0) & \mathbf{r}_m(1) & \mathbf{r}_m(2) & \cdots & \mathbf{r}_m(p-1) \\ \mathbf{r}_m(1) & \mathbf{r}_m(0) & \mathbf{r}_m(1) & \cdots & \mathbf{r}_m(p-2) \\ \mathbf{r}_m(2) & \mathbf{r}_m(1) & \mathbf{r}_m(0) & \cdots & \mathbf{r}_m(p-3) \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \mathbf{r}_m(p-1) & \mathbf{r}_m(p-2) & \mathbf{r}_m(p-3) & \cdots & \mathbf{r}_m(0) \end{bmatrix};$$

$$a=\begin{bmatrix} a_1 \\ a_2 \\ a_3 \\ \vdots \\ a_p \end{bmatrix};\quad r=\begin{bmatrix} \mathbf{r}_m(1) \\ \mathbf{r}_m(2) \\ \mathbf{r}_m(3) \\ \vdots \\ \mathbf{r}_m(p) \end{bmatrix}$$

Because autocorrelation matrix $\Re$ is a Toeplitz matrix (symmetrical and positively defined), the system can be resolved using Levinson-Durbin algorithm.

Observation

If $\Re^{-1}$ exists, the system can be solved with $a=\Re^{-1}r$.

That is the well known covariate method.

3.4. LEVINSON-DURBIN RECURRENT ALGORITHM

Prediction coefficients a_k can be easily found with autocorrelation method using Levinson-Durbin recurrent algorithm:

P1. For $i=\overline{1,p}$

$$E_0 = r(0)$$

$$k_i=\frac{\left(r(i)-\sum_{j=1}^{i-1}a_{i-1}(j)r(i-j)\right)}{E_{i-1}}$$

P2. For $j=1,2,...,i-1$

$$a_j(i)=k_i$$

$$a_i(j)=a_{i-1}(j)-k_i a_{i-1}(i-j)$$

$$E_i=(1-k_i^2)E_{i-1}$$

Example: For p = 2

$$E_0=r(0);\ \ k_1=r(1)/r(0);$$

$$a_1(1)=k_1=r(1)/r(0);$$

$$E_1=(1-k_1^2)E_0=\frac{r^2(0)-r^2(1)}{r(0)};$$

$$k_2=\frac{r(2)r(0)-r^2(1)}{r^2(0)-r^2(1)};$$

$$a_2(2)=k_2=\frac{r(2)r(0)-r^2(1)}{r^2(0)-r^2(1)};$$

$$a_2(1)=a_1(1)-k_2a_1(1)=\frac{r(1)r(0)-r(1)r(2)}{r^2(0)-r^2(1)};$$

$$a_1=a_2(1);\ a_2=a_2(2)$$

Observation

With found linear prediction coefficients is possible to calculate cepstral coefficients using differential equation (18).

REFERENCES

1. Anton, Gh., Radu, Gh., Socaciu, T., *Ecuaţii cu diferenţe şi scheme de calcul*, Editura Albastră, Cluj-Napoca, 2008;
2. Huang, H., Acero, A., Hon, H.W., *Spoken Language Processing – A Guide to Theory, Algorithms and Systems Development*, , New Jersey, Prentice–Hall, 2001;
3. Mateescu, A., Dumitriu, N., Stanciu, L., *Semnale şi Sisteme*, Editura Teora, Bucureşti, 2002;
4. Mafra, A.T., Simoes, M.G., *Text independent automatic speaker recognition using self organizing maps*, 39th IAS Annual Meeting Conference, Record of the Industry Applications Conference, 2004;
5. Rabiner, L., Juang, B.H., *Fundamentals of speech recognition*, Prentice Hall, Englewood Cliffs, New Jersey, 1993;
6. Tokuda, K., Masuko, T., Kobayashi, T., Imai, S., *Mel-generalized cepstral analysis – a Unified Approach to Speech Spectral Estimation*, Proc. of ICSLP-94, 1994, pp. 1043-1046.