SOME COMMENTS ON THE DESIGN AND IMPLEMENTATION OF FIR FILTERBANKS FOR SPEECH RECOGNITION

Jay T. Rubinstein * Harvey F. Silverman **

Laboratory for Engineering Man/Machine Systems (LEMS)
Division of Engineering
Brown University
Providence, RI 02912

Abstract

A program has been written which can generate banks of FIR filters using optimal-equiripple design. The composite responses of these filterbanks are very flat and have a stopband rejection of approximately 60 dB. Also, for purposes of speech recognition, energy estimates from filterbank output are frequently used, and are derived from oversampled, rectified input. A very substantial reduction in the number of calculations can be made if filter outputs are decimated while introducing a tolerable amount of error. This idea has been experimentally evaluated.

I. Introduction

Several studies have shown that FIR digital filters are an appropriate choice for the analysis of signals for speech recognition (Refs. 1,2,3). The linear-phase property implies that an ultra-flat composite response is achievable. Unfortunately, it is quite tedious to design a filterbank with the ultra-flat property by hand, even using the optimal design program of Parks, McClellan and Rabiner (PMR) (Refs. 4,5).

FIR filterbanks with a flat composite response have been designed by means of windowing, but, due to non-optimality, long filters were required when narrow-transition regions were specified (Ref. 6). The windowing technique produces predictable, reflected transition bands about the filterbank breakpoints. This symmetry of transition slope results in a flat composite response if the half-amplitude points of two adjacent filters are located at the desired frequency breakpoint between them. In this paper, optimal-equiripple design is automated for filterbanks, using a similar criterion to that of Ref. 6 for the setting of breakpoints.

Each filter output from an FIR filterbank normally outputs at the sample rate, FS, (say, 10kHz for recognition). It is is then rectified and lowpass filtered to give an energy estimate. A recognizer obtains these estimates from each filter at a rate of from 50 to 100 Hz. We hypothesize that the rectified signal is greatly oversampled when used for this purpose. Thus, much computation can be saved by applying the filters at a rate substantially lower than FS, which is feasible only for FIR filters. This technique, however, introduces some error in the resulting energy estimates, which should increase with the decimation. If the error is predictable and, for practical purposes, small, it is possible either to compute a given order filterbank more quickly, or to improve the resolution of the filterbank.

II. Automatic Design of Optimal FIR Filterbanks

To generate a standard bandpass filter, PMR requires the specification of the number of points of the filter, six band edges and three ideal band-gain values. For our purposes, the band-gain values will always be fixed to 0.0, 1.0, 0.0. (PMR also allows the designer to weight the importance in each band. It has been found experimentally that unity weighting for all bands works best for our filterbank program, and thus unity weighting will be assumed throughout.) As the first band-edge specification will always be 0.0, and the last band edge 0.5, only four band edges need be given to a call to PMR. Thus, for the kth filter of a filterbank, we must specify E1(k), the upper edge of the lower stopband, E2(k), E3(k), the lower and upper passband edges, respectively, and E4(k), the lower edge of the upper stopband. PMR returns the filter impulse response and ripple specifications.

---

* Currently at University of Washington School of Medicine, Seattle, WA

** Work partially supported by NSF Grant ECS-8113484

17.10

 CH1841-6/83/0000-0812 $1.00 © 1983 IEEE ICASSP 83, BOSTON

In our program, the number of points for all filters is considered to be the same, N. This, as well as the number of filters, M, in the filterbank, and the ideal breakpoints, B(k), specify the system. (Note that B(k) is specified in normalized frequency.) The algorithm that our program implements is as follows:

1. Input the filterbank specifications N, M, and B(k), k=0 to M. (Note: B(0) and B(M) define the skirts of the composite filterbank.)

2. Calculate all the filter widths, W(k) = B(k)-B(k-1) for k = 1 to M.

3. Determine the narrowest(minimum) filter width WN. Set S=WN/2.

4. Set E1(k) = B(k-1)-S for k=1 to M.

5. Set E4(k) = B(k)+S for k=1 to M.

6. Set E2(k) = E3(k) = (B(k)-B(k-1))/2 +B(k-1) for k=1 to M.

7. Start a loop on k = 1 to M.

   a. For cases where W(k) < 0.045 then skip to Step 7.e.

   b. For W(k) >= 0.045 correction is required. The passband spread is defined as PS(k) = 2*W(k)*W(k).

   c. E2(k) = E2(k)-PS(k)

   d. E3(k) = E3(k)+PS(k)

   e. End of Loop

8. Start a loop on k = 1 to M

   a. Call PMR using N, E1(k), E2(k), E3(k), and E4(k)

   b. Take 1024-point DFT of impulse response.

   c. Locate the half-amplitude points by search, FL(1) and FH(1).

   d. Set: DL(1) = FL(1)-B(k-1) and DH(1) = FH(1)-B(k).

   e. E2(k;0) = E2(k), E3(k;0) =E3(k).

   f. Start a loop on j =1 to 6.

      i) Convergence factor C(j)=1.0.

      ii) E2(k;j) = E2(k;j-1)-C(j)*DL(j).

      iii) E3(k;j) = E3(k;j-1)-C(j)*DH(j).

      iv) Call PMR using N, E1(k), E2(k;j), E3(k;j), and E4(k).

      v) Take 1024-point DFT of impulse response.

      vi) Locate the half-amplitude points by search, FL(j+1) and FH(j+1).

      vii) DL(j+1) = FL(j+1)-B(k-1), and DH(j+1) = FH(j+1)-B(k).

      viii) If DL(j+1) > 0.0 and DH(j+1) < 0.0 skip to Step 8.f.xi

      ix) Filter too wide; narrow it: C(j) = 0.15*C(j).

      x) Return to Step 8.f.ii

      xi) End of loop on j.

   g. End of loop on k.

9. End of Algorithm.

It is important to note that there are several important exceptions to the basic algorithm as listed above. A few of these are listed as follows:

E1(1) = E2(1)-0.03 to prevent excessive overshoot on the first filter.

Iteration is only for E3(1), and not E2(1).

C(1) is 0.6 for the first filter, rather than 1.0.

If B(M) > 0.475 then PS(M) is split to PSL(M) = 1.67*PS(M), and PSH(M) = 0.167*PS(M).

The program, written in FORTRAN, and a full explanation of the algorithm and exceptions may be found in Reference 7. The rates at which the transition bands converge to their desired locations and indeed whether they converge at all, is dependent upon a judicious first choice of filter parameters. We hypothesize that three or more filters will probably be desired and no more than two filters should overlap significantly at any one frequency. Because of this overlap restriction and the need for symmetry in allowed transition widths, no filter should be allowed a transition band beyond its breakpoint plus half the narrowest filter width.

III. Some Filterbank Examples

Figure 1 is a filterbank generated for use in the LEMS speech recognizer (Ref. 8). The input parameters for Figure 1 are:

N = 96 M = 6
B(k)=:.015 .05 .1 .15 .2 .25 .45

Figure 2 is a nine filter filterbank with three filters per octave. Its input parameters are:

N = 96   M = 9
B(k)=:.015 .05 .085 .12 .16 .2 .24 .32 .4 .48

In both cases, one should note the nearly "ideal" nature of the filterbanks; the composite passband is flat to +- 0.2dB (as opposed to a design where large excursions occur near transition bands) and the overall rejection is better than 60dB.

IV. Decimation for Energy Estimates

A signal from each filterbank filter is rectified (RMS or absolute value) and low-pass filtered and the result sampled at from 30-100Hz for input to a recognizer. We hypothesize that the input to the rectifier is oversampled for this purpose, and thus many calculations can be saved by applying each FIR filter at a rate lower than FS.

Figure 3 shows the error derived in passing a 750Hz sinewave, sampled at 10kHz through an absolute-value rectifier and a simple averager (rectangular lowpass filter) for 15 msec; error is given in dB relative to the integrated analytic solution. From Figure 3 it is clear that the sampling process introduces error at -51dB, and that one could decimate by four and introduce only -35dB error.

Figure 4 shows a similar result, except the input is the synthesized vowel ´AA´ as in fAther. No analytic solution was made, and thus the derived error is relative to the output of the digital system with no decimation. Here, one can decimate by three and still guarantee that only -35dB error is introduced. It should be noted that both results are for absolute-value rectification and simple averaging, as is the normal practice in hardware. If the cost is tolerable, then an FIR filter could be applied to the rectified signal, and less error introduced, or more decimation could be achieved. The reader is referred to Ref. 7 for a more complete treatment.

V. Summary

The filterbank program, given in Reference 7, can be used in any application which calls for an ultra-flat filterbank. For simple filterbanks - those with a few wideband filters - the program works extremely well and at a reasonable cost. For larger filterbanks the cost can be problematically large.

The results of the decimation experiment suggest that it is possible to decimate a signal by approximately three, without causing a large increase in estimation error.

References

1. Silverman, H. F. and Dixon, N. R. "State-Constrained Dynamic Programming(SCDP) for Discrete Utterance Recognition", Proceedings of 1980 ICASSP: Denver, pp.169-172

2. Dixon, N. R. and Silverman, H. F. "What are the Significant Variables in Dynamic Programming for Discrete Utterance Recognition", Proceedings of 1981 ICASSP: Atlanta, pp.728-731

3. Vickroy, Carolyn A., Silverman, H. F. and Dixon, N. R. "Study of Human and Machine Discrete Utterance Recognition(DUR)", Proceedings of 1982 ICASSP: Paris, pp.2022-2025

4. Rabiner, L. R., McClellan, J. H. and Parks, T. W. "FIR Digital Filter Design Techniques Using Weighted Chebychev Approximation", Proceedings of the IEEE, Vol. 63, No. 4, April, 1975

5. McClellan, J. H., Parks, T. W. and Rabiner, L. R. "A Computer Program for Designing Optimum FIR Linear Phase Digital Filters", IEEE Transactions on Audio and Electroacoustics, AU-21, No. 6, December, 1973

6. Schafer, R. W., Rabiner, L. R. and Herrmann, O. "FIR Digital Filter Banks for Speech Analysis", Bell System Technical Journal, Vol. 54, No. 3, March 1975

7. Rubinstein, Jay T. "Some Analysis and a Program for the Design of FIR Digital Filterbanks for Speech Recognition" ScM Thesis, Brown University, June 1983.

8. Niles, Les, Silverman, H. F. , and Dixon, N. R. "A Comparison of Three Feature Vector Clustering Procedures in a Speech Recognition Paradigm", Proc. ICASSP-83, forthcoming.

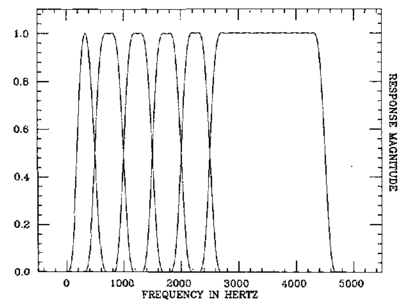

Fig. 1a

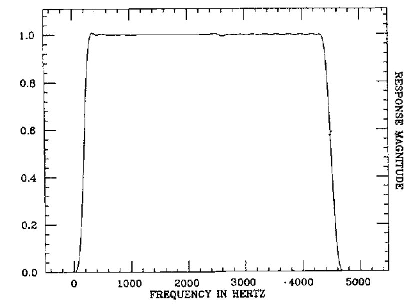

Fig. 1b

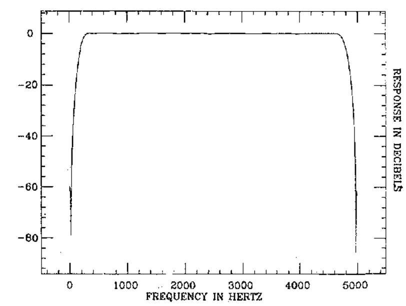

Fig. 1c

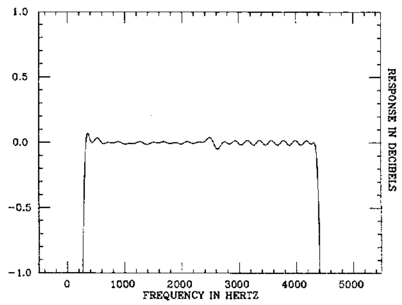

Fig. 1d

Six-Filter Filterbank Design

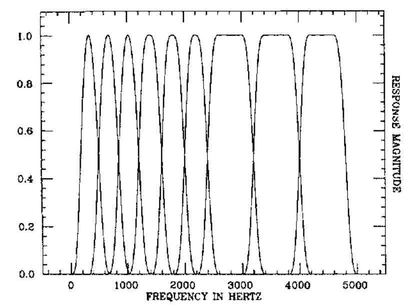

Fig. 2a

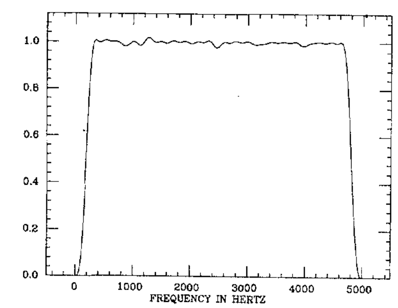

Fig. 2b

Nine-Filter Filterbank Design

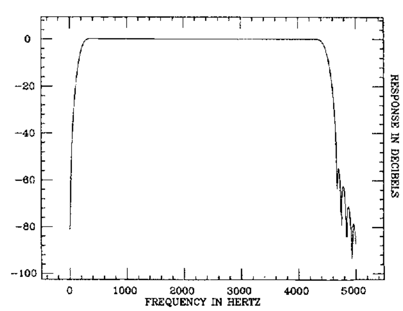

Fig. 2c

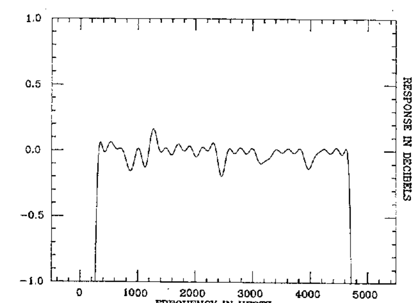

Fig. 2d

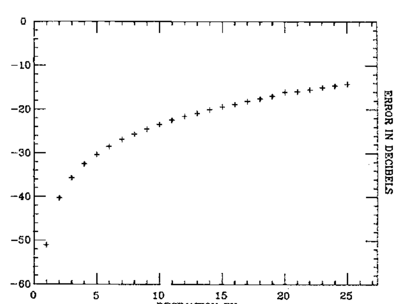

Fig. 3

Degradation of Energy Estimate for a Sine as Compared to an Analytic Solution

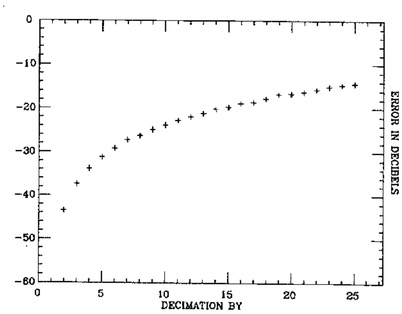

Fig. 4

Degradation of Synthetic, Voiced Speech as Compared Against No Decimation

17.10