The Use of Data Profiles in Class 2 and 3 Adaptive Filters

Ramfis Rivera-Colon and Claude S. Lindquist
Dept. of Electrical & Computer Engineering
University of Miami, Coral Gables, FL 33124, USA
Phone: (305) 284-3291 Fax: (305) 284-4044
rrivera@obsidian.eng.miami.edu

Abstract

Adaptive filters are optimum filters whose transfer functions adapt to changing input statistics. Such adaptation is usually needed because of fluctuating signal and/or noise conditions. These filters can be classified in order of increasing difficulty based upon the apriori assumptions about the signal and noise models. A Class 2/3 adaptive filter is a transitional filter between Class 2 and Class 3 filters. In this paper, we use data profiles to formulate Class 2/3 filters. Averaging provides an estimate of the signal and eliminates noise. Data profiles can also be used to formulate either Class 2 and/or Class 3 filters. Results show that this method provides improved estimates of expected signal and/or noise by smoothing the average signal profile. To demostrate this approach, these results are used to process EKG signals where no apriori information is available.

1 Introduction

Adaptive filtering is becoming increasingly popular. We have developed a variety of adaptive filtering algorithms using a somewhat unique approach. Class 1, 2 and 3 algorithms have been developed [1, Ch. 4].

Class 1 algorithms are the most reliable since models for both the signal and noise are known apriori. They only change their transfer functions when the expected signal $E\{S(j\omega)\}$ and/or the expected noise spectra $E\{N(j\omega)\}$ are updated.

Class 2 algorithms are derived from Class 1 filters by replacing the expected value of the signal spectrum by an estimate of it. In Class 2 filters, the function of the expected noise spectrum or the expected signal spectrum is unknown, but not both. Therefore the Class 2 filter performs less well than its Class 1 counterpart.

Class 3 algorithms require no apriori information about the signal and noise being processed. Therefore they are the most general. Class 3 filter algorithms utilize smoothing to obtain estimates of the expected signal, noise and their correlations. Both frequency domain smoothing [2],[3] and time domain smoothing [4] have been previously investigated.

The purpose of this paper is to obtain better estimates by smoothing the average signal profile rather than the signal itself. We anticipate this method will provide more reliable signal statistics.

2 Class 1–3 Estimation Filters

In the frequency domain, the transfer functions for Class 1–3 uncorrelated Wiener estimation filters are given as:

$$H_{ue1}(j\omega) = \frac{E\{|S(j\omega)|^2\}}{E\{|S(j\omega)|^2\} + E\{|N(j\omega)|^2\}} \tag{2.1}$$

$$H_{ue2}(j\omega) = \frac{E\{|S(j\omega)|^2\}}{E\{|S(j\omega)|^2\} + |R(j\omega) - kS(j\omega)|^2\}} \tag{2.2}$$

$$H_{ue3}(j\omega) = \frac{|< R(j\omega) >|^2}{< |R(j\omega)|^2 >} \tag{2.3}$$

It is assumed that the signal and noise are orthogonal, or equivalently, are uncorrelated and the noise has zero mean. The phase of these $H_{uex}(j\omega)$ transfer functions is always zero.

3 Class 1–3 Detection Filters

In the frequency domain, the transfer functions for Class 1–3 matched detection filters are given as:

$$H_{m1}(j\omega) = \frac{E\{S^*(j\omega)\}}{E\{|N(j\omega)|^2\}} \tag{3.1}$$

$$H_{m2}(j\omega) = \frac{E\{S^*(j\omega)\}}{E\{|R(j\omega) - kS(j\omega)|^2\}} \tag{3.2}$$

$$H_{m3}(j\omega) = \frac{< R^*(j\omega) >}{|R(j\omega) - < R(j\omega) > |^2} \tag{3.3}$$

The matched detection filter is obtained when we want to detect the signal and reject the rms noise. $H_{mx}(j\omega)$ weighs each spectrum input by the ratio of the conjugate signal spectrum to the noise power spectrum. The filter introduces a phase which is opposite to the signal phase. Thus all the output spectral components of the signal similar to the expected signal will be in phase, which tends to maximize its peak in the time domain. The output tends to be concentrated into a narrow pulse.

1058-6393/95 $4.00 © 1995 IEEE

The transfer functions for Class 1–3 inverse detection filters are given as:

$$H_{i1}(j\omega) = \frac{1}{E\{S(j\omega)\}} \tag{3.4}$$

$$H_{i2}(j\omega) = \frac{1}{R(j\omega) - kE\{N(j\omega)\}} \tag{3.5}$$

$$H_{i3}(j\omega) = \frac{1}{< R(j\omega) >} \tag{3.6}$$

The inverse filter outputs an impulse when only the signal and no noise is applied.

The transfer functions for Class 1–3 high-resolution uncorrelated Wiener detection filters are given as:

$$H_{hr1}(j\omega) = \frac{E\{S^*(j\omega)\}}{E\{|S(j\omega)|^2\} + E\{|N(j\omega)|^2\}} \tag{3.7}$$

$$H_{hr2}(j\omega) = \frac{E\{S^*(j\omega)\}}{E\{|S(j\omega)|^2\} + |R(j\omega) - kS(j\omega)|^2\}} \tag{3.8}$$

$$H_{hr3}(j\omega) = \frac{< R^*(j\omega) >}{< |R(j\omega)|^2 >} \tag{3.9}$$

The high-resolution detection filter outputs an impulse when both the signal and noise are applied. It maximizes the output signal to a total rms output ratio. It reduces to a matched detection filter at frequencies where the spectral SNR is low and to an inverse filter at high spectral SNR. The filter results from a combination of an estimation (Wiener) filter and an inverse detection filter. The Wiener estimation filter estimates the signal $S(j\omega)$ and the inverse filter converts the signal into a pulse.

4 Class 2/3 Filters

In Class 2 filters, the estimate $\widehat{E}\{S(j\omega)\}$ is assumed to be k times the expected signal spectrum $E\{S(j\omega)\}$. The expected signal spectrum $E\{S(j\omega)\}$ is known but the magnitude scaling constant k is unknown apriori. The constant k must be estimated via some aposteriori algorithm.

If the constant k from the Class 2 filters cannot be estimated, a different algorithm not involving k should be used. For example, in the Class 2 filter denominator containing k, the values $E\{S(j\omega)\}$ and $E\{R(j\omega)\}$ can be replaced by the Class 3 filter denominator which involves only $E\{R(j\omega)\}$ by setting $k = 1$. With this modification, the new Class 2/3 uncorrelated Wiener estimation transfer function is:

$$H_{ue}(j\omega) = \frac{E\{|S(j\omega)|^2\}}{< |R(j\omega)|^2 >} \tag{4.1}$$

The Class 2/3 detection transfer functions are:

$$H_m(j\omega) = \frac{E\{S^*(j\omega)\}}{|R(j\omega) - < R(j\omega) > |^2} \tag{4.2}$$

$$H_{hr}(j\omega) = \frac{E\{S^*(j\omega)\}}{< |R(j\omega)|^2 >} \tag{4.3}$$

This Class 2/3 algorithm is simpler to evaluate since k does not need to be estimated. $H_{i3}(j\omega)$ may be considered a Class 2/3 algorithm. We expect such filters to perform less well than the Class 2 filters, but better than the Class 3 forms.

5 Data Profiles

In practice there are many applications having waveforms that are repetitive but not periodic. Examples of these are the EKG and EEG signals. Conventional algorithms which do not exploit the repetitive nature of the signal are not as effective. By creating data profiles we can establish better estimates for the signal and noise. This is sometimes called "ensemble averaging" or "coherent averaging" in biomedical applications where an establish event trigger is used to align the signals.

Data profiles are obtained by properly sequencing repetitive data. The profile length must extend over the data period T_p seconds so the data is adequately characterized. The number of profiles depends on the amount of data available. If the stationarity period of the data is T_s seconds, then the maximum number of profiles available for use is $M \approx round(T_s/T_p)$. For EKG analysis, we usually use up to two minutes of data which provide around 150 profiles.

6 EKG Example

For example, consider the one-half minute EKG shown in Fig. 1 [5]. The data is segmented and peak aligned to form an EKG profile shown in Fig. 2. The EKG has been partitioned into individual heart-beat sections having a length just under one second using the following *MATLAB* software commands [6]:

Profile Pre-processing

To normalize the data and remove the lower envelope (i.e., slowly varying dc level in the data) we use:

```
x=x-min(x); x=x/max(x);
% Normalizing ekg data between 0 and 1.
dx=[0,diff(x)];
% Find derivative of x and padding the dx
  vector.
thres1=mean(x)-0.5*std(x);
% Defining threshold for lower peaks.
R1=find((dx(1:N-1)<0) & (dx(2:N)>0) &
  (x(1:N-1)<thres1));
% Finding lower intersections -- R1.
envelope=spline(R1,x(R1),1:length(x))'
% Envelope formed from using spline
  interpolation.
x=x-envelope;
% Normalized data with lower envelope
  removed.
```

The peak-alignment/segmentation section consists of the following commands:

```
thres2=max(x)-2*std(x);
% Defining threshold for higher peaks.
R2=find((dx(1:N-1)>0) & (dx(2:N)<0) &
```

```
  (x(1:N-1)>thres2));
% Finding higher intersections -- R2.
col=Ncol/2;
% Desired # of columns for profiles.
R=R((R>col) & (R<N-col));
% Do not extend beyond ends.
for i=1:length(R)
  y(i,:)=x(R(i)-col:R(i)+col);
end
% Segment row x inside matrix.
```

Other algorithms can be used to align the data like centroid or correlation. The next step is to set the energy of each profile to unity using:

```
y(i,:)=y(i,:)/(sqrt(sum((y(i,:).^ 2));
% Root Energy Normalization.
```

The last step is to plot the profile data as:

```
[L M]=size(y);
% Creating indices for plotting.
profile=(1:L)' * ones(1:M);
% Generates offset matrix to profile data.
z=50*y + profile;
% For plotting all the profiles.
plot(z','-')
% Plotting all the profiles in linear
  scale with normalized amplitude.
```

7 Class 2/3 EKG Filters

The uncorrelated Wiener estimation and high-resolution detection filters given by (4.1) and (4.3) will now be formulated. The expected signal is estimated by taking the mean of the 25 EKG profiles using

$$E\{s(t)\} = \frac{1}{25}\sum_{i=1}^{25} r(t,i) = ifft(E\{S(j\omega)\}) \quad (7.1)$$

$$E\{S(j\omega)\} = \frac{1}{25}\sum_{i=1}^{25} R(j\omega,i) = fft(E\{s(t)\}) \quad (7.2)$$

where $r(t,i)$ is the i^{th} profile. The expected input power spectrum is estimated by taking the mean of the 25 EKG power spectrum profiles as

$$\begin{aligned} <|R(j\omega)|^2> &= \frac{1}{25}\sum_{i=1}^{25} |R(j\omega,i)|^2 \\ &= fft(E\{|S(j\omega)|^2\} + E\{|N(j\omega)|^2\}) \end{aligned} \quad (7.3)$$

The results of (7.2) and (7.3) are used to form the $H_{ue}(j\omega)$ and $H_{hr}(j\omega)$ transfer functions.

The original EKG data was very clean with little background noise. We decided to degrade the data by adding uniformly-distributed random noise at about a 20 dB peak signal-to-rms noise SNR as shown in Fig. 5. The expected signal estimate is shown in Fig. 3 using (7.1). Note by comparing these two figures that because of the averaging, the noise has been greatly reduced ($1/\sqrt{25} = 14$ dB) but the peak signal remains about the same. This raises the effective SNR to about 34 dB. The subroutine for adding noise and manipulating the signal is shown below:

Forming EKG Signal and Noise Spectra
Forming H_{ue} Transfer Function

```
n=rand(1:25*128); r=x+n;
% Adding random noise to the profiles.
R=fft(r')';
% Getting input spectrum R.
S=mean(R)';
% Signal after averaging out noise from R.
S(1)=0;
% Getting rid of the DC component in the S
  spectrum.
R_2=R.*conj(R);
% Input squared spectrum.
R_2m=mean(R_2)';
% Mean input squared spectrum.
Hue=S_2./(R_2m);
% Forming the Hue transfer function.
Hue(1)=0;
% Getting rid of the DC component in the
  filter gain.
```

The signal and noise spectral envelopes and the R-wave signal in time are shown in Fig. 4. Consider that the signal spectral envelope bandwidth is ω_s and the noise cross-band bandwidth is ω_n. Due to the -40 dB/dec asymptotic rolloff, these frequencies are related as:

$$SNR = (\omega_n/\omega_s)^2 \quad (7.4)$$

Therefore, $\omega_n = \sqrt{SNR}\,\omega_s$. As a rough approximation, the Wiener estimation filter has $H_{ue}(j\omega) = 1$ for $\omega < \sqrt{SNR}\,\omega_s$ and zero elsewhere.

We note with the Class 2/3 Wiener uncorrelated estimation filter that with a SNR of 10 dB, the bandwidth of the filter reduces to $10^{1/4} \approx 2$ times the effective bandwidth of the signal, allowing only some signal frequencies as shown in Fig. 6. As we increase the SNR to 20 dB, the filter bandwidth is $10^{1/2} \approx 3$ times, moving the effective bandwidth of the signal as shown in Fig. 8.

To further adjust the bandwidth, we add a power spectral weighting factor by using

$$H_{uew}(j\omega) = H_{ue}(j\omega)\ |E\{S(j\omega)\}| \quad (7.5)$$

This weighted estimation filter adaptively reduces the filter passband to the width of the signal spectrum allowing the filter to depend directly on ω_s rather than on $\sqrt{SNR}\,\omega_s$. The new weighted gain is shown in Fig. 10. The filter gain of the high-resolution detection filter is shown in Fig. 12.

Forming the Weighted H_{ue} Transfer Function
Forming H_{hr} Transfer Function

```
Huew=Hue.*abs(S)/max((S(2:128));
% Power spectral weight to adjust BW.
```

```
Huew(1)=0;
% Getting rid of DC component.
S_c=mean(conj(R))';
% Signal after averaging out
  noise from conj(R).
Hhr=S_c./(R_2m);
% Hhr transfer function.
Hhr(1)=0;
% Getting rid of DC component.
```

The output profiles are shown in Figs. 7, 9 and 11. The estimated output profiles using the weighting factor are greatly improved compared to the non-weighted version due to the reduction in the high frequency components. The high-resolution detection filter profiles have narrow output detection pulses occurring where the R waves are centered at about time $n = 65$. We could have chosen any other position of the EKG waveforms to output this detection pulse.

Moving Pulse to R-Wave Position

```
Chr=(Hhr*ones(1:25)).*R';
% Hi-resolution filter output in freq.
chr=real(ifft(Chr));
% Hi-resolution filter output in time.
chr=[chr(:,65:128),chr(:,1:64)];
% Moving detection pulse to the R-wave
  position.
```

8 Conclusions

This paper has shown the use of data profiles in designing filters. The profiles are formed by segmenting the input using a pre-processing algorithm. They are further aligned using a peak-alignment algorithm. The resulting profile is averaged using a mean to obtain expected signal. The averaging reduces the rms noise by a factor equal to the square root of the number of profiles, but not the signal.

The expected signal or its spectrum is then smoothed using different frequency domain transfer functions. Due to the half bandwidth, the weighted H_{uew} filter increases the width of the R wave by a factor of 2. Simulations show that this method of Class 2/3 filtering is robust to input variations. We feel that profiling has many applications which we are presently investigating.

References

[1] Lindquist, C.S., *Adaptive & Digital Signal Processing with Adaptive Filtering Applications*, Steward & Sons, FL, 1989.

[2] Lindquist, C.S., "Frequency domain algorithms for class 3 adaptive filters", Proc. 28th Midwest Symp. on Circuits & Systems, CA, pp. 178–181, Aug. 1985.

[3] Rizq, N.N., "Theory and design of class 3 adaptive filters", M.S.E.E. Thesis, University of Miami, FL, Dec. 1986.

[4] Rivera-Colon, R., C.S. Lindquist and S.P. Reddy, "Class 3 adaptive filters using time domain smoothing", Proc. 27th Asilomar Conf. on Signals, Systems & Computers, CA, Nov. 1–3, 1993.

[5] Lindquist, C.S., "Radial basis function transforms and their use in signal processing", Proc. 27th Asilomar Conf. on Signals, Systems & Computers, CA, Nov. 1-3, 1993.

[6] Moler, C., J. Little and S. Bangert, *PRO-MATLAB User's Guide*, The MathWorks, Inc., 1991.

Fig. 1 - EKG Waveforms for 25 Heart Beats

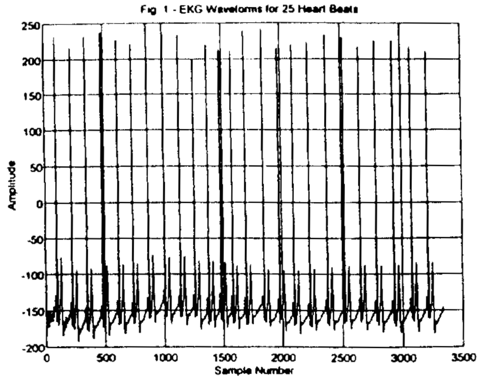

Fig. 2 - Input E

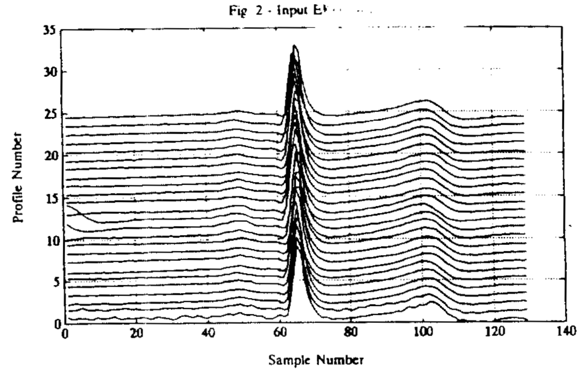

Fig. 3 - Expected Signal (Expected EKG Profile)

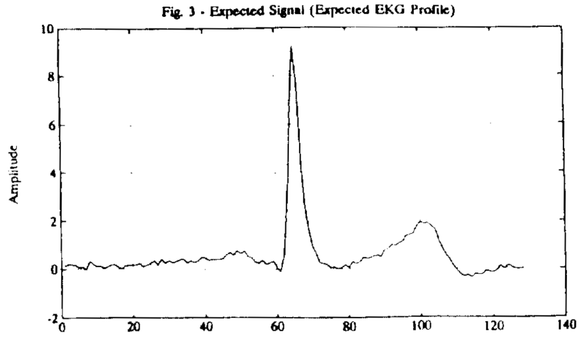

Fig. 4 (a) R-Wave, (b) R-Wave Spectrum |S(jw)| and White Noise Spectrum |N(jw)|

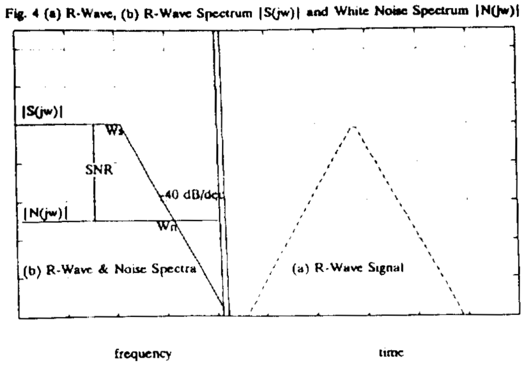

Fig. 5 - Noisy Input EKG Profiles (SNR = 20 dB)

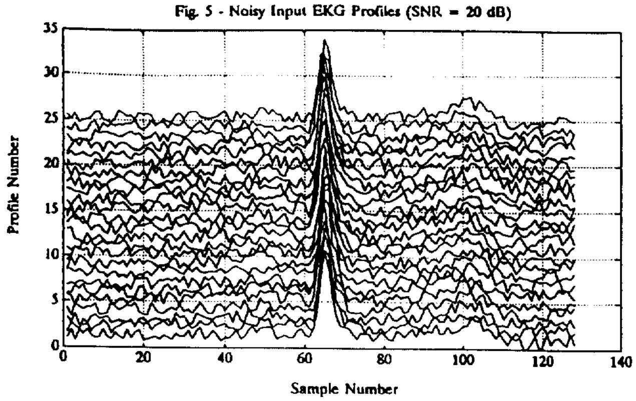

Fig. 6 - Hue Filter Gain (SNR = 10 dB)

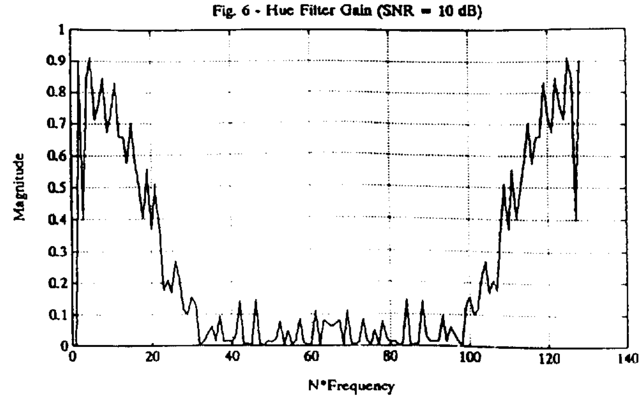

Fig. 7 - Hue Estimation Output Profiles

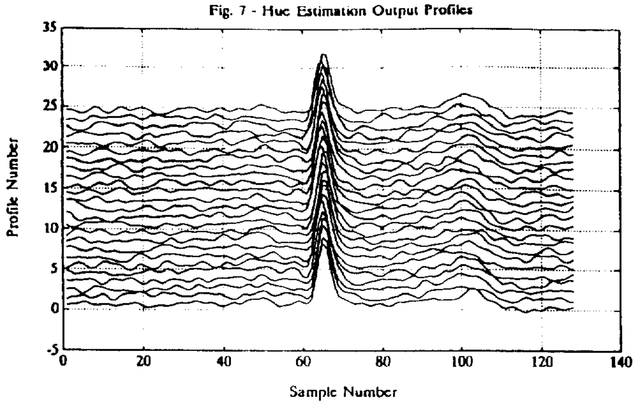

Fig. 8 - Hue Filter Gain (SNR = 20 dB)

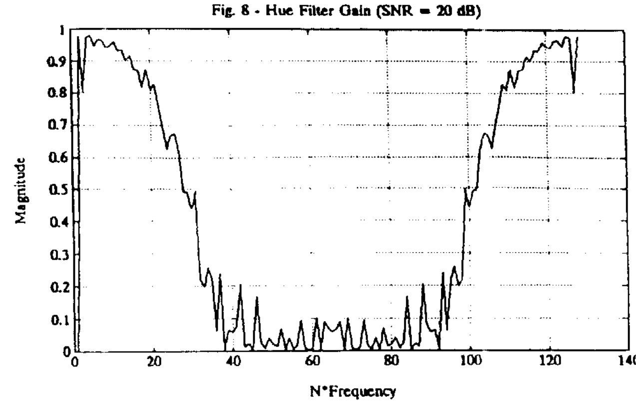

Fig. 9 - Weighted Estimation Hue Output Profiles

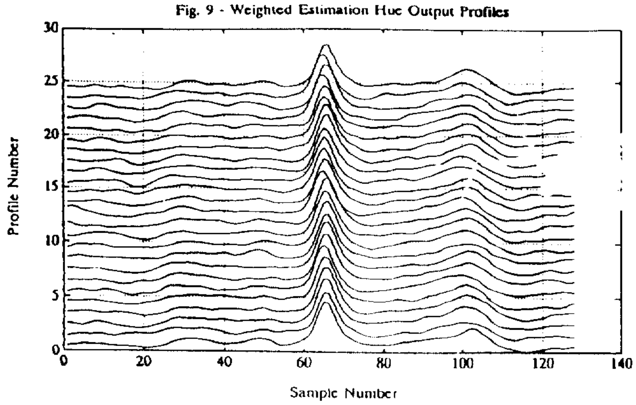

Fig. 10 - Weighted Hue Filter Gain (SNR = 20 dB)

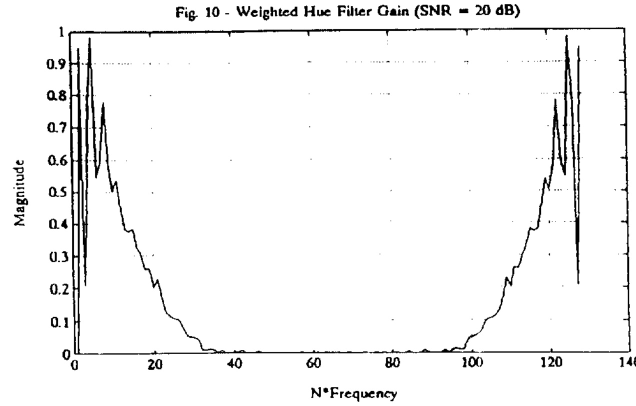

Fig. 11 - High-Resolution Detection Hhr Output Profiles

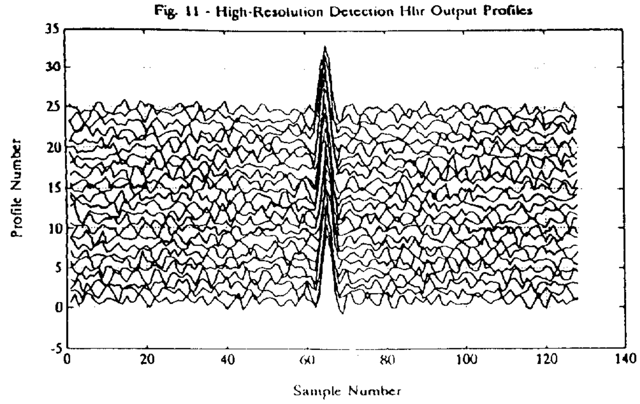

Fig. 12 - Hhr Filter Gain (SNR = 20 dB)