

Title:

***READILY IMPLEMENTED  
ENHANCED SINUSOID  
DETECTION IN NOISE***

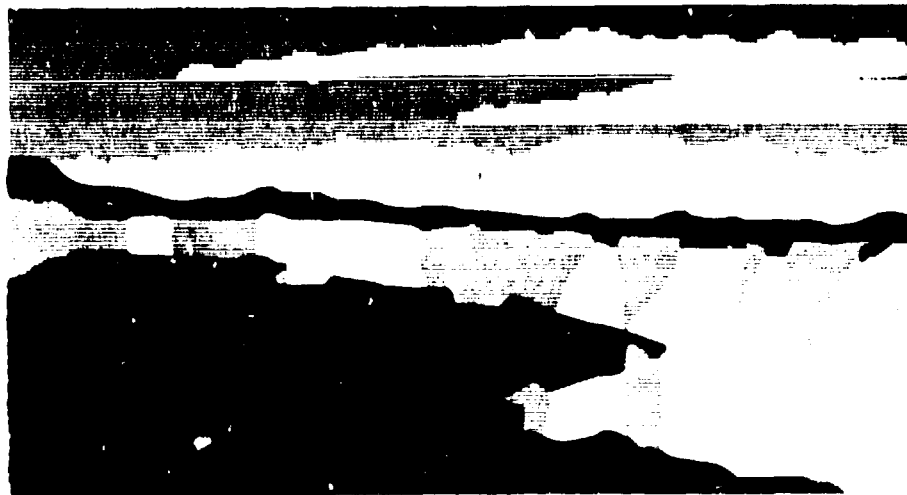
Author(s):

***Keith V. Lindsay***

Submitted to:

OST1

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.



**Los Alamos**  
NATIONAL LABORATORY

Los Alamos National Laboratory, an affirmative action/equal opportunity employer, is operated by the University of California for the U.S. Department of Energy under contract W 7405 ENG 36. By acceptance of this article, the publisher recognizes that the U.S. Government retains a nonexclusive, royalty-free license to publish or reproduce the published form of this contribution, or to allow others to do so, for U.S. Government purposes. The Los Alamos National Laboratory requests that the publisher identify this article as work performed under the auspices of the U.S. Department of Energy.

*So*

## **Readily Implemented Enhanced Sinusoid Detection in Noise**

Keith V. Lindsay

**ABSTRACT-** Significant efforts have been devoted, spanning many years, to the problem of sinusoid detection in noise. Many of these efforts have produced superb, yet complex, algorithms which may be difficult to use for a wide segment of the Digital Signal Processing (DSP) community. This paper presents a simple, easily implemented and highly effective method which solves this problem. This method severely degrades non-sinusoidal noise while leaving the embedded sinusoid(s) relatively undisturbed. The algorithm, simply put, exploits the difference between the net effect of integration and differentiation of sinusoids versus the effect of these operations on random noise and other signal sequences. The cross-correlation of a sine wave with its differentiated (and/or integrated) self is quite high. Conversely, the cross-correlation of a noise sequence with its differentiated (and/or integrated) self is much lower. Therefore, it is reasonable to assume that for sequences consisting of a sinusoid in noise, significant signal-to-noise-ratios (SNRs) in the correlation results are achievable using a combination of differentiation (and/or integration) and cross-correlation operations on such sequences. This technique has been applied to actual Doppler radar data, as well as to synthesized data, with excellent improvement in signal detection capability.

## I. INTRODUCTION

The problem of sinusoid detection in noise has wide application to systems such as radar, sonar, geophysics, and communications. Many techniques, developed over recent decades, have improved the ability to solve the problem. The simplest and, perhaps, most commonly used technique, is to Fourier Transform a candidate time sequence and examine the resulting spectra against a prescribed threshold criteria. Some of the primary methods of enhancing this approach are the use of traditional average periodogram analysis [1], as well as more advanced auto-regressive (AR), moving average (MA), auto regressive moving average (ARMA), maximum likelihood, and other methods [2] [3] to detect and accentuate periodicities, particularly in the case of closely spaced sinusoids. Implementation of these techniques, particularly with respect to existing DSP hardware structures, can be arduous at best. In addition, while sophisticated, robust techniques may encompass the method presented here, this method is very effective and easy to implement for the class of problem for which it was designed. A bonus to using the presented method is that it may be executed recursively with the recursion depth dependent on the expected performance and allowable computation time.

## II. ALGORITHM

The fundamental problem this technique challenges is the detection of low-level sinusoid signal(s) in noise. Large-signal detection is not precluded, but the computational expense is unnecessary, since conventional techniques are adequate in this case. The logic which lead to its development begins with a consideration of the fundamental discrete mathematical operations which could be applied to a discrete time series such that sinusoidal functions are either enhanced or left undisturbed, while the competing (non-sinusoidal) noise elements are somehow suppressed. This line of thought may lead to more sophisticated mathematical operations being applied in a similar way. However, the basic operations of differentiation and integration, with respect to sine and cosine functions are simple and predictable.

Let

$$x(t) = \sin(\omega t) \quad 2.1$$

then

$$\begin{aligned} \frac{d(x(t))}{dt} &= \omega \times \cos(\omega t) & 2.2 \\ &= \omega \times \sin\left(\omega t + \frac{\pi}{2}\right) \end{aligned}$$

Similarly,

$$\begin{aligned} \int x(t) dt &= -\frac{1}{\omega} \times \cos(\omega t) & 2.3 \\ &= \frac{1}{\omega} \times \sin\left(\omega t - \frac{\pi}{2}\right) \end{aligned}$$

With the exception of a  $\pm\pi/2$  phase shift and the scale factor, the sinusoid characteristics are undisturbed. Furthermore, since the  $\omega$  component in the scale factor (resulting from *differentiation* of a sinusoid) is not a function of time, it will cancel its inverse scale factor (resulting from *integration* of a

sinusoid) in the frequency domain. Noise containing random phase shifts, as well as signals exhibiting time-dependent phase characteristics, will be corrupted -- a desired characteristic in this case. The effect of the presented enhancement technique on chirped wave forms is not discussed in this paper.

While the characteristics of purely sinusoidal functions are preserved by differentiation and integration operations, the same is not true of noise sequences containing non-sinusoidal elements. For example, if some portion of the time series noise sequence is characterized by the arbitrary function  $z(t)$ , such that, over a short interval, the function is characterized by

$$z(t) = t^2 \tag{2.4}$$

then

$$\frac{d}{dt} z(t) = 2t = z'(t) \tag{2.5}$$

and, similarly for integration,

$$\int z(t) dt = \frac{t^3}{3} + C \tag{2.6}$$

Taking the case of the differential, since it becomes plainly obvious, further differentiation of equation 2.5 yields

$$\frac{d}{dt} z'(t) = 2 = z''(t) \tag{2.7}$$

and

$$\frac{d}{dt} z''(t) = 0 = z'''(t) \tag{2.8}$$

Clearly, the cross-correlation between  $z(t)$  and  $z'''(t)$  would be zero. On the other hand, the cross-correlation of  $x(t)$  (2.1) with  $x'(t)$  (2.2) would be very high. This distinction sets the logic for the algorithm in that it suppresses non-sinusoidal noise characteristics while leaving the sinusoidal elements relatively undisturbed. It follows that degrees of performance can, therefore, be linked to the number of times the sequence is differentiated and/or integrated and collectively cross-correlated, making the

algorithm performance level a function of the computation time and available machine storage.

Implementation of the algorithm on machines capable of performing the elementary tasks of discrete differentiation, integration, and convolution (usually via the FFT) is straightforward, making this approach particularly applicable to current DSP board architectures. Figure 2.1 shows the algorithm in diagram form.

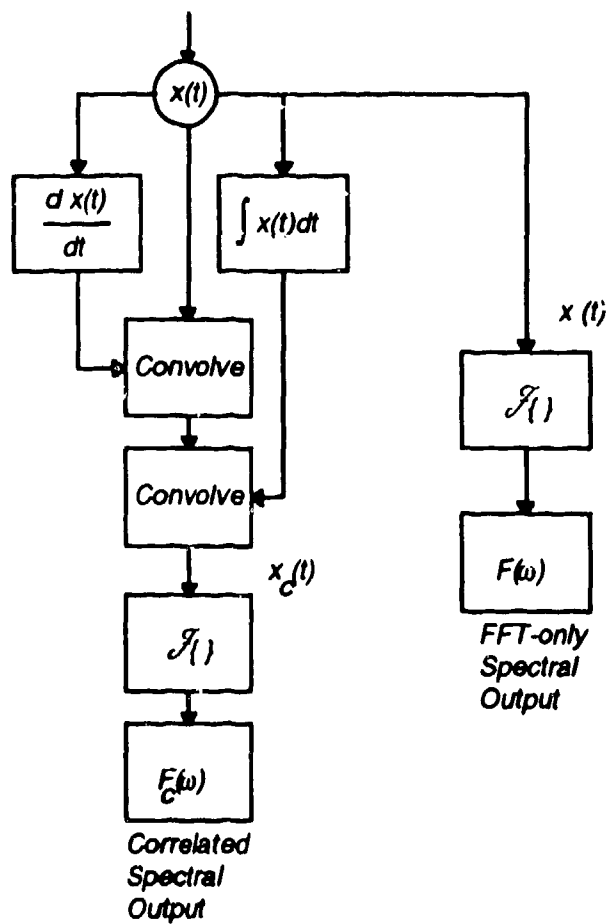


Figure 2.1. Detection Enhancement Algorithm Diagram.

### III. IMPLEMENTATION

Implementing this algorithm using a typical signal processing package or DSP development system is very straightforward, provided that the following library functions and operations exist or can be readily coded:

Difference Equation	$[y(n) = x(n) - x(n-1)]$
Cumulative Sum (Integration)	$[y(n) = \sum_{i=0}^{i=n} x(n)]$
Complex Discrete Fourier Transform (FFT)	
Standard complex math operations	[+, -, x, /, abs]
Discrete Hilbert Transform	(useful, but not required)

Table 3.1. DSP Library Functions and Operations.

This algorithm was originally intended to be applied to Doppler radar data consisting of in-phase (I) and quadrature (Q) time sequences. The quadrature sequence  $[x_q(n)]$  is, basically, the in-phase sequence  $[x_i(n)]$  shifted either  $+\pi/2$  or  $-\pi/2$ . A synthetic quadrature sequence can optionally be obtained from single-sequence time series signals by performing a Hilbert Transform of the sequence and assigning the result the "quadrature" sequence tag for the sake of this discussion. The reason for performing this function becomes clear in a moment. The data may be optionally windowed to reduce edge effects.

Then, let

$$x(n) = x_i(n) + i \times x_q(n), \quad n = 1, 2, \dots, N \quad 3.1$$

Next, let

$$\begin{aligned} x^{-1}(n) &= x(n) - x(n-1), \quad n = 2, 3, \dots, N \\ x^{-1}(1) &= 0 \end{aligned} \quad 3.2$$

and let

$$x^{+1}(n) = \sum_{i=0}^{i=n} x(i), \quad n = 1, 2, \dots, N \quad 3.3$$

At this point, we want to obtain the cross-correlation between  $x(n)$ ,  $x^{-1}(n)$ , and  $x^{+1}(n)$ . This can be done in the time domain (at the expense of computation time) or in the frequency domain (at the expense of storage requirements). In the time domain:

$$x_c(n) = x(n) * x^{-1}(n) * x^{+1}(n) \quad 3.4$$

If we choose to compute  $X_c(k)$  in the frequency domain, the time domain sequences are padded with three times the number of zeros (i.e., if  $x(n)$ ,  $x^{-1}(n)$ , and  $x^{+1}(n)$  are each 1024 points long, an additional 3072 zeros are appended to each sequence). Then, using the FFT, obtain

$$X(k) = \mathcal{F}\{x(n)\} \quad 3.5$$

$$X^{-1}(k) = \mathcal{F}\{x^{-1}(n)\} \quad 3.6$$

$$X^{+1}(k) = \mathcal{F}\{x^{+1}(n)\} \quad 3.7$$

$X_c(k)$  is obtained next:

$$X_c(k) = \left[ X(k) \times X^{-1*}(k) \right] \times X^{+1*}(k) \quad 3.8$$

At this point, the spectra  $X_c(k)$  may be examined for prominent sinusoids. We would choose to stop at this point if only single-sequence time series signals were processed. However, if the in-phase and quadrature series were processed, an additional step can further exaggerate the presence of the sinusoid(s). This is because the Fourier Transform of a complex sinusoid (cisoid) in the form of



$$y(t) = \sin(\omega t) \pm i \times \cos(\omega t) \quad 3.9$$

will produce a delta function in either the positive or negative half of the frequency domain (but not in both sides). Therefore, an additional step to further enhance a complex sinusoid is taken by forming the sequence\*

$$\Gamma_c(k) = X_c(k) - X_c(-k) \quad 3.10$$

The result,  $\Gamma_c(k)$ , can now be compared to a desired threshold criteria in order to make an ultimate decision on signal presence / absence. In the case of Doppler signals,  $\Gamma_c(k)$  has the added advantage of preserving the target direction (inbound or outbound) indication.

---

\* Suggested by Scott Robinson, LANI/IT-6.

#### IV. RESULTS

Quantitative measurements using synthetic data with various sinusoid SNRs were performed and are documented here for comparison purposes. The specified SNR for the synthesized time series was obtained using band limited noise combined with a sine wave at an arbitrary specified frequency. Sampled sequences consisting of 8192 points were generated using a sample rate of 50.1 KHz. The mean of the generated, band-limited noise sequence,  $N(n)$ , was measured and the appropriate scaling of the mean sine wave,  $S(n)$ , was performed such that the resulting SNR was

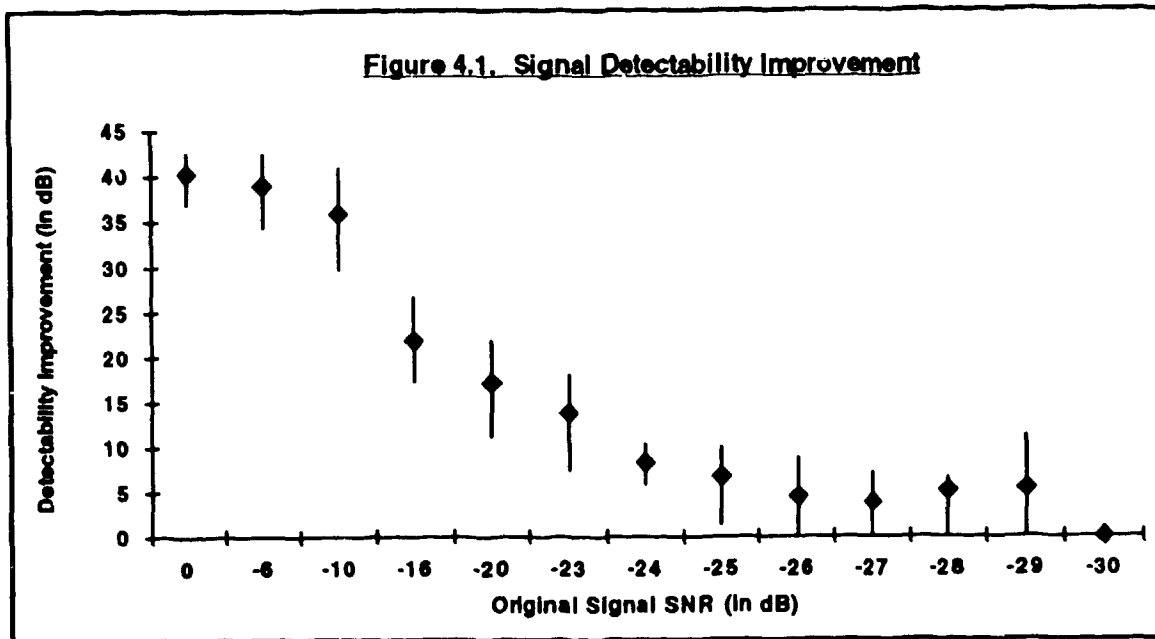
$$SNR = \frac{E\{|S(n)|\}}{E\{|N(n)|\}} \quad 4.1$$

An arbitrary discrete frequency of 300 Hz was chosen to represent an embedded sinusoid in a noise series band limited to 20 KHz. Synthesized time series SNRs from 0 to -30 dB were generated in order to quantitatively compare the enhanced signal detection results to the more traditional FFT-only technique. Table 4.1 presents the enhancement improvement results of the comparison for each trial.

Specified SNR	Trial #									
	1	2	3	4	5	6	7	8	9	10
0	36.98	39.09	42.58	40.14	41.76	39.75	40.22	42.55	39.98	39.65
-6	39.68	42.43	39.25	40.02	36.47	39.71	37.3	34.35	39.29	41.75
-10	37.1	36.92	29.75	34.64	36.06	36.34	35.18	40.86	38.96	34.01
-16	17.31	26.64	24.29	22.88	21.18	20.68	22.14	21.77	19.97	22.08
-20	11.236	18.825	18.44	16.199	15.22	17.896	16.943	21.81	17.476	18.912
-23	11.463	13.728	13.339	17.939	16.877	13.527	17.315	12.589	15.381	7.476
-24	9.434	8.218	6.999	10.241	7.198	7.786	5.886	7.837	9.161	9.574
-25	7.4067	6.397	8.082	7.178	9.718	3.923	7.559	6.848	9.923	1.369
-26	4.806	6.195	4.1059	4.623	2.028	4.549	N/A	0.905	8.7407	5.705
-27	6.679	7.114	4.313	1.859	0.414	1.242	6.088	3.218	3.557	N/A
-28	N/A	6.377	3.627	4.543	N/A	6.538	N/A	3.35	5.131	6.276
-29	N/A	N/A	3.598	N/A	N/A	1.253	N/A	N/A	11.082	N/A
-30	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	0.2443

Table 4.1 Sinusoid Detectability Improvement (in dB).

Averages were computed for the results in Table 4.1. These averages (the diamond markers in the plot) and the associated range of values are plotted in Figure 4.1.



Clearly, there exists a limit to both methods of detecting the signal. In the case of the above 20 KHz bandwidth, 8192-point sequences, the SNR limit is about -27 dB for reliable detection using the enhanced technique. The "bump" in the curve at -28 and -29 dB is due to the decreasing number of valid detections. Table 4.2 shows the percentage success rate of detection results (using ten random sequences per SNR level) for the selected method. Additional signal data could either be used to create longer sequences (as opposed to 8192-point sequences) or could be used to perform frame-to-frame averaging. Either approach would allow detection below the above-stated -27 dB limit. The point of the exercise, however, is to demonstrate the *detection improvement* using the presented technique over the FFT-only method for signals above the detection limit.

SNR (dB)	% Detect (FFT-only method)	% Detect (Enhanced method)
0	100	100
-6	100	100
-10	100	100
-16	100	100
-20	100	100
-23	100	100
-24	90	100
-25	90	100
-26	90	90
-27	60	90
-28	30	70
-29	10	30
-30	0	10

Table 4.2. Percent Detection Success.

Appendix A provides a representative set of the processed synthetic signals used for the above results for the reader's inspection. This appendix is organized such that it contains an FFT-only plot, followed by the enhanced version, to form pairs arranged from 0 to -30 dB SNR.

#### IV. CONCLUSION

A simple, effective technique to enhance detection of sinusoids in noise has been presented. It is easily implemented with standard DSP analysis packages and on DSP hardware using readily available "canned" software routines. Although the lower limit of the FFT-only approach is about the same for the presented technique, SNRs above this limit are significantly enhanced and detection reliability is improved in the region just above the minimum detectable threshold. This technique provides the practicing DSP engineer or signals analyst with a powerful, yet easily crafted tool. A sample M-file for 386-MATLAB\* which implements the enhancement technique is provided in Appendix B.

This paper has presented some representative synthetic data samples for comparison. The new method has been successfully applied to actual Doppler radar data with excellent results [4]. Other successful applications of this technique, or comparisons to detection methods found to be superior, are welcomed.

---

\* 386-MATLAB is a product of *The MathWorks, Inc.* based in South Natick, MA.

#### IV. REFERENCES

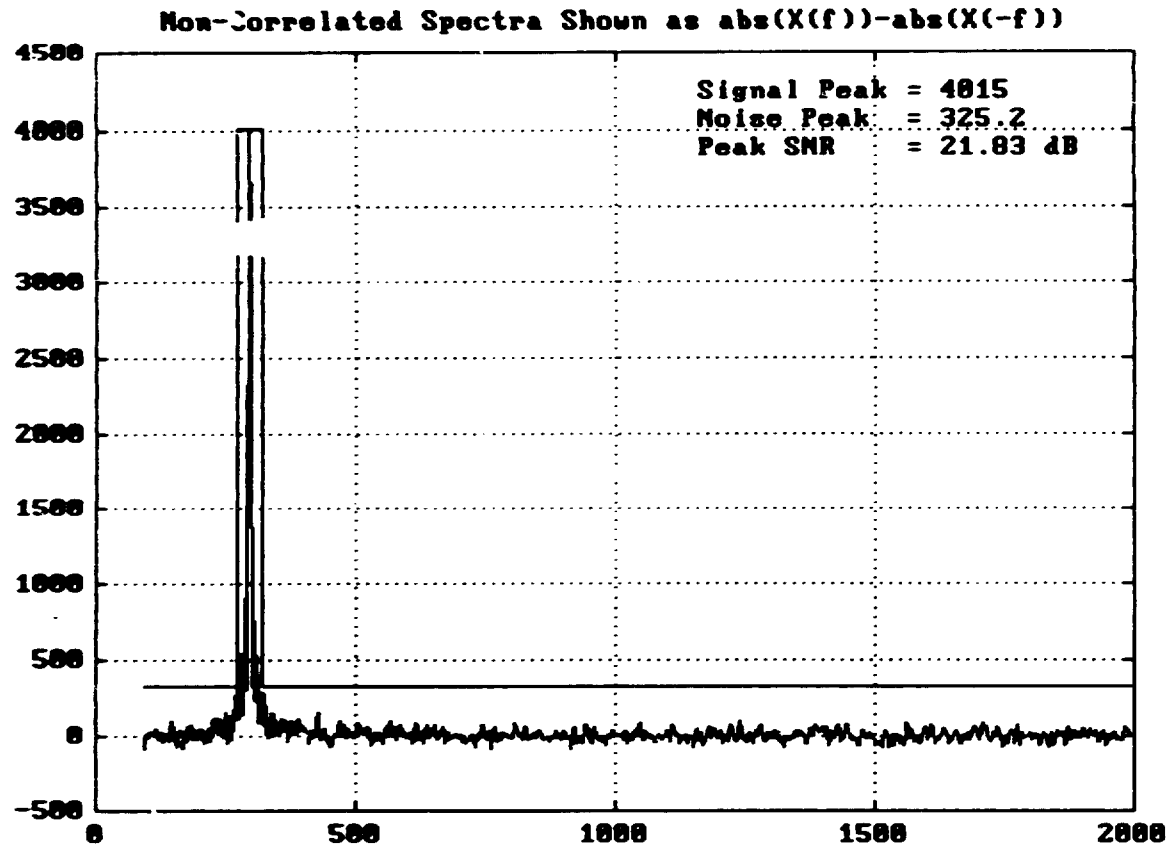
- [1] Alan V. Oppenheim and Ronald W. Schaffer, *Discrete-Time Signal Processing*. Englewood Cliffs, NJ: Prentice Hall, 1989.
- [2] Steven M. Kay and Stanley Lawrence Marple, Jr, "Spectrum Analysis-A Modern Perspective," *Proc IEEE*, vol. 69, pp. 1380-1419, Nov. 1981.
- [3] Steven M. Kay, "Robust Detection by Autoregressive Spectrum Analysis," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-30, pp. 256-269, Apr. 1982.
- [4] Robert Morrison, Scott Robinson, Keith Lindsay, and Dave Beck, *CW Doppler Radar Work at LANL/IT-6, Ongoing Research*, Nov. 1991.

## Appendix A

### TYPICAL RESULTS

#### List of Figures

Figure A1. FFT-only $\Gamma(k)$ (0 dB case).....	A.1
Figure A2. Correlated (Enhanced) $\Gamma_C(k)$ (0 dB case) .....	A.2
Figure A3. FFT-only $\Gamma(k)$ (-6 dB case) .....	A.3
Figure A4. Correlated (Enhanced) $\Gamma_C(k)$ (-6 dB case).....	A.4
Figure A5. FFT-only $\Gamma(k)$ (-10 dB case).....	A.5
Figure A6. Correlated (Enhanced) $\Gamma_C(k)$ (-10 dB case) .....	A.6
Figure A7. FFT-only $\Gamma(k)$ (-16 dB case).....	A.7
Figure A8. Correlated (Enhanced) $\Gamma_C(k)$ (-16 dB case) .....	A.8
Figure A9. FFT-only $\Gamma(k)$ (-20 dB case).....	A.9
Figure A10. Correlated (Enhanced) $\Gamma_C(k)$ (-20 dB case) .....	A.10
Figure A11. FFT-only $\Gamma(k)$ (-23 dB case).....	A.11
Figure A12. Correlated (Enhanced) $\Gamma_C(k)$ (-23 dB case) .....	A.12
Figure A13. FFT-only $\Gamma(k)$ (-24 dB case).....	A.13
Figure A14. Correlated (Enhanced) $\Gamma_C(k)$ (-24 dB case) .....	A.14
Figure A15. FFT-only $\Gamma(k)$ (-25 dB case).....	A.15
Figure A16. Correlated (Enhanced) $\Gamma_C(k)$ (-25 dB case) .....	A.16
Figure A17. FFT-only $\Gamma(k)$ (-26 dB case).....	A.17
Figure A18. Correlated (Enhanced) $\Gamma_C(k)$ (-26 dB case) .....	A.18
Figure A19. FFT-only $\Gamma(k)$ (-27 dB case).....	A.19
Figure A20. Correlated (Enhanced) $\Gamma_C(k)$ (-27 dB case) .....	A.20
Figure A21. FFT-only $\Gamma(k)$ (-28 dB case).....	A.21
Figure A22. Correlated (Enhanced) $\Gamma_C(k)$ (-28 dB case) .....	A.22
Figure A23. FFT-only $\Gamma(k)$ (-29 dB case).....	A.23
Figure A24. Correlated (Enhanced) $\Gamma_C(k)$ (-29 dB case) .....	A.24
Figure A25. FFT-only $\Gamma(k)$ (-30 dB case).....	A.25
Figure A26. Correlated (Enhanced) $\Gamma_C(k)$ (-30 dB case) .....	A.26

Figure A.1. FFT-only  $T(k)$  (0 dB case).



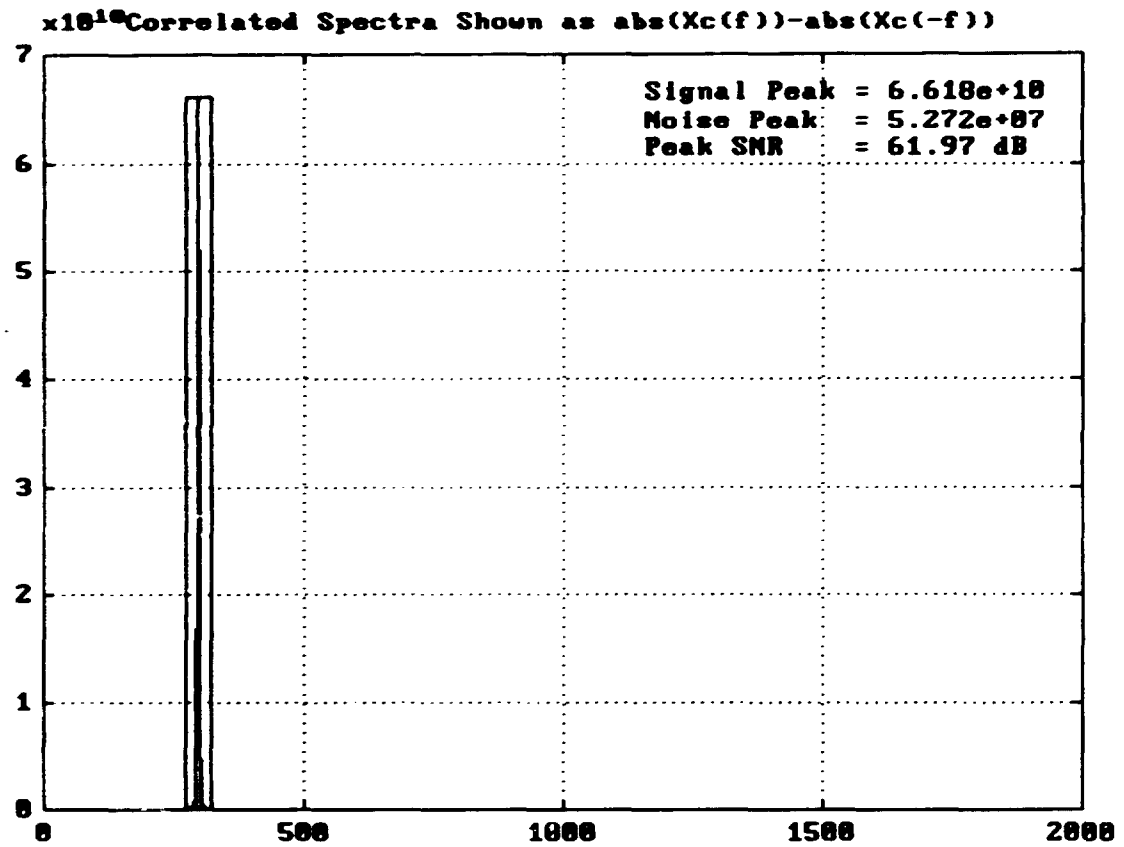
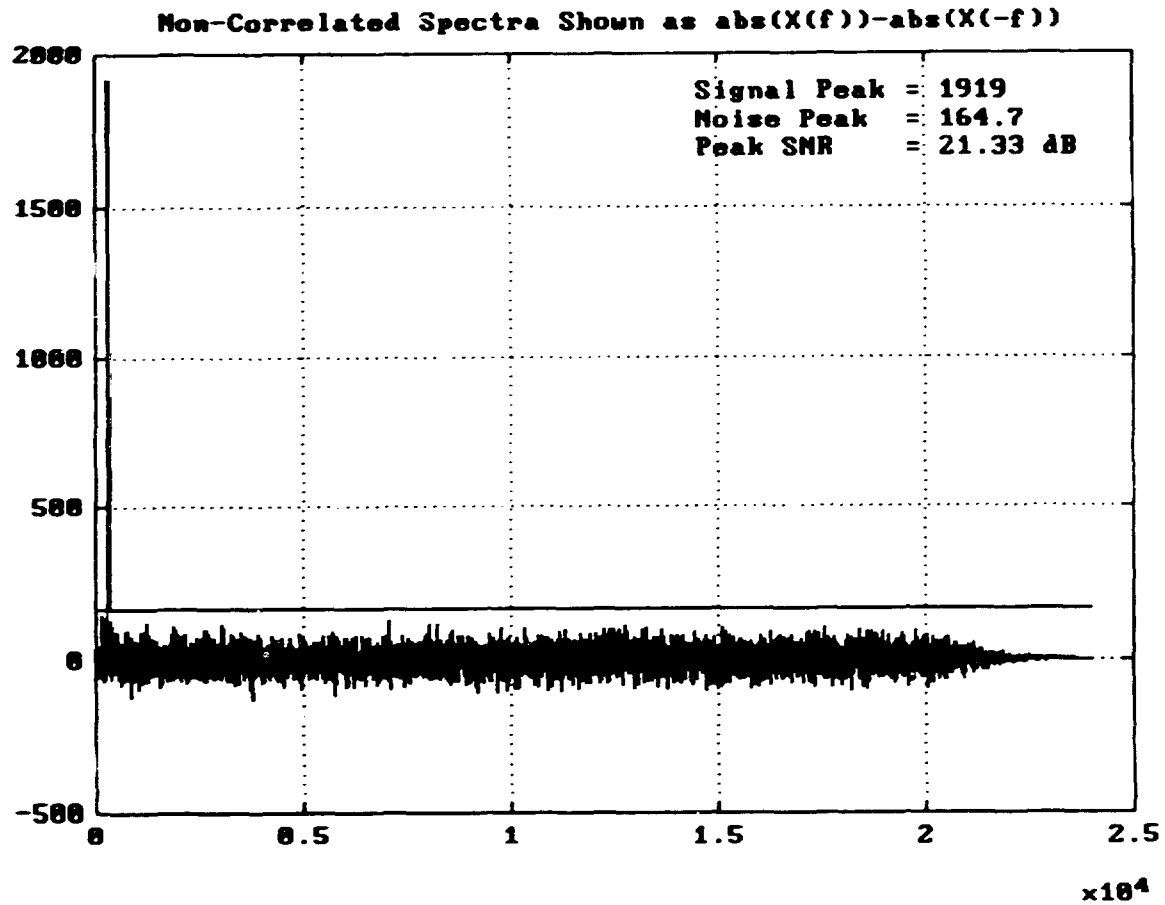


Figure A2. Correlated (enhanced)  $P_c(k)$  (0 dB case).

A.3

Figure A.3. PPT-only  $T(k)$  (-6 dB case).



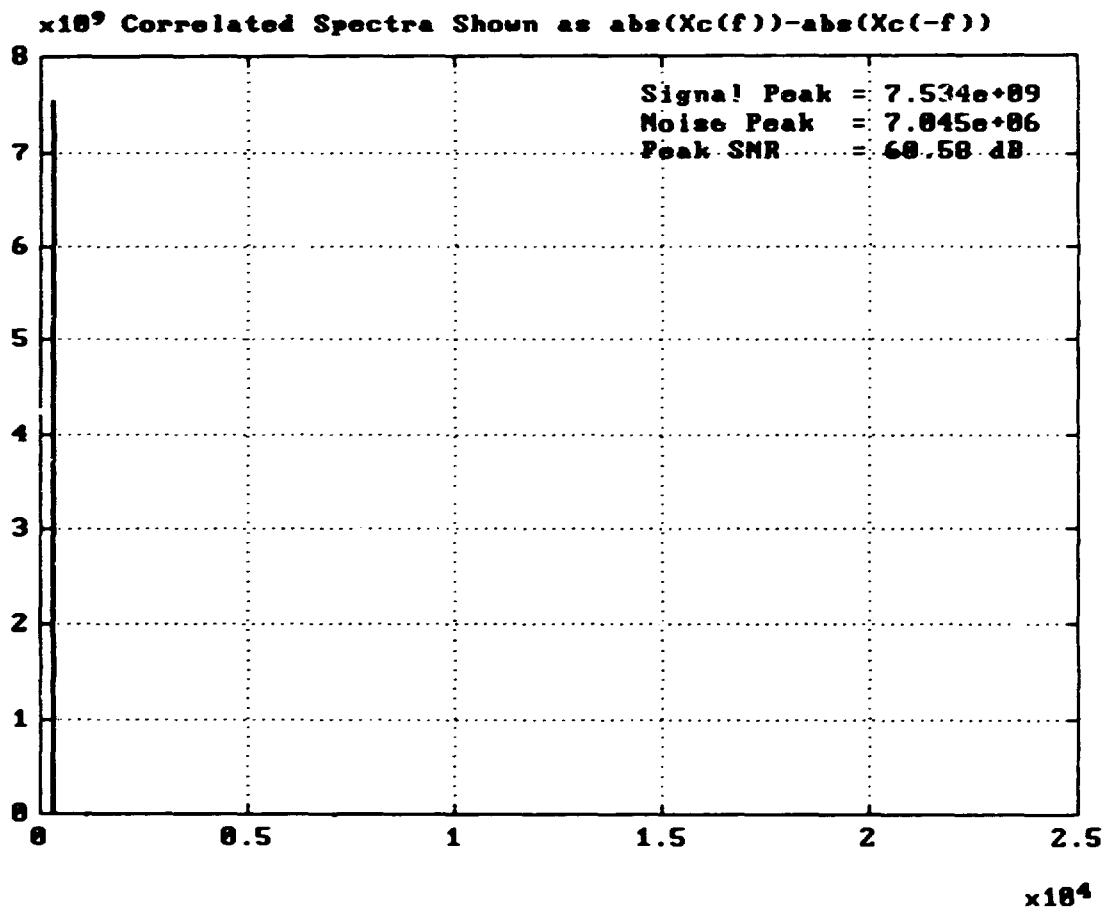


Figure A4. Correlated (Unhindered)  $T_c(k)$  (-6 dB case).

A.5

Figure A.5. PFI-only T(k) (-10 dB case).

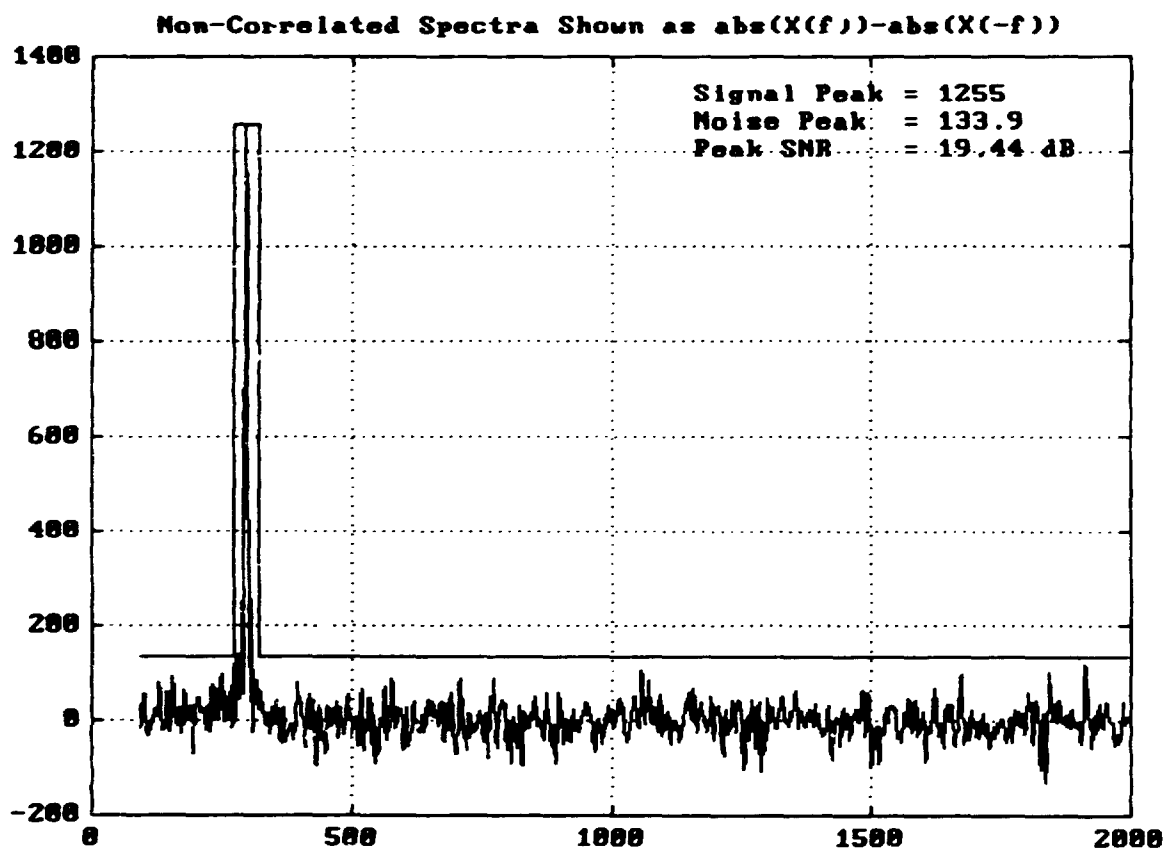
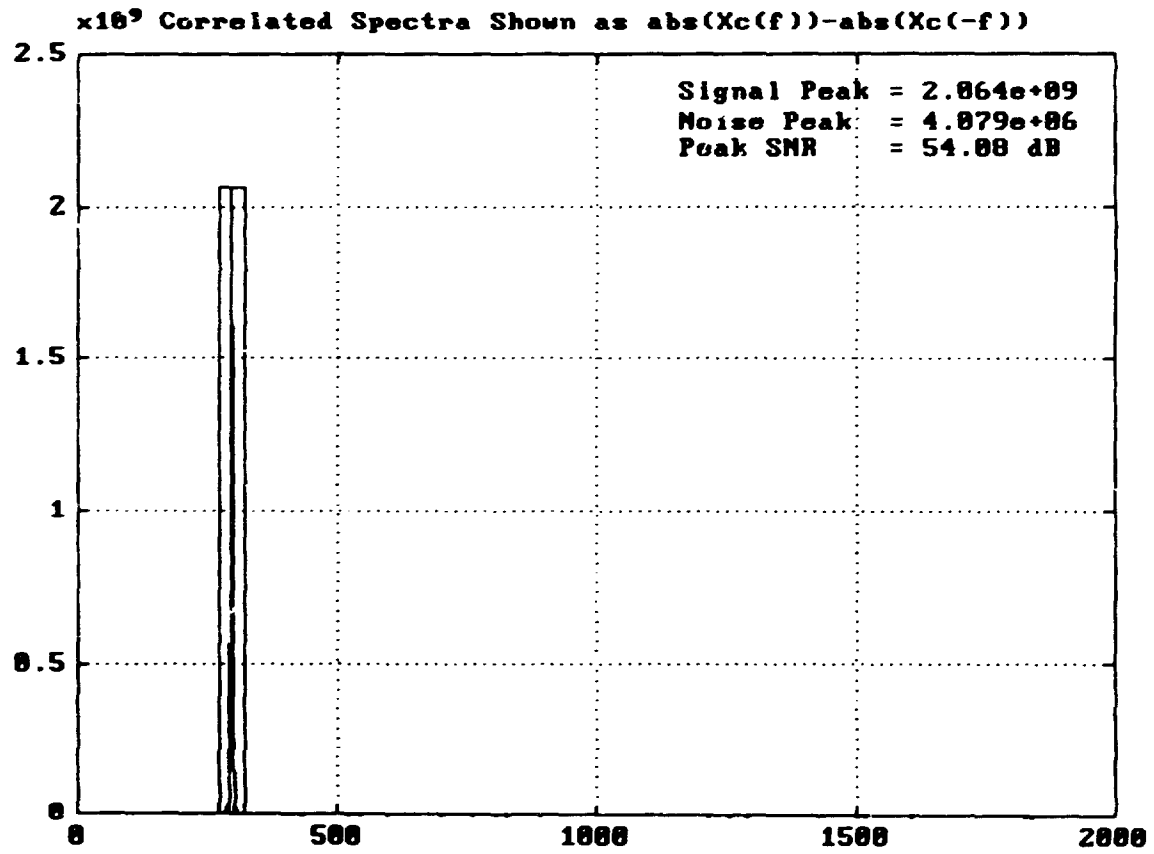


Figure A6. Correlated (Enhanced)  $F_c(k)$  (-10 dB case).

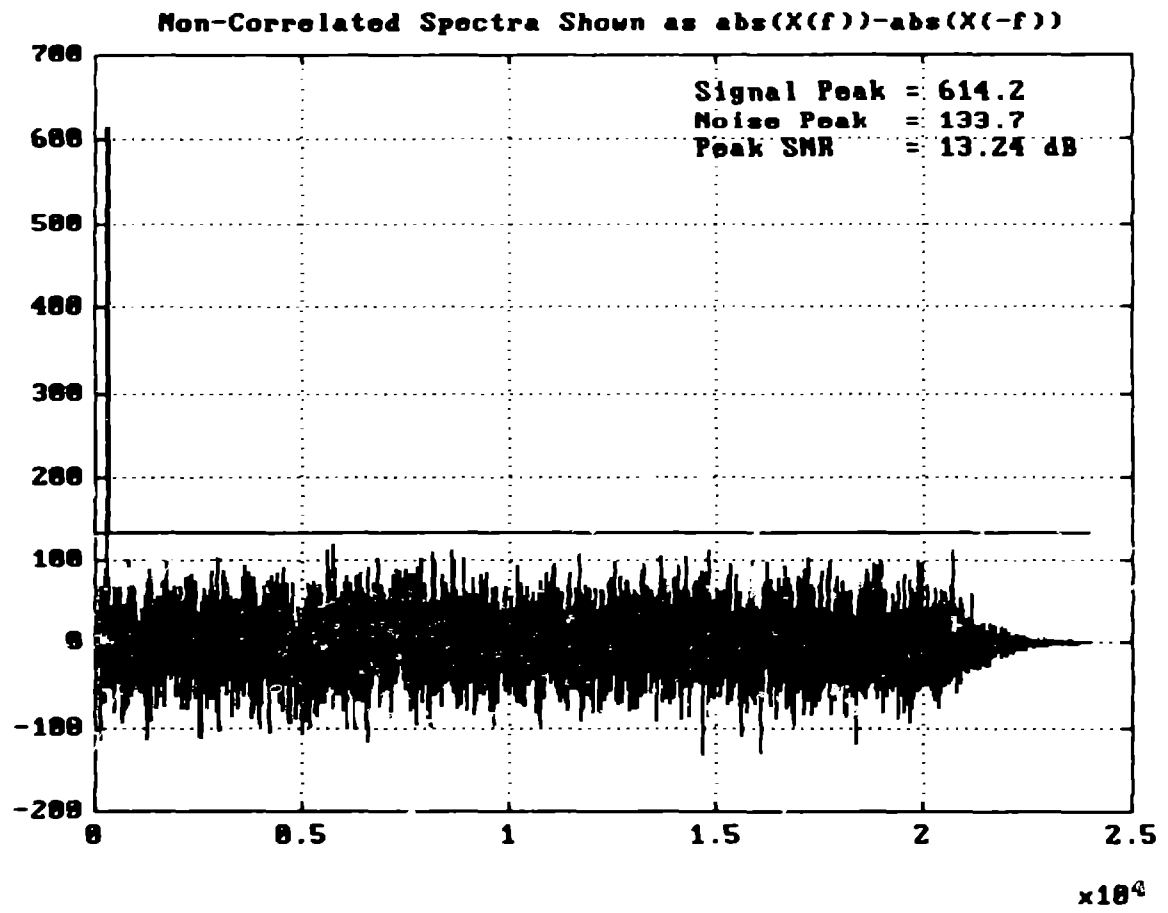


Figure A7. I-I-only F(K) (-16 dB case).

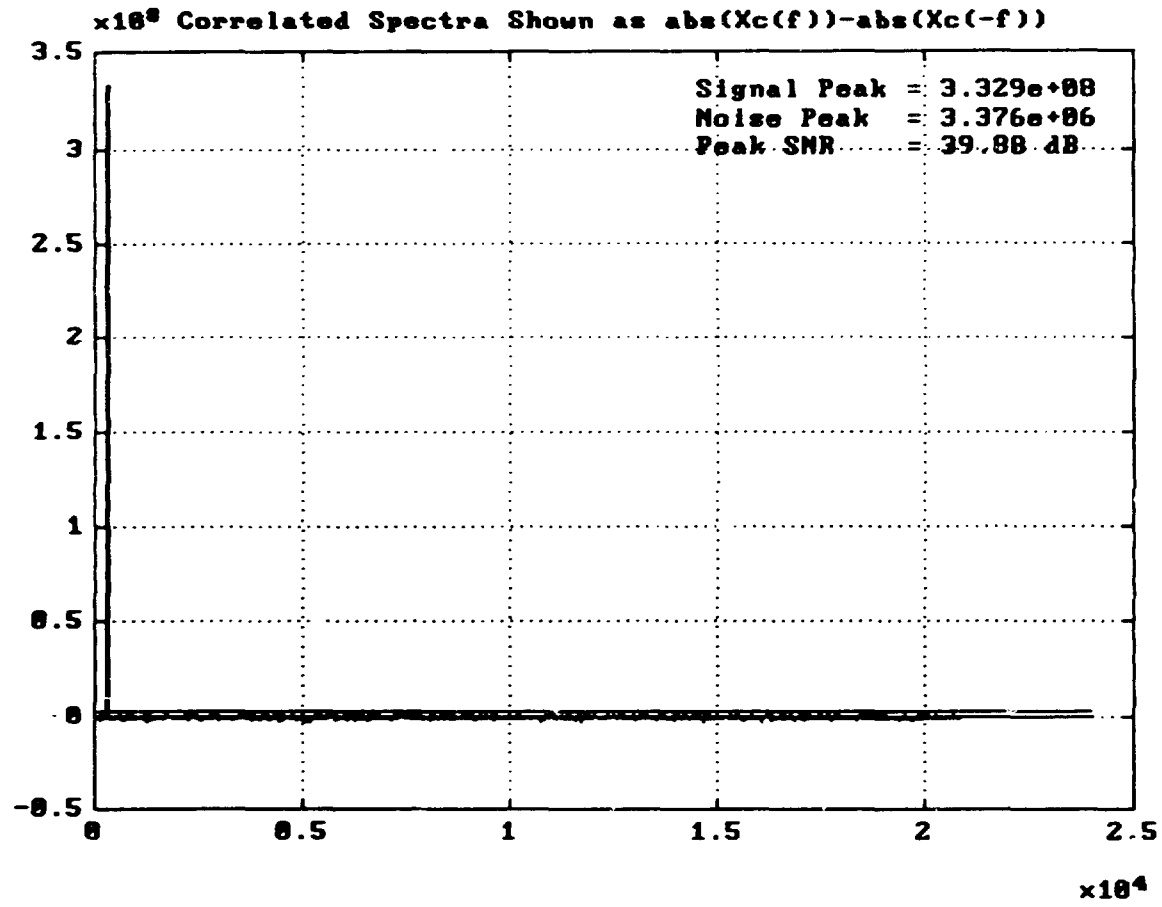
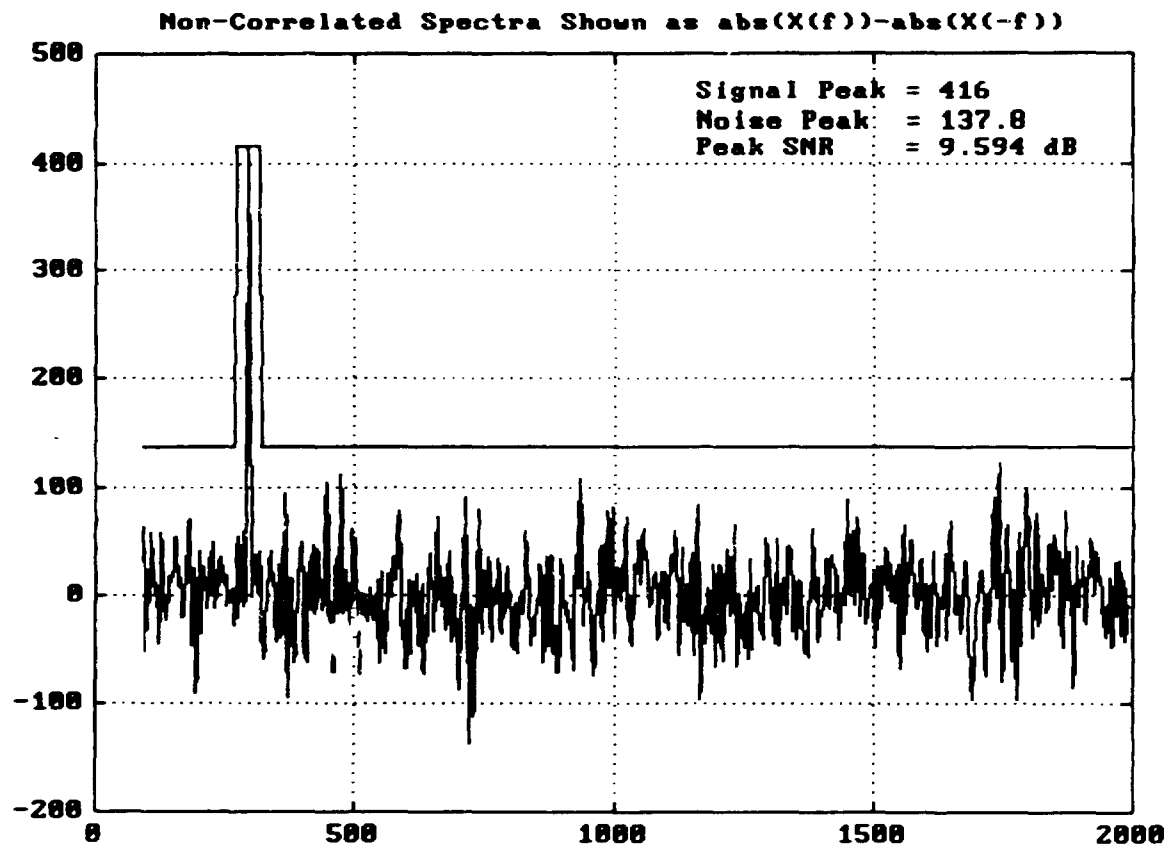
Figure A.8. Correlated (Enhanced)  $F_c(k)$  (-16 dB case).

Figure A9. I:T:1-only T(k) (-20 dB case).





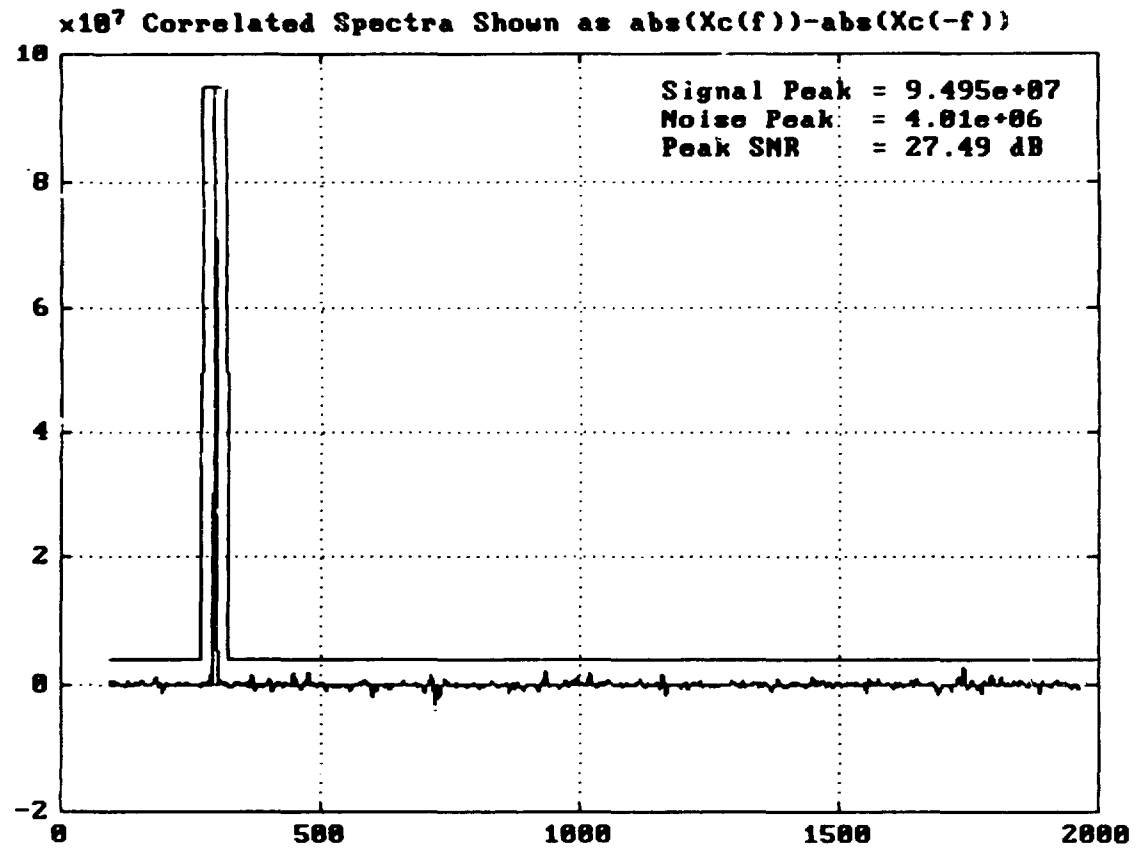
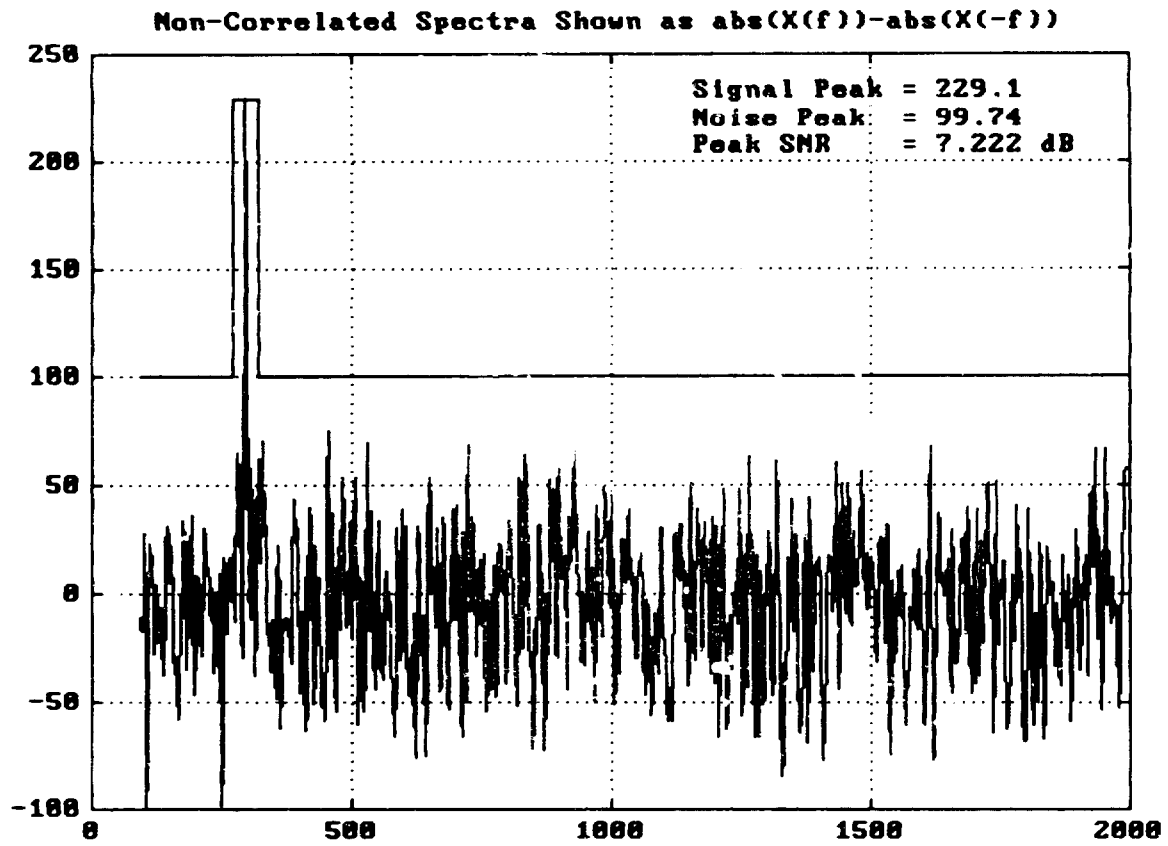


Figure A10. Correlated (Enhanced)  $\Gamma_c(k)$  (-20 dB case).

Figure A.11. FFT-only  $F(k)$  (-23 dB case).

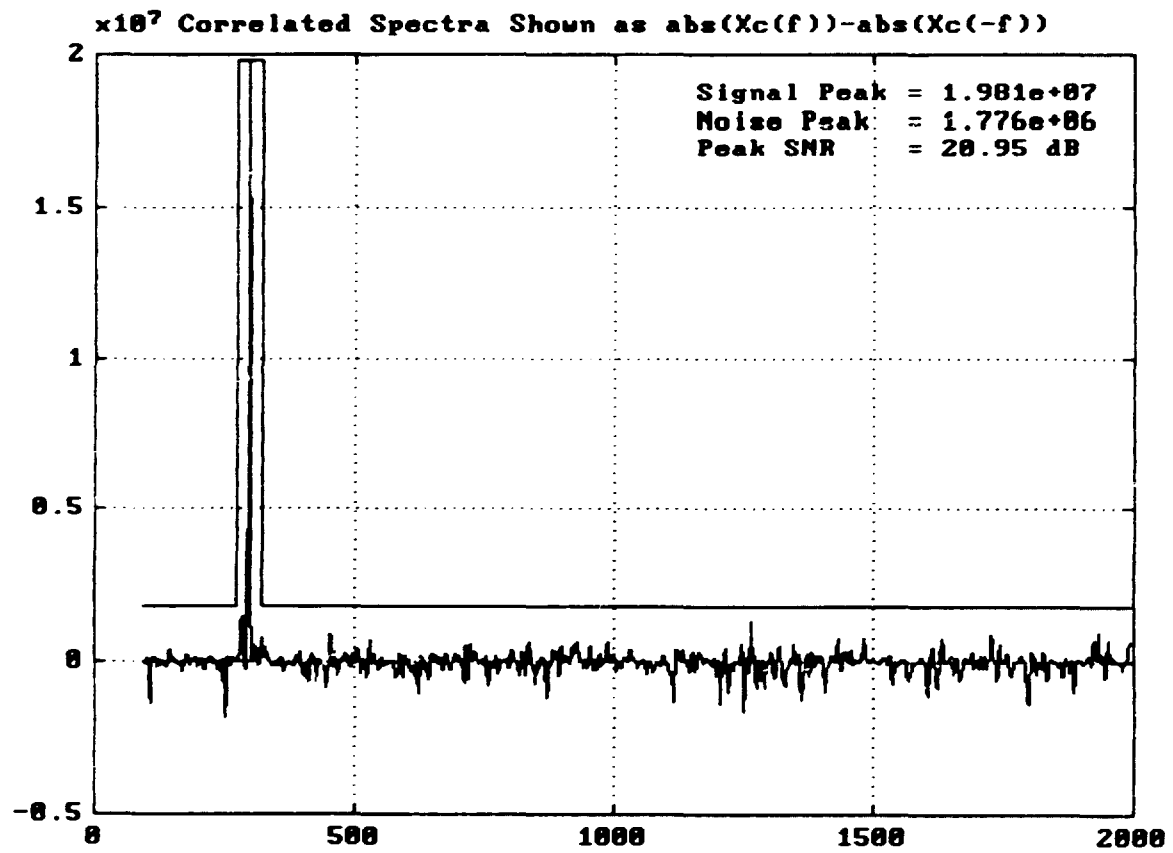
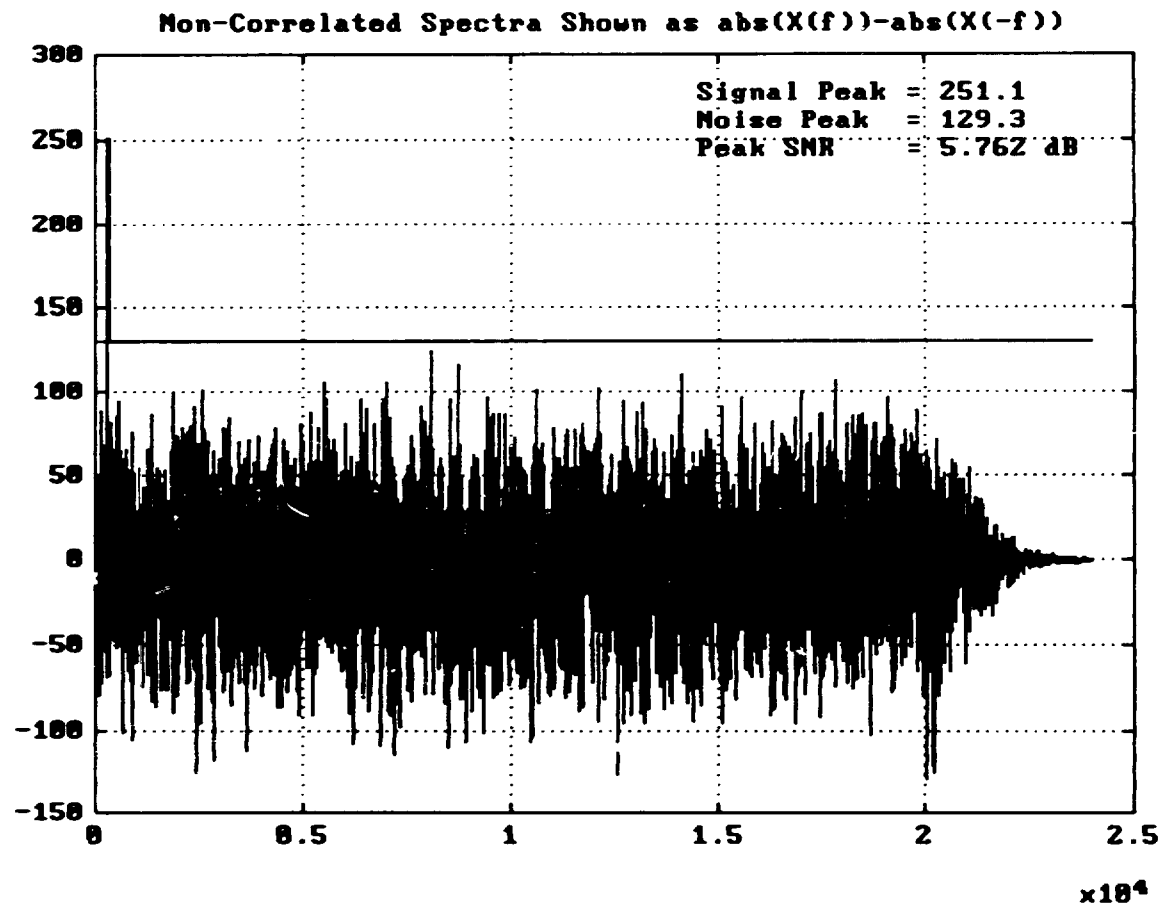


Figure A12. Correlated (Enhanced)  $T_c(k)$  (-23 dB case).

Figure A.13. H-F-only T(K) (-24 dB case).



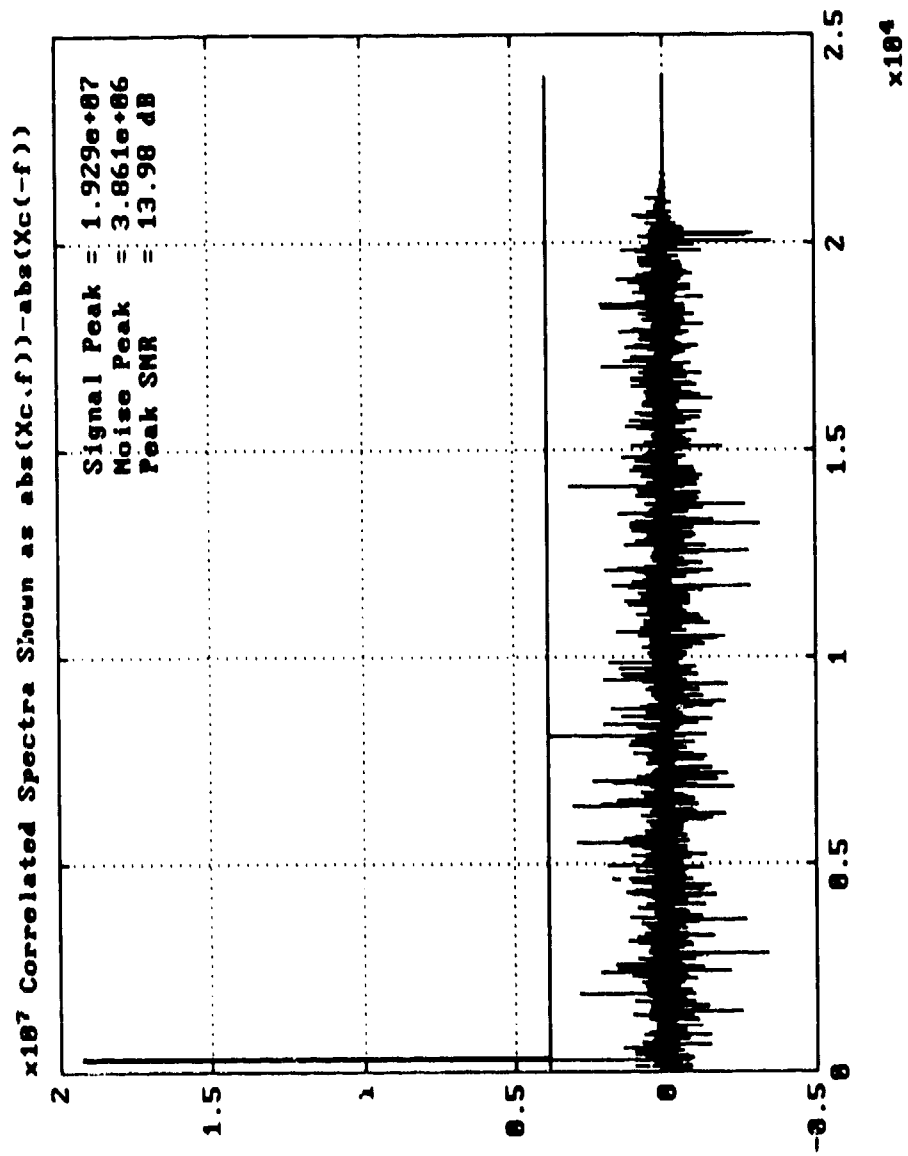


Figure A14. Correlated (Enhanced)  $\Gamma_C(k)$  (-24 dB case).

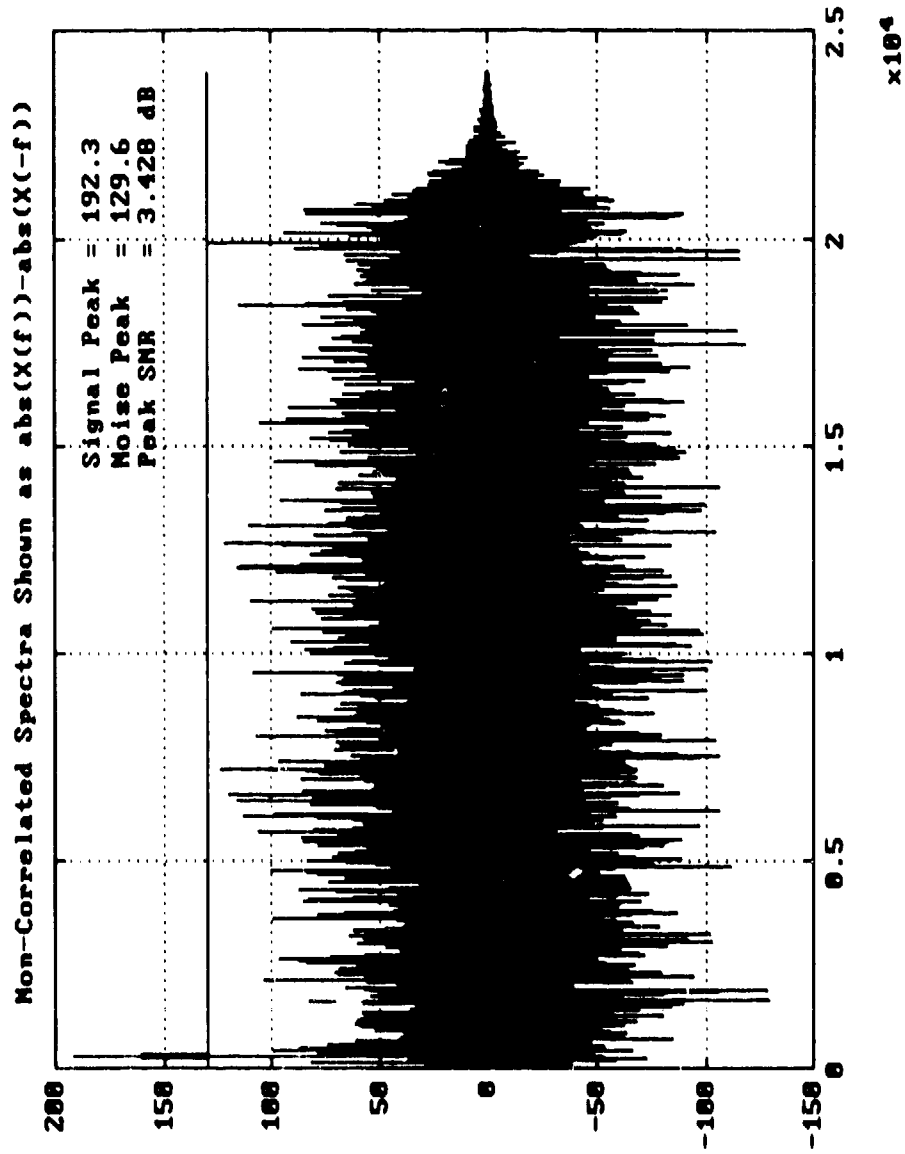


Figure A15. FFT-only  $\Gamma(k)$  (-25 dB case).

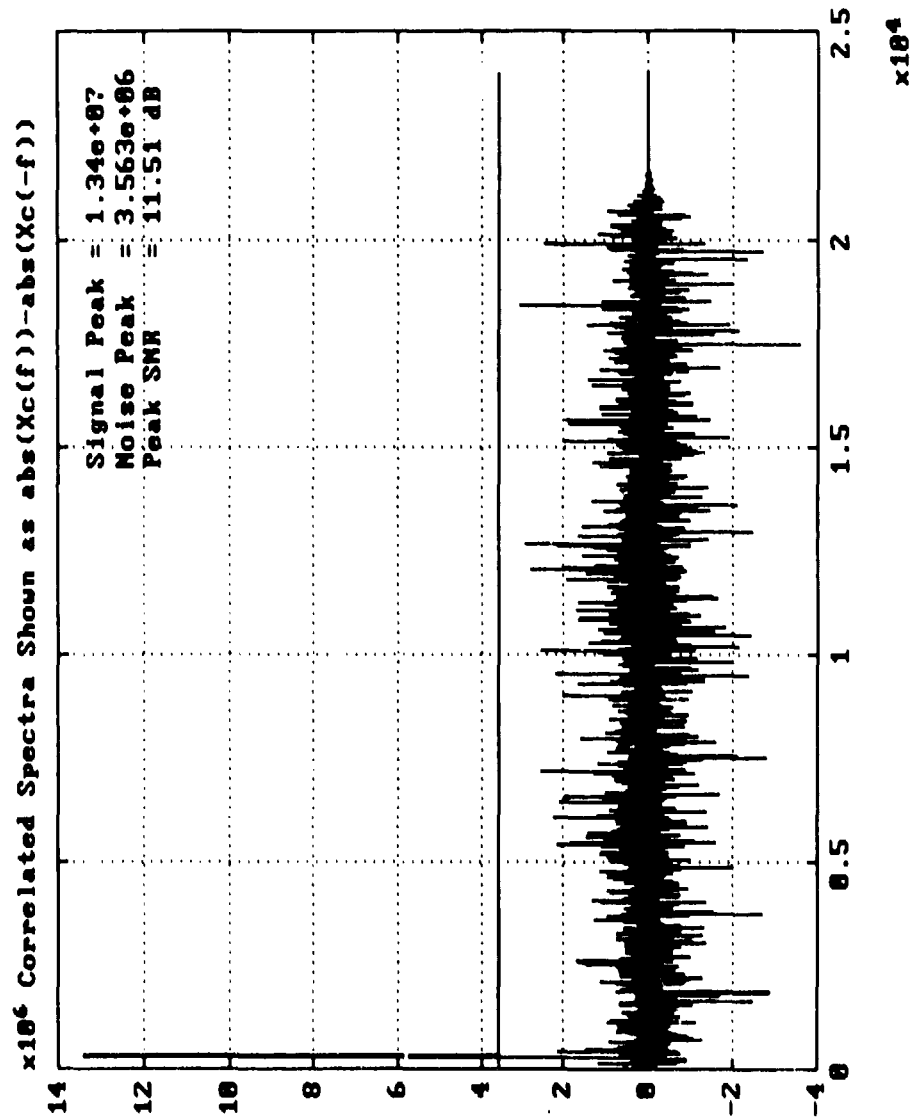


Figure A16. Correlated (Enhanced)  $\Gamma_C(k)$  (-25 dB case).

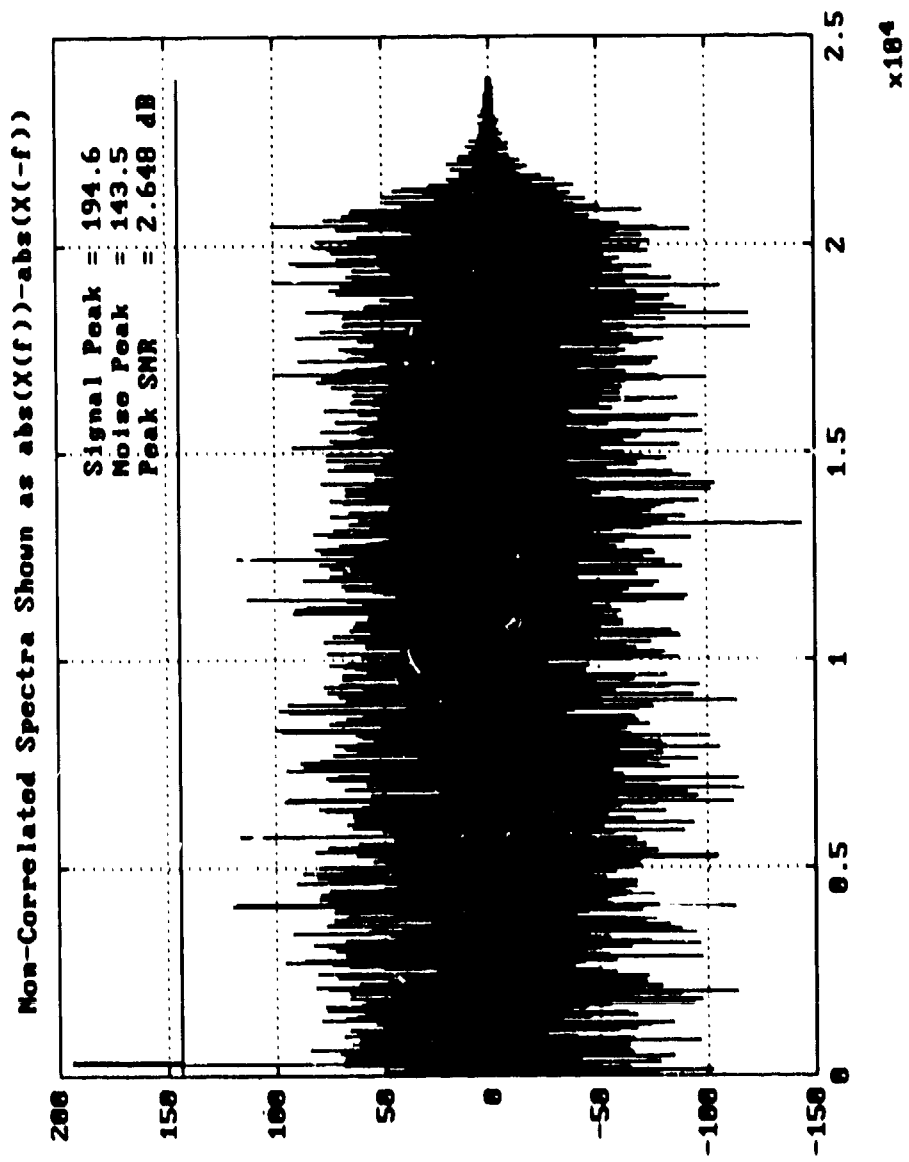


Figure A17. 1F1-only  $\Gamma(k)$  (-26 dB case).



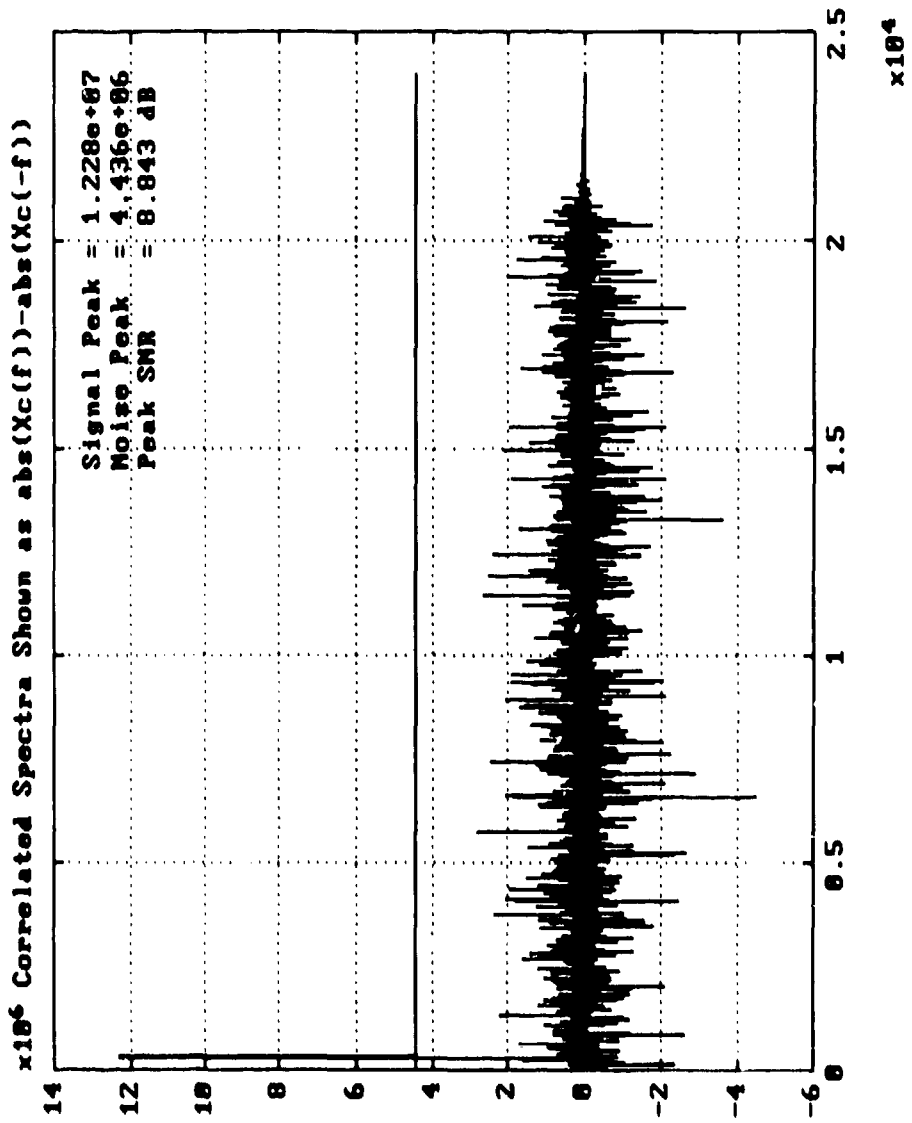


Figure A18. Correlated (Enhanced)  $\Gamma_C(k)$  (-26 dB case).

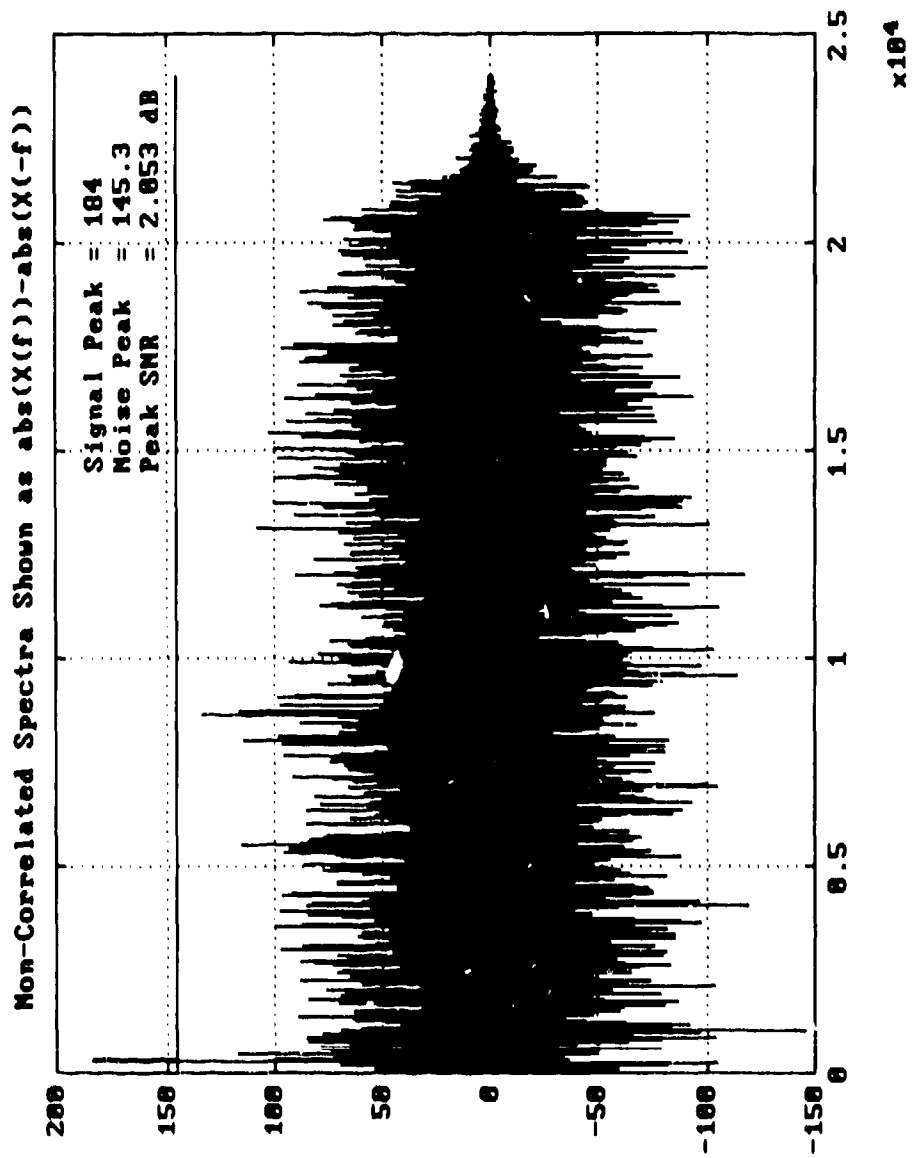


Figure A19. FFT-only  $\Gamma(k)$  (-27 dB case).

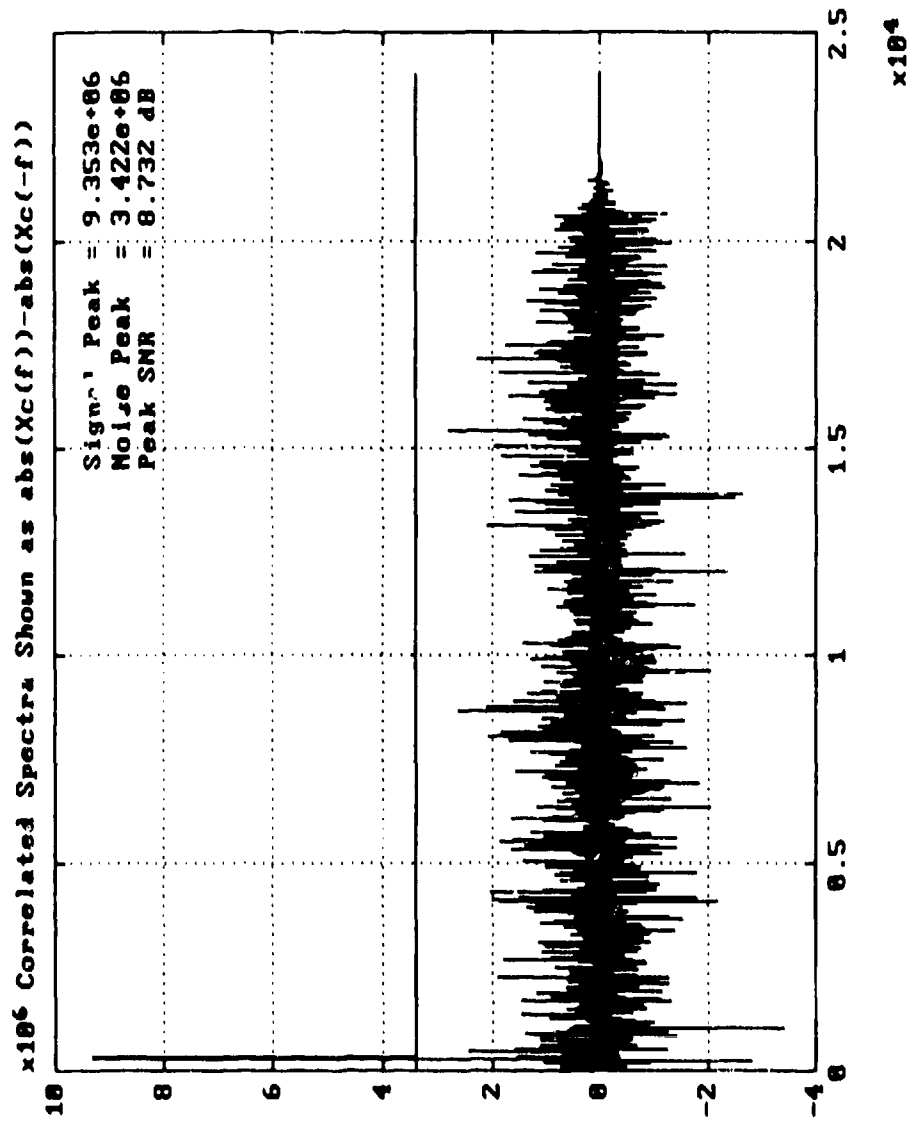


Figure A20. Correlated (Enhanced)  $\Gamma_C(k)$  (-27 dB case).

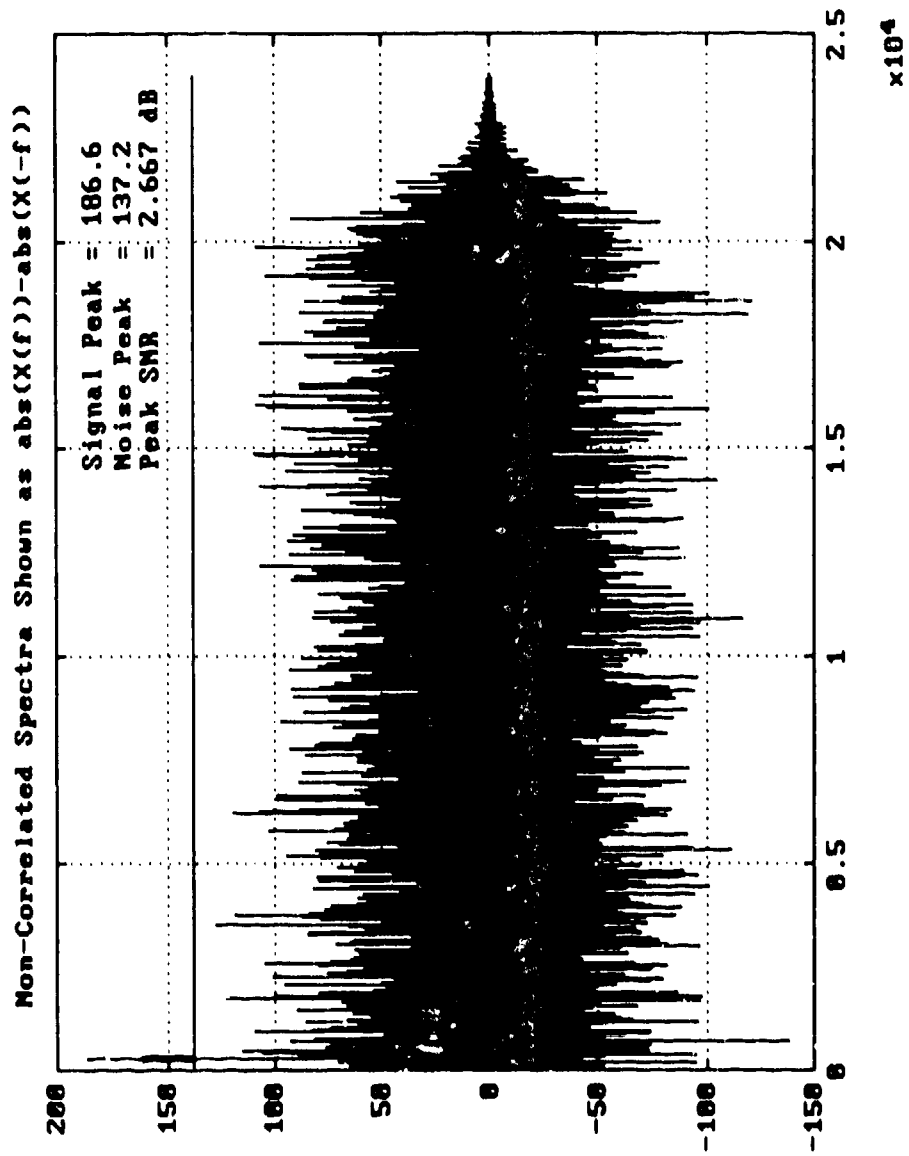


Figure A21. FFT-only  $\Gamma(k)$  (-28 dB case).

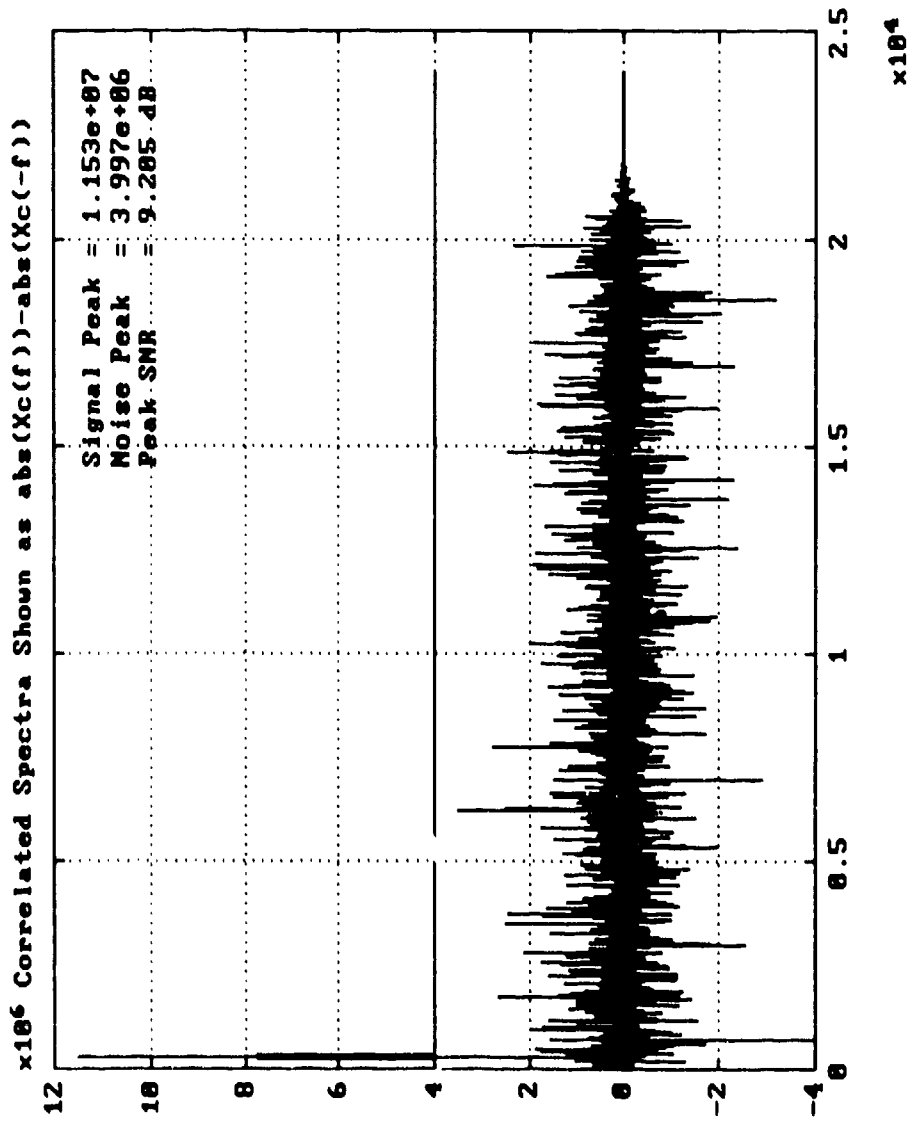


Figure A22. Correlated (Enhanced)  $\Gamma_C(k)$  (-28 dB case).

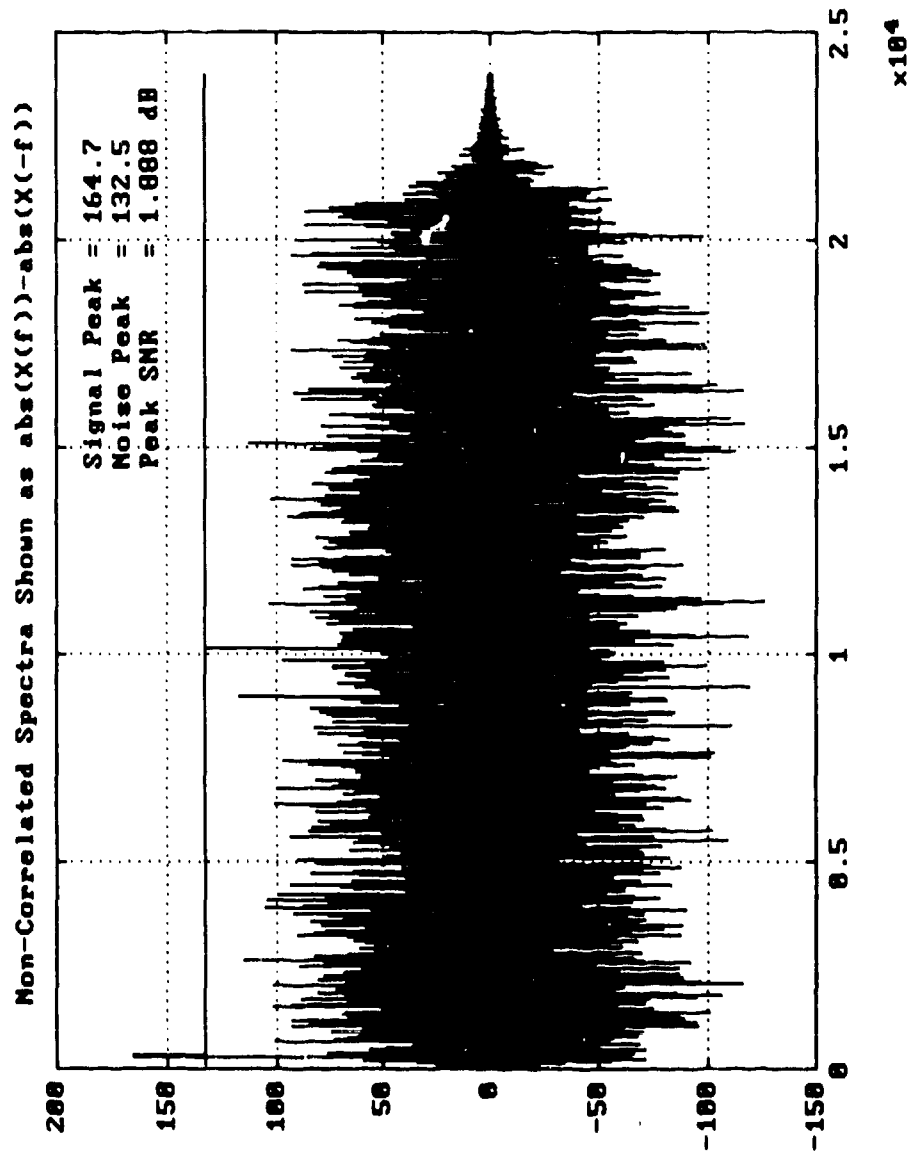


Figure A23. FFT-only  $\Gamma(k)$  (-29 dB case).

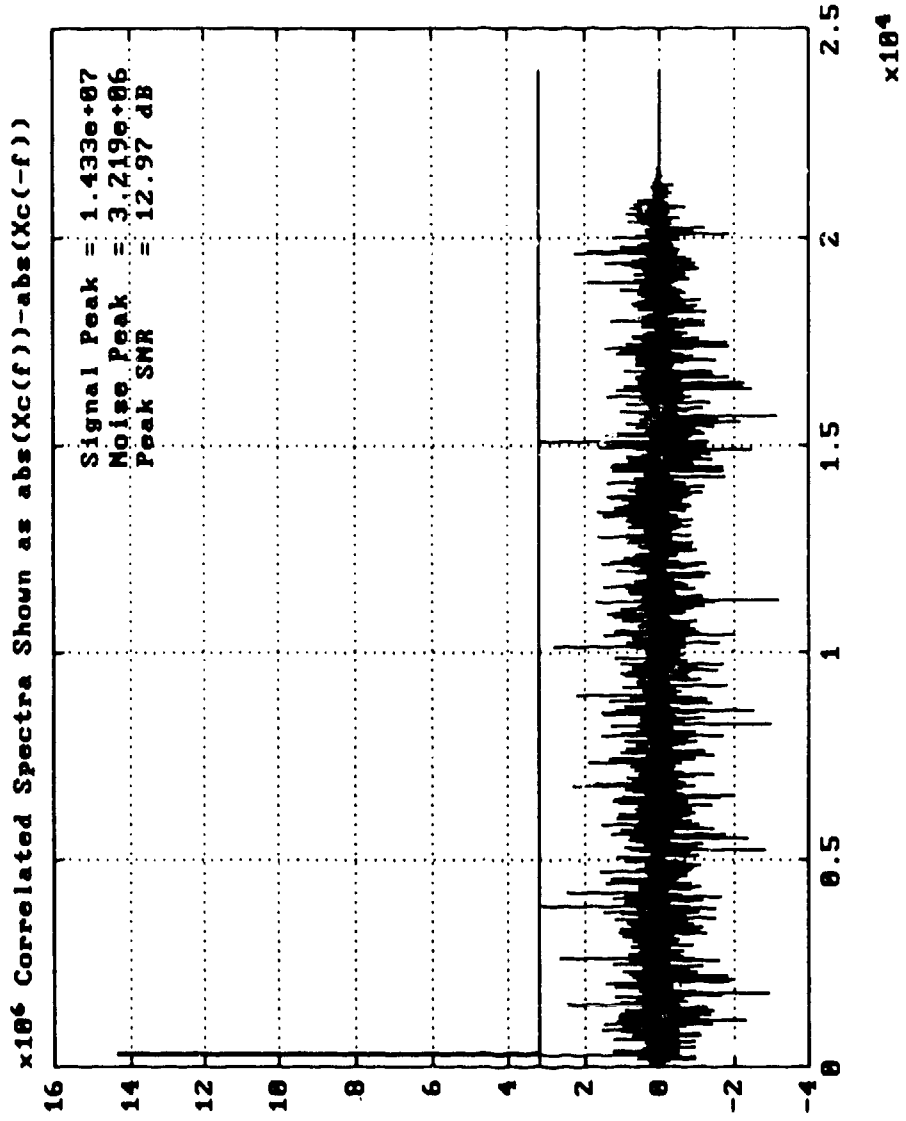


Figure A24. Correlated (Enhanced)  $\Gamma_C(k)$  (-29 dB case).

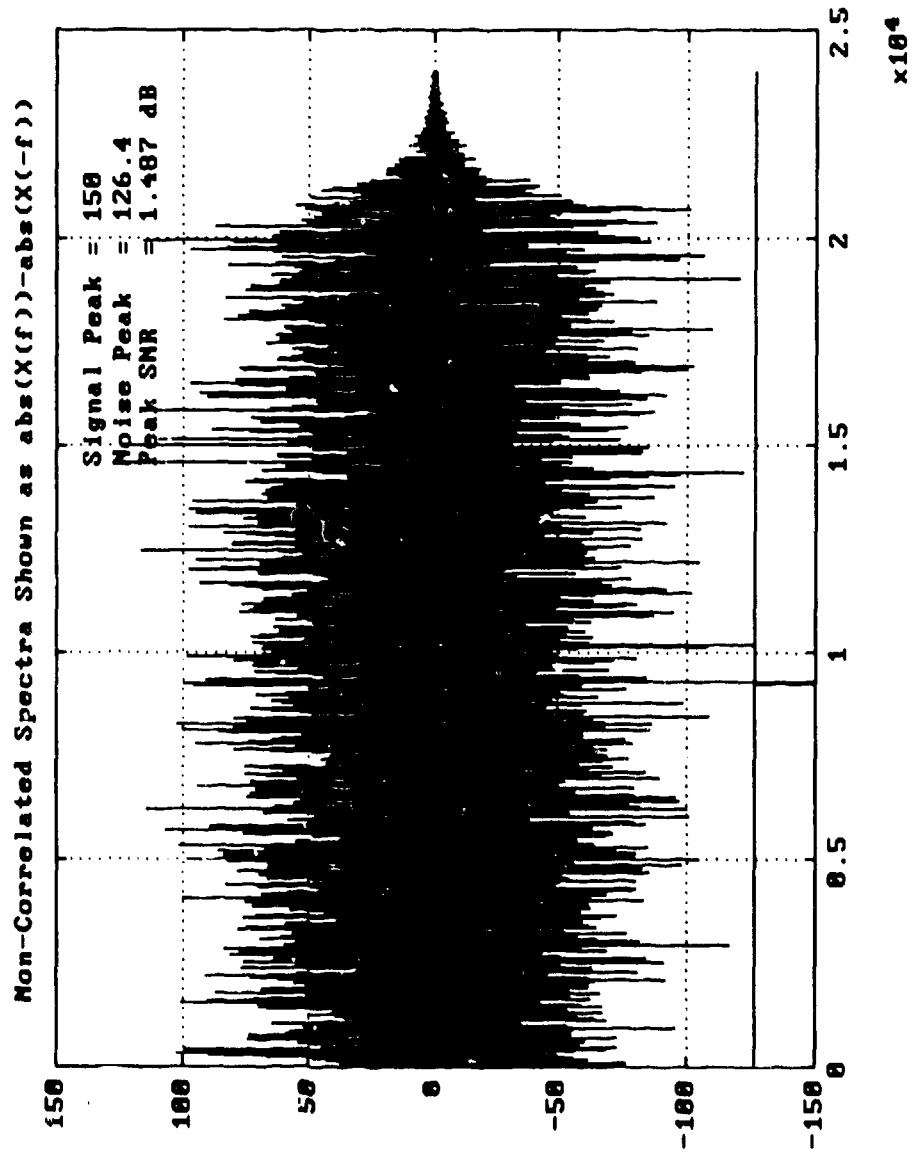


Figure A25. FFT-only  $\Gamma(k)$  (-30 dB case).



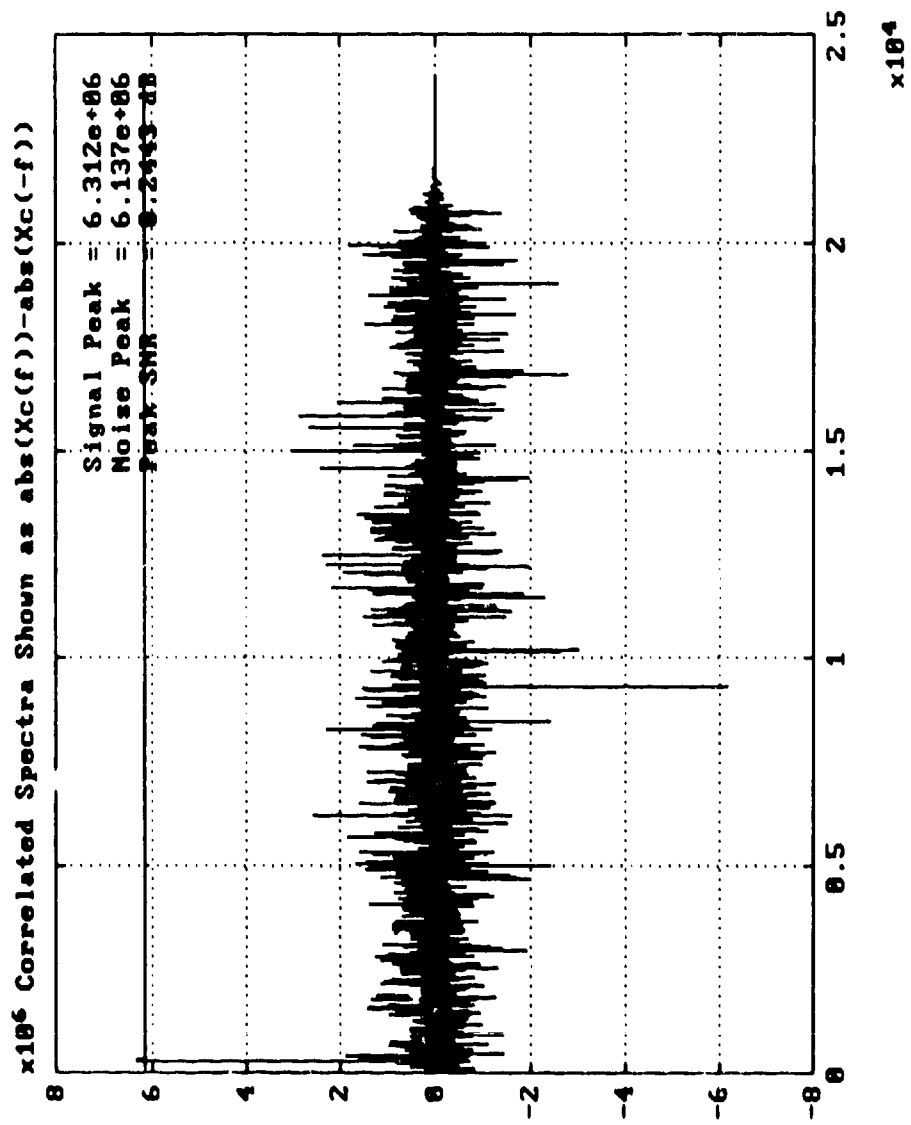


Figure A26. Correlated (Enhanced)  $\Gamma_C(k)$  (-30 dB case).

## Appendix B

### SAMPLE M-FILE

The following M-file was developed to demonstrate the enhancement technique and to compare it to the FFT-only technique. The plot outputs are self-explanatory.

```
function sig_det(radar_sig,lwr_f,upr_f)
% SIG_DET(RADAR_SIG,LWR_F,UPR_F)
% SIG_DET is a function designed detect a doppler target
% in a digitized signal stream. RADAR_SIG is an 8192x2
% data array. LWR_F is the beginning freq (Hz) for (a-b) display and
% UPR_F is the upper freq in Hz.
% This routine forms a complex signal in the form:
%   x=x_in_phase + j * x_quadrature_phase.
% Next, the routine calculates the first differential and the
% first integral of the sequence. The next step is to calculate
% Cxx'(int) (f) = [ X * conj(X') ] * conj(X_integrated).
% Finally, the difference between the positive and negative halves
% of the frequency spectrum are obtained and the results plotted.
% above operations.

a=1:8192;a=a-a;
samp_lwr=fix(lwr_f*8192*4/50000);
samp_upr=fix(upr_f*8192*4/50000);
f=1:8192*4;f=-25006.1035156+f*(50000/(8192*4));
t=1:8192;t=t*(1/50000);
%           w=hanning(8192)';
%           w=[w a a a];
% The above may be used to window the data if desired. Use the
% remarked statements about ten lines down, if applying the window.

cntr=(upr_f-lwr_f)/2+lwr_f;
bw=(upr_f-lwr_f)/2;

x_i=radar_sig(1:8192,1);x_i=x_i-mean(x_i);
radar_sig(:,1)=radar_sig(:,1)-mean(radar_sig(:,1));
x_q=radar_sig(1:8192,2);x_q=x_q-mean(x_q);
radar_sig(:,2)=radar_sig(:,2)-mean(radar_sig(:,2));
x_i_mean=mean(abs(x_i));
x_q_mean=mean(abs(x_q));
x_q=x_q*(x_i_mean/x_q_mean);
```

```

x_i=[x_i a a a];

x_q=[x_q a a a];

x_i=x_i-mean(x_i);x_q=x_q-mean(x_q);
%           x_i=x_i.*w;
%           x_q=x_q.*w;

% The following 7 lines are the key to the technique.

x=x_i+i*x_q;
y=[0 diff(x)];
z=cumsum(x);
y=y-mean(y);
z=z-mean(z);
xf=fft(x);yf=fft(y);zf=fft(z);
xyz=abs((xf.*conj(yf)).*conj(zf));

clg

x_fm=mean(abs(xf));xyzm=mean(xyz);
a=abs(fftshift(xf));b=fftshift(xyz);
a(16383)=x_fm;a(16384)=x_fm;a(16385)=x_fm;
b(16383)=xyzm;b(16384)=xyzm;b(16385)=xyzm;
clg;
subplot(211),plot(t,radar_sig(:,1));
title('Time Domain I & Q Data')
xlabel('In-Phase Data');
subplot(212),plot(t,radar_sig(:,2)*(x_i_mean/x_q_mean));
xlabel('Quadrature Data');

clg
subplot(211),plot((16384-samp_upr:16384-samp_lwr),a(16384-samp_upr:16384-samp_lwr));
title('Complex FFT of I + IQ Time Data')
grid;
xlabel('X(-f)');
subplot(212),plot((16384+samp_lwr:16384+samp_upr),a(16384+samp_lwr:16384+samp_upr));
xlabel('X(f)');
grid;

clg
subplot(211),plot((16384-samp_upr:16384-samp_lwr),b(16384-samp_upr:16384-samp_lwr));
title('Correlated Frequency Spectra')
grid;
xlabel('X(-f)');
subplot(212),plot((16384+samp_lwr:16384+samp_upr),b(16384+samp_lwr:16384+samp_upr));
xlabel('X(f)');
grid;

a=abs(xf(2:16384));
b=abs(xf(32768:-1:16386));
c=a-b;
clg;

```

```

[s_peak,s_pos]=max(abs(c(samp_lwr-2:samp_upr-2)));

if c(samp_lwr-2+s_pos)<0
    pk_pol=-1;
else
    pk_pol=1;
end

n_peak1=max(abs(c(samp_lwr-2:samp_lwr-2+s_pos-15)));

n_peak2=max(abs(c((samp_lwr-2+s_pos+15):samp_upr-2)));

nn=[n_peak1 n_peak2];n_peak=max(nn);

snr_curve=1:(samp_upr-samp_lwr+1);
snr_curve=snr_curve./snr_curve*n_peak*pk_pol;
s_peaka=1:31;s_peaka=s_peaka./s_peaka*s_peak*pk_pol;
snr_curve(s_pos-15:s_pos+15)=s_peaka;

plot(f(16384+samp_lwr:16384+samp_upr),c(samp_lwr-2:samp_upr-
2),f(16384+samp_lwr:16384+samp_upr),snr_curve);
grid;
title('Non-Correlated Spectra Shown as abs(X(f))-abs(X(-f)) )
peak_snr=20*log10(s_peak/n_peak);
text(.60,.88,['Signal Peak = ',num2str(s_peak)],'sc')
text(.60,.85,['Noise Peak = ',num2str(n_peak)],'sc')
text(.60,.82,['Peak SNR = ',num2str(peak_snr),' dB'],'sc')

a=abs(xyz(2:16384));
b=abs(xyz(32768:-1:16386));
c=a-b;
clg;

[s_peak,s_pos]=max(abs(c(samp_lwr-2:samp_upr-2)));

if c(samp_lwr-2+s_pos)<0
    pk_pol=-1;
else
    pk_pol=1;
end

n_peak1=max(abs(c(samp_lwr-2:samp_lwr-2+s_pos-15)));

n_peak2=max(abs(c((samp_lwr-2+s_pos+15):samp_upr-2)));

nn=[n_peak1 n_peak2];n_peak=max(nn);

snr_curve=1:(samp_upr-samp_lwr+1);

```

```

snr_curve=1:(samp_upr-samp_lwr+1);
snr_curve=snr_curve./snr_curve*n_peak*pk_pol;
s_peaka=1:31;s_peaka=s_peaka./s_peaka*s_peak*pk_pol;
snr_curve(s_pos-15:s_pos+15)=s_peaka;

plot(f(16384+samp_lwr:16384+samp_upr),c(samp_lwr-2:samp_upr-
2),f(16384+samp_lwr:16384+samp_upr),snr_curve);
grid;
title('Correlated Spectra Shown as abs(Xc(f))-abs(Xc(-f))')

peak_snr=20*log10(s_peak/n_peak);
text(.60,.88,['Signal Peak = ',num2str(s_peak)],'sc')
text(.60,.85,['Noise Peak = ',num2str(n_peak)],'sc')
text(.60,.82,['Peak SNR = ',num2str(peak_snr),' dB'],'sc')

end

```