The resulting approximation for X_∞ is then as given in (9). We use this as our starting function and adjust for finite values of K. We first look at the error curves versus P_d for the $K = 1$ case. The curves are smooth from $0.05 \leq P_d \leq 0.872$ so we fit a curve for $N = 10$ and $P_{fa} = 10^{-6}$, in the midrange of the parameter set. The resulting errors are small for $P_d \leq 0.872$ but grow exponentially for P_d above 0.872. We design an additional exponential curve fit for $0.872 \leq P_d \leq 0.99$. The behavior of these curves should vary approximately as the inverse of K and so K divides these equations. These approximations work well over a wide variety of P_{fa} but, to extend the range, we need to add an additional term sensitive to P_{fa}, for $0.872 \leq P_d \leq 0.99$. The final result is given in the equation (10) above and, for the extension, in (11).

DAVID A. SHNIDMAN
Raytheon Company
Bedford, MA 01730
E-mail: (shnidman@alum.mit.edu)
(Now with RADET Consulting, 11 Somerset Rd., Lexington, MA 02420-3519)

REFERENCES

[1] Marcum, J. I. (1960)
A statistical theory of target detection by pulsed radar.
IRE Transactions on Information Theory, **IT-6** (Apr. 1960), 59–267.

[2] Swerling, P. (1960)
Probability of detection for fluctuating targets.
IRE Transactions on Information Theory, **IT-6** (Apr. 1960), 269–308.

[3] Swerling, P. (1965)
More on detection of fluctuating targets.
IRE Transactions on Information Theory, **IT-11** (July 1965), 459–460.

[4] Swerling, P. (1970)
Recent developments in target models for radar detection analysis.
AGARD Avionics Tech. Symposium Proceedings, Istanbul, Turkey, May 25–29, 1970.

[5] Mitchell, R. L., and Walker, J. F. (1971)
Recursive methods of computing detection probabilities.
IEEE Transactions on Aerospace and Electronic Systems, **AES-7** (July 1971), 671–676.

[6] Shnidman, D. A. (1995)
Radar detection probabilities and their calculations.
IEEE Transactions on Aerospace and Electronic Systems, **31** (July 1995), 928–950.

[7] Mills, R. F., and Prescott, G. E. (1996)
A comparison of various radiometer detection models.
IEEE Transactions on Aerospace and Electronic Systems, **32** (Jan. 1996), 467–473.

[8] Neuvy, J. (1970)
An aspect of determining the range of radar detection.
IEEE Transactions on Aerospace and Electronic Systems, **AES-6** (July 1970), 514–521.

[9] Albersheim, W. J. (1981)
A closed-form approximation to Robertson's detection characteristics.
Proceedings of the IEEE, **69** (July 1981), 839.

[10] Barton, D. K. (1988)
Modern Radar System Analysis.
Boston, MA: Artech House, 1988, 68.

[11] North, D. O. (1963)
An analysis of the factors which determine signal/noise discrimination in pulsed carrier systems.
Proceedings of the IEEE, **51**, 7 (July 1963), 1015–1027.

[12] Sankaran, M. (1963)
Approximations to the non-central chi-square distribution.
Biometrica, **50** (1963), 199–204.

[13] Widder, D. V. (1961)
Advanced Calculus (2nd ed.).
Englewood Cliffs, NJ: Prentice-Hall, 1961, 371.

MATLAB Code for Plotting Ambiguity Functions

A MATLAB code capable of plotting ambiguity functions of many different radar signals is presented. The program makes use of MATLAB's sparse matrix operations, and avoids loops. The program could be useful as a pedagogical tool in radar courses teaching pulse compression.

The ambiguity function (AF) is the matched-filter response to a Doppler-shifted version of the signal it is matched to [1, Secs. 3.4 and 5.1]. The version calculated by our code is

$$|\chi(\tau,\nu)| = \left| \int_{-\infty}^{\infty} u(t)u^*(t-\tau)\exp(j2\pi\nu t)dt \right| \quad (1)$$

where $u(t)$ is the complex envelope of the radar signal.

The code allows inputting many different signals and provides control over many plot parameters. The program allows over-sampling of the signal with much finer resolution than needed for calculating the delayed signal. This makes it possible to compute a diluted picture (fewer delay-Doppler grid points) of the ambiguity function with low computational effort, while sampling the signal with a sufficiently large sampling rate.

Table I suggests input parameters for twelve different ambiguity plots of various signals. We also present two examples of AFs plotted using the code. Fig. 1 (corresponding to line 2 in Table I) presents the AF of a single linear-FM pulse [1, Sec. 7.2]. Fig. 2 (corresponding to line 12 in Table I) presents the AF of a coherent train of 8 stepped frequency pulses [1, Sec. 8.6] with additional interpulse amplitude weighting.

Manuscript received November 26, 2001; revised March 28, 2002; released for publication May 6, 2002.

IEEE Log No. T-AES/38/3/06450.

Refereeing of this contribution was handled by L. M. Kaplan.

TABLE 1
Recommended Parameters

Signal	Fig.	Signal code	Freq. Code	Fmax *Mtb	K Dop	τmax/ Mtb	N delay	Over sample
Pulse		1	none	4	60	1.1	60	10
LFM	1	ones(1,51)	.0031*(-25:25)	6	60	1.1	60	10
Weighted LFM		sqrt(chebwin(51,50))'	.0031*(-25:25)	6	60	1.1	60	10
Costas 7		ones(1,7)	[4 7 1 6 5 2 3]	12	60	1.1	80	20
Barker 13		[1 1 1 1 1 -1 -1 1 1 -1 1 -1 1]	none	10	60	1.1	60	10
Frank 16		[1 1 1 1 1 j -1 -j 1 -1 1 -1 1 -j -1 j]	none	10	60	1.1	60	10
P4 25		exp(j*pi*(1/25*(0:24).^2-(0:24)))	none	15	80	1.1	80	20
Complementary pair		[1 1 -1 0 0 0 0 0 0 0 1 j 1]	none	10	60	1.1	60	10
Pulse train		[1 0 0 0 0 1 0 0 0 0 1 0 0 0 0 1 0 0 0 1 0 0 0 0 1]	none	15	80	1.05	100	10
Pulse train		[1 0 0 0 0 1 0 0 0 0 1 0 0 0 0 1 0 0 0 1 0 0 0 0 1]	none	12	80	.042	60	10
Stepped freq. pulse train		[1 0 0 0 0 1 0 0 0 0 1 0 0 0 0 1 0 0 0 1 0 0 0 0 1]	.78*[0 0 0 0 0 1 0 0 0 0 2 0 0 0 0 3 0 0 0 0 4 0 0 0 0 5]	12	80	.042	60	10
Weighted Stepped freq. pulse train	2	sqrt(chebwin(36,50))'.*[1 0 0 0 0 1 0 0 0 0 1 0 0 0 0 1 0 0 0 1 0 0 0 0 1 0 0 0 0 1]	.7*[0 0 0 0 0 1 0 0 0 0 2 0 0 0 0 3 0 0 0 0 4 0 0 0 0 5 0 0 0 0 6 0 0 0 0 7]	16	70	.03	60	5

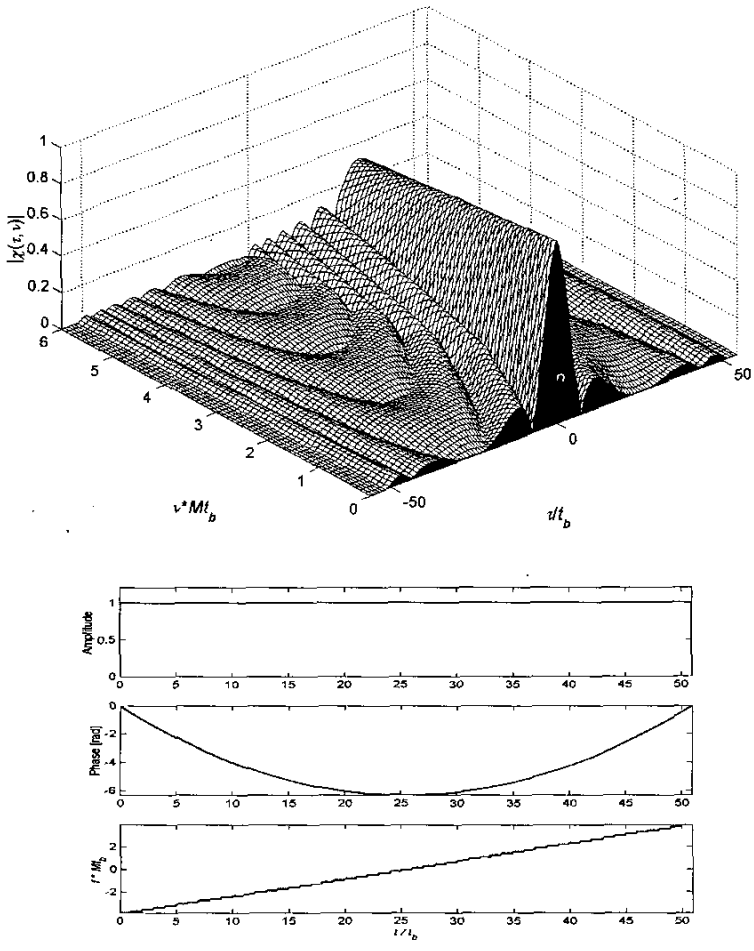

Fig. 1. Linear FM.

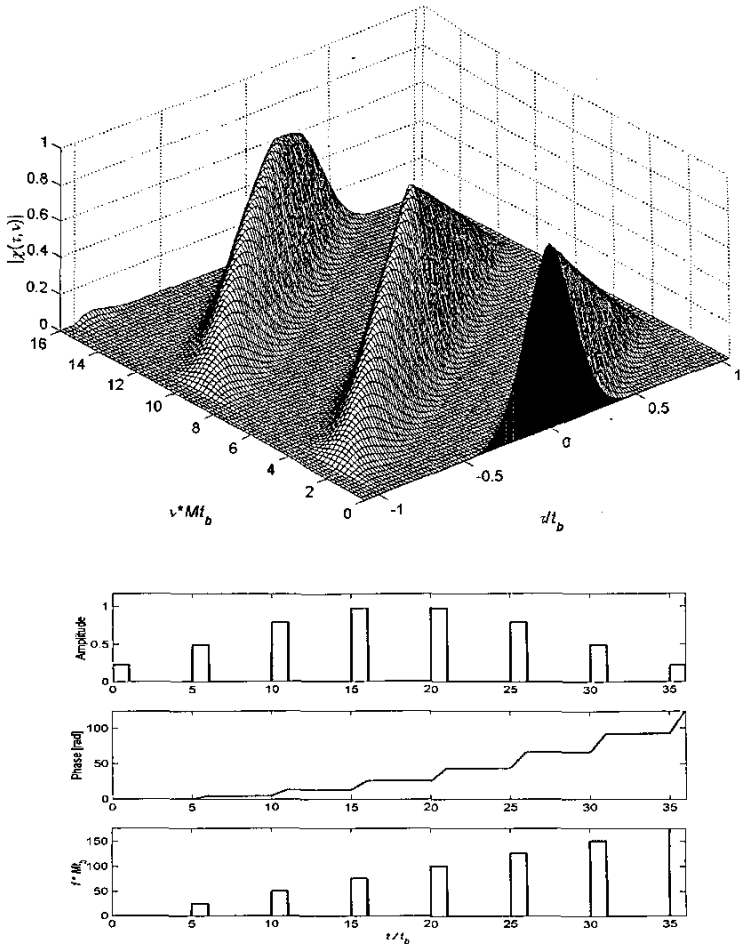

Fig. 2. Weighted stepped-frequency pulse train.

Because of the symmetry of the AF with respect to the origin, we found it sufficient to plot only two of the four quadrants. This provided an opportunity to display the zero-Doppler cut of the AF, which is the magnitude of the autocorrelation function. We preferred to plot $|\chi(\tau,\nu)|$ rather than $|\chi(\tau,\nu)|^2$, in order to emphasize sidelobes without resorting to log-scale. The code could be easily modified to produce a log-scale plot.

The program also produces a second figure with subplots of three characteristics of the signal: amplitude, phase and frequency.

For the sake of brevity the MATLAB m-file, listed in Table II, appears without comments. A version containing comments can be found on the Internet at: http://www.eng.tau.ac.il/~nadav/amb-func.html.

In that location one can also find a more elaborate MATLAB program for plotting AFs. This second program utilizes a graphic user interface (GUI). The GUI offers many preset signals, with an option to modify them. The user can define a new signal and store it. The GUI also makes it easy to modify the plot parameters.

ELI MOZESON
NADAV LEVANON
Dept. of Electrical Engineering–Systems
Tel Aviv University
P.O. Box 39040
Tel Aviv, 69978
Israel

REFERENCES

[1] Rihaczek, A. W. (1969)
Principles of High Resolution Radar.
New York: McGraw-Hill, 1969.

TABLE 2
MATLAB Code

```
% ambfn1.m plots the ambiguity function of a signal u_basic (row vector)
% The ambiguity function is defined as:
% a(t,f) = abs ( sumi( u(k)*u'(i-t)*exp(j*2*pi*f*i) ) )
% The user is prompted for the signal data:
% u_basic is a row complex vector representing amplitude and phase
% f_basic is a corresponding frequency coding sequence
% The duration of each element is tb (total duration of the signal is tb*(m_basic-1))
% F is the maximal Doppler shift
% T is the maximal Delay
% K is the number of positive Doppler shifts (grid points)
% N is the number of delay shifts on each side (for a total of 2N+1 points)
% The code allows r samples within each bit
% Written by Eli Mozeson and Nadav Levanon, Dept. of EE-Systems, Tel Aviv University
% clear all

u_basic=input(' Signal elements (row complex vector, each element last tb sec) = ? ');
m_basic=length(u_basic);
fcode=input(' Allow frequency coding (yes=1, no=0) = ? ');
if fcode==1
    f_basic=input(' Frequency coding in units of 1/tb (row vector of same length) = ? ');
end
F=input(' Maximal Doppler shift for ambiguity plot [in units of 1/Mtb] (e.g., 1)= ? ');
K=input(' Number of Doppler grid points for calculation (e.g., 100) = ? ');
df=F/K/m_basic;
T=input(' Maximal Delay for ambiguity plot [in units of Mtb] (e.g., 1)= ? ');
N=input(' Number of delay grid points on each side (e.g. 100) = ? ');
sr=input(' Over sampling ratio (>=1) (e.g. 10)= ? ');
r=ceil(sr*(N+1)/T/m_basic);
if r==1
   dt=1;
   m=m_basic;
   uamp=abs(u_basic);
   phas=uamp*0;
   phas=angle(u_basic);
   if fcode==1
      phas=phas+2*pi*cumsum(f_basic);
   end
   uexp=exp(j*phas);
   u=uamp.*uexp;
else                          % i.e., several samples within a bit
   dt=1/r;                   % interval between samples
   ud=diag(u_basic);
   ao=ones(r,m_basic);
   m=m_basic*r;
   u_basic=reshape(ao*ud,1,m);  % u_basic with each element repeated r times
   uamp=abs(u_basic);
   phas=angle(u_basic);
   u=u_basic;
   if fcode==1
       ff=diag(f_basic);
       phas=2*pi*dt*cumsum(reshape(ao*ff,1,m))+phas;
       uexp=exp(j*phas);
       u=uamp.*uexp;
   end
end
t=[0:r*m_basic-1]/r;
tscale1=[0 0:r*m_basic-1 r*m_basic-1]/r;
dphas=[NaN diff(phas)]*r/2/pi;

figure(1), clf, hold off % plot the signal parameters
subplot(3,1,1)
plot(tscale1,[0 abs(uamp) 0],'linewidth',1.5)
ylabel(' Amplitude ')
axis([-inf inf 0 1.2*max(abs(uamp))])
subplot(3,1,2)
plot(t, phas,'linewidth',1.5)
axis([-inf inf -inf inf])
ylabel(' Phase [rad] ')
subplot(3,1,3)
plot(t,dphas*ceil(max(t)),'linewidth',1.5)
axis([-inf inf -inf inf])
xlabel(' \itt / t_b ')
ylabel(' \itf * Mt_b ')
```

(continued)

TABLE 2
MATLAB Code *(continued)*

```
dtau=ceil(T*m)*dt/N;
tau=round([0:1:N]*dtau/dt)*dt;
f=[0:1:K]*df;
f=[-fliplr(f) f];
mat1=spdiags(u',0,m+ceil(T*m),m);
u_padded=[zeros(1,ceil(T*m)),u,zeros(1,ceil(T*m))];
cidx=[1:m+ceil(T*m)];
ridx=round(tau/dt)';
index = cidx(ones(N+1,1),:) + ridx(:,ones(1,m+ceil(T*m)));
mat2 = sparse(u_padded(index));
uu_pos=mat2*mat1;
clear mat2 mat1
e=exp(-j*2*pi*f'*t);
a_pos=abs(e*uu_pos');
a_pos=a_pos/max(max(a_pos));
a=[flipud(conj(a_pos(1:K+1,:))) fliplr(a_pos(K+2:2*K+2,:))];
delay=[-fliplr(tau) tau];
freq=f(K+2:2*K+2)*ceil(max(t));
delay=[delay(1:N) delay(N+2:2*N)];
a=a(:,[1:N,N+2:2*N]);
[amf amt]=size(a);
cm=zeros(64,3);
cm(:,3)=ones(64,1);

figure(2), clf, hold off % Ambiguity function
mesh(delay, [0 freq], [zeros(1,amt);a])
hold on
surface(delay, [0 0], [zeros(1,amt);a(1,:)])
colormap(cm)
view(-40,50)
axis([-inf inf -inf inf 0 1])
xlabel(' {\it\tau}/{\itt_b}','Fontsize',12);
ylabel(' {\it\nu}*{\itMt_b}','Fontsize',12);
zlabel(' |{\it\chi}({\it\tau},{\it\nu})| ','Fontsize',12);
hold off
```

Doppler Compensation For Binary Phase-Coded Waveforms

Codes used in phase-coded pulse compression waveforms suffer from Doppler mismatch. Based on an analysis of the effects of Doppler mismatch on binary phase-coded waveforms, a class of codes is proposed (hereafter referred to as the component codes), its derivation and processing is described, and the performance results are analyzed. This technique solves the problem of the Doppler mismatch of binary phase-coded waveforms at the expense of about 3 dB loss in signal-to-noise ratio (SNR). This technique may be used in tracking applications, such as accurately and rapidly measuring and tracking targets of unknown speeds. Simulation results demonstrate the validity of this technique.

Manuscript received January 25, 2001; revised November 30, 2001 and April 13, 2002; released for publication May 25, 2002.

IEEE Log No. T-AES/38/3/06451.

Refereeing of this contribution was handled by E. S. Chornoboy.

I. INTRODUCTION

Because of limitations in radar transmitter peak power, there is a conflict between long-range detection and range resolution [1, 2]. Pulse compression signals are an effective solution to the power/bandwidth problem, and among the phase-coded signals, binary phase-coded signals are of considerable interest due to their simple realization.

Matched filtering and correlation processing have been used to realize low-loss pulse compression, but Doppler causes mismatch in processing [1–4]. Doppler mismatch causes losses and time spreading of the peak response of the matched filtering and makes the sidelobe levels larger, thus affecting range resolution and decreasing the probability of detection. Doppler mismatch also affects the performance [5–8] of the subsequent sidelobe suppression filters used to reduce the time sidelobe and decreases the PSL (peak sidelobe level) [1, 3, 8]. Hence, it is important to eliminate or reduce Doppler mismatch as much as possible.

In phase-coded waveform applications, there are many ways for Doppler compensation, such as using multiple Doppler channels or employing a bank of digital filters [1–4]. The conventional methods compensate Doppler mismatch "point to point," as illustrated in Fig. 1 for one Doppler channel or one