1993-03

Resolution in radar mapping

Anderson, Michael D.

Monterey, California: Naval Postgraduate School

REPORT DOCUMENTATION PAGE

Form Approved
OMB No 0704-0188

1. AGENCY USE ONLY (Leave blank)	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	March 1993	Master's Thesis

4. TITLE AND SUBTITLE
Resolution in Radar Mapping

5. FUNDING NUMBERS

6. AUTHOR(S)
Michael D. Anderson

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Naval Postgraduate School
Monterey, CA 93943-5000

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the US Government.

12a. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution is unlimited

12b. DISTRIBUTION CODE

13. ABSTRACT (Maximum 200 words) Signal processing has led to great performance gains in radar mapping. The most critical feature of these systems is cell size, which determines resolution. Cell size is defined by range resolution and azimuth resolution.

Range resolution is improved through pulse compression. Phase of frequency modulation of a waveform yields increased bandwidth and shorter effective pulse width without reducing total signal energy. Several families of codes are investigated emphasizing matched filter output and doppler tolerance.

Azimuth resolution is improved through beam sharpening. Several beam sharpening techniques are illustrated with radar images provided by Hughes Aircraft. Range bin output plots demonstrate the effectiveness of these techniques.

With these techniques, "near-SAR" quality output can be obtained from real beam mapping radars allowing the real-time and all aspect capabilities of real beam systems to be more fully employed in tactical missions.

14. SUBJECT TERMS	15. NUMBER OF PAGES
Resolution; Pulse Compression; Beam Sharpening; Radar Mapping	90
	16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
UNCLAS	UNCLAS	UNCLAS	UL

Approved for public release; distribution is unlimited

RESOLUTION IN RADAR MAPPING

by

Michael D. Anderson
Lieutenant, United States Navy
B.S.E.E., United States Naval Academy, 1987

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN ELECTRICAL ENGINEERING

from the

NAVAL POSTGRADUATE SCHOOL
March 1993

ABSTRACT

Signal processing has led to great performance gains in radar mapping. The most critical feature of these systems is cell size, which determines resolution. Cell size is defined by range resolution and azimuth resolution.

Range resolution is improved through pulse compression. Phase or frequency modulation of a waveform yields increased bandwidth and shorter effective pulse width without reducing total signal energy. Several families of codes are investigated emphasizing matched filter output and doppler tolerance.

Azimuth resolution is improved through beam sharpening. Several beam sharpening techniques are illustrated with radar images provided by Hughes Aircraft. Range bin output plots demonstrate the effectiveness of these techniques.

With these techniques, "near-SAR" quality output can be obtained from real beam mapping radars allowing the real-time and all aspect capabilities of real beam systems to be more fully employed in tactical missions.

TABLE OF CONTENTS

ACKNOWLEDGMENTS

I wish to express my gratitude to Dr. Gurnam Gill who endeavors to take the "black magic" out of radar engineering.

I am deeply indebted to Tom Kennedy, T.D. (Buzz) Elliot, Dr. Kapriel Krikorian and Kurt Tarhan of Hughes Aircraft who went well out of their way to provide materials, documentation and explanations of the "black magic" they are building into radar systems.

Many thanks to Paul for his teamwork these past two years and to Griswell for his encouragement.

Special thanks to my wife, Karen, for her support and for putting up with me through the process.

I. INTRODUCTION

The evolution of radar has reached a new stage. Like many other areas of technical development, software is now driving innovation and providing new levels of performance. Digital signal processing has permeated every aspect of radar, from the shape of the transmitted pulse to the type of return display. One application which has greatly benefitted from signal processing advances is radar mapping.

A. RADAR MAPPING

One of the primary advantages of radar mapping is all-weather capability. Clouds, fog and haze generally do not limit system capability. In addition, radar mapping can be done day or night. Finally, both the angle and direction of illumination are controllable. For these reasons, radar mapping is invaluable in environmental mapping as well as tactical surveillance. Airborne radars are able to fill a number of mission roles through mapping.

Navigation is one of the roles for radar mapping. Numerous applications exist to fill this role. Terrain following involves vertical scanning with horizon information fed directly to flight control to allow extreme low altitude approach. Terrain avoidance is similar, but incorporates horizontal sweeps to avoid obstacles. TERCOM, or terrain contour mapping, is used to guide a platform over a timed preprogrammed ground trajectory. This mode assumes a known contour map. Radar map-matching involves guiding a platform over a pre-designated course over previously mapped terrain, as the Tomahawk TLAM cruise missile does. Either real-beam or SAR mapping can be used for this function.

In air-to-ground applications, weapon delivery can be entirely dependent on radar mapping. All weather attack capability is demonstrated in the scenario of Figure 1. [1] In this case, a SAR processing system is demonstrated. A first SAR map gives the system operator an overview of the attack area. By designating a point with a cursor, a second finer-detail image is mapped of the

immediate target area. By marking the precise target with a cursor, the operator initializes a sequence in which the system provides steering guidance to the target and controls weapon release.

In the air-to-ground battle, radar mapping also proves invaluable in reconnaissance and surveillance. The ability to distinguish and localize targets from stand-off distance provides a great advantage to friendly ground forces. Long-range targeting data and advance warning of large troop movements can provide the margin of victory in battle. Finally, radar mapping can also be quite useful in the area of battle damage assessment.

For several of these mission roles, SAR mapping has been the method of choice. With recent developments in real-beam mapping, however, real beam is expanding its role in warfare. The value of real beam mapping was demonstrated in Operation Desert Storm. A vital mission of the U.S. Air Force and other allied air assets was the location and destruction of both fixed and mobile ground-to-ground missile or Scud launchers. This became a high-priority cat and mouse game as efforts to protect Israel and Saudi Arabia from missile attack required searching great expanses of desert for relatively small targets. In effect, this mission resembled the proverbial search for a needle in a haystack. While Synthetic Aperature Radar (SAR) could provide the necessary resolution to detect these launchers, the time required for processing of the data would slow the searches considerably. With its all-aspect real time capability, real beam systems could search a much greater area in less time. This is critical when dealing with mobile targets and quick response tactics.

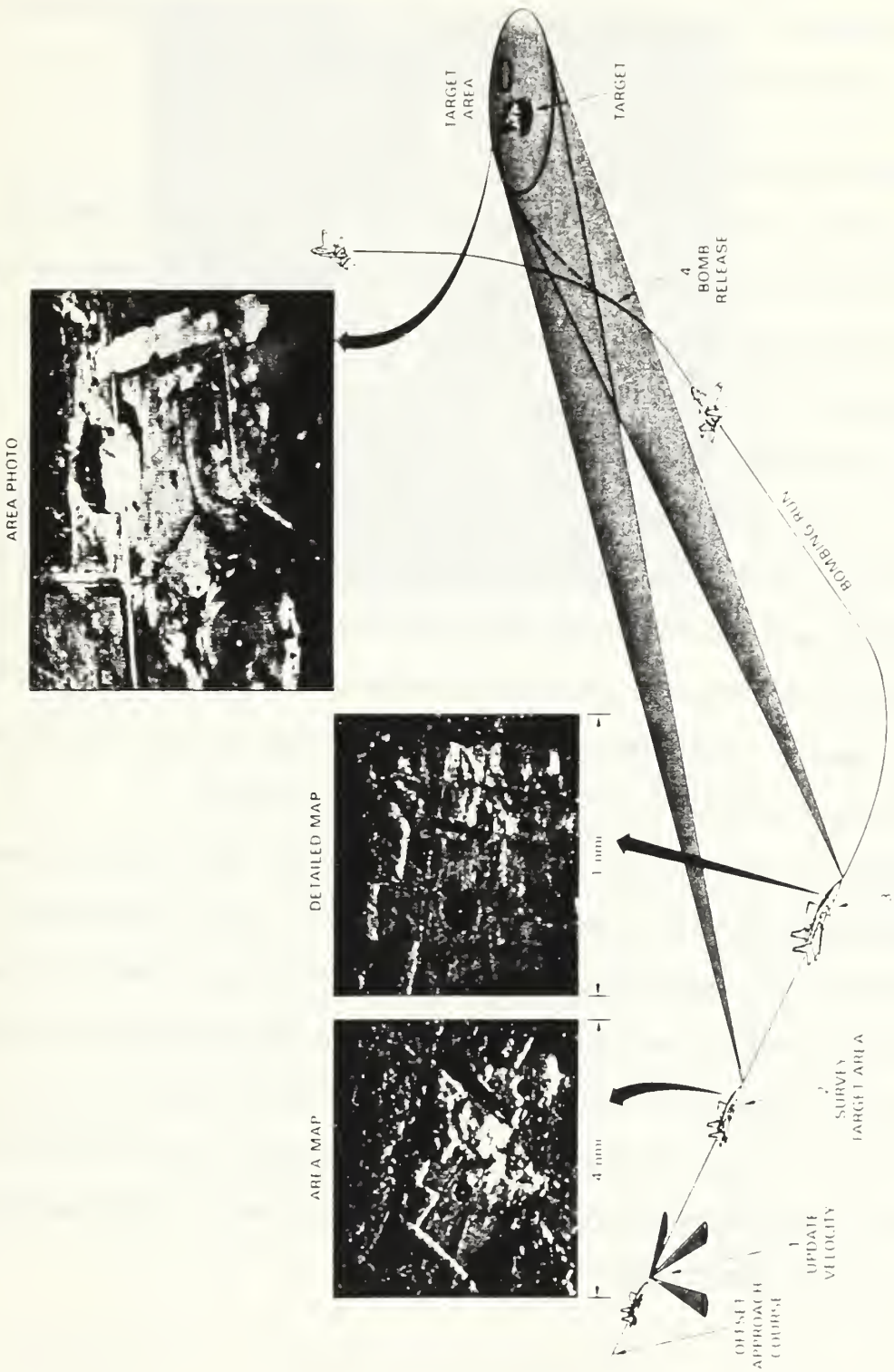


Figure 1. Radar target detection, localization and targeting

As envisioned, a real beam system would perform rapid sweeps of the desert and localize potential targets and areas of interest. A SAR system would then be employed to provide greater detail and targeting information. In this way, limited assets could be best employed to meet the threat as quickly as possible.

B. CELL SIZE AND RESOLUTION

As alluded to above, the effectiveness of radar mapping is, to a large extent, dependent on system cell size at operational range. Cell or "pixel" size is defined as a rectangle whose sides are the system range and azimuth resolution distances. Driving the choice of cell size are considerations such as object size, processing requirements, interpretation requirements and, of course, cost. A demonstration of the effects of cell size is presented in Figures 2 and 3. [1] In this figure, all portions of the shape were assumed to reflect equally with full cells lighter in the corresponding map and partially full cells darker according to the strength of the return. In actuality, several mappings from different aspects with different frequencies and polarizations would have to be combined to generate this quality return. Obviously, with smaller cell size, detection and localization of small objects is improved as well as discrimination of detail in larger objects or areas. Table 1 [1] provides typical cell size requirements for square mapping of common map features.

In general, it is easier to achieve fine range resolution than fine azimuth resolution. Resolution in range is approximately 492 feet per microsecond of pulse width. In order to improve range resolution beyond this figure, all that needs to be done is to reduce the pulse width of the system. As will be discussed in Chapter II, when there are limits to the physical reduction of pulse width, pulse compression can be used to achieve additional improvement in resolution. The matched filter and ambiguity diagram, as presented and implemented in Appendices A and B, will be used to illustrate several techniques. However, the more significant limitation for real beam systems has always been the azimuth resolution.

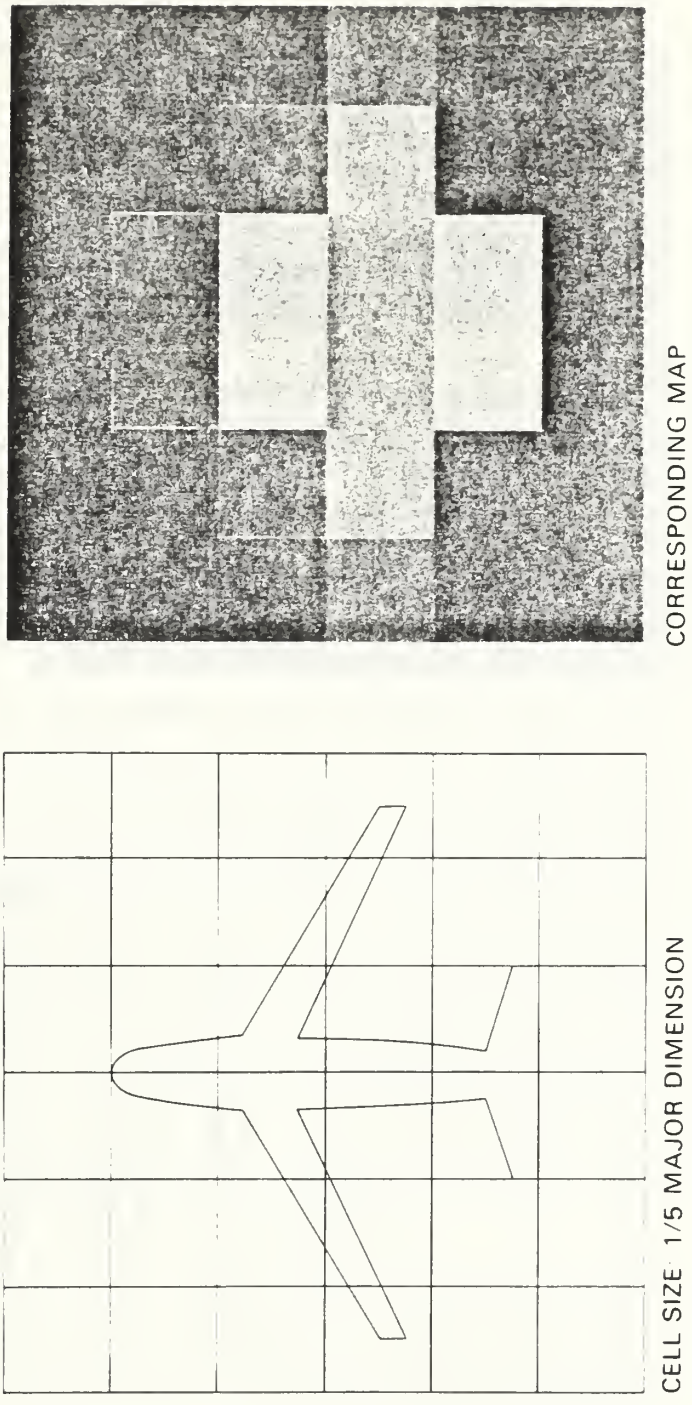


Figure 2. Simulated resolution of cell size equal to 1/5 object size.

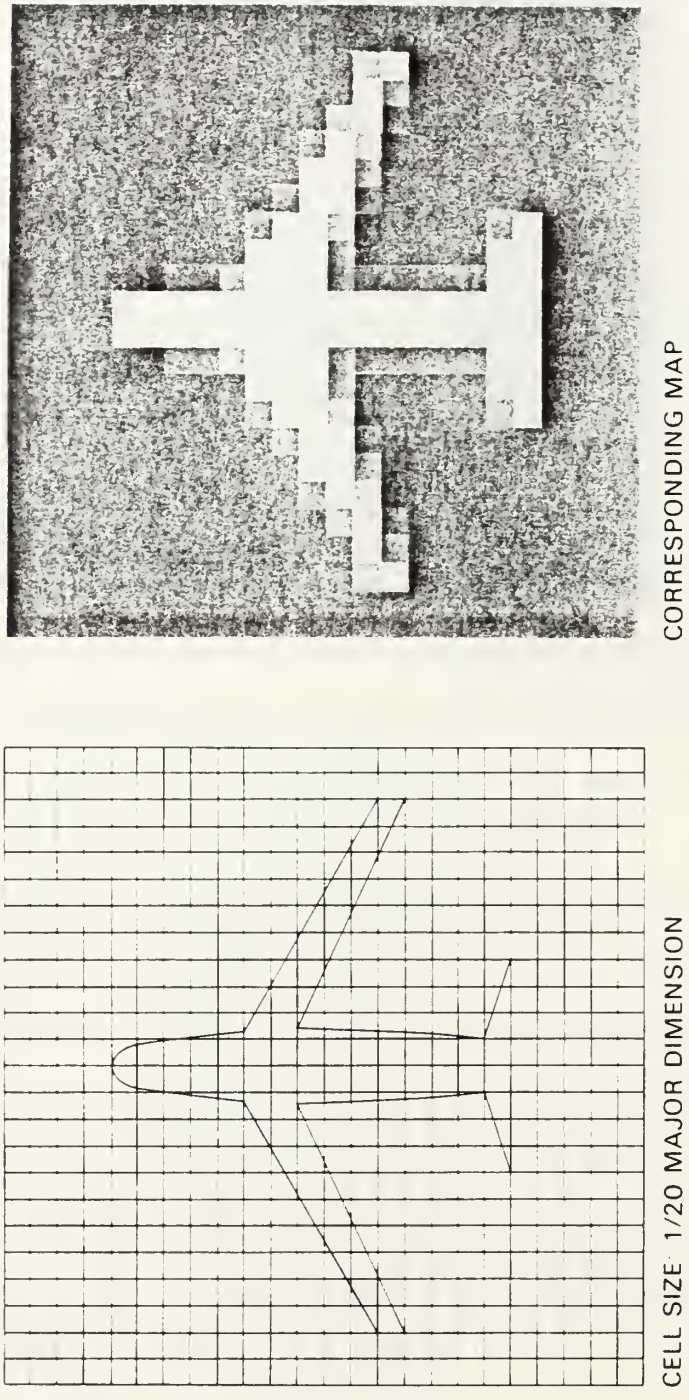


Figure 3. Simulated resolution of cell size equal to 1/20 object size.

TABLE 1
CELL SIZE REQUIREMENTS

Features to be Resolved	Cell Size
Coastlines, large cities, outlines of mountains	500 ft
Major highways, variations in fields	60-100 ft
"Road map" details: city streets, large buildings, small airfields	30-50 ft
Vehicles, houses, small buildings	5-10 ft

Azimuth resolution of a standard system is roughly the 3 dB beamwidth of the antenna times the range. At five nautical miles, a 16 foot array operating in X band has an azimuth resolution of approximately 200 feet. To recognize the shape of even large objects at long distance, the military has had to resort to SAR. Since the beamwidth varies proportionally with wavelength and inversely with antenna length, to improve resolution, it is necessary to operate at shorter wavelengths or with larger antennas. There are limits on choice of wavelength and antenna size. Limitations on machining tolerance and phase alignment make the construction of large antennas a difficult task. Airframe requirements complicate this task even further. Improvements in signal processing, however, have advanced to the point that "near SAR" quality images can be obtained from real beam systems. This is especially true for point reflectors. Chapter III will present a discussion of azimuth resolution and several beam sharpening techniques. These techniques will include actual beam sharpening algorithms as well as image enhancement techniques which are applied after initial beam sharpening. The comparative performance of these techniques will be demonstrated by the quality of through a set of images obtained through the application of algorithms developed by the Advanced Systems Division of Hughes Aircraft Company.

As these techniques are further developed and refined, real beam radar mapping will undoubtedly be applied to a great many situations which require SAR systems today. With faster

scan time, all aspect capability and good resolution, real beam systems can provide a great advantage in tactical operations.

II. RANGE RESOLUTION

A. WAVEFORM SELECTION

Long operational range and fine range resolution are two of the primary design considerations in a mapping radar. These apparently contradictory goals are achieved by proper selection of radar waveform. The radar waveform is defined by its carrier frequency, pulse repetition frequency (PRF), pulse width and intrapulse modulation. The tradeoffs involved in the selection of waveform parameters are described in this section.

The selection of carrier frequency is an important first step in waveform design. The propagation characteristics of radar change rapidly in the microwave region. Nathanson [2] provides the following guidance. Attenuation due to rain (in dB) is proportional to $f^{2.8}$ and backscatter from rain and small particles varies as f^4 dB over most of this region. Ionospheric effects also vary with frequency and are of concern at frequencies below 3 GHz, while polar operations are affected by aurora backscatter below 2 GHz. Table 2 [1] defines the radar frequency bands and common usages. The carrier frequency also has implications for azimuth resolution, though this will be discussed at greater length in the next chapter.

Once the operating frequency has been chosen, the next issue to be considered is the selection of pulse repetition frequency (PRF). The most widely used method of range measurement is pulse delay ranging. It is simple, avoids the problem of transmitter leakage noise, provides easy range measurement, and is extremely accurate. With this form of operation, however, great care must be taken when selecting the PRF. The choice of PRF is crucial because it determines to what extent the ranges and doppler frequencies observed by the radar will be ambiguous.

The returns from objects separated by the system unambiguous range are received simultaneously. The echoes from a target, therefore, must compete with ground return not only

from the target's own range but from every range that is separated from it by a whole multiple of the unambiguous range. Any overlapping of the return profiles can result in target echoes and ground clutter passing through the same doppler filter, even though the true doppler frequencies of the target and clutter are quite different. This becomes a significant problem if the received frequency is less than the width of the true doppler profile, as it must often be to reduce range ambiguities. Reducing the PRF causes the mainlobe clutter to occupy a large portion of the receiver passband. As the percentage of the passband occupied by the mainlobe clutter increases, it becomes difficult to reject clutter on the basis of doppler frequency without rejecting a large percentage of the target echoes as well. The lower the PRF, the more severe this effect becomes. With ground clutter being widely dispersed, range and doppler ambiguities greatly compound the problem of isolating target echoes.

Once the decision regarding PRF has been resolved, the designer's attention turns to the selection of pulse width. Ideally, to achieve both long detection range and fine range resolution, a system should transmit extremely narrow pulses of exceptionally high peak power. Since there are limitations on the amount of peak power which can be transmitted, to obtain long range at PRF's low enough for pulse delay ranging, fairly wide pulses must be transmitted. Pulse length determines the range resolution. To improve range resolution, a designer must shorten the pulse width. As pulse length is decreased, however, so is the total energy contained in the pulse. Eventually, a point is reached where further energy decrease in the signal is unacceptable due to detection requirements. With transmitter peak power and receiver sensitivity improvements coming slowly, and only at great cost, a clear limit seemed to exist in the detection-range resolution tradeoff. In search of a way around these limits, it was found that the limits on range resolution can be circumvented by coding successive increments of the transmitted pulse with phase or frequency modulation. This led to the development of linear frequency-modulation pulse compression.

TABLE 2
RADAR FREQUENCY APPLICATIONS

Bands	Frequency Range	Usage Summaries
HF	3-30 MHZ	Over the horizon (OTH) Radar. Very-long-range with lower resolution and accuracy.
VHF UHF	30-300 MHZ 300-1000 MHZ	Long-range line-of-sight (LOS) surveillance. Low-medium resolution and accuracy. No weather effects.
C	1-2 GHZ	Long-range surveillance/enroute traffic control. Medium resolution, slight weather effects. (200nmi)
S	2-4 GHZ	Short-range surveillance. Terminal air traffic control. Moderate weather effects. (60nmi)
C	4-8 GHZ	Long-range tracking w/high accuracy. Airborne weather detection. Increased weather effects in rain.
X	8-12 GHZ	Short-range track. Missile Guidance. Mapping/weather.
K_u	12-18 GHZ	High-resolution/synthetic aperture mapping. Satellite altimetry. Short-range or good weather required.
K	18-27 GHZ	Limited use due to water vapor attenuation.
K_a	27-40 GHZ	Very-high-resolution mapping. Airport surveillance.
Millimeter	40-300 GHZ	Smart munitions, remote sensing. Weather limited.

If a matched filter is used, the sensitivity of a radar receiver depends only on the total energy contained in the signal. The time distribution of the energy is unimportant. Within the constraints set by frequency and PRF selection, resolution then becomes the determining factor in the form of the transmitted energy. The single measure which best expresses the range resolution properties of a signal is the effective bandwidth. Long duration and high bandwidth are not incompatible if a signal contains rapid or irregular structure changes such as in pulse compression. Three types of pulse compression will be discussed in this chapter, linear frequency modulation, pseudo-random binary coding and pseudo-random polyphase coding.

B. LINEAR FREQUENCY MODULATION

Linear frequency modulation (LFM), also known as "chirp," is the result of increasing or decreasing the radio frequency of a transmitted signal at a constant rate throughout each pulse. This form of pulse compression enables very large compression ratios, and is simple to implement. There is a slight ambiguity between range and doppler using this method. A positive doppler shift will cause the signal to leave the filter sooner. The radar system can not distinguish between a target at a slightly closer range and one with a positive doppler shift at a slightly farther range.

To resolve a LFM signal, it is passed through a linear filter with a frequency change equal to the inverse of the pulse width. This must occur over a time equal to the compressed pulse width. The pulse compression ratio (PCR) achieved is equal to the ratio of the transmitted pulse width to the compressed pulse width. This is also equal to the pulse width times the frequency change which is known as the "time-bandwidth product."

C. PSEUDO-RANDOM BINARY CODES

In many applications, digital implementation of pulse compression is preferred over analog. There are many advantages to digital implementation, including:

- freedom from ringing due to impedance mismatches/analog filters
- reproducible response
- high peak-to-sidelobe ratios without weighting
- change bandwidth by changing clock frequency
- change the waveform and PCR by controlling the number of time samples and by digitally switching the matched filter
- compatibility with MTI and other signal processing.

Despite the advantages of digital implementation, significant limitations still exist due to range sidelobes. Range sidelobes exist due to the sin(x)/x shape of the energy spectrum of the uncompressed pulse. As the uncompressed pulse passes through the filter, the higher frequency spectral sidelobes travel faster than the main lobe frequencies, and the lower frequency sidelobes travel slower. The spectral shape of the energy then is reflected in the shape of the compressed pulse amplitude vs time plot. This has an effect similar to antenna sidelobes. Range sidelobes imply wasted energy and can result in false targets. Amplitude weighting, such as the use of Hamming window, can be used to minimize the sidelobes. This weighting, however, often has the effect of broadening the compressed pulse width or "main lobe" of the energy form.

The binary coding is actually a phase reversal of 0 or 180 degrees impressed upon the carrier signal. These phase sequences are usually represented by a series of +1 and -1. The resulting signal is wideband with bandwidth equivalent to twice the code clock rate. The range resolution of these codes is determined by the transmitted chip width. A chip is defined as one coded segment of the transmitted pulse.

The following are several of the more popular binary phase coding sequences. Most are easily implemented and display good autocorrelation characteristics. This class of waveform, however, does suffer from significant doppler phase shift limitations.

1. Barker Codes

These are a select number of pseudo-random codes which come close to the desirable feature of zero range sidelobes. With these sequences, the sidelobes of the compressed pulse are limited to the amplitude of the uncompressed individual segment, while the main lobe amplitude is multiplied by the number of bits in the code. Matched filter outputs for these sequences have normalized range sidelobes of only $1/L^2$ relative to the mainlobe. The primary drawback of these codes is that the longest known code is only 13 bits long. It has been proven that no other examples of these codes exist for odd lengths, and no even codes for lengths less than 6084. While it has not been proven, it is probable that no others exist at all. The known Barker codes, along with their known autocorrelation peak sidelobes and integrated sidelobes are given in Table 3. [2] The limitation to length 13 is serious, since it does not allow complete decoupling between average power and resolution. A second drawback is that the matched filter output of these waveforms degrades rapidly with doppler shift. Large sidelobes result in the autocorrelation and appear as spikes other than at the origin in the ambiguity diagram. This effect is demonstrated in Figure 4 which shows the matched filter output without doppler shift and the ambiguity diagram for a Barker 13 sequence.

TABLE 3
THE KNOWN BARKER CODES

Length	Code Elements	PSL, dB	ISL, dB
1	+		
2	+ -, + +	-6.0	-3.0
3	+ + -	-9.5	-6.5
4	+ + - +, + + + -	-12.0	-6.0
5	+ + + - +	-14.0	-8.0
7	+ + + - - + -	-16.9	-9.1
11	+ + + - - - + + + + - - + -	-20.8	-10.8
13	+ + + + + - - + + - + - +	-22.3	-11.5

2. Compound Barker Codes

One attempt to circumvent the length limitation of the Barker codes is known as compound Barker codes. This coding process can best be described through the use of the example below:

Barker length 4:	+	+	-	+	sidelobe level -12.0dB
Barker length 3: =>	(+ + -)	(+ + -)	(+ + -)	(+ + -)	sidelobe level - 9.5dB
Compound Barker length 12:	+ + -	+ + -	- - +	+ + -	sidelobe level - 9.5dB

Each bit of one sequence represents either a true or inverted replication of the second sequence. The resulting length and pulse compression ratio is equal to the product of the two original code lengths. While the range sidelobes will no longer be equal in amplitude, the main lobe to sidelobe amplitude ratio will be equal to the lowest original code used. Figure 5 provides the matched filter output and ambiguity diagram for Complementary Barker length 25 (5 in 5). This figure demonstrates that these longer codes are very sensitive to doppler phase shift.

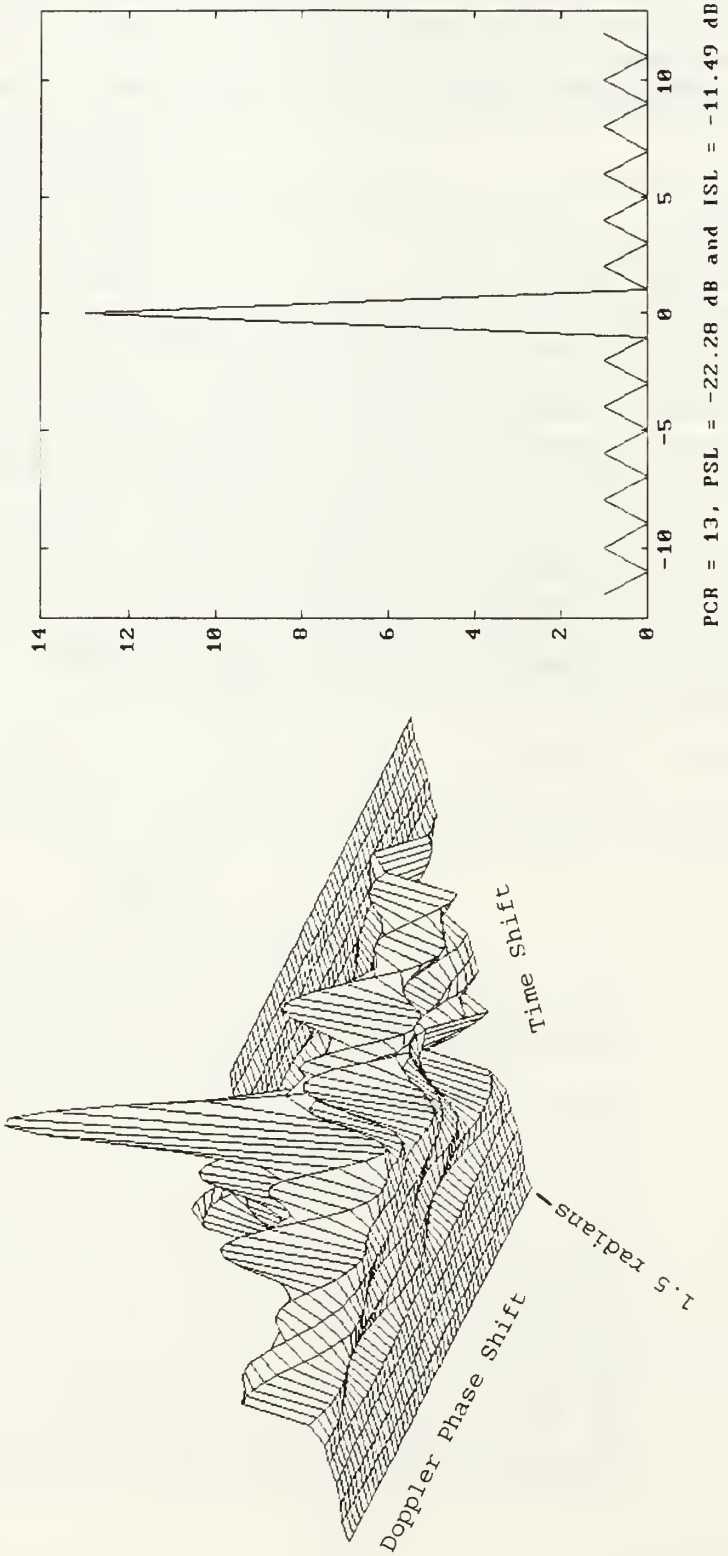


Figure 4. Ambiguity diagram and matched filter output for Barker length 13 code.

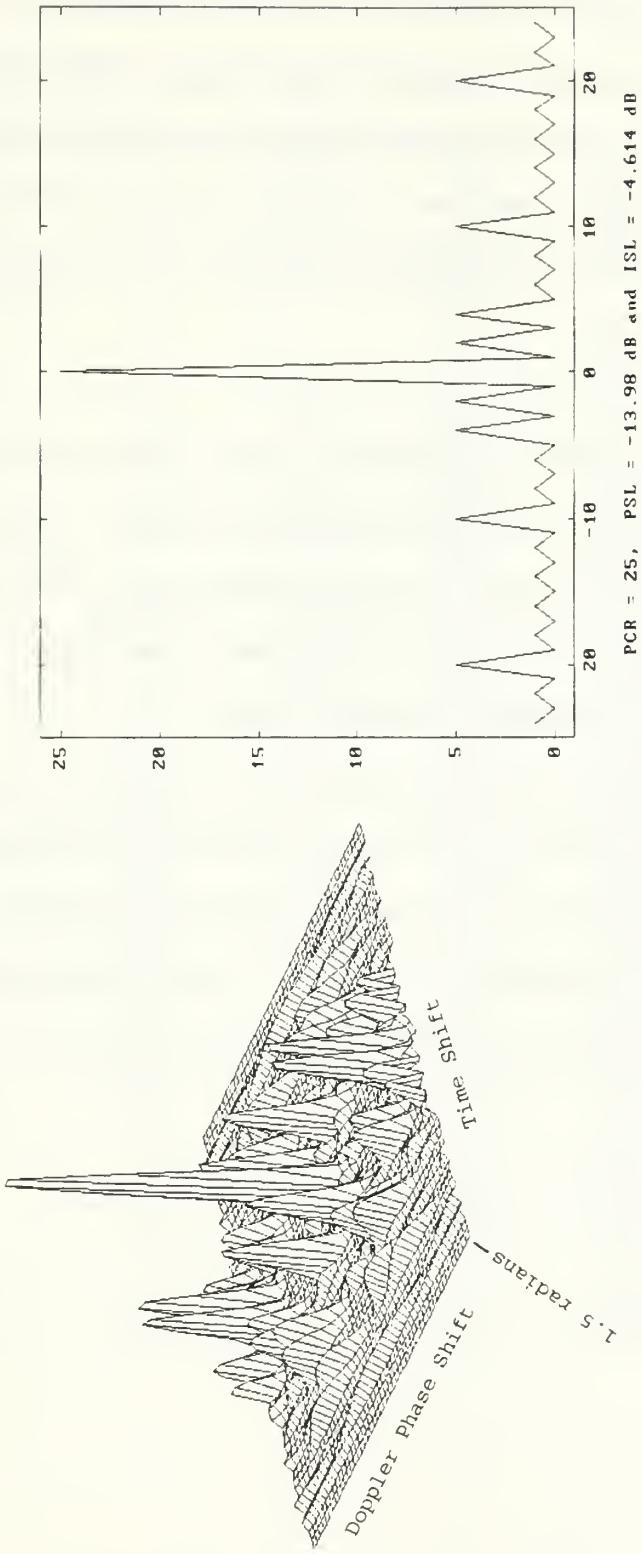


Figure 5. Ambiguity diagram and matched filter output of a compound Barker length 25 code.

3. Minimum Peak Sidelobe (MPS) Codes

Another effort has been the compilation of the minimum sidelobe codes. These are simply the binary sequences which attain the lowest PSL for given lengths. Table 4 [2] gives the MPS codes for lengths 14 through 48. These codes were discovered by an exhaustive search of all possibilities. Figure 6 gives the matched filter outputs and ambiguity diagrams for the MPS code length 5 (a Barker) and a nonideal code of the same length for comparison. Note the higher peak sidelobes in the non-optimal code.

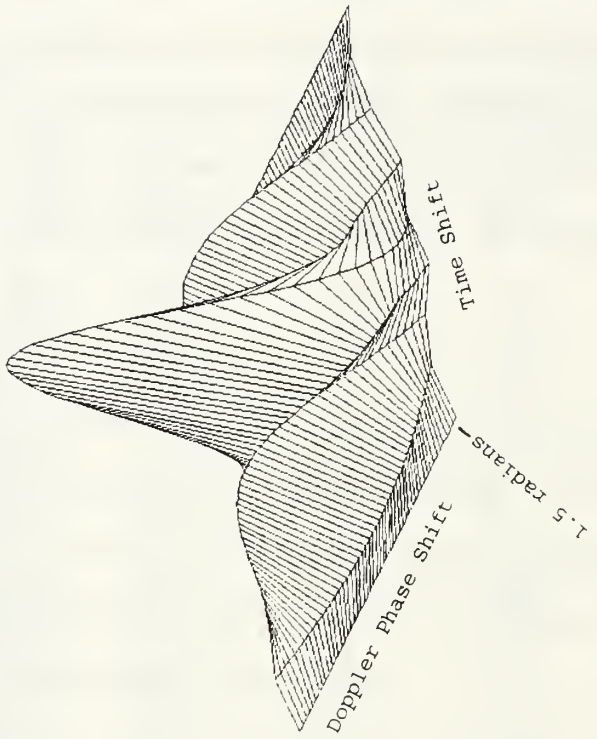
4. Complementary Codes

There are four known kernels of complementary code. A kernel is a basic length of code which cannot be shortened by inverting standard generation procedures. The known kernels are given in Table 5 [3] below. Corresponding sidelobes produced by the A and B sequences have opposite phases. By alternately modulating successive pulses and switching the matched filter, sidelobes are canceled upon integration of successive pulses.

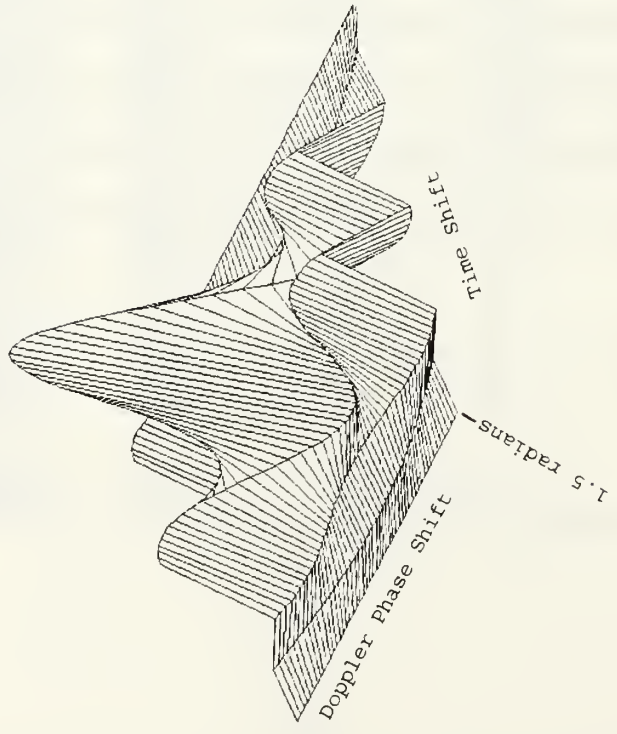
Longer sequences can be built up by "chaining" to form what is known as composite complementary sequences. Chained codes have sidelobes greater than a single code, but these cancel upon integration of an even numbers of pulses. The topic of complementary codes and their ambiguity diagrams were explored in Akita [3]. Figures 7 and 8, from this source, present an example of matched filter output and the corresponding ambiguity diagram for a complementary sequence of length 10.

TABLE 4
MINIMUM PEAK SIDELOBE (MPS) CODES

Length	Number	PSL	ISL, dB	Sample code (octal)
7	1	1	−9.12	047
8	16	2	−6.02	227
9	20	2	−5.28	327
10	10	2	−5.85	0547
11	1	1	−10.83	1107
12	32	2	−8.57	4657
13	1	1	−11.49	12637
14	18	2	−7.12	12203
15	26	2	−6.89	14053
16	20	2	−6.60	064167
17	8	2	−6.55	073513
18	4	2	−8.12	310365
19	2	2	−6.88	1335617
20	6	2	−7.21	1214033
21	6	2	−8.12	5535603
22	756	3	−7.93	03466537
23	1021	3	−7.50	16176511
24	1716	3	−9.03	31127743
25	2	2	−8.51	111240347
26	484	3	−8.76	216005331
27	774	3	−9.93	226735607
28	4	2	−8.94	1074210455
29	561	3	−8.31	2622500347
30	172	3	−8.82	4305222017
31	502	3	−8.56	05222306017
32	844	3	−8.52	00171325314
33	278	3	−9.30	31452454177
34	102	3	−9.49	146377415125
35	222	3	−8.79	000745525463
36	322	3	−8.38	146122404076
37	110	3	−8.44	0256411667636
38	34	3	−9.19	0007415125146
39	60	3	−8.06	1146502767474
40	114	3	−8.70	02104367035132
41	30	3	−8.75	03435224401544
42	8	3	−9.41	04210756072264
43	24	3	−8.29	000266253147034
44	30	3	−7.98	017731662625327
45	8	3	−8.18	052741461555766
46	2	3	−8.12	0074031736662526
47	2	3	−8.53	0151517641214610
48	8	3	−7.87	0526554171447763



(a)



(b)

Figure 6. a) Ambiguity diagram for the MPS length 5 code (+++-+).
b) Ambiguity diagram for a non-optimal code of length 5 (++--+).

TABLE 5
THE KNOWN COMPLEMENTARY KERNELS

LENGTH	SEQUENCE
2	A = {-1,-1} B = {-1,+1}
10	A = {-1,+1,+1,-1,+1,-1,+1,+1,+1,-1} B = {-1,+1,+1,+1,+1,+1,+1,-1,-1,+1}
10	A = {+1,-1,+1,-1,+1,+1,+1,+1,-1,-1} B = {+1,+1,+1,+1,-1,+1,+1,-1,-1,+1}
26	A = {+1,-1,+1,+1,-1,-1,+1,-1,-1,-1,-1,+1,-1, +1,-1,-1,-1,-1,+1,+1,-1,-1,-1,+1,-1,+1} B = {-1,+1,-1,-1,+1,+1,-1,+1,+1,+1,+1,+1,-1,-1, -1,-1,-1,-1,-1,+1,+1,-1,-1,-1,+1,-1,+1}

The complementary codes, like all binary phase codes, are limited by doppler sensitivity. Any doppler shifts encountered must be small, or pulse widths must be short, since it is the total phase shift across the pulse which matters. Excessive phase shift across the waveform will result in serious degradation of the amplitude vs. time plot.

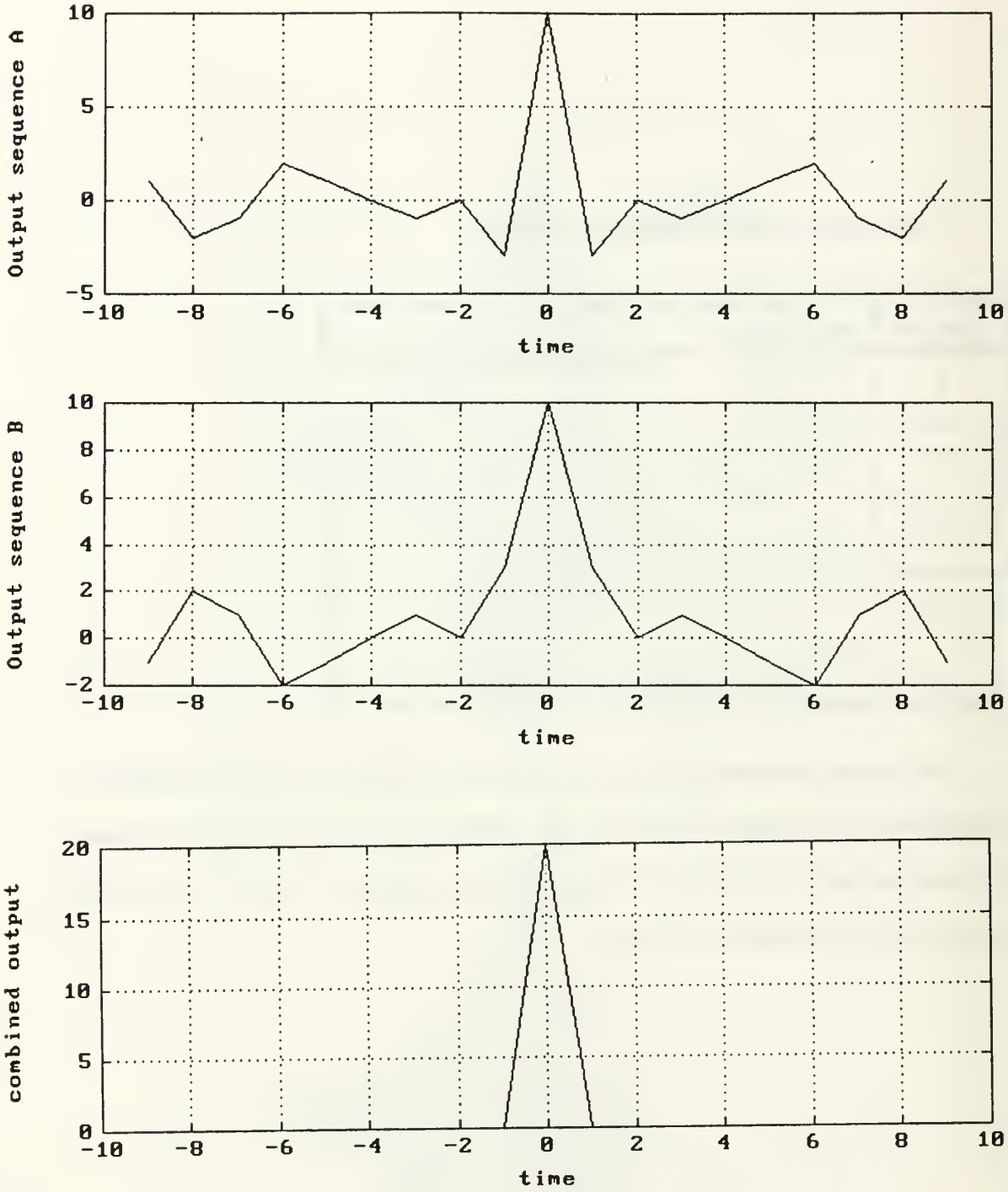


Figure 7. Matched filter output of complementary length 10 kernels; sequence A, sequence B and summation.

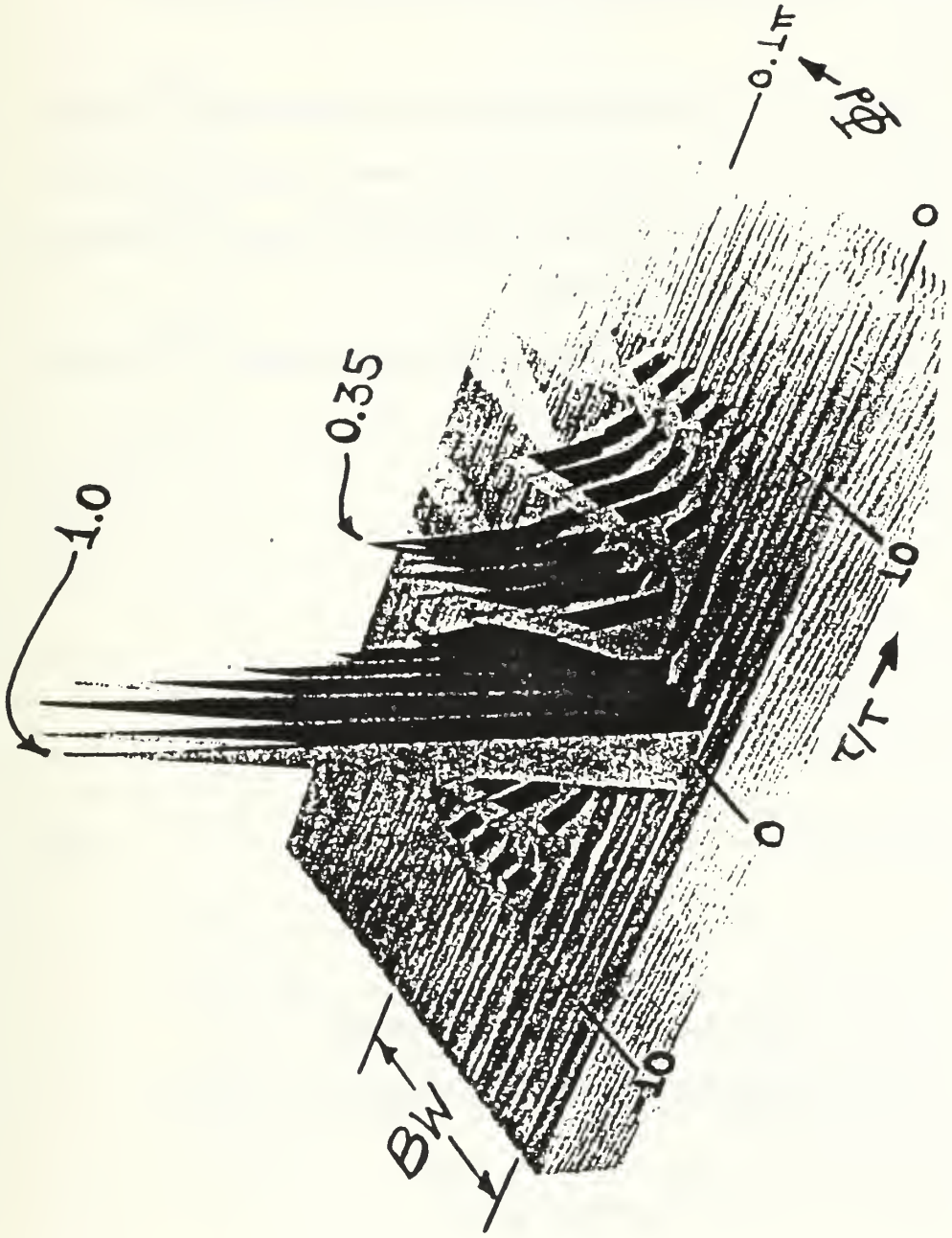


Figure 8. Ambiguity diagram of complementary length 10 kernel.

C. POLYPHASE CODES

Polyphase codes are another class of pseudo-random sequences. With these codes, the bits can consist of any number of different harmonically related phases, rather than only 0 or 180 degrees. Several such codes are available, with some of the more common ones described below.

1. Frank Codes

The Frank codes are a series of inphase and quadrature (IQ) samples taken at the Nyquist rate of a "step-chirp" waveform. The formation of the code is demonstrated by Figure 9. [4] To generate this code, P phase increments are defined by dividing 360/P. P groups of P segments are then generated according to the following procedure:

The first phase of each group is 0. The phases of the remaining segments in each group increase in increments of

$$\Delta\theta = (G-1)*(P-1)*\theta \tag{2.1}$$

where

G is the group number and

θ is the basic increment.

The resulting code is given by

$$\theta(i,j) = (2\pi/N)(i-1)(j-1) \tag{2.2}$$

For example:

P=4:

Group 1: segment 1=0, segment 2=0, segment 3=0, segment 4=0

Group 2: segment 1=0, segment 2=(360/4)(1)(1)=90, segment

3=(360/4)(1)(2)=180, etc.

The resulting code is shown at the bottom of Figure 9.

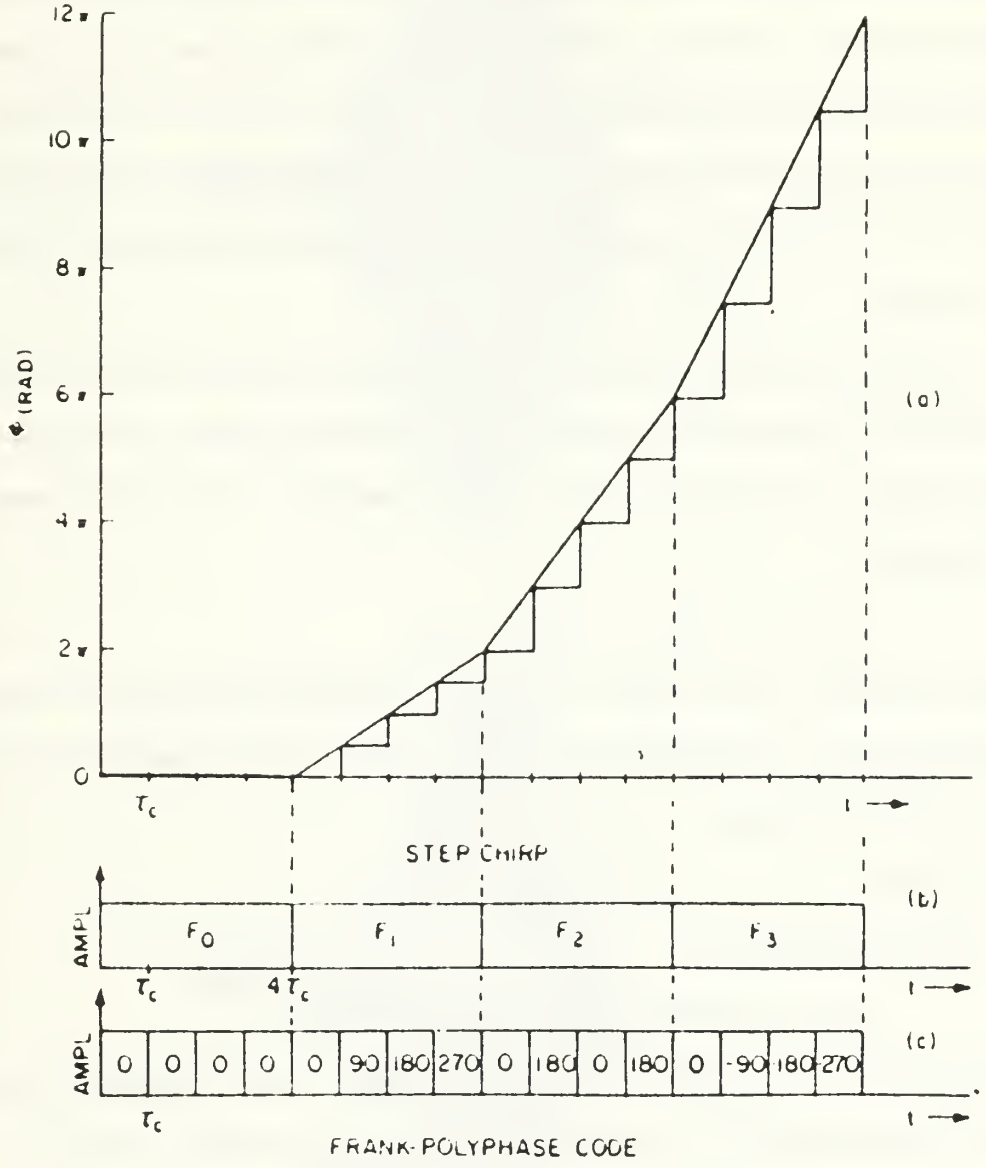


Figure 9. Derivation of Frank code from a step-chirp waveform.

For a given number of segments, a Frank code provides the same pulse compression ratio as a binary phase code and the same peak to sidelobe ratio as a Barker code. Yet, by using more phases (increasing P), the codes can be made any length. As P is increased, however, the size of the fundamental phase increment decreases, making performance more sensitive to externally introduced phase shifts and imposing more severe restrictions on uncompressed pulse width and maximum doppler shift. This is demonstrated in Figures 10 and 11 showing degradation of the ambiguity diagrams for increasing length Frank codes. Another limitation of the Frank codes is that the largest phase change between elements occurs at the center of the sequence. Because of this, receiver bandwidth limitations have especially significant effects on the center peak to sidelobe ratio.

2. P1 Code

To alleviate the problems of bandwidth limitation present in the Frank codes, the code groups representing the j frequencies were rearranged in time order of transmission. This placed the lowest amount of phase increment in the center to preserve the main lobe. This led to the creation of the P1 code given by

$$\Theta(i,j) = -(\pi/N)[N-(2j-1)][(j-1)N+(i-1)] \qquad (2.3)$$

When N is odd, the P1 code is simply the Frank code rearranged to have conjugate symmetry around the DC term. The autocorrelation functions of the two codes are nearly identical, but the P1 has no bandwidth limitations.

3. P2 Code

The P2 code is given by

$$\Theta(i,j) = \{(\pi/2)[(N-1)/N]-(\pi/N)(i-1)\}\{N+1-2j\} \qquad (2.4)$$

It has an autocorrelation function nearly identical in magnitude to both the Frank and P1 codes, and has no bandwidth limitations. The difference is that P2 is real as opposed to complex

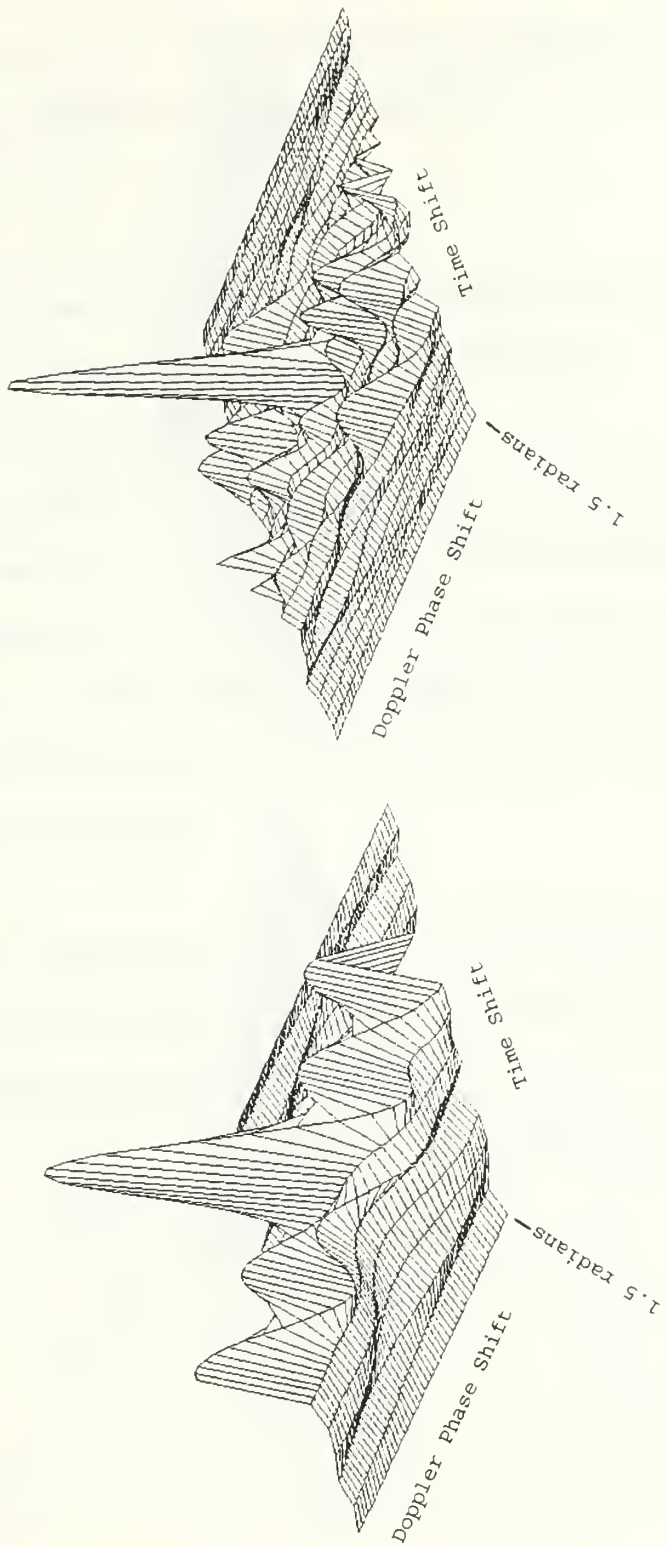


Figure 10. Frank codes of P=3 (length 9) and P=4 (length 16).

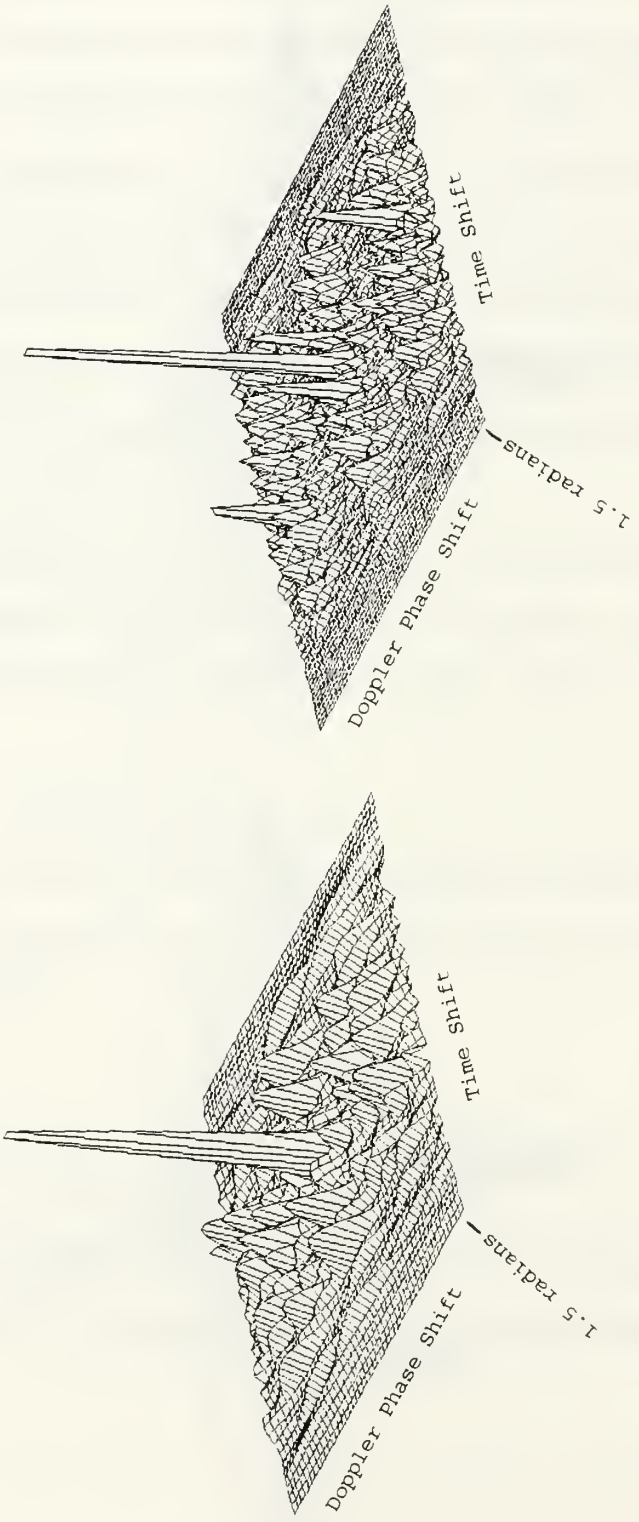


Figure 11. Frank codes of P=5 (length 25) and P=7 (length 49).

because it is symmetrical. Figure 12 [4] demonstrates the improved bandwidth tolerance of both P1 and P2 over the Frank codes. In this example, bandlimited results show a 6 db improvement in peak sidelobe level for P1 over Frank code. Both P1 and P2 still remain fairly doppler intolerant, however. This is a characteristic of the analog step chirp waveform from which both, along with the Frank code, are derived.

4. P3 Code

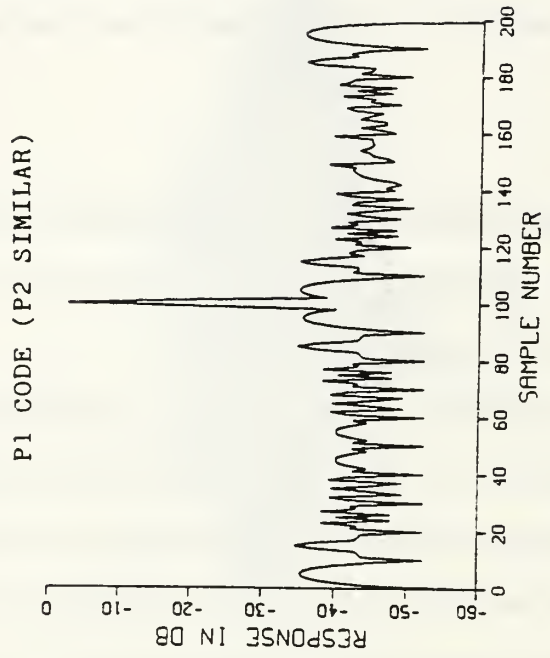
Since the linear chirp waveform is much more tolerant of doppler than the step chirp, the P3 code was developed from it. The code is written as

$$\Theta(i) = \pi(i-1)^2/\rho \text{ as } (i) \text{ varies from 1 to } \rho \qquad (2.5)$$

In this equation ρ represents the pulse compression ratio. The resulting waveform is just as doppler tolerant as the analog linear chirp, but suffers from the same bandwidth limitations as the Frank codes. Figure 13 demonstrates the improved doppler tolerances of the P3 code. While the unshifted matched filter output shows the Frank to have a superior peak-to-sidelobe ratio by 4 db, the doppler shifted signals clearly show the superiority of P3. Of particular note is the elimination of grating lobes and preservation of the image lobe.

5. P4 Code

A waveform both doppler tolerant and not bandwidth limited resulted from rearranging the P3 code. The resulting code is known as P4 and is written as $\Theta(i) = [\pi(i-1)^2/\rho]-\pi(i-1)$. Figure 14 [4] demonstrates the bandwidth limitation improvement of P4 over P3.



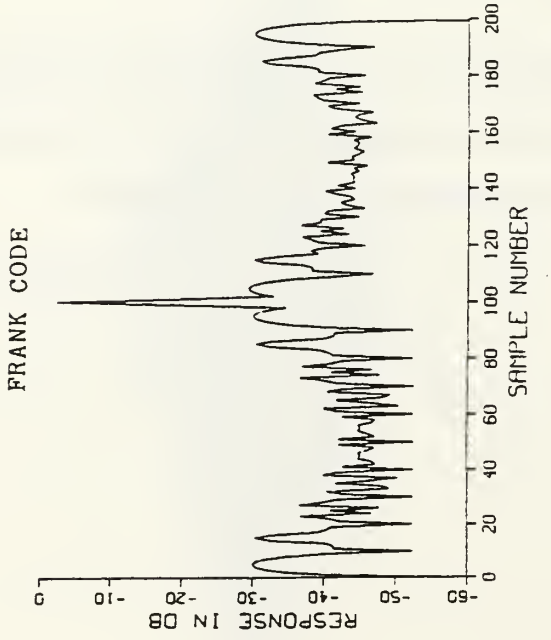


Figure 12. Effect of precompression bandlimiting of Frank and P1 codes of length 100.

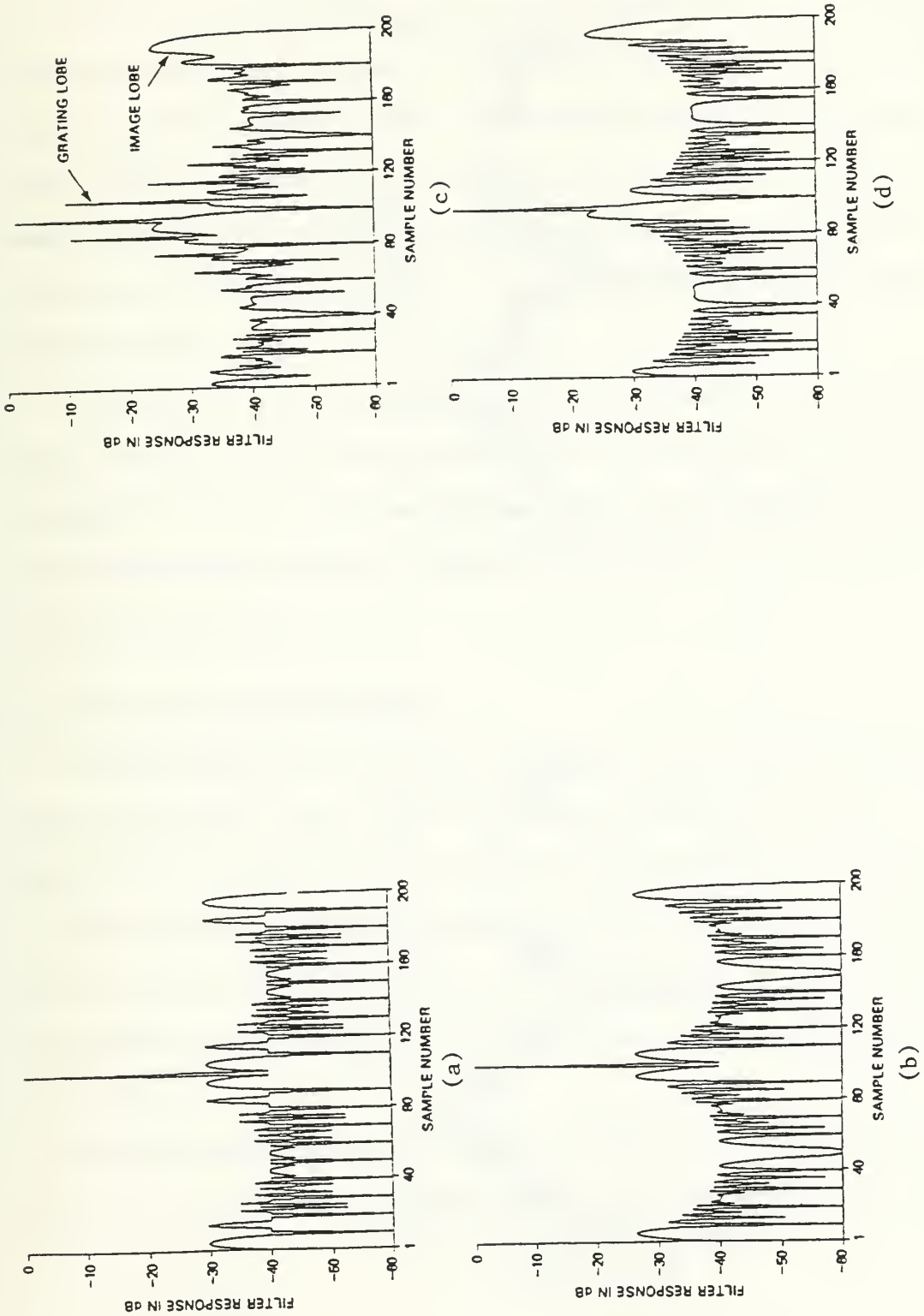


Figure 13. a) and b) are Frank and P3 code matched filter outputs for length 100. c) and d) show matched filter outputs for the doppler phase shifted case.

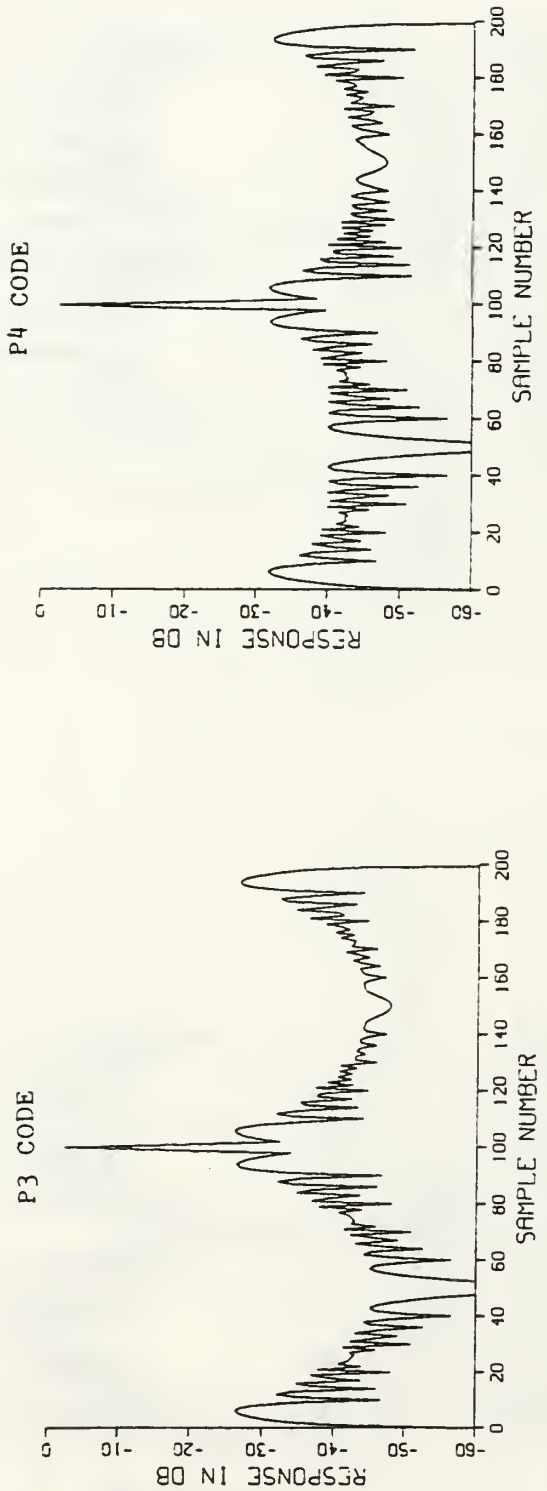


Figure 14. Bandlimited matched filter outputs for P3 and P4 codes length 100.

III. AZIMUTH RESOLUTION

With pulse compression providing a means of achieving good detection range and fine range resolution, the largest remaining constraint in a mapping radar is azimuth resolution. As mentioned in the introduction, in an unprocessed real beam system, azimuth resolution is determined by the antenna beamwidth. While beamwidth control is adequate for target detection, prohibitively large antennas result for the narrow beams required to determine detailed characteristics such as the number, size or separation of targets. The need for information of this sort led to the development of synthetic aperature radar.

Unfortunately SAR systems also have limitations. These systems are unable to map at nose aspect, are expensive, and are hardware dependent, Most importantly, they are comparatively slow in mapping due to the extensive amount of signal processing required. These limitations have led to renewed efforts to improve the angular resolution of real beam systems. These efforts are referred to as real beam sharpening.

A. BEAM SHARPENING TECHNIQUES

Beam sharpening reflects two areas of effort: actual beam sharpening algorithms, and image enhancement techniques which are applied after initial beam sharpening. While some implementations will combine some of both, they will be presented individually in this paper.

These techniques enhance angular designation accuracy in all aspect scenarios. Further, real-time post-processing is possible to accomplish this without extending the radar frame time. Even while implementing these additional processors, real beam systems can map a 60 degree swath in just 1-2 seconds.

Techniques already in use on the F-15E provide a 2-1 enhancement in designation accuracy. Test results of new techniques being developed at Hughes Aircraft promise beam sharpening of 6:1

with 6 dB SNR and up to 25:1 for a 20 dB SNR return. These estimates are based on isolated corner reflector return. The following pages will provide a brief introduction into a number of these techniques.

1. Combination of Sum and Difference Channels

Useful for resolving point targets, this method was developed at Hughes Aircraft and is currently implemented in the F-15E. Figure 15 [5] demonstrates the concept. The boresight null in the difference (D) channel pattern forms a sharp peak when combined with the sum (S) channel pattern. A weighting factor (K) controls the beam sharpening ratios. A value of (K) equal to .75 is common. The output (S_c) is given by

$$S_c = |S| - K|D| \tag{3.1}$$

This technique can also be applied using sum and synthetic difference channels. If difference channel information is not available, synthetic difference channel data may be generated through the use of anti-symmetric weights. The difference channel data (D) is generated as shown below.

$$D_j = \Sigma(1\text{-}k)\, w_k \;\; S_{j+k} \tag{3.2}$$

where w_k is a suitable set of anti-symmetric weights, (j) is the radar pulse index, and S_{j+k} is the sum channel data for each pulse and set of weights. The optimum weights (w_k) to maximize the gain slope of the synthetic difference pattern vary as

$$w_k \propto dG(\Theta)/d\Theta \tag{3.3}$$

where:

$G(\Theta)$ = antenna azimuth gain

Θ = k(scan rate)/PRF

The sum and synthetic difference channel data are then combined as for the previous case.

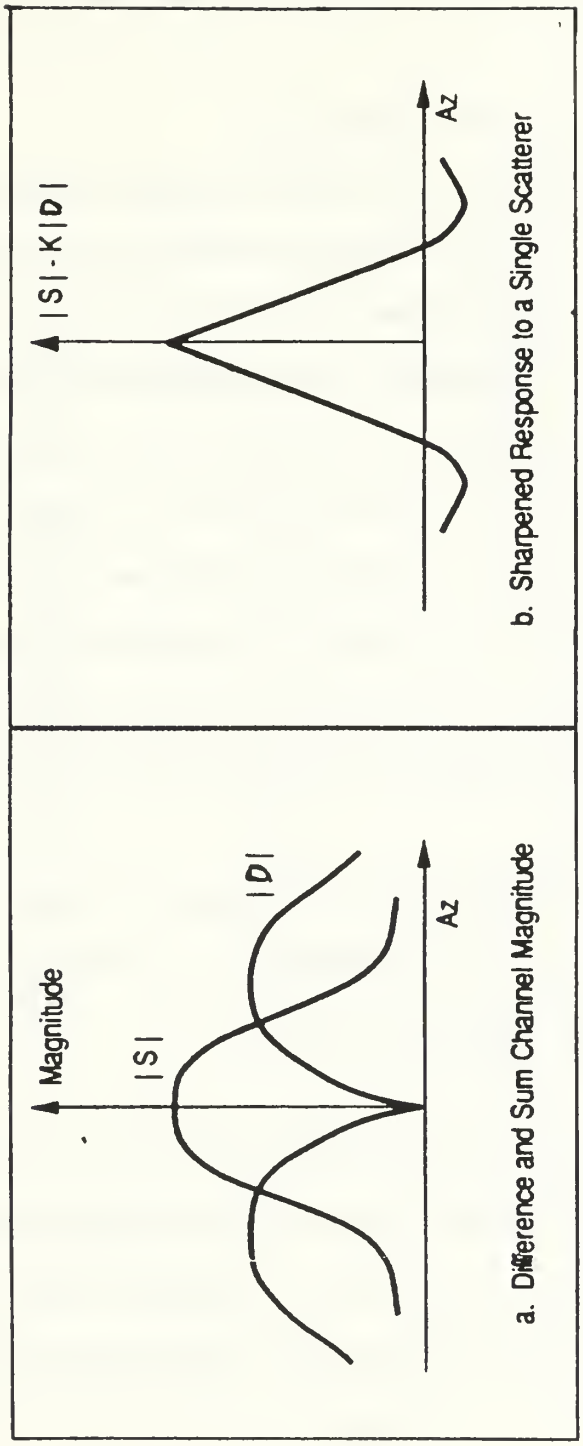


Figure 15. Combination of sum and difference channels resulting in attenuation of response away from boresight.

2. Monopulse Beam Sharpening

Also used for point targets, this technique uses sum and difference channels to form an azimuth discriminant on the I/Q data for each return in each range bin. The discriminant (a) for each pulse (j) is given by

$$a = k\ \mathrm{Re}((S_j^* \cdot D_j)/(S_j^* \cdot S_j)) \tag{3.4}$$

where k is a scaling factor given by 1/(discriminant slope)(angular bin spacing), S_j is the sum channel data for each pulse and D_j is the difference channel data for each pulse. Re is the real operator and * denotes complex conjugation. This discriminant value gives the distance of each scatterer from the center of the range bin. As the scatterer enters the beam, each subsequent pulse will show the scatterer moving closer to the center as the beam scans across it. Eventually, the discriminant will reach zero, cause a sign change and move out of the beam. Once the discriminant is formed for a pulse and range bin, the value of the discriminant assigns the return to one of the azimuth bins which each range bin is divided into. The magnitude of the sum channel data is what is actually stored in the bin. With each pulse, data is continually added to the bins. Upon completion of the scan, the accumulated sum of the contents of each bin is displayed thereby localizing scatterers.

Synthetic monopulse beam sharpening is also possible. It involves the sum channel and synthetic difference channel. The synthetic difference channel is formed as previously described, and is then applied as in regular monopulse beam sharpening.

3. Inverse Filtering

In a large class of imaging systems, spatial degradation can be modeled by a linear-shift-invariant impulse response and additive noise. In these cases, restoration can be accomplished through linear filtering. This is illustrated as part of an algorithm in Figure 16. [6] The image degradation is modeled by a system with a particular transfer function $H_D(x,y)$. Noise N(x,y) is

$$F_O(x, y) = \int_{-\infty}^{\infty}\int F_I(\alpha, \beta) H_D(x - \alpha, y - \beta)\, d\alpha\, d\beta + N(x, y)$$

or

$$F_O(x, y) = F_I(x, y) \circledast H_D(x, y) + N(x, y)$$

$$\hat{F}_I(x, y) = \int_{-\infty}^{\infty}\int F_O(\alpha, \beta) H_R(x - \alpha, y - \beta)\, d\alpha\, d\beta$$

Figure 16. Continuous image restoration model.

added to the signal resulting in the observed image field $F_O(x,y)$. After passing through a restoration filter, an approximation to the original image field F is obtained.

Inverse filtering reflects the earliest attempts towards image restoration. In this process, the transfer function of the degrading system is inverted to yield a restored image. One application of this technique is the modeling of low-resolution data as the convolution of high resolution data with a known spatial antenna pattern. With a known transfer function for the antenna and a good estimate of the mean square of the observed data, the mean square of the high-resolution signal can be reconstructed through appropriate deconvolution techniques. This assumes that the original and observed data are random processes. If source noise is present, however, an additive reconstruction error will result. This error can be quite large at spatial frequencies for which the spatial image degradation transfer function is small (typically, at high spatial frequencies). This impairs the quality of high detail regions of an image. Proposals have been made to reduce susceptibility to noise. One method involves using a restoration filter with a transfer function which is multiplied by a unit step function over the areas where the image spectrum is expected to exceed the noise spectrum, and is zero everywhere else. This strikes a compromise between noise suppression and loss of high-frequency image detail. Other filter implementations, such as the Wiener filter and its variations, provide better performance in the presence of noise.

4. Maximum Entropy

Maximum entropy is a nonlinear technique which can be used to resolve closely spaced signals. It is an extension of inverse filtering which incorporates linear prediction to better resolve an image scene. The technique is applied to magnitude data in each range bin. As scatterers enter the beam, they cause predictable transients due to the roll-off of the antenna beam shape. A Fourier transform of the data is then taken and scaled to remove the effect of the antenna pattern. This results in a set of autocorrelation coefficients. A set of predictive linear equations is applied and solved to further resolve the data which has been corrected for actual antenna length and generate a

second set of coefficients. A second Fourier transform is then performed, this time on the coefficients. The resulting transform is related to image intensity with the poles of the coefficients denoting the scatterers.

5. Hughes Advanced Discrimination Technique (HADT)

Further work has been done in this field by personnel at Hughes Aircraft Advanced Systems Group. While much of their work is proprietary in nature and cannot be discussed in detail, the following figures show the improvement in resolution which can be achieved with today's signal processing.

Figure 17 [5] provides a comparison of the various beam sharpening techniques for both point and extended targets. Note that for point targets, the maximum entropy, monopulse beam sharpening and HADT methods provide excellent results. For extended targets, the maximum entropy and HADT work well with the exception of some peaking in the return just before roll-off. A Hughes modified inverse filter has also attained good results on extended targets. (Figure 18 [5] shows the real return of an air-to-ground radar with a 2.5 degree beamwidth mapping two corner reflectors spaced three degrees apart.) Even though the separation is greater than the 3 dB beam width, the objects are indistinguishable in the raw return. In the HADT processed response, however, the targets are readily distinguished. Figure 19 [5] provides a comparison of SAR results, unprocessed real beam return and sharpened real beam ground mapping (RBGM) return. While targets in the unprocessed real beam image appear as smears due to the wide beam width sweeping across them, the HADT sharpened image reveals the SAR processed image in clarity.

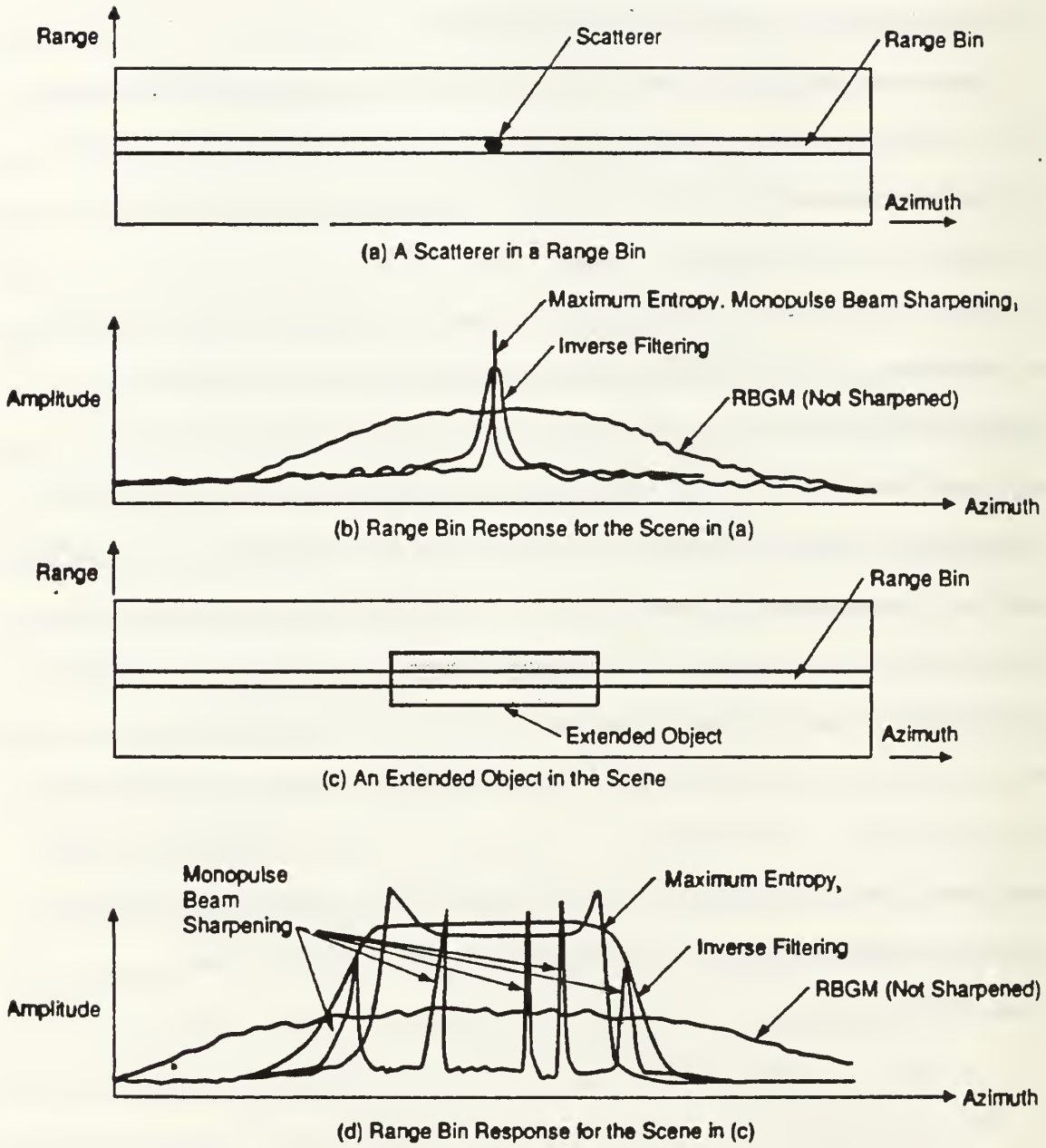


Figure 17. Relative effectiveness of commom beam sharpening techniques on point reflectors and extended objects.

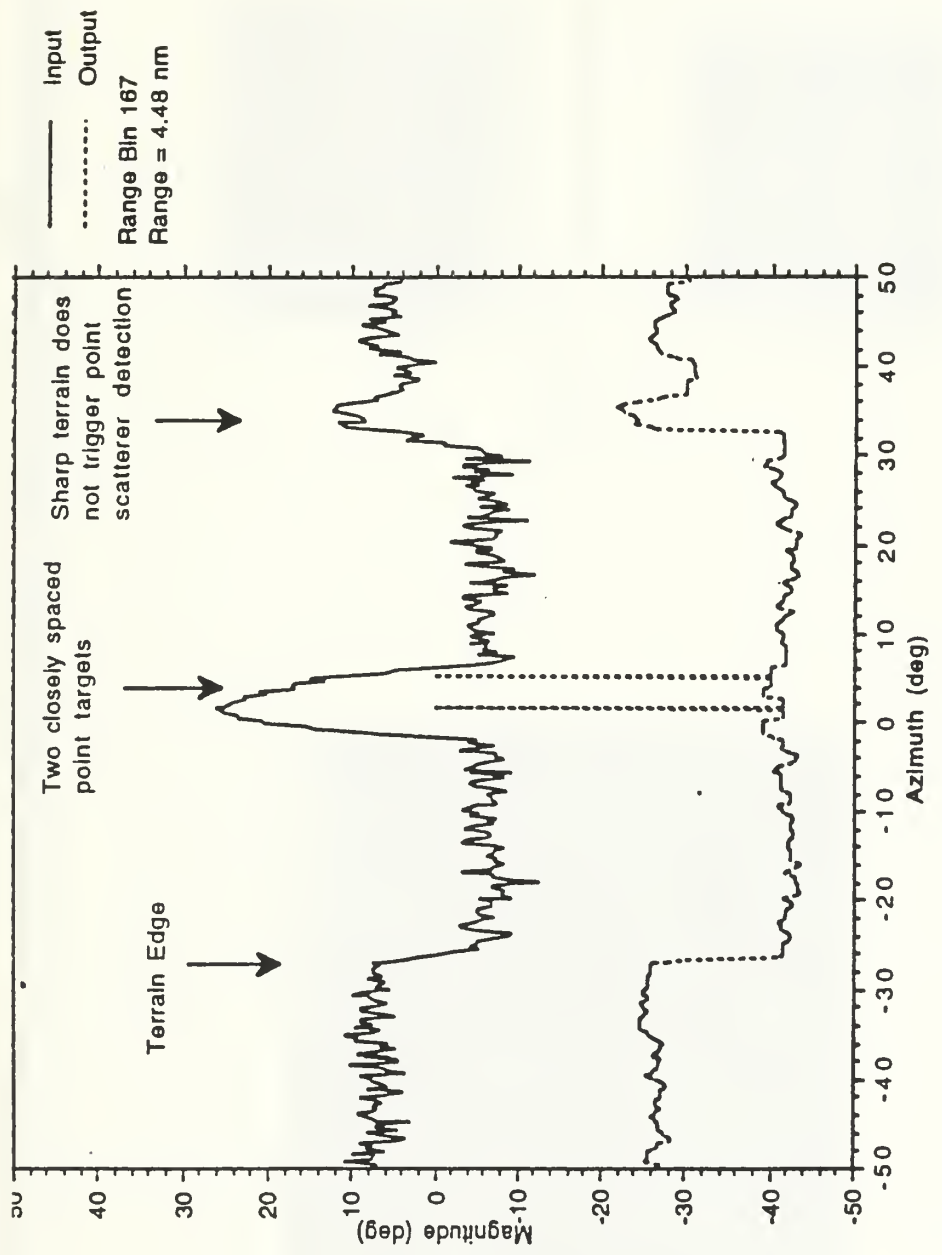
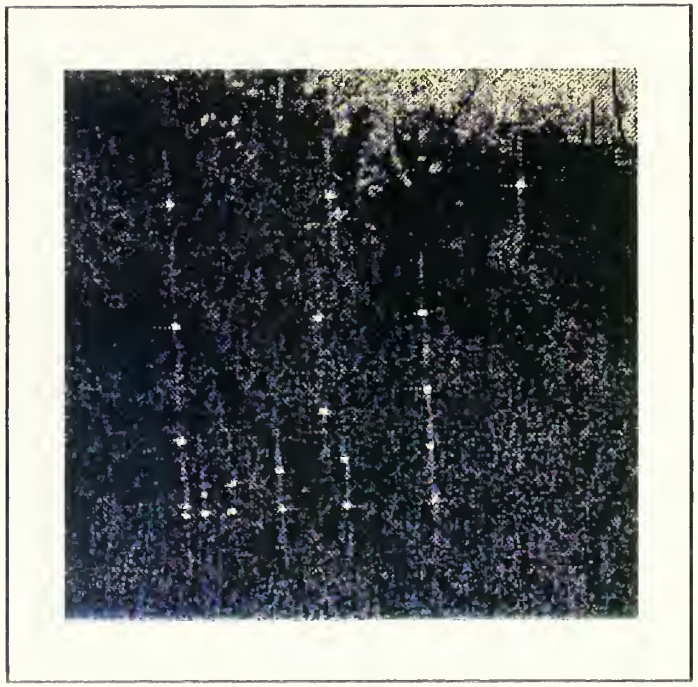


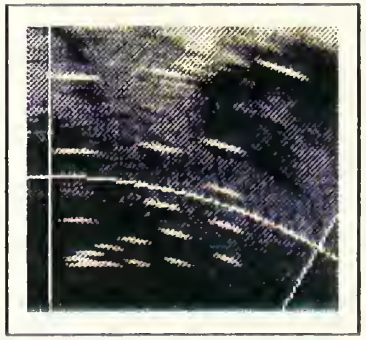
Figure 18. Raw and processed range bin return of corner reflectors.

RBGM Beam Sharpening Example: Corner Reflector Array

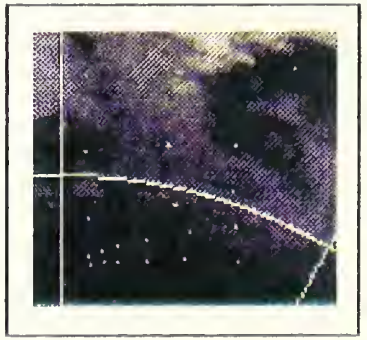
HUGHES



SAR Map (Resolution 42 ft)



RBGM Before Beam Sharpening



RBGM After Beam Sharpening

Figure 19. Comparison of SAR, unprocessed real beam and sharpened real beam displays.

B. IMAGE ENHANCEMENT

Once the beam sharpening techniques discussed above have been applied, image processing can further improve the output and generate more accurate and useful information. In order to implement these techniques, a high throughput processor is required such as the APG-70, APG-73 or Hughes CIP. Several image processing techniques which have been widely implemented are presented below.

1. Histogram Flattening

In many images, detail in the darker regions is imperceptible. Signal processing is required to alter brightness or improve contrast. One method which works well is histogram modification. Both adaptive and non-adaptive histogram modification techniques exist. In principle, these techniques involve rescaling the image to some desired form. A common method of scaling is to find the average value of the histogram and normalize quantized bands of pixels against it. By allowing fewer levels of quantization, higher contrast of features is obtained. This process, however, does result in an increase in quantization error.

2. Edge Detection

A common task in radar mapping is to determine the extent and position of extended targets such as airfields and roads. One method of performing this task is through the detection of luminescent edges or discontinuities between reasonably smooth regions. The only digital images which exhibit step edges are artificially generated test patterns and images. Due to low-pass filtering prior to digitization, any digital image resulting from optical or radar images of real scenes will have reduced edge slopes. For this reason, additional processing is required to identify and localize the boundaries of any extended objects.

One of the primary methods of edge detection is differential detection. This can take the form of either first order derivative or second order derivative processing. In the first order case, some form of spatial first order differentiation is performed, with the resulting edge gradient being

compared to a threshold value. A gradient above threshold denotes the presence of a boundary. This process is demonstrated in Figure 20. [7]

The simplest method of generating gradients is to compute the running difference of pixel magnitude along rows and columns. Diagonal gradients can be obtained by forming a running difference of diagonal pixel pairs. The running difference technique is highly susceptible to small fluctuations in object luminance. Object boundaries are also not well defined. Modifications do exist to better define edge locations. One such method which has provided good results is to simply take the pixel differences separated by a null. Luminance fluctuation problems have been limited through the use of weighted averages and moving windows. Larger windows improve the detection of edges in high noise environments, but the computational requirements increase rapidly with size. A sample C language program to perform first order derivative edge detection is presented in Embree [7]. This program uses convolution smoothing and vertical edge detection with a 3X3 moving window called Sobel detection.

In second order derivative processing, a change in the polarity of the second derivative signifies an edge. One implementation utilizes the Laplacian which is zero if the intensity is constant or changing linearly. A greater rate of change will result in a sign change at the point of inflection. The location of the zero crossing signifies the assigned location of the edge. A shortcoming of the Laplacian is that it produces a fairly noisy result. Extraneous edge points are often introduced, and especially strong responses result from corner reflectors.

An alternate implementation is to first estimate the edge direction. The second order derivative is then taken along the edge. The difficulty in this method lies in estimating the edge direction. First order detection techniques are sometimes used for this purpose.

A second method of edge detection is model fitting. This involves fitting the observed data in a region of pixels to matrix mappings of possible edge types. Once a close fit is found, an edge is deemed to exist and the region of pixels is assigned the properties of the model selected. In this

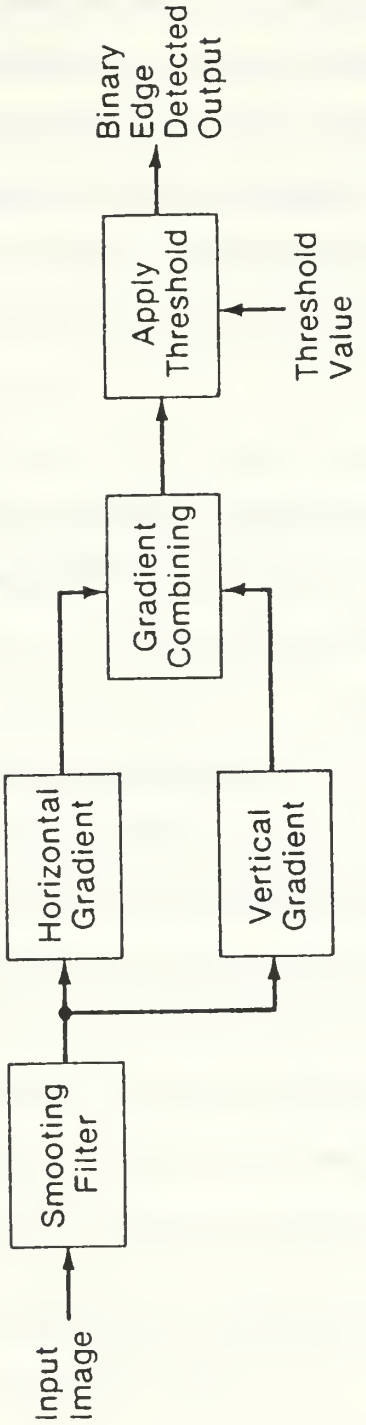


Figure 20. Formulation of running gradients and threshhold edge detection.

method, the match is made if the mean square error is below a threshold value. The model fitting method requires much more computation than the derivative detection methods.

Pratt [6] provides an excellent comparison of these methods with several implementations of each. The implementations are evaluated with respect to three criteria. The first, *good detection*, refers to the ability of the system to detect edges without undue false detections. The selection of threshold mirrors the tradeoffs of probability of detection (P_d) and probability of false alarm (P_f) in target detection. The second, *good localization*, refers to the edge being designated as close to edge center as possible. The third, *single response*, states that only one peak above threshold should exist for each edge crossing.

3. Line (boundary) Detection

Similar techniques are used in the detection and refining of image lines. A line can be modeled as a series of closely spaced parallel edges. Weighted averages and model fitting for orientation are often used to limit susceptibility to noise.

4. Thresholding and Centroiding

This technique can detect and localize point scatterers as well as determine the extent of large objects. Thresholds are determined by local noise estimates. An m out of n detection scheme is employed to separate point reflectors from extended objects. If greater than m cells in an n cell window exceed the threshold, then an extended object is determined to exist in the space. If fewer than m threshold crossings exist, then a centroiding algorithm is applied at the peak crossing. This algorithm estimates the location and amplitude of the point object and designates it to the appropriate cell. The rest of the window is then replaced by the local background. An estimate of the location for a point scatterer is obtained by summing the weighted amplitudes of each cell and dividing by an unweighted sum.

5. Median Filter

Noise can exist in one of two forms. It can be contained in a separate portion of the frequency spectrum from the signal, or it can coexist with the signal across the spectrum of interest. In the first instance, linear filtering can be quite effective. In the second case, filtering to remove noise will also degrade the signal. An example of the second type of noise is "salt and pepper" noise. It often results from A/D converter problems or digital transmission errors. These bit errors result in impulses in an otherwise smooth sequence. Linear filtering to remove these "specks" in an image can result in blurred images and loss of high frequency information. An effective method of removing this noise is median filtering.

Embree [7] describes median filtering as a simple four step process:

1) A window of contiguous data is selected for each output point. This window can be reflected as adjacent time samples in a sequence, or as neighboring pixels in an image.

2) The data is sorted by signal value, from high to low.

3) The central value is selected as the median.

4) This median value is used as the filter output.

The advantage of this process is that sharp edges are preserved and not blurred as in averaging filters. A twist to this process is *conditional median filtering*. This technique avoids unnecessary corruption of the signal while maintaining the impulse removal capability of the previous filter. The process is as follows:

1) Select window.

2) Sort data from high to low.

3) Determine median.

4) The output of the filter is the median *if* the absolute value of the difference between the median and input is above a chosen threshold.

Otherwise, the input is passed through as the output.

Median filtering is highly dependent on the type of sort used. In image processing, max or min values are often used to alter shrink or grow bright areas in an image. This is known as erosion or dilation. Selection of output can also be used to control contrast.

Figures 21 and 22 [7] show the effect of median and conditional median filtering. These plots are the result of the program MEDIAN which is shown in Embree [7]. The "flat topping" evident in the figure is the result of median filtering a cosine wave. This is unwanted when filtering sinusoidal signals, but is acceptable for image row and column processing. It should be noted that the amount of "flat topping" is reduced through the use of conditional median filtering.

Richards [8] demonstrates a further usage of median filtering to enhance noncoherent radar data. The algorithm diagrammed in Figure 23a [8] shows the combined usage of median and inverse filtering. A median filter is used to segment real air-to-ground radar data into two classes, potential point targets and background. The separation of the data overcomes the contradictory processing goals of targets and background, namely to enhance target spikes while smoothing background clutter. Once separated, the two classes of data are passed through deconvolution filters. This inverse filtering removes the known spatial antenna pattern from the data to yield a high resolution image. The resulting data is then recombined to yield an improved return. A lower order median filter is also used as the background component constraint in Figure 23a. [8] This was done to further control clutter spikes and ringing in the background which would otherwise be accentuated by the far lower diagrams. Figure 23b [8] shows the effect of this process on millimeter-wave data. Notice that the target response is sharpened in each case while the background has been noticeably smoothed.

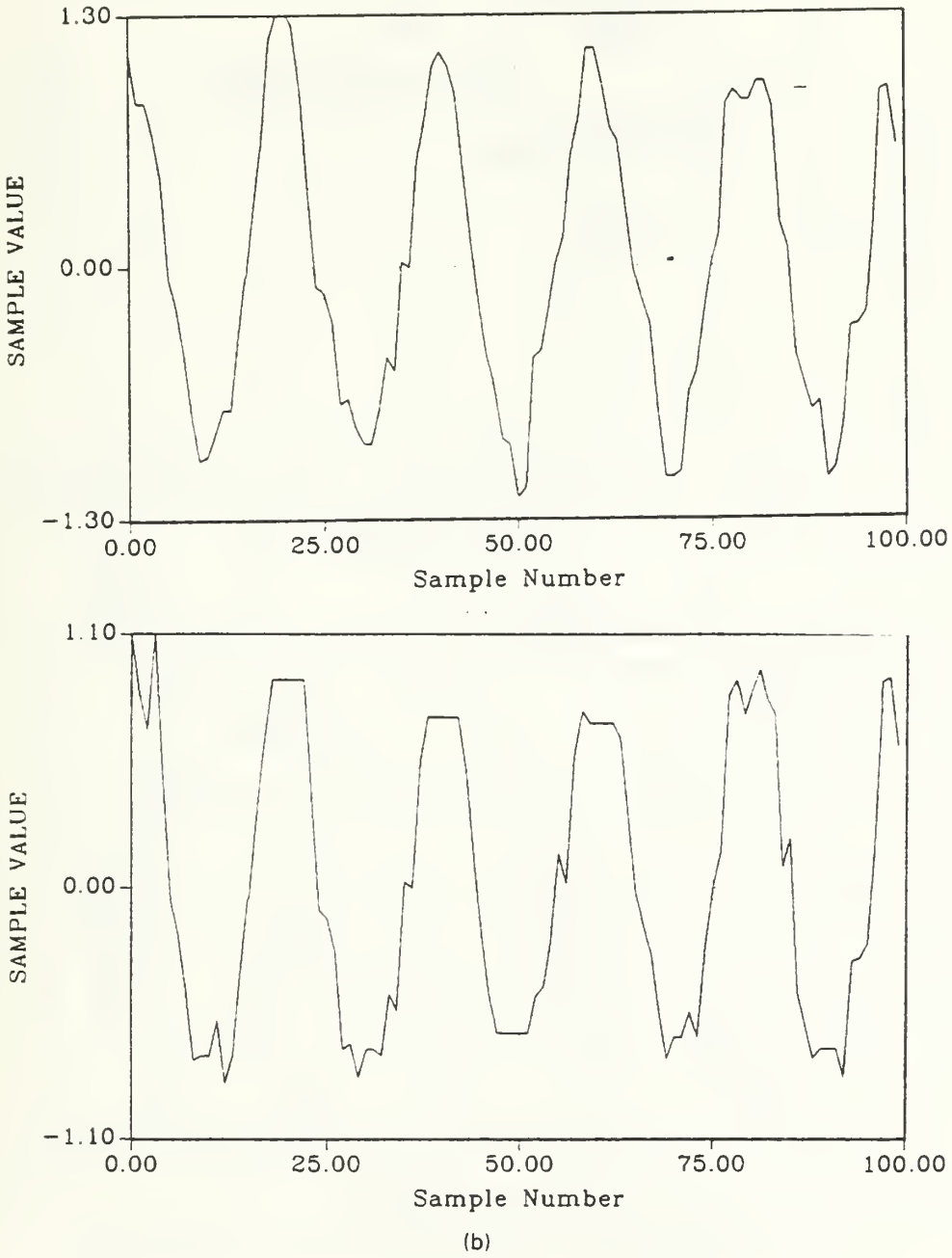


Figure 21. Median filtering with sort lengths of 3 and 9 threshhold = 0.

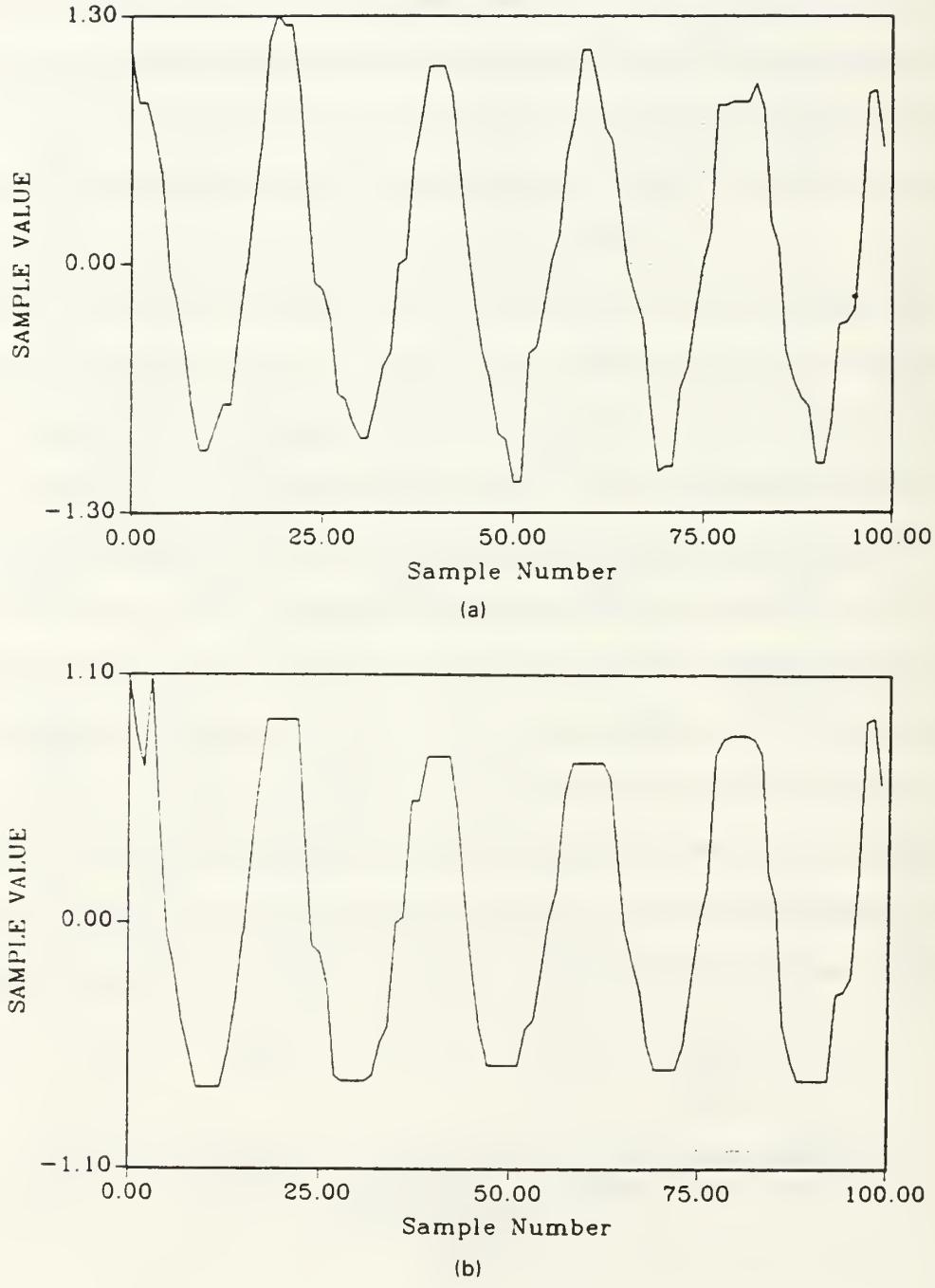
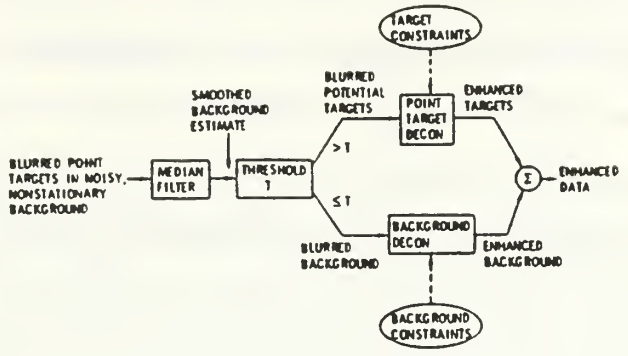
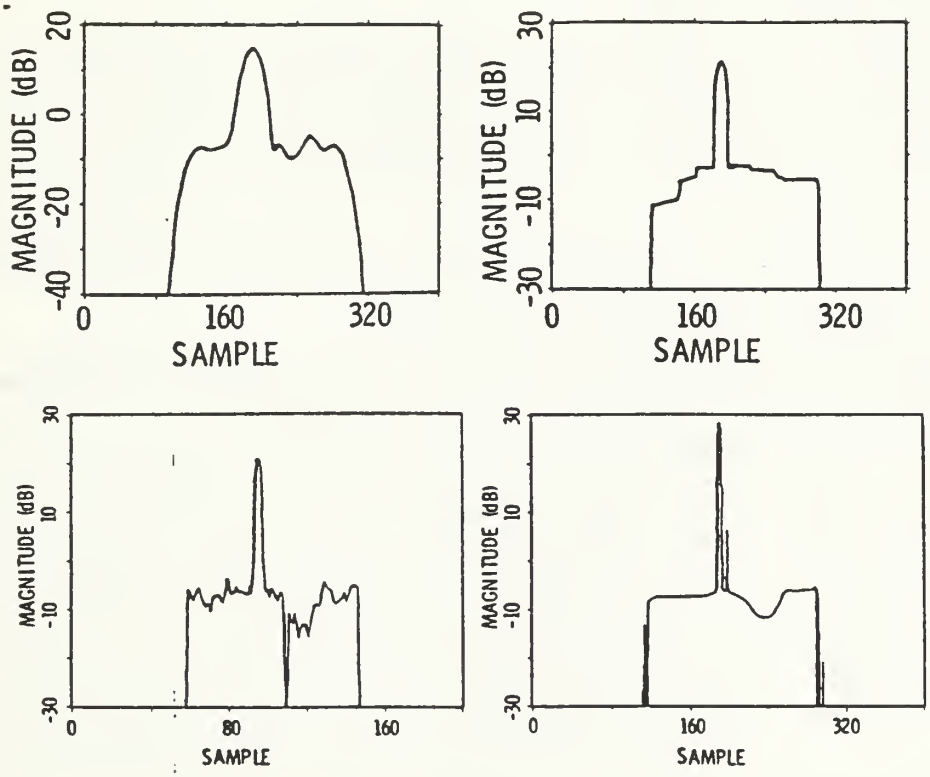


Figure 22. Conditional median filtering with sort lengths 3, threshhold .1, and sort length 9, threshhold .2.



(a)



(b)

Figure 23. An inverse filtering process utilizing median filtering (a), sampled data and filtered results shown below (b).

IV. CONCLUSIONS AND RECOMMENDATIONS

While the various methods of beam sharpening have been treated separately in this paper, in practice their implementation is often combined. This was demonstrated in the discussion of inverse filtering and the example from Richards [8] in which used median filters in the deconvolution process. In one Hughes system, the Hughes Advanced Discrimination Technique is used to detect the location of point scatterers, a modified inverse filter sharpens extended objects, median filtering is used to reduce impulse noise, edge detection algorithms are employed to identify roads and runways, and histogram flattening is implemented to improve image contrast and emphasize low contrast detail. When used in a system with high pulse compression, small mapping cells and improved resolution are obtained.

Through the combined usage of these techniques, beam sharpening ratios of 6:1 can be achieved for isolated corner reflectors with a 6 dB signal-to-noise ratio (SNR). With a 20 dB SNR, a 25:1 beam sharpening ratio can be achieved for corner reflectors and a 13:1 improvement for extended objects [5]. To illustrate the value of this capability, the following example will compare SAR and sharpened real beam resolutions for a fictional scenario.

Assuming operation at 12 GHz, the border of X and K_u bands, the system wavelength will equal .082 feet. If antenna length is assumed to be 10 feet, then the azimuth resolutions can be easily calculated. Stimson [1] gives the azimuth resolution of a SAR system as one half the physical antenna length. A comparison of resolution for real beam, sharpened real beam and SAR systems is shown in Table 6. From the unsharpened real beam results, it is not hard to see why these systems have been limited in application. The sharpened real beam columns, however, demonstrate "near SAR" quality. These examples do assume corner reflectors. The beam sharpening results for extended objects would only be approximately half as good.

TABLE 6
COMPARISON OF SAR AND REAL BEAM RESOLUTIONS

RANGE (in nmi)	SAR RESOLUTION	UNSHARPENED REAL BEAM	SHARPENED REAL BEAM (6 dB)	SHARPENED REAL BEAM (20 dB)
50	5 ft	2,461 ft	410 ft	98 ft
20	5 ft	984 ft	164 ft	39 ft
10	5 ft	492 ft	82 ft	20 ft

This level of improvement implies great potential for future mapping radars. Referring back to Table 1 on page 7, with a 20 dB SNR highways can be mapped to over 50 miles. "Road map" level of detail is available to over 20 miles. Taking into consideration that Scud launchers and other tactical targets are much larger than standard vehicles and are very well represented by corner reflectors, Table 6 demonstrates the capability to perform the mission proposed in the introduction of high-speed search and target localization. Further, these results were obtained by processing magnitude data only. In future efforts, these techniques may yield even better results through the processing of IQ data.

Figures 24 through 29, provided by Hughes, demonstrate the effectiveness the techniques discussed in this paper. These figures provide a visual comparison of the results of these techniques and SAR imaging. This data was collected by an F-15 in tests on another project. Efforts and equipment dedicated to mapping in future tests may also yield further improved results.

Resolution in range is equally impressive. As mentioned in Chapter II, bandwidth is the best determinant of performance. Since the 3dB bandwidth of a pulse is approximately the inverse of the pulse width, fine range resolution implies the need for high bandwidth. Continuing improvements in system bandwidth performance make it easier to achieve small cell size.

Continuing efforts to increase signal bandwidth and improve azimuth resolution yield numerous opportunities for further research. Of particular interest in the future will be continuing efforts to

improve the extended object capability of mapping systems. Image processing can play a great role in this effort. With stand-off weapons and associated tactics evolving at an extraordinary rate, radar mapping will have a large role to play in any future conflicts.

HUGHES

SAR Map: Edwards AFB

Figure 24. SAR Map:Edwards AFB.

HUGHES

RBGM Map: Edwards AFB
Before Beam Sharpening

Figure 25. RBGM Map:Edwards AFB, before beam sharpening.

HUGHES

RBGM Map: Edwards AFB
After Beam Sharpening

Figure 26. RBGM Map:Edwards AFB, after beam sharpening.

HUGHES

SAR Map: Corner Reflectors,
Rosamond Dry Lake

Figure 27. SAR Map:Corner Reflectors, Rosamond Dry Lake.

HUGHES

RBGM Map: Corner Reflectors
Before Beam Sharpening

Figure 28. RBGM Map:Corner Reflectors, before beam sharpening.

HUGHES

RBGM Map: Corner Reflectors
After Beam Sharpening

Figure 29. RBGM Map:Corner Reflectors, after beam sharpening.

APPENDIX

MATCHED FILTER OUTPUT AND THE AMBIGUITY DIAGRAM

A. THE AMBIGUITY DIAGRAM

One of the primary tools which will be used to evaluate these coding schemes is the ambiguity diagram. The ambiguity diagram represents the response of a matched filter to a waveform, and its doppler shifted versions, reflected from a point scatterer.

1. Development

The output of a matched filter can be represented as the cross correlation between a transmitted signal and the received signal. Neglecting noise, Skolnick [9] presents this as

$$\int S_r(t)S^*(t-T_R')dt \tag{A.1}$$

where

$S(t)$ is the transmitted signal

$S_r(t)$ is the received signal.

T_R' is the estimate of the time delay

The transmitted and received signals are assumed to be of the form

$$S(t) = u(t)e^{j2\pi ft} \tag{A.2}$$

$$S_r(t) = u(t-T)e^{j2\pi (f+fd)(t-T)} \tag{A.3}$$

where

f is the carrier frequency

f_d is the doppler shift

T is the time delay

Substituting these representations into (A.1) and simplifying by setting T=0, f=0 and -$T_R'=T_R$, the output of the matched filter becomes

$$X(T_R,f_d)= \int u(t)u^*(t+T_R)e^{j2\pi fdt} \tag{A.4}$$

where

u(t) is the complex modulation function

|u(t)| is the envelope of the real signal

With this representation, a positive f_d denotes a closing target while a positive T_R denotes a target beyond the reference delay. The ambiguity function is the magnitude of this equation squared.

The ambiguity diagram has a number of important properties as mentioned earlier. Several include:

$|X(T_R,f_d)|^2=|X(0,0)|^2=(2E)^2$ maximum value (A.5)

$|X(-T_R,-f_d)|^2=|X(T_R,f_d)|^2$ symmetry (A.6)

$\iint |X(T_R,f_d)|^2 dT_R df_d=(2E)^2$ constant total volume (A.7)

Eq. (A.7) is known as the waveform uncertainty principle and illustrates the problems and tradeoffs associated with waveform selection.

2. Usage

While the ambiguity diagram is of limited use as a practical design tool, it is helpful in examining the limitations and utilities of particular classes of radar waveforms. By studying the ambiguity diagram for a particular waveform, judgements can be made regarding its suitability for various applications.

Waveforms are selected to satisfy five major requirements. These are:
1) detection, 2) measurement accuracy, 3) resolution, 4) ambiguity and 5) clutter rejection. All of these characteristics can be visually evaluated using the ambiguity diagram. The ambiguity diagram

allows the user to intuitively evaluate waveforms at a glance. General characteristics are presented clearly without intensive mathematical formulation, as demonstrated below.

1) Detection is independent of the transmitted waveform and depends only on the level of energy which exceeds a chosen threshold at the origin. The maximum value of the ambiguity function occurs at the origin with a value of $(2E)^2$. The threshold must be set below this value, but high enough to avoid any secondary peaks.

2) Range accuracy is dependent on transmitted bandwidth (B) and doppler accuracy is dependent on the pulse width. The effects of these elements on the ambiguity function are shown in Figure 30. [9] Time delay (range) and frequency (doppler shift) are plotted on the horizontal axes with the ambiguity function magnitude plotted on the vertical axis. As will also be shown later, the volume under the ambiguity diagram remains constant. This is known as the radar waveform uncertainty law. Care must be taken in attempting to compress the diagram along either the range or frequency axis. (The waveform uncertainty principle forces a trade-off between range and doppler accuracy. Since the volume under the ambiguity function must remain constant, efforts to compress the diagram along one axis may result in a spreading of the diagram along the other.

3) Resolution is closely related to measurement accuracy. A waveform with good resolution will always have good accuracy. A waveform with good accuracy, however, may not have good resolution due to the presence of sidelobes. Waveform resolution is displayed as two overlapping ambiguity diagrams displaced by the range and Doppler spacing of the targets. While waveforms with narrow spikes may provide excellent accuracy and resolution in the main peak, they may also have large sidelobes. Sidelobes from a large target or ground clutter may contain enough energy to overpower a smaller target's central peak.

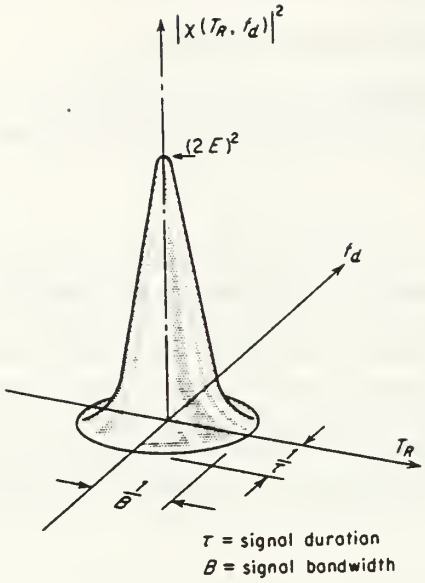
4) As alluded to above, peaks other than at the origin yield ambiguity in the measurement of range and doppler. Decisions with regard to pulse train length and PRF can be quickly evaluated by observing the presence and height of secondary peaks with respect to the main peak.

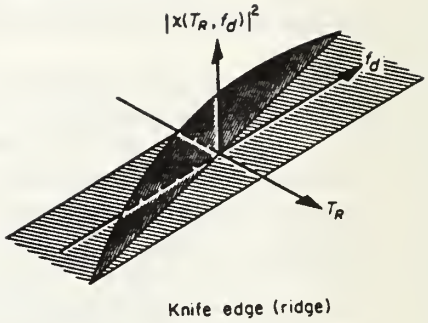
5) Clutter rejection ability is visualized by insuring ambiguity function peaks do not exist in doppler or range bins with high clutter.

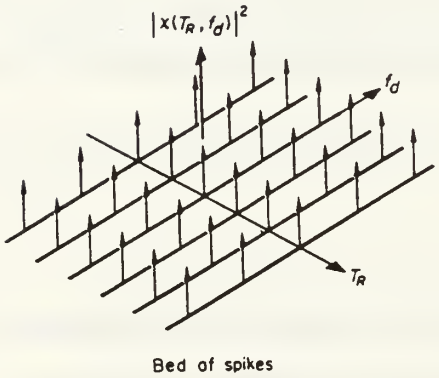
3. Types

There are five major classes of waveforms which give rise to three types of ambiguity diagrams. These types are the "knife edge", "bed of spikes" and "thumbtack". These are demonstrated in Figures 30. [9]

The "knife edge", or "ridge", diagram is obtained from a single pulse of sine wave. Linear frequency modulation rotates this ridge through the fd,T plane.







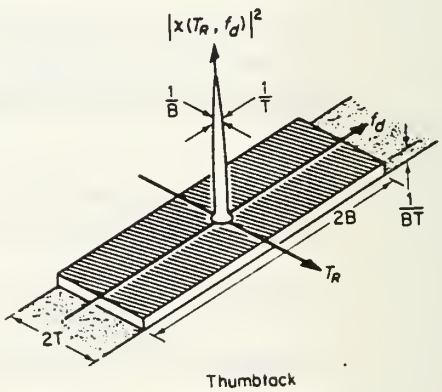


Figure 30. The ambiguity diagram, ideal approximation and three major classes.

The "bed of spikes" results from a periodic train of pulses. The form of each of the spikes is dependent upon the waveform of the individual pulses. This is representative of pulse-Doppler and MTI systems.

The "thumbtack" ambiguity diagram results from noise or pseudonoise codes. These codes include Pseudorandom, Barker and Polyphase codes. This thumbtack usually exists on a large pedestal of range-Doppler sidelobes which can be quite significant.

B. AMBIGUITY DIAGRAM PLOTTING PROGRAM

```
%  ============================================================
%  =  ambigUus                              Michael D. Anderson  =
%  =                                                             =
%  =     A menu driven plotting program to draw ambiguity       =
%  =  diagrams for various pulse compression code families.     =
%  =  AmbigUus is the main, calling menu subroutines and        =
%  =  performing the mesh plot.                                  =
%  ============================================================

clear
clg
CODE=[];

intro           % (Calls subroutine to provide instructions)
waveform        % (Calls subroutine to route user through menus.
                %  Menu programs implement selected codes and
                %  return for phase incrementation and plotting)


% (Scale the ambiguity diagram by selecting max reach of the
   doppler axis.)

s=input('Enter max doppler phase shift across the pulse in
         radians: ')

l=length(CODE);

for k=1:50;                    % (Implement the iterative phase
   step=(s*2)/(50-1);          %  shift across the code for each
   shift=(-s)+(k-1)*step;      %    increment  along  the  doppler
                               %  axis.)

   for j=1:l;
      v(j)=shift*j+CODE(j);
   end

   ua=i*CODE;
   va=i*v;
   ub=exp(ua);              % (Digitalizes the phase code values)
   vb=exp(va);              % (Digitalizes the shifted phase code)
   output=xcorr(ub,vb);     % (Performs the autocorrelation)
   x=abs(output);           % (The matched filter output)
   X=x.^2;                  % (The ambiguity function is the
   A(k,:)=X;                %  square of the filter function)
end
mesh(A)                     % (Plots the 3-D ambiguity diagram)
```

```
% ================================================================
%  =  intro                                  Michael D. Anderson    =
%  =                                                               =
%  =       Greets users and instructs them of the purpose          =
%  =    and usage of the program.                                  =
% ================================================================

clc
echo on
%       Welcome to the program ambigUus.  This program will enable
%
%   you to view the ambiguity diagram for a number of your favorite
%
%    waveforms.
%
%       The ambiguity diagram is a 3-D plot showing the response of
%
%   a matched filter receiver to reflections from point scatterers.
%
%    Range and frequency shifts are read along the horizontal axes,
%
%    as the matched filter response is plotted along the vertical
%
%    axis.
%
%        This plot can be used to evaluate the critical properties
%
%    of detection, range and doppler accuracy, range and doppler
%
%   resolution, range and doppler ambiguity, and clutter rejection.
%
%                    (press any key to continue)
echo off

pause;
end
```

```
%  ================================================================
%  =   waveform                          Michael D. Anderson      =
%  =                                                              =
%  =     A menu program allowing the user to select which         =
%  =   waveform to implement.                                     =
%  ================================================================

clg
clc

echo on
%       Two major classifications of radar waveforms are pseudo-
%
%   random binary codes and polyphase codes.  Both can be modeled
%
%   with this program.
%
%       Pseudo-random binary codes consist of Barker, Compound
%
%   Barker and Complementary codes.
%
%       Polyphase codes consist of Frank codes and the P-series
%
%   codes.  These are digital representations of LFM and step-chirp
%
%   waveforms.
%
%       1)      Barker code
%       2)      Compound Barker code
%       3)      Pseudorandom code
%       4)      Polyphase code
%       5)      Frank code
%       6)      P1 code
%       7)      P2 code
%       8)      P3 code
%       9)      P4 code
%       0)      quit program
%
%
echo off
n=input('Select the waveform type to process:  ');

if n==1
        barkform
elseif n==2
        compound
elseif n==3
        randform
elseif n==4
        polyform
elseif n==5
        frnkform
```

```
elseif n==6
        p1form
elseif n==7
        p2form
elseif n==8
        p3form
elseif n==9
        p4form
else
death

end
```

```
%  ================================================================
%  =   barkform                                Michael D. Anderson  =
%  =                                                                =
%  =     Allows user to select code length to plot and loads        =
%  =  the vector CODE with the code phases.                         =
%  ================================================================

clear
clc


echo on

%       The Barker codes are a special set of binary phase codes
%
%   which yield equal sidelobes after passage through a matched
%
%   filter.  These codes have pulse compression ratios equal to
%
%   their length.  The (+) and (-) in the sequences represent
%
%   phase shifts of zero and pi radians.  The known Barker codes
%
%   are as follows:
%
%       selection       code #          code sequence
%           1           2A              +-
%           2           2B              ++
%           3           3               ++-
%           4           4A              ++-+
%           5           4B              +++-
%           6           5               +++-+
%           7           7               +++--+-
%           8           11              +++---+--+-
%           9           13              +++++--++-+-+
%

echo off

n=input('Enter your choice, selection # ');
   if n==1,
         CODE=[0 pi];
      elseif n==2,
         CODE=[0 0];
      elseif n==3,
         CODE=[0 0 pi];
      elseif n==4,
         CODE=[0 0 pi 0];
      elseif n==5,
         CODE=[0 0 0 pi];
      elseif n==6,
         CODE=[0 0 0 pi 0];
```

```
      elseif n==7,
         CODE=[0 0 0 pi pi 0 pi];
      elseif n==8,
         CODE=[0 0 0 pi pi pi 0 pi pi 0 pi];
      elseif n==9,
         CODE=[0 0 0 0 0 pi pi 0 0 pi 0 pi 0];
   end
end
```

```
% ================================================================
%  =  compound                                Michael D. Anderson    =
%  =                                                                 =
%  =     Allows user to select inner and outer codes to generate =
%  =  compound barker codes.  Originated from "combined.m"       =
%  =  with Capt. Paul Ohrt, RCA for EC 4970, 1992.                =
% ================================================================

clear

BARKER2=[0,pi];
BARKER3=[0,0,pi];
BARKER4=[0,0,pi,0];
BARKER5=[0,0,0,pi,0];
BARKER7=[0,0,0,pi,pi,0,pi];
BARKER11=[0,0,0,pi,pi,pi,0,pi,pi,0,pi];
BARKER13=[0,0,0,0,0,pi,pi,0,0,pi,0,pi,0];

clc
echo on
%     This subroutine generates Compound Barker codes.
%
%  The known Barker codes have lengths of n= 2,3,4,5,7,11 or 13
%
%  You need to choose the length of the inner Barker code, "m"
%
%  and outer Barker code, "n".  The length will be m*n and the
%
%  PSL will be at the level of the shortest Barker code used.
%

echo off
m=input('ENTER the # of the inner Barker Code to use, m = ');
n=input('ENTER the # of the outer Barker Code to use, n = ');

%    Set selected inner Barker Code

if m==2
     in_CODE=BARKER2;
elseif m==3
     in_CODE=BARKER3;
elseif m==4
     in_CODE=BARKER4;
elseif m==5
     in_CODE=BARKER5;
elseif m==7
     in_CODE=BARKER7;
elseif m==11
     in_CODE=BARKER11;
elseif m==13
     in_CODE=BARKER13;
else
```

```
     compound   % starts sub program over if improper "m" entered
end

%    Set selected outer Barker Code

if n==2
     out_CODE=BARKER2;
elseif n==3
     out_CODE=BARKER3;
elseif n==4
     out_CODE=BARKER4;
elseif n==5
     out_CODE=BARKER5;
elseif n==7
     out_CODE=BARKER7;
elseif n==11
     out_CODE=BARKER11;
elseif n==13
     out_CODE=BARKER13;
else
     compound   % starts sub program over if improper "n" entered
end

%    Take Cosine of phase changes and develop "m in n" BARKER CODE

in_cos=cos(in_CODE);
out_cos=cos(out_CODE);

CODE=[];

for count=1:n
     INC=out_cos(count).*in_cos;
     CODE=[CODE, INC];
end

CODE=acos(CODE);
end
```

```
% ============================================================
%  =   randform                                Michael D. Anderson   =
%  =                                                                 =
%  =       Allows user to select code length to plot for a           =
%  =    polyphase code using shift register implementation.          =
%  =    Program originated from "random.m" developed with            =
%  =    Capt. Paul Ohrt RCA for EC4970, 1992.                        =
% ============================================================

clear
clc

echo on
%     When a large pulse compression ratio is required,
%
%  pseudorandom codes are often used.  These approximate noise
%
%  modulated signals with "thumbtack" ambiguity functions and
%
%  excellent range and doppler resolution.
%
%       A commmon method of generating these codes is the modulo 2
%
%  adder or exclusive-OR gate shift register decribed in Skolnik
%
%  ch 11.  That method is used here.  The only restrictions are
%
%  that the initial conditions of the register can not all be zero
%
%  (or the output will simply be a sequence of zeros), and that (p)
%
%  can not equal (q) (both shift inputs can not be taken from the
%                                                              n
%  same bit).  The length of the resulting code will be  2 -1.
%
%                       (hit any key to continue)
echo off

pause;
clc

n=input('ENTER the length of the shift register (# of bits): ');
p=input('ENTER the first feedback bit: # ');
q=input('ENTER the second feedback bit: # ');

echo on
%
%
%       To enter your initial conditions, simply enter a string of
%
%   ones and zeros.
%
```

```
%   ex.  Shift register length = 7.  Initial conditions = 1001100
%
%
echo off

IC=0;

flag=1;
while flag=1
   IC=input('Enter your string of length [n]: ');
   if length(IC)=n,
      flag=0
   end
end

CODE=[IC];                %  initial start point

MLS=2^n-1;                % maximal length sequence

for count = (n+1):MLS

     XOR=CODE(p)|CODE(q);      % OR function from Matlab

     if CODE(p)&CODE(q)==1;
          XOR=0;          % to ensure XOR conditions met
     else
          XOR=XOR;
     end

     CODE=[XOR,CODE];     % update code values from shift
end

CODE=acos(CODE);

end
```

```
%  ================================================================
%  =  polyform                              Michael D. Anderson   =
%  =                                                              =
%  =     Allows user to select code length to plot for a simple =
%  =  polyphase sequence and loads the CODE vector.  Program    =
%  =  originated from polyphas.m with Capt. Paul Ohrt RCA       =
%  =  for EC 4970, 1992.                                         =
%  ================================================================

clear
clc

echo on
%     This subroutine implements a POLYPHASE coding
%  sequence.  The POLYPHASE code developed here is a sequence
%  between 0 and pi in n increments.
%  You must specify how many increments are to be used.
%
echo off

n=input('ENTER the # of increments the Code is to use, n = ');

% (Generate polyphase vector)

k=0:(n-1);              % (increment vector)
phi=pi.*k.^2./n;
CODE=cos(phi)+i.*sin(phi);

end
```

```
% ================================================================
%  =   frnkform                                Michael D. Anderson   =
%  =                                                                 =
%  =     Allows user to select length of a frank code and loads      =
%  =  the vector CODE with the code phases.  From "frank.m"          =
%  =  generated in 1992 for EC 4970.                                 =
% ================================================================

clear
clc

echo on
%     This subroutine implements Frank Coding.  It is equal to
%
%  Barker coding in PSL, but is not limited in length.
%
%     Frank coding is a step-wise approximation of a
%
%  linear FM or "chirp" code.   To use this coding scheme,
%
%  enter the number of phase subdivisions N.  The PCR
%
%  will be N squared.
%
echo off

N=input('N =  ');
step = 2*pi/N;

for k = 1:N
   for j = 1:N
      count = (k-1)*N + j;          % Generate the Frank Code Matrix
      x(count) = (k-1)*(j-1);       % Store in vector form
   end
end

stepx = step*x;    % combine the Frank Code Matrix with the actual
                   % phase increment to generate the phase sampling

CODE=cos(stepx)+j.*sin(stepx);

end
```

```
% ================================================================
%  =   p1form                                  Michael D. Anderson   =
%  =                                                                 =
%  =    Allows user to select code length for P1 code and loads  =
%  =  the vector CODE with the code phases.  Derived from       =
%  =  frnkform.m 1992.                                               =
% ================================================================

clear
clc

echo on
%     This subroutine implements P1 coding, a Frank code
%
%  rearranged to avoid bandwidth limitations.  The ambiguity
%
%  diagram will be almost identical to the Frank code since both
%
%  are derived from the same waveform.
%
%     Enter the number of phase subdivisions N.
%
echo off

N=input('N =  ');
step = pi/N;

for k = 1:N
   for j = 1:N
      count = (k-1)*N + j;
      x(count) = ((j-1)*N+(k-1));
      x(count) = x(count)*(N-(2*j-1));
   end
end

stepx = (-1)*step*x;

CODE=cos(stepx)+j.*sin(stepx);

end
```

```
% ================================================================
%  =   p2form                                  Michael D. Anderson   =
%  =                                                                 =
%  =       Allows user to select code length for P2 and loads        =
%  =    the vector CODE with the code phases.                        =
% ================================================================

clear
clc

echo on
%     This subroutine implements P2 coding.  P2 coding
%
%  is a real vice complex version of P1.  Results should
%
%  be indistinguishable from Frank and P1 codes.
%
%     Enter the number of phase subdivisions N.
%
echo off

N=input('N =  ');
step = (pi/2)*((N-1)/N);

for k = 1:N
   for j = 1:N
      count = (k-1)*N + j;
      x(count) = step-(pi/N)*(k-1);
      x(count) = x(count)*(N+1-2*j);
   end
end

CODE=cos(x)+j.*sin(x);

end
```

```
% ================================================================
%  =   p3form                                  Michael D. Anderson   =
%  =                                                                 =
%  =       Allows user to select code length for P3 and loads        =
%  =    the vector CODE with the code phases.                        =
% ================================================================

clear
clc

echo on
%     This subroutine implements P3 coding.  This code
%
%  is derived from a linear chirp rather than the step
%
%  chirp as the Frank, P1 and P2 codes.  Because of this,
%
%  it will show the better doppler tolerance of the linear
%
%  frequency modulated (LFM) signal.
%
echo off

N=input('N =  ');
step = pi/N;

   for j = 1:N
      x(j) = (j-1)^2;
   end

stepx = step*x;

CODE=cos(stepx)+j.*sin(stepx);

end
```

```
% ================================================================
%  =   p4form                                  Michael D. Anderson   =
%  =                                                                 =
%  =       Allows user to select code length of a P4 and loads       =
%  =    the vector CODE with the code phases.                        =
% ================================================================

clear
clc

echo on
%     This subroutine implements P4 coding.  P4 codes
%
%  are simply P3 codes rearranged to insure that the
%
%  greatest phase shift between code bits occurs at the
%
%  edges of the sequence rather than at the center.  This
%
%  identicle in conncept to the reworking of Frank codes to
%
%  P1 to remove the bandwidth limitations.
%
%     Enter N, the number of phase subdivisions to use.
%
echo off

N=input('N =  ');
step = pi/N;

   for j = 1:N
      x(j) = (j-1);
      x(j) = (step*x(j)^2)-(pi*x(j));
   end


CODE=cos(x)+j.*sin(x);

end
```

LIST OF REFERENCES

1. Stimson, G.W., "Introduction to Airborne Radar," Hughes Aircraft Co., El Segundo, CA, 1983.

2. Nathanson, F.E., *Radar Design Principles, second ed.*, McGraw-Hill, Inc. NY, 1991.

3. Akita, R.M. "An Investigation of the Narrow-band and Wide-band Ambiguity Functions for Complementary Codes," Master's Thesis, Naval Postgraduate School, Monterey, CA, 1968.

4. Lewis, L. and Kretchmer, F.F. Jr., "New Polyphase Pulse Compression Waveforms and Implementation Techniques," *Advances in Radar Techniques*, ed. J. Clarke Peter Peregrinus Ltd., London, UK., 1985. pp. 496-500.

5. Unpublished presentation materials, Hughes Aircraft Advanced Systems.

6. Pratt, W. K., *Digital Image Processing*, John Wiley & Sons, Inc., NY, 1991.

7. Embree, P.M. and Kimble, B., *C Language Algorithms for Digital Signal Processing*, Prentice Hall, Englewood Cliffs, N.J., 1991.

8. Richard, M.A., Morris, C.E. and Haynes, M.H., "Iterative Enhancement of Noncoherent Radar Data," Proceedings 1986 IEEE International Conference on Acoustics, Speech and Signal Processing, April 1986.

9. Skolnick, M.L., *Introduction to Radar Systems*, McGraw-Hill Publishing Co., NY, 1980.

INITIAL DISTRIBUTION LIST

1.	Defense Technical Information Center Cameron Station Alexandria, Virginia 22304-6145	2
2.	Library, Code 52 Naval Postgraduate School Monterey, California 93943-5000	2
3.	Chairman, Code EC Department of Electrical and Computer Engineering Naval Postgraduate School Monterey, California 93943-5000	1
4.	Dr. Gurnam Gill, Code EC/Gl Department of Electrical and Computer Engineering Naval Postgraduate School Monterey, California 93943-5000	3
5.	Dr. David C. Jenn, Code EC/Jn Department of Electrical and Computer Engineering Naval Postgraduate School Monterey, California 93943-5000	1
6.	T.D. (Buzz) Elliot Hughes Aircraft Radar Systems Group Bldg. R8/N511 P.O. Box 92426 Los Angeles, California 90009-2426	2
7.	LT Michael D. Anderson 789 Cornwallis Drive Mount Laurel, New Jersey 08054	4