1986

The Design and Implementation of Surface Acoustic Wave Devices for Linear FM and Nonlinear FM Pulse Compression Technique for Radar Applications

James C. Walker
University of Central Florida

THE DESIGN AND IMPLEMENTATION OF SURFACE ACOUSTIC WAVE DEVICES FOR LINEAR FM AND A NEW NON-LINEAR FM PULSE COMPRESSION TECHNIQUE FOR RADAR APPLICATIONS

BY

JAMES C. WALKER
B.S.E., University of Central Florida, 1985

THESIS

Submitted in partial fulfillment of the requirements
for the degree of Master of Science in Engineering
in the Graduate Studies Program of the College of Engineering
University of Central Florida
Orlando, Florida

Fall Term
1986

ABSTRACT

The purpose of this work is to design, fabricate, and compare dispersive SAW devices using both linear FM and a new non-linear FM scheme. This new non-linear FM scheme uses the Blackman function as the modulating signal of the FM waveform. Up-chirped and V-chirped devices for both linear and the new non-linear FM scheme and their corresponding matched filters are compared.

Design considerations are discussed in detail. An efficient sampling algorithm (which can also be applied to other non-linearly modulated FM waveforms) developed to facilitate the design of the SAW devices is presented.

ACKNOWLEDGEMENTS

I would like to thank my entire committee, especially my chairman Dr. Madjid Belkerdid, for their support and guidance throughout this endeavor. A special thanks goes to Mr. Carl Bishop for his assistance in the fabrication of the devices and to Mr. Samuel Richie for his help in the development of the necessary software to write the magnetic tape for photomask production. Additionally, I would like to thank two very good friends, Keith Lindsay for all of the encouragement and advice he provided and Ron Booher who originally introduced the subject of this thesis.

Most importantly, I thank my wife, Denise, for the love and support she has given me during this trying period; and my two children, Jennifer and Kevin, for the inspiration they provide me.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER I

INTRODUCTION

Radar Fundamentals

In its simplest form, a radar system transmits a signal and waits for a reflected signal (echo) from a possible target. If an echo does exist, it will be similar to the transmitted signal, though exhibiting some differences. Information about the target's location, size, velocity, and direction of movement can be extracted from these differences. If on the other hand, there is no echo, it is assumed that no target exists (in that direction) and the signal is transmitted in another direction (Tzannes 1985).

The radar equation is used to determine the maximum range at which a target can be detected. If the system emits a signal with initial average power of P_T watts and meets an object at a distance R from the radar, the return power of the echo, P_R, is given by

$$P_R = K \frac{P_T}{R^4} \qquad (1)$$

where K depends on various other parameters of the system such as target cross section and antenna gain. A useful form of the radar equation is found by denoting the minimum detectable power by S_{min}, and solving for R,

$$R_{max} = \left(\frac{KP_T}{S_{min}}\right)^{\frac{1}{4}} \tag{2}$$

where R_{max} represents the maximum range of the system (Tzannes 1985).

Range information is found by transmitting a pulse and measuring the time delay, Δt, between the transmitted and received pulse. Since electromagnetic energy travels at the speed of light, c, the distance to the target will be

$$R = \frac{c\Delta t}{2} \tag{3}$$

where the factor of 2 accounts for the round trip (Cook and Bernfeld 1967).

To measure this delay, a distinct point of reference is required. Though the beginning or end of the pulse may have sharp edges for reference, these edges become rounded or obscured when band-limited or in the presence of noise. To minimize the range error, the time cross-correlation function of the incoming echo signal with the original pulse is taken using a matched filter, which maximizes the signal to noise ratio (SNR) in the presence of white Gaussian noise. This implies that a distinct point of reference must be present at the output of the matched filter. Good range resolution requires the output of the matched filter to have a high sharp peak, and only a single peak, so that a number of targets can be distinguished from one another (Rihaczek 1969).

A very narrow pulse would possess the characteristics needed for good range resolution, but due to power limitations in the transmitter, the maximum range detection capability would be limited. Hence, in order to meet range requirements, the radar system designer seems to be faced with a trade-off. This trade-off is reduced using pulse compression techniques, which is another significant advantage of using matched filters. The pulse can actually be made as wide as necessary to meet range requirements, then coded with wideband modulation information to meet the range resolution requirement (Cook and Bernfeld 1967).

If a target is moving toward a radar system, then the frequency of the return signal will be higher than that of the transmitted signal. Similarly, if the target is moving away from the radar, the frequency of the return signal will be lower than that of the transmitted signal. This phenomenon is known as the Doppler effect and the amount of change in frequency (f_D), is used to compute the target's radial velocity (ν_r). If ν_r is much less than the speed of light, the doppler frequency is approximated by

$$f_D = \frac{2\nu_r}{C} f_t \qquad (4)$$

where f_t represents the transmission frequency (Wheeler 1967).

Linear FM

The most common form of pulse compression used in modern radar systems is linear frequency modulation (chirp), which is realized by linearly varying the carrier frequency of the FM waveform during the interval of the pulse. Chirp radar provides good range and velocity measurements, as well as resolution (Tzannes 1985). Range requirements (within limits) can be met by increasing the width of the transmitted pulse. Using matched filters, this can be accomplished without sacrificing bandwidth.

Though it would seem that chirp offers all the desired characteristics of a radar signal, this is not the case. It measures range very well when the target is not moving (no Doppler shifts), but when the target is moving, range cannot be calculated directly. Though this range error can be accounted for when there is only one target, it becomes a problem when there are a number of targets within one pulse period with differing velocities. Therefore, chirp radar is most suitable for applications where the expected differences in velocity are relatively small during any single pulse period (Cook and Bernfeld 1967). Another disadvantage of the chirp radar is that the autocorrelation time sidelobes are relatively high, which could cause range resolution problems (Booher 1985).

Non-Linear FM

Many approaches have been investigated in an effort to minimize the problems inherent to linear FM. One such approach is to vary the frequency of the transmitted pulse in a non-linear fashion (non-linear FM). Work done by Booher suggests there may be some merit in varying the pulse frequency using the Blackman function as the modulating waveform. He showed that under certain conditions, this new non-linear FM could perform better than linear FM when range resolution is of primary concern (Booher 1985). This pulse is further investigated to see if a wider range of velocities can be detected without introducing significant range error due to Doppler shifts.

SAW Devices

Surface acoustic wave (SAW) technology offers a means of processing complex waveforms onto devices that are much smaller and more reliable than previous techniques. The planar nature of SAW devices allows them to be fabricated using standard lithography techniques used by the semiconductor industry. This process is highly repeatable and relatively inexpensive.

Impulse response model design techniques (Hartmann, Bell and Rosenfeld 1973) are used in order to provide a straightforward approach to producing sample devices which implement both linear FM and the new non-linear FM pulse compression waveforms.

Design considerations and implementation procedures are presented in detail.

CHAPTER II

OBJECTIVE OF PROPOSED WORK

The objective of this thesis is to further investigate the characteristics of the new non-linear FM waveform introduced by Booher. SAW devices will be designed and photomasks will be generated using both linear FM and the new non-linear FM function. Additionally, V-chirped devices using both linear FM and the new non-linear FM function will be investigated. Design considerations will be discussed and computer aided design software will be written. Comparisons will be made on the impulse response, frequency response, and matched filter responses of the filters. The Doppler effect on both linear FM and the new non-linear FM waveform will be simulated.

CHAPTER III

WAVEFORM DESCRIPTIONS

Using impulse response design techniques, SAW device design is straightforward if the time waveforms can be sampled accurately (Hartmann, Bell, and Rosenfeld 1973). This chapter is concerned with the derivation of the general form of the linear and new non-linear FM waveforms. Though up-chirped, down-chirped and V-chirped SAW devices are considered, symmetry arguments are used (Chapter IV) so only the up-chirped forms of the equations are necessary. Additionally, plots of the spectra, matched filter outputs, and spectra of the matched filter outputs for both linear FM and the new non-linear FM are presented. With the exception of figures 1 and 2, the plots provided in this chapter were accomplished using the FFT and graphic capabilities of the surface acoustic wave computer aided design program (SAWCAD) developed at the University of Central Florida (Malocha and Richie 1984).

FM Waveforms

In general, an FM waveform, $X_{FM}(t)$, can be represented by the equation (Ziemer and Tranter 1985)

$$X_{FM}(t) = A \cos \left[\omega_c t + \left(K_f \int_o^t m(\tau) d\tau + \phi_o\right)\right] \tag{5}$$

where:

$m(\tau)$ = the modulating signal

A = the peak amplitude of carrier

ω_c = the angular frequency when m(t) is zero

K_f = the frequency deviation constant

ϕ_o = the initial phase

The phase deviation, $\phi(t)$, of $X_{FM}(t)$ is

$$\phi(t) = K_f \int_o^t m(\tau)d\tau + \phi_o \quad (6)$$

The instantaneous frequency, $\omega_i(t)$, of $X_{FM}(t)$ is found by taking the derivative of the bracketed term in equation (5) with respect to time which yields

$$\omega_i(t) = \omega_c + K_f\, m(t) \quad (7)$$

Linear FM

Linear FM is an FM waveform whose instantaneous frequency changes in a linear fashion with respect to time. Figure 1 shows the modulating signal used to accomplish this. Note, the remaining equations (8-16) in this chapter are only valid from 0 to T since they represent pulses, not continuous waveforms. The linear FM modulating signal, $m_{LFM}(t)$, is given by the equation

$$m_{LFM}(t) = \frac{at}{T} \quad (8)$$

The instantaneous frequency of the linear FM waveform, $\omega_{LFM}(t)$ is found by substituting equation (8) into equation (7) to yield

$$\omega_{LFM}(t) = \omega_c + \frac{aK_f t}{T} \quad (9)$$

The phase deviation of the linear FM waveform, $\phi_{LFM}(t)$, is found by substituting equation (8) into equation (6) and performing the integration such that

$$\phi_{LFM}(t) = \frac{aK_f}{2T} t^2 + \phi_o \quad (10)$$

Now, the linear FM waveform, $X_{LFM}(t)$, can be written by substituting equation (10) into equation (5), which yields

$$X_{LFM}(t) = A \cos (\omega_c t + \frac{aK_f}{2T} t^2 + \phi_o) \quad (11)$$

Non-Linear FM

This new non-linear FM uses a shifted form of the Blackman function as the modulating signal of the FM waveform (Booher 1985). Shown in Figure 2, the modulating signal of the new non-linear FM waveform, $m_{NLFM}(t)$, is represented by

$$m_{NLFM}(t) = a\{0.43 + 0.5 \cos[\frac{\pi(t-T)}{T}] + 0.07 \cos[\frac{2\pi(t-T)}{T}]\} \quad (12)$$

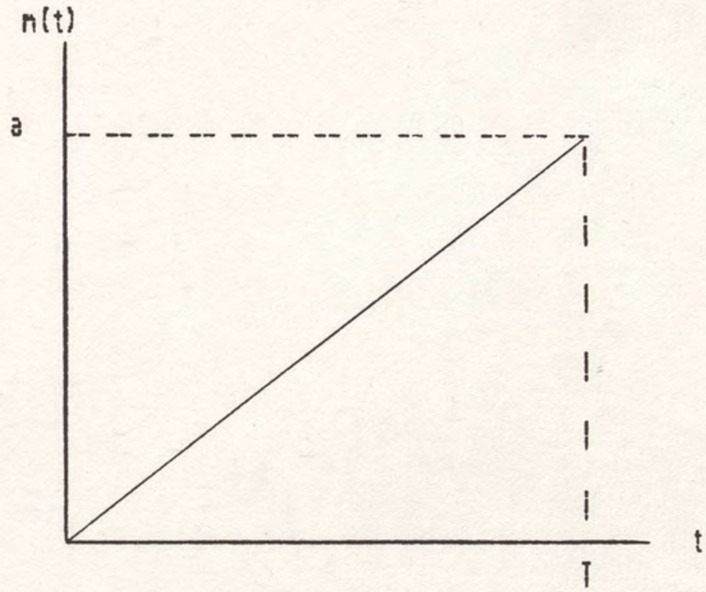

Figure 1. Modulating signal for linear FM.

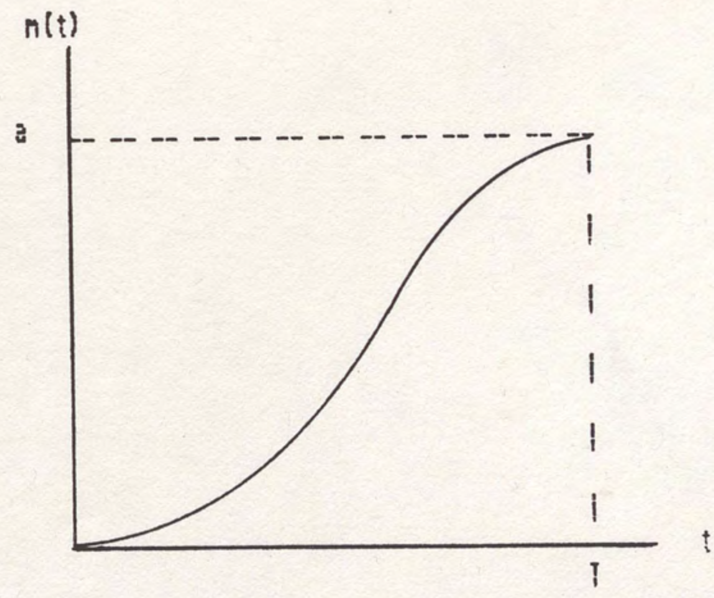

Figure 2. Modulating signal for new non-linear FM.

The instantaneous frequency of the new non-linear FM waveform, $\omega_{NLFM}(t)$, is found by substituting equation (12) into equation (7) to yield

$$\omega_{NLFM}(t) = \omega_c + aK_f\{0.43 + 0.5 \cos[\frac{\pi(t-T)}{T}] + 0.07 \cos[\frac{2\pi(t-T)}{T}]\} \quad (13)$$

The phase deviation of the new non-linear FM waveform, $\phi_{NLFM}(t)$, is found by substituting equation (12) into equation (6) such that

$$\phi_{NLFM}(t) = aK_f \int_o^t \{0.43 + 0.5 \cos[\frac{\pi(\tau-T)}{T}] + 0.07 \cos[\frac{2\pi(\tau-T)}{T}]\}d\tau + \phi_o \quad (14)$$

Performing the required integration yields

$$\phi_{NLFM}(t) = aK_f\{0.43t + \frac{0.5T}{\pi} \sin[\frac{\pi(t-T)}{T}] + \frac{0.07T}{2\pi} \sin[\frac{2\pi(t-T)}{T}]\} + \phi_o \quad (15)$$

Finally, the new non-linear FM waveform, $X_{NLFM}(t)$, can be written by substituting equation (15) into equation (5), which yields

$$X_{NLFM}(t) = A \cos\Big\{\omega_c t + aK_f\{0.43t + \frac{0.5T}{\pi} \sin[\frac{\pi(t-T)}{T}] + \frac{0.07T}{2\pi} \sin[\frac{2\pi(t-T)}{T}]\} + \phi_o\Big\} \quad (16)$$

Design Characteristic Waveforms

In general, it is very difficult (if not impossible) to physically see the varying frequency of practical FM waveforms.

For this reason, computer plots of the actual designed chirped waveforms will not be included, though a hypothetical example of an up-chirped linear FM waveform (Figure 3) and a V-chirped linear FM waveform (Figure 4) has been included for pedagogical purposes. The waveform of Figure 3 chirps from 40 MHz to 100 MHz in approximately 0.1 usec, while the waveform of Figure 4 chirps from 40 MHz to 100 MHz then back to 40 MHz in approximately 0.1 usec. The waveforms of the actual designs chirped from 60 MHz to 80 MHz in approximately 3 usec.

Frequency Response of the Filters

The spectrum of the linear FM waveform can be represented in closed form (Cook and Bernfeld 1967). Normalized amplitude plots of the linear and new non-linear FM spectra are shown in figures 5 and 6, respectively. These plots were accomplished by performing an FFT on the actual time waveforms used in the SAW device design.

The frequency responses of the receiving (matched) filters were found by multiplying the impulse response of the transmitting filter with that of its corresponding matched filter. Figures 7 and 8 show the frequency responses of the matched filter outputs. It should be noted that since the new non-linear FM actually has two peaks, linear FM would perform better in terms of velocity resolution.

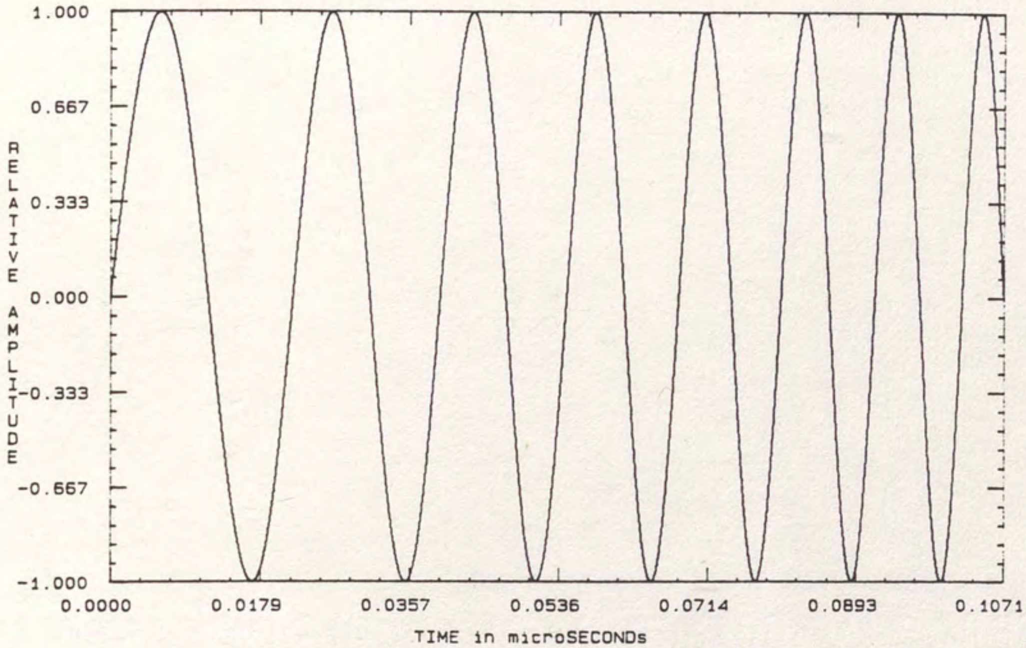

Figure 3. Up-chirped FM waveform.

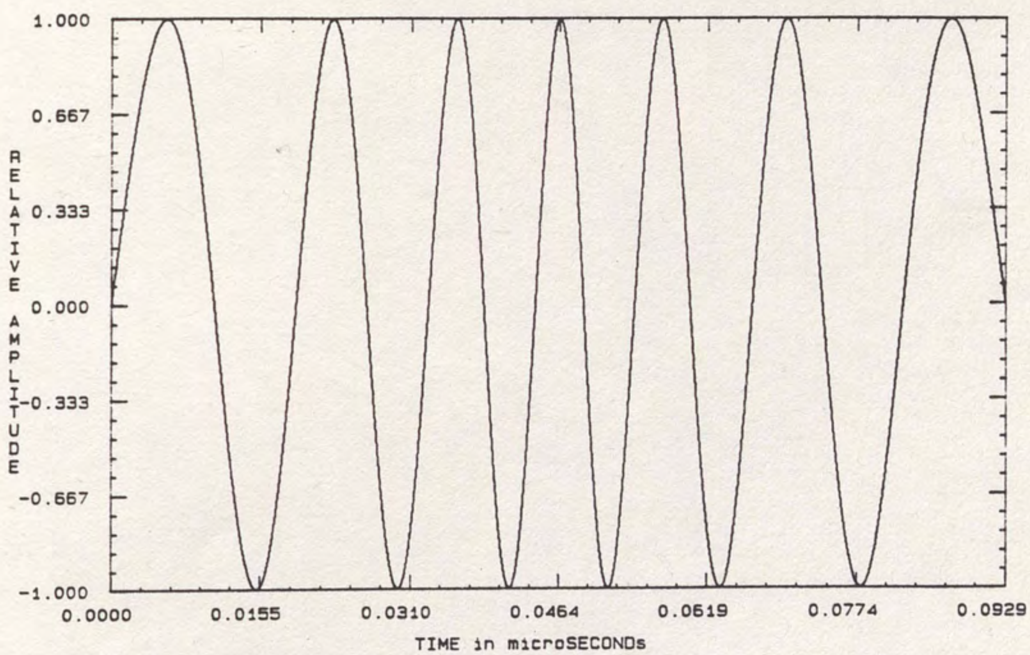

Figure 4. V-chirped FM waveform.

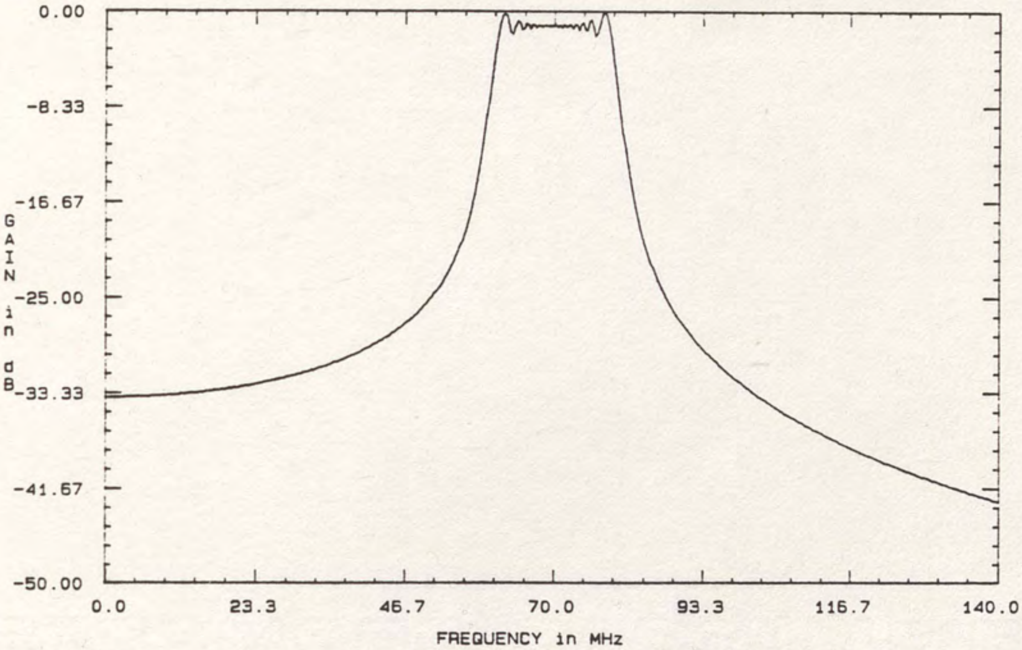

Figure 5. Frequency response of an up or down chirped linear FM waveform.

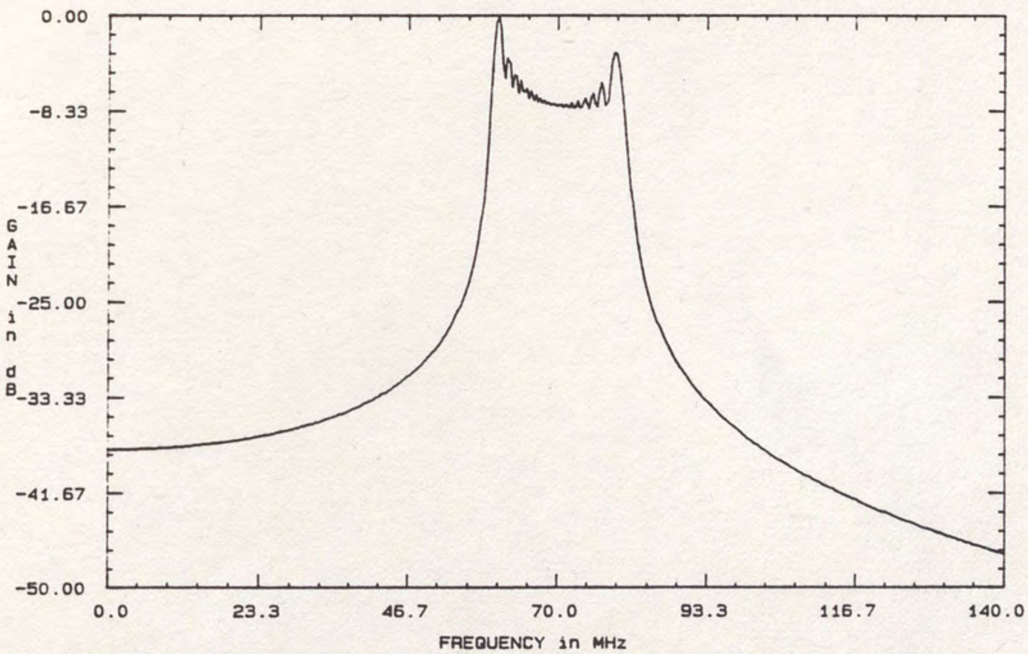

Figure 6. Frequency response of an up or down chirped non-linear FM waveform.

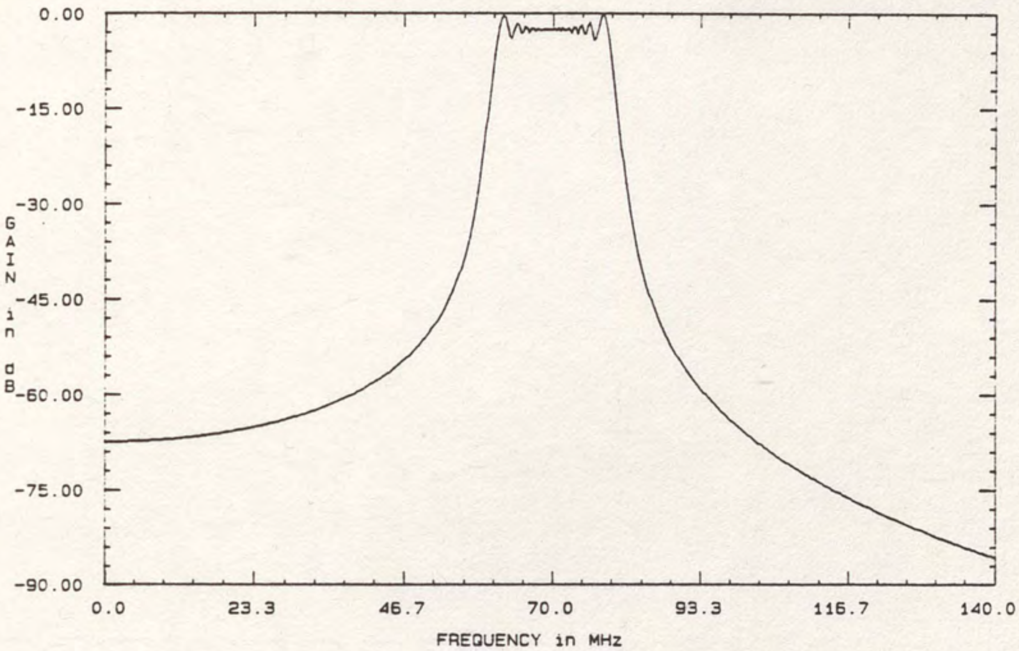

Figure 7. Frequency response of the linear FM matched filter pair.

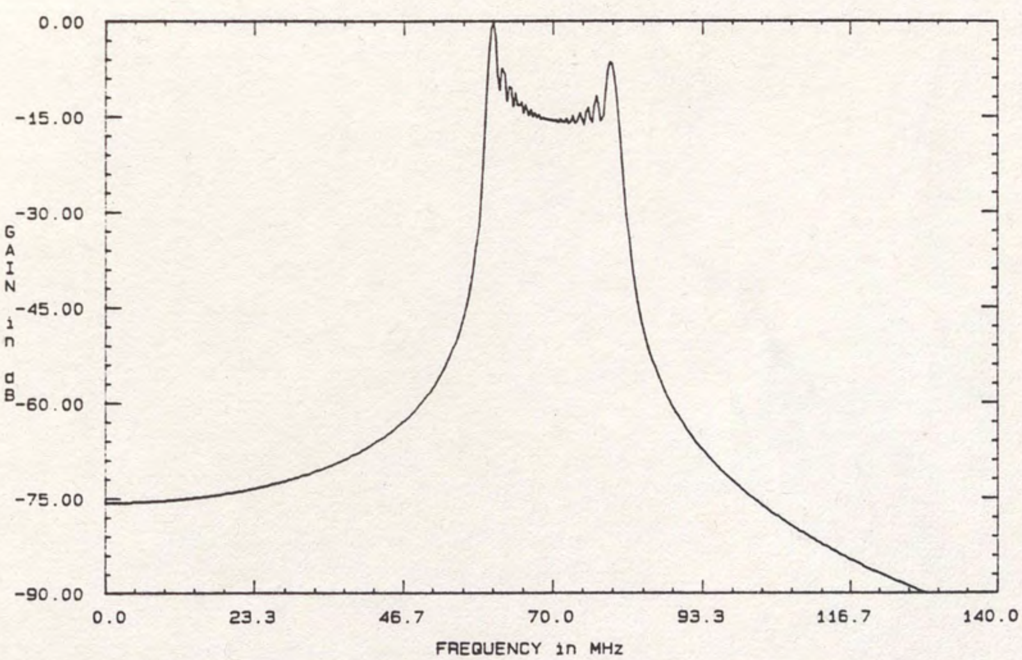

Figure 8. Frequency response of the new non-linear FM matched filter pair.

Time Responses of the Matched Filters

The time response of the matched filters (the autocorrelation function) is found by taking the inverse FFT of the corresponding matched filter frequency response. Matched filter outputs for linear and the new non-linear FM waveforms are presented in figures 9 and 10 respectively. The sidelobe level of the linear FM is approximately 13.7 dB down, while that of the non-linear FM is only 7.6 dB down. In this case, the linear FM provides better range resolution.

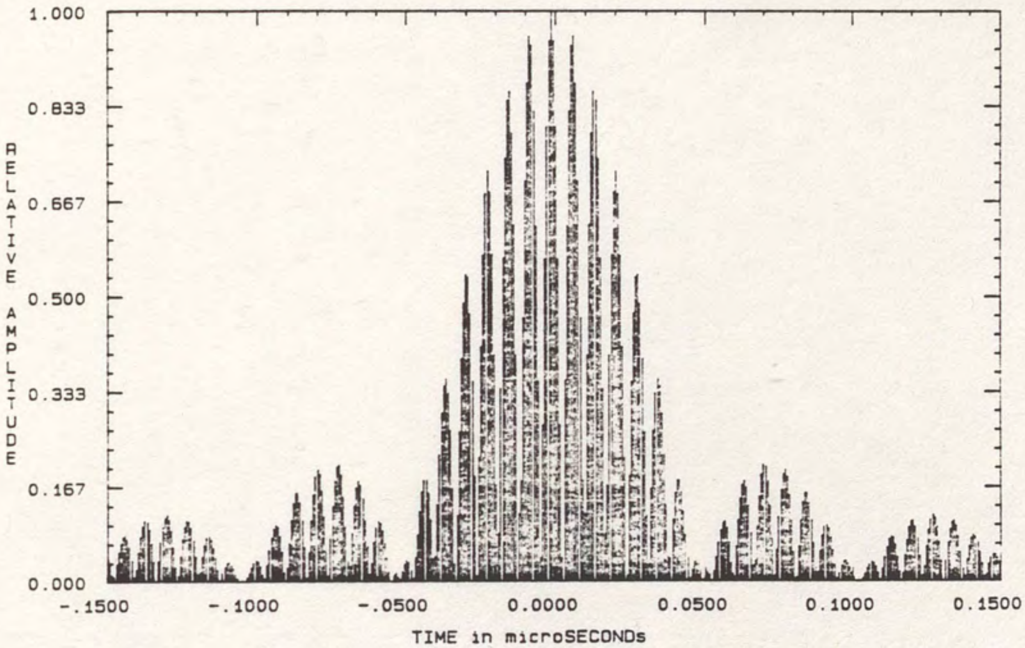

Figure 9. Time response of the linear FM matched filter pair.

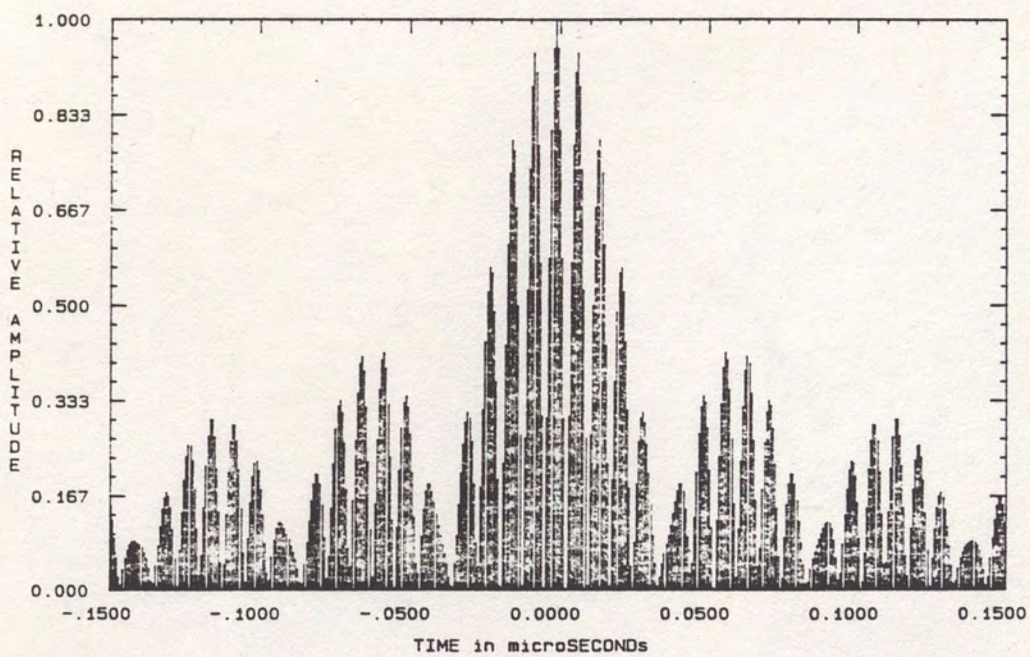

Figure 10. Time response of the new non-linear FM matched filter pair.

CHAPTER IV

DOPPLER SIMULATION

The Doppler effect can be simulated on a computer by shifting the center frequency of the transmitted pulse and processing the resulting signal through the original matched filter. This was accomplished by taking the FFT of the frequency shifted up-chirped waveform, and multiplying it with the FFT of the original down-chirped waveform. An inverse FFT was then performed on the product to yield the desired Doppler shifted matched filter response. The simulation was performed on both linear FM and the new non-linear FM waveforms for several Doppler frequency shifts (f_D), and on the V-chirped versions of the waveforms.

Distortion effects on the output of the original matched filter caused by Doppler shift conditions are loss in peak amplitude and a time shift of the waveform. For linear FM, the amplitude degradation is essentially bounded by a triangle with its baseline extending from -T to +T. This triangle is the auto-correlation of the rectangular envelope used and this relationship is a property of the linear FM waveform. For f_D less than half the total chirped bandwidth, a good approximation for the time shift, t_s, of the linear FM waveform is (Cook and Bernfeld 1967)

$$t_s = - \frac{f_D}{\Delta f} T \qquad (17)$$

Both of these distortion effects can be easily verified from the linear FM Doppler simulation presented here.

All of the following plots are of the matched filter output for the Doppler frequency indicated. The center frequency of the receiving filter was set at 70 MHz, the dispersion time at 3 usec, and the bandwidth at 20 MHz. The center frequency of the Doppler shifted pulse was set at 70 MHz plus the indicated Doppler frequency shift, the dispersion time at 3 usec, and the bandwidth at 20 MHz.

Figures 11 and 12 are for a Doppler frequency shift of 0, for linear FM and the new non-linear FM respectively. The following plots in this simulation are scaled to the peak amplitudes of these two in order to show the relative amplitude degradation due to the Doppler shifts.

Figures 13, 15, 17, 19, and 21 represent Doppler frequency shifts of +1, +3, +5, +7, and -5 MHz, respectively, for the linear FM. The resulting Doppler distorted waveforms are as predicted by the triangular envelope for amplitude degradation and equation (17) for the time shift. The results are summarized in Table 1.

The Doppler distorted waveforms for the new non-linear FM are presented in figures 14, 16, 18, 20, and 22 for the same

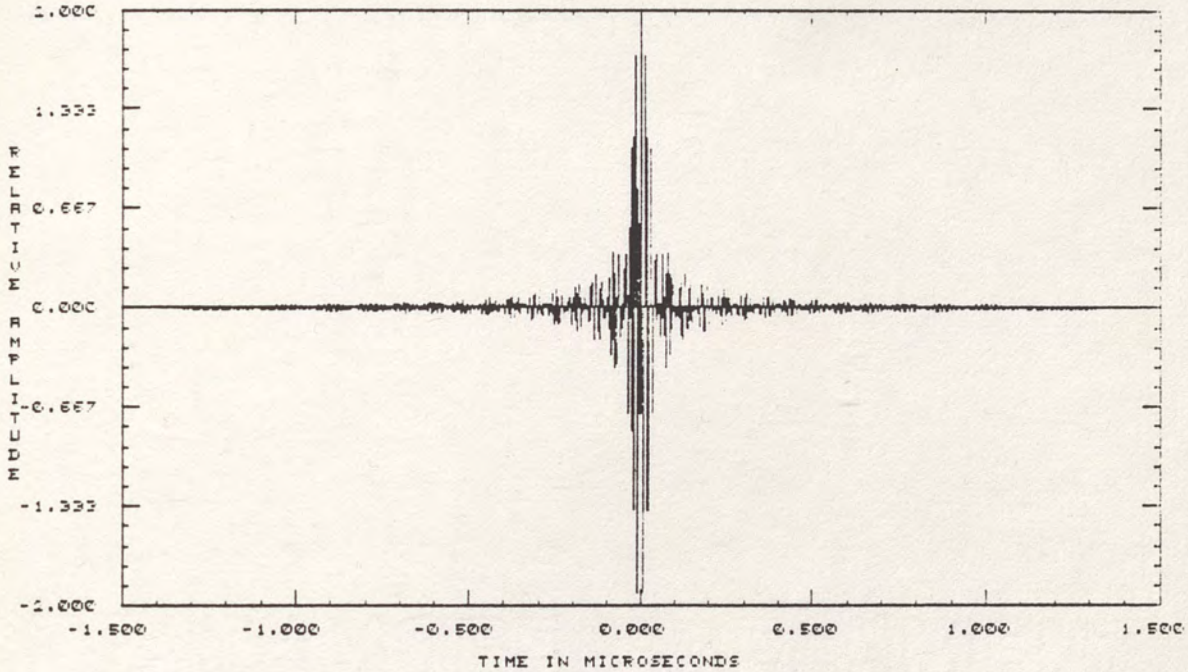

Figure 11. Linear FM matched filter output with no Doppler frequency shift.

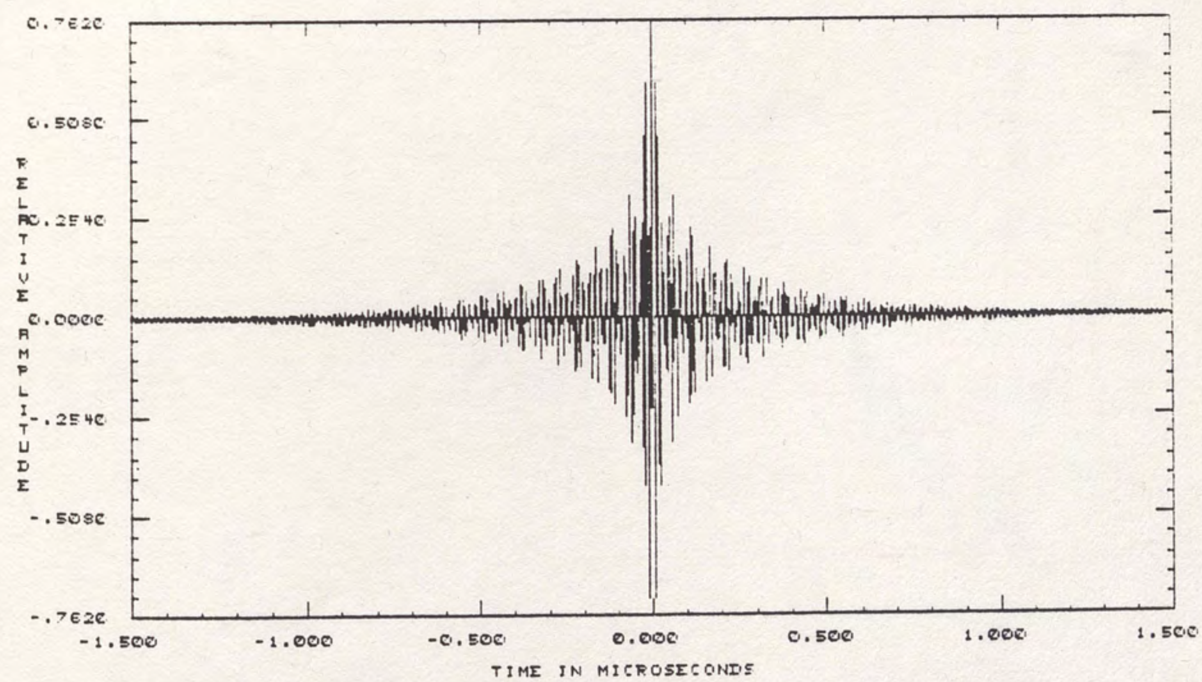

Figure 12. New non-linear FM matched filter output with no Doppler frequency shift.

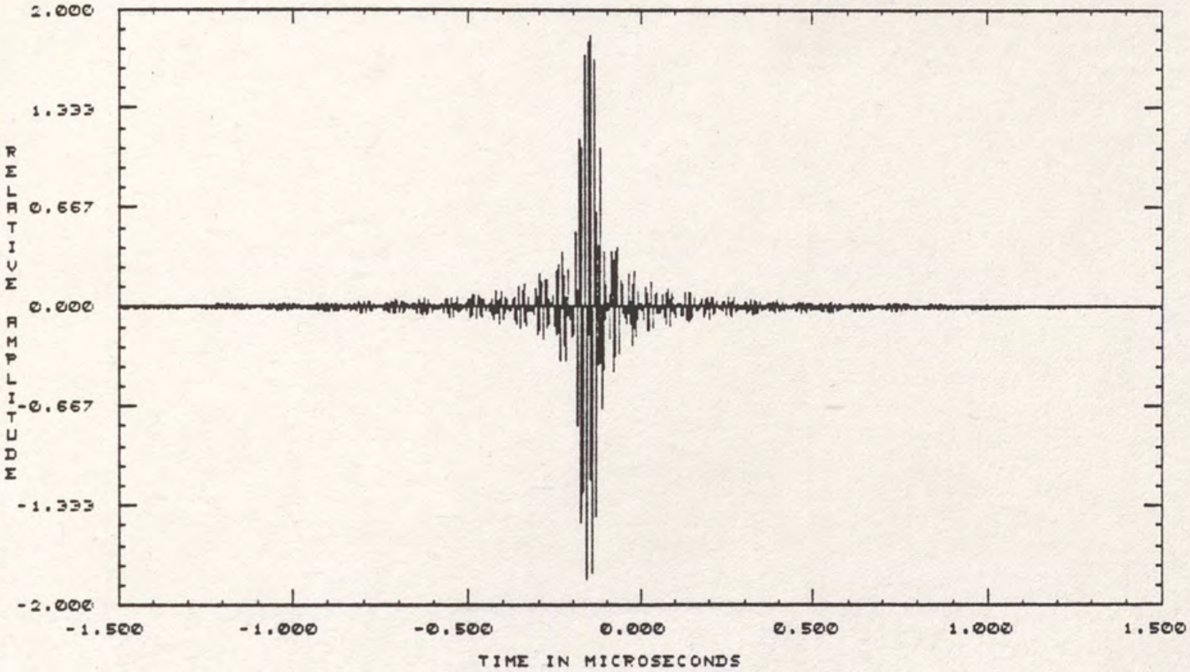

Figure 13. Linear FM matched filter output with Doppler frequency shift of 1 MHz.

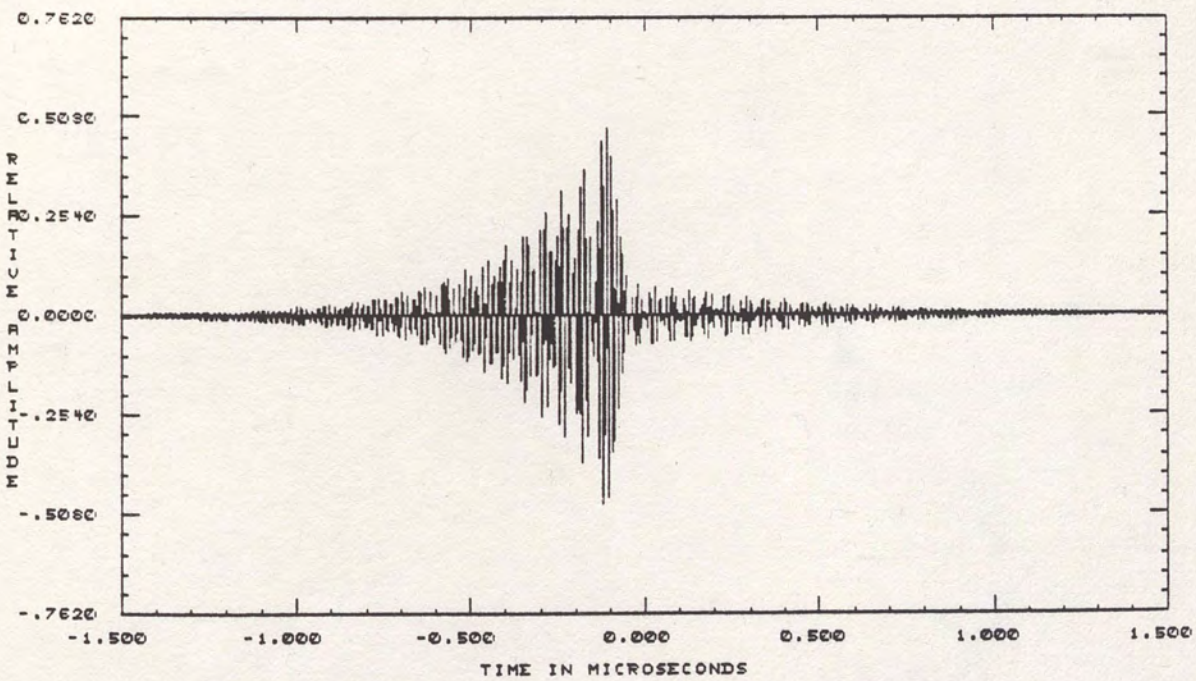

Figure 14. New non-linear FM matched filter output with Doppler frequency shift of 1 MHz.

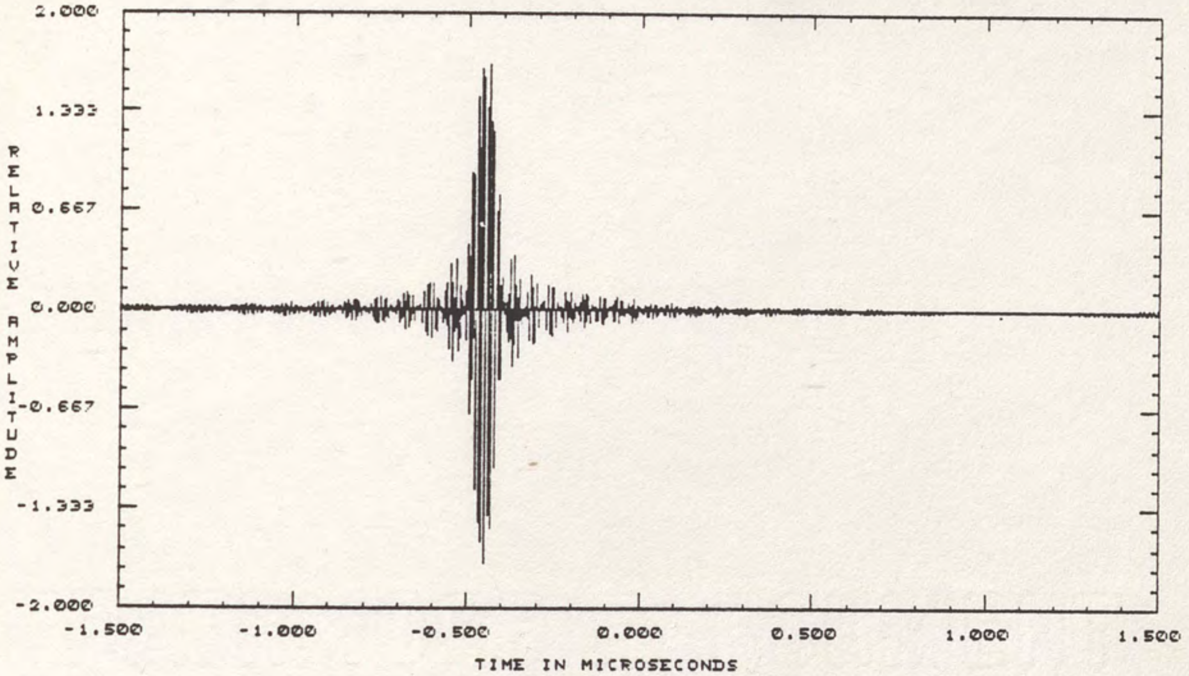

Figure 15. Linear FM matched filter output with Doppler frequency shift of 3 MHz.

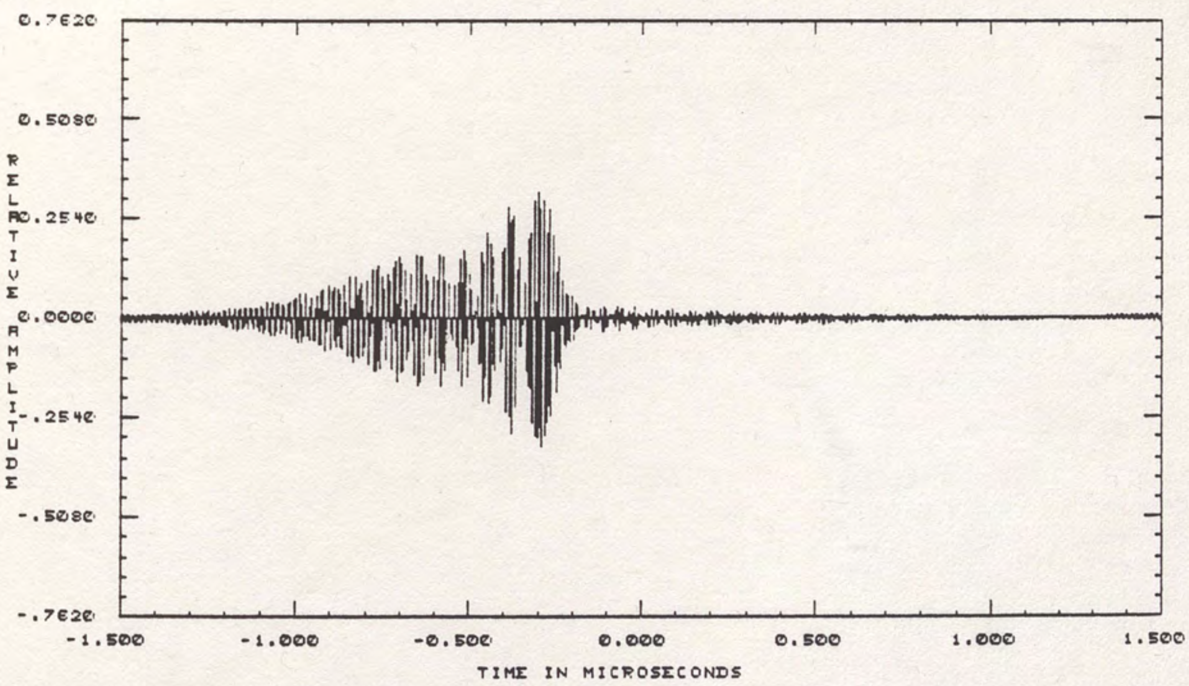

Figure 16. New non-linear FM matched filter output with Doppler frequency shift of 3 MHz.

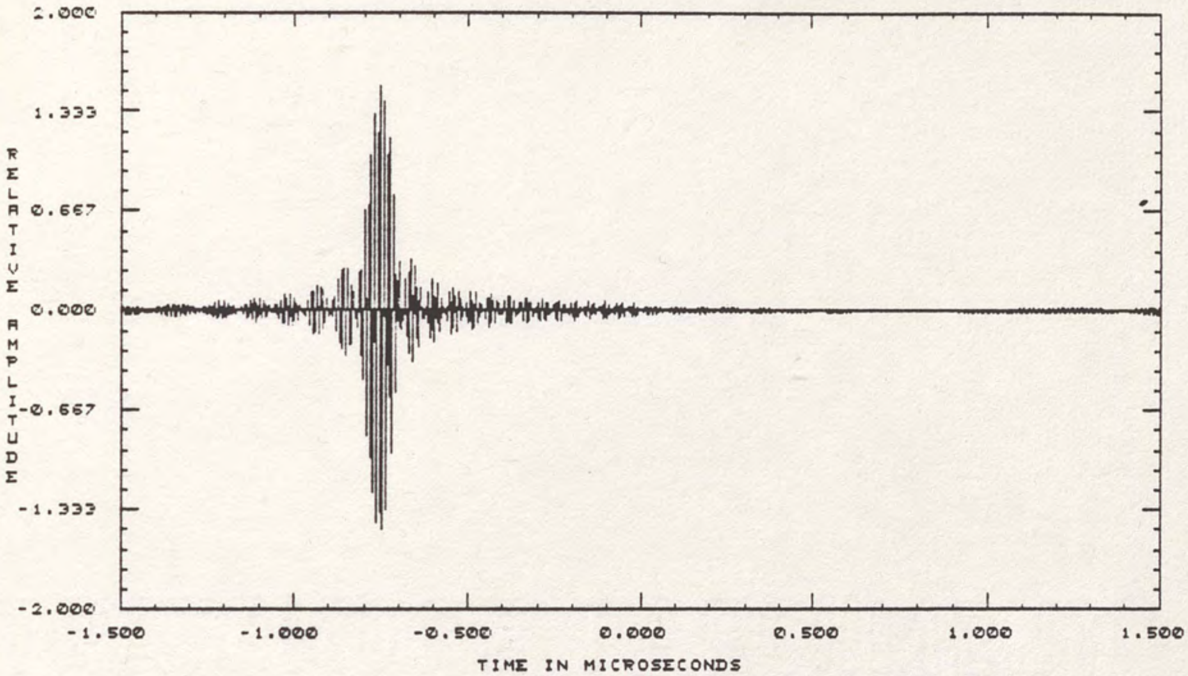

Figure 17. Linear FM matched filter output with Doppler frequency shift of 5 MHz.

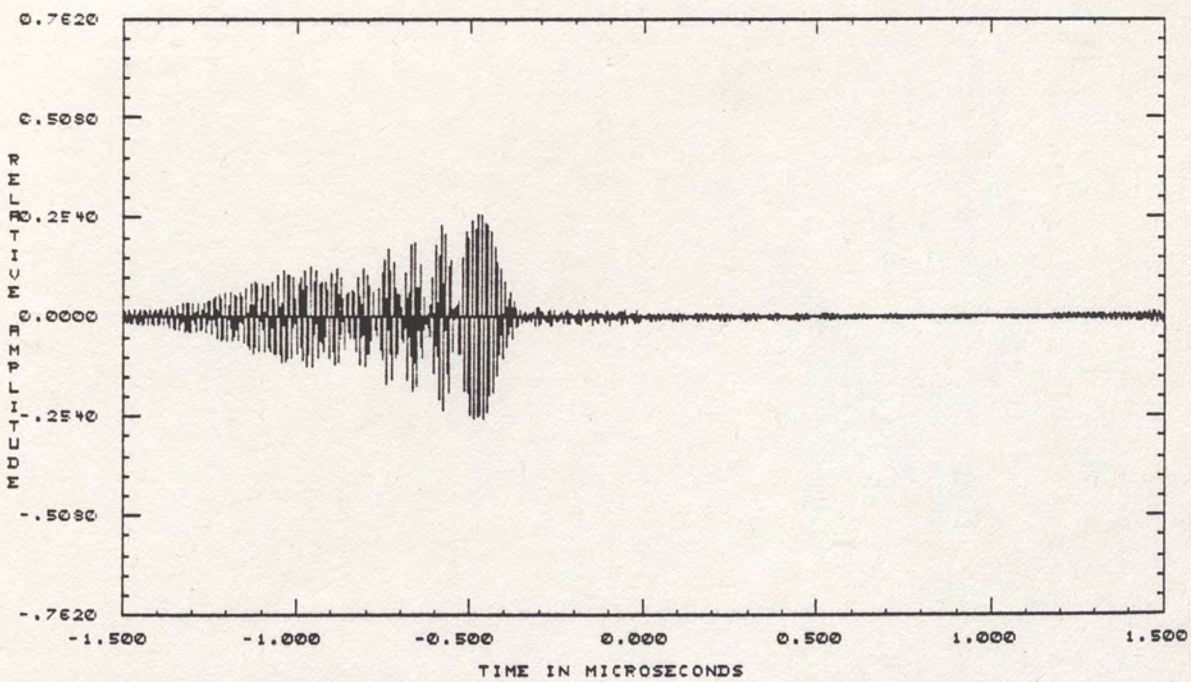

Figure 18. New non-linear FM matched filter output with Doppler frequency shift of 5 MHz.

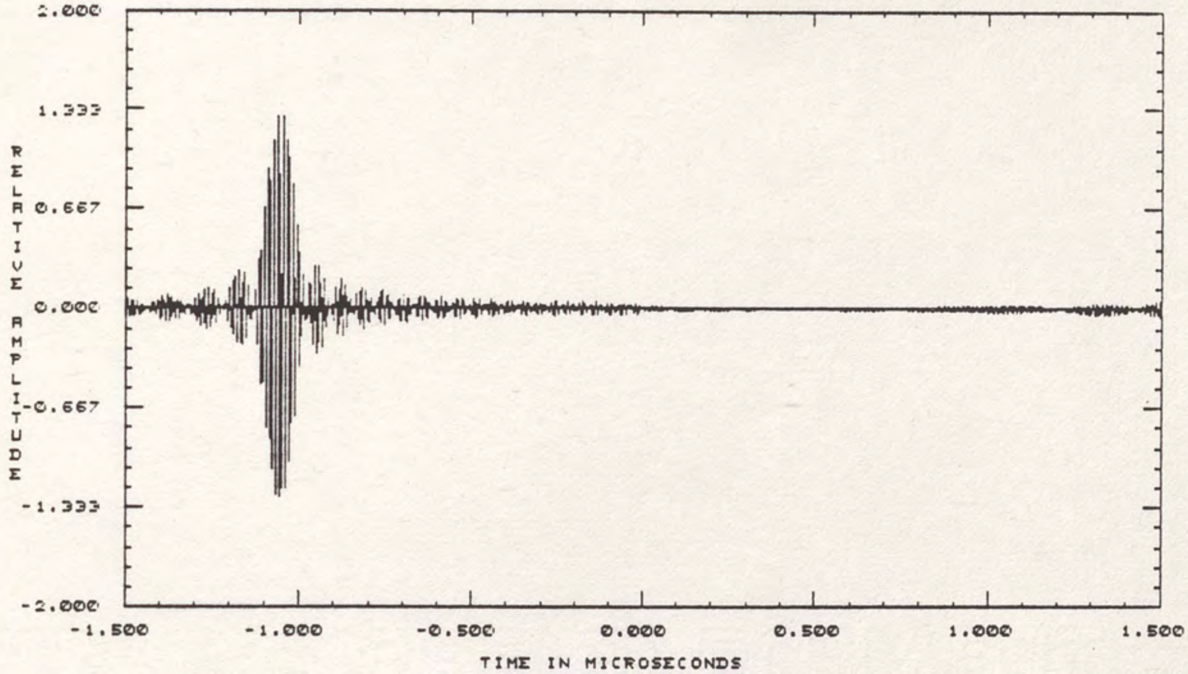

Figure 19. Linear FM matched filter output with Doppler frequency shift of 7 MHz.

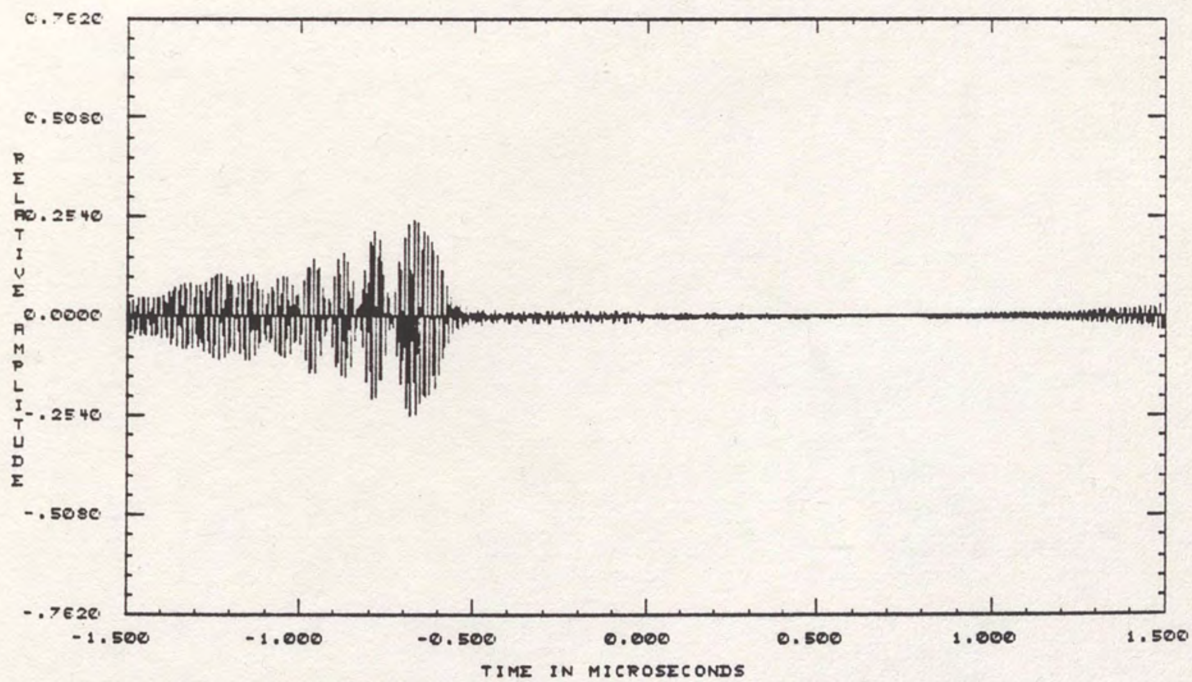

Figure 20. New non-linear FM matched filter output with Doppler frequency shift of 7 MHz.

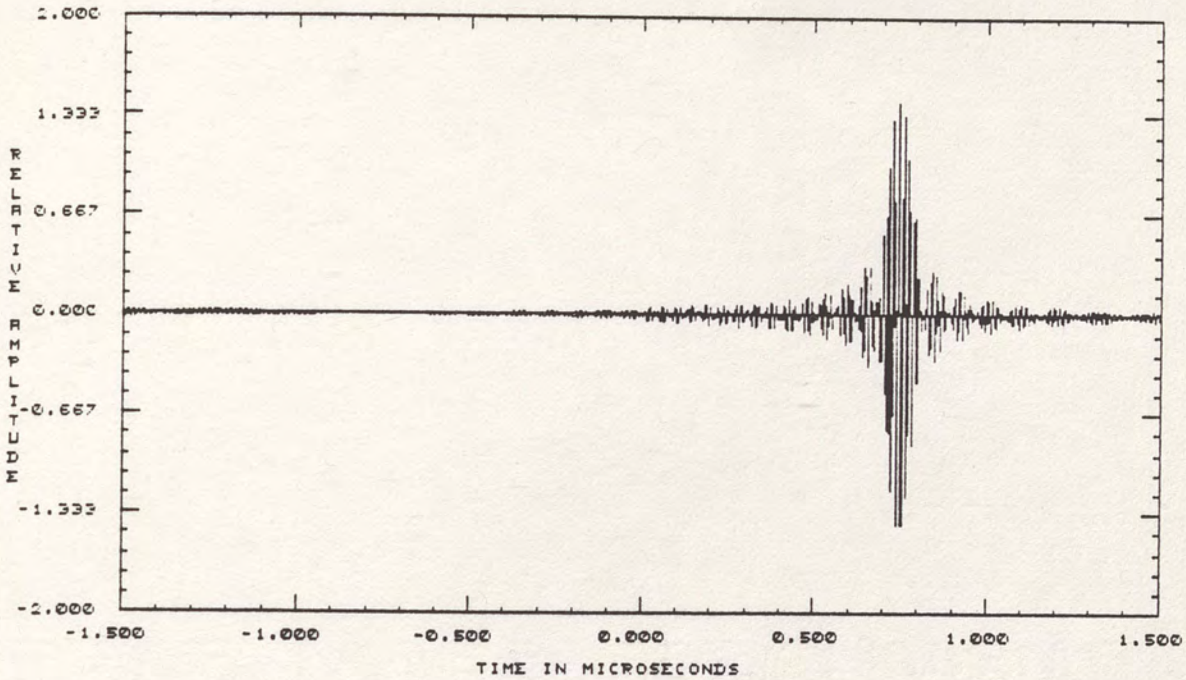

Figure 21. Linear FM matched filter output with Doppler frequency shift of -5 MHz.

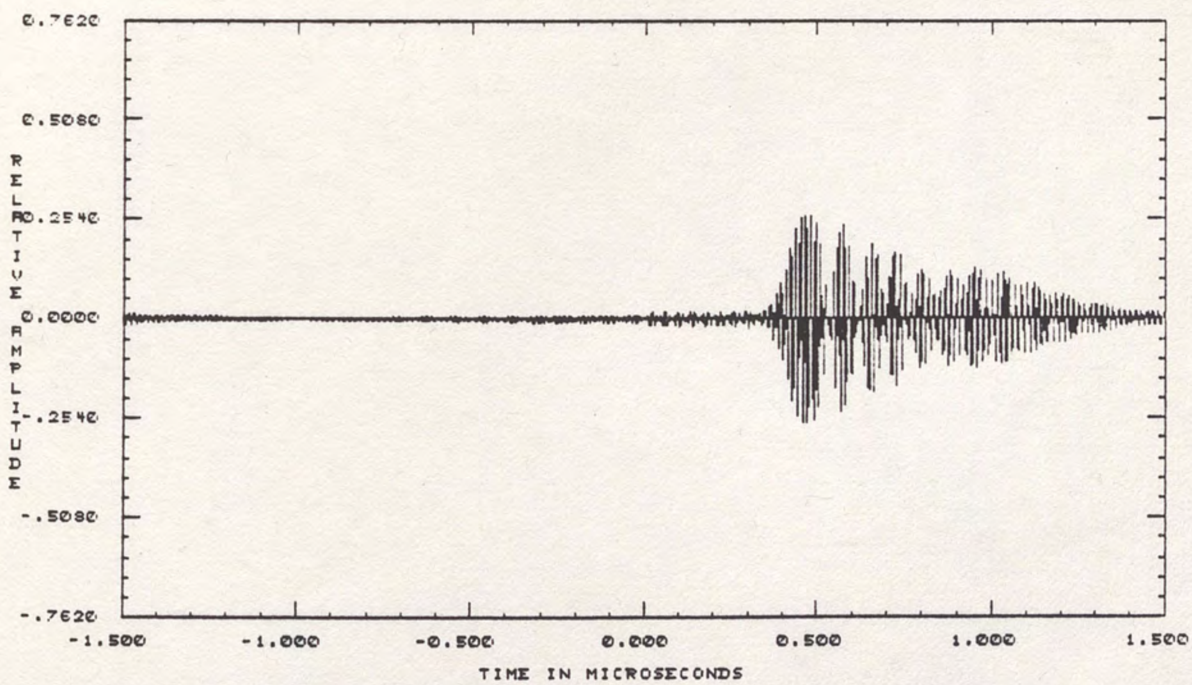

Figure 22. New non-linear FM matched filter output with Doppler frequency shift of -5 MHz.

respective Doppler frequency shifts used in the linear FM simulation. These results are summarized in Table 2.

TABLE 1

DOPPLER DISTORTION EFFECTS ON LINEAR FM

DOPPLER FREQUENCY (MHz)	AMPLITUDE DEGRADATION (dB)	TIME SHIFT (usec)
1	0.45	-0.15
3	1.41	-0.45
5	2.50	-0.75
7	3.74	-1.05
-5	2.50	+0.75

TABLE 2

DOPPLER DISTORTION EFFECTS ON THE NEW NON-LINEAR FM

DOPPLER FREQUENCY (MHz)	AMPLITUDE DEGRADATION (dB)	TIME SHIFT (usec)
1	3.84	-0.12
3	8.12	-0.30
5	9.39	-0.48
7	9.86	-0.68
-5	9.39	+0.48

Though the amplitude degradation due to Doppler shifts is much more severe for the new non-linear FM, a significant improvement is seen in terms of time shift distortion. This characteristic could be exploited to yield a system which is capable of detecting a larger range of velocities at the expense of more power in the transmitted pulse.

Another significant characteristic is that the sidelobes on one side (the side toward the zero velocity point) for the new non-linear FM waveform have virtually disappeared when subjected to Doppler shifts. This provides information not immediately apparent in the linear FM waveform. First of all, it serves as a flag to indicate whether a target is moving or not. Secondly, if the target is moving, it indicates whether it is moving toward or away from the radar.

It should be noted that equation (17) applies to a system in which the transmitted pulse was an up-chirp and the matched filter impulse response was a down-chirp. If the system were reversed such that the transmitted pulse was down-chirped, the time shift would be in the opposite direction. If the transmitted pulse chirped up and down (V-chirp), then there would be a shift in both directions.

The V-chirp offers a means of measuring target velocity without the need for a bank of narrow band filters. When a V-chirped pulse meets a moving target, the resulting matched filter output has two maxima due to the Doppler effect which are symmetric about the zero velocity point. This essentially creates its own reference point from which the time shift can be measured and the velocity computed. Actually, the V-chirp measures speed, not velocity, since there is no way of knowing for a given maximum, whether it was caused by the up- or down-chirped portion of the pulse, thus losing the sense of target direction. Also, due to the presence of two maxima, target resolving capability is limited.

Figures 23 and 24 are for a Doppler frequency shift of 0, for linear FM and the new non-linear FM respectively. Figures 25 and 26 represent a Doppler frequency shift of -5 MHz and are scaled to the peak amplitude of their respective 0 Doppler frequency shift waveforms in order to show the relative amplitude degradation. Tables 3 and 4 summarize the results of the Doppler simulation for the linear V-chirp and the new non-linear V-chirp, respectively.

TABLE 3

DOPPLER DISTORTION EFFECTS ON V-CHIRPED LINEAR FM

DOPPLER FREQUENCY (MHz)	AMPLITUDE DEGRADATION (dB)	TIME SHIFT (usec)
0	0.00	0.00
-5	8.92	0.40

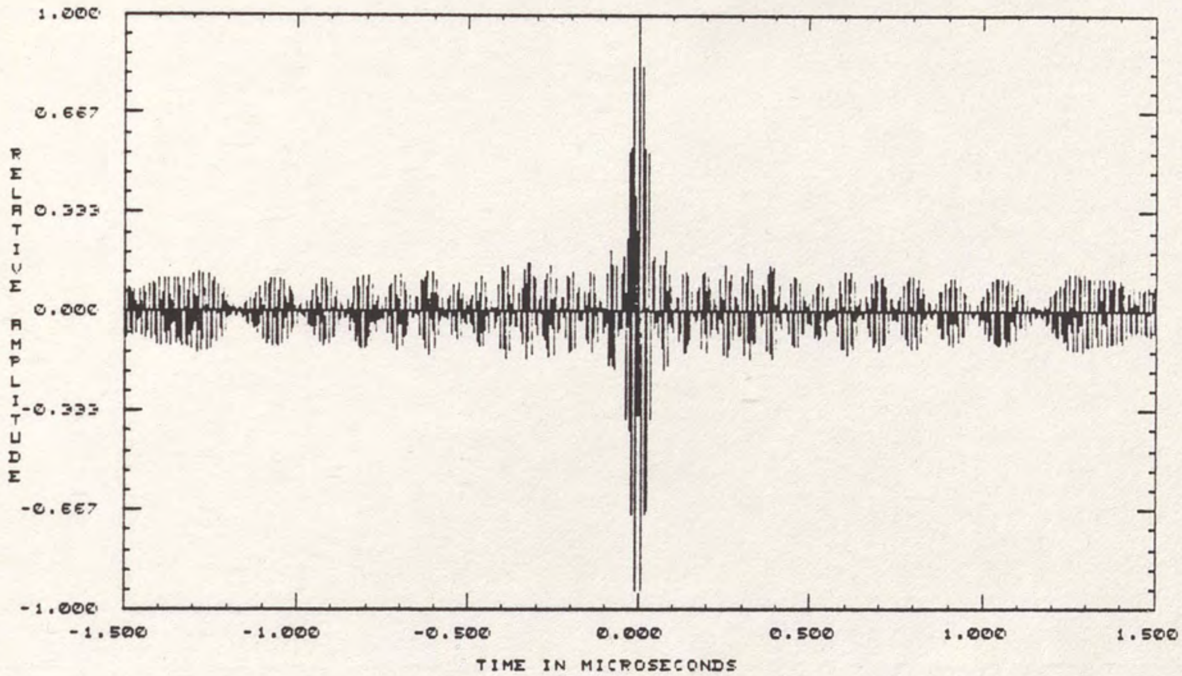

Figure 23. V-chirped linear FM matched filter output with no Doppler frequency shift.

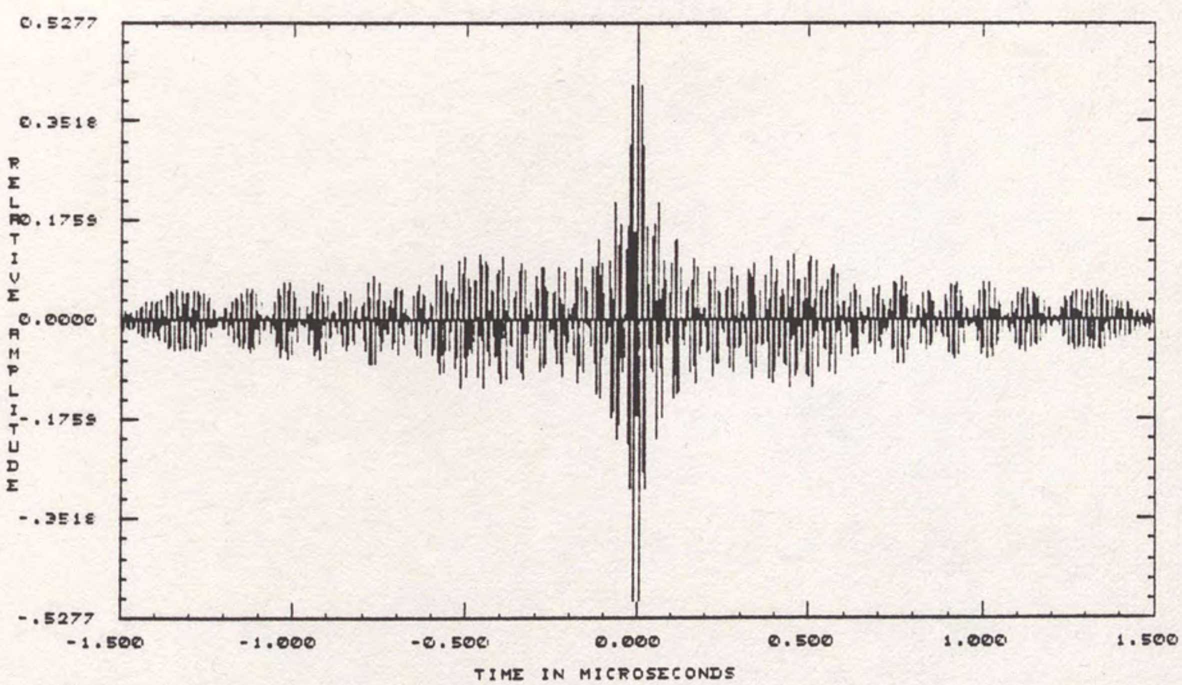

Figure 24. V-chirped non-linear FM matched filter output with no Doppler frequency shift.

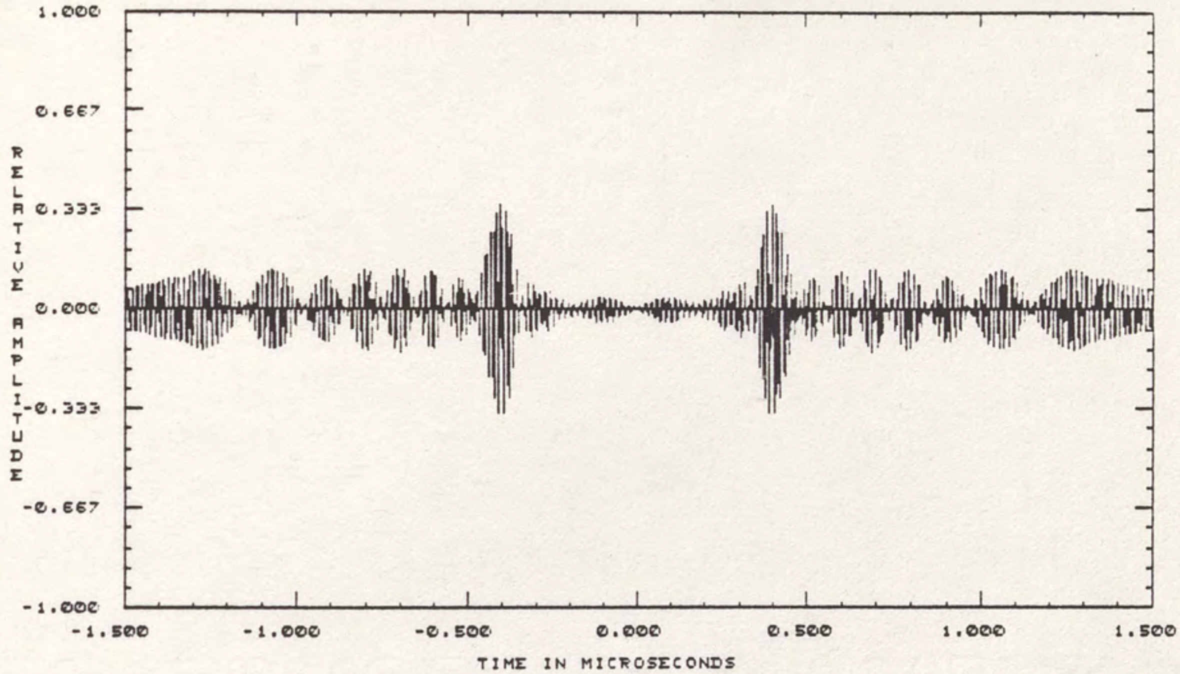

Figure 25. V-chirped linear FM matched filter output with Doppler frequency shift of -5 MHz.

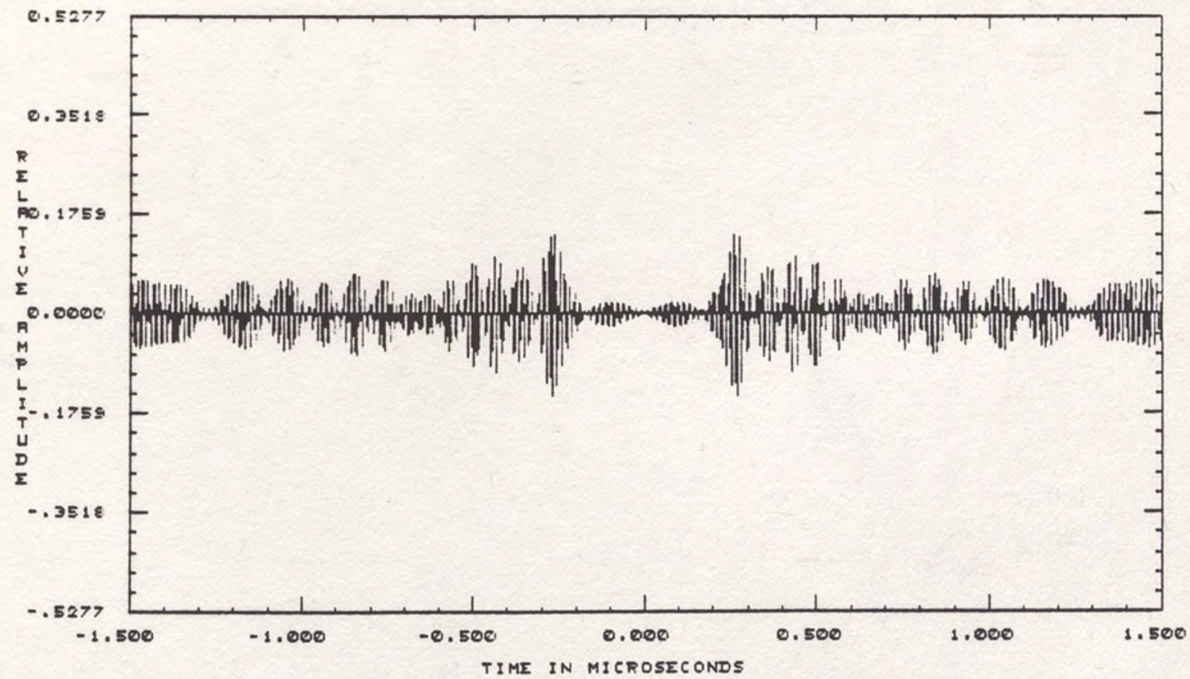

Figure 26. V-chirped non-linear FM matched filter output with Doppler frequency shift of -5 MHz.

TABLE 4

DOPPLER DISTORTION EFFECTS ON V-CHIRPED NON-LINEAR FM

DOPPLER FREQUENCY (MHz)	AMPLITUDE DEGRADATION (dB)	TIME SHIFT (usec)
0	0.00	0.00
-5	11.66	0.26

As before, linear FM offers less amplitude degradation, while the new non-linear FM provides less time shift distortion. For V-chirp, this time shift distortion is actually desired, since it is used to compute velocity. Also, no range error is introduced when the midpoint between the two maxima is taken as the reference. Again, if the need for a more dynamic range of velocities exists, perhaps the new non-linear FM should be considered.

CHAPTER V

SAW DEVICE DESIGN CONSIDERATIONS

The linear FM and the new non-linear FM time waveforms can be completely described mathematically in a relatively simple form. In contrast, the mathematical representation of the frequency spectrum of these signals is very complex and approximations must be used in the derivation. Impulse response model design of SAW devices lends itself very well to this situation because of the correspondence between the location of the electrodes on the transducer and the signal generated by an impulse of acoustic energy traveling under the electrodes (Hartmann, Bell, and Rosenfeld 1973).

Critical Points of the Chirped Waveform

The proper positioning of the transducer electrodes is dependent on the location of the positive peaks (peaks), negative peaks (valleys), and nulls of the waveform. The distance between electrodes will vary proportional to the changing frequency.

Closed Form Solution

For the linear FM waveform, the critical points can be found in closed form. By normalizing equation (11) and allowing the initial phase, ϕ_o, to be $-\pi/2$, the following form of the equation is realized

$$X_{LFM}(t) = \sin\left(\omega_c t + \frac{\mu}{2} t^2\right) \tag{18}$$

where $\mu = aK_f/T$ is the chirp slope. The peaks of the waveform are found when $X_{LFM}(t)$ is equal to 1. Allowing t_p to represent the time at which the peaks occur the equation becomes

$$\sin\left(\omega_c t_p + \frac{\mu}{2} t_p^2\right) = 1 \tag{19}$$

which can be written as

$$\frac{\mu}{2} t_p^2 + \omega_c t_p - \sin^{-1}(1) = 0 \tag{20}$$

or simply

$$\frac{\mu}{2} t_p^2 + \omega_c t_p - \left(\frac{\pi}{2} + 2n\pi\right) = 0 \tag{21}$$

where n is an integer and accounts for multiple peaks. Using the quadratic formula to solve for t_p and ignoring the negative root yields

$$t_p = \frac{-\omega_c + \sqrt{\omega_c^2 + \mu\pi(4n + 1)}}{\mu} \tag{22}$$

The valleys and nulls of the waveform can be found in a similar matter by setting $X_{LFM}(t)$ in equation (18) equal to -1 and 0, respectively. The results of doing so gives

$$t_v = \frac{-\omega_c + \sqrt{\omega_c^2 + \mu\pi(4n - 1)}}{\mu} \tag{23}$$

and

$$t_n = \frac{-\omega_c + \sqrt{\omega_c^2 + 2\mu n\pi}}{\mu} \tag{24}$$

where t_v and t_n represent when the valleys and nulls of the waveform occur.

A closed form solution for the critical points of the new non-linear FM waveform is not as easy. From equation (16), it can be seen that finding the critical points is no longer a matter of solving a simple quadratic. It was suggested by Booher (1985) that this task could be accomplished using an iterative approach of oversampling. Although this approach is valid, a more efficient way of doing this is possible.

The Walkerdid Algorithm

Although this algorithm was developed for finding the critical points of the new non-linear waveform discussed in this thesis, it can also be applied to linear FM or other non-linear FM waveforms. The algorithm is applied to the up-chirped (increasing frequency) form of the FM equation. Symmetry is used if the desired critical points were for the down-chirped or V-chirped form of the equation.

First, define Td as the desired amount of dispersion time in the waveform and Ta as the actual amount of dispersion time it takes to go from the minimum to maximum frequency and such that the final waveform starts on a null and has positive slope and ends on a null and has negative slope as shown in Figure 27. For up- or down-chirped waveforms, Ta is approximately equal to Td and for V-chirped waveforms, Ta is approximately equal to Td/2. Letting $X_u(t)$ represent the up-chirped waveform, $X_d(t)$ represent the down-chirped waveform, and $X_v(t)$ represent the V-chirped waveform, symmetry suggests

$$X_d(t) = X_u(Ta - t) \tag{25}$$

and

$$X_v(t) = \begin{cases} X_u(t) & 0 \leq t \leq Ta \\ X_u(2Ta-t) & Ta \leq t \leq 2Ta \end{cases} \tag{26}$$

For up or down-chirped linear FM, the value of Ta, (Ta_{LFM}), is found by evaluating equation (18) for $t = Ta_{LFM}$ and setting it equal to zero (want to end on a null) such that

$$\sin\,[(\omega_c + \frac{\mu}{2}\,Ta_{LFM})Ta_{LFM}] = 0 \tag{27}$$

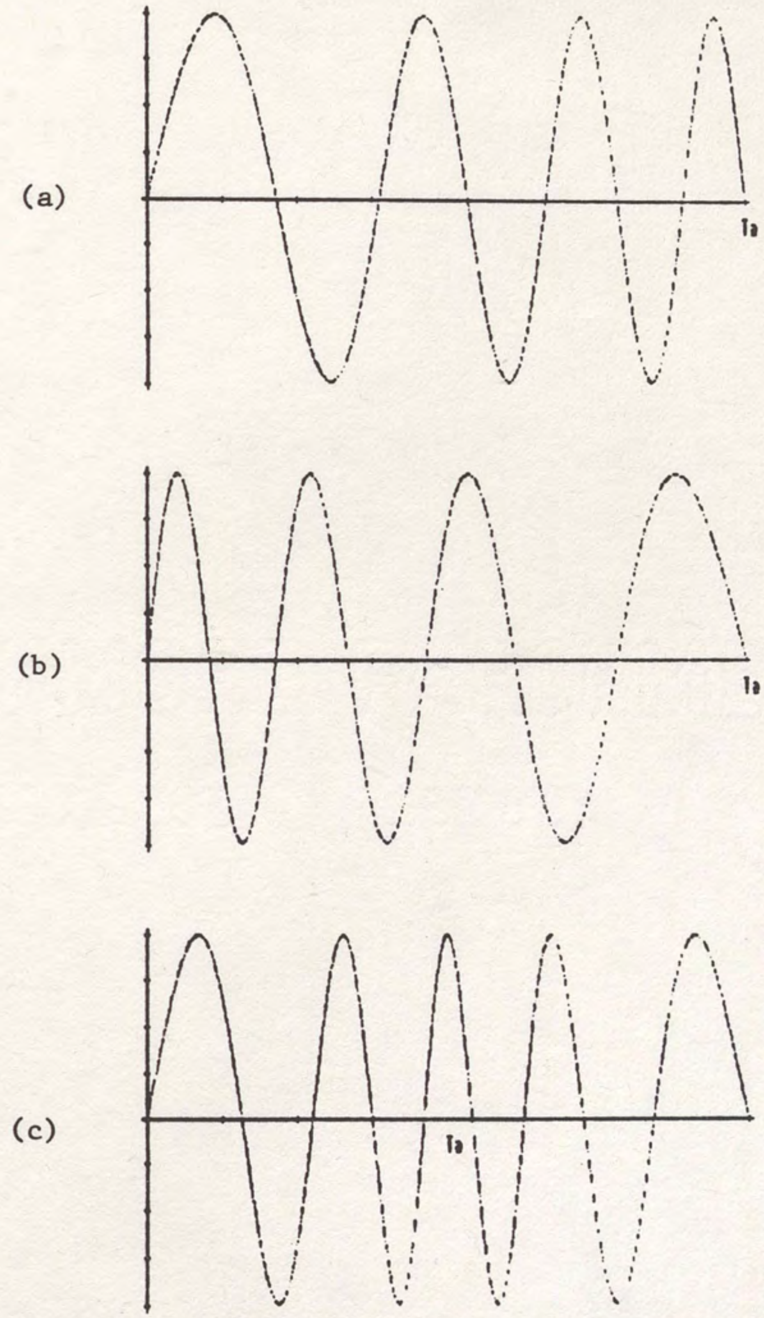

Figure 27. Geometry of chirped waveforms
(a) up-chirped waveform
(b) down-chirped waveform
(c) V-chirped waveform

Now, noting that $\mu = aK_f/T = aK_f/Ta_{LFM}$, equation (27) is written as

$$\sin \left[(\omega_c + \frac{aK_f}{2})\, Ta_{LFM}\right] = 0 \tag{28}$$

or

$$Ta_{LFM}(\omega_c + \frac{aK_f}{2}) = n\pi \tag{29}$$

The requirement that the waveform ends with negative slope is met by restricting n to only odd values and solving equation (29) for Ta_{LFM} such that

$$Ta_{LFM} = \frac{(2n + 1)\pi}{\omega_c + (\frac{aK_f}{2})} \tag{30}$$

Next, solve equation (30) for n

$$n = \frac{(\omega_c + \frac{aK_f}{2})\, Ta_{LFM}}{2\pi} - \frac{1}{2} \tag{31}$$

For Ta_{LFM} approximately equal to Td, Td is substituted into equation (31) in place of Ta_{LFM} and n_{LFM} is chosen as the closest integer (CINT) to n resulting in

$$n_{LFM} = \text{CINT}\left[\frac{(\omega_c + \frac{aK_f}{2})\, Td}{2\pi} - \frac{1}{2}\right] \tag{32}$$

Now, using n_{LFM} in place of n in equation (30) yields

$$Ta_{LFM} = \frac{(2n_{LFM} + 1)\pi}{\omega_c + (\frac{aK_f}{2})} \tag{33}$$

For V-chirped linear FM, Ta (Ta_{VLFM}) is found by evaluating equation (18) at t = Ta_{VLFM} and setting it equal to 1 so that the waveform is symmetric about Ta_{VLFM} and there is no sudden phase reversal. No restriction is placed on n in this case because Ta_{VLFM} can occur at any peak. In a similar manner as above, except now Td/2 replaces Ta when solving for n, n_{VLFM} can be shown to be

$$n_{VLFM} = \text{CINT}\left[\frac{(\omega_c + \frac{aK_f}{2})\ Td}{4\pi} - \frac{1}{8}\right] \tag{34}$$

and Ta_{VLFM} is found as

$$Ta_{VLFM} = \frac{(2n_{VLFM} + \frac{1}{2})\pi}{(\omega_c + \frac{aK_f}{2})} \tag{35}$$

The value of Ta for the new non-linear FM waveform is found when equation (16) is normalized, the initial phase set to $-\pi/2$, and Ta substituted for t, yielding

$$X_{NLFM}(Ta) = \sin\left[(\omega_c + 0.43\ aK_f)\ Ta\right] \tag{36}$$

Following the same procedure used for linear FM, the values of Ta and n for up or down-chirped non-linear FM can be shown to be

$$n_{NLFM} = \text{CINT}\left[\frac{(\omega_c + 0.43\ aK_f)\ Td}{2\pi} - \frac{1}{2}\right] \tag{37}$$

$$Ta_{NLFM} = \frac{(2n_{NLFM} + 1)\pi}{\omega_c + 0.43\ aK_f} \tag{38}$$

and for V-chirped non-linear FM

$$n_{VNLFM} = \text{CINT}\left[\frac{(\omega_c + 0.43\ aK_f)\ Td}{4\pi} - \frac{1}{8}\right] \tag{39}$$

$$Ta_{VNLFM} = \frac{(2n_{VNLFM} + \frac{1}{2})}{(\omega_c + 0.43\ aK_f)} \tag{40}$$

In the up or down-chirped waveforms, the value n+1 represents the total number of nulls, (n+1)/2 is the total number of peaks, and (n-1)/2 is the total number of valleys. For the V-chirped waveforms, 4n+2 represents the total number of nulls, 2n+1 is the total number of peaks, and 2n is the total number of valleys.

Now, knowing that the first critical point (a null) is at t=0, the function is increasing in frequency and the next critical point is a peak, the Walkerdid algorithm can be implemented.

Step 1: Call the location of the currently known critical point t_1.

Step 2: Find the instantaneous frequency at t_1 and call it f_1.

Step 3: Calculate t_2 using $t_2 = (t_1 + 1/4f_1)$. Since the frequency is increasing, t_2 will be slightly past the next critical point.

Step 4: Find the instantaneous frequency at t_2 and call it f_2.

Step 5: Calculate t_3 using $t_3 = (t_1 + 1/4f_2)$. Since f_2 is higher than f_1, t_3 will occur slightly before the next critical point.

Step 6: Calculate the argument of the chirped waveform at t_2 and at t_3 and call them arg_2 and arg_3, respectively.

Step 7: Calculate y_2 and y_3. If the currently unknown critical point is a peak or valley use $y_2 = \cos(arg_2)$, $y_3 = \cos(arg_3)$ otherwise use $y_2 = \sin(arg_2)$, $y_3 = \sin(arg_3)$. The nulls of the waveform occur when the function goes to zero, the peaks and valleys occur when the derivative of the function goes to zero.

Step 8: Using linear interpolation calculate the value t from the equation $t = t_2 - y_2(t_3-t_2)/(y_3-y_2)$. This step essentialy "homes" in on the currently unknown critical point.

Step 9: Calculate the argument of the chirped waveform at t and call it arg.

Step 10: Calculate y. If the currently unknown critical point is a peak or valley use $y = \cos(arg)$, otherwise use $y = \sin(arg)$.

Step 11: If $y=0$ go to step 15.

Step 12: If the sign of y equals the sign of y_2, let $y_2=y$ and $t_2=t$.

Step 13: If the sign of y equals the sign of y_3, let $y_3=y$ and $t_3=t$.

Step 14: Go to step 8.

Step 15: Store t as the current critical point. If t is less than Ta then goto step 1.

Step 16: If the desired critical points were for down-chirped or V-chirped waveforms, use the symmetry equations (25) and (26) to adjust them. End.

Structural Layout of Transducers

The transducer's geometry and the waveform are related by the acoustic velocity, ν_a, of the substrate. The wavelength, λ, can be calculated from

$$\lambda = \frac{\nu_a}{f} \tag{41}$$

The correspondence between the waveform and structural layout of the dispersive transducer used in the design is shown in Figure 28. Double electrodes were used to help reduce reflections. The x-position of each electrode corresponds to the midpoint between critical points. In order to maintain a 50% duty cycle (approximately equal spacing and electrode widths) the instantaneous frequency, f_i, corresponding to the electrode's x-position must be calculated. The widths of the electrodes are $\lambda_i/8$, where λ_i represents the instantaneous wavelength.

In order to acccount for the frequency dependence on the output of the SAW device, the transducer must be properly apodized. For an even crested waveform, this apodization is dependent on $f_i(t)^{-3/2}$, that is the amplitude of output is inversely proportional to the instantaneous frequency (Hartman, Bell and Rosenfeld 1973). The largest amount of overlap, the acoustic beamwidth W_a, occurs at the lowest frequency and was chosen as 100 λ_o (the wavelength at the center frequency). The amount of overlap corresponding to each peak and valley was found using $f_i^{-3/2}$, then normalizing to W_a. These overlaps were then used to determine the y-position and height of each electrode. Figure 29 shows the effect of the apodization.

In order to maintain 50% metallization, dummy electrodes are used. These dummy electrodes were placed such that the electrode gaps were $\lambda_i/8$. Bus bar heights were chosen large enough (10 mils) to allow for bonding without having to add additional bonding pads. Figure 30 shows the transducer with dummy electrodes.

The width of the non-dispersive transducer was chosen to be 1.5 λ_o. The distance between transducers was 50 mils and a 10 mil wide bar was placed midway between transducers to help reduce RF feedthrough. Figure 31 shows the complete SAW device.

The FORTRAN source code used in the design of the devices is provided in the Appendix. The files created are in SAWCAD STRUCTURE format (Malocha and Richie 1984). The variable ISNUM is the dimension of the X, Y, W, A, IREP, XD, and YD arrays where

X = horizontal position of the lower left-hand corner of the rectangle

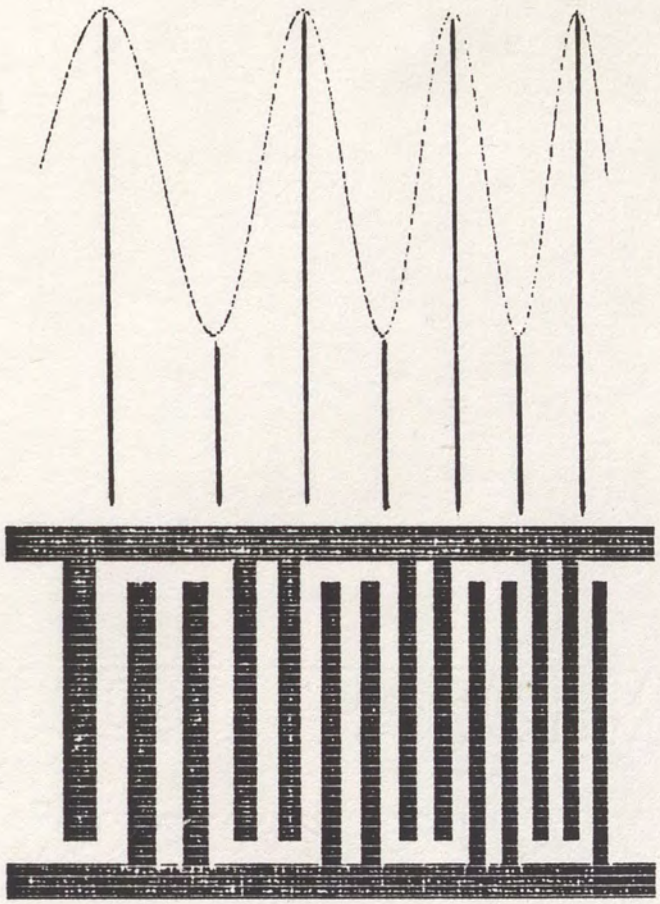

Figure 28. Waveform to transducer correspondence.

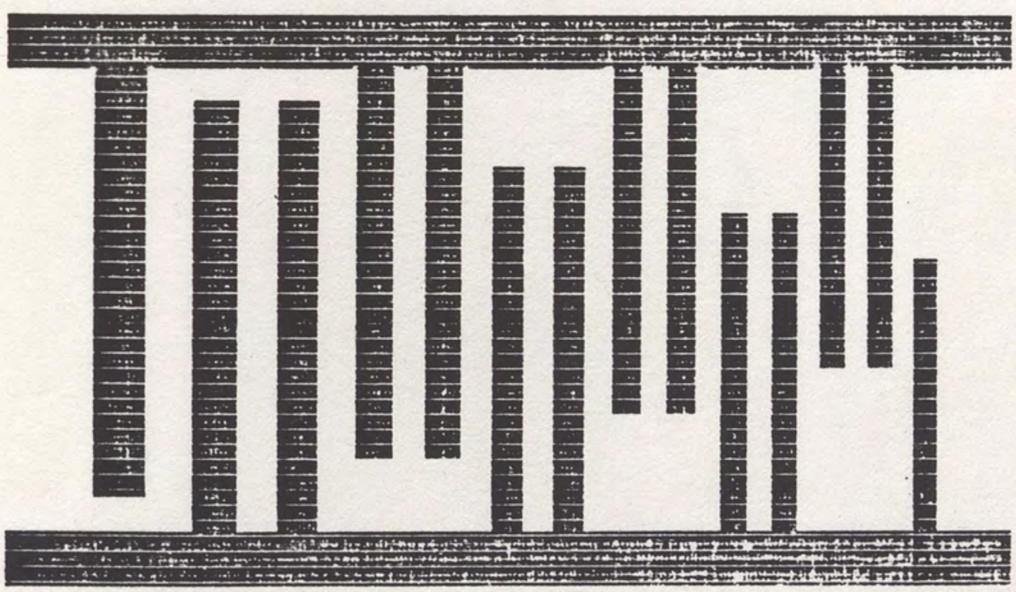

Figure 29. The effect of apodization.

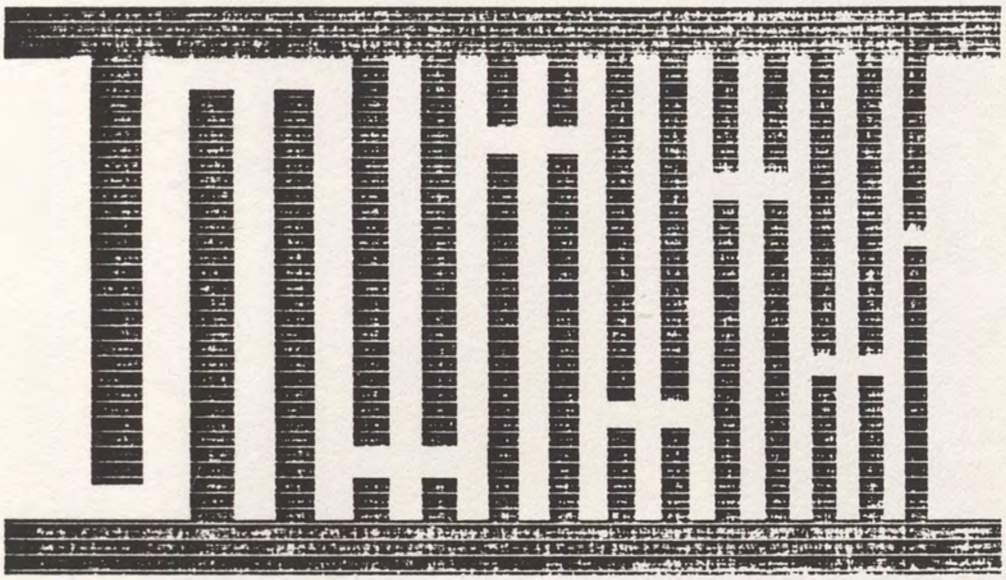

Figure 30. The placement of dummy electrodes.

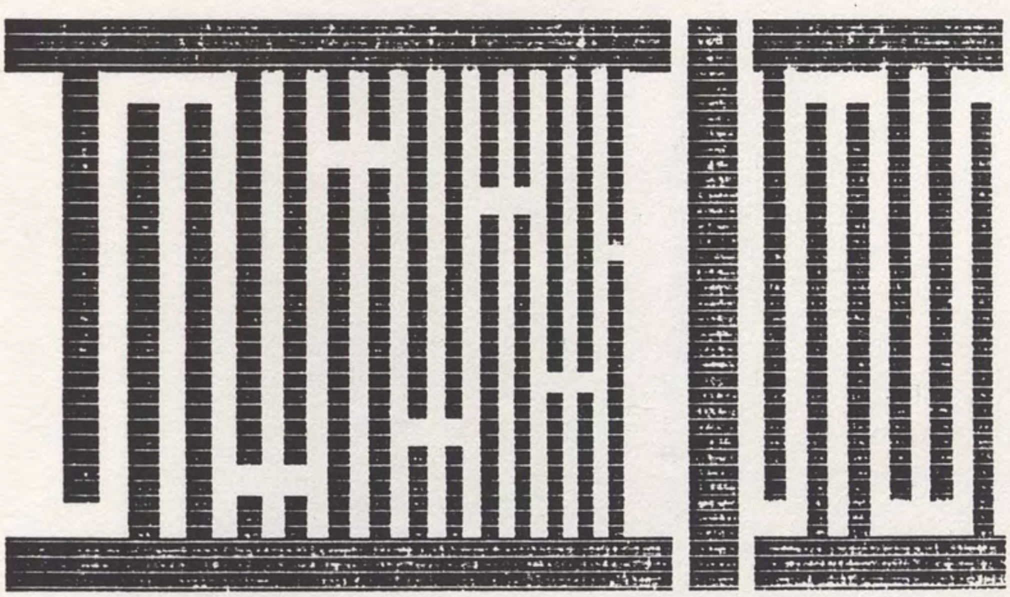

Figure 31. The SAW device with both transducers.

Y = vertical position of the lower left-hand corner of the rectangle

W = width of the rectangle

H = height of the rectangle

A = angle orientation (always 0)

IREP = number of incremental repetitions (always 1)

XD = incremental step in the X direction (always 0)

YD = incremental step in the Y direction (always 0)

The devices were designed for a center frequency of 70 MHz, a 20 MHz bandwidth, a 3 usec dispersion time and for fabrication on Y-cut Z-propogating lithium niobate (50% metallization acoustic velocity = 3448 m/s). The devices were approximately 12.003 mm in width and 5.483 mm in height.

CHAPTER VI

DESIGN IMPLEMENTATION

Once the STRUCTURE file has been created, a photomask must be produced so that the devices can be fabricated. Though the facility for producing photomasks is not available at the University of Central Florida, local industry (in this case, SAWTEK of Orlando) often provides the necessary support.

Photomask Generation

To make full use of the photomask, the STRUCTURE files were stepped and repeated until all of the available space was utilized. The final product had four copies of each of the four designs (linear FM, non-linear FM, and the V-chirped versions of each) plus some additional designs by other students. The format of the STRUCTURE data had to be converted to a format known as ELECTROMASK in order to be understood by the pattern generator.

ELECTROMASK data uses a body centered coordinate system, assumes dimensional data is in units of tenths of microns and angle data is in tenths of degrees, and has maximum aperture limitations. The subprogram, SBREAK2, was written to convert general STRUCTURE data (to include any arbitrary angle and any number of repeats) to ELECTROMASK data and is included in the Appendix.

Fabrication

The fabrication of devices was accomplished using the new clean room facilities at the University of Central Florida. For a detailed explanation of the fabrication process see Vigil and Yapp (1984). Briefly, the process consists of the following steps:

1) Clean the lithium niobate wafer.
2) Deposit aluminum using flash evaporation or sputtering techniques.
3) Apply photoresist to the metallized surface and soft bake.
4) Use the mask aligner to make contact between the wafer and mask, then expose to the ultraviolet source.
5) Develop the wafer to remove the exposed photoresist.
6) Place the wafer in an aluminum etch to remove the desired metal.
7) Dice the wafer to separate the individual devices.
8) Mount the devices to a header.
9) Bond the devices to the header.
10) Apply absorbing material to the ends of the devices to reduce edge reflections.

A few problems were encountered during fabrication which would affect the performance of the devices. One such problem was the inability to achieve a smooth layer of photoresist in step 3. This resulted in a number of fingers on the devices being open or shorted. Another problem was created back when the STRUCTURE data was stepped and repeated. There should have been enough distance left between the devices to allow for a sufficient amount of absorbing material to be placed at the ends of the device.

CHAPTER VII

CONCLUSIONS

Fundamental radar concepts have been reviewed, emphasizing the need for matched filtering and the desirable characteristics of the transmitted pulse and its autocorrelation function. A new non-linear FM pulse compression technique has been reviewed using linear FM as the basis of comparison. A simulation of the Doppler effect on the matched filter output has been accomplished, including the effects on V-chirped signals. SAW device design considerations have been presented and an efficient algorithm for finding the critical points of a chirped waveform was introduced. Computer software was written and is presented in the Appendix to implement the SAW device design and the subsequent conversion of data for photomask generation. SAW device fabrication was briefly reviewed.

The Doppler simulation showed that the new non-linear FM, unlike linear FM, could be used as a moving target indicator by examining only the matched filter output in the time domain. Furthermore, the new non-linear FM offers more dynamic range than the linear FM because the time shift is less than that of linear FM.

In conclusion, it is recommended a new photomask be produced which provides a layout allowing for a sufficient amount of absorbing material to be placed at the ends of the devices. Also, a simulation which includes the combination of the new non-linear FM with windowing and the Doppler effect may add to the resolution capabilities of the pulse. Further research could include the new non-linear FM's potential in a multiple target environment.

APPENDIX

COMPUTER PROGRAMS

```
c       program chirpy
c       -----------------------------------------------------------

c       date of last revision:                  09-16-86

c       for more information contact:           D.C.  Malocha
c                                               J.C.  Walker

c       -----------------------------------------------------------

        program chirpy

        character*1 ask
        common/pat1/ x(6000),y(6000),w(6000),h(6000),a(6000),
     2  irep(6000),xd(6000),yd(6000),isnum,iref

c       ** MENU **

500     call cls
        write(*,*) '                        Dispersive SAW Device Design'
        write(*,*) ' '
        write(*,*) ' '
        write(*,*) ' '
        write(*,*) '                                [C]reate File'
        write(*,*) ' '
        write(*,*) '                                [P]rint File'
        write(*,*) ' '
        write(*,*) '                                [S]ave File'
        write(*,*) ' '
        write(*,*) '                                [Q]uit'
        write(*,*) ' '
        write(*,*) ' '
        write(*,*) ' '
        write(*,*) ' '
        write(*,*) ' '
        write(*,*) ' Enter Choice'
        write(*,*) ' '
        read(*,600) ask
600         format(a1)
        if (ask.eq.'c'.or.ask.eq.'C') call create
        if (ask.eq.'p'.or.ask.eq.'P') call prnt
        if (ask.eq.'s'.or.ask.eq.'S') call savit
        if (ask.eq.'q'.or.ask.eq.'Q') goto 700
        goto 500

700     stop
        end

c       -----------------------------------------------------------

        subroutine cls

        do 800 i=1,22
            write(*,*)' '
```

```
800     continue

        return
        end

c       ---------------------------------------------------------------

        subroutine create

        character*1 ask
        common/pat1/ x(6000),y(6000),w(6000),h(6000),a(6000),
     2  irep(6000),xd(6000),yd(6000),isnum,iref
        common/cheq/  akf,arg,fi,ifm,pi,t,ta,wi,wl
        real time(3000),ov(3000),s(6000)

c       ** INPUT & VERIFICATION **

1000    call cls
1100    write(*,*) ' '
        write(*,*) ' 1) Linear FM'
        write(*,*) ' 2) Non-Linear FM'
        write(*,*) ' '
        write(*,*) ' Enter Choice (1,2)'
        write(*,*) ' '
        read(*,*) ifm
        if(ifm.ne.1.and.ifm.ne.2)then
            write(*,*) ' input error - try again'
            goto 1100
            endif

1200    write(*,*) ' '
        write(*,*) '1) Up or Down Chirp'
        write(*,*) '2) "V" Chirp'
        write(*,*) ' '
        write(*,*) ' Enter Choice (1,2)'
        write(*,*) ' '
        read(*,*) ishape
        write(*,*) ' '
        if(ishape.ne.1.and.ishape.ne.2)then
            write(*,*) ' input error - try again'
            goto 1200
            endif

        write(*,*) ' '
        write(*,*) ' Minimum Frequency ( MHz )?'
        write(*,*) ' '
        read(*,*) flt

        write(*,*) ' '
        write(*,*) ' Maximum Frequency ( MHz )?'
        write(*,*) ' '
        read(*,*) fht

        write(*,*) ' '
```

```
       write(*,*) ' Dispersion Time ( usec )?'
       write(*,*) ' '
       read(*,*) tdt

1300   write(*,*) ' '
       write(*,*) ' Data in units of :  1) wavelengths'
       write(*,*) '                     2) millimeters'
       write(*,*) '                     3) micrometers'
       write(*,*) '                     4) mils'
       write(*,*) ' '
       write(*,*) ' Enter Choice (1,2,3,4)'
       write(*,*) ' '
       read(*,*) iscale
       if(iscale.lt.1.or.iscale.gt.4)then
           write(*,*) ' input error - try again'
           goto 1300
           endif

       write(*,*) ' '
       write(*,*) ' SAW Velocity ( meters/sec )?'
       write(*,*) ' '
       read(*,*) va

       call cls
       if(ishape.eq.1)write(*,*) ' Up or Down Chirped'
       if(ishape.eq.2)write(*,*) ' "V" Chirped'
       write(*,*) ' '
       if(ifm.eq.1)write(*,*) ' Linear FM'
       if(ifm.eq.2)write(*,*) ' Non-Linear FM'
       write(*,*) ' '
       write(*,*) ' Minimum Frequency = ',flt,' MHz'
       write(*,*) ' '
       write(*,*) ' Maximum Frequency = ',fht,' MHz'
       write(*,*) ' '
       write(*,*) ' Dispersion Time = ',tdt,' usec'
       write(*,*) ' '
       if (iscale.eq.1) write(*,*) ' Units = wavelengths'
       if (iscale.eq.2) write(*,*) ' Units = millimeters'
       if (iscale.eq.3) write(*,*) ' Units = micrometers'
       if (iscale.eq.4) write(*,*) ' Units = mils'
       write(*,*) ' '
       write(*,*) ' SAW Velocity = ',va,' meters/sec'
       write(*,*) ' '
       write(*,*) ' '
       write(*,*) ' Is this correct (y/n)?'
       read(*,1400) ask
1400       format(a1)
       if (ask.eq.'n'.or.ask.eq.'N') goto 1000
       write(*,*) ' '
       write(*,*) ' '
       write(*,*) ' PROCESSING'

c      ** INITIALIZATION **
```

```
      pi=4.0*atan(1.0)
      fl=flt*1.0e6
      fh=fht*1.0e6
      td=tdt*1.0e-6
      if (ishape.eq.2) td=td/2.0
      akf=2.0*pi*(fh-fl)
      wl=2.0*pi*fl
      wh=2.0*pi*fh

      if (ifm.eq.1.and.ishape.eq.1) then
          n=td*(0.5*fl+0.5*fh)
          ta=(2.0*n+1.0)/(fl+fh)
          endif
      if (ifm.eq.1.and.ishape.eq.2) then
          n=td*(0.5*fl+0.5*fh)+0.25
          ta=(2.0*n+0.5)/(fl+fh)
          endif
      if (ifm.eq.2.and.ishape.eq.1) then
          n=td*(0.57*fl+0.43*fh)
          ta=(2.0*n+1.0)/(1.14*fl+0.86*fh)
          endif
      if (ifm.eq.2.and.ishape.eq.2) then
          n=td*(0.57*fl+0.43*fh)+0.25
          ta=(2.0*n+0.5)/(1.14*fl+0.86*fh)
          endif

      if (ishape.eq.1) then
          ncp=4*n+3
          nf=4*n+2
          endif
      if (ishape.eq.2) then
          ncp=4*n+2
          nf=8*n+2
          endif

      isnum=2*nf+17
      fo=(fl+fh)/2.0
      rlo=va/fo
      height=100.0           ! Acoustic Beamwidth in wavelengths
      wa=height*rlo
      bbh=0.000254/rlo       ! bus bar height is 10 mils
      d=0.00127/rlo          ! 50 mils between transducers
      utw=1.5                ! unweighted transducer is 1.5 wavelengths wide

c     ** CALCULATE NULLS, PEAKS & VALLEYS **

      time(1)=0.0
      t1=0.0
      do 1700 i=2,ncp
          r=i
          t=t1
          call freq
          f1=fi
          t2=t1+17.0/64.0/f1
```

```
          t=t2
          call freq
          f2=fi
          call argument
          if ((r/2.0).gt.(i/2)) ypos=sin(arg)
          if ((r/2.0).eq.(i/2)) ypos=cos(arg)
          ypos2=ypos
          t3=t1+15.0/64.0/f2
          t=t3
          call argument
          if ((r/2.0).gt.(i/2)) ypos=sin(arg)
          if ((r/2.0).eq.(i/2)) ypos=cos(arg)
          ypos3=ypos
          if (sgn(ypos2).eq.sgn(ypos3)) then
              write(*,*) 'ERROR: design specifications cannot be met!'
              write(*,*) '       try either a longer dispersion time'
              write(*,*) '       or smaller bandwidth.'
              write(*,*) ' '
              write(*,*) '  ENTER  < CNTRL > Y  to return to system'
1499          goto 1499
              endif
1500      t=t2-ypos2*(t3-t2)/(ypos3-ypos2)
          if (t.eq.t2.or.t.eq.t3) goto 1550
          call argument
          if ((r/2.0).gt.(i/2)) ypos=sin(arg)
          if ((r/2.0).eq.(i/2)) ypos=cos(arg)
          if (sgn(ypos).eq.0) t1=t
          if (sgn(ypos).eq.sgn(ypos2)) then
              t2=t
              ypos2=ypos
              endif
          if (sgn(ypos).eq.sgn(ypos3)) then
              t3=t
              ypos3=ypos
              endif
          if (ypos) 1500,1600,1500
1550      t1=t2
          if (abs(ypos3).lt.abs(ypos2)) t1=t3
1600      time(i)=t1
1700  continue

      if (ishape.eq.1) goto 1900
      do 1800 i=1,(ncp-1)                ! use symmetry for "V" Chirp
          time(ncp+i)=2.0*ta-time(ncp-i)
1800  continue


c     ** OVERLAPS **

1900  ovmax=0.0
      do 2000 i=2,nf,2
          t=time(i)
          if (ishape.eq.2.and.t.gt.ta) t=2.0*ta-t
          call freq
          ov(i/2)=fi**(-1.5)
```

```
          if (ov(i/2).gt.ovmax) ovmax=ov(i/2)
2000    continue

        do 2100 i=2,nf,2                ! nomalize to acoustic beamwidth
          ov(i/2)=ov(i/2)/ovmax*height
2100     continue

c       ** WIDTHS & X POSITIONING **

        do 2200 i=1,nf
          t=(time(i)+time(i+1))/2
          t1=t
          if (ishape.eq.2.and.t.gt.ta) t=2.0*ta-t
          call freq
          w(i)=1.0/8.0/fi*fo
          w(nf+i)=w(i)
          x(i)=t1*fo-w(i)/2.0
          x(nf+i)=x(i)
2200    continue

c       ** HEIGHTS & Y POSITIONING **

        rk=height+1.0                   ! rk is the distance between bus bars
2250    h(1)=(rk+ov(1)+bbh)/2.0
        h(nf+1)=rk+bbh-h(1)-w(1)
        y(1)=rk+3.0*bbh/2.0-h(1)
        y(nf+1)=bbh/2.0
        do 2300 i=2,(nf-2),2
          h(i)=rk+ov(i/2)+bbh-h(i-1)
          h(nf+i)=rk+bbh-h(i)-w(i)
          h(i+1)=h(i)
          h(nf+i+1)=h(nf+i)
          r=i
          if ((r/4.0).gt.(i/4)) then
            y(i)=bbh/2.0
            y(nf+i)=rk+3.0*bbh/2.0-h(nf+i)
            endif
          if ((r/4.0).eq.(i/4)) then
            y(i)=rk+3.0*bbh/2.0-h(i)
            y(nf+i)=bbh/2.0
            endif
          y(i+1)=y(i)
          y(nf+i+1)=y(nf+i)
2300    continue
        h(nf)=rk+ov(nf/2)+bbh-h(nf-1)
        h(2*nf)=rk+bbh-h(nf)-w(nf)
        y(nf)=bbh/2.0
        y(2*nf)=rk+3.0*bbh/2.0-h(2*nf)

c        Make sure it's ok. If an electrode less than half
c        the bus bar height then we need more distance between
c        the bus bars to make it implementable (though this
c        does not guarantee a practical design).
```

```
      iflag=0
      do 2400 i=1,(2*nf)
          if (h(i).lt.(bbh/2.0)) then
              iflag=iflag+1
              endif
2400  continue
      if (iflag.gt.0) then
          rk=rk+1.0
          if (rk.gt.2.0*height) then
              write(*,*) 'distance between bus bars is now:'
              write(*,*) rk, ' wavelengths'
              write(*,*) ' '
              write(*,*) 'acoustic beamwidth is:'
              write(*,*) height, 'wavelengths'
              write(*,*) ' '
              write(*,*) 'Enter < CNTRL Y > if you want to abort'
              endif
          goto 2250
          endif

c     ** BUS BARS **

      x(2*nf+1)= -0.25
      x(2*nf+2)= -0.25
      if (ishape.eq.1) w(2*nf+1)=ta*fo+0.5
      if (ishape.eq.2) w(2*nf+1)=2.0*ta*fo+0.5
      w(2*nf+2)=w(2*nf+1)
      h(2*nf+1)=bbh
      h(2*nf+2)=bbh
      y(2*nf+1)=0
      y(2*nf+2)=rk+bbh

c     ** NON-DISPERSIVE TRANSDUCER **

      x(2*nf+3)=w(2*nf+1)+d*0.8
      x(2*nf+4)=w(2*nf+2)+d*0.8
      w(2*nf+3)=utw+d*0.4
      w(2*nf+4)=utw+d*0.4
      h(2*nf+3)=bbh
      h(2*nf+4)=bbh
      y(2*nf+3)=0.0
      y(2*nf+4)=rk+bbh
      do 2500 i=1,(utw*4)
          ri=i
          x(2*nf+4+i)=x(2*nf+3)+1.0/16.0+(ri-1.0)/4.0+d*0.2
          w(2*nf+4+i)=1.0/8.0
          h(2*nf+4+i)=(rk+height+bbh)/2.0
          if (i.eq.1.or.i.eq.4.or.i.eq.5) then
              y(2*nf+4+i)=rk+3.0*bbh/2.0-h(2*nf+4+i)
              endif
          if (i.eq.2.or.i.eq.3.or.i.eq.6) then
              y(2*nf+4+i)=bbh/2.0
```

```
                endif
            x(2*nf+10+i)=x(2*nf+4+i)
            w(2*nf+10+i)=1.0/8.0
            h(2*nf+10+i)=rk+bbh-h(2*nf+4+i)-1.0/8.0
            if (i.eq.1.or.i.eq.4.or.i.eq.5) then
                y(2*nf+10+i)=bbh/2.0
                endif
            if (i.eq.2.or.i.eq.3.or.i.eq.6) then
                y(2*nf+10+i)=rk+3.0*bbh/2.0-h(2*nf+10+i)
                endif
2500    continue
        x(isnum-2)=w(2*nf+1)+0.4*d
        w(isnum-2)=d/5.0
        h(isnum-2)=rk+2.0*bbh
        y(isnum-2)=0.0

c       ** SORTING **

        do 2600 i=1,(isnum-2)
            s(i)=x(i)
2600    continue
        x(1)=s(2*nf+1)
        x(2)=s(2*nf+3)
        x(3)=s(2*nf+4)
        x(4)=s(2*nf+2)
        j=0
        do 2700 i=5,(2*nf+3),2
            j=j+1
            x(i)=s(j)
2700    continue
        j=nf
        do 2800 i=6,(2*nf+4),2
            j=j+1
            x(i)=s(j)
2800    continue
        x(2*nf+5)=s(isnum-2)
        j=2*nf+4
        do 2900 i=(2*nf+6),(isnum-3),2
            j=j+1
            x(i)=s(j)
2900    continue
        j=2*nf+4+utw*4.0
        do 3000 i=(2*nf+7),(isnum-2),2
            j=j+1
            x(i)=s(j)
3000    continue
        do 3100 i=1,(isnum-2)
            s(i)=y(i)
3100    continue
        y(1)=s(2*nf+1)
        y(2)=s(2*nf+3)
        y(3)=s(2*nf+4)
        y(4)=s(2*nf+2)
        j=0
```

```
      do 3200 i=5,(2*nf+3),2
          j=j+1
          y(i)=s(j)
3200  continue
      j=nf
      do 3300 i=6,(2*nf+4),2
          j=j+1
          y(i)=s(j)
3300  continue
      y(2*nf+5)=s(isnum-2)
      j=2*nf+4
      do 3400 i=(2*nf+6),(isnum-3),2
          j=j+1
          y(i)=s(j)
3400  continue
      j=2*nf+4+utw*4.0
      do 3500 i=(2*nf+7),(isnum-2),2
          j=j+1
          y(i)=s(j)
3500  continue
      do 3600 i=1,(isnum-2)
          s(i)=w(i)
3600  continue
      w(1)=s(2*nf+1)
      w(2)=s(2*nf+3)
      w(3)=s(2*nf+4)
      w(4)=s(2*nf+2)
      j=0
      do 3700 i=5,(2*nf+3),2
          j=j+1
          w(i)=s(j)
3700  continue
      j=nf
      do 3800 i=6,(2*nf+4),2
          j=j+1
          w(i)=s(j)
3800  continue
      w(2*nf+5)=s(isnum-2)
      j=2*nf+4
      do 3900 i=(2*nf+6),(isnum-3),2
          j=j+1
          w(i)=s(j)
3900  continue
      j=2*nf+4+utw*4.0
      do 4000 i=(2*nf+7),(isnum-2),2
          j=j+1
          w(i)=s(j)
4000  continue
      do 4100 i=1,(isnum-2)
          s(i)=h(i)
4100  continue
      h(1)=s(2*nf+1)
      h(2)=s(2*nf+3)
      h(3)=s(2*nf+4)
      h(4)=s(2*nf+2)
```

```
      j=0
      do 4200 i=5,(2*nf+3),2
          j=j+1
          h(i)=s(j)
4200  continue
      j=nf
      do 4300 i=6,(2*nf+4),2
          j=j+1
          h(i)=s(j)
4300  continue
      h(2*nf+5)=s(isnum-2)
      j=2*nf+4
      do 4400 i=(2*nf+6),(isnum-3),2
          j=j+1
          h(i)=s(j)
4400  continue
      j=2*nf+4+utw*4.0
      do 4500 i=(2*nf+7),(isnum-2),2
          j=j+1
          h(i)=s(j)
4500  continue

c     ** SCALING **

      if (iscale.eq.1) sf=1.0
      if (iscale.eq.2) sf=rlo*1000.0
      if (iscale.eq.3) sf=rlo*1.0e6
      if (iscale.eq.4) sf=rlo*1.0e5/2.54
      do 4600 i=1,isnum
          x(i)=(x(i)-xmin)*sf
          y(i)=(y(i)-ymin)*sf
          w(i)=w(i)*sf
          h(i)=h(i)*sf
4600   continue
4700   do 4800 i=1,isnum
          a(i)=0.0
          irep(i)=1
          xd(i)=0.0
          yd(i)=0.0
4800   continue

      return
      end

c     ------------------------------------------------------------

      subroutine freq

c     This subroutine finds the instantaneous
c     frequency of the chirped waveform

      common/cheq/  akf,arg,fi,ifm,pi,t,ta,wi,wl
```

```
      if (ifm.eq.1) wi=wl+akf*t/ta
      if (ifm.eq.2) then
          wi=wl+akf*(0.43+0.5*cos(pi*(t-ta)/ta)
     2       +0.07*cos(2.0*pi*(t-ta)/ta))
          endif
      fi=wi/2.0/pi

      return
      end

c     ------------------------------------------------------------

      subroutine argument

c     Subroutine calculates the argument of chirp waveform
c     equation. Used by calling program to find critical
c     points. If seeking a null of the waveform, sin(arg)
c     is used. If seeking a peak or valley, the derivative
c     of the function, cos(arg), is used. Either way, the
c     calling program is looking for the overall result to
c     go to zero.

      common/cheq/  akf,arg,fi,ifm,pi,t,ta,wi,wl

      if (ifm.eq.1) then
          arg=wl*t+akf/2.0/ta*t*t
          endif
      if (ifm.eq.2) then
          arg=wl*t+akf*(0.43*t+0.5*ta/pi*sin(pi*(t-ta)/ta)+
     2       (0.07*ta)/(2.0*pi)*sin(2*pi*(t-ta)/ta))
          endif

      return
      end

c     ------------------------------------------------------------

      integer function sgn(x)

      if (x.eq.0.0) sgn=0
      if (x.gt.0.0) sgn=1
      if (x.lt.0.0) sgn=-1

      return
      end

c     ------------------------------------------------------------

      subroutine prnt

      common/pat1/ x(6000),y(6000),w(6000),h(6000),a(6000),
     2 irep(6000),xd(6000),yd(6000),isnum,iref
```

```
      return
      end

c     -----------------------------------------------------------
c     this is just a modified version of subroutine writeo

      subroutine savit
      common/pat1/ x(6000),y(6000),w(6000),h(6000),a(6000),
     2 irep(6000),xd(6000),yd(6000),isnum,iref

      character filout*10

c     terminal input

100   write(6,*) ' <<< SYSTEM WRITE >>>'
      write(6,*)' '
      write(6,1005)
1005  format(x,'ENTER output file name : ==> ',$)
      read(5,1000,err=10) filout
1000  format(a10)

      open(10,file=filout,status='unknown',err=10)
      close(10,status='delete')
      open(10,file=filout,status='unknown',err=10)
      goto 20
10    write(6,*)' *** ERROR in file name - try again ***'
      goto 100

c     initialize data

20    icont=11
      itype=-1
      fo=1.0
      tflo=-1.0
      tfhi=1.0
      num=2
      zero=0.0
      iref=1

c     write out file data

      write(10,2000) icont
2000  format(x,'icont=',i5)

      write(10,2001) itype
2001  format(x,'itype=',i4)

      write(10,2002) fo
```

```
2002    format(x,'fo   =',e17.9)

        write(10,2003) tflo
2003    format(x,'tflo =',e17.9)

        write(10,2004) tfhi
2004    format(x,'tfhi =',e17.9)

        write(10,2005) num
2005    format(x,'num  =',i4)

        write(10,*) zero,zero
        write(10,*) zero,zero

        write(10,2006) isnum
2006    format(x,'isnum=',i5)

        write(10,2007) iref
2007    format(x,'iref =',i1)

        do 400 i=1,isnum
           write(10,*) x(i),y(i),w(i),h(i),a(i),irep(i),xd(i),yd(i)
400     continue


        close(10)

        return
        end

c       ------------------------------------------------------------
```

```
      subroutine sbreak2(nmark,nflash)

c     ---------------------------------------------------------------

c          For information contact: D.C. Malocha
c                                   S.M. Richie
c                                   J.C. Walker

c     ---------------------------------------------------------------

c          Date of last revision: August 19, 1986

c     ---------------------------------------------------------------

c     This subroutine is used when generating magtapes using the
c     ELECTROMASK (SAWTEK) format. Structure data (body centered, mm)
c     is taken and new rectangles are generated to eliminate repeti-
c     tions. Angle data is resolved such that it is in the range from
c     0 to 90 degrees; swapping height and width data if necessary.
c     Rectangles are broken up into smaller rectangles if necessary
c     to satisfy maximum aperture limitations.

c     ---------------------------------------------------------------


      integer nrec(99)
      character chstr*11
      real*8 chnum
      byte sbuf(512,2000)

      common/chconv/ chstr,chnum
      common/pat1/ x(36000),y(36000),w(36000),h(36000),a(36000),
      2            irep(36000),xd(36000),yd(36000),isnum,iref

      common/buffer/ sbuf
      common/address/ numrec,numbyte,nrec,ntape
      common/flash/ xg,yg,wg,hg,ag,irg,xi,xmin,xmax,ymin,ymax


c     maximum aperture for SAWTEK is 1.524 mm

      apmax=15240                  ! units are tenths of micrometers
      pi=4*atan(1.0)


c      load first begining of record symbol

       if(numrec.gt.1.or.numbyte.gt.1)then
           numbyte=1
           numrec=numrec+1
       endif
       istrlen=1
```

```
      chstr(1:1)='<'
      call ldsbuf(istrlen)
      nmark=nmark+1

c     angle data is resolved by first putting the data in the range
c     from -180 to 180 degrees then adjusting the angle (and height
c     and width data if necessary) such that all angle data is in the
c     range from 0 to 90 degrees (angle data is in tenths of degrees)

      do 1000 i=1,isnum
500       if(a(i).gt.1800)a(i)=a(i)-3600
          if(a(i).lt.-1800)a(i)=a(i)+3600
          if(a(i).gt.1800.or.a(i).lt.-1800)goto 500
          if(a(i).lt.0)a(i)=a(i)+1800
          if(a(i).ge.900)then
              a(i)=a(i)-900
              ww=w(i)
              w(i)=h(i)
              h(i)=ww
          endif
1000  continue

c     initialize

      do 5000 i=1,isnum
         wg=w(i)
         hg=h(i)
         ag=a(i)
         agr=a(i)*pi/1800        ! convert tenths of degrees to radians
         irg=irep(i)
         nwrep=1
         nhrep=1
         wov=0
         hov=0
         if(wg.gt.apmax) then
             nwrep=wg/apmax+1
1100         wov=(nwrep*apmax-wg)/(nwrep-1)
             if(wov.lt.10.0)then
                 nwrep=nwrep+1
                 goto 1100
             endif
             wg=apmax
         endif
         if(hg.gt.apmax)then
             nhrep=hg/apmax+1
1200         hov=(nhrep*apmax-hg)/(nhrep-1)
             if(hov.lt.10.0) then
                 nhrep=nhrep+1
                 goto1200
             endif
             hg=apmax
         endif

         do 4000 j=0,(irg-1)                          ! repeats
```

```
          do 3000 k=0,(nwrep-1)                        ! widths
              do 2000 l=0,(nhrep-1)                    ! heights
                  xi=x(i)+xd(i)*j
                  yi=y(i)+yd(i)*j
                  if(agr.eq.0)then
                      xg=xi-w(i)/2+wg/2+(wg-wov)*k
                      yg=yi-h(i)/2+hg/2+(hg-hov)*l
                      goto 1500
                  endif
                  xip=xi*cos(agr)+yi*sin(agr)
                  yip=yi*cos(agr)-xi*sin(agr)
                  xgp=xip-w(i)/2+wg/2+(wg-wov)*k
                  ygp=yip-h(i)/2+hg/2+(hg-hov)*l
                  xg=xgp*cos(agr)-ygp*sin(agr)
                  yg=ygp*cos(agr)+xgp*sin(agr)
1500              if(numbyte.gt.472)then
                      sbuf(numbyte,numrec)='.'
                      sbuf(1,numrec+1)='<'
                      numbyte=2
                      numrec=numrec+1
                  endif
                  call flash2(nmark,nflash)
2000          continue
3000      continue
4000    continue
5000  continue

c     load end of record symbol for last rec of file

      istrlen=1
      chstr(1:1)='.'
      call ldsbuf(istrlen)
      nmark=nmark+1

      return
      end
```

REFERENCES

Booher, Ronald A. "New Non-Linear FM Pulse-Compression Technique for Radar Applications." Master's Thesis, University of Central Florida, Orlando, 1985.

Cook, Charles E., and Bernfeld, Marvin. Radar Signals - An Introduction to Theory and Application. New York: Academic Press, 1967.

Hartmann, C.S.; Bell, D.T., Jr.; and Rosenfeld, R. "Impulse Model Design of Acoustic Surface-Wave Filters." IEEE Transactions on Sonics and Ultrasonics 20 (April 1973): 81-93.

Malocha, D.C., and Richie, S.M. "Computer Aided Design of Surface Acoustic Wave Bi-directional Transducers." Final Report to Texas Instruments, March 1984.

Rihaczek, August W. Principles of High Resolution Radar. New York: McGraw Hill Book Company, 1969.

Tzannes, Nicolaos S. Communication and Radar Systems. Englewood Cliffs, NJ: Prentice Hall, 1985.

Vigil, A.J., and Yap, R.L. "Fabrication of Surface Acoustic Wave Devices." National Vacuum Society Conference, St. Petersburg, FL, 1984.

Wheeler, Gershon H. Radar Fundamentals. Englewood Cliffs, NJ: Prentice Hall, 1967.

Ziemer, R.E., and Tranter, W.H. Principles of Communications. Boston: Houghton Mifflin Company, 1985.