A DESIGN ALGORITHM FOR ARBITRARY PHASE AND AMPLITUDE SURFACE ACOUSTIC WAVE FILTERS

by

MARK S. SUTHERS, B.ENG., M.ENG.

A Thesis

Submitted to the School of Graduate Studies

in Partial Fulfilment of the Requirements

for the Degree

Doctor of Philosophy

McMaster University

1978

ARBITRARY PHASE AND AMPLITUDE SURFACE ACOUSTIC WAVE FILTERS

DOCTOR OF PHILOSOPY (1978)
(Electrical Engineering)

McMASTER UNIVERSITY
Hamilton, Ontario

TITLE: A Design Algorithm for Arbitrary Phase and Amplitude Surface Acoustic Wave Filters

AUTHOR: Mark S. Suthers, B.Eng. (McMaster University)
M.Eng. (McMaster University)

SUPERVISOR: Professor C.K. Campbell

ABSTRACT

The subject of this study is the development of a computer-aided-design tool that filter designers could use to obtain nonlinear phase and nonsymmetric amplitude responses for surface acoustic wave devices. To accomplish this, a design algorithm was developed that uses a Fast Fourier Transform to realize the time domain transform of the frequency domain specifications required by the designer. From this time domain solution an interdigital transducer (IDT) is synthesized that will approximate the design specifications. The next step in the algorithm is to analyse the synthesized IDT using an accurate charge distribution model for the IDT (the charge distribution model is developed in this thesis which gives the frequency domain response for the IDT). The frequency response is then compared to the input specifications so as to generate an error function. This error function is then used to predistort the input frequency domain specifications and the whole synthesis/analysis procedure is repeated iteratively until an acceptable solution is obtained. The algorithm is programmed and used to design 2 filters (3 separate IDT designs) for the nonlinear phase and nonsymmetric amplitude specifications for television intermediate frequency (TVIF) specifications. Using these designs, two filters were fabricated and tested to show the validity of the algorithm.

TABLE OF CONTENTS

TABLE OF CONTENTS (continued)

TABLE OF CONTENTS (continued)

LIST OF FIGURES

LIST OF FIGURES (continued)

ACKNOWLEDGEMENTS

The author would like to express his appreciation for the consideration and supervision received from the supervisor, Dr. C.K. Campbell. The author also acknowledges the constructive criticism and interest of the other members of the supervisory committee: Dr. S. Haykin and Dr. J.P. Marton.

I would like to thank my colleagues: Messrs. P. Edmonson, P. Nanayakkara, J.P. Reilly and D.G. Seiler, who provided a stimulating work environment.

I also acknowledge: Mr. C. Goulson and the Canadian Institute for Metal Working for the use of their terminal; Mr. R. Martin and Mr. J. Jewell of the McMaster Computing Centre for their help with permanent file manipulation and management. The congeniality of all of the Computing Centre's operations staff is also very much appreciated.

Mr. R. Grant of the Electrical Engineering Department of the University of Waterloo is thanked for his help in the production of the photo-masks. The artwork of Mr. G. Kappel is much appreciated.

I also thank P. Dillon of the Word Processing Centre for the typing of this manuscript and finally I would like to thank my wife, Carol, for seeing me through this ordeal.

CHAPTER 1

INTRODUCTION

1.1 Principles of Surface Acoustic Waves

Surface Acoustic Wave (SAW) devices (Figure 1.1) are essentially transversal filters which, in the transmitting mode, are analogous to end fire antenna arrays (Figure 1.2). The frequency response in the end fire direction is a function of the spacing between the elements and the relative amplitude and phase of the fields produced by each elemental source. This response is given by:

$$H(\omega) = \sum_{\ell=1}^{L} A_\ell \, e^{-j(k_x x_\ell + \phi_\ell)} \tag{1.1}$$

where: $H(\omega)$ is the frequency response along the x-axis

A_ℓ is the relative amplitude of the ℓ^{th} element

k_x is the propagation constant along the x-axis

x_ℓ is the position of the ℓ^{th} element measured from the summation point

ϕ_ℓ is the relative phase of the ℓ^{th} element.

Equation (1.1) is a summation of exponentials and it may be noted that there are no poles (frequency at which the denominator goes to zero) in the expression. This is due to the fact that the receiving array is not electrically connected to the transmitting array. This gives an added degree of freedom from lumped element filter design.

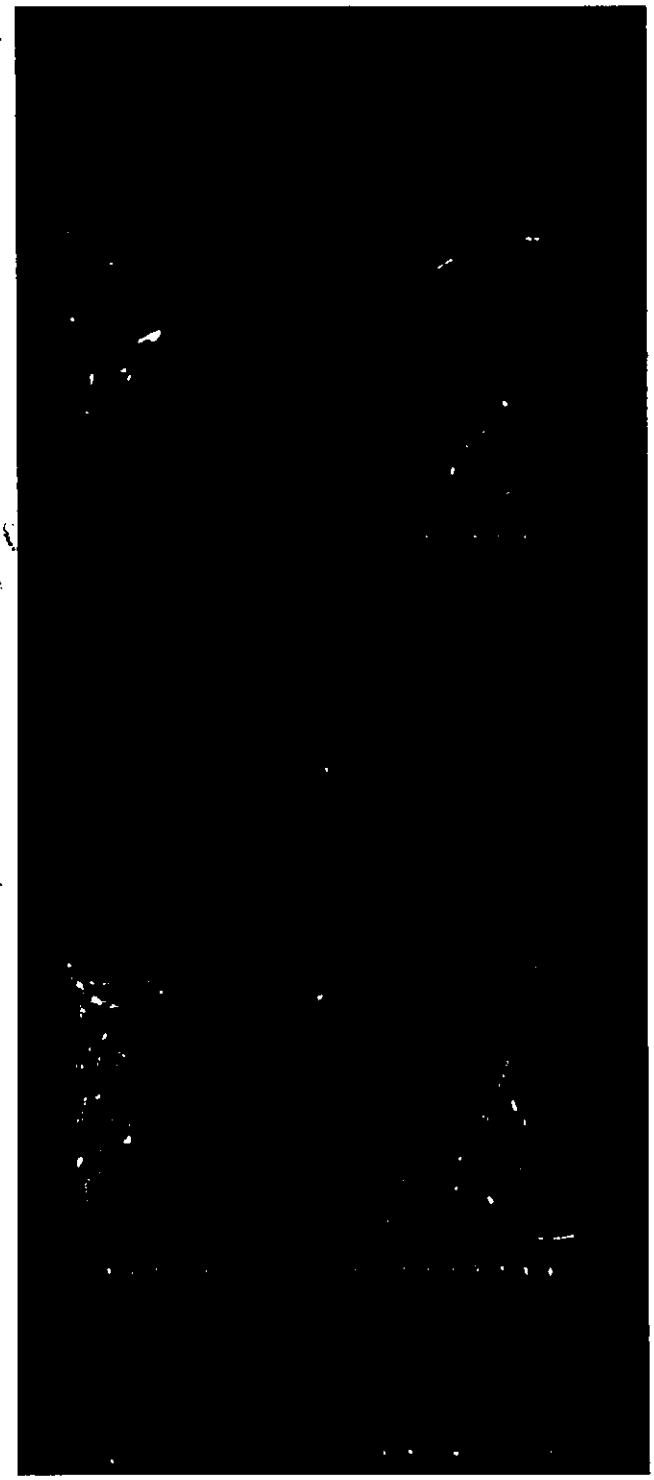


Figure 1.1: Surface Acoustic Wave Filters

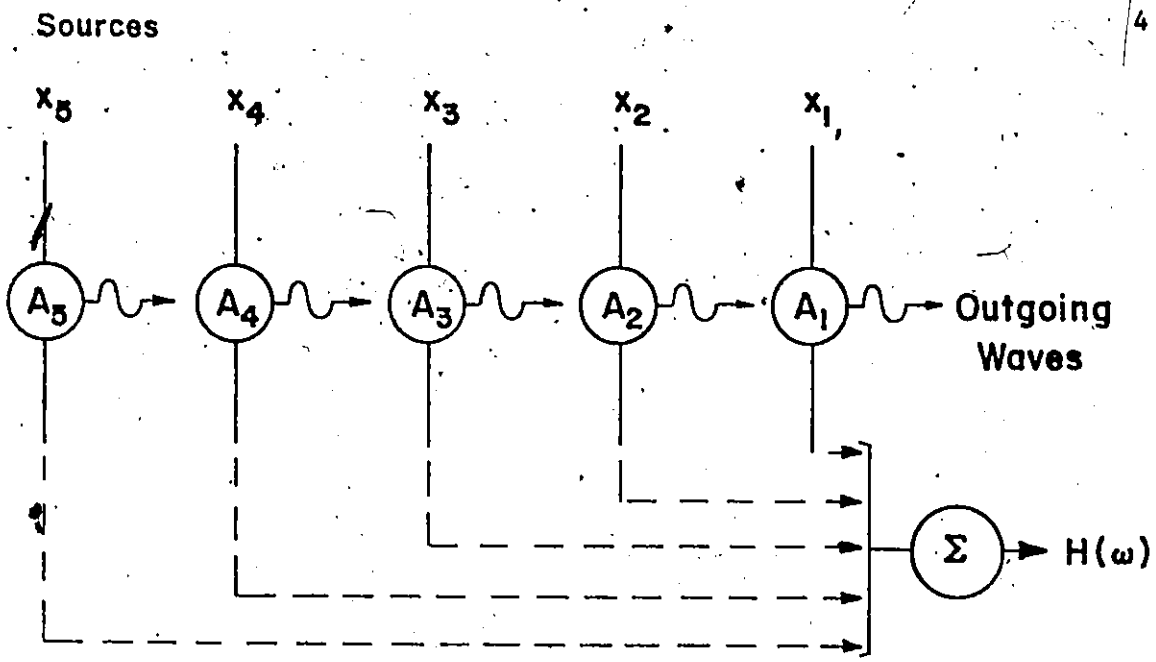
In the receiving mode, the transducer is similar to a finite-impulse response (FIR) digital filter as seen in Figure 1.2 where the taps are allowed to vary in position and are weighted as to their relative sample amplitude. The response of this receiving system is given by an equation of the same form as (1.1).

Thus by varying the lengths of the transducer fingers (elemental amplitude or tap weight) and by varying the finger position (elements position in the array or sample time in the digital filter analogy), a desired frequency response may be met. The above analogies are the simplest explanations possible which explain how SAW devices produce filtering action.

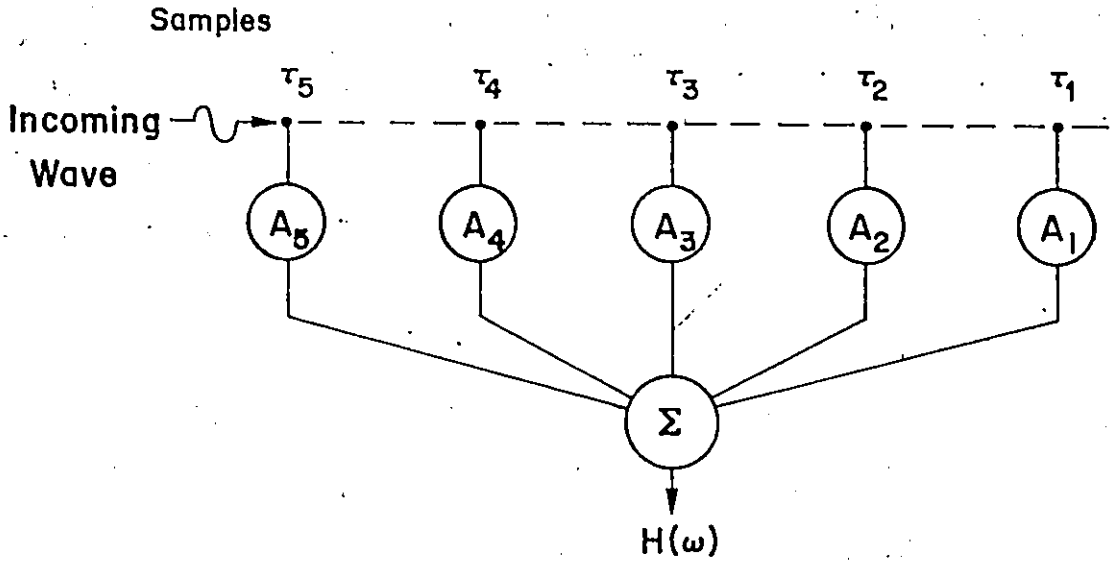
1.2 Why SAW Filters?

This question is answered by the fact that these devices may be made economically, and small, and they do not require alignment once the device is made. Since the acoustic wavelength on the surface of the substrate is approximately 10^{-5} times the wavelength of the corresponding electro-magnetic wave, the principles of microwaves may be applied to the 10 MHz to 1 GHz band of frequencies. Also, the technology of production is planar lithography, i.e. micro-electronic fabrication procedures, which are well established as precise economical production techniques.

The physical structure of a surface acoustic wave (SAW) filter is shown in Figure 1.3. Since the waves are very short (35 μm at 100 MHz), it is necessary that the surface of the substrate be very smooth.

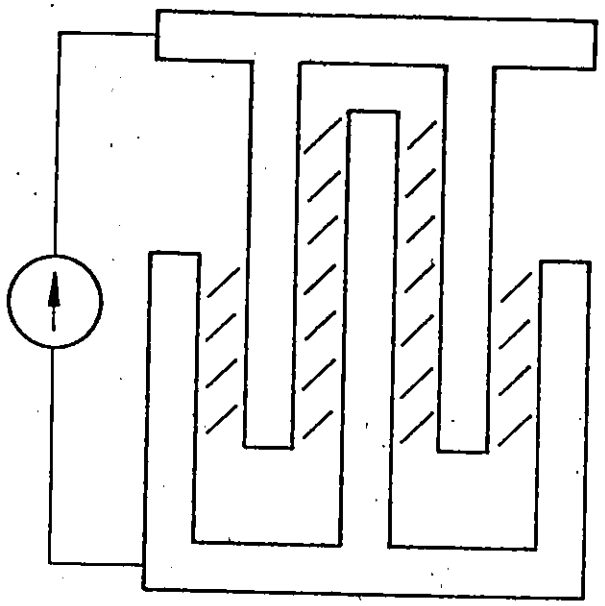


End-fire antenna array transmitter

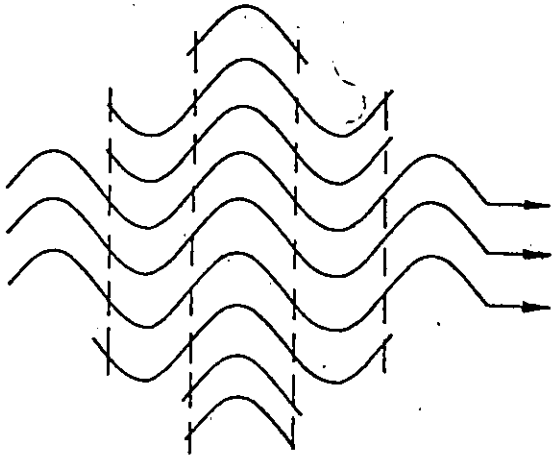


Finite impulse response digital filter

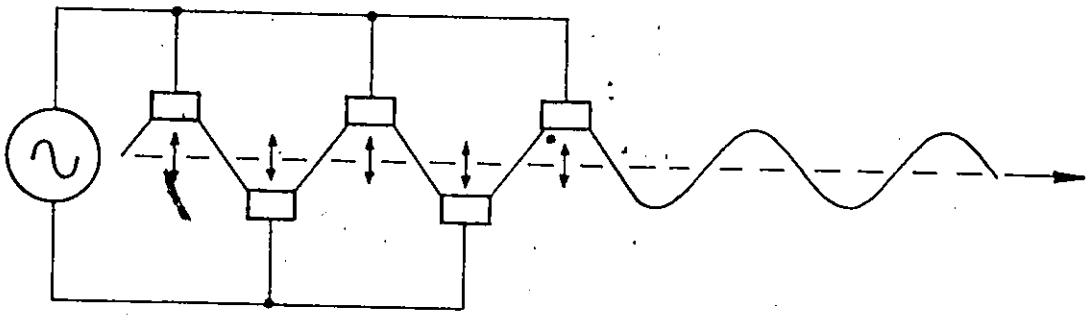
Figure 1.2: Antenna end-fire array and FIR digital filter analogies to SAW filters



Top view of interdigital transducer geometry (aluminum finger pattern showing apodization)



Surface acoustic wave energy distribution leaving IDT region due to impulse excitation



Side view showing IDT fingers causing substrate surface distortion

Surface acoustic wave leaving IDT region at resonant frequency of transducer

Figure 1.3: Interdigital transducer interacting with piezoelectric substrate

Any irregularities in the surface will tend to scatter energy and thus degrade the device response. As the name implies, surface acoustic waves are mechanical in nature and therefore the electrical-to-mechanical and mechanical-to-electrical transduction must be accomplished simply and efficiently. The most obvious, and thus most common, solution to this transduction requirement is to use piezoelectric substrates. Since application of an electric field to a piezoelectric material causes a mechanical deformation to the material then it is seen from Figure 1.3 that the application of voltages to the aluminum fingers on the surface of the substrate will cause a periodic deformation. If the applied voltage is alternated in a sinusoidal manner then at those frequencies at which constructive interference occurs, acoustic radiation will develop causing mechanical waves to propagate along the surface. Since surface acoustic wave velocities are slower than bulk wave velocities, the wave deformation in the interior of the crystal tends to propagate ahead of the surface deformation for a given wave crest. This causes a "dragging" effect on the wave which tends to pull the wave toward the surface of the crystal and is the cause for approximately 90% of the acoustic energy being contained within 1.5 λ_s (where λ_s = surface acoustic wave length) of the surface [1].

From [2], the piezoelectric transduction process is given by the following pair of equations:

$$[S] = [\underline{d}] \cdot [E] + [s^E] : [T] \qquad (1.2a)$$

$$[D] = \epsilon_o \cdot [\epsilon^S] \cdot [E] + [e] : [S] \qquad (1.2b)$$

where: [S] is the 1x6 stress vector measured in meters/meter i.e. dimensionless

[d] is the 3x6 piezoelectric strain tensor measured in meters/volt

[E] is the 1x3 electric field vector in volts/meter

$[s^E]$ is the 6x6 compliance tensor for constant E field measured in $meters^2/Newton$

[T] is the 1x6 stress vector measured in $Newtons/meter^2$

[D] is the 1x3 electric displacement density vector measured in $coulombs/meter^2$

$[\varepsilon^S]$ is the 3x3 relative permittivity tensor at constant stress (dimensionless)

$[\varepsilon_o]$ is the permittivity of free space

[e] is the 6x3 piezoelectric stress constant tensor in $coulombs/meter^2$

The point to note in these equations (1-2) is that

$$[S] = [d] \cdot [E] \qquad \text{for } [T] = 0 \qquad (1.3a)$$

$$[D] = [e] : [S] \qquad \text{for } [E] = 0 \qquad (1.3b)$$

Thus from (1.3a), strain (S) is a function of an applied electric field (E) through the piezoelectric strain coefficients (d). Therefore, the mechanical deformation is directly related to charge applied through metallized regions on the surface which gives the electrical to mechanical transduction process.

Similarly, from (1.3b) the electric displacement density (D) is

a function of the mechanical strain (S) through the piezoelectric stress coefficients (e) thus relating the charge at the receiving transducer to the mechanical deformation caused by an incident wave.

1.3 Design Techniques

Referring to (1.1), it is noted that there is a strong resemblance to the frequency domain response being related to the spatial domain of the transducer fingers through the Fourier transformation in a truncated discrete form. Initially, attempts at a desired frequency domain response were obtained by evaluating the inverse Fourier transform. From the infinite time domain solution, a window function was applied to realize a finite solution. Then the finger length was taken as a function of the amplitude of the inverse Fourier transform (IFT) and the finger's position at the $\pm n\pi$ points of the IFT. The polarity was obtained from the zero crossings of the phase response of the IFT. References such as [3], [4] and [5] illustrate this technique very well. This procedure is shown graphically in Figure 1.4. The variation in the length of the fingers, known as apodization, is equivalent to varying the amplitude of each finger of a constant overlap IDT since the acoustic amplitude in the far field is a function of the charge on the finger. Any technique of varying the charge on a finger in a manner equal to the amplitude of the inverse Fourier transform solution will give the desired results, but the most common technique is the use of apodization which is used in this study. Alternate schemes are: spectral weighting [6], series weighting [7] and

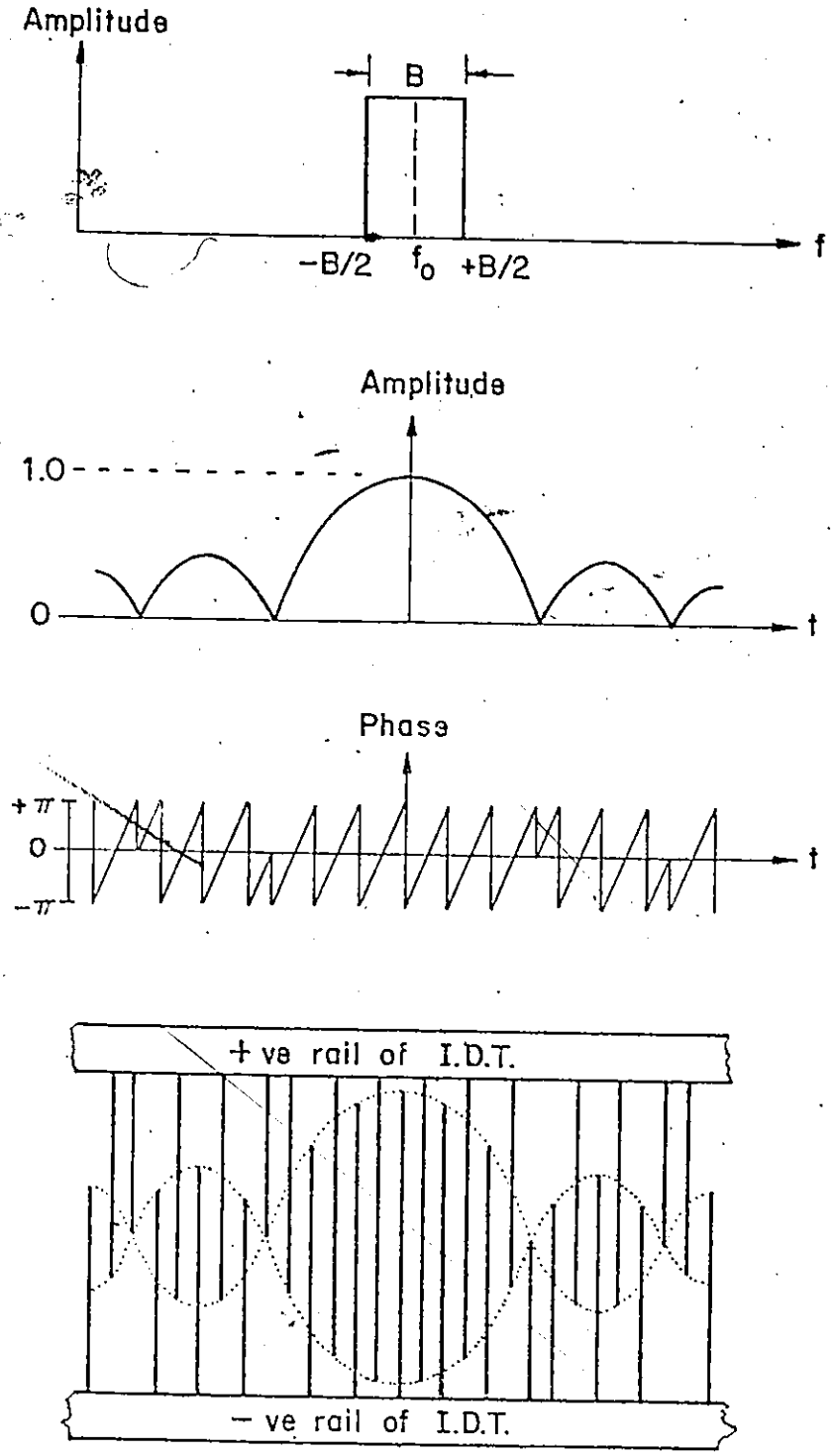


Figure 1.4: Obtaining IDT geometry from Fourier transform

withdrawal weighting [8].

Naturally, the next step from the classical Fourier transform design approach was to use a Fast Fourier Transform (FFT) [4], [9]. If the desired bandpass is symmetric about f_o in amplitude and linear in phase, then the bandpass may be translated to baseband and the FFT applied for the band from 0 Hz to some frequency above the band-edge. The resulting time domain transform is finite in duration and of course an appropriate window function may be applied to reduce passband ripple and to suppress side lobes [9], [10].

It must be noted at this point that the above techniques have been restricted to linear phase/symmetric bandpass filter specifications. While this is often desirable, there are many filters that require different specifications.

1.4 Arbitrary Phase

Working with baseband specifications, it is possible to obtain a deviation from linear phase, if the specifications are Hermitian. This technique is illustrated in [3] and [11]. If the frequency domain specification is symmetric in amplitude and anti-symmetric in phase about the centre frequency (Hermitian specifications) then, these specifications may be Inverse Discrete Fourier Transformed (IDFT) to give real equally-spaced samples. Since, most SAW filters to date have been designed on the basis of equally-spaced fingers, then this type of response is quite easily obtained using either the classical transform or the IDFT.

The above technique does not give a truly arbitrary phase response; it is restricted to an anti-symmetric phase response.

1.5 Arbitrary Amplitude

Split electrodes have come to be used in general so as to eliminate ripple due to reflections which add constructively at certain frequencies [1]. Since the spatial sampling with split electrodes is twice that of single electrodes, it was noted in [12] that the amplitude response could be made non-symmetric by giving different lengths to each half of the split electrode. In effect, the synchronous frequency (frequency at which finger spacing resonates) is at twice the centre frequency (f_o) of the desired response. There is also an image at the frequency $f = 3 f_o$. So long as the desired response and the image are constructed so as to be symmetric about the frequency $f = 2 f_o$ then the filter may easily be constructed using the previously discussed design techniques. Note that this response is also Hermitian (i.e. symmetric amplitude and anti-symmetric phase about the frequency $f = 2 f_o$).

1.6 Building Block

The next obvious development to realize a given response was to design a bank of filters that would be driven and detected in parallel. Each filter would be a piece-wise fit to the desired response. These concepts (minus experimental results) are given in [13]. From these concepts, evolved the idea of combining the building blocks into a single IDT (interdigital transducer) pattern, which was shown in [14].

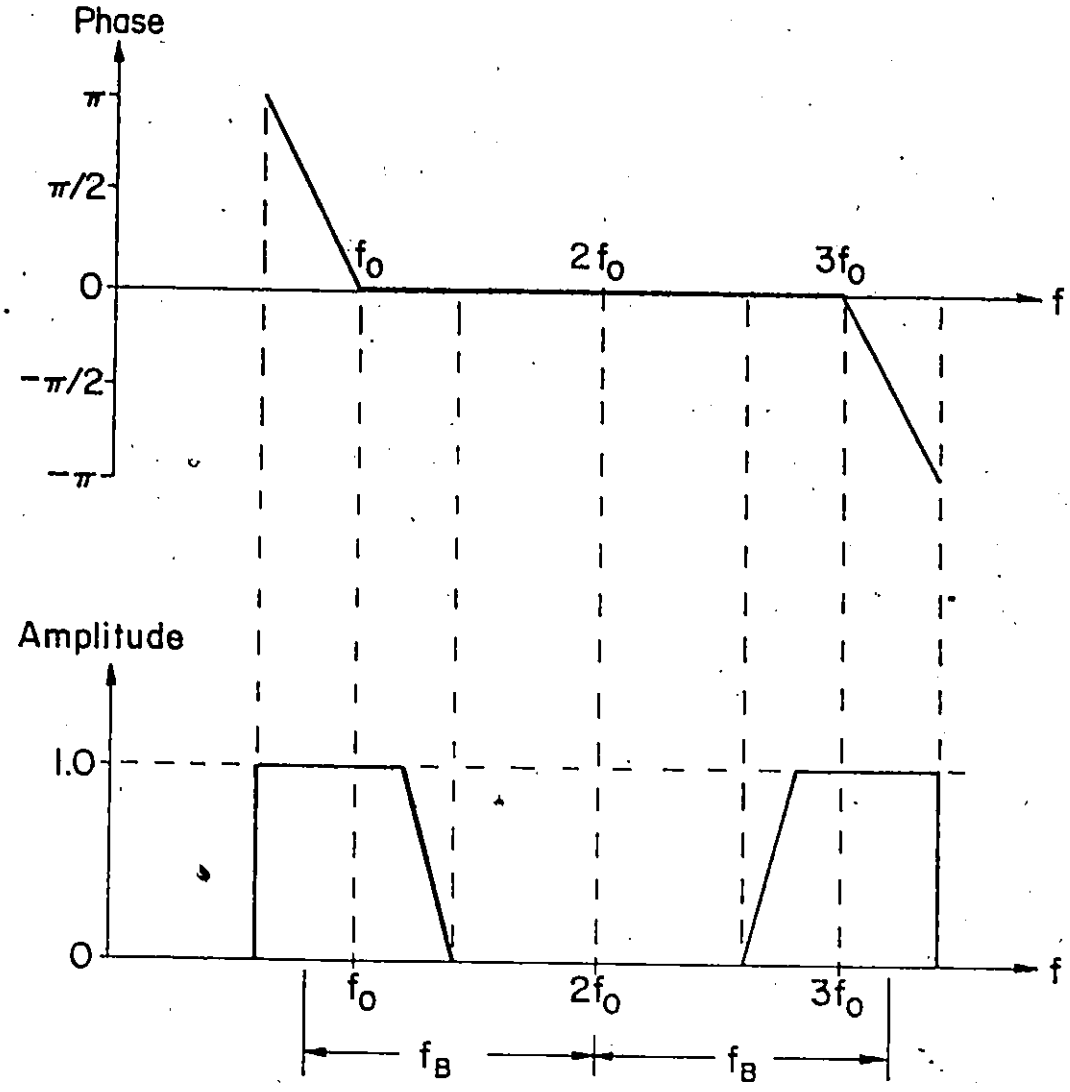
This technique has the advantage of being simple and effective in producing responses to specifications that are non-symmetric in amplitude and non-linear in phase; however, it is still a piece-wise fit.

1.7 Arbitrary Phase and Amplitude

The most elegant solution to the arbitrary Phase/Amplitude design problem is given in [15]. Essentially, this technique combines the concepts described in Sections 1.4 and 1.5. By using the split electrode concept as described before, a Hermitian function is constructed about $f = 2 f_o$ where f_o is the centre frequency of the desired bandpass.

The desired amplitude response must be imaged at $f = 3 f_o$ such that a symmetric amplitude function is obtained about $f = 2 f_o$. Similarly, the desired phase response must be anti-symmetrically imaged at $f = 3 f_o$ so that the resulting specifications give a real equally spaced solution when the inverse Fourier transform is obtained. The other interdigital transducer (IDT) is designed to select only the desired bandpass and to reject the image bandpass. Of course the other IDT may be designed as a single bandpass linear phase filter or it may have a dual bandpass specification so long as the higher frequency image is at some frequency other than the upper image of the first IDT.

These concepts are illustrated in Figure 1.5 and will be further developed in Chapter 7 where these concepts are refined.



Specifications are Hermitian about $f=2f_o$

if for any f_B:

the specification at $f=2f_o-f_B$ is A+jB then,

the specification at $f=2f_o+f_B$ is A-jB

Figure 1.5: Dual bandpass Hermitan specifications about $f=2f_o$

1.8 Scope of Thesis

Since the development of surface acoustic wave (SAW) devices for filter applications, there has been a need to have a design technique that allowed for arbitrary amplitude and phase specifications. The easiest, and thus the first SAW filters obtained, had linear phase and symmetric amplitude responses; however, many systems require a nonlinear phase and a non-symmetric amplitude response. The exceptions to the rule of linear phase/symmetric amplitude responses for SAW's have been the "chirp" filters used in pulse compression for RADAR systems and some other concepts that have been explained in this chapter. The responses of these filters were generally limited to linear phase/non-symmetric amplitude or anti-symmetric phase/symmetric amplitude. The only publication known to this author that describes a design algorithm for arbitrary phase/amplitude specifications is given by Reference [16]. This algorithm required a "cut-and-try" optimization loop since no modelling of the device was included in the algorithm. It must be emphasized that no synthesis procedure presently available will give a SAW response as predicted by the synthesis. Thus the algorithm as described in this thesis involves: (a) synthesis (b) modelling of the synthesized IDT geometry (interdigital transducer) (c) obtaining the frequency response of the IDT based on the model (d) comparing the response to the desired specification and thus creating new pre-distorted specifications to be used in a new synthesis. Thus, the geometry of the transducer is obtained in an iterative multipass optimization loop. This procedure is shown in Figure 1.6 illustrating

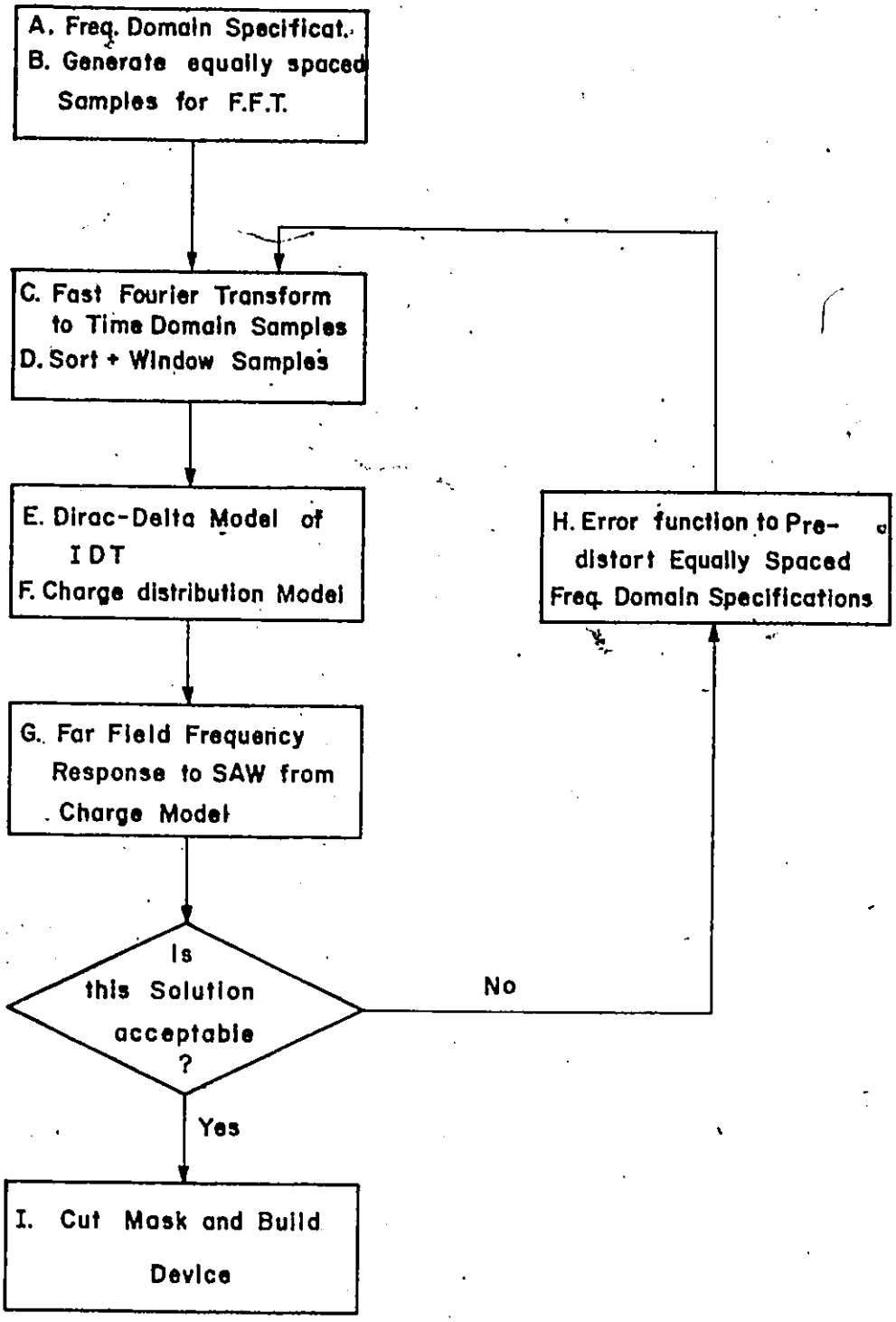


Figure 1.6: Algorithm Block Diagram

the design procedure.

1.9 Description of Algorithm

The algorithm described in this thesis begins with the specification in the frequency domain of the insertion loss and the phase characteristics desired for the filter (Step A of Figure 1.6). From these criteria, a set of frequency domain equally-spaced samples is generated (Step B) which is stored for future reference. These complex numbers are then fed into a Fast Fourier Transform (FFT) routine (Step C). The output of the transform is then sorted and a Kaiser window function is applied (Step D). The next step gives the Dirac-delta-model ([1], [2], [3], [4]) of the IDT by selecting the position and length of each finger of the IDT in the time domain (Step E) which is stored on the system mass storage. From this model, the physical charge distribution model is obtained for the transducer (Step F) by solving the equations for the electrostatic solution for the metallized transducer pattern. This is done in a numerical iterative manner. Once the physical model has been obtained, then the far-field frequency response is obtained (Step G) which is stored on the system. Plots of the specifications, the apodization (pattern showing finger lengths and positions) function and the responses are produced along with printout of the associated numerical values. If the response is not acceptable then the original specifications to the FFT are distorted in such a way such that the error is minimized (Step H) and the process is repeated from Step C. This process is repeated until an acceptable solution is

obtained at which point the mask for the IDT geometry may be made (Step I). From this point on, standard lithographic techniques are used to produce the SAW filter.

A description of the algorithm's significant elements is given including the following topics:

Chapter 2: - the synthesis of the transducer geometry using a Fast Fourier transform technique

Chapter 3: - development of the charge distribution model for the IDT that was synthesized in Chapter 2. The mathematical development is given in Appendix I.

Chapter 4: - development of the far-field frequency response analogous to an end fire antenna array.

Chapter 5: - development of the numerical optimization techniques that are used to minimize the design errors.

Chapter 6: - notes on programming of the algorithm.

Chapter 7: - experimental results of two attempts at television intermediate frequency (TVIF) filter specifications. The first filter has a single bandpass non-symmetric, nonlinear phase IDT in the device. The second filter uses a dual band Hermitian specification centred at 55 MHz for the IDT. Both filters have the same single bandpass rectangular linear phase receiving IDT.

Chapter 8: - conclusions and some suggestions of how the algorithm might further be developed and refined.

CHAPTER 2

SYNTHESIS OF THE INTERDIGITAL TRANSDUCER

2.1 The Fast Fourier Transform

An efficient method for obtaining the time domain transformation from the frequency domain specifications is the Fast Fourier Transform (FFT) algorithm. This procedure is adequately described in many references such as [9] and [17]. The actual transform routine may be written by the filter designer or, if available, a computer system scientific subroutine package may be used.

Typically, a FFT works with lowpass specifications, and then the frequency-shifting theorem is applied to produce the desired bandpass response. However, the lowpass specification restricts the design to a symmetric amplitude and anti-symmetric phase response as is seen in Figure 2.1. To overcome this limitation, it is necessary to consider the desired bandpass and sample both below and above the centre frequency. To increase the sample density in the region of interest, it is necessary to frequency translate the desired bandpass such that only the adjacent stop bands will be considered in the transformation. This is illustrated in Figure 2.2. After the transformation, the sample rate of the time domain solution may be increased so as to bring the passband up to the desired frequency. However, for example, if the region from f = 1/2 f_o to f = 1 1/2 f_o is transformed, then the mathematics imply image bandpasses at f = 0.0 Hz, 2 f_o etc. (where f_o is the centre

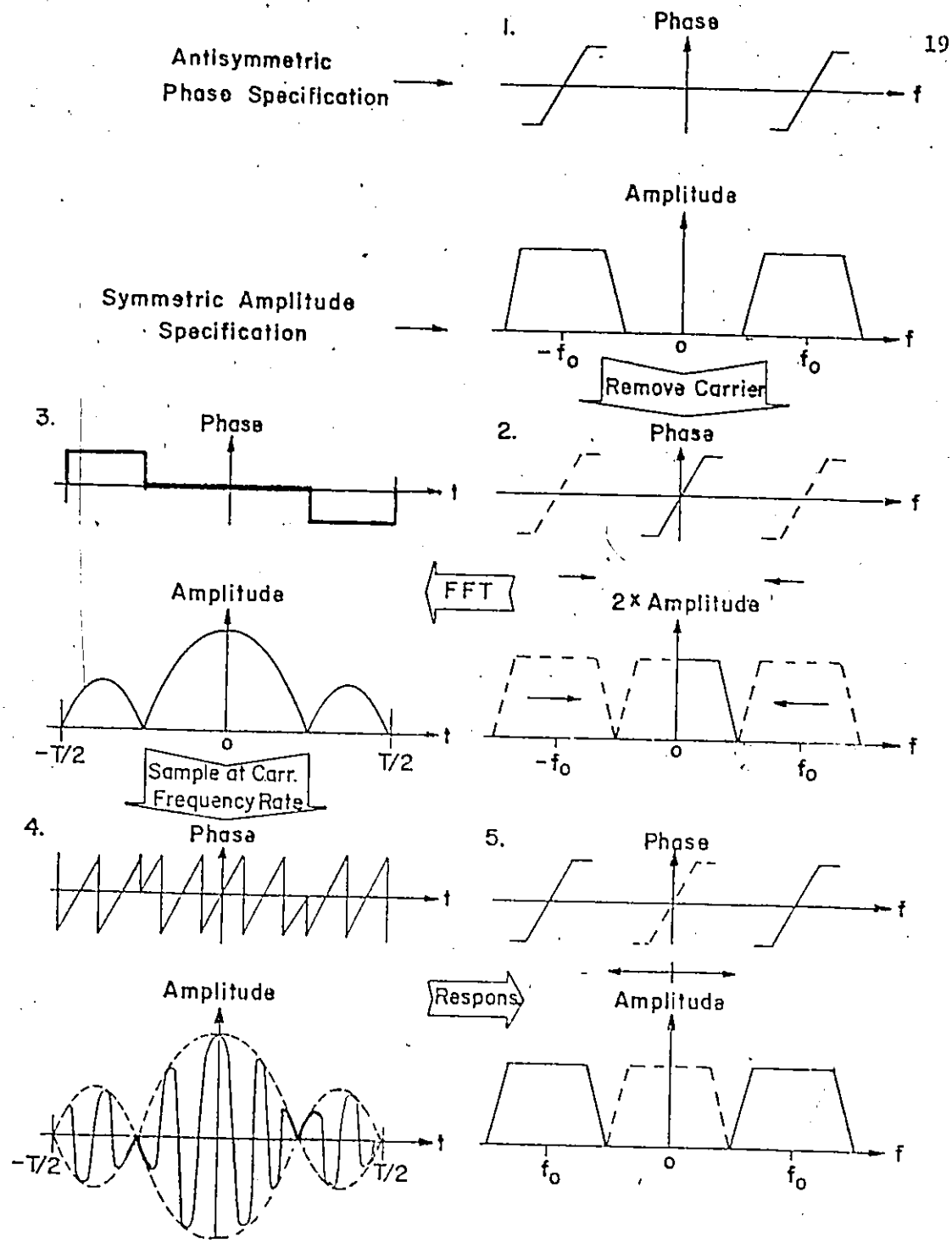


Figure 2.1: Lowpass FFT response when resonated at bandpass

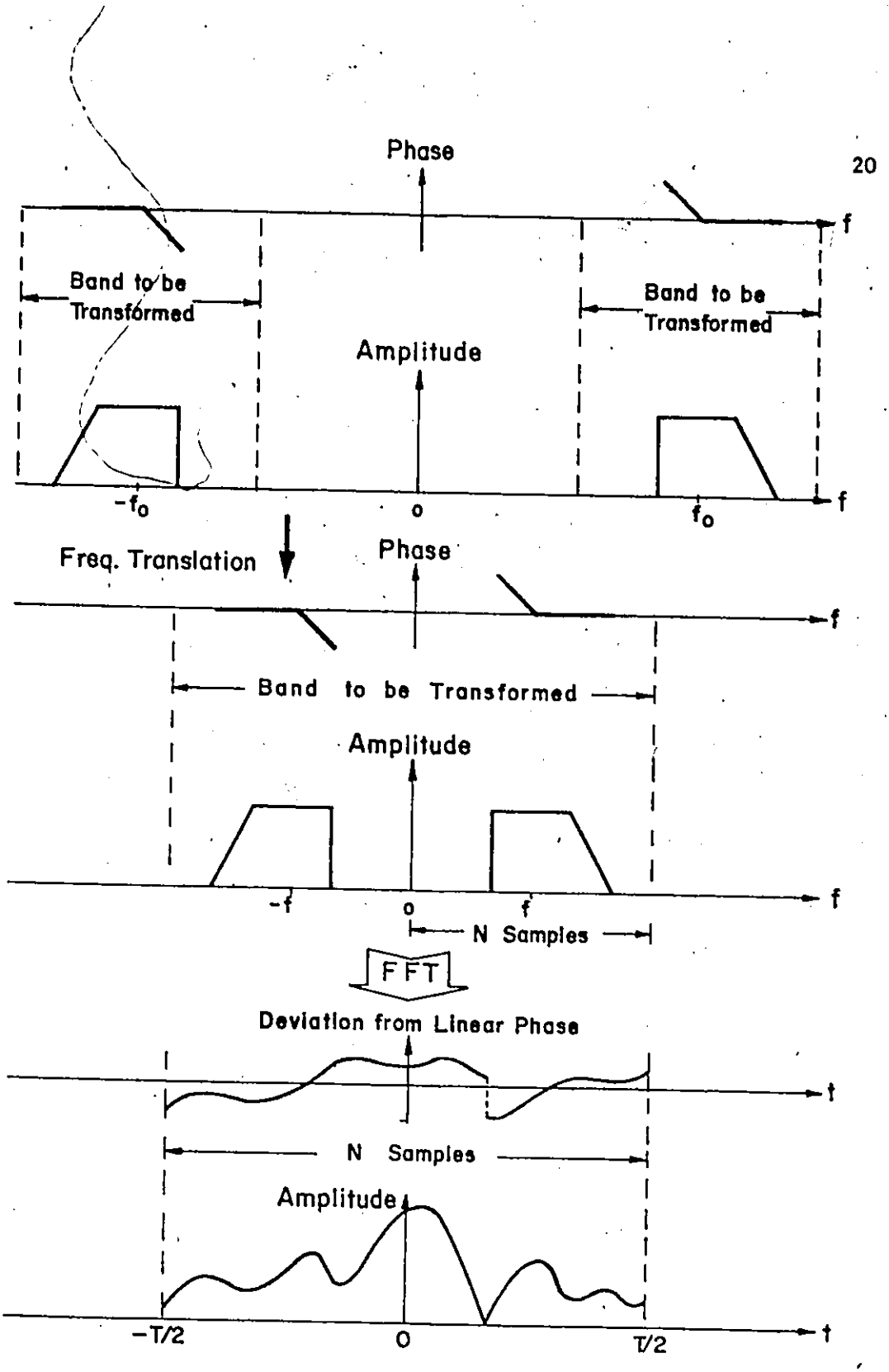


Figure 2.2: Bandpass FFT response when frequency translated

frequency of the bandpass). In reality, the physical nature of the surface acoustic wave interdigital transducer (SAW IDT) prevents this from happening at the lower frequencies while the upper harmonics may be suppressed, if necessary, by simple RC external filtering techniques. This is usually not necessary as the amplifiers will be chosen such that they will pass only up to, and including, the frequencies of interest.

Thus, it is feasible to restrict the region of the frequency domain to be transformed and therefore reduce the required number of samples. This is necessary if the computation is to be kept to a reasonable amount of time and storage.

It should be noted at this time that the length of the IDT is determined by: the number of samples in the transformation and the band of the frequency domain. From [17] it is stated that:

$$N \Delta t = T$$

$$\Delta f = T^{-1}$$

$$N \Delta f = RLBW \times FO$$

where: N is the number of samples in both the frequency and time domains

Δt is the time domain spacing

Δf is the frequency domain spacing

T is the period of the time domain being considered thus the IDT's length

RLBW is the relative frequency domain considered

FO is the centre of the frequency domain that is transformed.

It is easily seen that: $T = N/(RLBW \times FO)$. By choosing RLBW to include only the passband and adjacent stopbands of interest, the length of the IDT corresponding to T may be chosen by selecting the appropriate value of N. The length of the surface of the substrate covered by the IDT will be given by: $L = v\,T$

where: L is the length of the IDT

v is the surface acoustic wave velocity corrected for metallization

T is the period in time covered by the IDT.

2.2 Input of Fast Fourier Transform

In the programme written to illustrate this algorithm, the input DATA supplied by the designer is as presented below:

(a) IG is the power of 2 that specifies the number of sample points in the frequency and time domains which will be transformed. Typically IG = 6, 7, 8 such that there are N = 32, 64, 128 points.

(b) NP is the number of points in the frequency domain specified in terms of dB of insertion loss and degrees of phase deviation from linear phase.

(c) RLBW is the per unit bandwidth to be transformed. Typically this is from 0.5 FO to 2.0 FO.

(d) FO is the centre frequency of the region to be transformed and is usually chosen in the approximate centre of the desired bandpass. Units are MHz.

(e) next follow the NP specifications which are given as:

(i) the frequency in MHz

(ii) the insertion loss in dB

(iii) the phase deviation in degrees.

The specifications are entered from the lowest frequency through to the highest frequency. All specified frequencies must be contained in the region to be transformed and a linear interpolation between points is assumed.

2.3 Conditioning the DATA

Having obtained the raw DATA, the frequency domain samples must be generated in the form required by the FFT process. The region to be transformed is divided into $N = 2^{IG}$ frequencies between (1 - 1/2 RLBW)FO and (1 + 1/2 RLBW)FO. The points below the first specified frequency are given the same specifications of insertion loss and phase as the lowest supplied DATA point and the points above the highest specified frequency are given the highest frequency point's specifications. All of the points in between are given specifications obtained from a linear interpolation between the adjacent DATA points. Because of the discrete equal spacing of the specifications, it should now be obvious why the transformed region is minimized, otherwise an excessively large number of points would be required so as to accurately represent the desired continuous response by the sampled points.

Next, the specifications (which are stored on the system mass storage for future comparison) are transformed to the rectangular linear

co-ordinate system as a set of N complex numbers. Since FORTRAN IV is the system language, the complex numbers are legitimate input arguments.

2.4 Interpreting the FFT Output

After applying the FFT routine, the resulting set of numbers (usually complex) must be sorted because of the "bit-reversed" order of the output as is shown in [17]. These time domain samples are then transformed into the linear-polar coordinate system with the amplitude vector allowed to have both positive and negative values so as to eliminate phase discontinuities at the nulls of the amplitude response. At these points, the phase changes typically by 180° if the amplitude is restricted to positive values.

The phase of each one of the samples at this step in the process will be restricted to the range of -90° to $+90^{\circ}$. Therefore it is necessary to remove the discontinuous steps between samples. The result of this process is a continuously increasing phase response (with linear interpolation between sample points) from the first through the last sample point. The centre sample (N/2) is taken as t = 0 and is taken as the reference for the centre of the IDT to be developed from these samples.

It should be noted that since the bandpass being transformed is not at baseband, then the transformation will have a linear phase component as well as the phase deviation from linear phase which gives rise to the discontinuities mentioned above. This is analogous to the phase response of an intermediate frequency in communications theory and

is illustrated in Figure 2.3.

2.5 Dirac-Delta Model

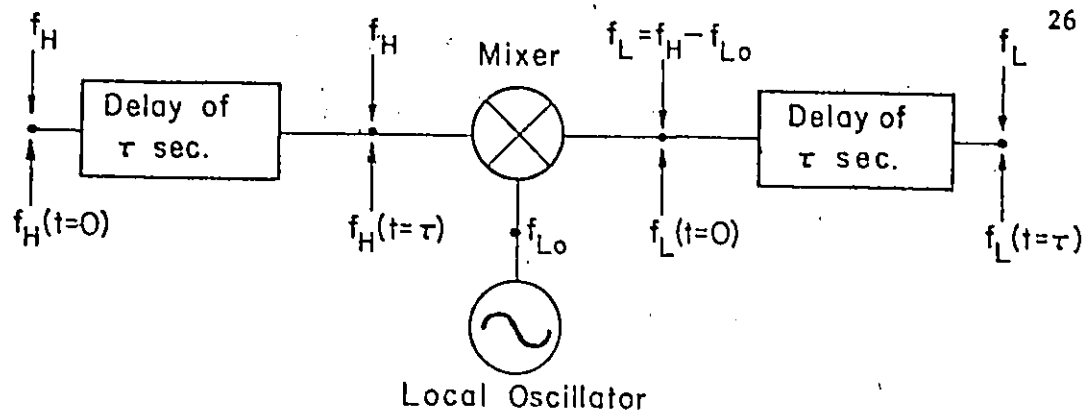
Having conditioned the time domain response as described above it is now possible to synthesize the IDT. In principle, the fingers of the transducer are placed at any point in the phase response that has a phase of 0^{o}, $\pm 180^{o}$, $\pm 360^{o}$, $\pm 540^{o}$ .... Those phases that are even multiples of 180^{o} will have a positive pad connection and those with odd multiples of 180^{o} will have a negative pad connection. The length of the finger overlap will be scaled according to the amplitude response at the appropriate phase position. The amplitude response is linearly interpolated between the evenly spaced samples of the Fast Fourier Transform solution.

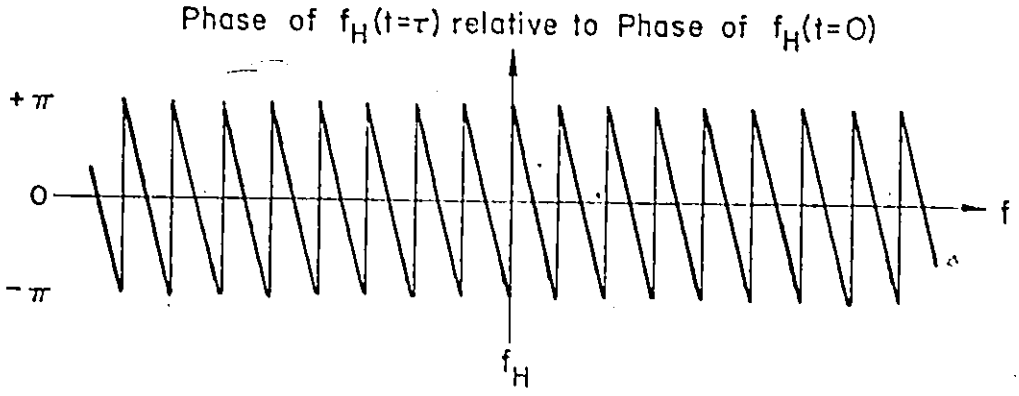
It should be noted also that the negative amplitudes will be interpreted as a change of the rail connection that was obtained from the phase criterion. While this process seems more cumbersome than restricting the amplitude response to positive values and working with the discontinuous phase response, it actually reduces the number of samples that would be required to accurately establish finger lengths and positions in the region of the discontinuities.

The above procedure is shown schematically in Figure 2.4.

2.6 Limitations of Dirac-Delta Model

At this point the raw DATA of finger length, pad connection (polarity of finger) and the position in the time domain has been





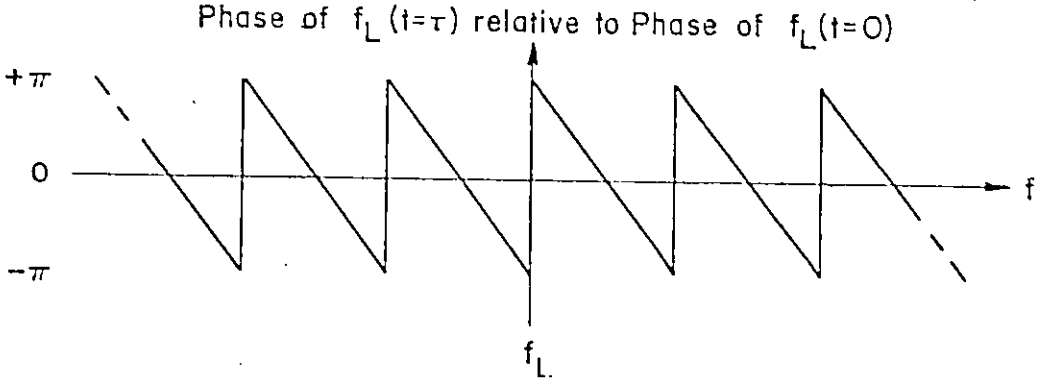


Figure 2.3: Analogy to IF phase response

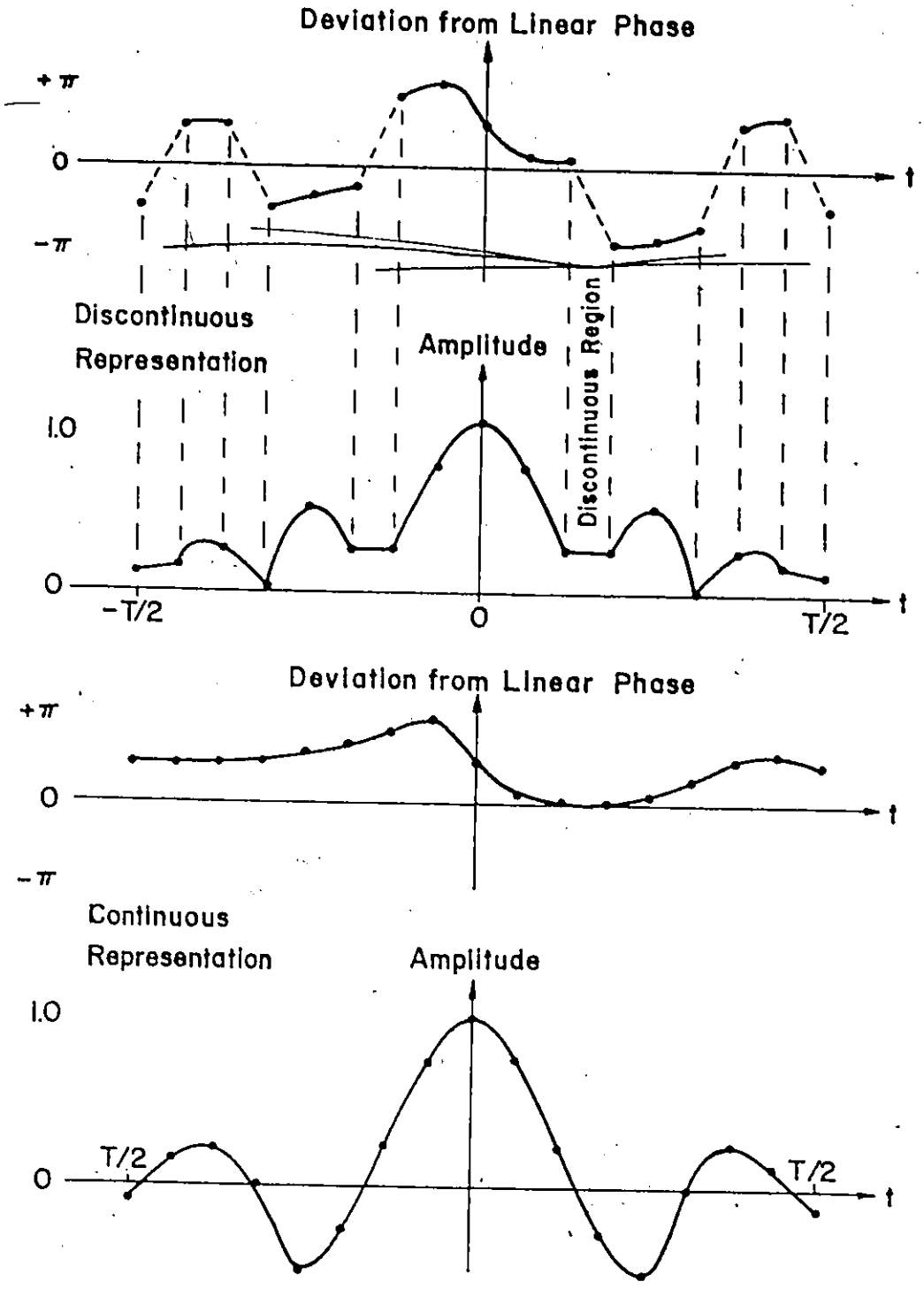


Figure 2.4: Effects of discontinuities in phase response of transform

obtained. In the past, most designers have simply translated this DATA to the surface spatial domain of the acoustic medium producing devices on this basis. Since that technique has produced experimental results with up to 15% deviation from the theoretical predictions, the next step in refining the design process is to use a more accurate analysis technique and from this, hopefully, to be able to design another transducer with updated input specifications such that the results of the next synthesis will be closer to the desired specifications. In the next chapter, a more accurate model will be described.

CHAPTER 3

CHARGE DISTRIBUTION MODEL OF AN INTERDIGITAL TRANSDUCER

3.1 Charge to Acoustic Deformation

It has been suggested [2] that the acoustic wave generated by a finger of an interdigital transducer (IDT) is related to the charge distribution on that finger by the following relationship:

$$a_s = \frac{k_s}{2\sqrt{\omega\ \epsilon_f / L}} \int_A j\ \omega\ q(x_o)\ e^{-jk_x(x-x_o)}\ dx_o \qquad (3.1)$$

where: a_s is the acoustic wave deformation at x

$k_s = \frac{2\Delta V}{V_\infty}$ is the surface wave coupling constant

ω is the frequency of excitation in radians/sec.

ϵ_f is the effective dielectric constant

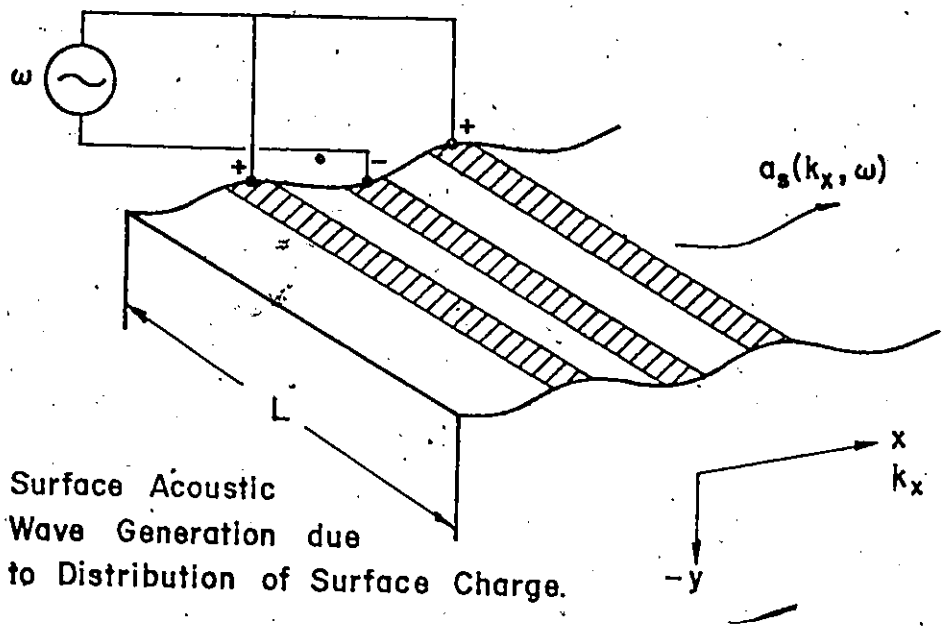
L is the length of the finger

$q(x_o)$ is the charge on the finger as a function of x_o

k_x is the SAW propagation constant in the x direction

x_o is defined as that region of $-b \leq x \leq b$ on which the finger is deposited.

These quantities are depicted in Figure 3.1. The quantities on the right hand side of (3.1) are all known except for the charge distribution function $q(x_o)$ and it is this quantity that needs to be



Surface Acoustic Wave Generation due to Distribution of Surface Charge.

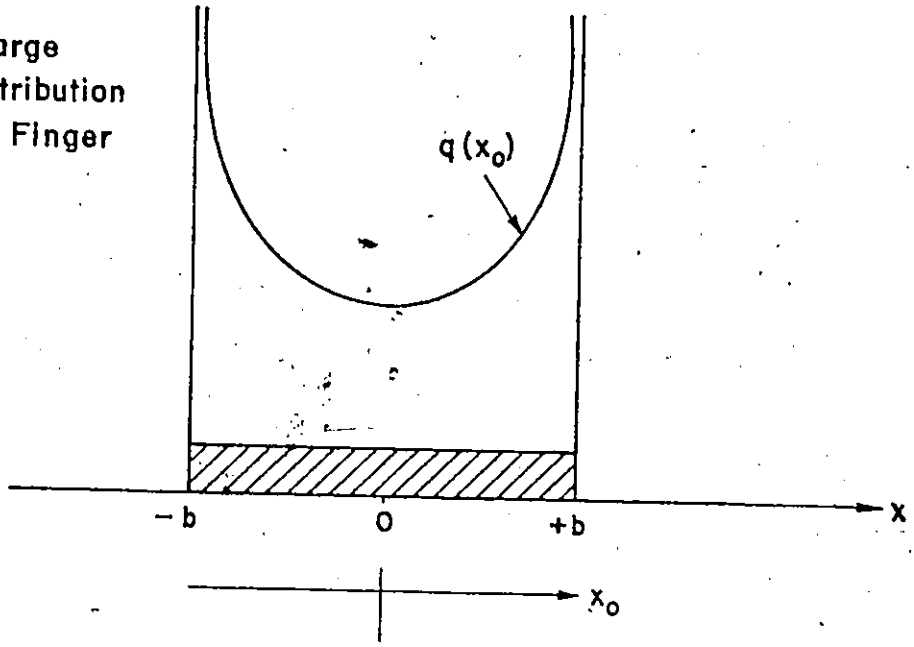


Figure 3.1: Acoustic wave generation of IDT

established.

Before solving for $q(x_o)$, it is necessary to point out some assumptions that are made with respect to SAW IDT's. Firstly, as has been stated in the introduction, the apodization (variable finger length) is taken to be equivalent to varying the applied voltage (and thus the charge since adjacent fingers form the plates of a capacitor) to each finger of a constant overlap IDT (i.e. all fingers have the same length). This approximation is quite accurate for longer (greater than 10 λ_s [18] where λ_s is the SAW length and finger centres have approximately a $\lambda_s/2$ separation) fingers near the centre of the transducer. Those fingers that are shorter or near the ends of the array require correction factors to account for mechanisms such as diffraction. Secondly, so as to make the solution tractable, the electrostatic problem is taken as a two dimensional situation ($y = q(x_o)$ as a function of x). That is, the charge distribution is assumed to be constant over the length of the finger to a first approximation. The end effects of the shorter overlaps will be compensated for empirically, by scaling at a later point.

3.2 Electrostatic Potential

The electrostatic potential, at some point x on the surface of the substrate, due to the charge $q(x_o)$ of a finger centred at $x = x_o = 0$ is given to a first approximation by [18] as:

$$\phi(x) = -\frac{1}{\pi \varepsilon_f} \int_A q(x_o) \ln |x-x_o| \, dx_o \qquad (3.2)$$

The X-Y plane is taken to be through the centre of the apodization of the IDT and it is assumed that the fingers are apparently of infinite length.

For any point x_p on the x-axis of the substrate, the voltage potential will be given by the summation of the electrostatic potentials of all the potentials for all the fingers in that IDT.

i.e. $$V(x_p) = -\frac{1}{\pi \epsilon_f} \sum_{\ell=1}^{L} \int_{A_\ell} q_\ell(x_o) \ln |x_\ell - x_o| \, dx_o \qquad (3.3)$$

where the ℓ subscript denotes the particular finger of the array whose charge effect is being evaluated. Also, x_ℓ is the distance from the point x_p on the substrate to the centre of the ℓ^{th} finger, thus the origin of x_ℓ is $x = x_o = 0$ for each finger and is redefined for each finger.

3.3 Charge on a Finger

The charge distribution on a single metal strip has been shown to be given by [19] as:

$$q(x_o) = \frac{C_o}{\sqrt{b^2 - x_o^2}} \qquad (3.4)$$

and it has been suggested [18] that the charge distribution of a finger in an IDT array can be modelled by:

$$q(x_o) = \frac{C_0 + C_1 x_o + C_2 x_o^2}{\sqrt{b^2 - x_o^2}} \qquad (3.5)$$

where: C_0, C_1 and C_2 are constant coefficients that are a function of the finger's position in the transducer and b is 1/2 the finger width. The linear term deviation, C_1x_o in (3.5), from the isolated strip distribution of equation (3.4) is caused by the fact that a finger nearer one end of an IDT than the other will have a skewed or non-symmetric charge distribution. The quadratic term corresponding to $C_2x_o^2$ is due to the proximity of other fingers on either side of the finger under consideration which tends to attract more charge to the edges than would be the case with an isolated strip. Thus, the problem has now been reduced to finding the coefficients, C_0, C_1 and C_2, for each finger of an IDT.

3.4 Finding the Charge Distribution Coefficients

Since all the fingers of the IDT are connected in parallel to either the positive or negative rails, it is known that the voltage boundary condition will be $\pm V/2$ at any point on the fingers. By selecting the outer edges and the centre points of each finger, then 3 equations may be written for each finger at which the electrostatic potential may be calculated. The electrostatic potential will be a function of the charge distribution on all of the fingers in the IDT array and these charges may be arranged such that the potential due to the charges will equal the boundary value of the applied voltage (again, it is pointed out that the voltage on each finger is taken as the relative length of that individual finger multiplied by the applied voltage and the finger is assumed to have a length equal to the longest

finger overlap). Thus for each finger, there will be the following 3 equations:

$$V_{ik} = -\frac{1}{\pi\,\epsilon_f}\sum_{\ell=1}^{L} C_{0\ell}\int_{-b_\ell}^{b_\ell}\frac{\ln|x_\ell - x_o|}{\sqrt{b_\ell^2 - x_o^2}}\,dx + C_{1\ell}\int_{-b_\ell}^{b_\ell}\frac{x_o \ln|x_\ell - x_o|}{\sqrt{b_\ell^2 - x_o^2}}\,dx_o + C_{2\ell}\int_{-b_\ell}^{b_\ell}\frac{x_o^2 \ln|x_\ell - x_o|}{\sqrt{b_\ell^2 - x_o^2}}\,dx \qquad (3.6a)$$

where: $i = 1, 2, 3$

$k = 1, 2, 3, \ldots, L$

putting:

$$\phi_{0\ell k} = -\frac{1}{\pi\,\epsilon_f}\int_{-b_\ell}^{+b_\ell}\frac{\ln|x_\ell - x_o|}{\sqrt{b_\ell^2 - x_p^2}}\,dx_o$$

$$\phi_{1\ell k} = -\frac{1}{\pi\,\epsilon_f}\int_{-b_\ell}^{b_\ell}\frac{x_o \ln|x_\ell - x_o|}{\sqrt{b_\ell^2 - x_o^2}}\,dx_o$$

$$\phi_{2\ell k} = -\frac{1}{\pi\,\epsilon_f}\int_{-b_\ell}^{+b_\ell}\frac{x_o^2 \ln|x_\ell - x_o|}{\sqrt{b_\ell^2 - x_o^2}}\,dx_o \qquad (3.6b)$$

Then a set of equations may be written:

$$\underset{\sim}{\phi}_{0i} \cdot \begin{bmatrix} C_{01} \\ C_{02} \\ \vdots \\ C_{0L} \end{bmatrix} + \underset{\sim}{\phi}_{1i} \cdot \begin{bmatrix} C_{11} \\ C_{12} \\ \vdots \\ C_{1L} \end{bmatrix} + \underset{\sim}{\phi}_{2i} \cdot \begin{bmatrix} C_{21} \\ C_{22} \\ \vdots \\ C_{2L} \end{bmatrix} = \begin{bmatrix} V_1 \\ V_2 \\ \vdots \\ V_L \end{bmatrix} \tag{3.7}$$

where: ϕ_0, ϕ_1 and ϕ_2 are L x L matrices and i = 1, 2, 3 corresponds to the 3 points on each finger i.e. the two edges and the centre where the potentials are calculated and then equated to the boundary values.

The authors of [8] wrote this set of equations as:

$$\underset{\sim}{\phi} \cdot \begin{bmatrix} C_{01} \\ C_{11} \\ C_{21} \\ \vdots \\ C_{0L} \\ C_{1L} \\ C_{2L} \end{bmatrix} = \begin{bmatrix} V_{11} \\ V_{21} \\ V_{31} \\ \vdots \\ V_{1L} \\ V_{2L} \\ V_{3L} \end{bmatrix} \tag{3.8}$$

where $\underset{\sim}{\phi}$ is a 3L x 3L matrix.

Then, they inverted $\underset{\sim}{\phi}$ to obtain the coefficients of the charge distribution matrix. It is obvious that this technique is restricted to relatively small problems (they illustrated the process with a 13 finger transducer), thus large realistic IDT's must be solved in a different manner.

In (3.8) it is noted that $V_{1\ell} = V_{2\ell} = V_{3\ell}$ for each finger, thus the set of equations may be arranged as in (3.7) and the left hand side may be evaluated for initial guesses of $C_{0\ell}$, $C_{1\ell}$ and $C_{2\ell}$ as:

$$\underset{\sim}{\Phi}_{0i} \cdot \begin{bmatrix} C_{01} \\ C_{02} \\ \vdots \\ C_{0L} \end{bmatrix} + \underset{\sim}{\Phi}_{1i} \cdot \begin{bmatrix} C_{11} \\ C_{12} \\ \vdots \\ C_{1L} \end{bmatrix} + \underset{\sim}{\Phi}_{2i} \cdot \begin{bmatrix} C_{21} \\ C_{22} \\ \vdots \\ C_{2L} \end{bmatrix} = \begin{bmatrix} P_{1i} \\ P_{2i} \\ \vdots \\ P_{Li} \end{bmatrix} \qquad (3.9)$$

where i = 1, 2, 3 which corresponds to the 3 points on each finger under consideration.

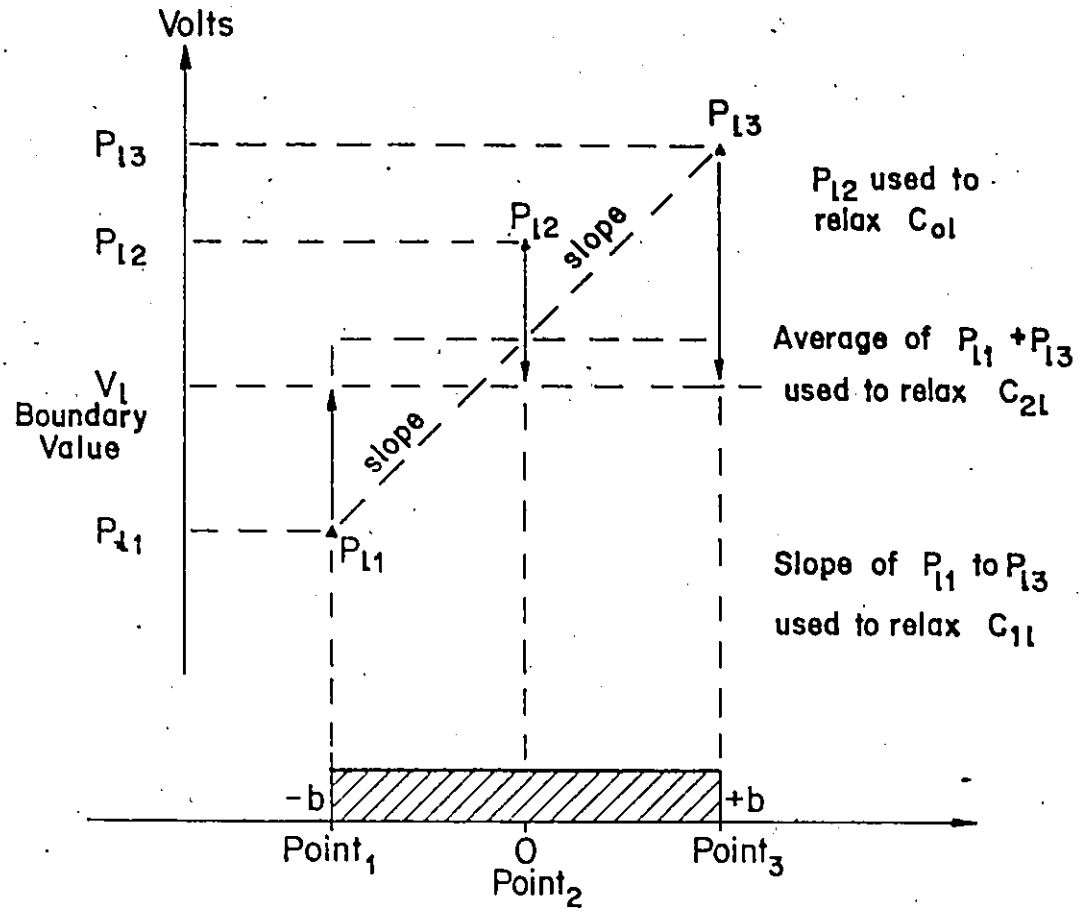
By comparing the calculated $P_{\ell i}$ to the imposed boundary value V_ℓ, it is possible to relax the coefficients $C_{0\ell}$, $C_{1\ell}$ and $C_{2\ell}$ in such a way that $P_{\ell 1}$, $P_{\ell 2}$ and $P_{\ell 3}$ are made to be equal to V_ℓ within an acceptable error criterion. Once this has been accomplished, then applying the optimum values of $C_{0\ell}$, $C_{1\ell}$ and $C_{2\ell}$ to (3.5) will give the charge distribution for each finger of the transducer.

The initial guess for $C_{0\ell}$ is taken proportional to the corresponding fingers amplitude from the FFT solution while $C_{1\ell}$ and $C_{2\ell}$ are set to zero. Then the potentials $P_{\ell 1}$, $P_{\ell 2}$ and $P_{\ell 3}$ are calculated and the relaxation process is begun. Referring to Figure 3.2, the relaxation of $C_{0\ell}$ is taken as:

$$C_{0\ell}{}_{t+1} = C_{0\ell}{}_{t} [1 + B_o (V_\ell - P_{\ell 2})] \qquad (3.10)$$

where B_o is a constant that includes scaling and an over/under relaxation constant. The t+1 subscript implies a newer value than the t subscript.

The initial value of $C_{1\ell}$ is taken as:



The values of C_{ol}, C_{1l} and C_{2l} are relaxed in such a way so as to make the potentials P_{11} , P_{12} and P_{13}, calculated from the charge distribution, approach the boundary value V_l

Figure 3.2: Relaxation of charge distribution coefficients

$$C_{1\ell} = \frac{P_{\ell 3} - P_{\ell 1}}{k_1} \tag{3.11}$$

where k_1 is a relational constant. In the following passes $C_{1\ell}$ is relaxed as:

$$C_{1\ell}{}_{t+1} = C_{1\ell}{}_{t} [1 + B_1 (P_{\ell 3} - P_{\ell 1})] \tag{3.12}$$

where B_1 is a constant including the under-relaxation constant.

The initial value of $C_{2\ell}$ is taken as:

$$C_{2\ell} = \frac{V_\ell - 1/2 (P_{\ell_3} + P_{\ell_1})}{K_2} \tag{3.13}$$

where K_2 is the equations's relational constant. There after C_{2n} is relaxed as:

$$C_{2\ell}{}_{t+1} = C_{2\ell}{}_{t} [1 + B_2 (V_\ell - 1/2 (P_{\ell 3} + P_{\ell 1}))] \tag{3.14}$$

where B_2 is the constant as before. No proof is offered in this thesis as to why the numerical technique causes the potentials to relax into their boundary values. The routine has shown itself to cause $P_{\ell 1}$, $P_{\ell 2}$ and $P_{\ell 3}$ to tend toward V_ℓ for successive passes and produces deviations from V_ℓ of less than 1% in 3 passes for those cases that have been investigated.

The only quantities to be established now are the integral functions ϕ_{0i}, ϕ_{1i} and ϕ_{2i}. While the solutions to these functions must have been obtained by the authors of [18], neither the development nor the functions themselves have been presented in the literature.

3.5 The Integral Functions

Considering the equations (3.6b) and putting:

$$I_{10} = -\int_{-b}^{b} \frac{\ln |x-x_o|}{\sqrt{b^2 - x_o^2}}\, dx_o$$

$$I_{20} = -\int_{-b}^{b} \frac{x_o \ln |x-x_o|}{\sqrt{b^2 - x_o^2}}\, dx_o$$

$$I_{30} = -\int_{-b}^{b} \frac{x_o^2 \ln |x-x_o|}{\sqrt{b^2 - x_o^2}}\, dx_o \qquad (3.16)$$

where the subscript & has been dropped. Then it may be shown as is given in Appendix I that the solution to these integrals is:

$$I_{10} = \pi \ln (2) \qquad \text{for } x = 0$$

$$I_{10} = \pi \ln \frac{2}{1 + \sqrt{1 - (b/x)^2}} \qquad \text{for } |x| \geq +b$$

$$I_{20} = 0 \qquad \text{for } -b \leq x \leq +b$$

$$I_{20} = -\pi \left(\sqrt{1 - (b/x)^2} - 1\right) \qquad \text{for } x > +b$$

$$I_{20} = +\pi \left(\sqrt{1 - (b/x)^2} - 1\right) \qquad \text{for } x < -b$$

$$I_{30} = \frac{\pi}{2} \left[\ln (2) - \frac{1}{2}\right] \qquad \text{for } x = 0$$

$$I_{30} = \frac{\pi}{2} \left\{\left(\frac{b}{x}\right)^2 \ln \left[\frac{2}{1+\sqrt{1-(b/x)^2}}\right] + \sqrt{1-(b/x)^2} - 1 - \frac{1}{2}\left(\frac{b}{x}\right)^2\right\} \qquad \text{for } |x| \geq b$$

(3.17)

These functions are shown graphically in Figure 3.3. The results have been normalized such that the arguments of the logarithms are dimensionless and such that the integrals disappear as $x \to \infty$. This at first may seen strange since x is a variable; however, it is noted that the resulting variable is b/x and this is a variable of x.

3.6 Summary

Thus, by applying the results of the integral equations (3.17) to (3.3b) for the appropriate conditions of x_ℓ, the potentials at each of the 3 point's on each finger may be evaluated using (3.9) with the values of $C_{0\ell}$, $C_{1\ell}$ and $C_{2\ell}$ being found as described in Section 3.4. Once this has been done, then an acceptable solution to the charge distribution on the IDT has been found (through (3.5)) and the results may now be applied to (3.1) so as to calculate the acoustic far field radiation and thus obtain the frequency response of the IDT. This will be described in the next chapter.

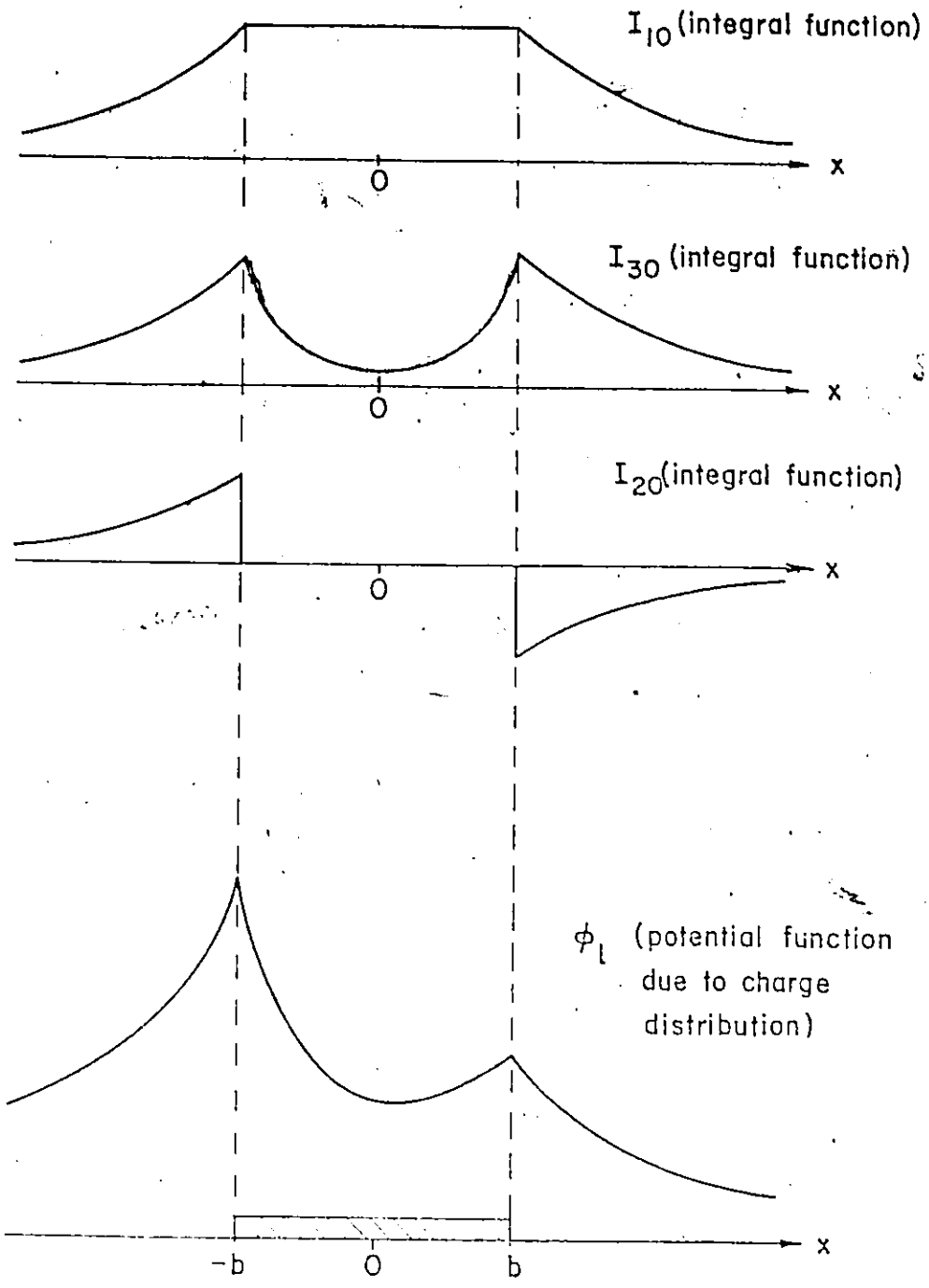


Figure 3.3: Potential functions due to charge distribution

CHAPTER 4

FREQUENCY RESPONSE OF AN INTERDIGITAL TRANSDUCER

4.1 General

Recalling equation (3.1)

$$a_s = \frac{k_s}{2\sqrt{\omega\,\varepsilon_f/L}} \int_A j\omega\, q(x_o)\, e^{-jk_x(x-x_o)}\, dx_o \qquad (4.1)$$

where the above quantities have been previously defined, it is now possible to calculate the acoustic deformation at some point x_p. By using the superposition of the waves generated by each finger in the transducer, and applying the appropriate solutions to $q(x_o)$ as established in Chapter 3. Then at some point x_p the response will be given by:

$$H(\omega) = \sum_{\ell=1}^{L} a_{s\ell}$$

$$= \frac{j\omega\, k_s}{2\sqrt{\omega\,\varepsilon_f/L}} \sum_{\ell=1}^{L} e^{-jk_x x_\ell} \int_{-b_\ell}^{+b_\ell} q_\ell(x_o)\, e^{+jk_x x_o}\, dx_o$$

$$= K \sum_{\ell=1}^{L} e^{-jk_x x_\ell}\, C_{o\ell} \int_{-b_\ell}^{b_\ell} \frac{e^{jk_x x_o}}{\sqrt{b_\ell^2 - x_o^2}}\, dx_o$$

$$+ C_{1\ell} \int_{-b_\ell}^{b_\ell} \frac{x_o \, e^{jk_x x_o}}{\sqrt{b_\ell^2 - x_o^2}} \, dx_o$$

$$+ C_{2\ell} \int_{-b_\ell}^{b_\ell} \frac{x^2 \, e^{jk_x x_o}}{\sqrt{b_\ell^2 - x_o^2}} \, dx_o \Bigg] \qquad (4.2)$$

where all of the constants have been grouped into K. Thus the 3 integrals of (4.2) need to be evaluated.

4.2 Integral Solutions

From equation (4.2) the integrals that need to be evaluated are:

$$I_{10} = \int_{-b}^{b} \frac{e^{j k_x x_o}}{\sqrt{b^2 - x_o^2}} \, dx_o$$

$$I_{20} = \int_{-b}^{b} \frac{x_o \, e^{j k_x x_o}}{\sqrt{b^2 - x_o^2}} \, dx_o$$

$$I_{30} = \int_{-b}^{b} \frac{x_o^2 \, e^{j k_x x_o}}{\sqrt{b^2 - x_o^2}} \, dx_o \qquad (4.3)$$

The solution to these integrals is given in Appendix II and the results are:

$$I_{10} = J_o\,(k_x\, b)$$

$$I_{20} = j\, b\, J_1\,(k_x\, b)$$

$$I_{30} = b^2\,[J_0\,(k_x\, b) - (k_x\, b)^{-1}\, J_1\,(k_x\, b)] \qquad (4.4)$$

where: J_0 and J_1 are Bessel functions of the 1st and 2nd kind

k_x is the propagation constant

b is 1/2 the finger width.

As before in Chapter 3, the normalization has been done with respect to x.

4.3 The Acoustic Wave

In general, the frequency response of an IDT is evaluated at the centre of the transducer (which corresponds to t=0 and x=0). However, since this algorithm deals with responses that are not the same in both directions, it is necessary to evaluate the response outside of the IDT in the direction in which the wave is desired to propagate. Substituting the solutions (4.4) into equation (4.1) gives:

$$a_s = \frac{j\, k_s e^{-jk_x x}}{2\sqrt{\omega\, \varepsilon_f / L}} \left[J_0(k_x\, b),\ jbJ_1(k_x\, b),\ b^2 \left\langle J_0(k_x\, b) - \frac{J_1(k_x\, b)}{k_x\, b} \right\rangle\right] \cdot \begin{bmatrix} C_0 \\ C_1 \\ C_2 \end{bmatrix}$$

$$= \frac{j\, k_s \sqrt{\omega L}\, e^{-jk_x x}}{2\sqrt{\varepsilon_f}} \left[J_0(k_x\, b),\ jbJ_1(k_x\, b),\ b^2 \left\langle J_0(k_x\, b) - \frac{J_1(k_x\, b)}{k_x\, b} \right\rangle\right] \begin{bmatrix} C_0 \\ C_1 \\ C_2 \end{bmatrix} \qquad (4.5)$$

Now by applying these results to (4.2) and doing the summation over the whole IDT under consideration, then the frequency domain response may be obtained.

4.4 Summary

Thus, the frequency response of the IDT will be given by

$$H(\omega) = K \cdot \sum_{\ell=1}^{L} e^{-j k_x x_\ell} \left[j \sqrt{\omega}\, C_{0\ell}\, J_0 (k_x b_\ell) \right.$$

$$\left. - \sqrt{\omega}\, b_\ell\, C_1\, J_1(k_x b_\ell) + j\sqrt{\omega}\, C_{i\ell}\, b_\ell^2 \left\langle J_0(k_x b_\ell) - \frac{J_1(k_x b_\ell)}{k_x b_\ell} \right\rangle \right] \qquad (4.6)$$

Then $H(\omega)$ is evaluated at each frequency ω_i at which the specifications have been established. By comparing the specifications to the responses at these specific frequencies, an iterative design technique may be developed so as to make the response approach the specifications. This iterative scheme is described in Chapter 5.

CHAPTER 5

NUMERICAL OPTIMIZATION

5.1 General

The sythesis procedures available to designers at this time all involve delta functions in the time domain from which the finger geometry is established. Since delta functions imply filamentary line sources (or taps in the receiving mode), this delta model does not adequately describe the physical response of the interdigital transducers (IDT). Also, applying a window function causes distortion of the desired bandpass by broadening the bandpass and rounding the edges. It is mainly because of these reasons that an iterative design technique is required if the desired response is to be met more accurately.

5.2 Error Function

Referring to Figure 5.1 an error function is calculated by substracting the calculated response of the synthesised IDT from the original specifications. The amplitudes are left on the logarithmic scale so that the error function is not "hard limited" at zero amplitude. In Figure 5.2 it is seen that if the error function (times its over/under relaxation constant) is greater than the original specification and is added linearly to the specifications then negative amplitude specifications result. These would have to be interpreted as phase reversals of 180° to the specifications which would cause

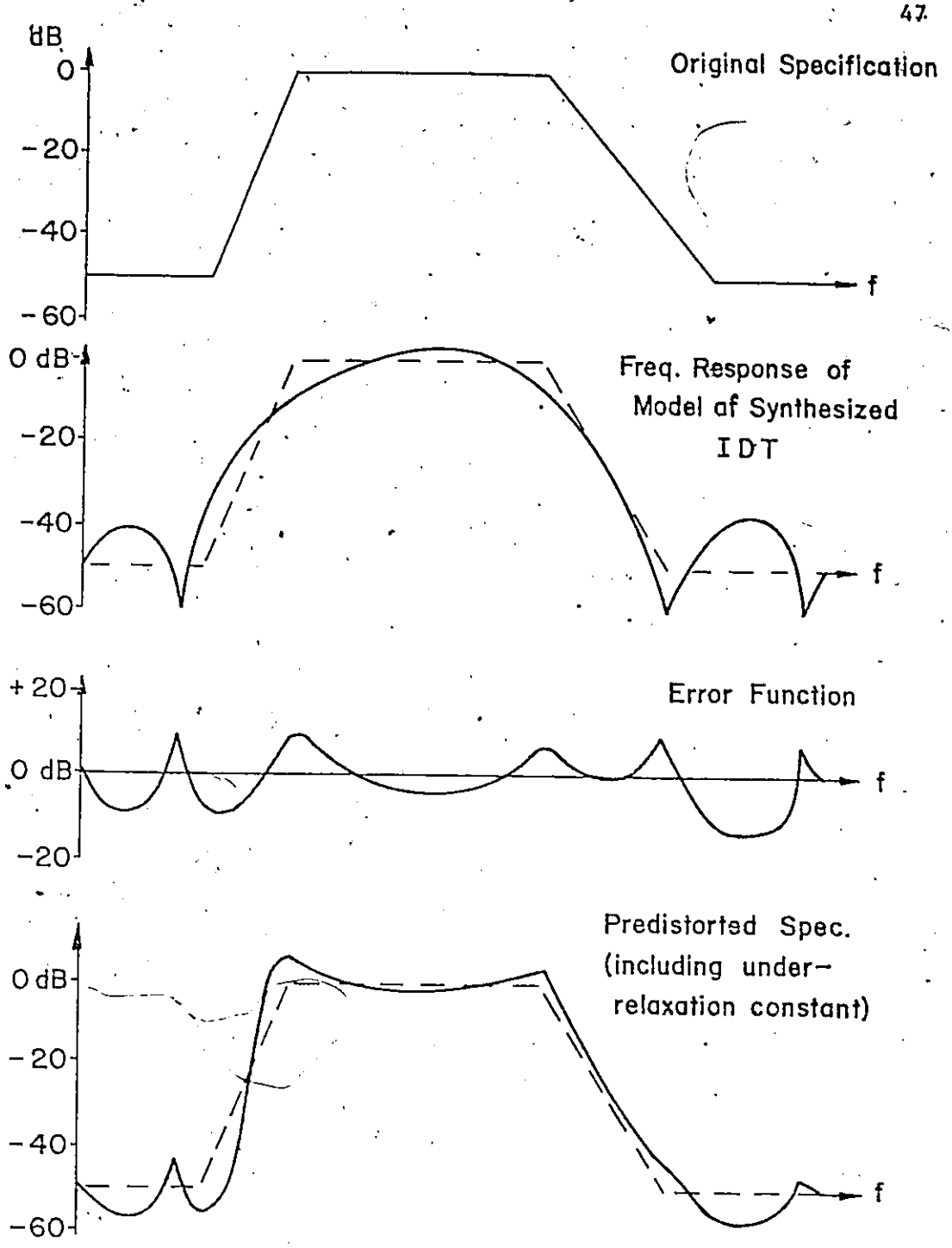
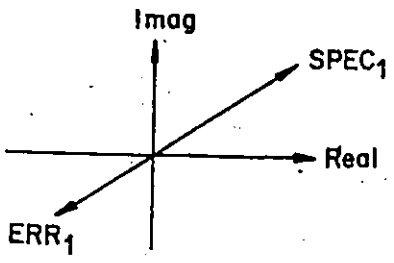


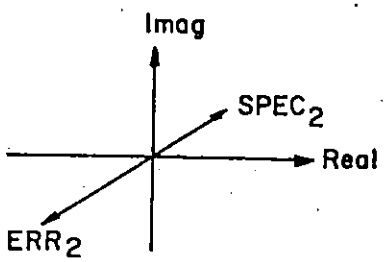
Figure 5.1: Development of error function and predistortion of specifications

Linear System: Predistortion is done on linear scale

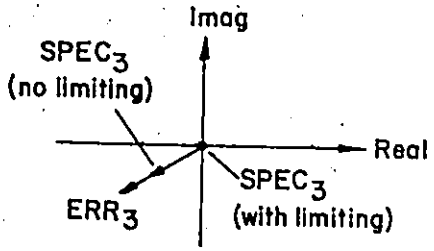
1.



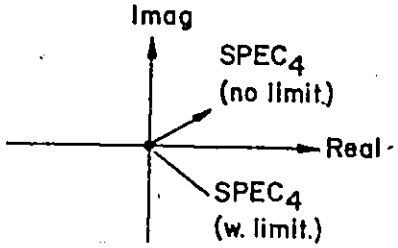
2.



3.

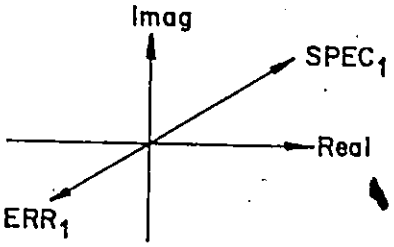


4.

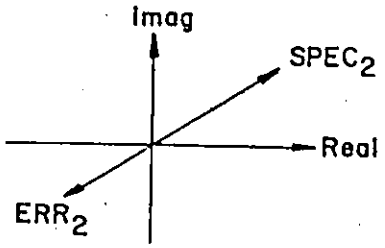


Logarithmic System: Predistortion is done on logarithmic scale

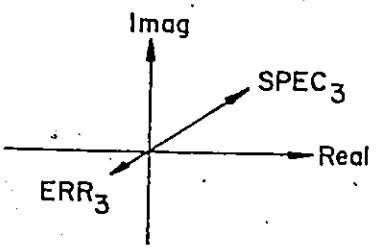
1.



2.



3.



4.

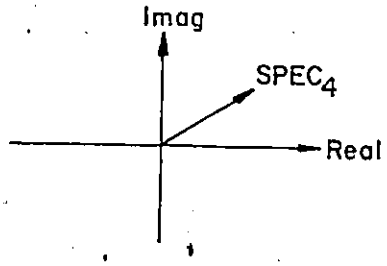


Figure 5.2: Hard limiting and soft limiting of error function

oscillations in the numerical process. By creating the error function on the logarithmic scale these problems are eliminated. The phase error function creates no such problem and is thus left in whatever angular linear measure is used in the routine. Thus, the error functions are given as:

$$\text{RMERR}(\omega_i) = \text{MDB}(\omega_i) - \text{RMDB}(\omega_i)$$

$$\text{RPERR}(\omega_i) = \text{PHASE}(\omega_i) = \text{RPHS}(\omega_i) \qquad (5.1)$$

where: $\text{RMERR}(\omega_i)$ is the amplitude error in dB

$\text{MDB}(\omega_i)$ is the desired response in dB

$\text{RMDB}(\omega_i)$ is the IDT response in dB

$\text{RPERR}(\omega_i)$ is the phase error in degrees

$\text{PHASE}(\omega_i)$ is the desired phase in degrees

$\text{RPHS}(\omega_i)$ is the phase response in degrees

The error function is calculated at each ω_i of the input set of specifications.

5.3 Predistortion of Specifications

Once the error functions have been calculated, then the error, multiplied by an under-relaxation factor, is added to the specifications that are fed into the Fast Fourier Transform (FFT) synthesis process as described earlier. An under-relaxation factor of 0.5 has proved to be the most efficient since higher values (up to 2.0) caused oscillations in the resulting numerical solutions. Thus, the specifications that are fed to the synthesis routine are:

$$MAG(\omega_i)_{t+1} = MAG(\omega_i)_t + 0.50 \times RMERR(\omega_i)_t$$

$$PHSE(\omega_i)_{t+1} = PHSE(\omega_i)_t + 0.50 \times RPERR(\omega_i)_t \qquad (5.2)$$

where: $MAG(\omega_i)$ is the variable amplitude input specifications to the FFT in dB

$PHSE(\omega_i)$ is the variable phase input specifications to the FFT in degrees.

The subscript t+1 implies the set of specifications that follow the pass where the t specifications were used. This process of predistortion is shown in Figure 5.1.

5.4 Summary

The iterative design process has been presented and because of the complexity of the problem, a totally automatic design algorithm is not suggested. It has proved expedient to build the system as an interactive computer-aided-design (CAD) technique which allows the designer to view the response of each synthesis. From these plots and numerical results, the designer is able to decide if the results are acceptable or if the procedure should be restarted with different design constraints.

It should also be noted that there is no need to optimize the phase criterion for amplitudes of less than -20 dB for example. Normally the phase response is only of interest in the bandpass and not in the stopband.

CHAPTER 6

PROGRAMMING THE ALGORITHM

6.1 General

To illustrate the design algorithm described in this thesis, a series of programmes have been written in FORTRAN IV for the CDC-6400 computing system at McMaster University, Hamilton, Canada. The programmes have been written for "batch" submission but could be used in an on-line interactive mode. Once developed to the point of having the design system on-line, then a further breaking up of the programmes so that routines such as analysis could be accessed at any time would be advantagous to the designer.

6.2 SAWINIT

Referring to the flow chart in Figure 6.1 the input DATA required by SAWINIT has been described in detail in Chapter 2. Recalling Section 2.2, block 1 of Figure 6.1 requires: the power of 2 (IG) that determines the number of samples (N) in the FFT; NP which is the number of specifications; the number of samples in the FFT; NP which is the number of specifications; the relative bandwidth of the transformation (RLBW); the centre frequency (FO) and then follow the NPP specifications of FRSP (NP,3).

Next in block 2 of Figure 6.1, the frequency domain from f = (1-RLBW) x FO to f = (1 + RLBW) x FO is divided into N = 2^{IG} frequency

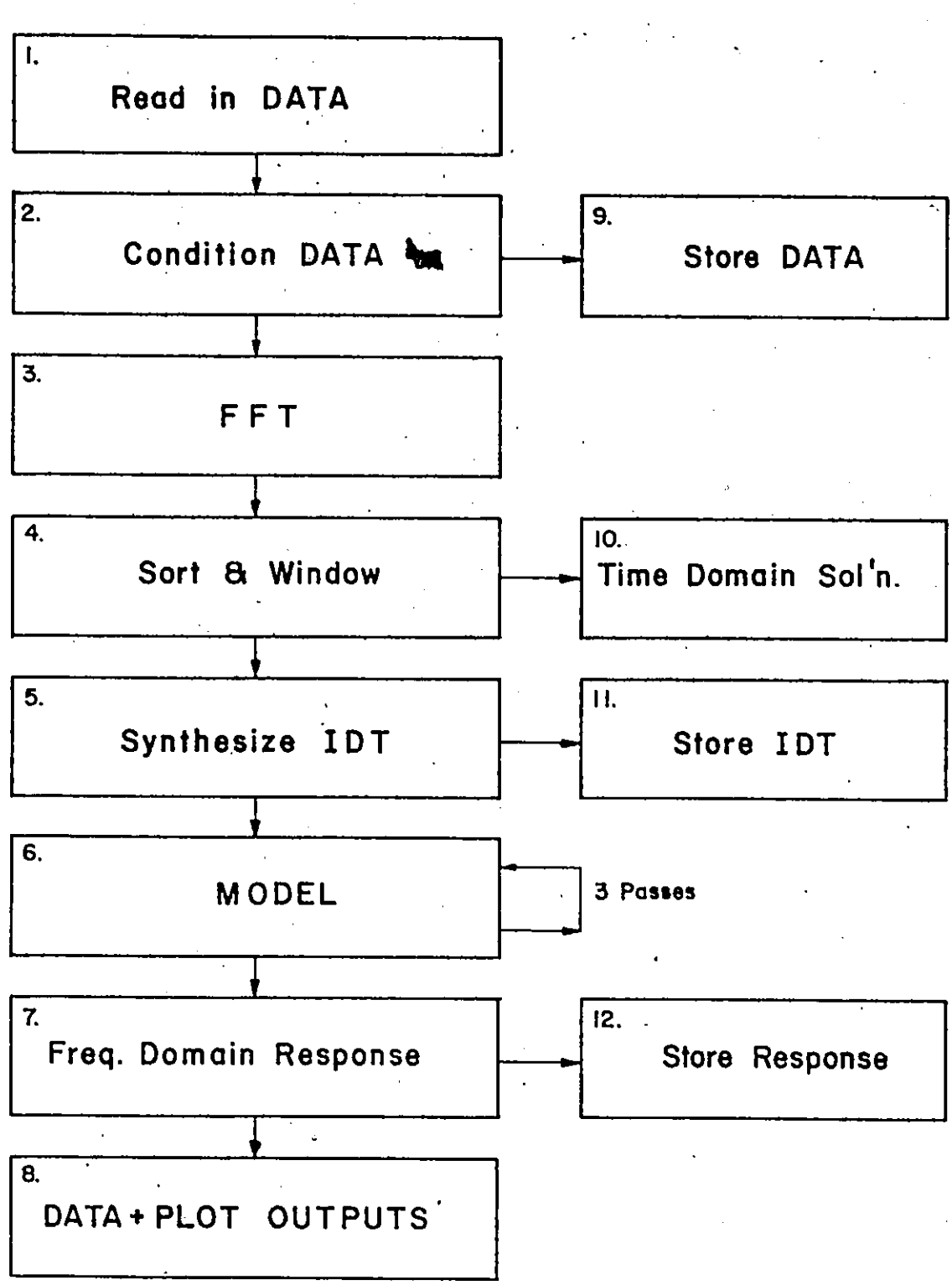


Figure 6.1: Flowchart of programme SAWINIT

points coded as FREQ(I). At each of these frequencies a desired specification MDB(I) for amplitude and PHASE(I) for phase is generated as a linearly interpolated fit to the DATA of FRSP. Those frequency sample points below FRSP(1,1) are given the specifications of FRSP(1,1) and those above FRSP(NP,1) takes its specifications.

For the first pass the input specifications to the FFT (block 3) are given the values of MAG(I) = MDB(I) and PHSE(I) = PHASE(I) which are then transformed into linear rectangular complex numbers for the transformation.

The output of the Fast Fourier Transform (FFT), is then sorted (block 4) so that the samples are in their correct order and so that t = 0 sec. is in the centre of the time domain solution i.e. the time domain solution goes from - T/2 to + T/2 and thus has its maximum amplitude near the centre of the domain.

Next a Kaiser-Bessel window function of:

$$WT(I) = \frac{J_0\left[(16/N)\left(\sqrt{NH^2 - |NH-I|^2}\right)\right]}{J_0\left[8.0\right]}$$

is applied to the time domain samples [20] where:

J_0 is the zero order Bessel function

N is the number of samples in the transform

NH = N/2

I is the sample from 1 to N

This is done to reduce the side lobes at the expense of broadening the bandpass and rounding the edges of the bandpass. From Step 4 the finite time domain solutions to the frequency domain specifications are plotted

in Step 10.

Having obtained the time domain solutions, the next step is to synthesize the IDT geometry (Step 5). After removing the phase discontinuities as explained in Chapter 3, the linear phase component due to the centre frequency is added to the phase response of the time domain solution as:

$$\phi(I) = 360.0^{\circ} \times FO \times DT \times I$$

where: FO is the centre resonant frequency in MHz

DT is the sample separation in micro-sec.

I is the sample number (I = 1, 2, ..., N)

Having changed the time domain samples as above then the finger lengths, positions and polarities are evaluated as described in Chapter 2 giving:

L is the number of fingers in IDT

LEN(I) is length of finger and sign gives polarity

TIM(I) is the time position in transducer structure relative to its physical position.

The basic IDT information of L, LEN and TIM are stored on the system mass storage (Step 11) so that the actual IDT geometry may be generated by the programme TAPEJOB if the designer wishes.

The next step of the charge-distribution modelling for the sythesized IDT is performed in Step 6. This procedure is described in detail in Chapter 3. The subroutine MODEL also calculates the acoustic response of each finger of the IDT from the charge distribution coefficients as described in (4.3) but without the phase delay $e^{-jk_x x_\ell}$.

This information is presented as complex numbers in the array AR(L) which is fed to the next routine FRQRSP (Step 7). FRQRSP calculates the actual frequency domain response of the transducer in the forward direction. This response (RMDB and RPHS) is stored on the system mass storage (Step 12) for use in SAWOPT. The final step of SAWINIT is to generate: the input specifications and the response in both plotted and numerical form. Also, the time domain solution is plotted. From this output, the designer can decide if the input specifications are valid or if the response meets his requirements. If further refinement of the design is wished then the programme SAWOPT is run as is described in the next section. The listing of SAWINIT is given in Appendix III.

6.3 SAWOPT

The programme SAWOPT (Appendix IV) is the same as the programme SAWINIT except for Steps 1 and 2. In Step 1, the desired specifications MDB and PHASE are obtained along with the predistorted specifications, MAG and PHSE, from the last pass of either SAWOPT or SAWINIT which ever preceded this run. The responses of the last synthesis, RMDB and RPHS, are compared to MDB and PHASE and then MAG and PHSE are distorted further as is described in Chapter 5. Then the whole process as in SAWINIT is repeated. This is shown in the flowchart of Figure 6.2.

The designer uses SAWOPT in successive passes in an attempt to minimize the error functions, RMERR and RPERR (Chapter 5), until a usable design is obtained. When RMDB and RPHS are acceptably small, the

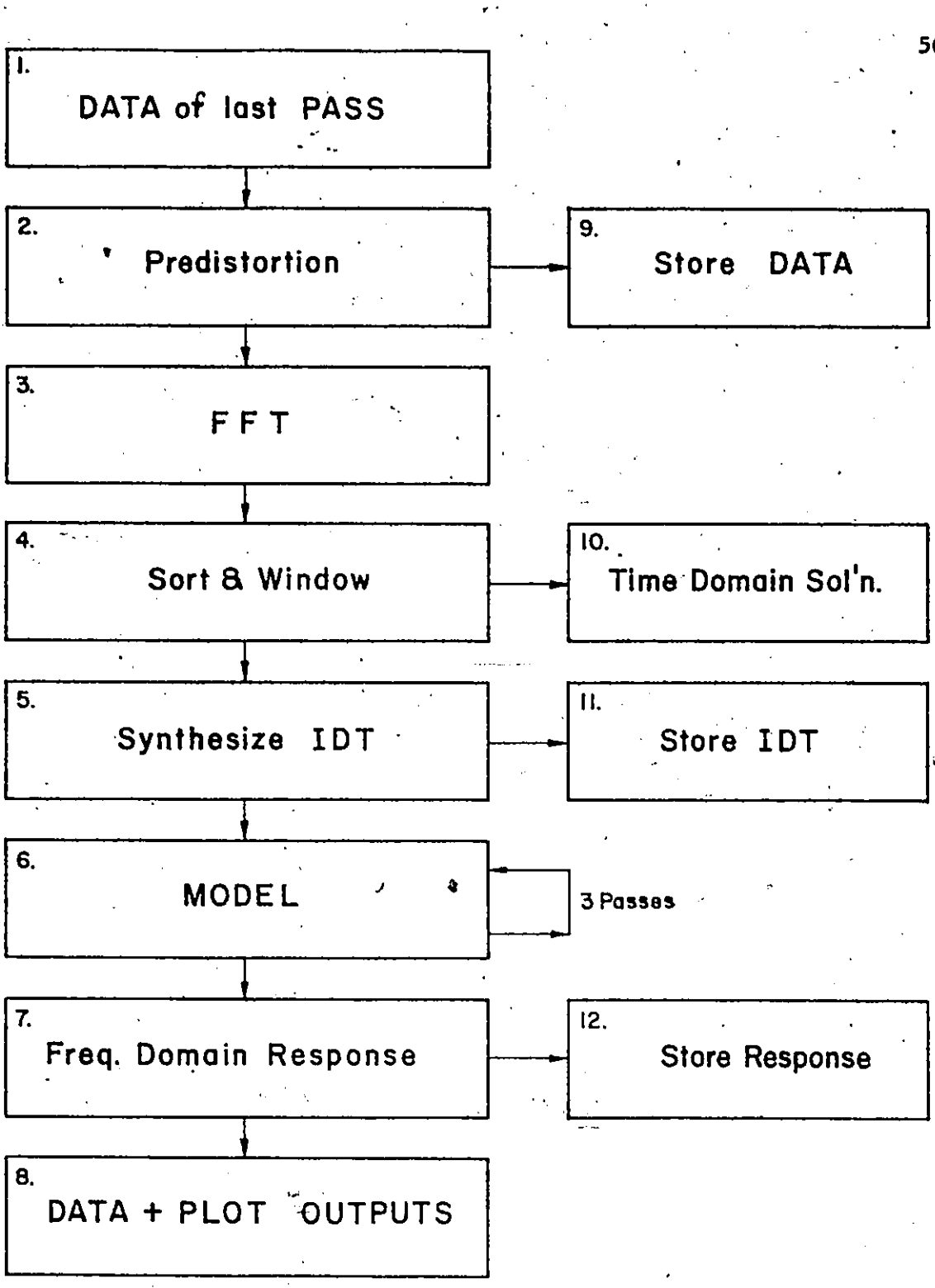


Figure 6.2: Flowchart of programme SAWOPT

designer uses the programme TAPEJOB to generate the transducer mask cutting instructions. TAPEJOB is described next.

6.4 TAPEJOB

The programme TAPEJOB takes the information of L, LEN and TIM (described previously) and generates cutting instructions for a computer controlled cutting table available at the University of Waterloo. Since this is a programme unique to this University (and Waterloo), details of TAPEJOB will not be given. The programme listing is given in Appendix V.

Also a plot is generated by TAPEJOB from the cutting instructions so as to verify that the instructions are correct before cutting the mask.

6.5 Summary

Initially the designer enters his desired specifications into SAWINIT. He then uses SAWOPT interactively until an acceptable transducer has been designed. Then TAPEJOB is used to produce the cutting instructions for the IDT's mask.

CHAPTER 7

EXPERIMENTAL RESULTS

7.1 General

The programmes described in Chapter 6 have been applied to the design of two television intermediate frequency (TVIF) filter designs. Two such filters have been produced which involve the design of three IDT's. The one IDT, arbitrarily called the receiving IDT, is designed as a symmetric linear phase transducer and is responsible for obtaining the stop band insertion loss of 50 dB or better. This transducer is used as the receiver in both filters. The transmitting IDT of the first filter is a single bandpass with the non-symmetric and non-linear specifications of the TVIF. The second filter has a dual bandpass transmitting specification such that the lower bandpass has the required TVIF specifications (Figure 7.1). The upper bandpass is the image of the lower bandpass such that the specifications are Hermitian about a centre frequency of 55 MHz. As a result this IDT can be fabricated with single finger electrodes since the synchronous frequency (frequency at which periodic finger spacing resonates) is placed in the stop band. This allows for a relaxation on the tolerances of fabrication. The receiving IDT accepts only the lower bandpass.

The receiving IDT and the first transmitting IDT have split electrodes while the second transmitting IDT has single fingers.

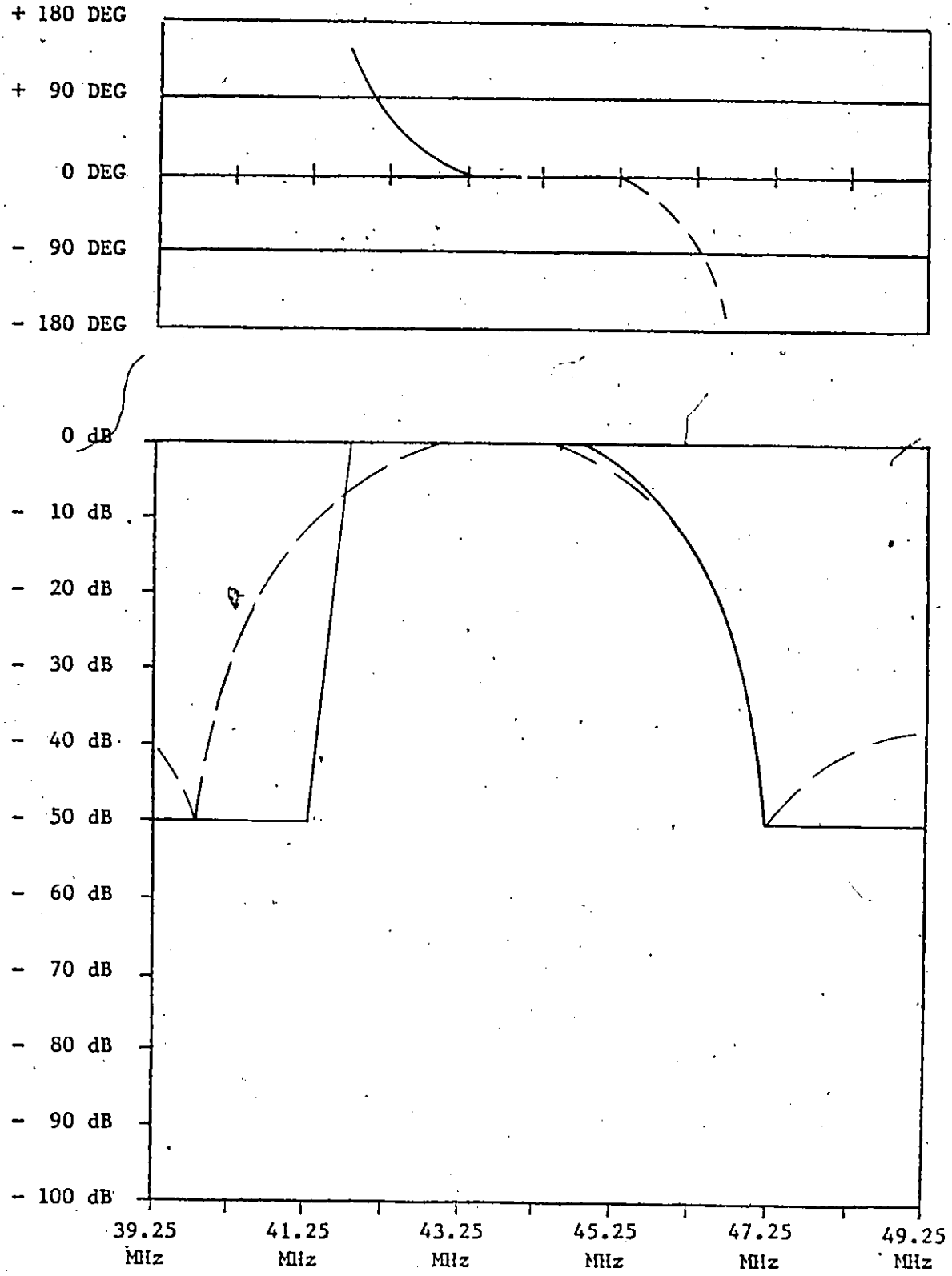


Figure 7.1: TVIF Filter Specifications. Ideal-Solid Line; Typical Lumped Element Compromise-Dashed Line.

7.2 Filter Specifications

The ideal (solid line) and typical compromise (dashed line) filter specifications [21] and [22] are given in Figure 7.1. Since the specifications for TVIF standards were produced at a time when only lumped element filters were feasible, the phase response was allowed to be non-linear so that television receivers would not require expensive phase equalization networks. The transmitted signal of commercial television is predistorted so that the product of the predistorted signal and the TVIF will give a linear phase video signal to the monitor. The phase specification is actually given as group delay:

$$D = -\frac{d\phi}{d\omega} \qquad (7.1)$$

where: D is the group delay in seconds

ϕ is the phase in radians

ω is the angular frequency in radians/second

but for the purposes of this thesis it has been converted to a phase specification as a deviation from linear phase in degrees.

7.3 Single Bandpass Filter

The first filter designed and built has a single bandpass specification. The solution of the 20^{th} pass of SAWOPT for the transmitting IDT and the 15^{th} pass of SAWOPT for the receiving IDT were used to produce the filter mask shown in Figure 7.2.

Photographs of the response of this filter are given in Figure 7.3 and an X-Y plot of the 10 MHz about the centre frequency is shown in

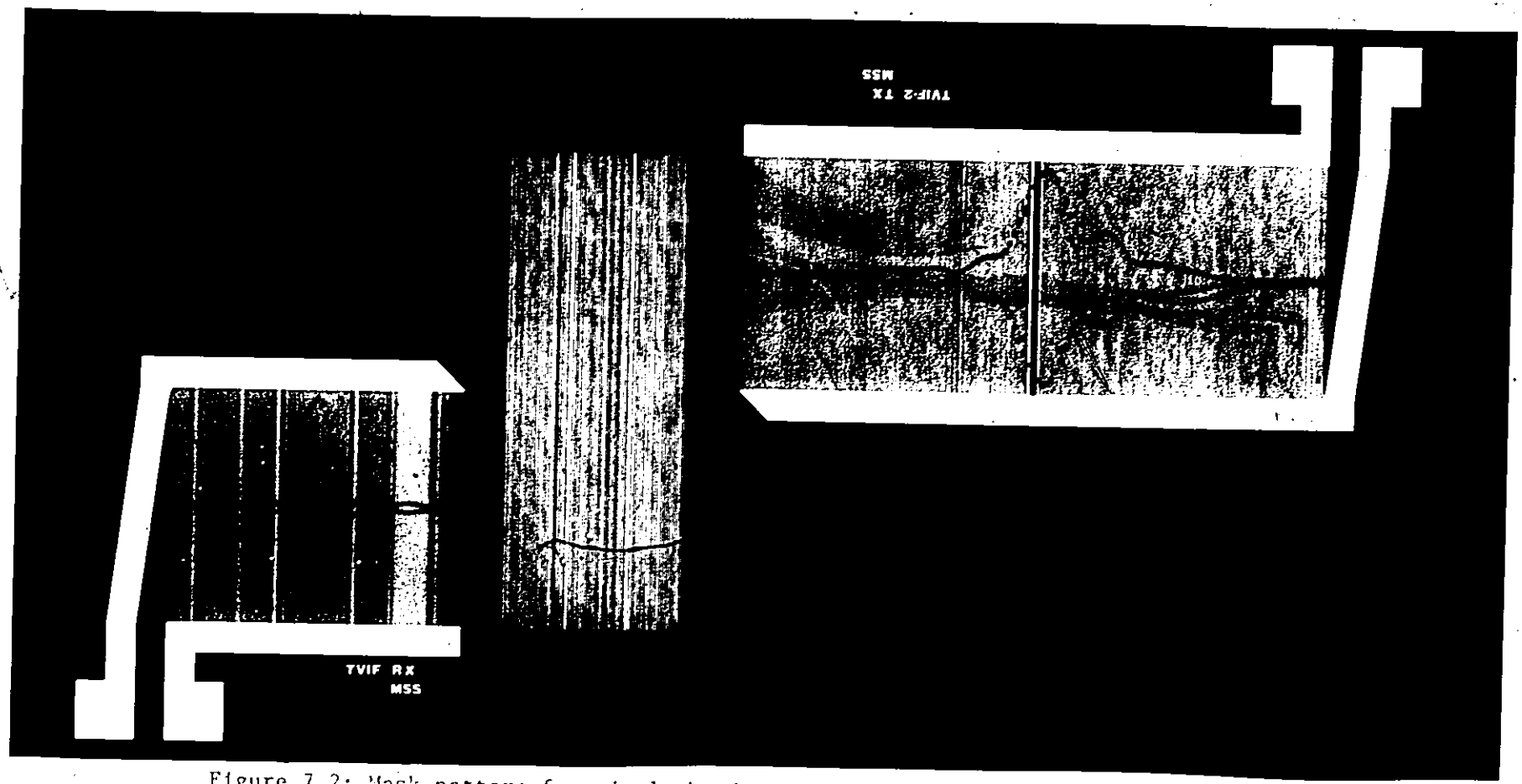
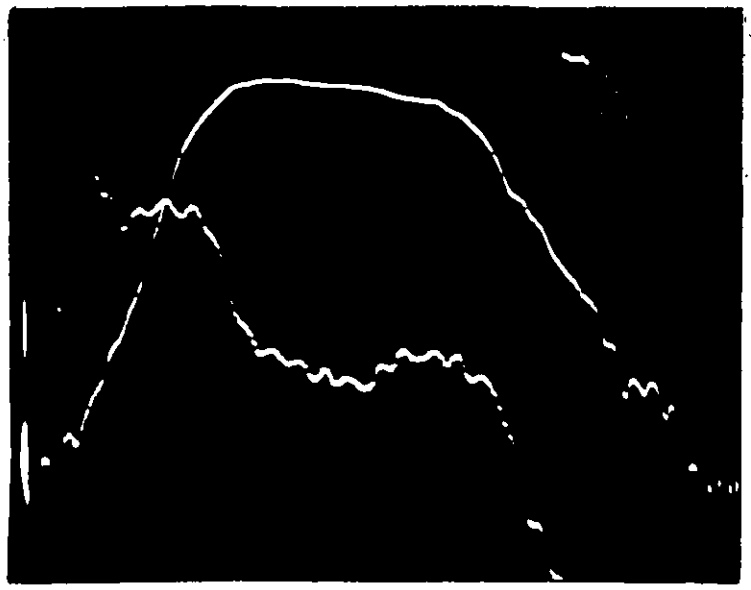


Figure 7.2: Mask pattern for single bandpass filter



39.25 MHz | 44.25 MHz | 49.25 MHz

1 MHz/div ; 10dB/div ; 45°/div

20 MHz | 70 MHz

5 MHz/div ; 10dB/div

Figure 7.3: Response of single bandpass filter

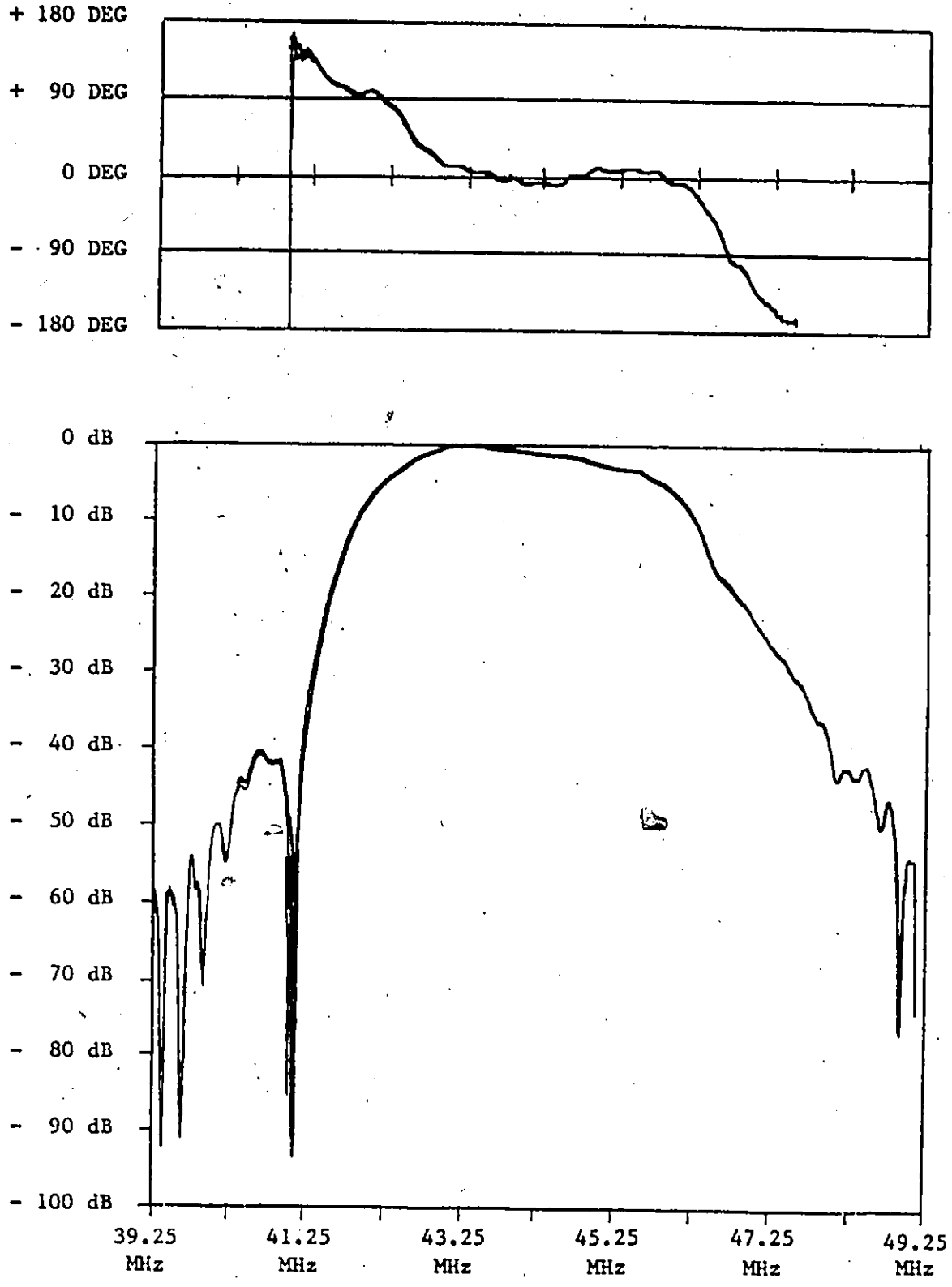


Figure 7.4: X-Y Plot of Single Bandpass Filter Response

Figure 7.4. It is noted that there is an error of approximately 250 kHz in the bandpass (too high) that is due to an error in the final reduction ratio. Also the stop band at f = 41.25 and f = 47.25 are not met. A redesign of the receiving transducer would resolve this problem.

7.4 Dual Bandpass Filter

The second filter built has a dual bandpass Hermitian specification (Figure 7.5). The receiving transducer accepts only the lower bandpass and rejects the upper. The 4^{th} pass of SAWOPT for the transmitting IDT and as before the 15^{th} pass of SAWOPT for the receiver were used to produce the second filter mask of Figure 7.6. Photographs of the response of the second filter are given in Figure 7.7 and an X-Y plot of the 10 MHz about the lower bandpass centre frequency is shown in Figure 7.8. As before the response is approximately 250 kHz high in frequency and the traps at f = 41.25 MHz and f = 47.25 MHz are not met because of the receiving transducer's response being too wide.

7.5 Measuring the Responses

An H.P. 8407A network analyser system (Figure 7.9) was used to obtain the filter responses. Since there is no normalisation in the system, the phase measurement requires an equal length in the reference channel. The only feasible way to account for the acoustic delay in the filter is to use a SAW delay line. Therefore, a delay line was constructed using two receiving transducers placed at the appropriate phase centres (corresponds to t = 0 or r = 0) for the receiving and

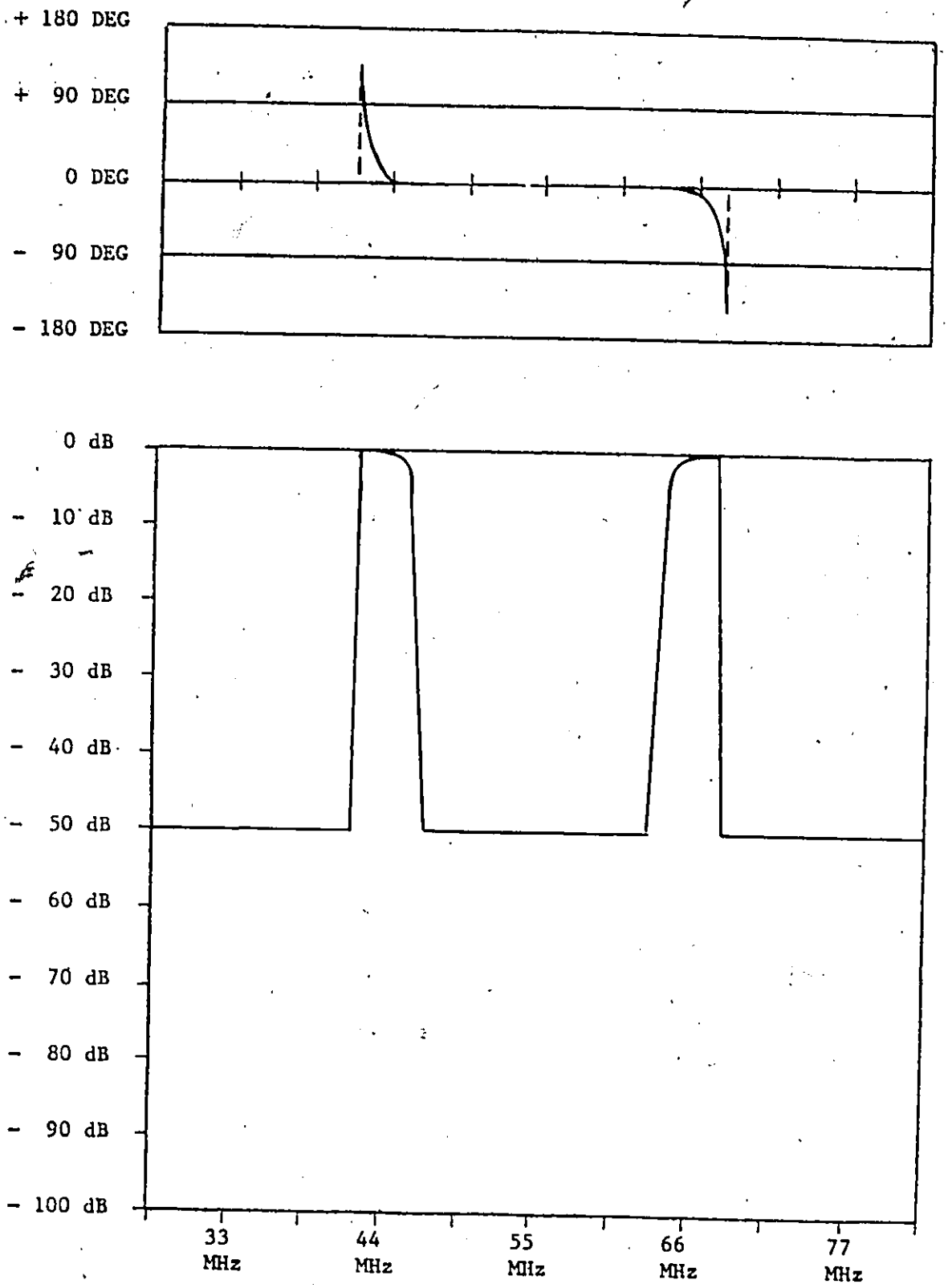


Figure 7.5: Hermitian Specifications for Dual Bandpass IDT

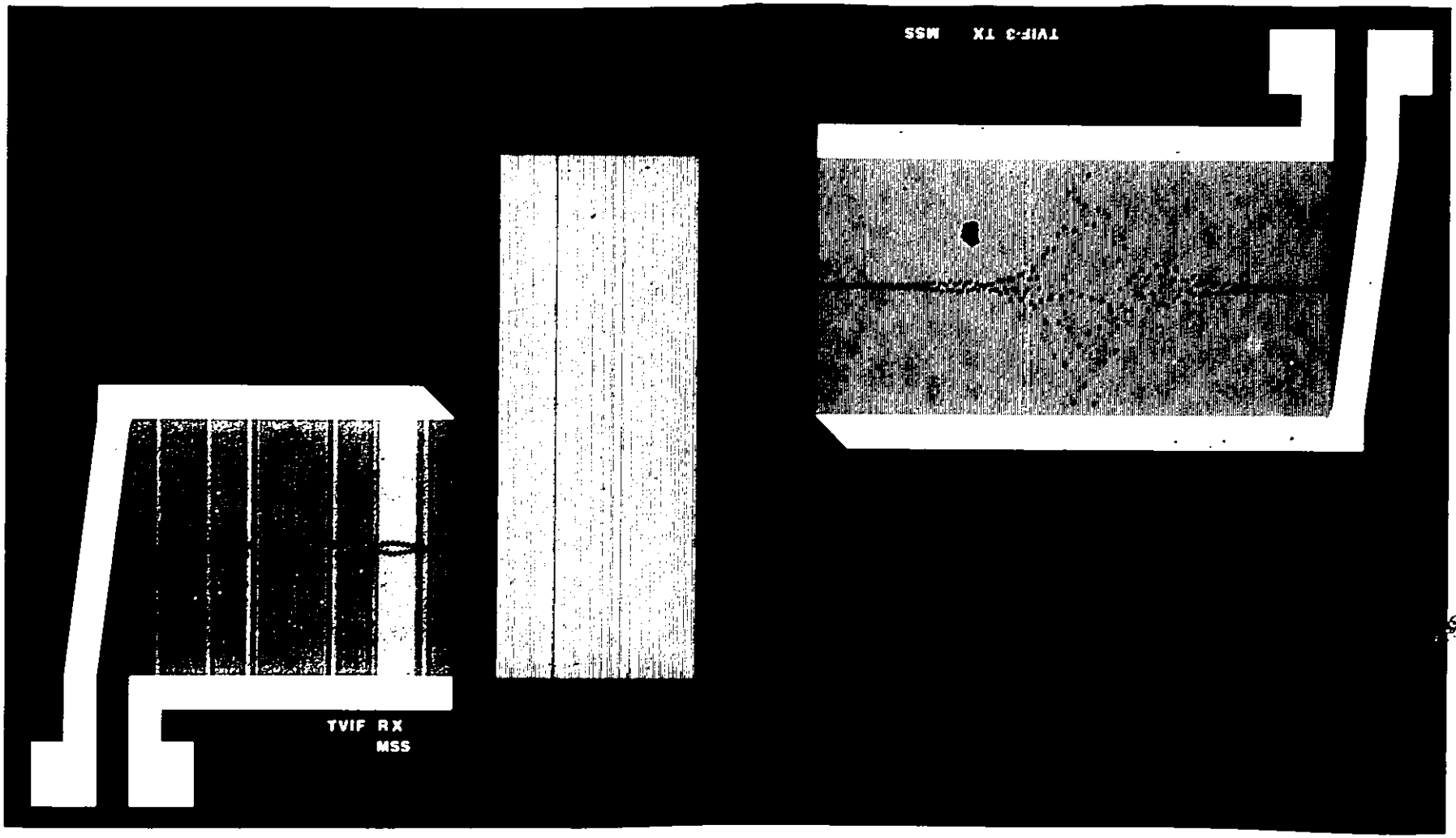
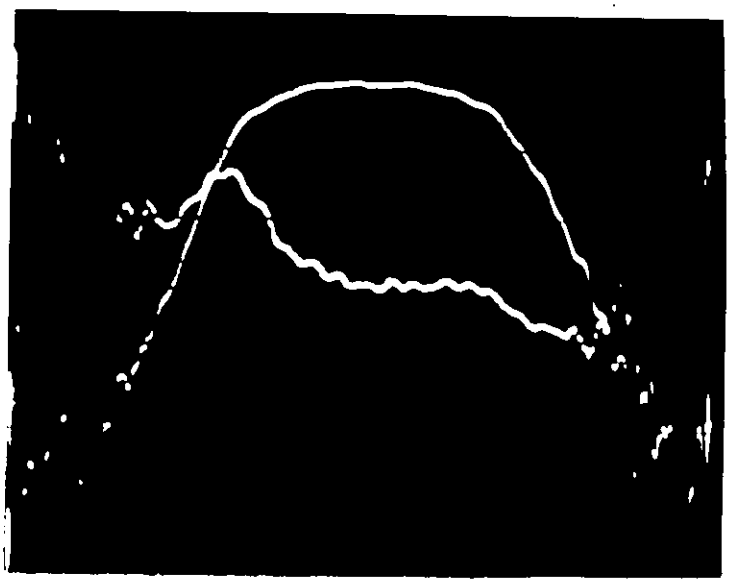
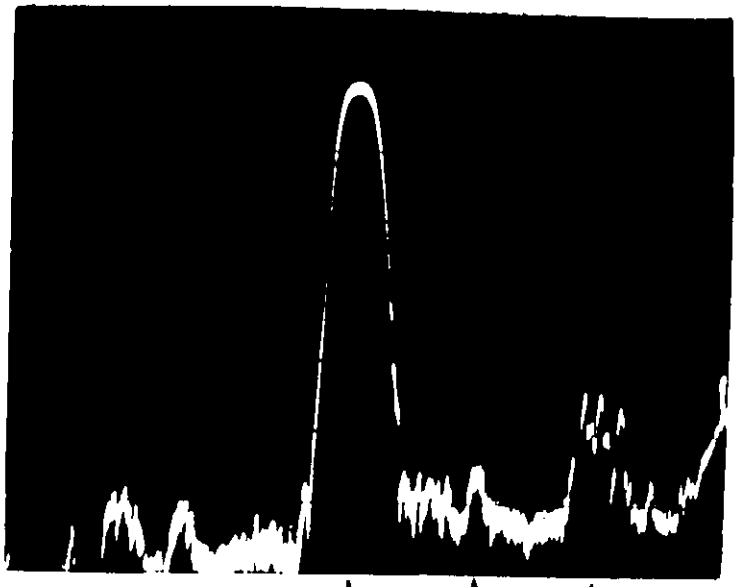


Figure 7.6: Mask pattern for dual bandpass filter



39.25 MHz 44.25 MHz 49.25 MHz

1 MHz/div ; 10dB/div ; 45°/div



10 MHz 55 MHz 80 MHz

7MHz/div
10 dB/div

accepted lower bandpass

rejected upper bandpass

Figure 7.7: Response of dual bandpass filter

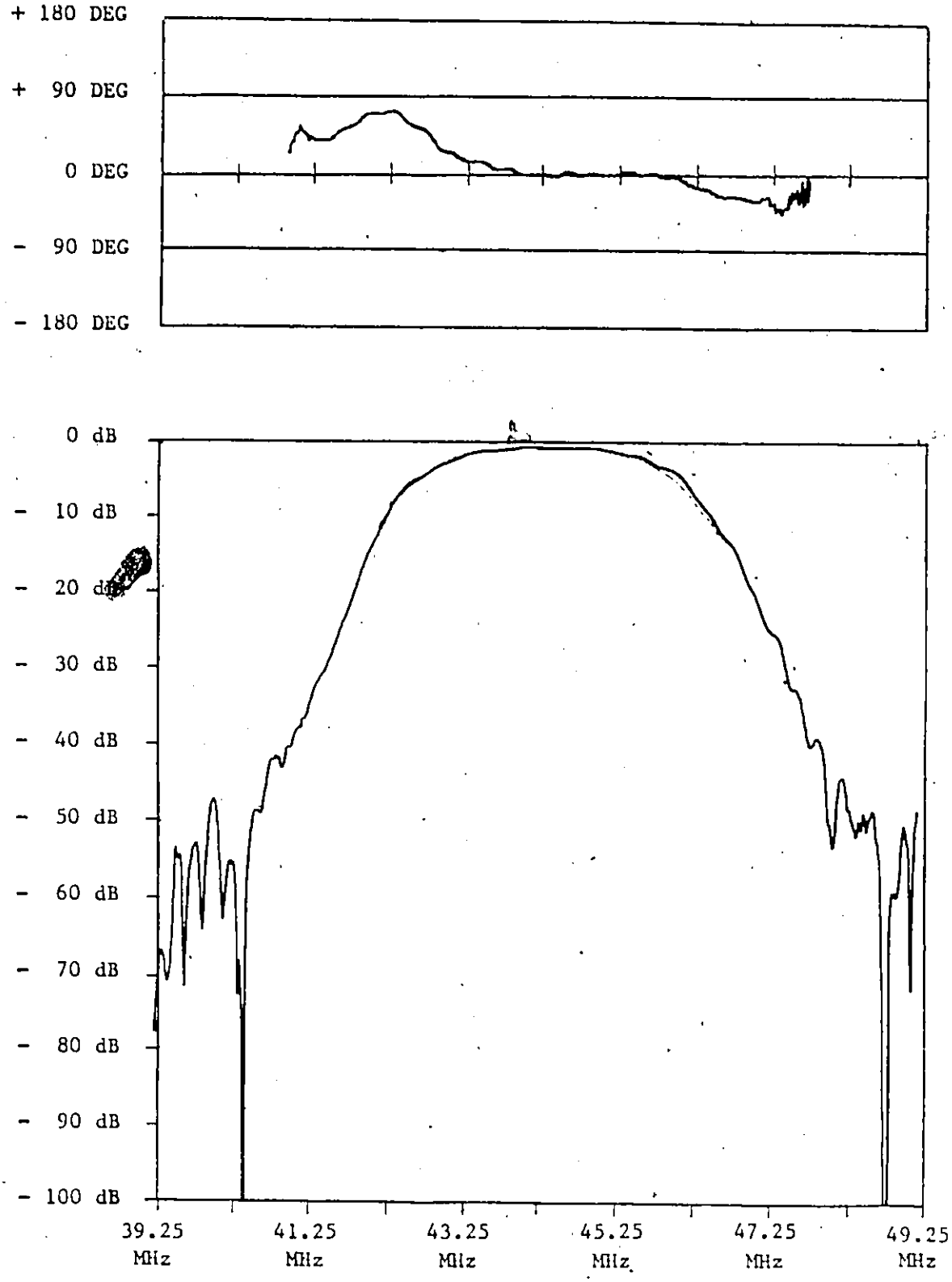


Figure 7.8: X-Y Plot of Dual Bandpass Filter Response

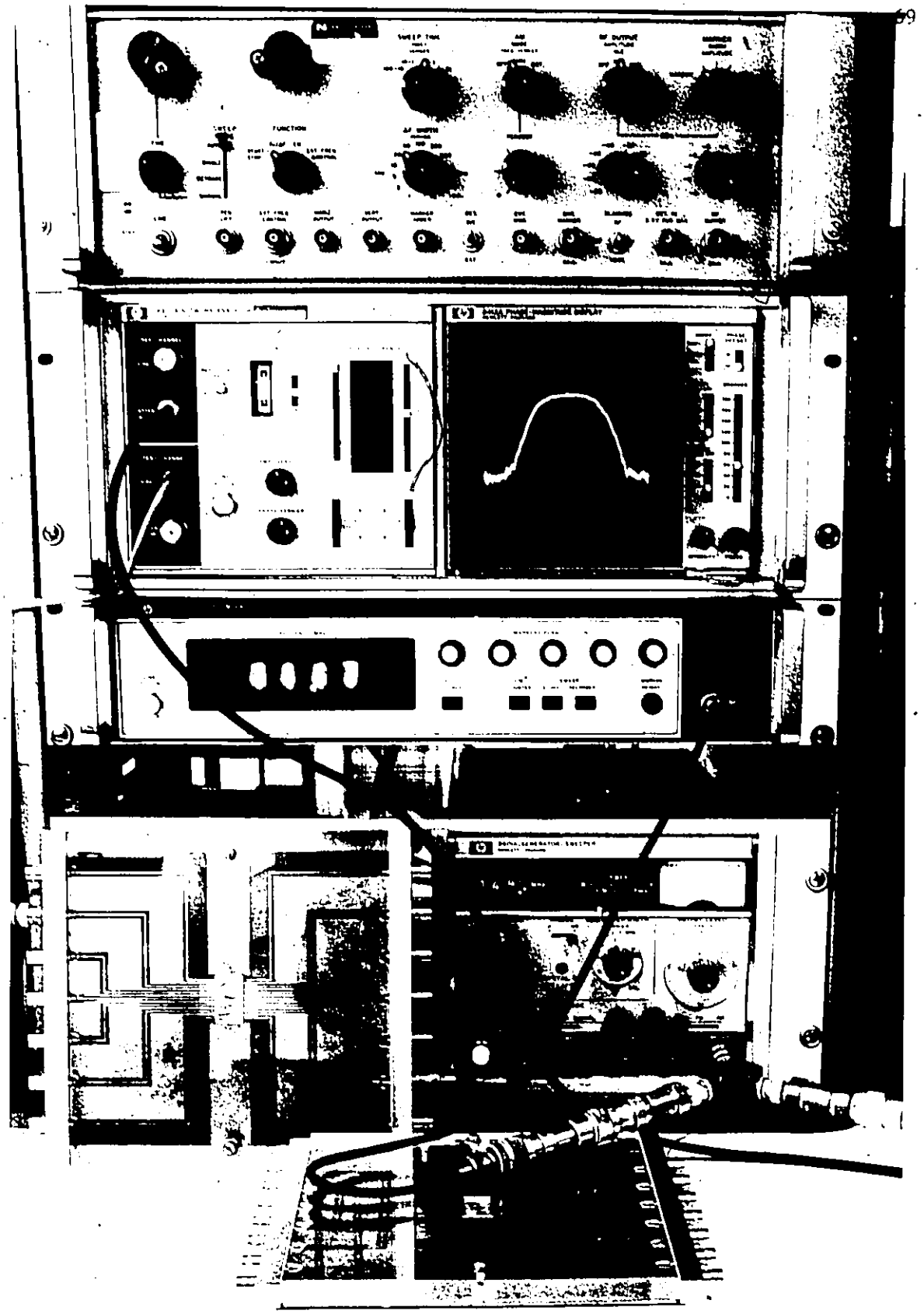


Figure 7.9: Measurement system for testing filters

transmitting IDT's in the master mask. The delay line mask is shown in Figure 7.10 and the response is shown in the X-Y plot of Figure 7.11 for the bandpass region. Another approximate delay line was used to show the linear phase in Figure 6.11. The amplitude response is twice the response of a single receiving IDT and it is obvious that this IDT needs to be redesigned if the traps of the TVIF responses are to be met. The ripple in the phase response is due to electromagnetic feed through in the package for the approximate delay line (a dual inline package which did not provide adequate shielding).

7.6 Summary

The experimental results given in this thesis are used as illustrations of how the algorithm works. Further design work (redesign of receiving IDT) would be required to obtain a valid design that would meet the trap requirements. The filters that have been designed, fabricated and tested indicate the validity of the design process and considering the lumped-element compromise filter (dashed lines in Figure 6.1), the obtained responses compare quite favourably. The receiving IDT theoretical response predicted the wide bandpass as was obtained.

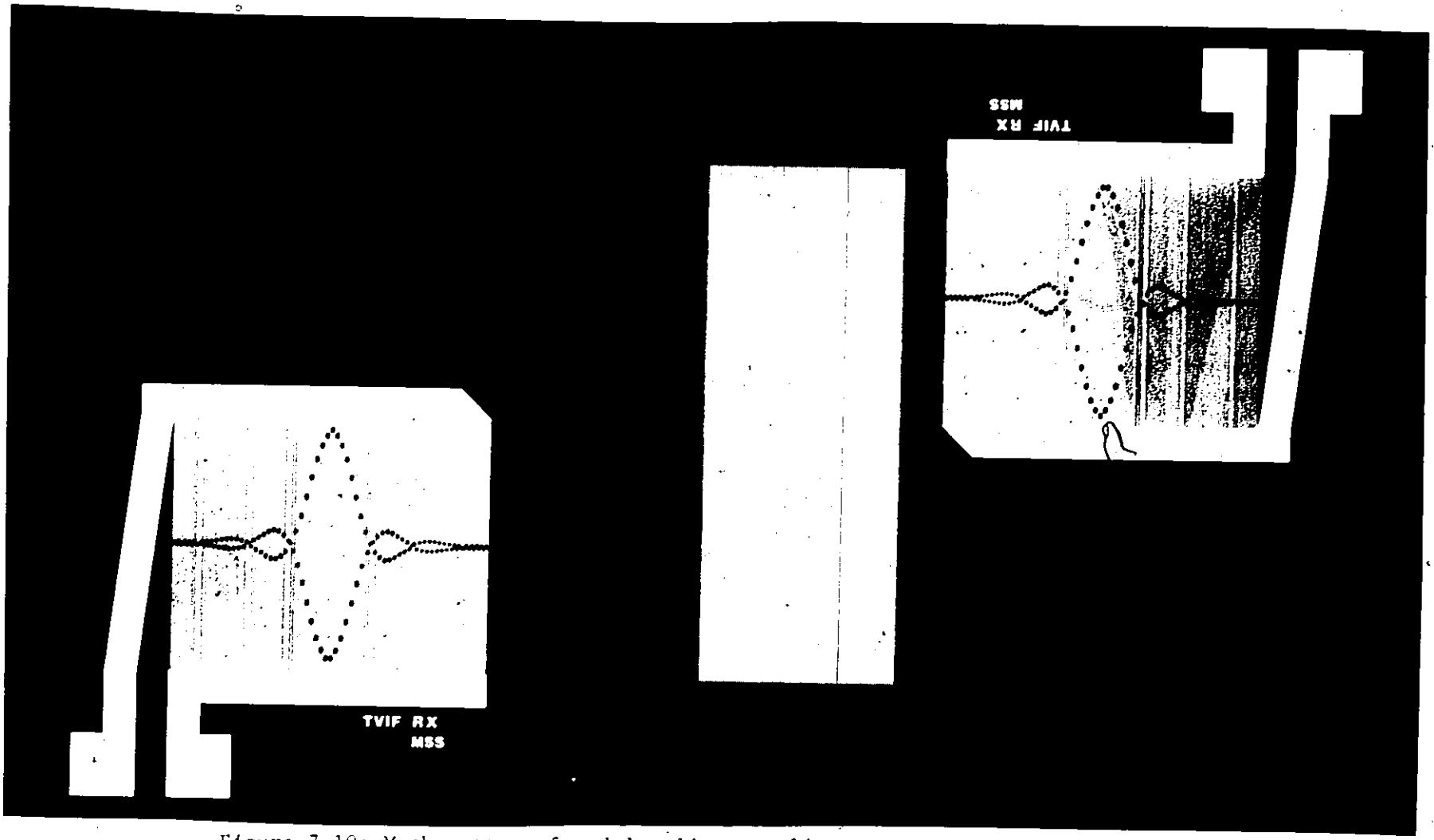


Figure 7.10: Mask pattern for delay-line equalizer

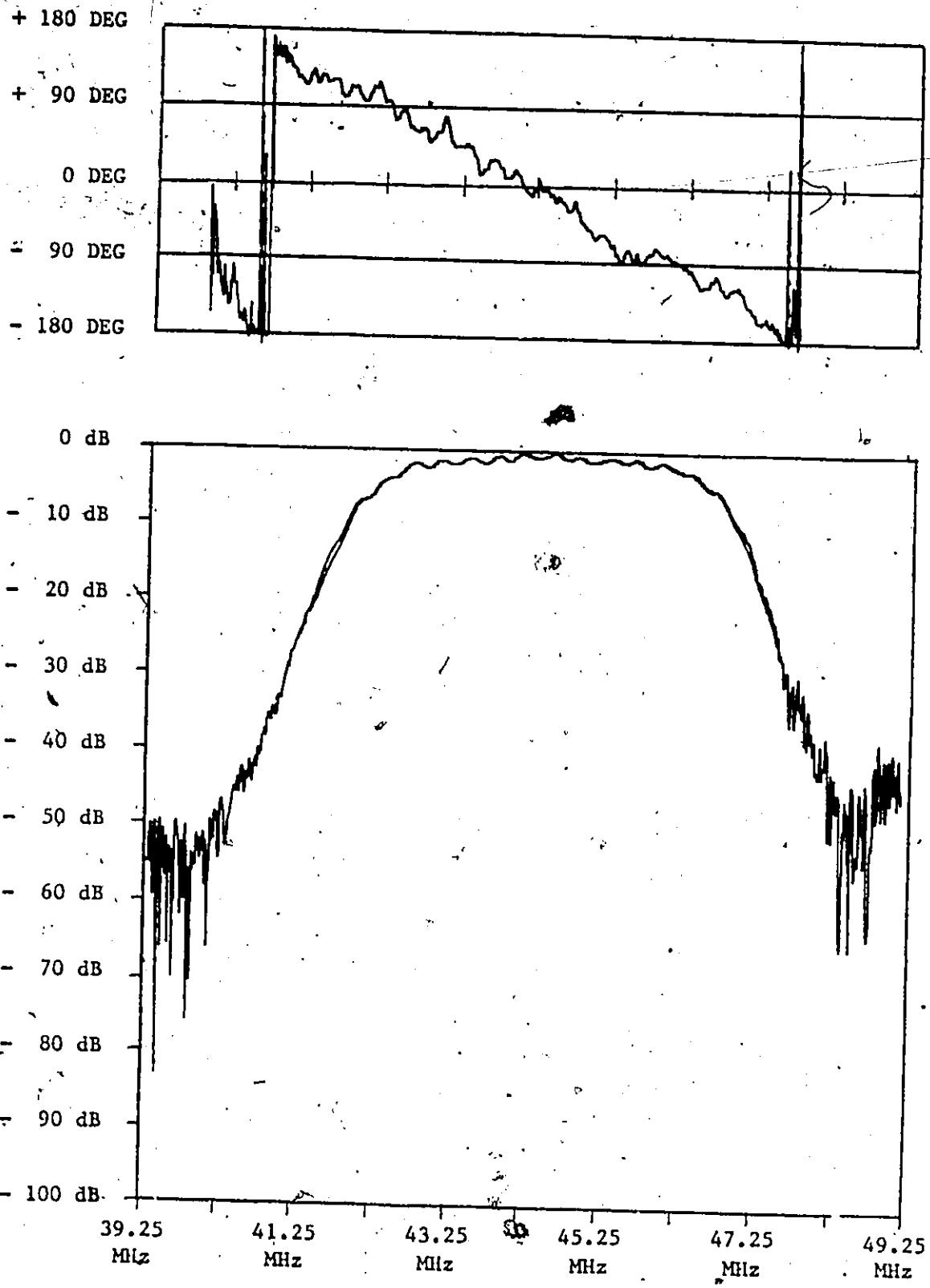


Figure 7.11: X-Y Plot of Delay Line Filter Response

CHAPTER 8

CONCLUSIONS

8.1 General

A design algorithm has been produced comprised of the following elements:

(a) Synthesis

(b) charge distribution model analysis

(c) frequency domain response

(d) numerical optimization of design to realize a desired response.

The algorithm has been programmed and the model has been developed mathematically and then some difficult filter responses were attempted. The designed filters were fabricated and tested showing the algorithm, the programmes and the model to all be quite valid.

Because of the broadness of this investigation, most of the areas that go to make up the algorithm could not be investigated in as much depth as would be desirable. The following sections suggest areas of further research, development and refinement.

8.2 Synthesis

In the synthesis, further work needs to be done to establish the final effect of the windowing function. Some windows such as Bartlett, Hamming and Hanning [20] were investigated in the course of this study

but the actual quantitative results of the various functions and their variable arguments have not been researched with respect to this design algorithm. A series of SAW filters need to be fabricated and measured using the algorithm with various windows to see what effect they have on the resulting device with respect to design time, band edge rounding, side lobe suppression and stop band insertion loss.

8.3 Charge Distribution Model

The charge model developed in this study is the solution to the model suggested in [18]. It has been used based on the claims of the authors of [18], however, further research on this model is warranted, particularly with respect to accounting for variable finger lengths.

The numerical solution to the charge distribution coefficients could be made much more efficient since only the nearby fingers will have a significant effect on a finger's charge distribution.

Also, it would be desirable if the variable finger length could be accounted for in the model rather than using the approximation in Chapter 3.

8.4 Frequency Response

The frequency response of the model does not include consideration of finger edge reflections. Since split electrodes or synchronous frequencies in the stop band have been used, this did not present a problem in this study. It is necessary under many other conditions, however, to use synchronous frequencies in the bandpass and

since lithography constraints at higher frequencies preclude the use of split electrodes, accouting for finger reflections is necessary if passband ripple is to be avoided.

Also, in much more advanced studies, diffraction in the far field could be included as well as accounting for the scattering at apodization cuts (finger ends).

8.5 Optimization

This area offers the greatest potential for further research. Attempts at optimizing the IDT geometry with respect to the desired and realized frequency responses were attempted in this study. While some positive results were obtained using this approach, the optimization was not very efficient since the response at all frequencies changed if one finger was moved or changed in length. Direct manipulation of the fingers is desirable but has not proved to very efficient in terms of computer time. Perhaps using the transform algorithm as described in this thesis until very closé results are obtained and then optimizing the IDT directly would prove effective.

Also, the possibility of using gradient search techniques should be investigated since the apodization function is well behaved near the solution.

8.6 Summary

There are at least four areas of significant potential for further research that will refine the design algorithm presented in this

thesis. Since surface acoustic wave filters offer such a significant improvement over lumped element filters in the 10 MHz to 1 GHz region, in size, ruggedness and realizable designs it is essential that effective and efficient design packages be provided to the design engineer that are easy to use and provide usable results without in-depth knowledge of how SAW devices work. The study presented in this thesis is an initial step in providing the design engineer with that facility.

APPENDIX I

SOLUTION TO CHARGE-DISTRIBUTION INTEGRALS

I.1 General

The integral equations of Chapter 3 to be solved are:

$$I_{10} = -\int_{-b}^{b} \frac{\ln |x-x_o|}{\sqrt{b^2 - x_o^2}}\, dx_o \qquad \text{(I.1a)}$$

$$I_{20} = -\int_{-b}^{b} \frac{x_o \ln |x-x_o|}{\sqrt{b^2 - x_o^2}}\, dx_o \qquad \text{(I.1b)}$$

$$I_{30} = -\int_{-b}^{b} \frac{x_o^2 \ln |x-x_o|}{\sqrt{b^2 - x_o^2}}\, dx_o \qquad \text{(I.1c)}$$

The table of integrals used in obtaining these solutions is Reference [23].

I.2 Solution to I_{10}

$$I_{10} = -\int_{-b}^{b} \frac{\ln |x-x_o|}{\sqrt{b^2 - x_o^2}}\, dx_o$$

using the transformations:

$$x_o = b\cos\theta$$

$$dx_o = -\, b\sin\theta\, d\theta$$

$$\sqrt{b^2 - x_o^2} = b\sin\theta \qquad \text{(I.2)}$$

(i) $x = 0$; making the substitutions of (I.2)

$$I_{10} = -\int_0^{\pi} \ln\,|-b\cos\theta|\, d\theta$$

$$= -\,2\int_0^{\pi/2} \ln\,(b\cos\theta)\, d\theta$$

$$= -\,2\,[\tfrac{\pi}{2}\ln\,(b) - \tfrac{\pi}{2}\ln\,(2)]$$

Thus

$$I_{10} = \pi\,\ln\,(2/b) \quad \text{for} \quad x = 0 \qquad \text{(I.3)}$$

(ii) $x = +b$; making the substitutions (I.2):

$$I_{10} = -\int_0^{\pi} \ln\,|b - b\cos\theta|\, d\theta$$

for $\theta = \pi$ the argument of ln is $b - b\cos\pi = 2b$ and for $\theta = 0$ the argument is >0 over region of integration which gives:

$$I_{10} = -\int_0^{\pi} \ln\,(b - b\cos\theta)\, d\theta$$

$$= -\,\pi\,\ln\,[\frac{b - \sqrt{b^2 - b^2}}{2}]$$

Thus

$$I_{10} = \pi\,\ln\,(2/b) \quad \text{for} \quad x = +b \qquad \text{(I.4)}$$

(iii) $x = -b$; making the substitutions of (I.2)

$$I_{10} = -\int_0^{\pi} \ln |-b - b\cos\theta|\, d\theta$$

$$= -\int_0^{\pi} \ln |(-1)\ (b + b\cos\theta)|\, d\theta$$

since $(b + b\cos\theta) \geq 0$ for $0 \leq \theta \leq \pi$ then:

$$I_{10} = -\int_0^{\pi} \ln (b + b\cos\theta)\, d\theta$$

gives

$$I_{10} = \pi \ln (2/b) \quad \text{for } x = -b \tag{I.5}$$

(iv) $x > +b$; with substitutions (I.2)

$$I_{10} = -\int_0^{\pi} \ln |x - b\cos\theta|\, d\theta$$

since $x > +b$ then argument of ln is > 0 thus

$$I_{10} = -\int_0^{\pi} \ln (x - b\cos\theta)\, d\theta$$

which gives

$$I_{10} = \pi \ln \left(\frac{2}{x + \sqrt{x^2 - b^2}}\right) \quad \text{for } x > +b \tag{I.6}$$

(v) $x < -b$; with substitutions (I.2)

$$I_{10} = -\int_0^{\pi} \ln |x - b\cos\theta|\, d\theta$$

$$= - \int_0^{\pi} \ln |(-1)\ (x + b\cos\theta)|\ d\theta$$

taking $x > +b$ then argument of ln will be +ve over $0 \leq \theta \leq \pi$ if (-1) is removed giving

$$I_{10} = \int_0^{\pi} \ln (x + b\cos\theta)\ d\theta$$

thus

$$I_{10} = \pi \ln \left(\frac{2}{x + \sqrt{x^2 - b^2}}\right) \quad \text{for} \quad x < -b \qquad \text{(I.7)}$$

Thus the first integral (I_{10}) has been solved for all the points on the x-axis where the electrostatic potential will be calculated.

I.3 Solution of I_{20}

Solving for I_{20}

$$I_{20} = - \int_{-b}^{b} \frac{x_o \ln |x-x_o|}{\sqrt{b^2 - x_o^2}}\ dx_o$$

$$= \int_{-b}^{b} [\ln |x-x_o|] \cdot \left[\frac{-x_o}{\sqrt{b^2 - x_o^2}}\right] dx_o$$

$$= \sqrt{b^2 - x_o^2} \ln |x-x_o| \Big|_{-b}^{b} - \int_{-b}^{b} (\sqrt{b^2 - x_o^2}) \left(\frac{d \ln |x-x_o|}{dx_o}\right) dx_o$$

$$= \sqrt{b^2 - b^2} \ln |x-b| - \sqrt{b^2 - b^2} \ln |x+b| - \int_{-b}^{b} \sqrt{b^2-x_o^2} \frac{d \ln |x-x_o|}{dx_o} dx_o$$

$$I_{20} = - \int_{-b}^{b} \sqrt{b^2-x_o^2} \frac{d \ln |x-x_o|}{dx_o} dx_o \qquad (I.8)$$

consider the transformation

$$u = x - x_o \qquad x_o = x - u$$

$$du = - dx_o \qquad dx_o = -du \qquad (I.9)$$

then

$$I_{20} = - \int_{x+b}^{x-b} \sqrt{(b^2-x^2) + (2x)u + (-1)u^2} \frac{d \ln |u|}{du} du \qquad (I.10)$$

(i) $x = 0$; taking (I.8) with $x = 0$

$$I_{20} = - \int_{-b}^{b} \sqrt{b^2-x_o^2} \frac{d \ln |-x_o|}{dx_o} dx_o$$

$$= - \int_{-b}^{0} \sqrt{b^2-x_o^2} \frac{d \ln (-x_o)}{dx_o} dx_o$$

$$= - \int_{0}^{b} \sqrt{b^2-x_o^2} \frac{d \ln (x_o)}{dx_o} dx_o$$

$$= - \int_{0}^{-b} \frac{\sqrt{b^2 - x_o^2}}{-x_o} dx_o - \int_{0}^{b} \frac{\sqrt{b^2 - x_o^2}}{x_o} dx_o$$

$$= + \int_{0}^{b} \frac{\sqrt{b^2 - x_o^2}}{x_o} dx_o - \int_{0}^{b} \frac{\sqrt{b^2 - x_o^2}}{x_o} dx_o$$

Thus $I_{20} = 0$ for $x = 0$ (I.11)

(ii) $x = +b$; taking (I.10) and putting $x = +b$ gives:

$$I_{20} = \int_{b+b}^{b-b} \sqrt{(b^2-b^2) + (2b)u + (-1)u^2}\, \frac{d \ln |u|}{du}\, du$$

$$= -\int_{2b}^{0} \frac{\sqrt{2bu - u^2}}{u}\, du$$

$$= -\{\sqrt{2bu-u^2} - \sqrt{2bu} \ln [\frac{\sqrt{2bu} + \sqrt{2bu-u^2}}{u}]\} \Big|_{2b}^{0}$$

$$= 0$$

Thus $I_{20} = 0$ for $x = +b$ (I.12)

(iii) $x = -b$; taking (I.10) and putting $x = -b$ gives:

$$I_{20} = -\int_{-b+b}^{-b-b} \sqrt{(b^2-b^2) - (2b)u + (-1)u^2}\, \frac{d \ln |u|}{du}\, du$$

$$= -\int_{0}^{-2b} \frac{\sqrt{-2bu - u^2}}{-u}\, du$$

$$= -\int_{0}^{2b} \frac{\sqrt{2bu - u^2}}{u}\, du$$

$$= -\{\sqrt{2bu - u^2} - \sqrt{2bu} \ln [\frac{\sqrt{2bu} + \sqrt{2bu - u^2}}{u}]\} \Big|_{0}^{2b}$$

$$= 0$$

Thus $I_{20} = 0$ for $x = -b$ (I.13)

(iv) $x > b$; taking (I.10) gives

$$I_{20} = -\int_{x+b}^{x-b} \sqrt{(b^2-x^2) + (2x)u + (-1)u^2}\, \frac{d \ln |u|}{du}\, du$$

$$= -\int_{x+b}^{x-b} \frac{\sqrt{(b^2-x^2) + (2x)u + (-1)u^2}}{u}\, du$$

$$= -\sqrt{U}\,\Big|_{x+b}^{x-b} - \frac{B}{2}\int_{x+b}^{x-b} \frac{du}{\sqrt{U}} - A\int_{x+b}^{x-b} \frac{du}{u\sqrt{U}} \qquad (I.14)$$

where $U = (b^2-x^2) + (2x)u + (-1)u^2$

$A = b^2 - x^2$

$\left.\begin{matrix} B = 2x \\ C = -1 \end{matrix}\right\}$

From (I.14) $\sqrt{U}\,\Big|_{x+b}^{x-b} = 0$

and $\frac{B}{2}\int_{x+b}^{x-b} \frac{du}{\sqrt{U}}$

$$= -\frac{B/2}{\sqrt{-1}} \sin^{-1} \left[\frac{-2u + 2x}{\sqrt{4Au + 4x^2}}\right]_{x+b}^{x-b}$$

$$= \pi x$$

and $A\int_{x+b}^{x-b} \frac{du}{u\sqrt{U}}$

$$= \frac{(b^2 - x^2)}{\sqrt{x^2 - b^2}} \sin^{-1} \left[\frac{2xu + 2b^2 - 2x^2}{|u| \sqrt{4b^2 - 4x^2 + 4x^2}} \right]_{x+b}^{x-b}$$

$$= \pi \sqrt{x^2 - b^2}$$

Thus combining the above terms into (I.14) gives

$$I_{20} = - \pi \left[\sqrt{x^2 - b^2} - x \right] \quad \text{for} \quad x > b \tag{I.15}$$

(v) $x < -b$; taking (I.10) gives

$$I_{20} = - \int_{x+b}^{x-b} \sqrt{(b^2-x^2) + (2x)u + (-1)u^2} \, \frac{d \ln |u|}{du} \, du$$

$$= - \int_{-(|x|-b)}^{-(|x|+b)} \frac{\sqrt{U}}{-u} \, du$$

$$= + \int_{|x|+b}^{|x|-b} \frac{\sqrt{U}}{u} \, du$$

which is the negative of case (iv). Thus

$$I_{20} = +\pi \left[\sqrt{x^2 - b^2} - x \right] \quad \text{for} \quad x < -b \tag{I.16}$$

Thus the second integral has been solved for all the points on the x-axis where the electrostatic potential was calculated.

I.4 Solution of I_{30}

Solving for I_{30}

$$I_{30} = -\int_{-b}^{b} \frac{x_o^2 \ln |x-x_o|}{\sqrt{b^2 - x_o^2}}\, dx_o$$

(i) $x = 0$; taking $x = 0$

$$I_{30} = -\int_{-b}^{b} \frac{x_o^2 \ln |-x_o|}{\sqrt{b^2 - x_o^2}}\, dx_o$$

$$= -2\int_{0}^{b} [x_o \ln(x_o)] \left[-\frac{x_o}{\sqrt{b^2 - x_o^2}} \right] dx_o$$

$$= \frac{b^2\pi}{2}\left[\ln\left(\frac{2}{b}\right) - \frac{1}{2}\right]$$

Thus $I_{30} = \frac{b^2\pi}{2}\left[\ln\left(\frac{2}{b}\right) - \frac{1}{2}\right]$ for $x = 0$ (I.17)

(ii) $x = +b$; taking $x = +b$ then:

$$I_{30} = -\int_{-b}^{b} \frac{x_o^2 \ln (b-x_o)}{\sqrt{b^2 - x_o^2}}\, dx_o$$

$$= -\int_{-b}^{b} \left[\frac{x_o^2}{\sqrt{b^2 - x_o^2}}\right] \cdot [x_o \ln (b-x_o)]\, dx_o$$

Integrating by parts gives:

$$I_{30} = [x_o \sqrt{b^2-x_o^2} \ln (b-x_o)]\Big|_{-b}^{b} - \int_{-b}^{b} \sqrt{b^2-x_o^2} \left[\ln (b-x_o) - \left(\frac{x_o}{b-x_o}\right)\right] dx_o$$

$$= \int_{-b}^{b} \frac{x_o \sqrt{b^2 - x_o^2}}{b - x_o} dx_o - \int_{-b}^{b} \sqrt{b^2 - x_o^2} \ln (b-x_o) \, dx_o \qquad (I.18)$$

Taking one term at a time and applying the transformation:

$$x_o = b\cos\theta \qquad dx_o = -b\sin\theta \, d\theta$$

$$\sqrt{b^2 - x_o^2} = b\sin\theta \qquad d\theta = -\frac{dx_o}{\sqrt{b^2 - x_o^2}}$$

gives for the second term of (I.18):

$$- \int_{-b}^{b} \sqrt{b^2 - x_o^2} \ln (b-x_o) \, dx_o$$

$$= \int_{\pi}^{0} b^2 \sin^2 \theta \ln (b - b\cos\theta) \, d\theta$$

$$= b^2 \int_{\pi}^{0} \ln (b-b\cos\theta) \, d\theta - \int_{\pi}^{0} b^2 \cos^2\theta \ln (b-b\cos\theta) \, d\theta$$

$$= - b^2 \int_{0}^{\pi} \ln (b-b\cos\theta) \, d\theta + \int_{-b}^{b} \frac{x^2 \ln (b-x_o)}{\sqrt{b^2 - x_o^2}} dx_o$$

$$= - b^2 \pi \ln \left[\frac{b + \sqrt{b^2 - b^2}}{2}\right] - I_{30}$$

$$= - b^2 \pi \ln (b/2) - I_{30} \qquad (I.19)$$

Now taking the first term of (I.18) and applying the tranformation:

$$u = b - x_o \qquad du = -dx_o$$

$$x_o = b - u \qquad dx_o = -du$$

gives:
$$\int_{-b}^{b} \frac{x_o \sqrt{b^2 - x_o^2}}{b - x_o} \, dx_o$$

$$= - \int_{b+b}^{b-b} \frac{(b-u) \sqrt{b^2 - b^2 + 2bu - u^2}}{u} \, du$$

$$= - \int_{2b}^{0} \frac{b \sqrt{2bu - u^2}}{u} du + \int_{2b}^{0} \sqrt{2bu - u^2} \, du$$

$$= - \frac{b^2 \pi}{2} - b^2 \pi \qquad (I.20)$$

Substituting the results (I.19) and (I.20) into (I.18) gives:

$$I_{30} = - \frac{b^2 \pi}{2} - b^2 \pi - b^2 \pi \ln \left(\frac{b}{2}\right) - I_{30}$$

$$2 I_{30} = - b^2 \pi \left[\ln \left(\frac{b}{2}\right) + \frac{3}{2}\right]$$

thus $$I_{30} = - \frac{b^2 \pi}{2} \left[\ln \left(\frac{b}{2}\right) + \frac{3}{2}\right] \quad \text{for } x = +b \qquad (I.21)$$

(iii) $x = -b$; taking $x = -b$

$$I_{30} = - \int_{-b}^{b} \frac{x_o^2 \ln |-b-x_o|}{\sqrt{b^2 - x_o^2}} dx_o$$

$$= - \int_{-b}^{b} \frac{x_o^2 \ln |(-1)\ (b+x_o)|}{\sqrt{b^2 - x_o^2}} dx_o$$

$$= - \int_{-b}^{b} \frac{x_o^2 \ln (b+x_o)}{\sqrt{b^2 - x_o^2}} dx_o$$

transferring $x = -b$ to $x = +b$

$$I_{30} = + \int_{b}^{-b} \frac{x_o^2 \ln (b-x_o)}{\sqrt{b^2 - x_o^2}} dx_o$$

and reversing order of integration

$$I_{30} = - \int_{-b}^{b} \frac{x_o \ln (b-x_o)}{\sqrt{b^2 - x_o^2}} dx_o$$

which is the same as for case (ii)

thus $I_{30} = - \frac{b^2 \pi}{2} [\ln (\frac{b}{2}) + \frac{3}{2}]$ for $x = -b$ (I.22)

(iv) $x > +b$; taking the region of $x > +b$

$$I_{30} = - \int_{-b}^{b} \frac{x_o^2 \ln |x-x_o|}{\sqrt{b^2 - x_o^2}} dx_o$$

since $x > b$ then $x - x_o > 0$ for region of integration

$$I_{30} = -\int_{-b}^{b} \frac{x_o^2 \ln (x-x_o)}{\sqrt{b^2 - x_o^2}}\, dx_o$$

$$= -\int_{-b}^{b} [x_o \ln (x-x_o)] \cdot \left[\frac{x_o}{\sqrt{b^2 - x_o^2}}\right] dx_o$$

Integrating by parts gives:

$$I_{30} = [x_o \sqrt{b^2-x_o^2} \ln (x-x_o)]_{-b}^{b} - \int_{-b}^{b} \sqrt{b^2-x_o^2} \left[\ln (x-x_o) - \frac{x_o}{(x-x_o)}\right] dx_o$$

$$= 0 - \int_{-b}^{b} \sqrt{b^2-x_o^2} \ln (x-x_o)\, dx_o + \int_{-b}^{b} \frac{x_o \sqrt{b^2-x_o^2}}{x - x_o}\, dx_o$$

Taking the first term and substituting:

$x_o = b\cos\theta$ $\qquad$ $dx_o = -b\sin\theta\, d\theta$

$\sqrt{b^2 - x_o^2} = b\sin\theta$ $\qquad$ $d\theta = -\dfrac{dx_o}{\sqrt{b^2 - x_o^2}}$

gives: $-\int_{-b}^{b} \sqrt{b^2 - x_o^2} \ln (x-x_o)\, dx_o$

$$= \int_{\pi}^{0} b^2 \sin^2\theta \ln (x-b\cos\theta)\, d\theta$$

$$= b^2 \int_{\pi}^{0} \ln (x-b\cos\theta)\, d\theta - \int_{\pi}^{0} b^2 \cos^2\theta \ln (x-b\cos\theta)\, d\theta$$

$$= - b^2\pi \ln \left[\frac{x + \sqrt{x^2 - b^2}}{2}\right] - I_{30} \qquad \text{(I.23)}$$

Taking the second term and substituting:

$$u = x-x_o \qquad du = -dx_o$$

$$x_o = x-u \qquad dx_o = -du$$

gives $$\int_{-b}^{b} \frac{x_o\sqrt{b^2 - x_o^2}}{x - x_o}\, dx_o$$

$$= \int_{x+b}^{x-b} \frac{(x-u)\sqrt{(b^2-x^2) + (2x)u + (-1)u^2}}{u}\, du$$

$$= \int_{x+b}^{x-b} \frac{x\sqrt{(b^2-x^2) + (2x)u + (-1)u^2}}{u}\, du$$

$$- \int_{x+b}^{x-b} \sqrt{(b^2-x^2) + (2x)u + (-1)u^2}\, du$$

$$= \pi [x\sqrt{x^2 - b^2} - x^2 - b^2/2] \qquad \text{(I.24)}$$

Combining the results (I.23) and (I.24) gives

$$I_{30} = - b^2\pi \ln \left[\frac{x + \sqrt{x^2-b^2}}{2}\right] - I_{30} + \pi \left[x\sqrt{x^2-b^2} - x^2 - \frac{b^2}{2}\right]$$

$$2\, I_{30} = - b^2\pi \ln \left[\frac{x + \sqrt{x^2-b^2}}{2}\right] + \pi \left[x\sqrt{x^2-b^2} - x^2 - \frac{b^2}{2}\right]$$

Thus $$I_{30} = \frac{\pi}{2}\left\{b^2 \ln\left[\frac{2}{x + \sqrt{x^2-b^2}}\right] + x\sqrt{x^2-b^2} - x^2 - \frac{b^2}{2}\right\}$$

for $x > b$ (I.25)

(v) $x < -b$; considering the region where $x < -b$

$$I_{30} = -\int_{-b}^{b} \frac{x_o^2 \ln|x-x_o|}{\sqrt{b^2 - x_o^2}}\, dx_o$$

$$= -\int_{-b}^{b} \frac{x_o^2 \ln|(-1)(|x|+x_o)|}{\sqrt{b^2 - x_o^2}}\, dx_o$$

$$= -\int_{-b}^{b} \frac{x_o^2 \ln(|x|+x_o)}{\sqrt{b^2 - x_o^2}}\, dx_o$$

transferring $x_o \to -x_o$

$$= +\int_{b}^{-b} \frac{x_o^2 \ln(|x|-x_o)}{\sqrt{b^2 - x_o^2}}\, dx_o$$

and reversing order of integration

$$= -\int_{-b}^{b} \frac{x_o^2 \ln(|x|-x_o)}{\sqrt{b^2 - x_o^2}}\, dx_o$$

Thus from case (iv):

$$I_{30} = \frac{\pi}{2}\left\{b^2 \ln\left[\frac{2}{|x| + \sqrt{x^2-b^2}}\right] + |x|\sqrt{x^2-b^2} - x^2 - \frac{b^2}{2}\right\}$$

for $x < -b$ (I.26)

I.5 Summary

Of course all the above integrals must have a reference constant such that $I_{10} = I_{20} = I_{30} = 0$ at $|x| = \pm\infty$. This is found to be the value of x thus the normalized solutions of the integrals are:

$$I_{10} = \pi \ln(2) \qquad \text{for } x = 0$$

$$I_{10} = \pi \ln\left[\frac{2}{1 + \sqrt{1-(b/x)^2}}\right] \qquad \text{for } |x| \geq +b$$

$$I_{20} = 0 \qquad \text{for } x = 0$$

$$I_{20} = -\pi\left(\sqrt{1 - (b/x)^2} - 1\right) \qquad \text{for } x > +b$$

$$I_{20} = +\pi\left(\sqrt{1 - (b/x)^2} - 1\right) \qquad \text{for } x < -b$$

$$I_{30} = \frac{\pi}{2}\left[\ln(2) - \frac{1}{2}\right] \qquad \text{for } x = 0$$

$$I_{30} = \frac{\pi}{2}\left\{\left(\frac{b}{x}\right)^2 \ln\left[\frac{2}{1 + \sqrt{1-(b/x)^2}}\right] + \sqrt{1-(b/x)^2} - 1 - \frac{1}{2}\left(\frac{b}{x}\right)^2\right\}$$

for $|x| \geq b$

APPENDIX II

SOLUTION OF WAVE EQUATION INTEGRALS

II.1 General

To integral equations of Chapter 4 to be solved are:

$$I_{10} = \int_{-b}^{b} \frac{e^{j k x_o}}{\sqrt{b^2 - x_o^2}} dx_o$$

$$I_{20} = \int_{-b}^{b} \frac{x_o\, e^{j k x_o}}{\sqrt{b^2 - x_o^2}} dx_o$$

$$I_{30} = \int_{-b}^{b} \frac{x_o^2\, e^{j k x_o}}{\sqrt{b^2 - x_o^2}} dx_o \qquad \text{(II.1)}$$

II.2 Solution to I_{10}

$$I_{10} = - \int_{-b}^{b} \frac{e^{j k_x x_o}}{\sqrt{b^2 - x_o^2}} dx_o$$

putting $x_o = b\cos\theta$

$$\sqrt{b^2 - x_o^2} = b\sin\theta$$

$$dx_o = -b\sin\theta\, d\theta \qquad \text{(II.2)}$$

gives: $$I_{10} = \int_0^{\pi} e^{jkb\cos\theta}\, d\theta$$

$$= \int_0^{\pi} \left[1 + j\,k\,b\cos\theta - \frac{(kb)^2}{2!} - j\,\frac{(kb)^3}{3!}\cos^3\theta + \frac{(kb)^4}{4!}\cos^4\theta + j\,\frac{(kb)^5}{5!}\cos^5\theta \ldots\right] d\theta$$

$$= \int_0^{\pi} \left[1 - \frac{(kb)^2}{2!}\cos^2\theta + \frac{(kb)^4}{4!}\cos^4\theta \ldots\right] d\theta$$

$$+ j\,k\,b \int_0^{\pi} \left[\cos\theta - \frac{(kb)^2}{3!}\cos^3\theta + \frac{(kb)^4}{5!}\cos^5\theta \ldots\right] d\theta$$

$$= \left[\pi - \frac{(kb)^2}{2!}\left[\frac{1}{2}\theta + \frac{1}{4}\sin 2\theta\right]\right]_0^{\pi} + \frac{(kb)^4}{4!}\left[\frac{3\theta}{8} + \frac{\sin 2\theta}{4} + \frac{\sin 4\theta}{32}\right]_0^{\pi} \ldots$$

$$+ j\,k\,b\left[\sin\theta\Big|_0^{\pi} - \frac{(kb)^2}{3!}(\sin\theta)(\cos^2\theta+2)\Big|_0^{\pi} + \frac{(kb)^4}{5!}\left(\frac{1}{5}\cos^4\theta\,\sin\theta\right)\Big|_0^{\pi}\right.$$

$$\left. + \frac{4}{5}\int_0^{\pi}\cos^2\theta\, d\theta \ldots\right]$$

$$= \pi\left[1 - \frac{(kb)^2.1}{2.2} + \frac{(kb)^4.3}{4.3.2.2.4} - \frac{(kb)^6.5.3}{6.5.4.3.2.2.4.6} \cdots\right]$$

$$+ j\,k\,b\,[0]$$

$$= \pi\left[1 - \frac{(kb)^2}{2^2} + \frac{(kb)^4}{2^2.4^2} - \frac{(kb)^6}{2^2.4^2.6^2} \cdots\right]$$

$$= \pi J_o (k b)$$

Reference [24]

Thus $I_{10} = \pi J_o (k b)$ is the solution to the first integral.

II.3 Solution to I_{20}

Using tranformations (II.2):

$$I_{20} = \int_0^{\pi} b\cos\theta \, e^{jkb\cos\theta}$$

$$= \int_0^{\pi} b\cos\theta \left[1 + j k b\cos\theta - \frac{(kb)^2\cos^2\theta}{2!} - j\frac{(kb)^2\cos^3\theta}{3!} \right.$$

$$\left. + \frac{(kb)^4\cos^4\theta}{4!} + j\frac{(kb)^5\cos^5\theta}{5!} \ldots \right] d\theta$$

$$= \frac{1}{k}\int_0^{\pi} \left[k b\cos\theta - \frac{(kb)^3\cos^3\theta}{2!} + \frac{(kb)^5\cos^5\theta}{4!} \ldots \right] d\theta$$

$$+ \frac{1}{k}\int_0^{\pi} \left[j (kb)^2\cos^2\theta - j\frac{(kb)^4}{3!}\cos^4\theta + j\frac{(kb)^6}{5!}\cos^6\theta \ldots \right] d\theta$$

$$= \frac{1}{k}[0] + \frac{jkb}{k}\left[kb\left(\frac{1}{2}\theta + \frac{1}{4}\sin 2\theta\right)\Big|_0^{\pi} - \frac{(kb)^3}{3!}\left(\frac{3\theta}{8} + \frac{\sin 2\theta}{4} + \frac{\sin 4\theta}{32}\right)\Big|_0^{\pi}\right.$$

$$\left. + \frac{(kb)^5}{5!}\left(\frac{1}{6}\cos^5\theta\sin\theta\right)\Big|_0^{\pi} + \frac{5}{6}\int_0^{\pi}\cos^4\theta\, d\theta\right) + \ldots \Big]$$

$$= j b \left[\frac{kb\pi}{2} - \frac{(kb)^3}{3!}\left(\frac{3\pi}{8}\right) + \frac{(kb)^5}{5!}\left(\frac{5}{6}\right)\left(\frac{3\pi}{8}\right) \ldots \right]$$

$$= j\pi b \left[\frac{kb}{2} - \frac{(kb)^3}{2^2.4} + \frac{(kb)^5}{2^2.4^2.6} + \ldots \right]$$

$$= j \pi b J_1(kb)$$

Reference [24]

Thus the solution to the second integral is

$$I_{20} = j \pi b J_1(kb)$$

III.4 Solution to I_{30}

Using the tranformation (II.2)

$$I_{30} = \int_0^{\pi} b^2 \cos^2\theta \, e^{jkb\cos\theta} \, d\theta$$

$$= \int_0^{\pi} b^2 (1 - \sin^2\theta) \, e^{jkb\cos\theta} \, d\theta$$

$$= b^2 \int_0^{\pi} e^{jkb\cos\theta} \, d\theta - \int_0^{\pi} b^2 \sin^2\theta \, e^{jkb\cos\theta} \, d\theta$$

$$= b^2 \pi J_o (k b) + \frac{1}{kb} \int_0^{\pi} [b\sin\theta] [-j k b\sin\theta \, e^{jkb\cos\theta}] d\theta$$

$$= b^2 \pi J_o (k b) + \frac{1}{kb} [b\sin\theta \, e^{jkb\cos\theta}\Big|_o^{\pi} - \int_0^{\pi} b\cos\theta \, e^{jkb\cos\theta} \, d\theta]$$

$$= b^2 \pi J_o (k b) - \frac{\pi b}{k} J_1 (k b)$$

Thus the solution to the third integral is:

$$I_{30} = b^2 \pi [J_o (k b) - \frac{1}{kb} J_1 (k b)]$$

II.5 Summary

The solution to the wave equation integrals in the far field (i.e. outside the transducer region) is:

$$I_{10} = \pi J_o (k\ b)$$

$$I_{20} = j \pi b J_1 (k\ b)$$

$$I_{30} = b^2 \pi [J_o (k\ b) - \frac{1}{kb} J_1 (k\ b)]$$

where: J_o is the Bessel function of the first kind

J_1 is the Bessel function of the second kind

k is the propagation constant

b is 1/2 the finger width

APPENDIX III

SAWINIT

The following is a listing of the initial programme SAWINT. It is written in FORTRAN IV and operates under the NOS system for CDC-6000 computer systems. System libraries used are:

(1) SSPLIB - scientific subroutines package

(2) PLOTVER - Versatec plotting library package.

The files that are saved are: TAPE 1 and TAPE 2.

```
SAWINIT. THIS PROGRAM DOES 1ST PASS OF SAW DESIGN.    SUTHERS
USER([redacted],[redacted])
CHARGE([redacted])
FTN.
SAVE,INPUT=SAWINIT.
      PROGRAM TST (INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT,TAPE1,TAPE2)


      COMPLEX FFTS,V,RIDT,AR
      REAL MAG
      REAL LEN
      REAL MDB, PI, IMAG, LENGTH ,RELBP, IMAGL
      DIMENSION AR(513)
      DIMENSION V(500),XP(500),B(500),TIM(500),LEN(500),FRSP(30,3),FREQ(
     C513),MDB(513),PHASE(513),X(513),Y(513),Z(513),RMDB(513),RPHS(513),
     CFFTS(513),RIDT(513),MAG(513),PHSE(513)
      DIMENSION C0(500),C1(500),C2(500)
      CALL      INPUT(N,NP,FRSP,RLBW,FREQ,MDB,PHASE,IG,FO,IC,ID)
      T=N/(FO*RLBW)
C     T IS THE PERIOD OF THE IDT IN MICROSECONDS.
      DT=T/N
C     DT IS THE TIME INTERVAL OF THE FFT SAMPLE POINTS.
      SPBD =FO*RLBW
      NZ = N+1
      NH=N/2
      NM=NH+1
      PI = ACOS (-1.0)
      IR=0
      DO 10 I=1,NZ
      IF (MDB(I).EQ.0.0.AND.IR.EQ.0) IR=I
      MAG(I)=MDB(I)
      AMP=10.0**(MDB(I)/20.0)
      PHSE(I)=PHASE(I)
      CALL RECT(AMP,PHSE(I),RR,RI)
      FFTS(I)=CMPLX(RR,RI)
10    CONTINUE
```

```
      IR=IR+3
      CALL PLTFRQ(N,FREQ,MAG,PHSE)
      S=N
C     INITIALIZATION IS COMPLETE


      CALL SAWFFT(FFTS,N,IG,X,Y,Z)
      X(NZ)=6.0
      Y(NZ)=Y(1)
      Z(NZ)=Z(1)
      CALL PLTFFT(NZ,X,Y,Z)
      FC=FO-SPBD/2.0
      DO 95 I=1,NH
      Z(NM+I)=Z(NM+I)-I*360.0*DT*(FO-FC)
      Z(NM-I)=Z(NM-I)+I*360.0*DT*(FO-FC)
95    CONTINUE
      CALL PHSSTR(NZ,Y,Z)
      CALL I0(8.0,DEN)
      DO 100 I=1,NH
      TM=FLOAT(NH**2-I**2)
      ARG=(16.0/N)*SQRT(TM)
      CALL I0(ARG,ANUM)
      WT=ANUM/DEN
      Y(NH+1+I)=Y(NH+1+I)*WT
      Y(NH+1-I)=Y(NH+1-I)*WT
100   CONTINUE
      CALL IDT(NZ,Y,Z,FO,SPBD,LEN,TIM,L)
      F1=FREQ(NH-2)
      F2=FREQ(NH+2)
      VEL=3410
      CALL MODEL(L ,LEN,TIM,B,XP,T,F1,F2,VEL,C0,C1,C2,  AR)
      CALL FRQRSP(L,AR,TIM,FREQ,RMDB,RPHS,NZ,T,RIDT)

      WRITE (1,20) NZ,L,IG,T,DT,SPBD,FO,FC,VEL
      RMAX=RMDB(IR)
      DO 50 I=1,NZ
```

```
      RMDB(I)=RMDB(I)-RMAX
      WRITE (1,30) FREQ(I),MDB(I),PHASE(I),RMDB(I),RPHS(I),MAG(I),PHSE(
     +I)
      IF (RMDB(I).LT.-100.0) RMDB(I)=-100.0
50    CONTINUE
20    FORMAT (3I5,6E10.5)
30    FORMAT (7E11.5)
      WRITE (2,81) L
      DO 70 I=1,L
      XL=XP(I)-B(I)
      XR=XP(I)+B(I)
      WRITE (2,80) LEN(I),XL,XR
70    CONTINUE
80    FORMAT (3E20.10)
81    FORMAT (I5)

      WRITE (6,152)
      DO 150 I=1,N
      WRITE (6,151) FREQ(I),MAG(I),MDB(I),RMDB(I),PHSE(I),PHASE(I),
     +RPHS(I)
150   CONTINUE
151   FORMAT (1H0,5X,F8.2,* MHZ*,F9.2,* DB*,F9.2,* DB*,F9.2,* DB*,F8.2,
     +* DEG*,F8.2,* DEG*,F8.2,* DEG*)
152   FORMAT (1H1,9X,*FREQ*,9X,*MAG*,9X,*MDB*,8X,*RMDB*,8X,*PHSE*,7X,
     +*PHASE*,8X,*RPHS*)

      CALL PLTFRQ(N,FREQ,RMDB,RPHS)
      CALL PLOT (X(N),Y(N),999)
      STOP
      END


      SUBROUTINE INPUT(N,NP,FRSP,RLBW,FREQ,MDB,PHASE,IG,FO,IC,ID)

C     THE INPUT SUBROUTINE READS IN THE SPECIFICATIONS IN THE FREQUENCY
C     DOMAIN AND INITIALIZES THE INPUT VECTOR FOR THE FFT.
```

```
C     N IS THE NUMBER OF  POINTS FOR THE FFT WHERE N=2**IG.-OUTPUT
C     NP IS THE NUMBER OF SPECIFIED FREQUENCY POINTS.-INPUT
C     FRSP IS A 3 X NP ARRAY SPECIFICATION OF INPUT FREQUENCIES.-INPUT
C     SPBD IS THE SPECIFIED BAND WIDTH.
C     FREQ IS THE VECTOR OF CALCULATED FREQUENCY POINTS FOR FFT.-OUTPUT
C     MDB IS THE VECTOR OF CALCULATED AMPLITUDES FOR FFT.-OUTPUT
C     PHASE IS THE VECTOR OF PHASE SPECIFICATIONS FOR FFT.-OUTPUT
C     IG IS THE POWER OF 2 SUCH THAT N=2**IG.-INPUT,OUTPUT
      DIMENSION FRSP(30,3),FREQ( 513),MDB( 513),PHASE( 513)
      REAL MDB
      READ (5,10) IG,NP,RLBW ,FO
10    FORMAT (2I10,2F10.5)
C     FO IS DESIGNATED AS CENTRE FREQUENCY.
      SPBD=FO*RLBW
C     SPBD IS THE SPECIFIED BAND OF FREQUENCIES.
      DO 31 I=1,NP
      READ (5,30) FRSP(I,1),FRSP(I,2),FRSP(I,3)
31    CONTINUE
30    FORMAT (3F10.2)
C     FRSP(I,1) IS FREQUENCY IN MHZ.
C     FRSP(I,2) DB OF INSERTION LOSS
C     FRSP(I,3) IS PHASE SPECIFICATIONS IN DEGREES.
      N=2**IG
      DF=SPBD/N
C     DF IS FREQUENCY INCREMENT BETWEEN POINTS
      K=2
C     FO IS THE CENTRE OF THE SPECIFIED BANDPASS
C     CALCULATE FREQUENCY POINTS.
      NZ=N+1
      NH=N/2
      NL=NH+1
      FREQ(NL)=FO
      DO 49 I=1,NH
      FREQ(NL-I)=FO-I*DF
      FREQ(NL+I)=FO+I*DF
49    CONTINUE
```

```
C     ENTER AMPLITUDE AND PHASE SPECIFICATIONS FOR EACH FREQUENCY.
      DO 50 I=1,NZ
      IF (FREQ(I).LT.FRSP(1,1)) MDB(I)=-FRSP(1,2)
      IF (FREQ(I).GT.FRSP(NP,1)) MDB(I)=-FRSP(NP,2)
      PHASE(I)=0.0
53    IF (K.GT.NP) GO TO 50
      IF (FREQ(I).LT.FRSP(K-1,1)) GO TO 50
      IF (FREQ(I).GT.FRSP(K,1)) GO TO 54
      S=(FREQ(I)-FRSP(K-1,1))/(FRSP(K,1)-FRSP(K-1,1))
      MDB(I)=FRSP(K-1,2)+(FRSP(K,2)-FRSP(K-1,2))*S
      MDB(I) = -MDB(I)
      PHASE(I)=FRSP(K-1,3)+(FRSP(K,3)-FRSP(K-1,3))*S
      GO TO 50
54    K=K+1
      GO TO 53
50    CONTINUE
      FC =(FRSP(NP,1)+FRSP(1,1))/2.0
      DO 60 I=1,N
      IF (FREQ(I)-FC) 60,61,62
60    CONTINUE
62    IF ((FREQ(I)-FC).LT.(FC-FREQ(I-1))) GO TO 61
      GO TO 63
61    FC=FREQ(I)
      IC=I
      GO TO 65
63    FC=FREQ(I-1)
      IC=I-1
65    ID=IC* ((FRSP(NP,1)-FRSP(1,1))/FC)
      RETURN
      END


      SUBROUTINE SAWFFT(FFT1,N,IG,X,Y,Z)

      COMPLEX FFT1,FFT2,W,WK,WP
      REAL IMAG
```

```
      DIMENSION FFT1(N),IFFT(513,2),FFT2(513),X(N),Y(N),Z(N)
      PI=ACOS(-1.0)
      L=N
C     FAST FOURIER TRANSFORM FOR SAW DESIGN
C     FREQ IS ARRAY OF FREQUENCIES OF  POINTS OF FFT
C     MDB IS ARRAY OF DB OF INSERTION LOSS FOR EACH FREQUENCY
C     PHASE IS ARRAY OF PHASE,S FOR EACH FREQUENCY
C     IFFT IS ARRAY KEEPING TRACK OF EXPONENTS FOR MULTIPLIERS W(J) IN
C     THE FFT PROCESS
C     FFT1 AND FFT2 ARE COMPLEX VECTORS OF FFT PROCESS
      DO 10 I=1,513
      DO 10 J=1,2
10    IFFT(I,J) = 0
C     CALCULATE THE EXPONENT FOR THE MULTIPLIER FOR THE FFT PROCESS
      AREAL = COS(2.*PI/N)
      IMAG = SIN(2.*PI/N)
      W = CMPLX(AREAL,IMAG)
      DO 70 M=1,IG
      L=L/2
      NH = N/2
      NM=NH+1
      DO 50 K=1,NH
      IFFT(K,2) = IFFT(2*K,1)
50    IFFT(K+NH,2)=IFFT(K,2) +L
C     CALCULATE NEXT VECTOR IN FFT PROCESS
      DO 60 K=1,N
      WK = W**IFFT(K,2)
      IF (IFFT(K,2).GT.IFFT(K,1)) GO TO 61
      WP = WK*FFT1(K+L)
      FFT2(K) = FFT1(K)+WP
      GO TO 60
61    WP = WK*FFT1(K)
      FFT2(K) = FFT1(K-L) + WP
60    CONTINUE
      DO 70 I=1,N
      FFT1(I) = FFT2(I)
```

```
      IFFT(I,1)=IFFT(I,2)
70    CONTINUE
C     PUT OUTPUT VECTOR (DFT) IN CORRECT ORDER.
      AMAX=0.0
      DO 82 K=1,N
      KI=0
81    KI=KI+1
      IF (IFFT(KI,1).EQ.K-1) GO TO 80
      GO TO 81
80    RR=REAL(FFT1(KI))
      RI=AIMAG(FFT1(KI))
      CALL POLAR (RR,RI,PL,PA)
      Y(K)=PL
      Z(K)=PA
      IF (Y(K).GT.AMAX) AMAX=Y(K)
      X(K)=5.0*IFFT(KI,1)/N
82    CONTINUE
      DO 83 I=1,N
      Y(I)=Y(I)/AMAX
83    CONTINUE
      DO 93 K = 1,NH
      YS = Y(K)
      Y(K)=Y(K+NH)
      Y(K+NH)=YS
      ZS = Z(K)
      Z(K) = Z(K+NH)
      Z(K+NH) = ZS
93    CONTINUE
      RETURN
      END


      SUBROUTINE IDT(NZ,Y,Z,FO,SPBD,LEN,TIM,LH)

      REAL LEN
      DIMENSION Y(NZ),Z(NZ),LEN(500),TIM(500)
```

```
      NH=NZ/2
      NM=NH+1
      N=NZ-1
      T=N/SPBD
      DT=T/N
C     T IS THE IDT PERIOD IN MICRO-SECONDS
C     DT IS THE TIME INCREMENT OF THE FFT SAMPLES.
C     ADD IN THE LINEAR PHASE COMPONENTS OF THE FREQUENCY SHIFT.
      ZM=Z(NM)
      DO 40 I=1,NH
      Z(NM+I)=Z(NM+I)-ZM+(I*DT*FO*360.0)
      Z(NM-I)=Z(NM-I)-ZM-(I*DT*FO*360.0)
40    CONTINUE
      L=0
      M=0
      LH=1
      LEN(250)=Y(NM)
      TIM(250)=0.0
      DO 10 I=1,NH
      IF (L.EQ.249) GO TO 21
C     NP GIVES THE NUMBER OF FINGERS BETWEEN FFT POINTS.
      NC=(+Z(NM-1+I)+0.5)/180.0
      PHASL=180.0*NC
      NP=(+Z(NM+I)-PHASL+0.5)/180.0
      IF (NP.EQ.0) GO TO 21
C     OBTAIN TIMES,LENGTHS AND POLARITIES OF FINGERS.
      DO 20 K=1,NP
      L=L+1
      LH=LH+1
      PH=(180.0*(NC+K)-Z(NM-1+I))/(Z(NM+I)-Z(NM-1+I))
      TIM(250+L)=(I-1+PH)*DT
      SN=(-1.0)**(NC+K)
      LEN(250+L)=SN*(Y(NM-1+I)+(Y(NM+I)-Y(NM-1+I))*PH)
20    CONTINUE
21    IF (M.EQ.249) GO TO 10
C     NP GIVES THE NUMBER OF FINGERS BETWEEN FFT POINTS.
```

```
      NC=(-Z(NM+1-I)+0.5)/180.0
      PHASL=180.0*NC
      NP=(-Z(NM-I)-PHASL+0.5)/180.0
      IF (NP.EQ.0) GO TO 10
C     OBTAIN TIMES,LENGTHS AND POLARITIES OF FINGERS.
      DO 30 K=1,NP
      M=M+1
      LH=LH+1
      PH=(180.0*(NC+K)+Z(NM+1-I))/(-Z(NM-I)+Z(NM+1-I))
      TIM(250-M)=-(I-1+PH)*DT
      SN=(-1.0)**(NC+K)
      LEN(250-M)=SN*(Y(NM+1-I)+(Y(NM-I)-Y(NM+1-I))*PH)
30    CONTINUE
10    CONTINUE
      MN=250-M-1
      DO 50 I=1,LH
      LEN(I)=LEN(MN+I)
      TIM(I)=TIM(MN+I)
50    CONTINUE
49    RETURN
      END


      SUBROUTINE FRQRSP(L,V,TIM,FREQ,RMDB,RPHS,NZ,T,RIDT)

      REAL LEN
      COMPLEX RIDT
      COMPLEX DEL,V
      DIMENSION V(L),TIM(L),FREQ(NZ),RMDB(NZ),RPHS(NZ),RIDT(NZ)
      N=NZ-1
      NM=N/2+1
C     L IS THE NUMBER OF FINGERS IN THE IDT
C     LEN IS THE LENGTH OF FINGERS + POLARITY
C     TIM IS POSITION IN THE TIME DOMAIN OF FINGERS IN MICRO-SECONDS
C     FREQ IS ARRAY OF FREQUENCY POINTS FOR FFT
C     RMDB IS AMPLITUDE RESPONSE IN DB
```

```
C     RPHS IS THE PHASE RESPONSE IN DEGREES
C     NZ IS NUMBER OF SPECIFIED POINTS TO BE EVALUATED
      PI=ACOS(-1.0)
      RMAX=0.0
C     DO FAR FIELD INTEGRATION
      DO 10 I=1,NZ
      RIDT(I)=CMPLX(0.0,0.0)
      DO 20 J=1,L
      ANG=2*PI*FREQ(I)*(-TIM(J)-T/2.0)
      DEL=CMPLX(COS(ANG),SIN(ANG))
      RIDT(I)=RIDT(I)+DEL*V(J)
20    CONTINUE
      RSRL=REAL(RIDT(I))
      RSIM=AIMAG(RIDT(I))
      RMDB(I)=SQRT(RSRL**2+RSIM**2)
      RPHS(I)=(ATAN(RSIM/RSRL))*(180.0/PI)
      IF (RSRL.LT.0.0.AND.RSIM.LT.0.0) RPHS=RPHS-180.0
      IF (RSRL.GT.0.0.AND.RSIM.LT.0.0) RPHS=RPHS+180.0
10    CONTINUE
      RMAX=RMDB(NM)
C     OBTAIN AMPLITUDE IN DB
      DO 30 I=1,NZ
      RMDB(I)=20*ALOG10(RMDB(I)/RMAX)
      IF (RMDB(I).LT.-100.0) RMDB(I)=-100.0
      RPHS(I)=RPHS(I)-RPHS(NM)
30    CONTINUE
21    FORMAT (1H0,5X,*FREQ =*,F5.1,* MHZ*,5X,*MDB =*,F10.3,* DB*,5X,
     C*PHS =*,F10.3,* DEG*)
      RETURN
      END


      SUBROUTINE MODEL(LZ,LEN,TIM,B,X,T,F1,F2,VEL,C0,C1,C2,  AR)

      COMPLEX AR
      REAL LEN,I1,I2,I3
```

```
      DIMENSION LEN(LZ),TIM(LZ),B(LZ),X(LZ),C0(LZ),C1(LZ),C2(LZ),
     CV(500,3),AR(LZ)
      PI=ACOS(-1.0)
      AL=ALOG(2.0)
      L=LZ-1
      VMAX=0.0
      DO 10 I=1,LZ
      IF (VMAX.LT.ABS(LEN(I))) VMAX=ABS(LEN(I))
      IF (VMAX.EQ.ABS(LEN(I))) IMAX=I
      X(I)=TIM(I)*VEL
      V(I,1)=0.0
      V(I,2)=0.0
      V(I,3)=0.0
10    CONTINUE
      B(1)=(X(2)-X(1))/4.0
      DO 20 I=2,L
      B(I)=(X(I+1)-X(I-1))/8.0
20    CONTINUE
      B(LZ)=(X(LZ)-X(L))/4.0

C     FINGER WIDTH AND POSITIONS HAVE BEEN FOUND.

      LH=LZ/2
      DO 30 I=1,LZ
      C0(I)=LEN(I)
      C1(I)=0.0
      C2(I)=0.0
30    CONTINUE

      CALL ELPOT(LZ,X,B,C0,C1,C2,V(IMAX,2),IMAX,2)
      FTR=ABS(LEN(IMAX)/V(IMAX,2))
      DO 41 I=1,LZ
      C0(I)=C0(I)*FTR
41    CONTINUE

      DO 42 I=1,LZ
```

```
      CALL ELPOT(LZ,X,B,C0,C1,C2,V(I,1),I,1)
      CALL ELPOT(LZ,X,B,C0,C1,C2,V(I,2),I,2)
      CALL ELPOT(LZ,X,B,C0,C1,C2,V(I,3),I,3)
42    CONTINUE
      BI30=0.5*ALOG(2.0)-0.75
      ORC=1.5

      DO 55 M=1,3

39    DO 43 I=1,LZ
      C0(I)=C0(I)+ORC*(LEN(I)-V(I,2))/AL
      C1(I)=C1(I)+ORC*(V(I,3)-V(I,1))/(2.0*B(I))
      C2(I)=C2(I)+ORC*(V(I,2)-(V(I,1)+V(I,3))/2.0)/BI30
43    CONTINUE


      CALL ELPOT(LZ,X,B,C0,C1,C2,V(IMAX,2),IMAX,2)
      FTR=ABS(LEN(IMAX)/V(IMAX,2))
      DO 50 K=1,LZ
      C0(K)=C0(K)*FTR
      C1(K)=C1(K)*FTR
      C2(K)=C2(K)*FTR
50    CONTINUE

      DO 51 I=1,LZ
      CALL ELPOT(LZ,X,B,C0,C1,C2,V(I,1),I,1)
      CALL ELPOT(LZ,X,B,C0,C1,C2,V(I,2),I,2)
      CALL ELPOT(LZ,X,B,C0,C1,C2,V(I,3),I,3)
51    CONTINUE

55    CONTINUE

52    FORMAT (1H0,5X,I5,7E15.4)


C     COEFICIENTS HAVE BEEN FOUND AND PRINTED.
```

```
      W1=SQRT(2.0*PI*F1)
      DO 70 I=1,LZ
      X1=2.0*PI*F1*B(I)/VEL
      CALL BESJ(X1,0,BJ01,0.0001,IER)
      CALL BESJ(X1,1,BJ11,0.0001,IER)
      ARR=W1*C0(I)*BJ01+C2(I)*W1*B(I)**2*(BJ01-BJ11/X1)
      ARI=W1*C1(I)*B(I)*BJ11
      AR(I)=CMPLX(ARR,ARI)
70    CONTINUE

C     ACOUSTIC RESPONSE FOR EACH FINGER HAS BEEN FOUND.

      RETURN
      END


      SUBROUTINE ELPOT(LZ,X,B,C0,C1,C2,VK,IK,IN)

      REAL I1,I2,I3
      DIMENSION X(LZ),B(LZ),C0(LZ),C1(LZ),C2(LZ)
      VK=0.0
      AL=ALOG(2.0)
      DO 41 J=1,LZ
      XR=X(IK)+(IN-2)*B(IK)-X(J)
      IF (XR.EQ.0.0) 43 ,48
48    BX=(B(J)/XR)**2
      ARG=1.0-BX
      IF(ARG.LE.0.0) GO TO 45
47    BS=SQRT(1.0-BX)
40    I1=ALOG(2.0/(1.0+BS))
      I2=BS-1.0
      I3=0.5*(BX*I1+BS-1.0-0.5*BX)
      IF (XR) 42,43,44
41    CONTINUE
      RETURN
```

```
 42     VK   =VK   +C0(J)*I1-C1(J)*I2+C2(J)*I3
        GO TO 41

 43     VK=VK+C0(J)*AL
        GO TO 41.

 44     VK =  VK +C0(J)*I1+C1(J)*I2+C2(J)*I3
        GO TO 41
 50     FORMAT (1H0,5X,I5,7E15.5)
 51     FORMAT (1H0,5X,I5,3E15.5)
 45     BS=0.0
        GO TO 40

        END


        SUBROUTINE PHSCPS(N,Z)

        DIMENSION Z(N)
        DO 10 K=1,N
        GO TO 11
 12     Z(K)=Z(K)-360.0
 11     IF (Z(K).GT.+181.0) GO TO 12
        GO TO 13
 14     Z(K)=Z(K)+360.0
 13      IF (Z(K).LE.-179.0) GO TO 14
 10     CONTINUE
        RETURN
        END


      SUBROUTINE PHSSTR(NZ,Y,Z)

      DIMENSION Y(NZ),Z(NZ)
      NH=NZ/2
```

```
      NM=NH+1
      DO 10 I=1,NH
14    DPHS=Z(NM+I)-Z(NM-1+I)
      IF (DPHS.GT.+90.0) GO TO 11
      IF (DPHS.LT.-90.0) GO TO 12
      GO TO 13
11    Z(NM+I)=Z(NM+I)-180.0
      Y(NM+I)=-Y(NM+I)
      GO TO 14
12    Z(NM+I)=Z(NM+I)+180.0
      Y(NM+I)=-Y(NM+I)
      GO TO 14
13    DPHS=Z(NM-I)-Z(NM+1-I)
      IF (DPHS.GT.+90.0) GO TO 15
      IF (DPHS.LT.-90.0) GO TO 16
      GO TO 10
15    Z(NM-I)=Z(NM-I)-180.0
      Y(NM-I)=-Y(NM-I)
      GO TO 13
16    Z(NM-I)=Z(NM-I)+180.0
      Y(NM-I)=-Y(NM-I)
      GO TO 13
10    CONTINUE
      RETURN
      END


      SUBROUTINE RECT(PL,PA,RR,RI)

C     RECT CONVERTS (PL,PA) TO (RR,RI)
C     PL IS THE POLAR CO-ORDINATE LENGTH.
C     PA IS THE POLAR CO-ORDINATE ANGLE IN DEGREES.
C     RR IS THE RECTANGULAR CO-ORDINATE REAL PART.
C     RI IS THE RECTANGULAR CO-ORDINATE IMAGINARY PART.
      PI =ACOS(-1.0)
      RI=PL*SIN(PI*PA/180.0)
```

```
      RR=PL*COS(PI*PA/180.0)
      RETURN
      END

      SUBROUTINE POLAR(RR,RI,PL,PA)
C     POLAR CONVERTS (RR,RI)TO (PL,PA)
C     RR IS THE RECTANGULAR CO-ORDINATE REAL PART.
C     RI IS THE RECTANGULAR CO-ORDINATE IMAGINARY PART.
C     PL IS THE POLAR CO-ORDINATE LENGTH.
C     PA IS THE POLAR CO-ORDINATE ANGLE IN DEGREES.
      PI=ACOS(-1.0)
      PL=SQRT(RR**2+RI**2)
      IF (RR.EQ.0.0) GO TO 10
      PA =ATAN(RI/RR)
      IF (RR.LE.0.0.AND.RI.GE.0.0) PA =PA+PI
      IF (RR.LE.0.0.AND.RI.LE.0.0) PA =PA-PI
      PA =PA*180.0/PI
      GO TO 15
10    IF (RI) 11,12,13
11    PA=-90.0
      PL=RI
      GO TO 15
12    PA=0.0
      PL=0.0
      GO TO 15
13    PA=+90.0
      PL=RI
15    RETURN
      END

      SUBROUTINE LBLF(YL)
      CALL LETTER (19,.1,.0,1.5,7.50,19HPHASE SPECIFICATION)
      CALL LETTER (23,.1,.0,1.5,1.2 ,23HAMPLITUDE SPECIFICATION)
```

```
      YL=5.75
      NA = 00550060006001440102B
      DO 10 I=1,10
      NA=NA+1000000000000B
      YL = YL-0.4
10    CONTINUE
      CALL LBPH(9.75)
      RETURN
      END


      SUBROUTINE LBLT(YL)
      CALL LETTER (15,.1   ,0.0 ,1.5,1.25,15HIDT APODIZATION)
      CALL LETTER (31,.1   ,0.0 ,1.5,7.5,31HIDT DEVIATION FROM LINEAR PHA
     CSE).
      NA = 006000560071B
      YL = 5.75
      DO 10 I=1,10
      YL = YL - 0.4
      NA = NA - 0001B
10    CONTINUE
      CALL LBPH(9.75)
      RETURN
      END


      SUBROUTINE LBPH(YPH)
      DIMENSION Y(5), NB(5)
      NB(1) = 00550061007000600040B
      NB(2) = 0055007100600040B
      NB(3) = 00600040B
      NB(4) = 0053007100600040B
      NB(5) = 00530061007000600040B
      Y(1) = 7.75
      Y(2) = 8.25
      Y(3) = 8.75
```

```
      Y(4) = 9.25
      Y(5) = 9.75
      NC=0060B
      DO 20 K=1,5
      YB=Y(K)+0.1
20    CONTINUE
      RETURN
      END

      SUBROUTINE PLTB(N)
      N=N
      CALL PLOT (0.0,0.0,3)
      CALL PLOT (0.0,10.5,2)
      CALL PLOT (8.6,0.0,3)
      CALL PLOT (8.6,10.5,2)
      CALL PLOT (1.5,1.75,3)
      CALL PLOT (7.5,1.75,2)
      CALL PLOT (7.5,5.75,2)
      CALL PLOT (1.5,5.75,2)
      CALL PLOT (1.5,1.75,2)
      CALL PLOT (1.5,7.75,3)
      CALL PLOT (7.5,7.75,2)
      CALL PLOT (7.5,9.75,2)
      CALL PLOT (1.5,9.75,2)
      CALL PLOT (1.5,7.75,2)
      CALL PLOT (1.5,8.25,2)
      CALL PLOT (7.5,8.25,2)
      CALL PLOT (7.5,8.75,2)
      CALL PLOT (1.5,8.75,2)
      CALL PLOT (1.5,9.25,2)
      CALL PLOT (7.5,9.25,2)
      DO 200 I = 1,10
      XA = I*0.6+1.5
      CALL PLOT (XA,1.75,3)
      CALL PLOT (XA,1.55,2)
```

```
      CALL PLOT (XA,8.85,3)
      CALL PLOT (XA,8.65,2)
200   CONTINUE
      DO 201 I = 1,10
      YA = I*0.4+1.75
      CALL PLOT (1.5,YA,3)
      CALL PLOT (1.4,YA,2)
201   CONTINUE
      CALL PLOT (1.5,1.75,3)
      RETURN
      END

      SUBROUTINE PLTIDT(TIM,LEN,L)
      DIMENSION TIM(L),LEN(L)
      REAL LEN
      CALL PLOT(0.0,0.0,3)
      CALL PLOT(0.0,10.5,2)
      CALL PLOT(8.5,10.5,3)
      CALL PLOT(8.5,0.0,2)
      CALL PLOT(1.5,2.0,3)
      CALL PLOT(1.5,8.0,2)
      CALL PLOT(2.0,8.0,2)
      CALL PLOT(2.0,2.0,2)
      CALL PLOT(1.5,2.0,2)
      CALL PLOT(7.0,2.0,3)
      CALL PLOT(7.0,8.0,2)
      CALL PLOT(7.5,8.0,2)
      CALL PLOT(7.5,2.0,2)
      CALL PLOT(7.0,2.0,2)
      DO 10 I=1,L
      X=4.5+2.0*LEN(I)
      Y=((TIM(I)-TIM(1))/(TIM(L)-TIM(1)))*6.0+2.0
      IF (LEN(I).LT.0.0) GO TO 11
      CALL PLOT(2.0,Y,3)
      CALL PLOT(X,Y,2)
```

```
      GO TO 10
11    CALL PLOT(7.0,Y,3)
      CALL PLOT(X,Y,2)
10     CONTINUE
      CALL PLOT(12.0,0.0,-3)
      RETURN
      END


      SUBROUTINE PLTFFT(NZ,X,Y,Z)
      DIMENSION X(NZ),Y(NZ),Z(NZ)
      CALL PLTB(128)
      CALL LBLT(5.75)
      N=NZ-1
      M=N-1
      XSCALE=(X(NZ)-X(1))/6.0
      YSCALE=0.5
      ZSCALE=540.0
      XO=-1.5*XSCALE
      YO=-3.75*YSCALE
      ZO=-8.75*ZSCALE
      XMIN=X(1)
      YMIN=-1.0
      ZMIN=-540.0
      XMAX=X(NZ)
      YMAX=1.0
      ZMAX=+540.0
      CALL PLTIN(XSCALE,YSCALE,XO,YO,XMIN,XMAX,YMIN,YMAX)
      DO 10 I=1,N
      CALL PLTLN(X(I),Y(I),X(I+1),Y(I+1))
10    CONTINUE
      CALL PLTIN(XSCALE,ZSCALE,XO,ZO,XMIN,XMAX,ZMIN,ZMAX)
      DO 20 I=1,N
      CALL PLTLN(X(I),Z(I),X(I+1),Z(I+1))
20    CONTINUE
      CALL PLOT(12.0,0.0,-3)
```

```
      RETURN
      END


      SUBROUTINE PLTFRQ(N,FREQ,MDB,PHASE)

      DIMENSION FREQ(N),MDB(N),PHASE(N)
      DF=(FREQ(N)-FREQ(1))/(N-1)
      XSCALE = (FREQ(N)-FREQ(1)-DF)/6.0
      YSCALE = 180.0
      XO = -1.5 * XSCALE+FREQ(1)
      YO = -8.75 * YSCALE
      XMIN = FREQ(1)-DF
      YMIN = -180.1
      XMAX = FREQ(N)
      YMAX = +180.1
      M=N-1
      CALL PLTIN(XSCALE,YSCALE,XO,YO,XMIN,XMAX,YMIN,YMAX)
      DO 60 I=1,M
      CALL PLTLN(FREQ(I),PHASE(I),FREQ(I+1),PHASE(I+1))
60    CONTINUE
      YSCALE = 25.0
      YO = -5.75*YSCALE
      YMIN = -100.1
      YMAX=+20.0
      CALL PLTIN(XSCALE,YSCALE,XO,YO,XMIN,XMAX,YMIN,YMAX)
      DO 70 I=1,M
      CALL PLTLN(FREQ(I),MDB(I),FREQ(I+1),MDB(I+1))
70    CONTINUE
      CALL PLTB(N)
      CALL LBLF(5.75)
      CALL PLOT(12.0,0.0,-3)
      RETURN
      END
```

APPENDIX IV

SAWOPT

The following is a listing of the optimization programme SAWOPT. It is written in FORTRAN IV and operates under the NOS system for CDC-6000 computer systems. System libraries used are:

(1) SSPLIB - scientific subroutines package.

(2) PLOTVER - Versatec plotting library package.

Files TAPE1 and TAPE2 are replaced while the old files TAPE1 and TAPE2 are renamed TAPE10 and TAPE20.

```
SAWOPT. THIS PROGRAM DOES SUCCESSIVE OPTIMISATIONS OF SAW DESIGN.   SUTHERS
USER([redacted],[redacted])
CHARGE([redacted])
FTN.
SAVE,INPUT=SAWOPT.
      PROGRAM TST (INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT,TAPE1,TAPE2,
     +TAPE10,TAPE20)


      COMPLEX FFTS,V,RIDT,AR
      REAL MAG
      REAL LEN
      REAL MDB, PI, IMAG, LENGTH ,RELBP, IMAGL
      DIMENSION AR(513)
      DIMENSION V(500),XP(500),B(500),TIM(500),LEN(500),FRSP(20,3),FREQ(
     C513),MDB(513),PHASE(513),X(513),Y(513),Z(513),RMDB(513),RPHS(513),
     CFFTS(513),RIDT(513),MAG(513),PHSE(513)
      DIMENSION C0(500),C1(500),C2(500)
      READ (10,20) NZ,L,IG,T,DT,SPBD,FO,FC,VEL
      IR=0
      DO 10 I=1,NZ
      READ (10,30) FREQ(I),MDB(I),PHASE(I),RMDB(I),RPHS(I),MAG(I),PHSE(
     +I)
      IF (MDB(I).EQ.0.0.AND.IR.EQ.0) IR=I
10    CONTINUE
      IR=IR+3
      PI = ACOS (-1.0)
      N=NZ-1
      NH=N/2
      NM=NH+1
C     PREDISTORTION OF SPECIFICATIONS.

      DO 205 I=1,N
      RMERR=MDB(I)-RMDB(I)
      RPERR=PHASE(I)-RPHS(I)
      IF (MDB(I).LT.-20.0) RPERR=0.0
```

```
      MAG(I)=MAG(I)+0.50*RMERR
      PHSE(I)=PHSE(I)+0.50*RPERR
      AMP=10.0**(MAG(I)/20.0)
      CALL RECT(AMP,PHSE(I),RRE,RIE)
      FFTS(I)=CMPLX(RRE,RIE)
205   CONTINUE
210   FORMAT (1H0,2X,9F12.6)
      CALL PLTFRQ(N,FREQ,MAG,PHSE)

      CALL SAWFFT(FFTS,N,IG,X,Y,Z)
      X(NZ)=6.0
      Y(NZ)=Y(1)
      Z(NZ)=Z(1)
      CALL PLTFFT(NZ,X,Y,Z)
      FC=FO-SPBD/2.0
      DO 95 I=1,NH
      Z(NM+I)=Z(NM+I)-I*360.0*DT*(FO-FC)
      Z(NM-I)=Z(NM-I)+I*360.0*DT*(FO-FC)
95    CONTINUE
      CALL PHSSTR(NZ,Y,Z)
      CALL I0(8.0,DEN)
      DO 100 I=1,NH
      TM=FLOAT(NH**2-I**2)
      ARG=(16.0/N)*SQRT(TM)
      CALL I0(ARG,ANUM)
      WT=ANUM/DEN
      Y(NH+1+I)=Y(NH+1+I)*WT
      Y(NH+1-I)=Y(NH+1-I)*WT
100   CONTINUE
      CALL IDT(NZ,Y,Z,FO,SPBD,LEN,TIM,L)
      F1=FREQ(NH-2)
      F2=FREQ(NH+2)
      VEL=3410
      CALL MODEL(L ,LEN,TIM,B,XP,T,F1,F2,VEL,C0,C1,C2,  AR)
      CALL FRQRSP(L,AR,TIM,FREQ,RMDB,RPHS,NZ,T,RIDT)
```

```
      WRITE (1,20) NZ,L,IG,T,DT,SPBD,FO,FC,VEL
      RMAX=RMDB(IR)
      DO 50 I=1,NZ
      RMDB(I)=RMDB(I)-RMAX
      WRITE (1,30) FREQ(I),MDB(I),PHASE(I),RMDB(I),RPHS(I),MAG(I),PHSE(
     +I)
      IF (RMDB(I).LT.-100.0) RMDB(I)=-100.0
50    CONTINUE
20    FORMAT (3I5,6E10.5)
30    FORMAT (7E11.5)
      WRITE (2,81) L
      DO 70 I=1,L
      XL=XP(I)-B(I)
      XR=XP(I)+B(I)
      WRITE (2,80) LEN(I),XL,XR
70    CONTINUE
80    FORMAT (3E20.10)
81    FORMAT (I5)

      WRITE (6,152)
      DO 150 I=1,N
      WRITE (6,151) FREQ(I),MAG(I),MDB(I),RMDB(I),PHSE(I),PHASE(I),
     +RPHS(I)
150   CONTINUE
151   FORMAT (1H0,5X,F8.2,* MHZ*,F9.2,* DB*,F9.2,* DB*,F9.2,* DB*,F8.2,
     +* DEG*,F8.2,* DEG*,F8.2,* DEG*)
152   FORMAT (1H1,9X,*FREQ*,9X,*MAG*,9X,*MDB*,8X,*RMDB*,8X,*PHSE*,7X,
     +*PHASE*,8X,*RPHS*)

      CALL PLTFRQ(N,FREQ,RMDB,RPHS)
      CALL PLOT (X(N),Y(N),999)
      STOP
      END
```

```
      SUBROUTINE PHSCPS(N,Z)

      DIMENSION Z(N)
      DO 10 K=1,N
      GO TO 11
12    Z(K)=Z(K)-360.0
11    IF (Z(K).GT.+181.0) GO TO 12
      GO TO 13
14    Z(K)=Z(K)+360.0
13     IF (Z(K).LE.-179.0) GO TO 14
10    CONTINUE
      RETURN
      END


      SUBROUTINE SAWFFT(FFT1,N,IG,X,Y,Z)
      COMPLEX FFT1,FFT2,W,WK,WP
      REAL IMAG
      DIMENSION FFT1(N),IFFT(513,2),FFT2(513),X(N),Y(N),Z(N)
      PI=ACOS(-1.0)
      L=N
C     FAST FOURIER TRANSFORM FOR SAW DESIGN
C     FREQ IS ARRAY OF FREQUENCIES OF  POINTS OF FFT
C     MDB IS ARRAY OF DB OF INSERTION LOSS FOR EACH FREQUENCY
C     PHASE IS ARRAY OF PHASE,S FOR EACH FREQUENCY
C     IFFT IS ARRAY KEEPING TRACK OF EXPONENTS FOR MULTIPLIERS W(J) IN
C     THE FFT PROCESS
C     FFT1 AND FFT2 ARE COMPLEX VECTORS OF FFT PROCESS
      DO 10 I=1,513
      DO 10 J=1,2
10    IFFT(I,J) = 0
C     CALCULATE THE EXPONENT FOR THE MULTIPLIER FOR THE FFT PROCESS
      AREAL = COS(2.*PI/N)
      IMAG = SIN(2.*PI/N)
      W = CMPLX(AREAL,IMAG)
```

```
      DO 70 M=1,IG
      L=L/2
      NH = N/2
      NM=NH+1
      DO 50 K=1,NH
      IFFT(K,2) = IFFT(2*K,1)
50    IFFT(K+NH,2)=IFFT(K,2) +L
C     CALCULATE NEXT VECTOR IN FFT PROCESS
      DO 60 K=1,N
      WK = W**IFFT(K,2)
      IF (IFFT(K,2).GT.IFFT(K,1)) GO TO 61
      WP = WK*FFT1(K+L)
      FFT2(K) = FFT1(K)+WP
      GO TO 60
61    WP = WK*FFT1(K)
      FFT2(K) = FFT1(K-L) + WP
60    CONTINUE
      DO 70 I=1,N
      FFT1(I) = FFT2(I)
      IFFT(I,1)=IFFT(I,2)
70    CONTINUE
C     PUT OUTPUT VECTOR (DFT) IN CORRECT ORDER.
      AMAX=0.0
      DO 82 K=1,N
      KI=0
81    KI=KI+1
      IF (IFFT(KI,1).EQ.K-1) GO TO 80
      GO TO 81
80    RR=REAL(FFT1(KI))
      RI=AIMAG(FFT1(KI))
      CALL POLAR (RR,RI,PL,PA)
      Y(K)=PL
      Z(K)=PA
      IF (Y(K).GT.AMAX) AMAX=Y(K)
      X(K)=6.0*IFFT(KI,1)/N
82    CONTINUE
```

```
      DO 83 I=1,N
      Y(I)=Y(I)/AMAX
83    CONTINUE
      DO 93 K = 1,NH
      YS = Y(K)
      Y(K)=Y(K+NH)
      Y(K+NH)=YS
      ZS = Z(K)
      Z(K) = Z(K+NH)
      Z(K+NH) = ZS
93    CONTINUE
      RETURN
      END


      SUBROUTINE IDT(NZ,Y,Z,FO,SPBD,LEN,TIM,LH)

      REAL LEN
      DIMENSION Y(NZ),Z(NZ),LEN(500),TIM(500)
      NH=NZ/2
      NM=NH+1
      N=NZ-1
      T=N/SPBD
      DT=T/N
C     T IS THE IDT PERIOD IN MICRO-SECONDS
C     DT IS THE TIME INCREMENT OF THE FFT SAMPLES.
C     ADD IN THE LINEAR PHASE COMPONENTS OF THE FREQUENCY SHIFT.
      ZM=Z(NM)
      DO 40 I=1,NH
      Z(NM+I)=Z(NM+I)-ZM+(I*DT*FO*360.0)
      Z(NM-I)=Z(NM-I)-ZM-(I*DT*FO*350.0)
40    CONTINUE
      L=0
      M=0
      LH=1
      LEN(250)=Y(NM)
```

```
      TIM(250)=0.0
      DO 10 I=1,NH
      IF (L.EQ.249) GO TO 21
C     NP GIVES THE NUMBER OF FINGERS BETWEEN FFT POINTS.
      NC=(+Z(NM-1+I)+0.5)/180.0
      PHASL=180.0*NC
      NP=(+Z(NM+I)-PHASL+0.5)/180.0
      IF (NP.EQ.0) GO TO 21
C     OBTAIN TIMES,LENGTHS AND POLARITIES OF FINGERS.
      DO 20 K=I,NP
      L=L+1
      LH=LH+1
      PH=(180.0*(NC+K)-Z(NM-1+I))/(Z(NM+I)-Z(NM-1+I))
      TIM(250+L)=(I-1+PH)*DT
      SN=(-1.0)**(NC+K)
      LEN(250+L)=SN*(Y(NM-1+I)+(Y(NM+I)-Y(NM-1+I))*PH)
20    CONTINUE
21    IF (M.EQ.249) GO TO 10
C     NP GIVES THE NUMBER OF FINGERS BETWEEN FFT POINTS.
      NC=(-Z(NM+1-I)+0.5)/180.0
      PHASL=180.0*NC
      NP=(-Z(NM-I)-PHASL+0.5)/180.0
      IF (NP.EQ.0) GO TO 10
C     OBTAIN TIMES,LENGTHS AND POLARITIES OF FINGERS.
      DO 30 K=1,NP
      M=M+1
      LH=LH+1
      PH=(180.0*(NC+K)+Z(NM+1-I))/(-Z(NM-I)+Z(NM+1-I))
      TIM(250-M)=-(I-1+PH)*DT
      SN=(-1.0)**(NC+K)
      LEN(250-M)=SN*(Y(NM+1-I)+(Y(NM-I)-Y(NM+1-I))*PH)
30    CONTINUE
10    CONTINUE
      MN=250-M-1
      DO 50 I=1,LH
      LEN(I)=LEN(MN+I)
```

```
      TIM(I)=TIM(MN+I)
50    CONTINUE
49    RETURN
      END


      SUBROUTINE FRQRSP(L,V,TIM,FREQ,RMDB,RPHS,NZ,T,RIDT)

      REAL LEN
      COMPLEX RIDT
      COMPLEX DEL,V
      DIMENSION V(L),TIM(L),FREQ(NZ),RMDB(NZ),RPHS(NZ),RIDT(NZ)
      N=NZ-1
      NM=N/2+1
C     L IS THE NUMBER OF FINGERS IN THE IDT
C     LEN IS THE LENGTH OF FINGERS + POLARITY
C     TIM IS POSITION IN THE TIME DOMAIN OF FINGERS IN MICRO-SECONDS
C     FREQ IS ARRAY OF FREQUENCY POINTS FOR FFT
C     RMDB IS AMPLITUDE RESPONSE IN DB
C     RPHS IS THE PHASE RESPONSE IN DEGREES
C     NZ IS NUMBER OF SPECIFIED POINTS TO BE EVALUATED
      PI=ACOS(-1.0)
      RMAX=0.0
C     DO FAR FIELD INTEGRATION
      DO 10 I=1,NZ
      RIDT(I)=CMPLX(0.0,0.0)
      DO 20 J=1,L
      ANG=2*PI*FREQ(I)*(-TIM(J)-T/2.0)
      DEL=CMPLX(COS(ANG),SIN(ANG))
      RIDT(I)=RIDT(I)+DEL*V(J)
20    CONTINUE
      RSRL=REAL(RIDT(I))
      RSIM=AIMAG(RIDT(I))
      RMDB(I)=SQRT(RSRL**2+RSIM**2)
      RPHS(I)=(ATAN(RSIM/RSRL))*(180.0/PI)
      IF (RSRL.LT.0.0.AND.RSIM.LT.0.0) RPHS=RPHS-180.0
```

```
      IF (RSRL.GT.0.0.AND.RSIM.LT.0.0) RPHS=RPHS+180.0
10    CONTINUE
      RMAX=RMDB(NM)
C     OBTAIN AMPLITUDE IN DB
      DO 30 I=1,NZ
      RMDB(I)=20*ALOG10(RMDB(I)/RMAX)
      IF (RMDB(I).LT.-100.0) RMDB(I)=-100.0
      RPHS(I)=RPHS(I)-RPHS(NM)
30    CONTINUE
21    FORMAT (1H0,5X,*FREQ =*,F5.1,* MHZ*,5X,*MDB =*,F10.3,* DB*,5X,
     C*PHS =*,F10.3,* DEG*)
      RETURN
      END


      SUBROUTINE PHSSTR(NZ,Y,Z)

      DIMENSION Y(NZ),Z(NZ)
      NH=NZ/2
      NM=NH+1
      DO 10 I=1,NH
14    DPHS=Z(NM+I)-Z(NM-1+I)
      IF (DPHS.GT.+90.0) GO TO 11
      IF (DPHS.LT.-90.0) GO TO 12
      GO TO 13
11    Z(NM+I)=Z(NM+I)-180.0
      Y(NM+I)=-Y(NM+I)
      GO TO 14
12    Z(NM+I)=Z(NM+I)+180.0
      Y(NM+I)=-Y(NM+I)
      GO TO 14
13    DPHS=Z(NM-I)-Z(NM+1-I)
      IF (DPHS.GT.+90.0) GO TO 15
      IF (DPHS.LT.-90.0) GO TO 16
      GO TO 10
15    Z(NM-I)=Z(NM-I)-180.0
```

```
      Y(NM-I)=-Y(NM-I)
      GO TO 13
   16 Z(NM-I)=Z(NM-I)+180.0
      Y(NM-I)=-Y(NM-I)
      GO TO 13
   10 CONTINUE
      RETURN
      END


      SUBROUTINE RECT(PL,PA,RR,RI)

C     RECT CONVERTS (PL,PA) TO (RR,RI)
C     PL IS THE POLAR CO-ORDINATE LENGTH.
C     PA IS THE POLAR CO-ORDINATE ANGLE IN DEGREES.
C     RR IS THE RECTANGULAR CO-ORDINATE REAL PART.
C     RI IS THE RECTANGULAR CO-ORDINATE IMAGINARY PART.
      PI =ACOS(-1.0)
      RI=PL*SIN(PI*PA/180.0)
      RR=PL*COS(PI*PA/180.0)
      RETURN
      END


      SUBROUTINE POLAR(RR,RI,PL,PA)

C     POLAR CONVERTS (RR,RI)TO (PL,PA)
C     RR IS THE RECTANGULAR CO-ORDINATE REAL PART.
C     RI IS THE RECTANGULAR CO-ORDINATE IMAGINARY PART.
C     PL IS THE POLAR CO-ORDINATE LENGTH.
C     PA IS THE POLAR CO-ORDINATE ANGLE IN DEGREES.
      PI=ACOS(-1.0)
      PL=SQRT(RR**2+RI**2)
      IF (RR.EQ.0.0) GO TO 10
      PA =ATAN(RI/RR)
      IF (RR.LE.0.0.AND.RI.GE.0.0) PA =PA+PI
```

```
      IF (RR.LE.0.0.AND.RI.LE.0.0) PA =PA-PI
      PA =PA*180.0/PI
      GO TO 15
10    IF (RI) 11,12,13
11    PA=-90.0
      PL=RI
      GO TO 15
12    PA=0.0
      PL=0.0
      GO TO 15
13    PA=+90.0
      PL=RI
15    RETURN
      END


      SUBROUTINE MODEL(LZ,LEN,TIM,B,X,T,F1,F2,VEL,C0,C1,C2,  AR)

      COMPLEX AR
      REAL LEN,I1,I2,I3
      DIMENSION LEN(LZ),TIM(LZ),B(LZ),X(LZ),C0(LZ),C1(LZ),C2(LZ),
     CV(500,3),AR(LZ)
      PI=ACOS(-1.0)
      AL=ALOG(2.0)
      L=LZ-1
      VMAX=0.0
      DO 10 I=1,LZ
      IF (VMAX.LT.ABS(LEN(I))) VMAX=ABS(LEN(I))
      IF (VMAX.EQ.ABS(LEN(I))) IMAX=I
      X(I)=TIM(I)*VEL
      V(I,1)=0.0
      V(I,2)=0.0
      V(I,3)=0.0
10    CONTINUE
      B(1)=(X(2)-X(1))/4.0
      DO 20 I=2,L
```

```
      B(I)=(X(I+1)-X(I-1))/8.0
20    CONTINUE
      B(LZ)=(X(LZ)-X(L))/4.0

C     FINGER WIDTH AND POSITIONS HAVE BEEN FOUND.

      LH=LZ/2
      DO 30 I=1,LZ
      C0(I)=LEN(I)
      C1(I)=0.0
      C2(I)=0.0
30    CONTINUE

      CALL ELPOT(LZ,X,B,C0,C1,C2,V(IMAX,2),IMAX,2)
      FTR=ABS(LEN(IMAX)/V(IMAX,2))
      DO 41 I=1,LZ
      C0(I)=C0(I)*FTR
41    CONTINUE

      DO 42 I=1,LZ
      CALL ELPOT(LZ,X,B,C0,C1,C2,V(I,1),I,1)
      CALL ELPOT(LZ,X,B,C0,C1,C2,V(I,2),I,2)
      CALL ELPOT(LZ,X,B,C0,C1,C2,V(I,3),I,3)
42    CONTINUE
      BI30=0.5*ALOG(2.0)-0.75
      ORC=1.5

      DO 55 M=1,3

39    DO 43 I=1,LZ
      C0(I)=C0(I)+ORC*(LEN(I)-V(I,2))/AL
      C1(I)=C1(I)+ORC*(V(I,3)-V(I,1))/(2.0*B(I))
      C2(I)=C2(I)+ORC*(V(I,2)-(V(I,1)+V(I,3))/2.0)/BI30
43    CONTINUE
```

```
      CALL ELPOT(LZ,X,B,C0,C1,C2,V(IMAX,2),IMAX,2)
      FTR=ABS(LEN(IMAX)/V(IMAX,2))
      DO 50 K=1,LZ
      C0(K)=C0(K)*FTR
      C1(K)=C1(K)*FTR
      C2(K)=C2(K)*FTR
50    CONTINUE

      DO 51 I=1,LZ
      CALL ELPOT(LZ,X,B,C0,C1,C2,V(I,1),I,1)
      CALL ELPOT(LZ,X,B,C0,C1,C2,V(I,2),I,2)
      CALL ELPOT(LZ,X,B,C0,C1,C2,V(I,3),I,3)
51    CONTINUE

55    CONTINUE

52    FORMAT (1H0,5X,I5,7E15.4)


C     COEFICIENTS HAVE BEEN FOUND AND PRINTED.

      W1=SQRT(2.0*PI*F1)
      DO 70 I=1,LZ
      X1=2.0*PI*F1*B(I)/VEL
      CALL BESJ(X1,0,BJ01,0.0001,IER)
      CALL BESJ(X1,1,BJ11,0.0001,IER)
      ARR=W1*C0(I)*BJ01+C2(I)*W1*B(I)**2*(BJ01-BJ11/X1)
      ARI=W1*C1(I)*B(I)*BJ11
      AR(I)=CMPLX(ARR,ARI)
70    CONTINUE

C     ACOUSTIC RESPONSE FOR EACH FINGER HAS BEEN FOUND.

      RETURN
      END
```

```
      SUBROUTINE ELPOT(LZ,X,B,C0,C1,C2,VK,IK,IN)

      REAL I1,I2,I3
      DIMENSION X(LZ),B(LZ),C0(LZ),C1(LZ),C2(LZ)
      VK=0.0
      AL=ALOG(2.0)
      DO 41 J=1,LZ
      XR=X(IK)+(IN-2)*B(IK)-X(J)
      IF (XR.EQ.0.0) 43 ,48
48    BX=(B(J)/XR)**2
      ARG=1.0-BX
      IF(ARG.LE.0.0) GO TO 45
47    BS=SQRT(1.0-BX)
40    I1=ALOG(2.0/(1.0+BS))
      I2=BS-1.0
      I3=0.5*(BX*I1+BS-1.0-0.5*BX)
      IF (XR) 42,43,44
41    CONTINUE
      RETURN

42    VK  =VK  +C0(J)*I1-C1(J)*I2+C2(J)*I3
      GO TO 41

43    VK=VK+C0(J)*AL
      GO TO 41

44    VK =  VK +C0(J)*I1+C1(J)*I2+C2(J)*I3
      GO TO 41
50    FORMAT (1H0,5X,I5,7E15.5)
51    FORMAT (1H0,5X,I5,3E15.5)
45    BS=0.0
      GO TO 40

      END
```

```
      SUBROUTINE LBLF(YL)
      CALL LETTER (19,.1,.0,1.5,7.50,19HPHASE SPECIFICATION)
      CALL LETTER (23,.1,.0,1.5,1.2 ,23HAMPLITUDE SPECIFICATION)
      YL=5.75
      NA = 005500600060014401028
      DO 10 I=1,10
      NA=NA+1000000000000B
      YL = YL-0.4
10    CONTINUE
      CALL LBPH(9.75)
      RETURN
      END


      SUBROUTINE LBLT(YL)
      CALL LETTER (15,.1  ,0.0 ,1.5,1.25,15HIDT APODIZATION)
      CALL LETTER (31,.1  ,0.0 ,1.5,7.5,31HIDT DEVIATION FROM LINEAR PHA
     CSE)
      NA = 0060005600718
      YL = 5.75
      DO 10 I=1,10
      YL = YL - 0.4
      NA = NA - 0001B
10    CONTINUE
      CALL LBPH(9.75)
      RETURN
      END


      SUBROUTINE LBPH(YPH)
      DIMENSION Y(5), NB(5)
      NB(1) = 0055006100700600040B
      NB(2) = 005500710060040B
      NB(3) = 00600040B
      NB(4) = 005300710060040B
```

```
      NB(5) = 0053006100700600040B
      Y(1) = 7.75
      Y(2) = 8.25
      Y(3) = 8.75
      Y(4) = 9.25
      Y(5) = 9.75
      NC=0060B
      DO 20 K=1,5
      YB=Y(K)+0.1
20    CONTINUE
      RETURN
      END


      SUBROUTINE PLTB(N)
      N=N
      CALL PLOT (0.0,0.0,3)
      CALL PLOT (0.0,10.5,2)
      CALL PLOT (8.6,0.0,3)
      CALL PLOT (8.6,10.5,2)
      CALL PLOT (1.5,1.75,3)
      CALL PLOT (7.5,1.75,2)
      CALL PLOT (7.5,5.75,2)
      CALL PLOT (1.5,5.75,2)
      CALL PLOT (1.5,1.75,2)
      CALL PLOT (1.5,7.75,3)
      CALL PLOT (7.5,7.75,2)
      CALL PLOT (7.5,9.75,2)
      CALL PLOT (1.5,9.75,2)
      CALL PLOT (1.5,7.75,2)
      CALL PLOT (1.5,8.25,2)
      CALL PLOT (7.5,8.25,2)
      CALL PLOT (7.5,8.75,2)
      CALL PLOT (1.5,8.75,2)
      CALL PLOT (1.5,9.25,2)
      CALL PLOT (7.5,9.25,2)
```

```
      DO 200 I = 1,10
      XA = I*0.6+1.5
      CALL PLOT (XA,1.75,3)
      CALL PLOT (XA,1.55,2)
      CALL PLOT (XA,8.85,3)
      CALL PLOT (XA,8.65,2)
200   CONTINUE
      DO 201 I = 1,10
      YA = I*0.4+1.75
      CALL PLOT (1.5,YA,3)
      CALL PLOT (1.4,YA,2)
201   CONTINUE
      CALL PLOT (1.5,1.75,3)
      RETURN
      END


      SUBROUTINE PLTIDT(TIM,LEN,L)
      DIMENSION TIM(L),LEN(L)
      REAL LEN
      CALL PLOT(0.0,0.0,3)
      CALL PLOT(0.0,10.5,2)
      CALL PLOT(8.5,10.5,3)
      CALL PLOT(8.5,0.0,2)
      CALL PLOT(1.5,2.0,3)
      CALL PLOT(1.5,8.0,2)
      CALL PLOT(2.0,8.0,2)
      CALL PLOT(2.0,2.0,2)
      CALL PLOT(1.5,2.0,2)
      CALL PLOT(7.0,2.0,3)
      CALL PLOT(7.0,8.0,2)
      CALL PLOT(7.5,8.0,2)
      CALL PLOT(7.5,2.0,2)
      CALL PLOT(7.0,2.0,2)
      DO 10 I=1,L
      X=4.5+2.0*LEN(I)
```

```
      Y=((TIM(I)-TIM(1))/(TIM(L)-TIM(1)))*6.0+2.0
      IF (LEN(I).LT.0.0) GO TO 11
      CALL PLOT(2.0,Y,3)
      CALL PLOT(X,Y,2)
      GO TO 10
11    CALL PLOT(7.0,Y,3)
      CALL PLOT(X,Y,2)
10     CONTINUE
      CALL PLOT(12.0,0.0,-3)
      RETURN
      END


      SUBROUTINE PLTFFT(NZ,X,Y,Z)
      DIMENSION X(NZ),Y(NZ),Z(NZ)
      CALL PLTB(128)
      CALL LBLT(5.75)
      N=NZ-1
      M=N-1
      XSCALE=(X(NZ)-X(1))/6.0
      YSCALE=0.5
      ZSCALE=540.0
      XO=-1.5*XSCALE
      YO=-3.75*YSCALE
      ZO=-8.75*ZSCALE
      XMIN=X(1)
      YMIN=-1.0
      ZMIN=-540.0
      XMAX=X(NZ)
      YMAX=1.0
      ZMAX=+540.0
      CALL PLTIN(XSCALE,YSCALE,XO,YO,XMIN,XMAX,YMIN,YMAX)
      DO 10 I=1,N
      CALL PLTLN(X(I),Y(I),X(I+1),Y(I+1))
10    CONTINUE
      CALL PLTIN(XSCALE,ZSCALE,XO,ZO,XMIN,XMAX,ZMIN,ZMAX)
```

```
      DO 20 I=1,N
      CALL PLTLN(X(I),Z(I),X(I+1),Z(I+1))
20    CONTINUE
      CALL PLOT(12.0,0.0,-3)
      RETURN
      END


      SUBROUTINE PLTFRQ(N,FREQ,DB,DEG)

      DIMENSION FREQ(N),DB(N),DEG(N)
      DF=(FREQ(N)-FREQ(1))/(N-1)
      XSCALE = (FREQ(N)-FREQ(1)-DF)/6.0
      YSCALE = 180.0
      XO = -1.5 * XSCALE+FREQ(1)
      YO = -8.75 * YSCALE
      XMIN = FREQ(1)-DF
      YMIN = -180.1
      XMAX = FREQ(N)
      YMAX = +180.1
      M=N-1
      CALL PLTIN(XSCALE,YSCALE,XO,YO,XMIN,XMAX,YMIN,YMAX)
      DO 60 I=1,M
      CALL PLTLN(FREQ(I),  DEG(I),FREQ(I+1), DEG(I+1))
60    CONTINUE
      YSCALE = 25.0
      YO = -5.75*YSCALE
      YMIN = -100.1
      YMAX=+100.0
      CALL PLTIN(XSCALE,YSCALE,XO,YO,XMIN,XMAX,YMIN,YMAX)
      DO 70 I=1,M
      DBA=DB(I)
      DBB=DB(I+1)
      IF (DBA.LT.-100.0) DBA=-100.0
      IF (DBB.LT.-100.0) DBB=-100.0
      CALL PLTLN(FREQ(I),DBA,FREQ(I+1),DBB)
```

```
70    CONTINUE
      CALL PLTB(N)
      CALL LBLF(5.75)
      CALL PLOT(12.0,0.0,-3)
      RETURN
      END
```

APPENDIX V

TAPEJOB

The following is a listing of the programme TAPEJOB which generates the cutting instructions in the format requires by the mask cutting facility of Waterloo. It is written in FORTRAN IV and operates under the NOS system for CDC-6000 computer systems. System libraries used are:

(1) PLOTVER - Versatec plotting library package

TAPE8 is the information from TAPE2 or TAPE20 that results from SAWOPT or SAWINIT.

TAPE7 is the cutting instruction file.

```
TAPEJOB.
USER([redacted],[redacted]))                                    SUTHERS
CHARGE([redacted])
FTN.
SAVE,INPUT=TAPEJOB.
      PROGRAM TST (INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT,TAPE8,TAPE7)
      DIMENSION XL(500),XR(500),Y(500),XA(500),XB(500),XC(500),XD(500)
      READ (8,10) N
      I=63B
      WRITE (7,35) I
35    FORMAT (A6)
      WRITE (7,31)
      WRITE (7,32)
      WRITE (7,34)

      YU=+16000.0
      YL=-16000.0
      DO 20 I=1,N
      READ (8,30) Y(I),XL(I),XR(I)
      B=0.0
      XA(I)=8.0*(XL(I)-B)+0.5
      XB(I)=8.0*(XL(I)+B)+0.5
      XC(I)=8.0*(XR(I)-B)+0.5
      XD(I)=8.0*(XR(I)+B)+0.5
      Y(I)=Y(I)*15500.0+0.5
20    CONTINUE

C     CUT PADS AND BENCH-MARKS.

      CALL PLTCUT(3,XA(1),YL)
      CALL PLTCUT(2,XD(N),YL)
      CALL PLTCUT(2,XD(N),-20000.0)
      XP=XA(1)+4000.0
      CALL PLTCUT(2,XP,-20000.0)
      CALL PLTCUT(2,XP,-24000.0)
      XP=XP+4000.0
```

```
CALL PLTCUT(2,XP,-24000.0)
CALL PLTCUT(2,XP,-32000.0)
CALL PLTCUT(2,XA(1),-32000.0)
CALL PLTCUT(2,XA(1),-16000.0)
CALL PLTCUT(3,XD(N),+16000.0)
CALL PLTCUT(2,XA(1),+16000.0)
XP=XA(1)-4000.0
CALL PLTCUT(2,XP,-16000.0)
CALL PLTCUT(2,XP,-32000.0)
XP=XP-8000.0
CALL PLTCUT(2,XP,-32000.0)
CALL PLTCUT(2,XP,-24000.0)
XP=XP+4000.0
CALL PLTCUT(2,XP,-24000.0)
CALL PLTCUT(2,XP,-16000.0)
XP=XA(1)-4000.0
CALL PLTCUT(2,XP,+16000.0)
CALL PLTCUT(2,XP,+20000.0)
XP=XD(N)-4000.0
CALL PLTCUT(2,XP,+20000.0)
CALL PLTCUT(2,XD(N),+16000.0)
CALL PLTCUT(3,+20.0,-24000.0)
CALL PLTCUT(2,+20.0,-32000.0)
CALL PLTCUT(2,-20.0,-32000.0)
CALL PLTCUT(2,-20.0,-24000.0)
CALL PLTCUT(2,+20.0,-24000.0)
CALL PLTCUT(3,+4000.0,-27980.0)
CALL PLTCUT(2,+4000.0,-28020.0)
CALL PLTCUT(2,-4000.0,-28020.0)
CALL PLTCUT(2,-4000.0,-27980.0)
CALL PLTCUT(2,+4000.0,-27980.0)
CALL PLTCUT(3,+20.0,+24000.0)
CALL PLTCUT(2,+20.0,+32000.0)
CALL PLTCUT(2,-20.0,+32000.0)
CALL PLTCUT(2,-20.0,+24000.0)
CALL PLTCUT(2,+20.0,+24000.0)
```

```
      CALL PLTCUT(3,+4000.0,+27980.0)
      CALL PLTCUT(2,+4000.0,+28020.0)
      CALL PLTCUT(2,-4000.0,+28020.0)
      CALL PLTCUT(2,-4000.0,+27980.0)
      CALL PLTCUT(2,+4000.0,+27980.0)

C     PADS AND BENCH-MARKS ARE CUT.

C     GENERATE LINE-CUT INSTRUCTIONS.

      DO 50 I=1,N
      CALL PLTCUT(3,XA(I),YL)
      CALL PLTCUT(2,XA(I),YU)
      CALL PLTCUT(3,XD(I),YU)
      CALL PLTCUT(2,XD(I),YL)
50    CONTINUE

C     GENERATE APODISATION-CUT INSTRUCTIONS.

      CALL NEWPEN(5)
      DO 60 I=1,N
      IF (Y(I).LT.0.0) GO TO 60
      BL=4.0*(XC(I)-XA(I))
      BT=8.0*BL
      IF (Y(I).GT.BT) YT=Y(I)-(2.0*BL)
      IF (Y(I).LE.BT) YT=Y(I)-2.0*BL*(+Y(I)/(BL*4.0)-(+Y(I)/(BL*8.0))
     +**2)
      CALL PLTCUT(3,XA(I),YT)
      CALL PLTCUT(2,XD(I),YT)
      Y(I)=Y(I)+500.0
60    CONTINUE

      DO 70 I=1,N
      IF (Y(I).LT.0.0) GO TO 70
      BL=4.0*(XC(I)-XA(I))
      BT=8.0*BL
```

```
      IF (Y(I).GT.BT) YT=Y(I)-(2.0*BL)
      IF (Y(I).LE.BT) YT=Y(I)-2.0*BL*(+Y(I)/(BL*4.0)-(+Y(I)/(BL*8.0))
     +**2)
      CALL PLTCUT(3,XA(I),YT)
      CALL PLTCUT(2,XD(I),YT)
      Y(I)=Y(I)-500.0
70    CONTINUE

      DO 80 I=1,N
      IF (Y(I).GT.0.0) GO TO 80
      BL=4.0*(XC(I)-XA(I))
      BT=-8.0*BL
      IF (Y(I).LT.BT) YT=Y(I)+(2.0*BL)
      IF (Y(I).GE.BT) YT=Y(I)+2.0*BL*(-Y(I)/(BL*4.0)-(-Y(I)/(BL*8.0))
     +**2)
      CALL PLTCUT(3,XA(I),YT)
      CALL PLTCUT(2,XD(I),YT)
      Y(I)=Y(I)-500.0
80    CONTINUE

      DO 90 I=1,N
      IF (Y(I).GT.0.0) GO TO 90
      BL=4.0*(XC(I)-XA(I))
      BT=-8.0*BL
      IF (Y(I).LT.BT) YT=Y(I)+(2.0*BL)
      IF (Y(I).GE.BT) YT=Y(I)+2.0*BL*(-Y(I)/(BL*4.0)-(-Y(I)/(BL*3.0))
     +**2)
      CALL PLTCUT(3,XA(I),YT)
      CALL PLTCUT(2,XD(I),YT)
      Y(I)=Y I)+500.0
90    CONTINUE
      WRITE (7,100)
      CALL PLOT(0.0,0.0,999)

10    FORMAT (I5)
30    FORMAT (3E20.10)
```

```
31    FORMAT (*D90*)
32    FORMAT (*G1*)
33    FORMAT (*D1*)
34    FORMAT (*D2*)
100   FORMAT (*M30*)

      STOP
      END


      SUBROUTINE PLTCUT(M,XD,YD)

      IX=IFIX(XD)
      IY=IFIX(YD)
      YP=(YD+34000.0)/6400.0
      XP=(XD+60000.0)/6400.0
      CALL PLOT(XP,YP,M)
      L=M-2
      IF (L) 10,20,30
10    WRITE (7,1) IX,IY
      RETURN
20    WRITE (7,2) IX,IY
      RETURN
30    WRITE (7,3) IX,IY
      RETURN

1     FORMAT (*X*,I7.7,*Y*,I7.7)
2     FORMAT (*X*,I7.7,*Y*,I7.7,*D1*)
3     FORMAT (*X*,I7.7,*Y*,I7.7,*D2*)

      END
```

APPENDIX VI

FABRICATION OF FILTERS

Once an acceptable design for a receiving and a transmitting transducer had been obtained, the programme TAPEJOB generated the cutting instructions in the proper format and stored these instructions as a file on the system mass storage. These files were accessed through a facility of the Canadian Insitute of Metal Working which translated the instructions onto paper tape. The paper tape was then taken to the University of Waterloo where the Rubylith master mask was cut for each transducer at 80x scale.

After peeling the Rubylith, the masks were taken to Hughes-Owens Reproduction Department in Hamilton where a 4:1 reduction was done of each mask giving a 20x negative transparency.

These transparencies were then assembled along with the multi-strip coupler on a 50 cm x 25 cm master filter mask here at McMaster. This mask was then reduced 20:1 to give the final positive mask on a Kodak 5 cm x 5 cm high resolution plate. The camera system (in the McMaster Electrical Engineering Department) is an HLC Microkon 1600 Microphotography Camera system that has a resolution at 200 line pairs/mm. and a 2.5 cm square image area.

Having produced the final positive mask, it now remains to prepare the substrate. The lithium niobate ($LiNBO_3$) substrates which are 2.5 cm in the z-direction and 1.25 cm in the x-direction are

prepared as follows:

(1) scrub in trichloroethylene using cotton swabs for 2 minutes.

(2) rinse with trichloroethylene.

(3) scrub in acetone with cotton swabs for 2 minutes.

(4) rinse with acetone.

(5) scrub in distilled water with cotton swabs for 2 minutes.

(6) rinse with distilled water.

(7) place on vacuum chuck of Headway Research Inc. Model EC101 spinner, spin at 6000 rpm and rinse substrates with pure filtered methyl- or ethyl-alcohol for two minutes.

(8) place in clean-covered petri dish and bake in oven at 85° C for 15 minutes.

(9) inspect under microscope for surface contamination and repeat steps (1) through (8) if necessary (NOTE: all the above steps are done on a clean air bench in as dustfree an environment as is possible).

(10) if the substrate(s) are acceptably clean then they are placed in an Edwards Model 12E3/1670 coating unit and are subjected to ionic (argon) cleaning at as low an atmosphere as is required to maintain ionization for 15 minutes.

(11) then approximately 3000 to 5000 Å of aluminum is evaporated onto the substrate surface using tungsten filament sources. The deposition rate is not critical since aluminum adheres well to lithium niobate.

(12) allow the coated substrates to thoroughly cool in the evaporator

(4 hrs), then remove the substrates via clean covered petri dishes back to the clean air bench with the Headway spinner.

(13) place metallized substrate on vacuum chuck and spin at 6000 rpm and rinse surface with pure filtered methyl- or ethyl-alcohol and spin dry for 2 minutes.

(14) stop spinner and cover surface with filtered Shipley AZ-1350B positive working photo resist, then spin at 6000 rpm for 2 minutes.

(15) place resist covered substrates in petri dish and bake at 85°c for 15 minutes.

(16) the substrates are now ready to be exposed. Place final mask with the emulsion side down on top of the resist covered substrate using the 2.5 cm x 1.25 cm master mask outline for alignment with substrate edges.

(17) expose to UV light source for 1 minute.

(18) place exposed substrate in Shipley AZ developer and observe development (without agitation) until all exposed resist has been dissolved.

(19) rinse substrate with distilled water.

(20) place substrate in acid etch composed of 25 parts phosphoric acid (H_3PO_4), 5 parts acetic acid ($C_2H_4O_3$) and 1 part nitric acid (HNO_3). Observe etching through microscope. Remove when etching is complete.

(21) rinse substrate with water and then remove unexposed photo resist with acetone.

(22) rinse with distilled water and blow dry with dry nitrogen.

(23) using RTV silicone based adhesive, glue substrate to 2.5 cm x 1.25 cm Tekform butterfly package.

(24) using Kulicke and Soffa model 472 ultra-sonic bonder, bond rail connections to appropriate pin connections of case.

(25) use RTV as acoustic absorber at ends of acoustic paths.

(26) test device.

REFERENCES

[1] G.W. Farnell and E.L. Adler, "An overview of acoustic surface-wave technology", Final Report to Communications Research Centre on DSS Contract 36001-3-4406, August 12, 1974.

[2] B.A. Auld, "Acoustic fields and waves in solids", Vol. I and Vol. II, John Wiley and Sons, New, York, N.Y., 1973.

[3] J.P. Reilly, "Realization of arbitrary frequency responses using surface acoustic wave devices", M.Eng. Thesis, McMaster University, Hamilton, Ontario, Canada, January 1977.

[4] D.H. Hurlburt, "The philosophy and design of surface acoustic wave filters", RCA Engineer, 19, 6, April/May 1974.

[5] M.H. El-Diwany, "Surface acoustic wave bandpass filter synthesis and design", M.Eng. Thesis, McMaster University, Hamilton, Ontario, Canada, May 1975.

[6] G.R. Nudd, M. Waldner and R.L. Zimmerman, "Design of unapodized surface-wave transducers with spectral weighting", IEEE Transactions on Microwave Theory and Techniques, Vol. MTT-22, No. 1, Jan. 1974, pp. 25-32.

[7] H. Engan, "Series-weighting of surface acoustic wave transducers", 1974 Ultrasonic Symposium Proceedings, IEEE Cat. #74 CHO 896-1SU, pp. 422-424.

[8] K.R. Laker, E. Cohen and A.J. Slobodnik, "Electric field ineractions within finite arrays and the design of withdrawal of weighted SAW filters of fundamental and higher harmonies", 1976 Ultrasonics Symposium Proceedings, IEEE Cat. #76 CH 1120-5SU, pp. 317-321.

[9] A.V. Oppenheim and R.W. Schafer, "Digital signal processing", Prentice Hall, Inc., Englewood Cliffs, N.J., 1975.

[10] S. Haykin, "Communications systems", John Wiley and Sons, New York, N.Y., 1978.

[11] J.P. Reilly, C.K. Campbell and M.S. Suthers, "The design of SAW bandpass filters exhibiting arbitrary phase and amplitude response characteristics", IEEE Transactions on Sonics and Ultrasonics, Vol. SU-24, No. 5, September 1977, pp. 301-305.

[12] R.F. Mitchell and D.W. Parker, "Synthesis of acoustic-surface-wave filters using double electrodes", Electronic Letters, 1974, 10, p. 512.

[13] R. LaRosa and S.J. Kerbel, "Synthesis of transfer functions by parallel-channel SAW filter banks", 1976 Ultrasonics Symposium Proceedings, IEEE Cat. #76 CH 1120-5SU, pp. 322-327.

[14] R.L. Miller and A.J. DeVries, "A simple building block method for design of SAW filters having nonlinear phase response"; 1976 Ultrasonics Symposium Proceedings, IEEE Cat. #76 CH 1120-5SU, pp. 553-557.

[15] T.J. Boege, G. Chae and W.S. Drammond, "Design of arbitrary phase and amplitude characteristics in SAW filters", 1976 Ultrasonics Symposium Proceedings, IEEE Cat. #76 CH 1120-5SU, pp. 313-316.

[16] R.F. Mitchell, "Surface acoustic wave transversal filters: their use and limitations", IEE Conf. Publ. 109, 1973, pp. 130-140.

[17] E.O. Brigham and R.E. Morrow, "The fast Fourier transform", IEEE Spectrum, Dec. 1967, pp. 63-70.

[18] H. Skeie and Ronnekleiv, "Electrostatic neighbour and end effects in weighted surface wave transducers", 1976 Ultrasonics Symposium Proceedings, IEEE Cat. #76 CH 1120-5SU, pp. 540-542.

[19] E. Halten, "Elektricitetslara", Stockholm, Sweden, 1953, pp. 56-59.

[20] F.J. Harris, "On the use of windows for harmonic analysis with the discrete Fourier transform", Proceedings of the IEEE, Vol. 66, No. 1, Jan. 1978, pp. 51-83.

[21] J. Zucker, "Surface acoustic wave devices", GTE Laboratories Profile, Two / 1974.

[22] J. Papay, "Private communication", Electrohome Ltd., Kitchener, Ont., Canada, 1976.

[23] S.M. Selby, "Standard math tables", The Chemical Rubber Company, 1970.

[24] M.R. Spiegel, "Mathematical Handbook of Formulas and Tables", Schaum's Outline Series, McGraw Hill Co., New York, N.Y., 1968.