1990

The extraction of the equivalent circuit parameters of surface acoustic wave resonators

Brent H. Horine
brent.horine@manhattan.edu

THE EXTRACTION OF THE EQUIVALENT CIRCUIT PARAMETERS OF SURFACE ACOUSTIC WAVE RESONATORS

BY

BRENT HANSEN HORINE
B.S.E.E., Marquette University, 1983

THESIS

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

Fall Term
1990

ABSTRACT

The application of the Electronic Industries Association suggested standard, EIA-512 [1], has been successfully applied to quartz crystal resonators for extraction of the equivalent circuit model parameters. This technique has been shown to be very successful in the operation range to several hundred megahertz. This same technique is proposed to be applied to many differing types of resonators through the full spectrum of frequencies. This thesis will discuss measurement and testing of Surface Acoustic Wave (SAW) resonators at the higher frequencies.

The packaging configuration in conjunction with SAW devices can lead to significant parasitic effects. In order to successfully extract the SAW crystal parameters and properly predict the measured frequency response, it is important to model the resulting equivalent circuit. This circuit includes both the quartz crystal resonator and all packaging effects. The approach used has yielded good results in the tested range up to approximately 1 GHz.

ACKNOWLEDGEMENTS

I would like to express my deep appreciation for the constant support, encouragement, and most of all, patience of my wife and children. I would also like to thank my friends in the church who helped me keep everything in perspective.

Much of the foundation for this work was laid by Donald C. Malocha. I am indebted to him for his many suggestions and encouragement. Finally, I would like to acknowledge the assistance of the EIA for providing me with some of their extraction software. The use of this software was a great help in the early phases of this research.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER I

INTRODUCTION

Proper oscillator design requires, in part, a thorough knowledge of the feedback path and the impedances presented to the active element. While the main response of the resonator is intended to be the feedback path, parasitic elements can result in additional paths that may satisfy the oscillation criteria of unity loop gain with n*360° of loop phase shift. While good resonator design seeks to reduce the parasitics, they cannot be eliminated entirely. Instead, a means of quantifying them must be developed in order that the oscillator designer may take appropriate measures to reduce their impact.

The study of parasitic feedback paths in a resonator is primarily concerned with its transmission characteristics. The reflection characteristics are also important since most UHF and L-band transistors are not unconditionally stable. An improper impedance at a transistor port can push the transistor into parasitic oscillation [2]. This oscillation arises from the feedback path presented by a finite reverse transmission (S12) of the active element. Because all of the S-parameters of the transistor and the impedance of the SAW resonator are functions of frequency, a study of the stability of the transistor is needed wherever the transistor has significant gain.

The availability of circuit modeling software has greatly simplified this task. Most commercially available software packages accept S-parameter data as input [3]. While it is relatively simple to collect S-parameter data from either actual devices or modeled devices using a SAW based modeling program, it is impractical to take advantage of the yield analysis and design centering capabilities of the circuit modeling software.

With an equivalent circuit which includes the main response and the parasitic effects, a more sophisticated approach may be used. For example, the series capacitance may be calculated as a function of the temperature to study the frequency versus temperature stability of the overall oscillator. Other possibilities include making the bond wire inductance a random variable in a Monte Carlo analysis in order to determine the effects of the variability of different assembly operators on the oscillators's spurious margin.

Besides yield studies, using an equivalent circuit approach facilitates design centering. The relationship between physical design parameters (such as metal thickness, cavity length, transducer length, etc.) is more clearly understood with respect to equivalent circuit parameters than with a series of S-parameters [4]. With the equivalent circuit approach to oscillator modeling, optimizers, using suitable constraints, can be applied on the resonator itself in order to optimize its design in conjunction with the oscillator circuit design. This helps the SAW designer choose the optimum physical design parameters.

The extraction technique described herein, is an adaptation of the Electronic Industries Association (EIA) standard for measurement of the equivalent electrical parameters of quartz crystal units (EIA-512) [1]. Successive application of this standard technique can extract not only the fundamental response of the SAW resonator, but also the packaging related parasitics and spurious longitudinal modes. Examples are presented for both single and dual port SAW resonators. Test circuits are modeled using typical component values in order to evaluate the extraction process itself independent of measurement errors and unanticipated parasitic responses. Several actual SAW resonators have been measured and extracted to demonstrate the practicality of the extraction technique.

CHAPTER II

SAW RESONATOR CHARACTERISTICS

In order to fully appreciate the extraction process, a review of SAW resonator characteristics is necessary. A typical dual port SAW resonator is illustrated in Figure 1. The basic response results from an acoustic wave traveling in a cavity formed by the proper separation of two distributed reflector arrays. Whenever the separation between the edgesof the arrays is an integer number of half-wavelengths, a resonant mode is supported.

In order to achieve high Qs, a long cavity is used, typically 150 to 250 half-wavelengths. This gives rise to spurious modes. For example, assuming that the device is designed with a 100 half-wavelength cavity, spurious modes will exist at frequencies where the physical cavity is 99 half-wavelengths and 101 half-wavelengths. These are referred to as the first longitudinal modes. An ideal cavity would support an infinite number of modes; however, the reflectors have a finite bandwidth over which they efficiently reflect the acoustic energy. In typical designs, only the fundamental mode and the first longitudinal modes exist. Because of the half-wavelength periodicity, the insertion phase of the longitudinal modes is opposite that of the fundamental mode. The response of these modes is illustrated in Figure 2.

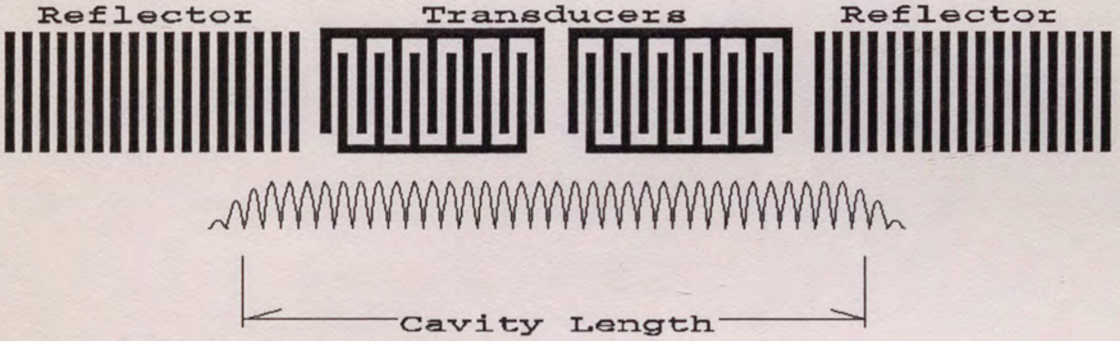

Figure 1: Layout of a typical dual port SAW resonator

In addition to being bounded in the longitudinal direction, the device is bounded in the transverse direction by the difference in acoustic velocity in the metallized versus unmetallized regions. This gives rise to transverse modes. Only a finite number of transverse modes can exist depending on the relative difference of the velocities and the aperture of the device [5]. Interdigitated transducers are used to convert electrical energy into acoustic energy and inject this energy into the cavity. Proper apodization of these transducers will result in efficient coupling to the fundamental transverse mode and very inefficient coupling to

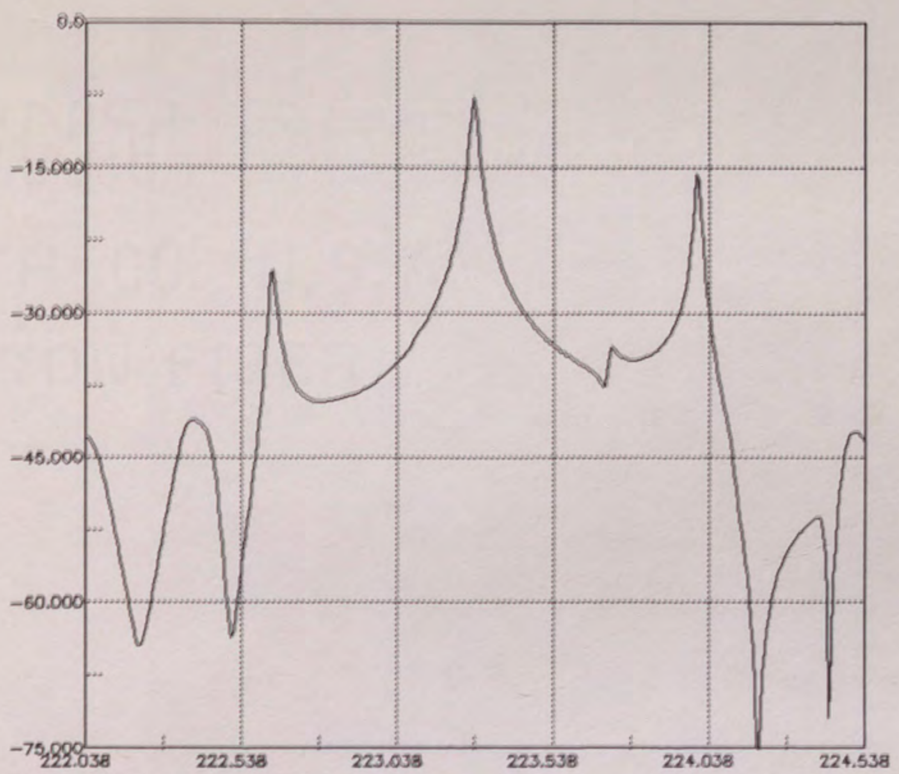

Figure 2: Typical dual port SAW resonator response

the higher order transverse modes. Because of the larger velocity difference encountered at high frequencies, resonators near 1 GHz sometimes exhibit distortion due to transverse moding. It will be demonstrated that it is possible to extract and analyze the longitudinal modes; however, because of the close frequency spacing to the fundamental mode, extracting transverse modes is not practical with this approach.

As mentioned previously, interdigitated transducers are used to couple energy in and out of the cavity. Either one or two transducers may be used, producing a single or dual port resonator, respectively. Generally, the transducers fill up the available cavity space in order to provide reasonable coupling to the load. The transducers have the same form as an interdigital capacitor and this will be present as an additional element in the equivalent circuits.

The resonator is most often packaged in a TO-39 type header. Bond wires connect the SAW die to the header leads. These bond wires can be represented in the equivalent circuit as a series inductor and resistor. The leads are insulated from the header base by a glass feedthrough. The glass acts as a dielectric and forms a capacitor between the lead and the base which is normally at ground potential. This capacitor and the bond wire resistance and inductance form a parasitic resonance with the transducer capacitance.

In conclusion, there are four significant resonances in a SAW resonator, 1) the main resonance, 2) the longitudinal modes, 3) the transverse modes, and 4) the package resonance. All except the transverse modes can be extracted using successive applications of the EIA-512 technique. The elements can then be included in a single circuit that can simultaneously describe the significant narrowband responses and the wideband response up to nearly 3 GHz.

CHAPTER III

REVIEW OF EQUIVALENT CIRCUITS

There have been a variety of equivalent circuits used for modeling resonators. Hafner [6] describes several bulk wave equivalent circuits with different levels of complexity. Generally, extra complexity is required at higher operating frequencies where parasitic elements are more significant. The equivalent circuits that have been applied to SAW resonators are extensions of Hafner's circuits with appropriate modifications for the unique characteristics of SAW resonators.

The basic equivalent circuit of a resonator is a series combination of an inductor, capacitor, and resistor. The resistor represents the loss of the resonator. The inductive and capacitive reactances cancel at resonance, with the inductor value being chosen in accordance with the Q of the resonator and the resistor value. The elements are often referred to as motional elements, because their orgin is in the motion of the crystal.

In both bulk wave and surface wave resonators, the transducers introduce an additional capacitance. The arrangement of these capacitors in a SAW device depends upon the number of transducers. In a single port device, the capacitor bridges the series resonance elements [7] as in Figure 3. The presence of this capacitor limits the tuning range of the resonator and also provides a parasitic feedback path. Shreve points out that in a dual port resonator, the transducer capacitance is in parallel with the input and output ports [8] as illustrated in Figure 4.

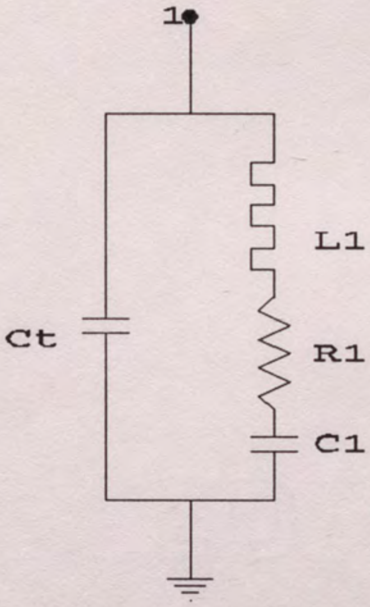

Figure 3: Basic single port equivalent circuit

Shreve also includes three resistors which represent the radiation conductance of each transducer and the transmission between the two transducers, respectively. Because the transducers are frequency selective, the resistors are only accurate very near the operating frequency. Since this effort seeks a model that is accurate over a several decade bandwidth, these resistors are normally left out.

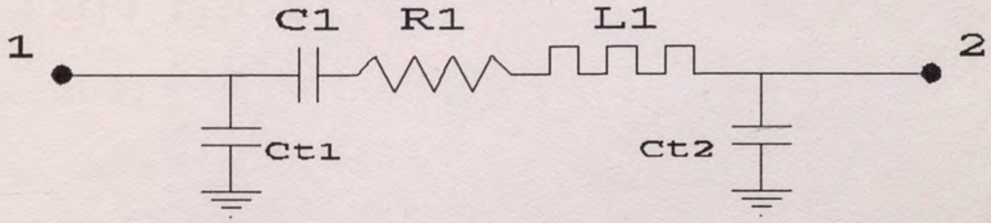

Figure 4: Basic dual port equivalent circuit

Fortunately, typical SAW resonators are built with relatively short transducers on quartz which has a low piezoelectric coupling constant. This results in a transducer impedance that is dominated by its capacitance and the real part is negligible

Gunes [9] adds a series holder inductance and resistance for evaluating VHF SAW resonators in HC-6 packages. This approach is also necessary in TO-39 packages, except rather than the elements representing the holder clips, they will represent the bond wire inductance and resistance, the bond resistance, and the post inductance and resistance.

As the center frequency increases, the geometry of the transducers decreases. This results in smaller transducer capacitances. At 300 MHz, a typical dual port transducer capacitance is 1.5 pF. This compares significantly with the package capacitance which is 0.5 to 1.25 pF. In a simple circuit, these capacitors are simply summed together as if they were in parallel. In actuality, the bond wire impedance separates them and they must be handled individually. The additional packaging related elements result in the circuits described in Figure 5 for single port devices and in Figure 6 for dual port devices.

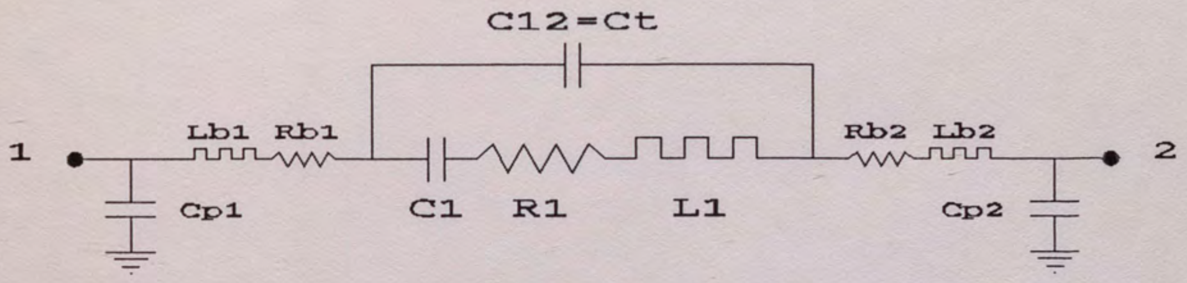

Figure 5: Full single port equivalent circuit in transmission mode

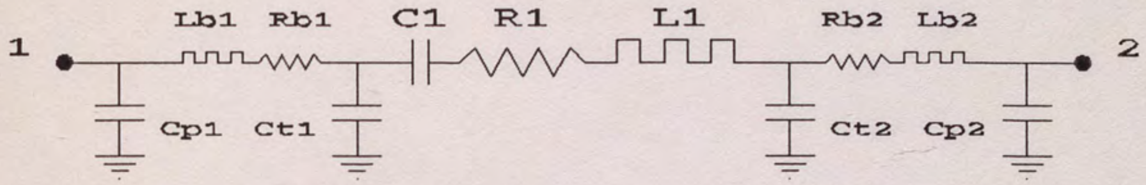

Figure 6: Full dual port equivalent circuit

Note that even the single port resonator has three terminals. This is because in practice, the base is isolated from the resonator itself and the rf connections are made through feedthrough leads. This allows the base to be firmly tied to ground potential even when the resonator is connected between the collector and base of the transistor as in a typical Pierce configuration. The circuit of Figure 7 results when the resonator is viewed as a true single port device. These circuits encompass two of the four types of resonances that exist in a SAW resonator. Longitudinal modes are included by adding parallel branches of the series resonance elements as illustrated in Figure 8. The 180° phase shifter is necessary for the proper vector addition of the signals

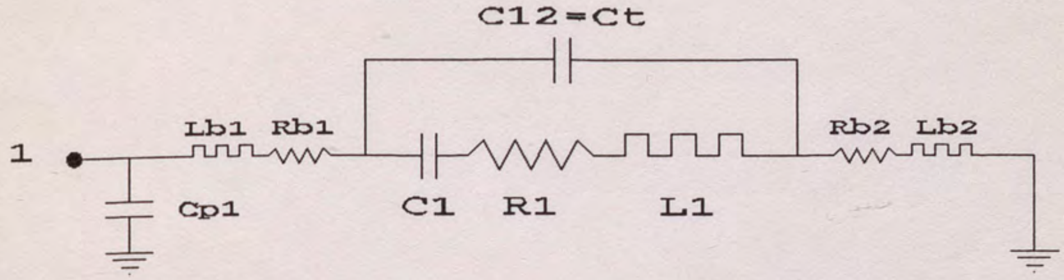

Figure 7: Full single port equivalent circuit with second terminal grounded

from each branch. The phase shifter can also be used in the circuit of Figure 6 to indicate a 180° resonator.

These four circuits are valid not only near center frequency but also across a frequency range from 1 MHz to over 2 GHz. They describe all of the significant spurious responses of a typical SAW resonator. They do not account for transverse modes, harmonic responses, or spurious bulk modes, none of which are significant for well designed SAW resonators. They also do not describe the delay line response of dual port resonators in the stopband region of the reflectors [10], although this response is again insignificant. The circuits are composed of only lumped,

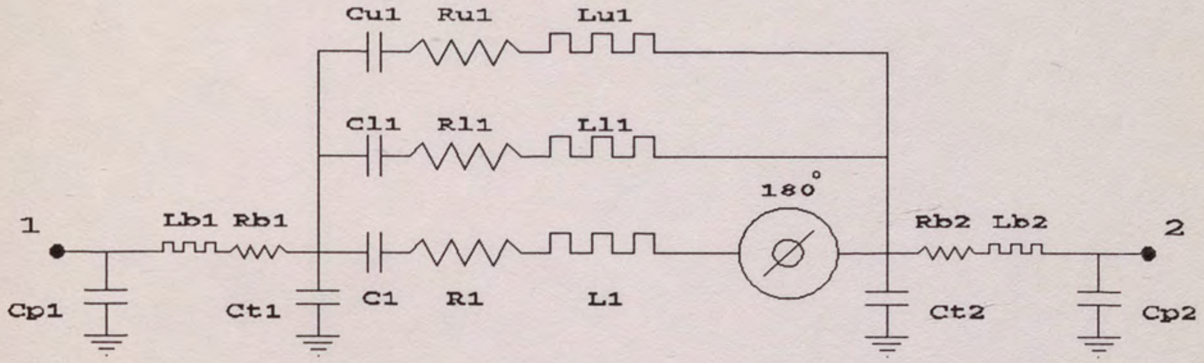

Figure 8: Full dual port equivalent circuit with longitudinal modes

passive elements and one idealized element (180° phase shifter). As will be seen, these elements are extracted without the need to resort to the iterative optimization techniques or initial guesses required by other methods [9,11]. The resulting circuits are easily loaded into one of the popular commercially available RF modeling software packages [2].

CHAPTER IV

PARAMETER EXTRACTION METHOD

The EIA-512 standard describes a procedure to extract the equivalent circuit parameters from a file of S-parameters. This method is based upon the equivalent circuit as illustrated in Figure 9. It is intended to extract a circuit which

> " . . . approximates the admittance/ impedance characteristic of a crystal unit over a relatively narrow range of frequencies near an isolated mode of vibration, at a particular excitation level. Non-linear amplitude behavior, or the presence of interfering modes of vibration not only interfere with the actual measurement, but also render the accepted equivalent network representation invalid [12]."

The EIA-512 method does not directly extract a circuit as shown in Figure 5 and Figure 6; however, these circuits may be split into wideband and narrowband subcircuits. For the single port resonator, two different circuits can be used. The first, diagrammed in Figure 10 has the topology of a two port resonator formed by the bondwire inductance and the transducer capacitance. The second circuit has its second port grounded as in Figure 11. This results in a single port resonator. The procedure can be repeated with the first port grounded in order to recover the value of the second port package capacitance, C_{p2} if some nonsymmetry is suspected.

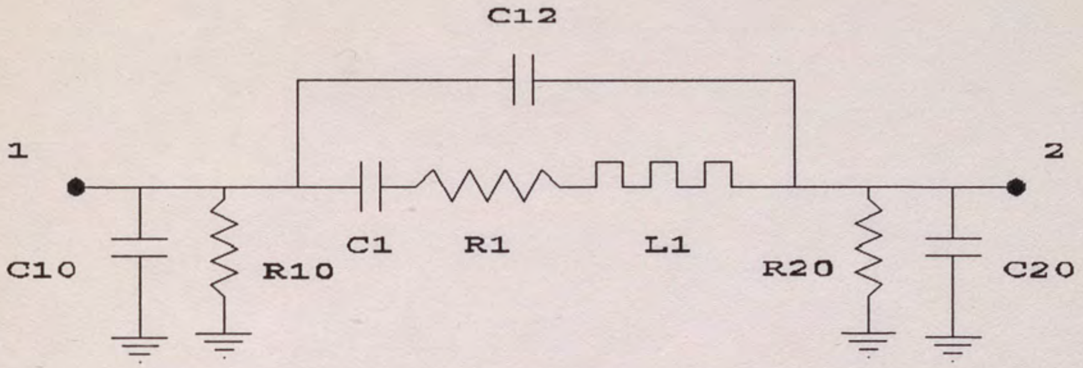

Figure 9: EIA-512 equivalent circuit

Note that since L_{b1} is in series with L_{b2} and R_{b1} is in series with R_{b2}, it will not be possible to extract their individual values, but only their sums. On the other hand, for the same reason, the allocation of these sums into individual elements has no significance on the modeled response. In practice, the values are split evenly between the two resistances and inductances.

The high frequency behavior of the dual port resonator can be represented by the circuits of Figure 12. In this case, a pair of single port circuits results. Note that all of the parasitic elements can be uniquely determined.

The resulting narrowband circuits are the circuits of Figure 3 and Figure 4. Now the appropriate extraction method

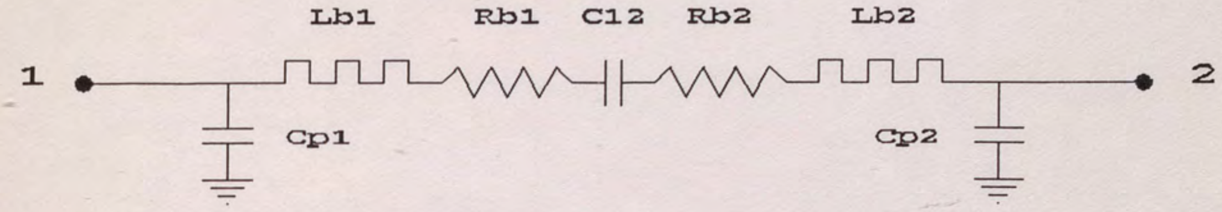

Figure 10: Package parasitic resonance equivalent circuit for single port resonators using a transmission measurement

can be applied to each subcircuit directly. The overall circuit can then be constucted by combining the subcircuits.

The actual implementation of the EIA-512 method follows the approach of Malocha, et. al [13] which is detailed in Appendix A. Two assumptions are necessary in order to apply this approach to the current problem:

1) the motional elements can be treated as open circuits at the resonant frequency of the parasitic elements, and

2) the capacitive susceptance across the series arm is a constant over the narrow extraction bandwidth.

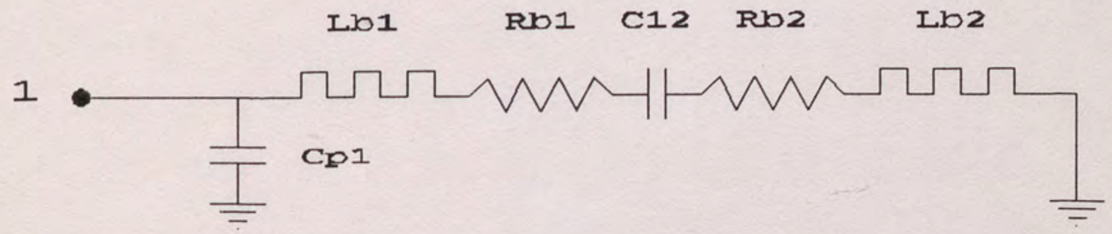

Figure 11: Package parasitic resonance equivalent circuit for single port resonators using reflection measurements

The first assumption is easy to justify due to the high Q nature of the SAW resonator. For example, the series arm of a 217 MHz, dual port resonator has a reactance of 18 MΩ at its parasitic resonance of 1.8 GHz. In contrast, the transducer capacitance that forms the parasitic resonance has a reactance of 45 Ω. The second assumption is just an approximation that is justified by the agreement of the modeled equivalent circuit response with the actual measured response.

Note that while the first assumption is valid, its converse is not necessarily acceptable. Specifically, the parasitic circuits have Qs on the order of 10 to 15; therefore, the elements can be significant in the area of the main resonance, especially for L-band resonators. For this

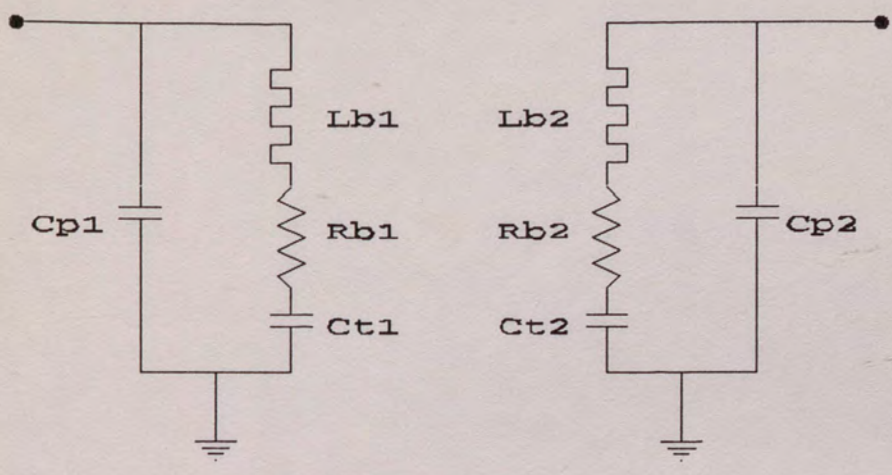

Figure 12: Package parasitic resonance equivalent circuit for dual port resonators

reason, the parasitic elements are extracted first and their effects are subtracted from the data describing the main resonance, then the motional elements are extracted. Longitudinal modes are handled in a similar fashion, although the first assumption can be more difficult to support if the mode does not rise several dB above the main response skirt. An iterative procedure would be helpful in this situation. The first assumption is the reason why transverse modes cannot be successfully extracted since they occur immediately on the high frequency skirt of the fundamental response and are greatly influenced by it.

The following steps summarize the extraction procedure:

1) Sample S-parameter data within approximately the 3 dB bandwidth of each resonance.

2) Extract C_p, L_b, R_b, and C_t.

3) Convert main resonance and longitudinal mode data to Y-parameters.

4) Subtract $B_p = 2\pi f \cdot C_p$ from the Y-parameters.

5) Convert Y- to Z-parameters and subtract $R_b + jX_{Lb}$.

6) Convert Z- to Y-parameters and subtract $B_t = 2\pi f \cdot C_t$.

7) Extract the motional elements for any longitudinal modes and subtract these elements from the remaining sets of Z-parameters.

8) Extract the motional elements for the main response.

With this approach, the capacitances adjacent to the SAW are extracted from the package parasitic data. Alternatively, the transducer capacitance C_t can be extracted from the main resonance data. The decision on which approach to take depends upon the quality of the high frequency package resonance data. If it occurs at a low enough frequency that fixturing and calibration can be accomplished very accurately and the Q of the package resonance is high enough that the narrowband approximation is easily supported, then it will likely be more accurate to obtain the value of the transducer capacitance from the high frequency data rather than the main response data. In practice, since both approaches use the same data set, it is convenient to evaluate both approches for the best accuracy.

The subtraction of the parasitic elements is described in appendix B. Note that it is not necessary to actually subtract B_t from the main resonance data. The extraction routine will produce the same results for the series arm. The source code for the extraction routine is listed in appendix C. Appendices D and E list the subtraction routines for single port and dual port models, respectively.

CHAPTER V

DATA COLLECTION

Data has been collected from several different resonators using a Hewlett Packard 8753 network analyzer [14]. All resonators were packaged in TO-39/TO-5 headers. The fixture consists of a brass housing with SMA coaxial connectors launching directly into the pin sockets. Calibration standards have been developed in order to perform a vector calibration at the pin sockets. The short standard is simply a brass block with leads soldered into it. The load consists of a brass block, a semi-rigid coaxial cable with the center conductor forming the hot pin, and a calibrated 50 ohms load that screws into the brass block and onto the center conductor at the other end of the coax cable. Because the characteristic impedance of 50 Ω is maintained throughout the cables, fixture, and load standard, the extra electrical length added by this approach is negligible. The open is a brass block with concave hemispheres milled out directly above the pin sockets. These are intended to provide a shielded open that is easily modeled mathematically. An SMA bullet is used at the cable connections for the through standard. Its electrical length is very close to that of the fixture.

The calibration standards are assumed to be identical to the HP-85033C 3.5 mm calibration kit [14]; these characteristics are built into the HP-8753 firmware. This approximation holds to over 2.5 GHz as supported by Figure 13 and Figure 14. In both of these plots, the analyzer has been calibrated using the 3.5 mm calibration kit and with the reference planes at the end of the cables. In Figure 13, the return loss of port one is plotted with the fixture calibration load in place.The return loss at 2.5 GHz is approximately 16 dB. The transmission response with a through in place is plotted in Figure 14. The loss of the fixture and through is less than 0.75 dB at 2.5 GHz.

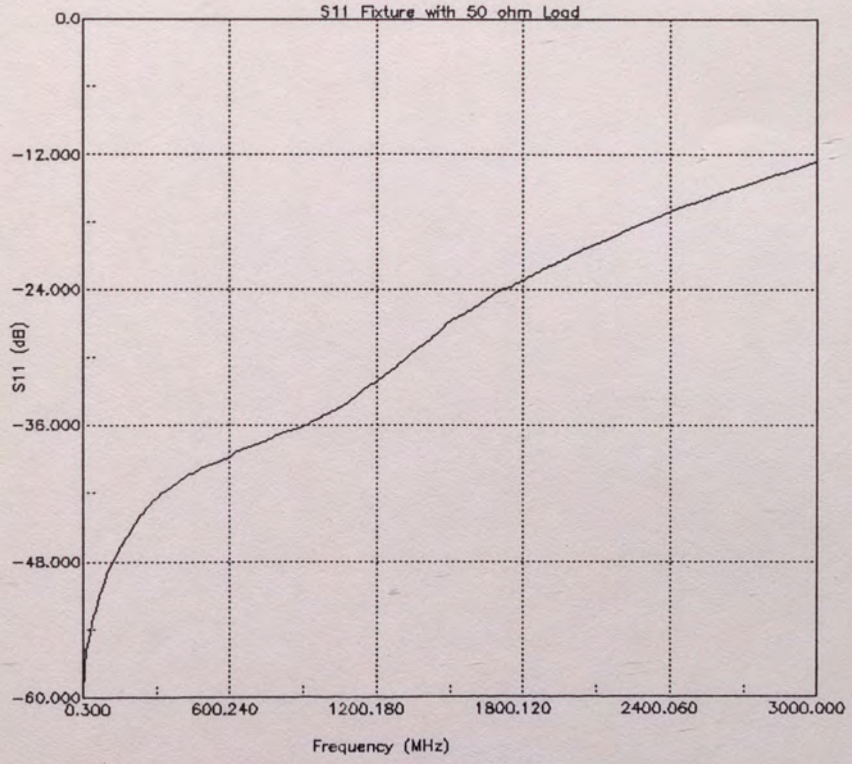

Figure 13: Fixture S11 with calibration load in place

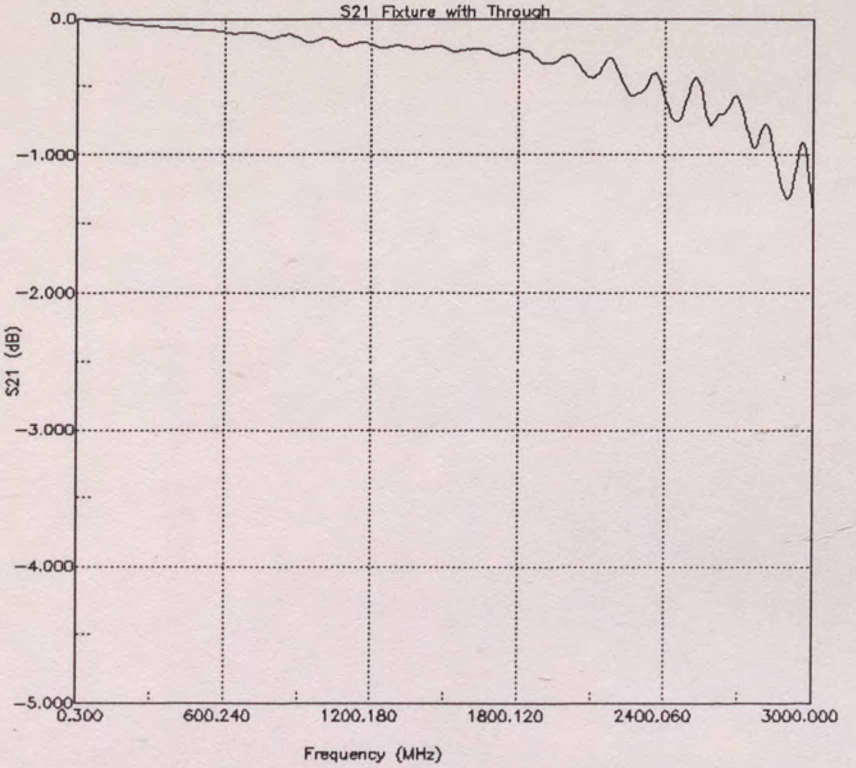

Figure 14: Fixture S21 with through in place

In the majority of tests, two sets of 801 points where sampled from the analyzer. The wideband data ranged from starting values of either 0.3 MHz or 10 MHz to a final value of 3 GHz. This provides 100 to 200 points over the narrow extraction bandwidth of the package resonance. This bandwidth was chosen by measuring the impedance magnitude at resonance and then locating the frequencies at which the impedance was twice the midband value. These frequencies defined the extraction bandwidth. The quality of the reactance versus frequency fit was evaluated graphically to determine if the bandwidth was too wide. The bandwidth was then modified and the elements were extracted again. This second extraction was rarely necessary.

The narrowband data was collected over various bandwidths centered about the main resonance. The extraction bandwidth was the 3 dB bandwidth of the resonator. The 3 dB bandwidth of a single port resonator is difficult to determine before extraction. The resulting variability of the extraction bandwidth for different devices and the good agreement of the data indicate that the actual bandwidth used is not critical.

CHAPTER VI

RESULTS

Parameter extractions have been performed on resonators with center frequecies ranging from 217 MHz to 1150 MHz. Four representive devices are presented. Two devices are single port resonators while the other two are dual port devices. In addition, two ideal test devices are presented. The data for these ideal test devices where generated by assuming typical values for each of the elements and modeling the circuits. Each of these devices will be reviewed by presenting the modeled response of the extracted circuit superimposed on the sampled data.

Single Port Resonators

Three, single port resonators are presented. The first is an ideal test device. The second is a relatively low frequency, high Q resonator at 303 MHz. The final device is a 744 MHz resonator.

Ideal Test Device

A single port test device was modeled at 200 MHz. It was evaluated in both a dual port mode (see Figure 5) and a single port mode (see Figure 7). Table I lists the component values used and the extracted results for each item under several different conditions. In Table II, the errors are listed for each item.

The package elements should resonate at 795.775 MHz. Using a single port high frequency circuit like Figure 11, the model predicts a resonance at 795.457 MHz or an error of -0.04%. This is very good agreement considering the low Q of the parasitic resonance. The individual elements agree within 3%. These elements, except C_t, were subtracted from the main response data. The full dual port model was then extracted using this data for calculating C_t. The errors for the main resonance values are all below 1%. The frequency error is only 0.21 ppm. The full dual port model was also extracted after subtracting the ideal package parasitic values from the main resonance data. The errors are now on the order of 0.1%, with the frequency error only 5 ppb. This indicates that the ability to accurately extract the package values determines the ultimate accuracy of the main resonance element values. In contrast, the basic dual port model exhibits errors as high as 44% and a frequency error of -14 ppm. Clearly, the full model is needed for high accuracy and it is necessary to exercise care in sampling and extracting the high frequency data in order to take advantage of the improved circuit.

The grounded single port model was also applied to this test device (see Figure 7). Again, main resonance extractions were performed with both ideal and extracted package parasitics and ignoring the parasitics. The results are very similar to the dual port model.

303 MHz Single Port Resonator

The equivalent circuit for the 303 MHz device is presented in Figure 15. Again, the wideband circuit is analyzed using the grounded single port high frequency model. The dual port equivalent circuit was then extracted. The transmission magnitude is presented in Figure 16 and the transmission phase is presented in Figure 17. In these and other plots, the modeled response of the equivalent circuit is given in the solid line style while the dash-dotted line style represents the raw data. Clearly, the agreement is excellent.

The susceptance versus conductance plot for the narrowband extraction is given in Figure 18. For these plots, the actual data use a solid line style while the fitted data are in a dotted line style. This data has been normalized to the midband conductance value in order to reduce roundoff errors in the extraction calculations. The reactance versus frequency data is plotted in Figure 19. Again, the agreement is excellent. Especially note the agreement of the second order fit near the root of the reactance plot. This supports the claim that a second order equation is sufficient.

The impedance magnitude for the complete model is plotted in Figure 20. The transmission response is shown in

			TABLE I: SINGLE PORT IDEAL TEST DEVICE			
ELEMENT	UNITS	ACTUAL	BASIC 2 PORT	PKG	FULL MODEL	FULL MODEL
				PARASITIC		IDEAL PKG
Cp1	pF	1.000	1.000	1.009	1.009	[1]
Lb1	nH	5.00	N/A	5.13	5.13	[5]
Rb1	ohm	5.00	N/A	4.95	4.95	[5]
C12=Ct	pF	4.000	3.088	3.902	3.980	4.000
L1	uH	596.83104	856.74059	N/A	599.35955	597.52068
C1	fF	1.061033	0.739168	N/A	1.056556	1.059808
R1	ohm	30.0	35.9	N/A	30.2	30.0
Lb2	nH	5.00	N/A	5.13	5.13	[5]
Rb2	ohm	5.00	N/A	4.95	4.95	[5]
Cp2	pF	1.000	1.000	1.009	1.009	[1]
Fs	MHz	200.000000	199.997155	795.457000	200.000042	200.000001
Qu		25000	30021	5	24948	25028

Actual: Typical component values of a one port resonator with package parasitics. They are used to generate a file of S-paramters to test the extraction routine.

DUAL PORT MODE

Basic 2 Port: Extracted element values neglecting the package parasitics.

PKG Parasitic: Extracted package parasitic values.

Full Model Extracted PKG: Extracted element values using the extracted package parasitic elements. The transducer capacitance is extracted from the main resonance data.

Full Model Ideal PKG: Extracted elements using actual values for the parasitics.

	TABLE I: CONTINUED		
ELEMENT	BASIC 1 PORT	FULL 1 PORT	FULL 1 PORT
		EXTRACTED PKG	IDEAL PKG
Cp1	N/A	1.009	[1]
Lb1	N/A	5.30	[5]
Rb1	N/A	4.95	[5]
C12=Ct	4.088	3.970	4.000
L1	846.10399	599.19348	597.43711
C1	0.748454	1.056849	1.059957
R1	35.9	30.2	30.0
Lb2	N/A	5.13	[5]
Rb2	N/A	4.95	[5]
Cp2	N/A	N/A	N/A
Fs	199.998046	200.000048	200.000005
Qu	29648	24948	250285

SINGLE PORT MODE

Basic 1 port: Extracted element values neglecting the package parasitics.

Full 1 Port Extracted PKG: Extracted element values using the extracted package parasitic elements. The transducer capacitance is extracted from the main resonance data.

Full 1 Port Ideal PKG: Extracted element values using the actual known values for the package elements.

TABLE II: SINGLE PORT IDEAL TEST DEVICE ERRORS

ELEMENT	UNITS	ACTUAL	BASIC 2 PORT	PKG	FULL MODEL	FULL MODEL
				PARASITIC		IDEAL PKG
Cp1	pF	1	0.00%	9.00%	9.00%	--
Lb1	nH	5	N/A	2.61%	2.61%	--
Rb1	ohm	5	N/A	-1.00%	-1.00%	--
C12=Ct	pF	4	-22.80%	-2.46%	-0.50%	0.00%
L1	uH	596.83104	43.5483%	N/A	0.4237%	0.1156%
C1	fF	1.061033	-30.3350%	N/A	-0.4219%	-0.1155%
R1	ohm	30	19.67%	N/A	0.67%	0.00%
Lb2	nH	5	N/A	9.00%	2.61%	--
Rb2	ohm	5	N/A	2.61%	-1.00%	--
Cp2	pF	1	0.00%	-1.00%	0.90%	--
Fs	MHz	200	-14.2 ppm	--	0.210 ppm	0.005 ppm
Qu		25000	20.08%	--	-0.21%	0.11%

	TABLE II: CONTINUED		
ELEMENT	BASIC 1 PORT	FULL 1 PORT	FULL 1 PORT
		EXTRACTED PKG	IDEAL PKG
Cp1	N/A	0.90%	--
Lb1	N/A	2.16%	--
Rb1	N/A	-1.00%	--
C12=Ct	2.20%	-0.75%	0.00%
L1	41.7661%	0.3958%	0.1015%
C1	-29.4599%	-0.3943%	-0.1014%
R1	19.67%	0.67%	0.00%
Lb2	N/A	2.61%	--
Rb2	N/A	-1.00%	--
Cp2	N/A	N/A	N/A
Fs	-9.77 ppm	0.240 ppm	0.025 ppm
Qu	18.59%	-0.21%	0.10%

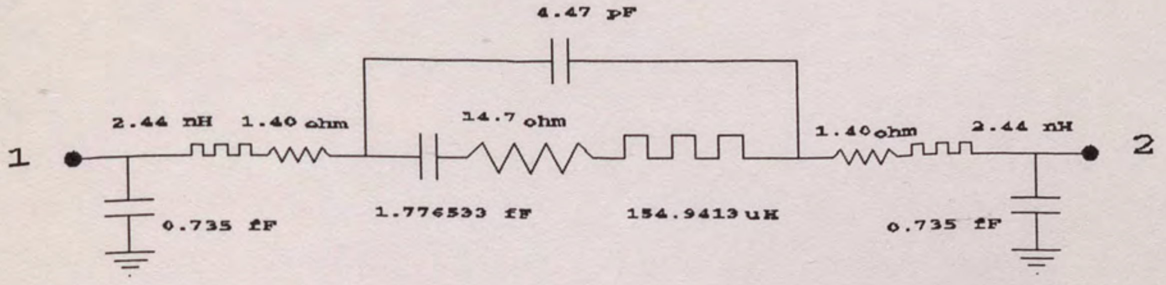

Figure 15: 303 MHz equivalent circuit -- dual port model

$$\begin{aligned} 2\ \mathit{Port}\colon\quad Q_L &= Q_u\frac{R_1}{R_L+R_S+R_1} \\ 1\ \mathit{Port}\colon\quad Q_L &= Q_u\frac{R_1}{R_S+R_1} \end{aligned} \tag{1}$$

Figure $\overset{21}{23}$. Note that it is difficult to determine the package resonance frequency from the transmission response. This is because in the dual port configuration, the resonator is loaded by 50 Ω on both ports instead of just one as on the single port configuration. Equation (1) clarifies the relationships between loaded and unloaded Q.

In this situation, $R_1 = R_b = 1.4\ \Omega$. The source and load impedances dominate the equation and the loaded Q of the dual port model is one-half that of the single port model.

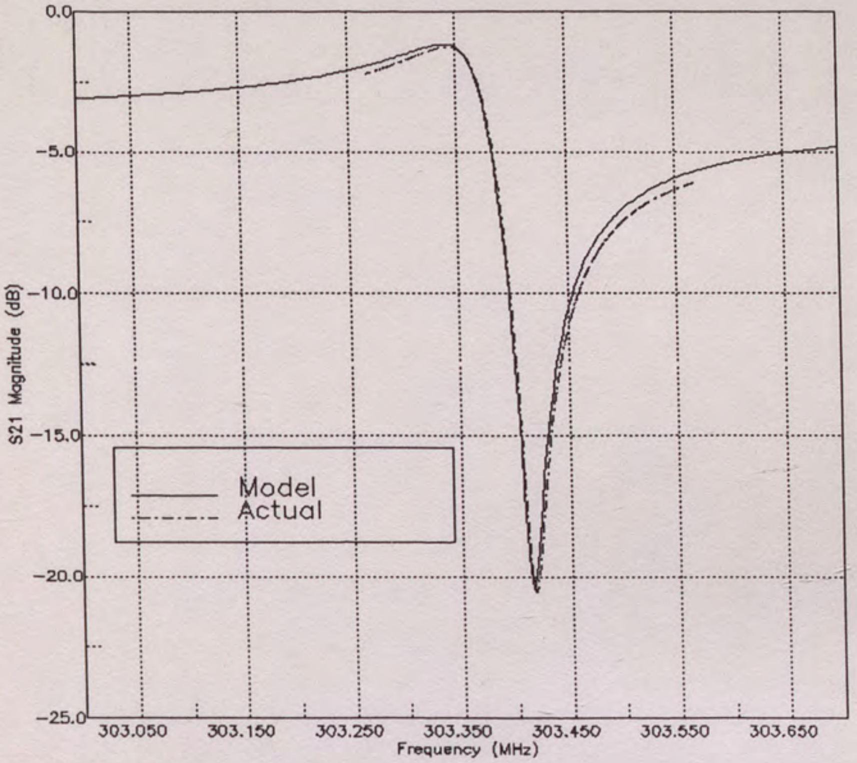

Figure 16: 303 MHz single port resonator in dual port model.

As mentioned previously, some of the capacitors can be determined from either the parasitic resonance data or the main resonance data. The bridging transducer capacitance, C_t, was determined from the parasitic data for Figure 16. On the other hand, the EIA-512 based dual port extraction yields this same capacitor and also C_{10} and C_{20} of Figure 9. In a single port device with the package capacitance removed, these capacitors represent an extremely small amount of stray capacitance. The extraction predicts stray capacitances of approximately 0.6 pF and a transducer capacitance of 2.76 pF. Even though the model is more complete, the agreement is worse in both the main resonance, Figure 22, and the wideband

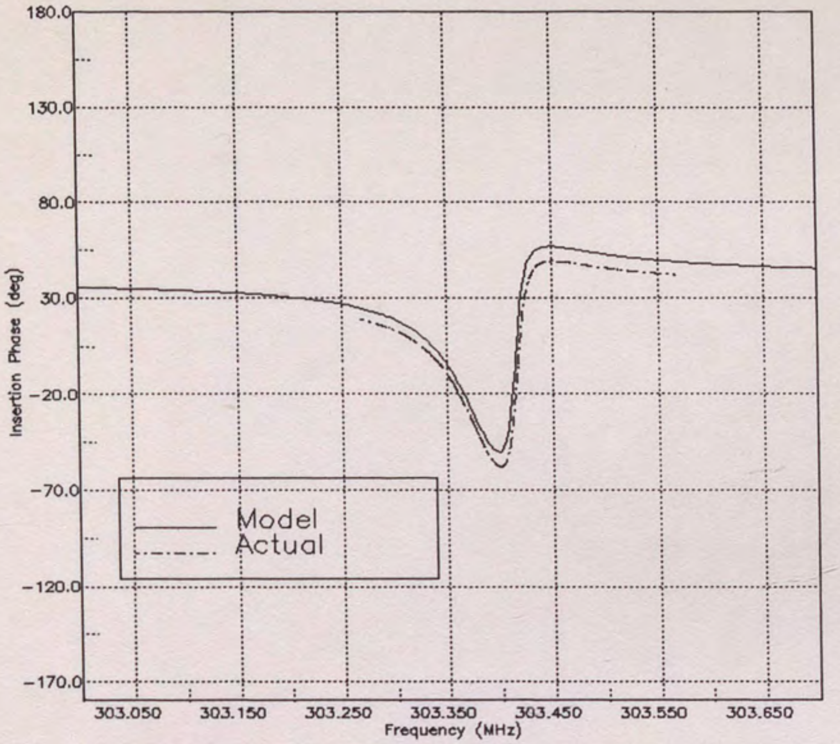

Figure 17: 303 MHz single port resonator transmission phase

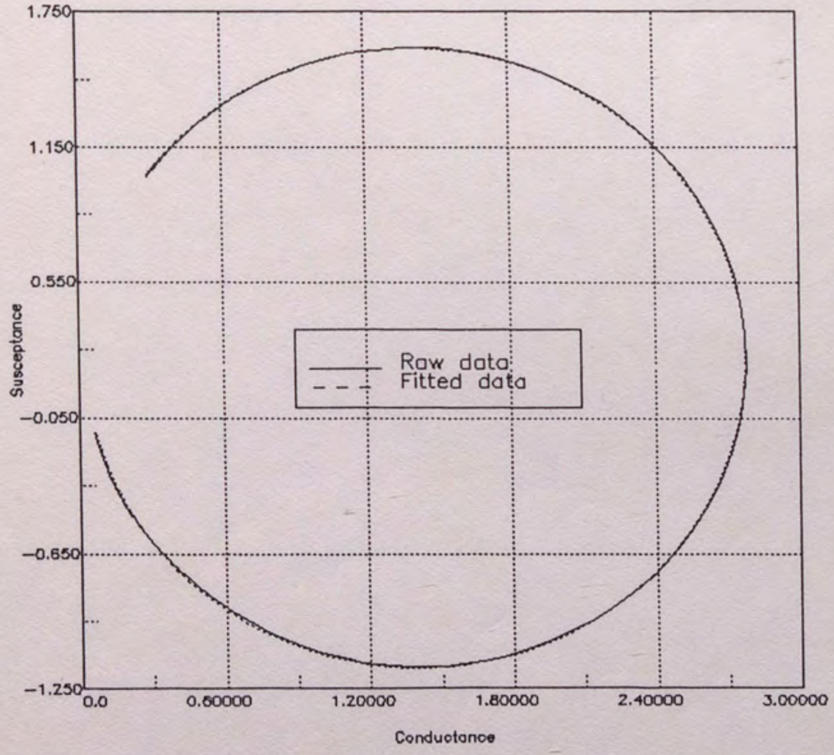

Figure 18: Susceptance versus Conductance for 303 MHz

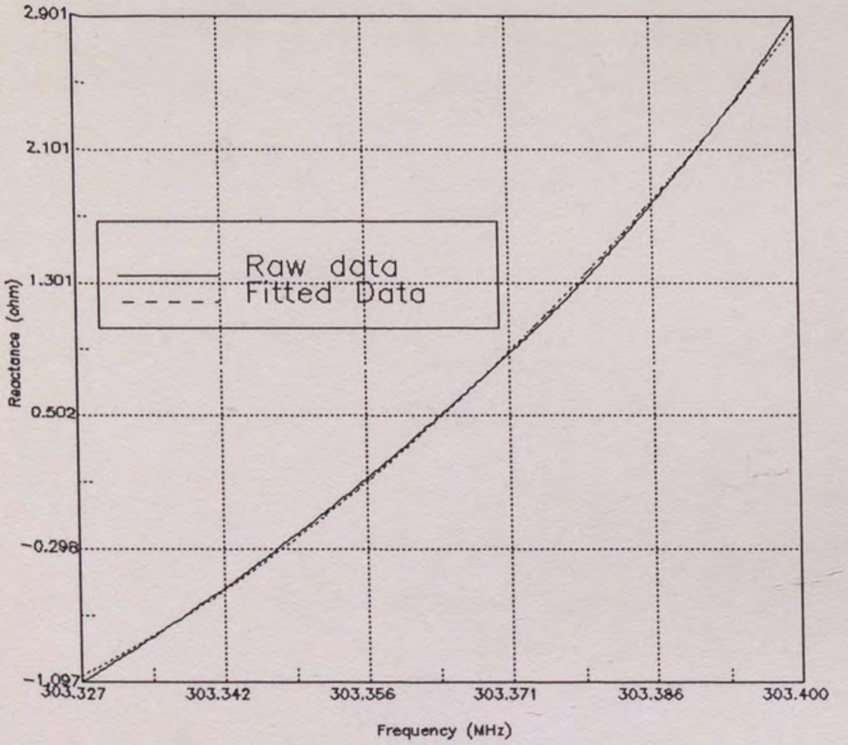

Figure 19: Reactance versus frequency 303 MHz

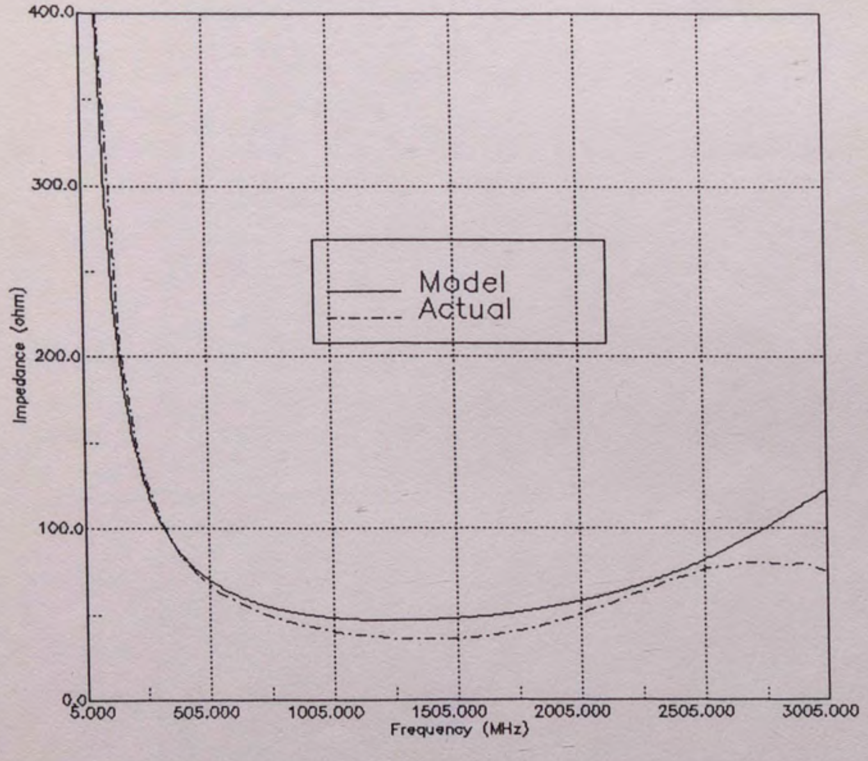

Figure 20: Wideband impedance 303 MHz resonator

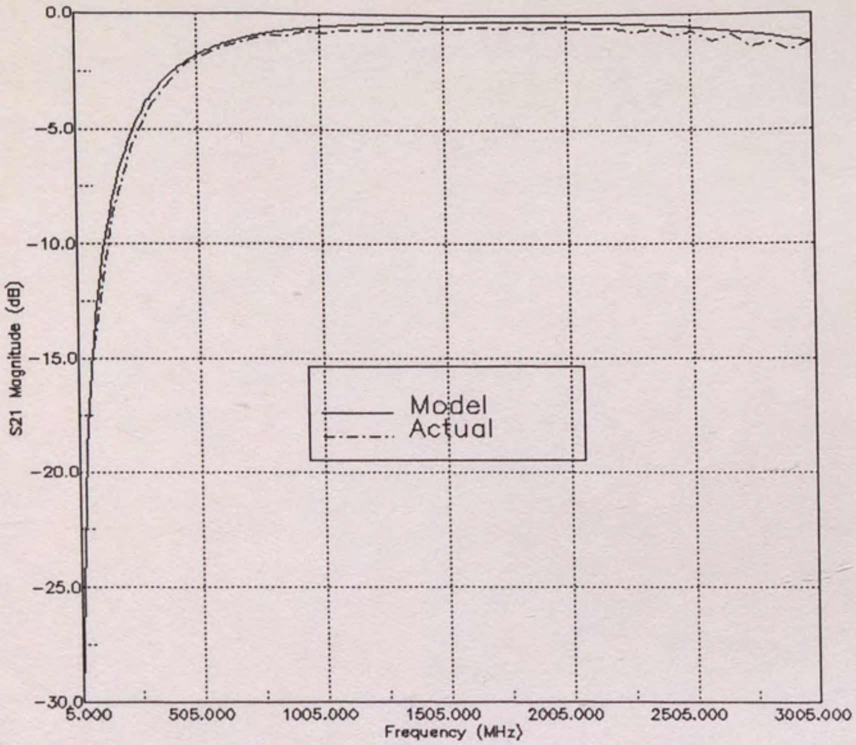

Figure 21: Wideband transmission 303 MHz resonator

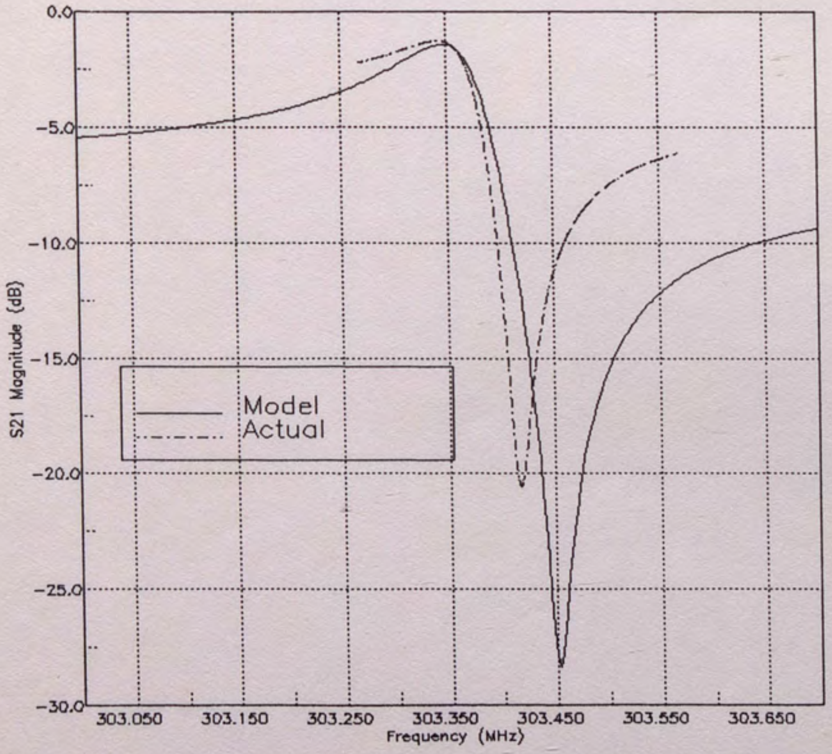

Figure 22: 303 MHz resonator with stray capacitance

response, Figure 23. It is therefore important to use only the degree of complexity which can be accurately extracted.

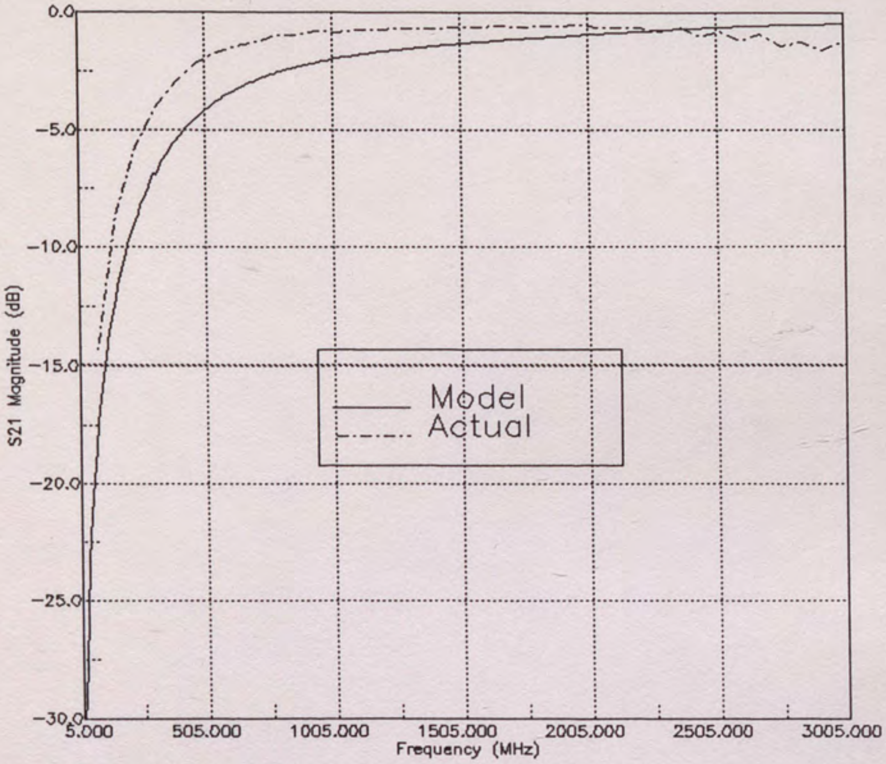

Figure 23: Wideband response 303 MHz resonator with stray capacitance

The resonator can also be modeled in the single port configuration of Figure 7. The agreement for this model is also excellent for both the main response, Figure 24, and the wideband response, Figure 25.

744 MHz Single Port Resonator

Higher frequency resonators are more challenging to extract for several reasons. Because the transducer capacitance is smaller, the package resonance is higher than

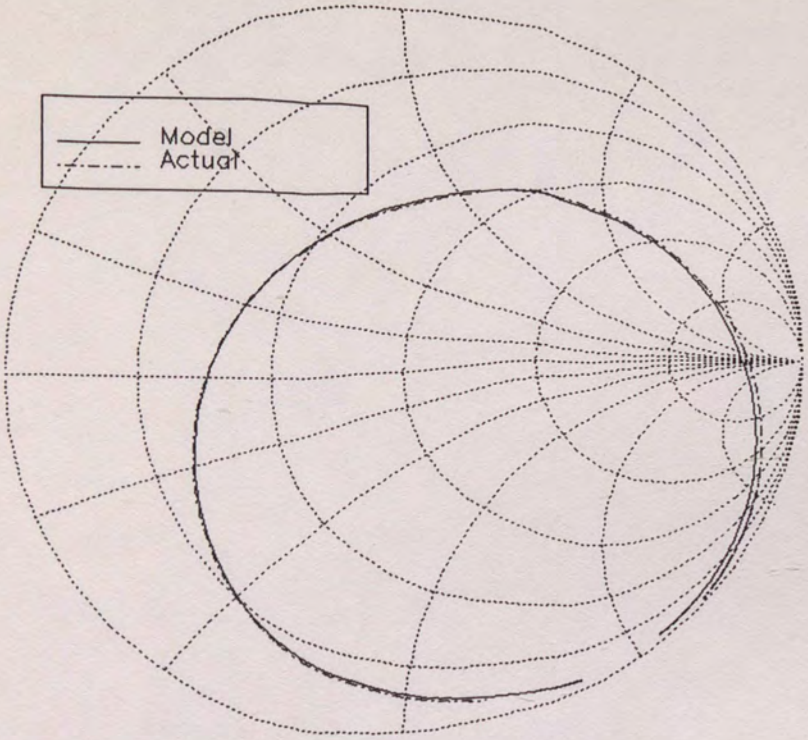

Figure 24: 303 MHz resonator using single port model

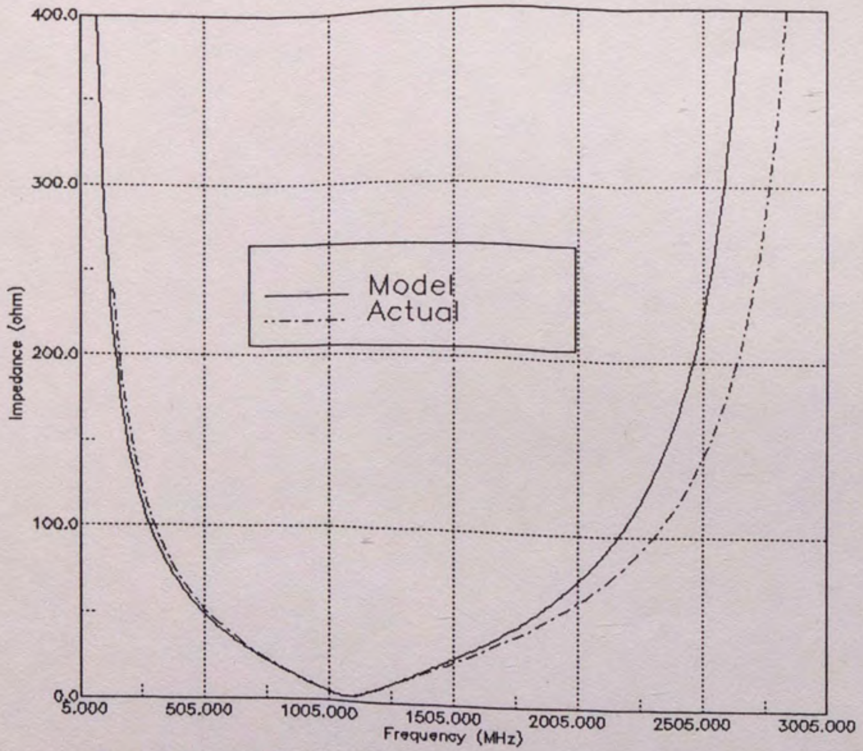

Figure 25: Wideband response 303 MHz using single port model

low frequency resonators. This places greater demands on the fixturing and calibration. Compounding the problem is the fact that the package capacitance and bond wire inductance remain the same. The bondwire reactance and the package susceptance, therefore increase. This ultimately means that greater accuracy is needed in extracting the package related elements in order to be able to accurately extract the main resonance values.

The extracted equivalent circuit for the 744 MHz single port resonator operated as a dual port device is diagrammed in Figure 26. The 744 MHz single port resonator has a package parasitic resonance at 1947 MHz. Figure 27 illustrates the susceptance versus conductance plot for this resonance and Figure 28 describes the reactance versus frequency fit. Both fits appear to be excellent.

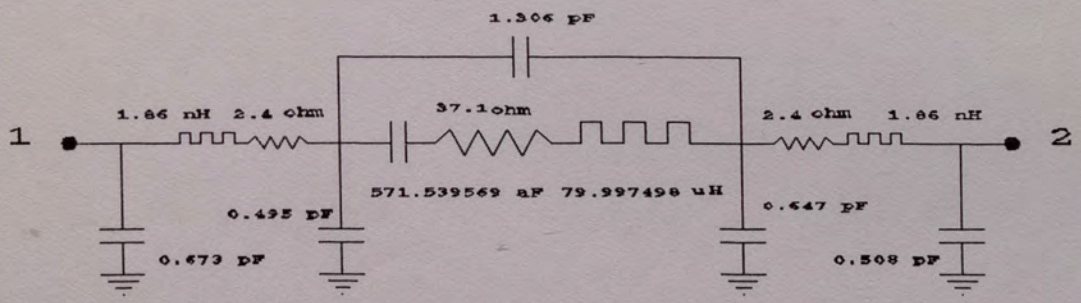

Figure 26: 744 MHz single port resonator equivalent circuit using dual port model

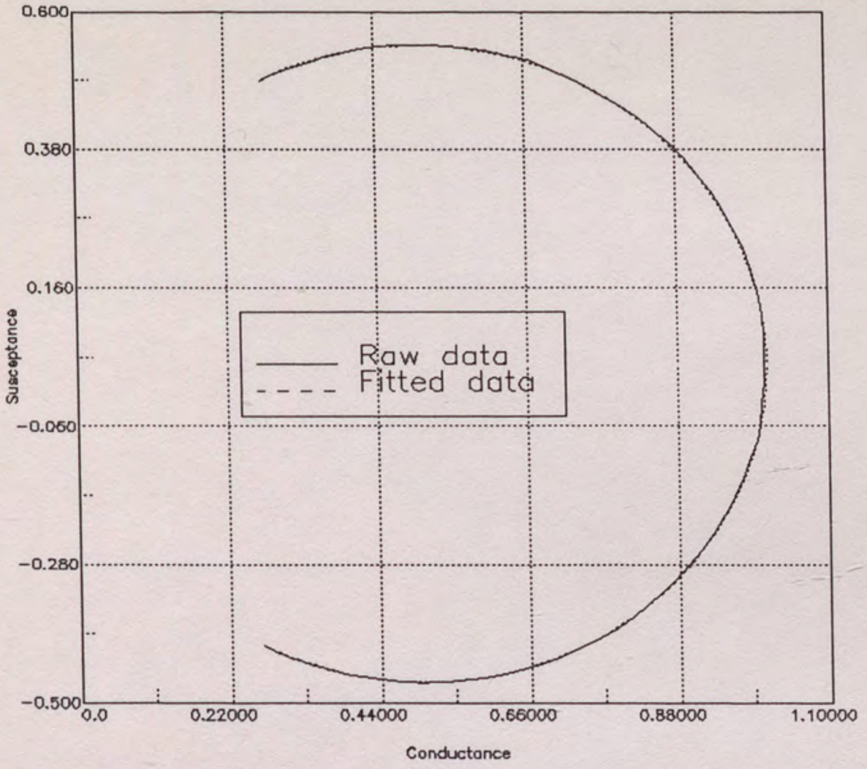

Figure 27: Susceptance versus conductance 744 MHz resonator

Figure 29 plots the transmission magnitude of the main response. The phase response is graphed in Figure 30. The package parasitic extraction predicts a transducer capacitance of 1.8 pF. Extracting from the main resonance data produces a transducer capacitance of 0.727 pF and stray capacitances C_{10}=0.5 pF, C_{20}=0.6 pF. The latter is plotted in the dotted line style. In the high frequency extraction C_{10} (or C_{20}) is in parallel with C_t; therefore, when the dual port circuit is constructed, the stray capacitance determined from the main resonance data is subtracted from the measured transducer capacitance to yield the true transducer capacitance. This is plotted in the solid line style and is clearly superior to

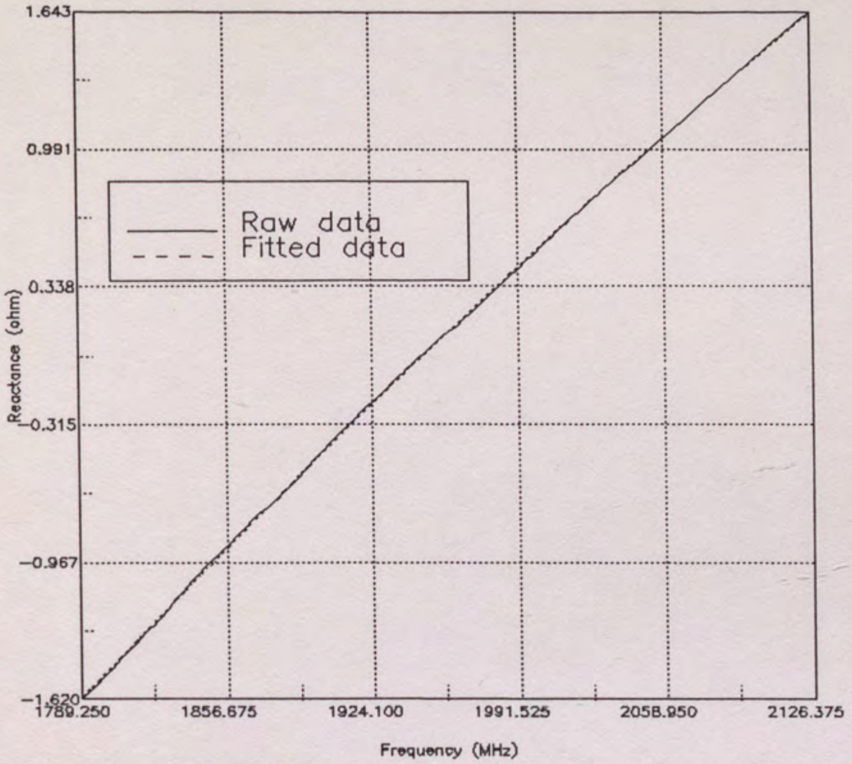

Figure 28: Reactance versus frequency 744 MHz resonator

relying solely on the main resonance data. S_{21} of the actual device is plotted in the dash-dotted line style.

As in the 303 MHz case, extracting the capacitance from the high frequency data produces better agreement. Note that in the high frequency extraction, the transducer capacitance is equivalent to C_1 of the EIA circuit, while in the main resonance, it is equivalent to C_{12}. In the ideal test case, the opposite ocurred: smaller error was acheived from the main resonance extraction (-0.50% and 0.00%) than the high frequency extraction (-2.46%) for the transducer capacitance. This suggests that there may be some parasitic effect in actual devices that has not been accounted for in the model.

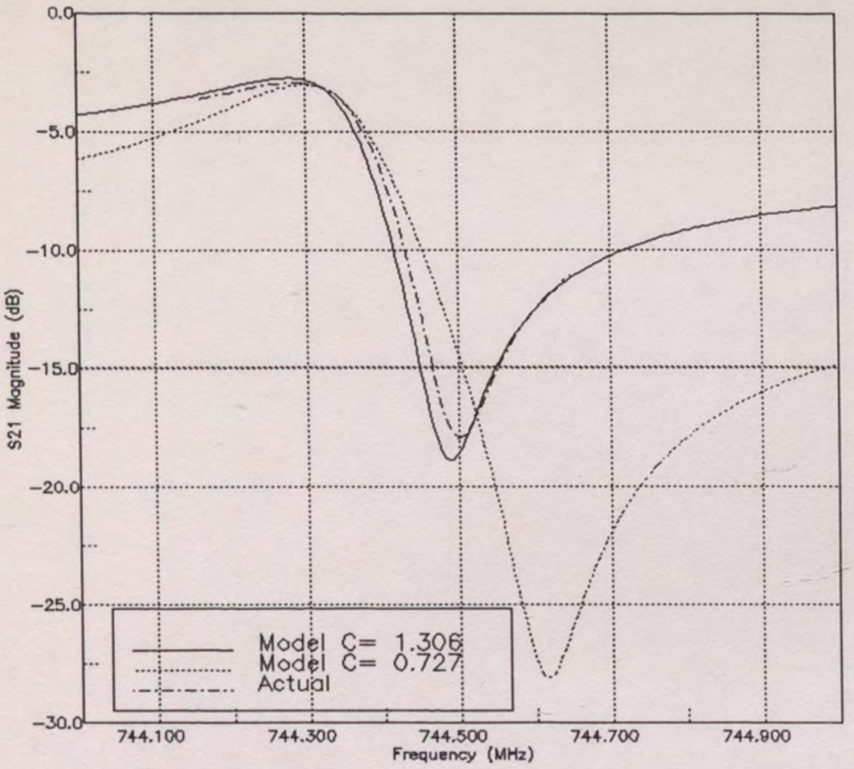

Figure 29: 744 MHz single port resonator

The quality of the fit even in the best case is not quite as good as the 303 MHz device. The single port model in Figure 31 also indicates some degradation from the lower frequency resonator. Figure 32 and Figure 33 shed some light on this matter. The order of the fit for the reactance in this situation could be higher. Alternatively, rather than increasing the order of the fit, the range of the fit could be limited to the region very near the root. In this case, a second order fit should yield excellent results.

Dual Port Resonators

As in the single port case, an ideal test resonator at 200 MHz is evaluated. The complete response of a 217 MHz

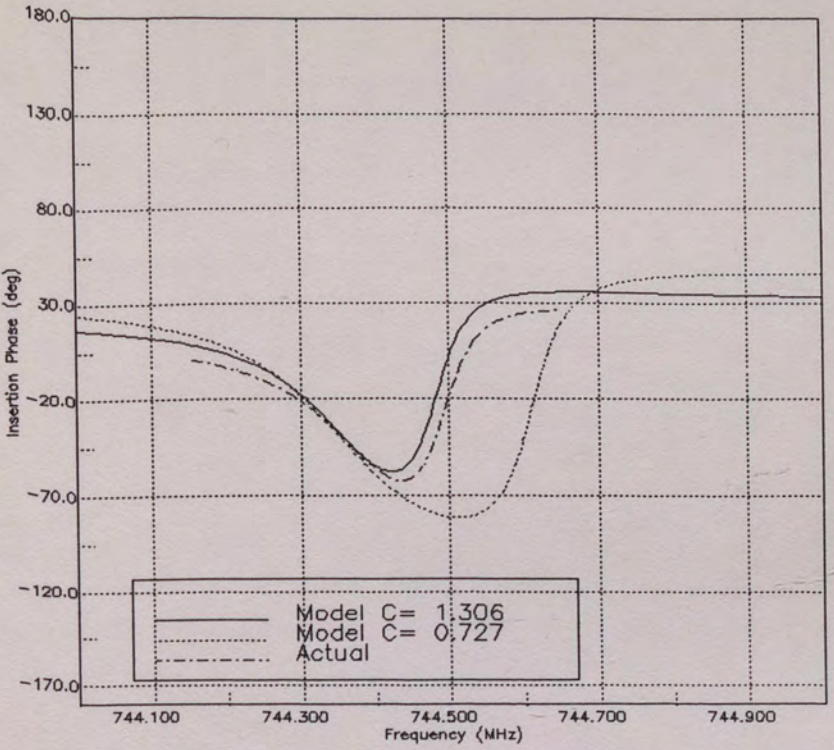

Figure 30: 744 MHz single port resonator phase response

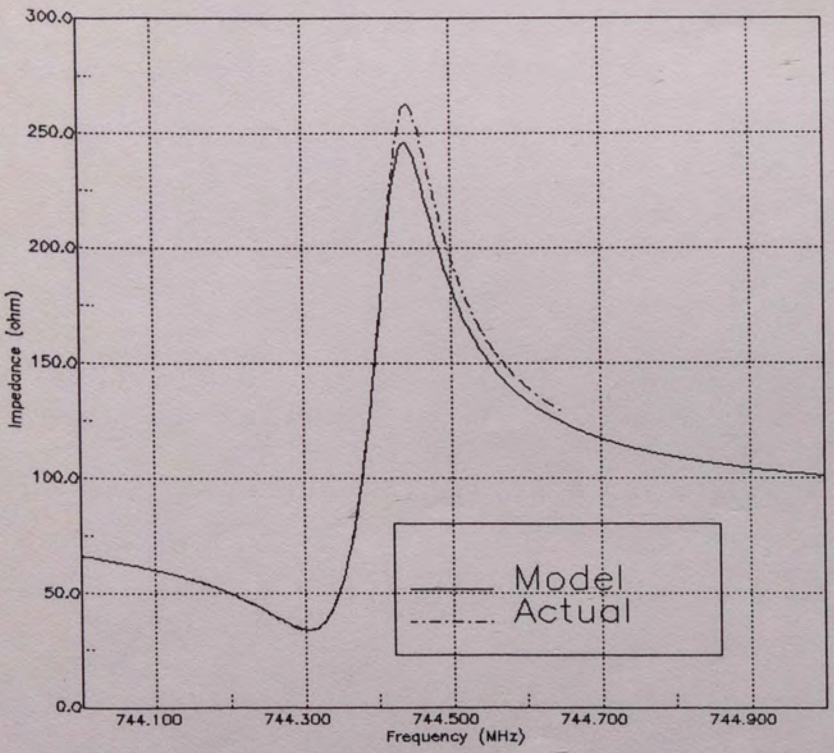

Figure 31: 744 MHz resonator in single port configuration

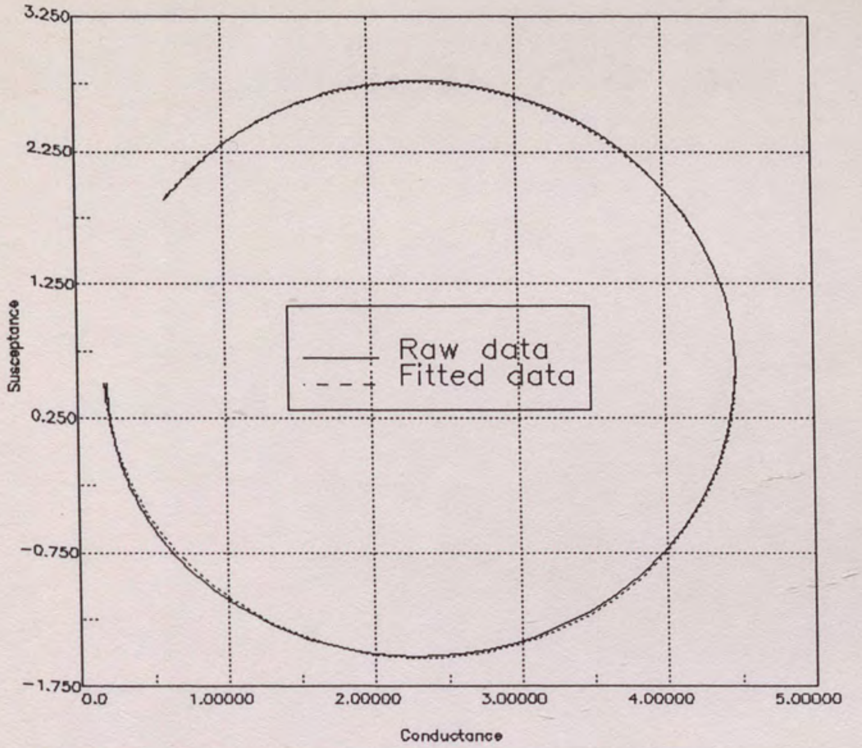

Figure 32: Susceptance versus conductance 744 MHz resonator

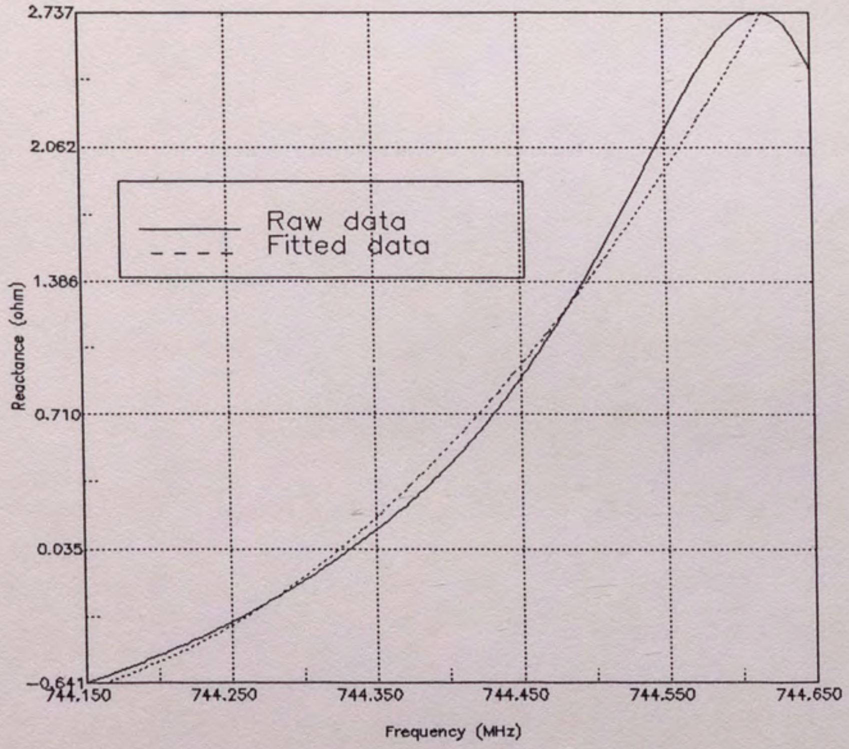

Figure 33: Reactance versus frequency 744 MHz resonator

resonator, including longitudinal modes, is modeled from extracted elements. Finally, a 567 MHz device is presented. Because a dual port resonator has approximately half as much transducer capacitance than a single port device, the package parasitic resonance is $\sqrt{2}$ times higher. Although higher frequency dual port resonators have been extracted, the agreement degrades because of the practical difficulties of extracting the parasitic resonance elements above 2 GHz.

Ideal Test Device

The test device for the dual port resonator was also designed at 200 MHz. The results of the various extractions are summarized in Table III for the element values and in Table IV for the errors. Again, the basic model exhibits significant deviations for all elements. The package parasitics are extracted with errors on the order of 1%. Using these values, the errors in the main resonance element values are at most 0.034%. Even more impressive, when the ideal package values are used in the subtraction process, the maximum error in the main resonance element value is 0.0014%. The agreement of the series resonant frequencies follow the same tendencies. The frequency of the basic circuit is 9.5 ppm too low. On the other hand, the full model is only 0.1 ppm high using the extracted package elements and is within 2.5 ppb of the actual frequency when using the ideal package elements.

TABLE III: DUAL PORT IDEAL TEST DEVICE

ELEMENT	UNITS	ACTUAL	BASIC	PKG	FULL MODEL	FULL MODEL
				PARASITIC	EXTRACTED PKG	IDEAL PKG
Cp1	pF	1.000	N/A	1.003	1.003	[1]
Lb1	nH	5.00	N/A	5.06	5.06	[5]
Rb1	ohm	5.00	N/A	5.00	5.00	[5]
Ct1	pF	2.000	3.032	1.978	1.997	2.000
L1	uH	596.83104	580.22799	N/A	597.03398	596.83953
C1	fF	1.061033	1.091415	N/A	1.060672	1.061018
R1	ohm	120.0	126.3	N/A	120.0	120.0
Ct2	pF	2.000	3.032	1.978	1.997	2.000
Lb2	nH	5.00	N/A	5.06	5.06	[5]
Rb2	ohm	5.00	N/A	5.00	5.00	[5]
Cp2	pF	1.000	N/A	1.003	1.003	[1]
Fs	MHz	200.000000	199.998103	1591.046847	200.000020	200.000000
Qu		6250	5773	10	6250	6250

Actual: Typical component values of a two port resonator with package parasitics. They are used to generate a file of S-parameter to test the extraction routine.

Basic: Extracted element values neglecting the package parasitics.

PKG Parasitic: Extracted package parasitic values.

Full Model Ideal PKG: Extracted element values using the actual known values of the package elements.

Full Model Extracted PKG: Extracted element values using the extracted package parasitic elements. The transducer capacitance is extracted from the main resonance data.

TABLE IV: DUAL PORT IDEAL TEST DEVICE ERRORS

ELEMENT	UNITS	ACTUAL	BASIC	PKG	FULL MODEL	FULL MODEL
				PARASITIC	EXTRACTED PKG	IDEAL PKG
Cp1	pF	1.000	N/A	0.30%	0.30%	--
Lb1	nH	5.00	N/A	1.20%	1.20%	--
Rb1	ohm	5.00	N/A	0.00%	0.00%	--
Ct1	pF	2.000	51.60%	-1.10%	-0.15%	0.00%
L1	uH	596.83104	-2.78%	N/A	0.0340%	0.0014%
C1	fF	1.061033	2.86%	N/A	-0.0340%	-0.0014%
R1	ohm	120.0	5.25%	N/A	0.00%	0.00%
Ct2	pF	2.000	51.60%	-1.10%	-0.15%	0.00%
Lb2	nH	5.00	N/A	1.20%	1.20%	--
Rb2	ohm	5.00	N/A	0.00%	0.00%	--
Cp2	pF	1.000	N/A	0.30%	0.30%	--
Fs	MHz	200.000000	-9.5 ppm	--	0.1 ppm	0.0 ppm
Qu		6250	-76.00%	--	0.00%	0.00%

217 MHz Dual Port Resonator

The study of the 217 MHz dual port resonator presents an opportunity to investigate the extraction of longitudinal modes. The modes in this device are clearly defined and can be extracted without resorting to subtracting the effects of it or the main response. The final circuit is given in Figure 34. Note that each mode is represented by its own series arm. The main resonance has a nominal 180° phase shift which is indicated by the ideal phase shifter. The longitudinal modes therefore have a nominal 0° insertion phase.

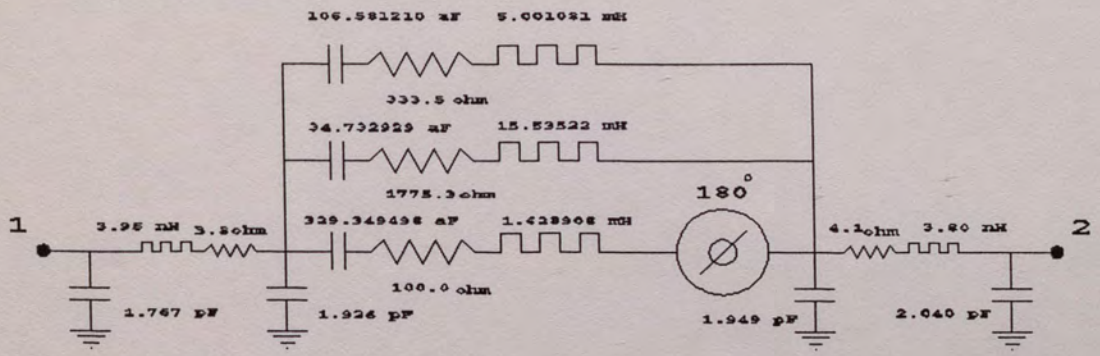

Figure 34: Equivalent circuit 217 MHz dual port 180° resonator with longitudinal modes

Figure 35 illustrates the fit of the main response over a relatively wide range, while Figure 36 zooms in on the extraction bandwidth. Because the dual port resonator does not have a transducer capacitance in the transmission path,

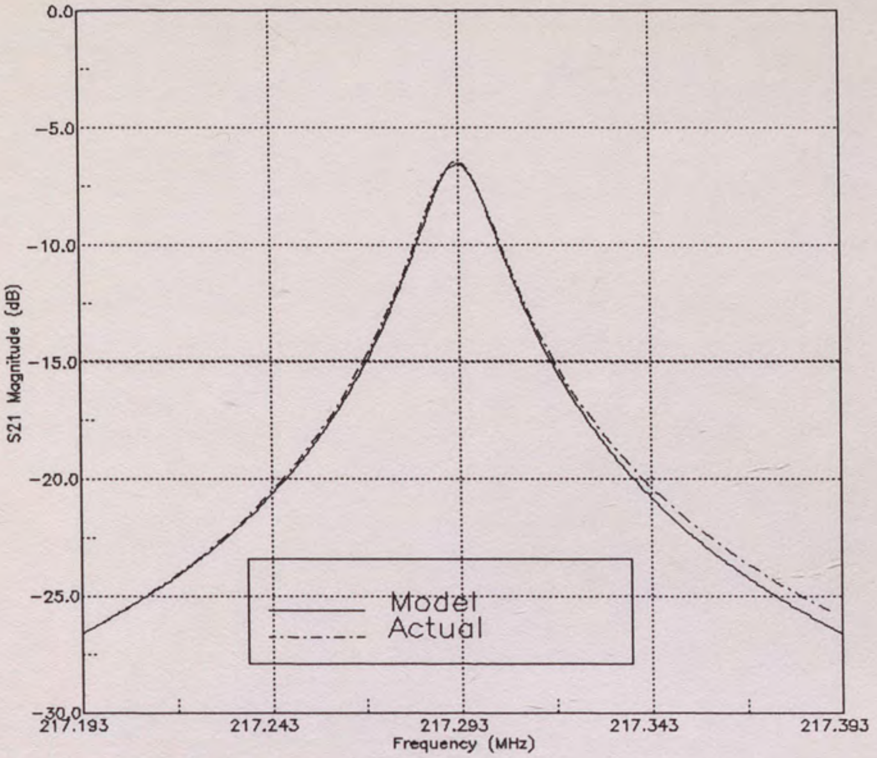

Figure 35: 217 MHz dual port 180° resonator

the passband response is much better defined than the single port device.

The circuit was constructed as usual. The package parasitic elements and the transducer capacitance where extracted from wideband impedance measurements at 1825 MHz to 1850 MHz using the circuit of Figure 12. Figure 37 shows the wideband fit. It appears to be characteristic of dual port devices that the model underestimates the impedance below the parasitic resonance and overestimates on the high frequency side. In addition, the anti-resonance near 2.65 GHz is damped out in the actual device. Perhaps there is a finite resistance path to ground that has not been added to the model.

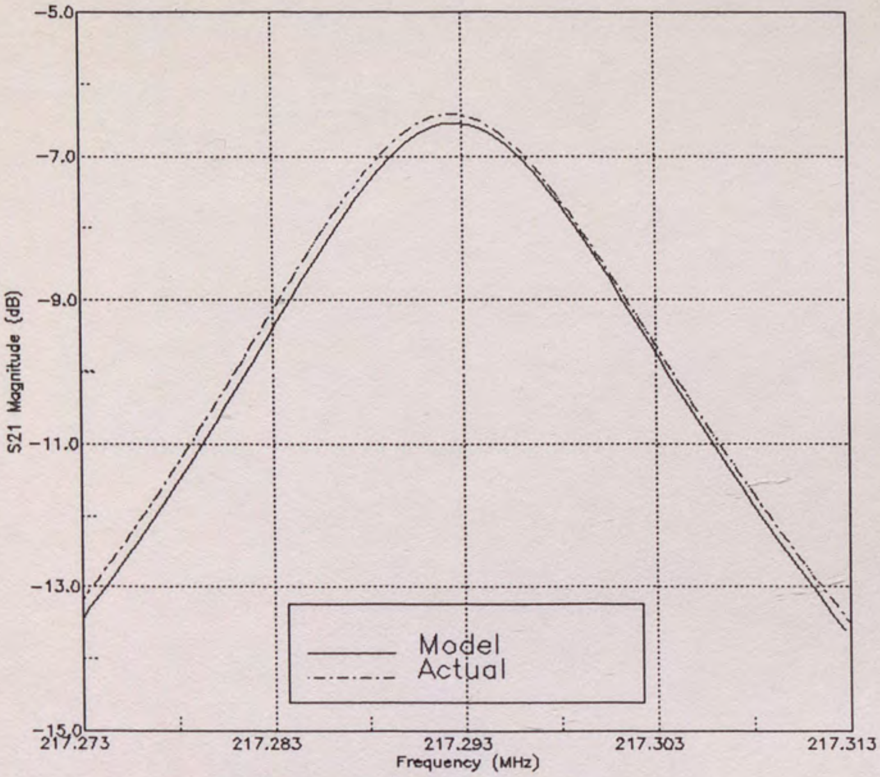

Figure 36: 217 MHz dual port 180° resonator narrowband view

By extracting each longitudinal mode separately, the composite response may be constructed. Figure 38 graphs the modeled and actual response. The agreement is excellent within the reflector stopband. The quantization in the modeled response is due to the software package used to combine the three resonant branches together and is not a reflection on the model itself. The transition between the main response and the lower longitudinal mode is handled nicely. There is an interfering mode in between the main resonance and the higher longitudinal mode. This is possibly a transverse mode. The responses below 216.5 MHz and above 218.5 MHz are due to the sidelobes of the reflector stopband

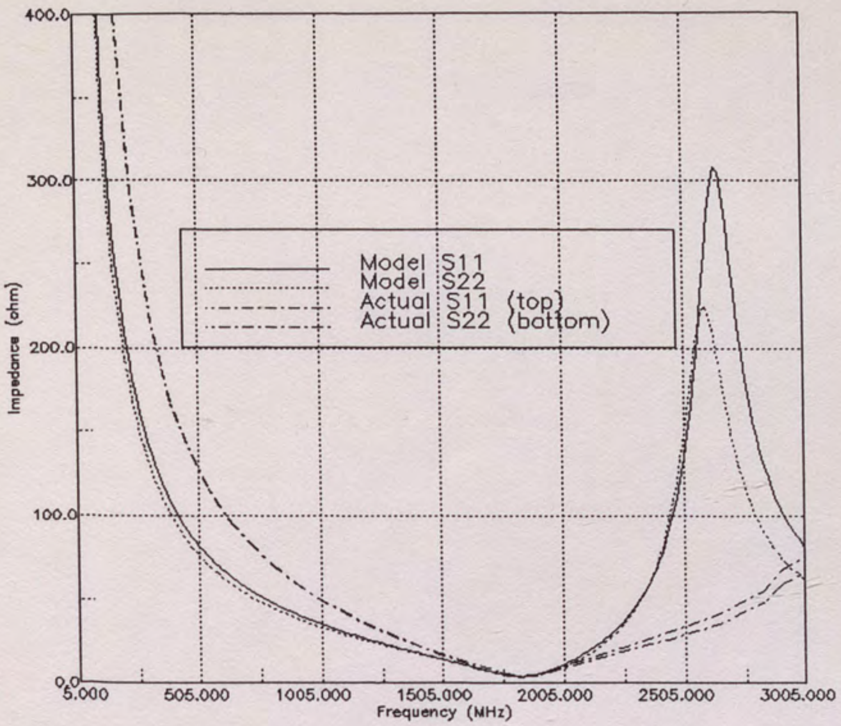

Figure 37: Wideband impedance magnitude response 217 MHz dual port 180° resonator

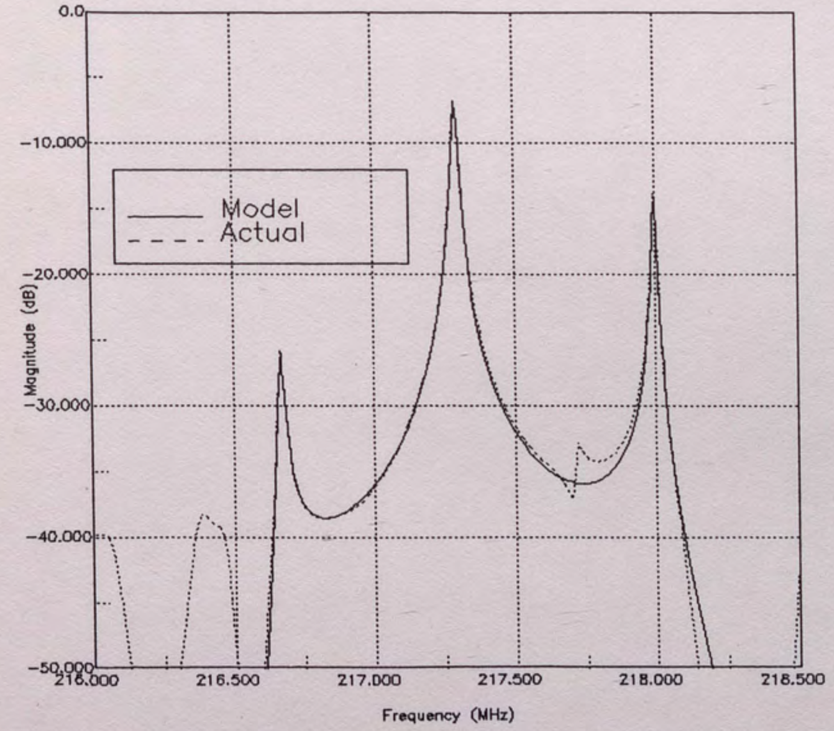

Figure 38: 217 MHz dual port 180° resonator with longitudinal modes

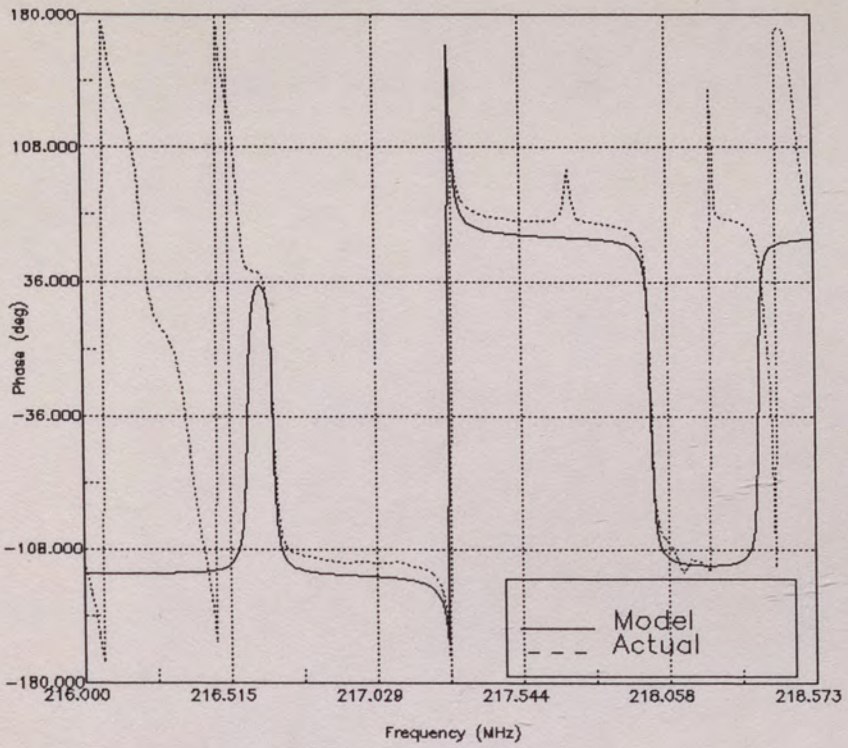

Figure 39: Phase response 217 MHz dual port 180° resonator with longitudinal modes

which is not modeled. Figure 39 illustrates that the insertion phase agreement is also very good.

567 MHz Dual Port Resonator

The 567 MHz resonator exhibits the highest loss of all the resonators presented herein. It is therefore natural that it also has the greatest error in this area, amounting to nearly 0.3 dB. In all other respects, it has characteristics similar to the devices previously discussed. The equivalent circuit is presented in Figure 40. The passband response is plotted in Figure 41 and Figure 42. Finally, the wideband impedance magnitude is graphed in Figure 43.

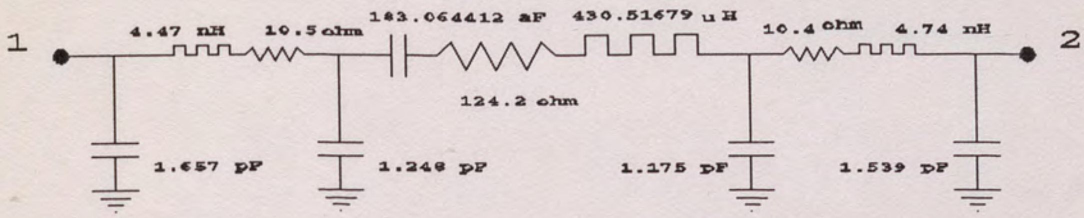

Figure 40: 567 MHz dual port resonator equivalent circuit

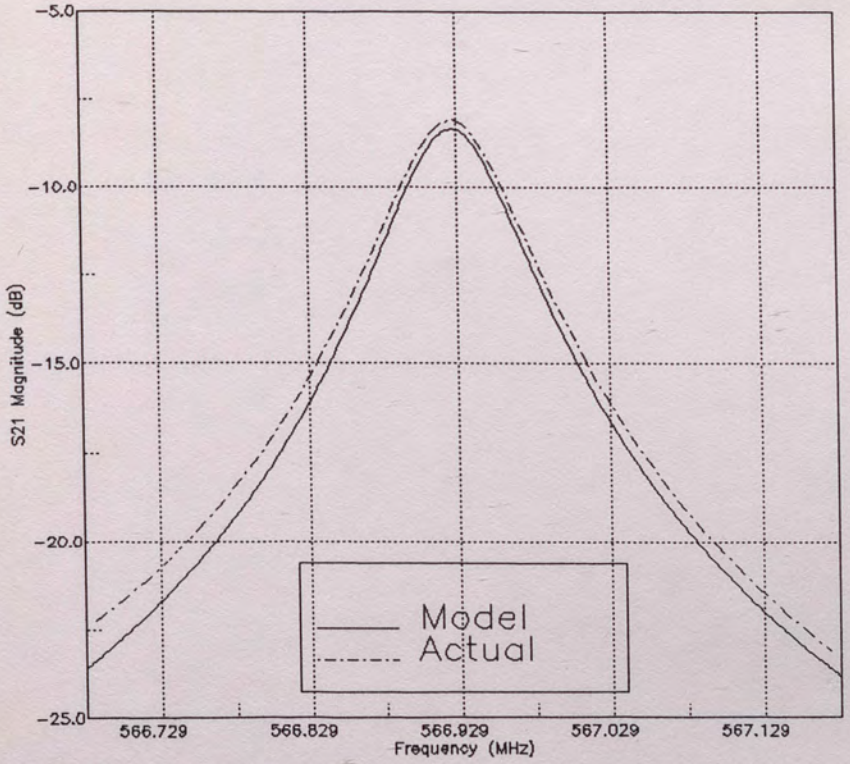

Figure 41: 567 MHz dual port resonator

Because the circuit of Figure 6 does not include C_{12}, the wideband response of a dual port resonator is not accurately

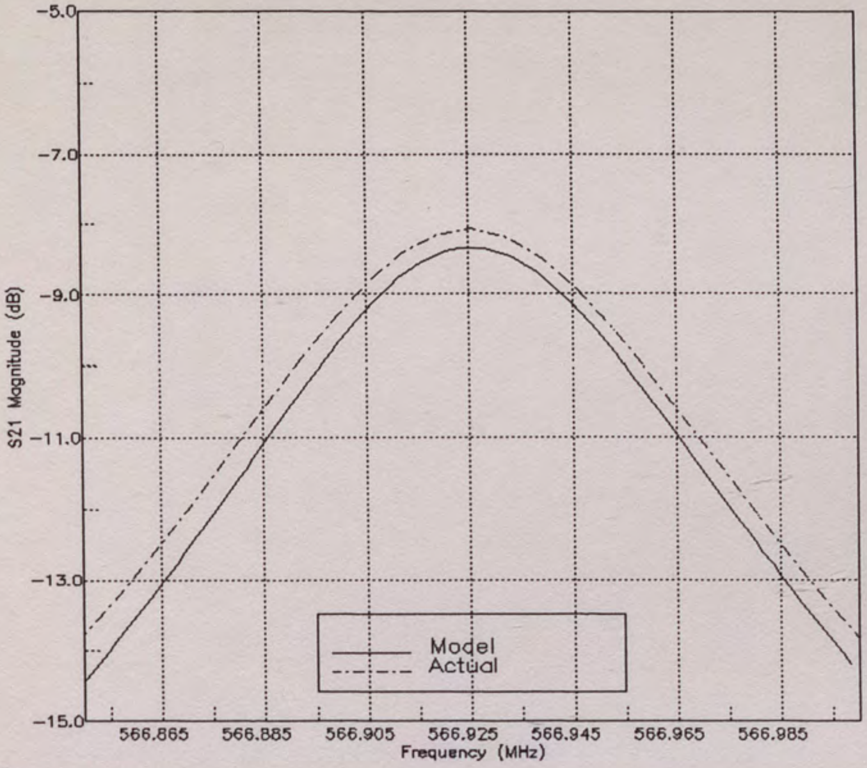

Figure 42: 567 MHz dual port resonator narrowband view

modeled. In actuality, the EIA extraction procedure does produce a value for this element; however, in a SAW device, the value is so low that it is susceptable to roundoff errors and measurement inaccuracies. In fact, for this device, the result is -0.241 pF. Figure 44 illustrates the susceptance versus conductance plot of the main resonance. Even though the coordinates are normalized, it is clear that the circle's center is below the real axis. It is not likely that this is representative of the true device. Rather, the electrical length mismatch between the calibration through standard and the actual electrical length of the fixture results in slightly inaccurate transmission phase measurements. The error in phase would obviously increase at higher frequencies.

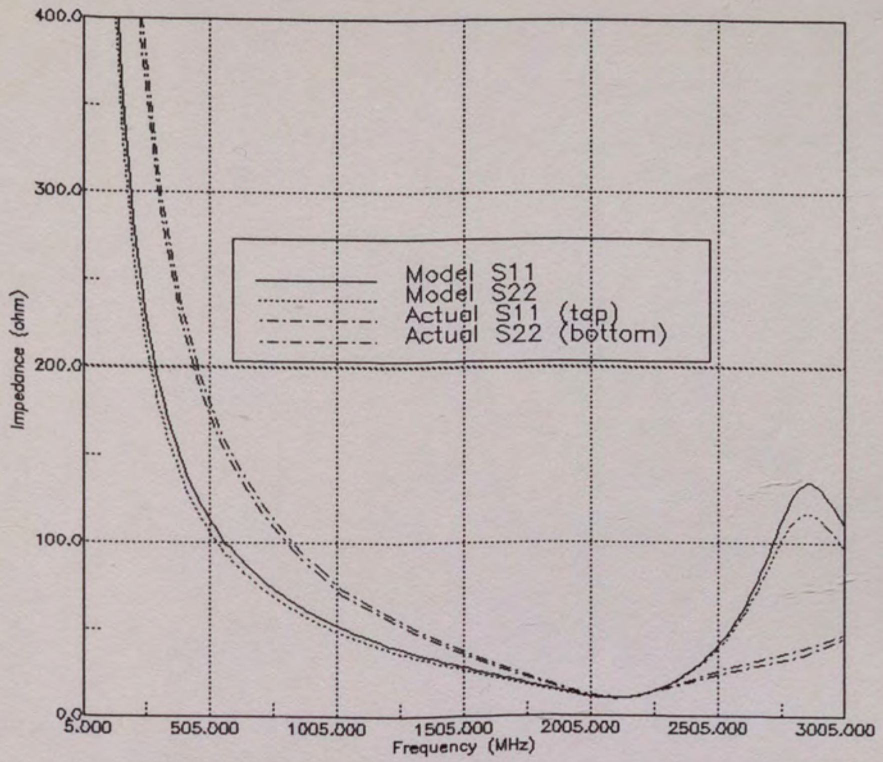

Figure 43: Wideband impedance magnitude 567 MHz dual port resonator

The response without C_{12} is plotted in a solid line style in Figure 45. The dotted line represents the same circuit except with C_{12} = 0.05 pF.

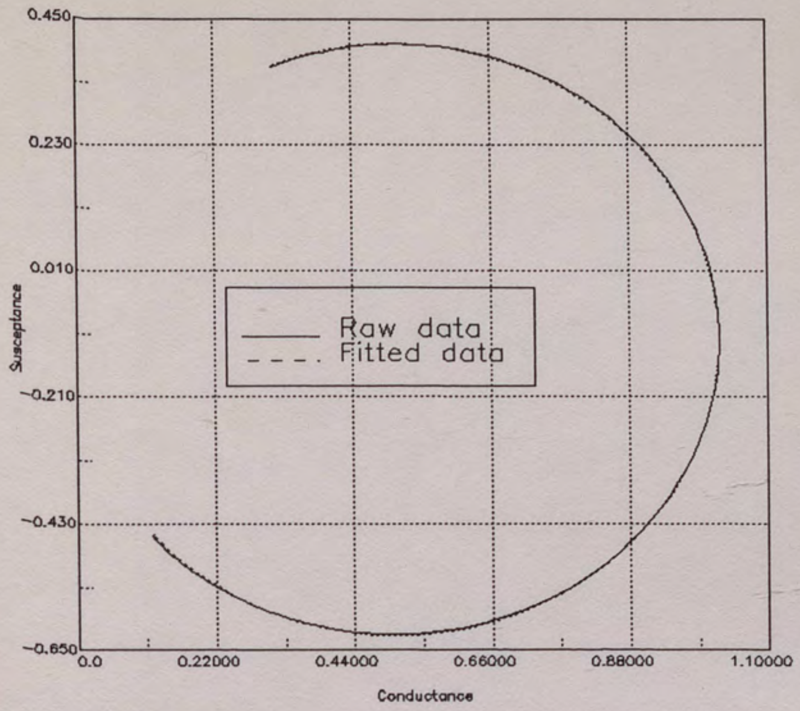

Figure 44: Susceptance versus conductance 567 MHz dual port

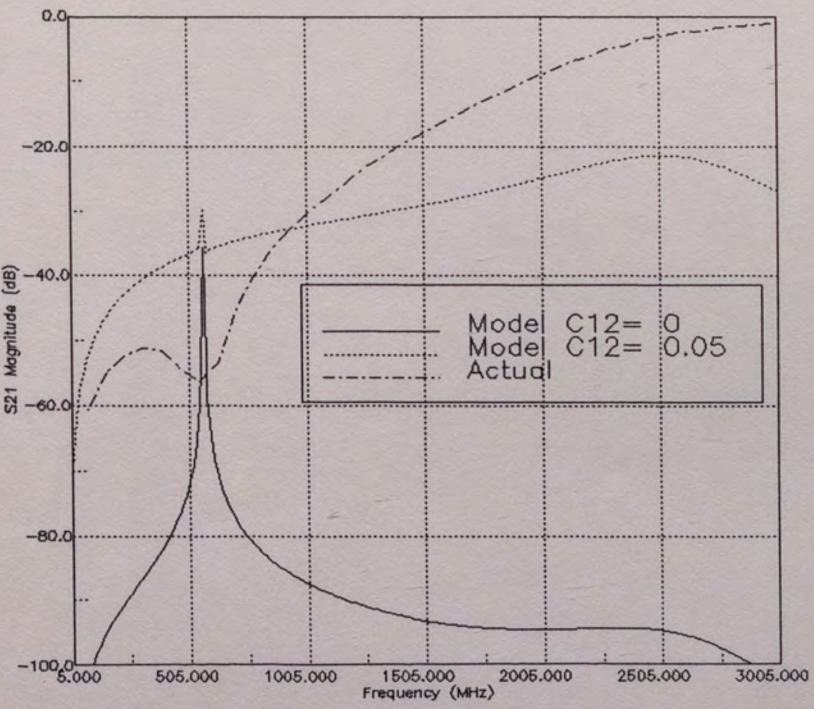

Figure 45: Wideband transmission response 567 MHz dual port resonator

CHAPTER VII

CONCLUSION

It has been demonstrated that a useful and accurate model for high frequency SAW resonators can be extracted by successive applications of the EIA-512 method. This method is based upon separating the complicated circuits into subcircuits of a form that the EIA-512 procedure can reduce. Bond wire inductance and resistance and package capacitance have been included in this fashion. Once these elements have been extracted from the high frequency, package parasitic resonance data, their effects are subtracted from the main resonance data by Z- and Y-parameter manipulations. Longitudinal modes can also be extracted in this manner.

It has been found that the transducer capacitance is usually more accurately determined from the package parasitic resonance data rather than the main resonance. Also, in the single port device this capacitance is best determined from reflection measurements instead of transmisison measurements because of the higher loaded Q available.

Six different resonators have been presented which support the validity of the theory, assumptions, and conclusions. The agreement of the more complete models is considerably better than for the basic models. The procedure is very easy to use and can be automated since no initial guesses are required. In addition, each subcircuit is extracted in a single pass. No iterative optimization routines are needed; although, if even better agreement is necessary, these routines could easily be added to the extraction procedure.

APPENDICES

APPENDIX A

EIA-512 EQUIVALENT CIRCUIT METHOD

ONE PORT THEORY

The basic model used for the one port model is illustrated in Figure 3. The goal of this theory development is to cast equations describing the response of the circuit into forms that are easily curve fitted. In addition, it is desired to have a high degree of independence between the various elements so that the curve fitting can proceed in a straightforward manner. The input admittance of this circuit is given by

$$\begin{aligned} Y &= jB_o + \frac{R_1 - jX_1}{R_1^2 + X_1^2} \qquad = jB_o + \frac{1}{R_1 + jX_1} \\ \textit{where}\quad X_1 &= j\omega L_1 - \frac{j}{\omega C_1} \qquad X_1 = \omega L_1 - \frac{1}{\omega C_1} \\ B_o &= j\omega C_o \qquad B_o = \omega C_o \end{aligned} \tag{2}$$

The admittance value, Y, is obtained by converting the measured S-parameters (see Appendix B). Since Y is known, the unknowns are B_o, R_1, and X_1. Three equations to solve for the three unknowns can be easily found by choosing three different measurement frequencies; however, because the SAW resonator is a high Q device, measurement and roundoff errors can be significant. By using a large number of frequencies, the random errors can be averaged to zero. Directly applying this to (2) results in a transcendental equation. Estimates of some of the component values can be made and an optimization routine used to extract the model. This approach does not lend itself to a generalized test procedure because of the need for estimates and the time consuming optimization. A more elegant solution is obtained with further manipulation of (2). The conductance and susceptance are found by equating real and imaginary parts resulting in

$$G = \frac{R_1}{R_1^2+X_1^2} \tag{3}$$

and

$$B = B_o - \frac{X_1}{R_1^2+X_1^2} \tag{4}$$

Rearranging, squaring, and summing equations (3) and (4) leads to equation (5).

$$G^2+(B-B_o)^2 = \frac{1}{R_1^2+X_1^2} \tag{5}$$

Note that the right hand side of (5) can be replaced by G/R_1 from (3) yielding

$$G^2+(B-B_o)^2 = \frac{G}{R_1} \qquad \textbf{(6)}$$

Now completing the square by adding $(1/2R_1)^2$ to each side results in an equation in the form of a circle.

$$(G-\frac{1}{2R_1})^2 + (B-B_o)^2 = [\frac{1}{2R_1}]^2 \qquad \textbf{(7)}$$

If the susceptance B is plotted against G, the center of the resulting circle is located at $(1/2R_1,B_o)$. The different combinations of (G,B) represent different frequencies. Since B_o is a function of frequency, the center will not be a constant, and therefore the plot will not be circular. This is where the second assumption, discussed in Chapter 4, is applied. By assuming that the shunt capacitive susceptance is constant over the extraction range, the equation can be forced to a circle with a center (G_o,B_o), where $G_o = 1/2R_1$. This leads to (8).

$$G^2 + B^2 = 2G_oG + 2B_oB - B_o^2 \qquad \textbf{(8)}$$

Error Reduction

Error reduction first requires error quantification. First let each data point be designated by G_i and B_i at the i^{th} frequency point. Next, following the notation of Malocha, et. al., the various constants in (8) are renamed for clarity:

$$\begin{aligned} M_i &= G_i^2 + B_i^2 \\ K_1 &= -B_o^2 \\ K_2 &= 2G_o \\ K_3 &= 2B_o \end{aligned} \tag{9}$$

leading to

$$M_i = K_1 + K_2G_i + K_3B_i$$

The mean squared error of the squared magnitude of the admittance is chosen for the error function:

$$\varepsilon = \sum_{i=1}^{I} (M_i-K_1-K_2G_i-K_3B_i)^2 \tag{10}$$

It is desired to choose the coefficients such that the error is minimized. This is done by setting the partial derivatives of the error with respect to each K_n equal to zero:

$$\begin{aligned} \frac{\partial\varepsilon}{\partial K_1} &= \sum_{i=1}^{I} 2\,[M_i-K_1-K_2G_i-K_3B_i] = 0 \\ \frac{\partial\varepsilon}{\partial K_2} &= \sum_{i=1}^{I} 2\,[M_i-K_1-K_2G_i-K_3B_i]\,(-G_i) = 0 \\ \frac{\partial\varepsilon}{\partial K_3} &= \sum_{i=1}^{I} 2\,[M_i-K_1-K_2G_i-K_3B_i]\,(-B_i) = 0 \end{aligned} \tag{11}$$

Simplifying and rewriting in matrix form yields

$$\begin{bmatrix} \sum K_1 & \sum K_2G_i & \sum K_3B_i \\ \sum K_1G_i & \sum K_2G_i^2 & \sum K_3G_iB_i \\ \sum K_1B_i & \sum K_2G_iB_i & \sum K_3B_i^2 \end{bmatrix} = \begin{bmatrix} \sum M_i \\ \sum M_iG_i \\ \sum M_iB_i \end{bmatrix} \tag{12}$$

This equation can be easily solved using any of several different techniques such as gaussian elimination.

Two of the equivalent circuit component values can now be determined from the Kn's.

$$\begin{gathered} R_1=\frac{1}{K_2} \quad or \quad G_o=\frac{K_2}{2} \\ B_o=\frac{K_3}{2} \quad or \quad B_o=\sqrt{-K_1} \end{gathered} \tag{13}$$

These correspond to specific points on the circle described by (8). An example of this circle is plotted in Figure 18 of Chapter 6. Hafner discusses these and other parameters and how they relate to a bulk crystal resonator. The series resistance corresponds to the inverse of the diameter? ~~radius~~. The static susceptance is the height of the center above the real axis.

Expected Values of Conductance and Susceptance

The calculation of G_o and B_o has taken advantage of the averaging effects of a least squares fit in order to minimize the errors. In order to carry this advantage further on to the calculation of the remaining parameters, the individual measured values must be replaced by their expected values. Because the measured values are actually complex pairs parameterized by frequency, it is necessary to choose a complex expected value without distorting the frequency information. This can be accomplished by moving the points along the radii of the circle. Mathmatically, this is described by (14).

$$\begin{aligned} \langle G_i \rangle &= G_o + Rad*\cos(\theta_i) \\ \langle B_i \rangle &= B_o + Rad*\sin(\theta_i) \\ where\ \theta_i &= \tan^{-1}\left[\frac{B_i - B_o}{G_i - G_o}\right] \\ Rad &= \sqrt{G_o^2 + B_o^2} \end{aligned} \quad (14)$$

<u>Determining L_1, C_1, and f_s</u>

The remaining element values may now be determined from $\langle G_i \rangle$ and $\langle B_i \rangle$ along with the frequency information that accompanies these points. In this case, the reactance X_1 is nonlinear with respect to frequency as indicated in (2). In order to apply a least squares to this system, it must first be linearized. This can be done by approximating the reactance by a polynomial, solving for its coefficients, and finally relating the polynomial to the individual element values.

By using equations (4) and (5), the reactance can be related to $\langle G_i \rangle$, $\langle B_i \rangle$, and G_o as described by (15).

$$X_{1_i} = \frac{-\langle \overline{B_i} \rangle}{\langle G_i \rangle^2 + \langle \overline{B_i} \rangle^2} \quad (15)$$

$$where\ \ \langle \overline{B_i} \rangle = \langle B_i \rangle - B_o$$

The reactance can be approximated by an n^{th} order power series. In order to simplify finding the root, the series is terminated at the third term.

$$X_1(f) = A_0 + A_1 f + A_2 f^2 \tag{16}$$

By using (16), an error function can be defined and is represented by the mean square error of the approximated reactance to the set of I measured points.

$$\varepsilon = \sum_{i=1}^{I} (\ X_i - A_0 - A_1 f_i - A_2 f_i^2\ )^2 \tag{17}$$

It is desired to find the set of A_n that gives the minimum error. As before, the partial derivative of the error function with respect to each coefficient is set to zero.

$$\begin{aligned}
\frac{\partial\varepsilon}{\partial A_0} &= \sum_{i=1}^{I} 2\left[X_{1_i} - A_0 - A_1 f_i - A_2 f_i^2\right] = 0 \\
\frac{\partial\varepsilon}{\partial A_1} &= \sum_{i=1}^{I} 2\left[X_{1_i} - A_0 - A_1 f_i - A_2 f_i^2\right](-f_i) = 0 \\
\frac{\partial\varepsilon}{\partial A_2} &= \sum_{i=1}^{I} 2\left[X_{1_i} - A_0 - A_1 f_i - A_2 f_i^2\right](-f_i^2) = 0
\end{aligned} \tag{18}$$

As before, this system of equations can be written in matrix form and solved for the coefficients A_n. The series resonant frequency, f_s, is defined as the frequency at which the series arm reactance, X_1, is zero. This frequency can therefore be found by setting (16) equal to zero, substituting the A_n's found from (18), and applying the quadratic formula.

$$f_s = \frac{-A_1 + \sqrt{A_1^2 - 4A_2A_0}}{2A_2} \tag{19}$$

Now the shunt capacitance can be determined from (20)

$$C_o = \frac{B_o}{2\pi f_s} \tag{20}$$

The resonant frequency of a series inductor and capacitor combination is given by

$$f_s = \frac{1}{2\pi\sqrt{L_1C_1}} \tag{21}$$

From (2), X_1 is given by

$$X_1 = \omega L_1 - \frac{1}{\omega C_1} = 2\pi fL_1 - \frac{1}{2\pi fC_1} \tag{22}$$

Taking the derivative of X_1 with respect to frequency and evaluating at the series resonant frequency produce the value of L_1.

$$\begin{aligned} \left.\frac{dX_1}{df}\right|_{f=fs} &= 2\pi L_1 + \left.\frac{1}{2\pi f^2 C_1}\right|_{f=fs=\frac{1}{2\pi\sqrt{L_1C_1}}} \\ &= 2\pi L_1 + \frac{4\pi^2 L_1 C_1}{2\pi C_1} \qquad (23) \\ L_1 &= \frac{1}{4\pi}\left.\frac{dX_1}{df}\right|_{f=fs} \end{aligned}$$

Finally, the series capacitance, C_1 is determined from the resonant frequency formula.

$$C_1 = \frac{1}{(2\pi f_s)^2\ L_1} \qquad (24)$$

DUAL PORT SAW RESONATORS

The dual port SAW resonator is extracted in much the same way as the single port with some additional steps to calculate the shunt capacitances and resistances. Consider the circuit of Figure 9 except replace the resistances with conductances to simplify the algebra. A Y-matrix can be written which describes this two port network.

$$Y = \begin{bmatrix} (G_{1_0}+j\omega C_{1_0})+(Y_c+j\omega C_o) & -Y_c-j\omega C_o \\ -Y_c-j\omega C_o & (G_{2_0}+j\omega C_{2_0})+(Y_c+j\omega C_o) \end{bmatrix} \qquad \textbf{(25)}$$

where: Y_c is the admittance of the series motional elements

Note that the admittance of the motional parameters and the bridging capacitive susceptance, C_o, are independent of the shunt capacitance and conductances. In fact, the motional and bridging elements can be directly extracted from $-Y_{12}$. Once Y_{12} is known, Y_{21} is know from symmetry and they can be added to Y_{11} and Y_{22}. The following equations result:

$$\begin{aligned} G_{10} &= RE\ \{\ Y_{11}+Y_{21}\ \} \\ C_{10} &= \frac{IM\ \{\ Y_{11}+Y_{21}\ \}}{\omega} \\ G_{20} &= RE\ \{\ Y_{22}+Y_{12}\ \} \\ C_{20} &= \frac{IM\ \{\ Y_{22}+Y_{21}\ \}}{\omega} \end{aligned} \qquad \textbf{(26)}$$

Since there are many sampled points, averaging can be applied to improve the accuracy.

$$
\begin{aligned}
G_{10} &= \sum \frac{RE\{Y_{11} + Y_{12}\}}{I} \\
C_{10} &= \sum \frac{IM\{Y_{11} + Y_{12}\}}{\omega I} \\
G_{20} &= \sum \frac{RE\{Y_{22} + Y_{21}\}}{I} \\
C_{20} &= \sum \frac{IM\{Y_{22} + Y_{21}\}}{\omega I}
\end{aligned}
\qquad (27)
$$

where I = number of samples

All of the elements of the dual and single port networks are determined. These equations have been developed into a Pascal named CKT. The source code is listed in Appendix C.

APPENDIX B

MATRIX CONVERSIONS

A series element at port 1 can be removed from a network by subtracting its impedance from Z_{11}. Likewise, a series impedance at port 2 can be subtracted from Z_{22} in order to remove its influence. On the other hand, a shunt element at a port is removed by subtracting its admittance from Y_{11} or Y_{22} as appropriate. An element that is across a network is removed by subtracting its admittance from each element of the admittance matrix of the network.

In order to accomplish the element removals, one must be able to convert back and forth between Y- and Z-matrices. In addition, the data points are gathered in S-parameter form, so a conversion from S- to Y-matrices is also required. These are well know equations and are reproduced here from Vendelin[2] for convenience.

$$Y_{11} = \frac{(1-S_{11})(1+S_{22})+S_{12}S_{21}}{(1+S_{11})(1+S_{22})-S_{12}S_{21}}$$

$$Y_{12} = \frac{-2S_{12}}{(1+S_{11})(1+S_{22})-S_{12}S_{21}}$$

$$Y_{21} = \frac{-2S_{21}}{(1+S_{11})(1+S_{22})-S_{12}S_{21}}$$

$$Y_{22} = \frac{(1+S_{11})(1-S_{22})+S_{12}S_{21}}{(1+S_{11})(1+S_{22})-S_{12}S_{21}}$$

$$Z_{11} = \frac{Y_{22}}{\Delta^{y}}$$

$$Z_{12} = -\frac{Y_{12}}{\Delta^{y}}$$

$$Z_{21} = -\frac{Y_{21}}{\Delta^{y}} \tag{28}$$

$$Z_{22} = \frac{Y_{11}}{\Delta^{y}}$$

$$\Delta^{y} = Y_{11}Y_{22}-Y_{12}Y_{21}$$

$$Y_{11} = \frac{Z_{22}}{\Delta^{z}}$$

$$Y_{12} = -\frac{Z_{12}}{\Delta^{z}}$$

$$Y_{21} = -\frac{Z_{21}}{\Delta^{z}}$$

$$Y_{22} = \frac{Z_{22}}{\Delta^{z}}$$

$$\Delta^{z} = Y_{11}Y_{22}-Y_{12}Y_{21}$$

APPENDIX C

SOURCE CODE FOR PROGRAM CKT

```
program ckt(input, output);




(* CKT.PASCAL reads in an S parameter file and extracts *)
(* an equivalent circuit for a SAW resonaotr using the  *)
(* EIA-512 technique.  It is closely based on D. C.     *)
(* Malocha, et al, "Quartz Resonator Model Measurement *)
(* and Sensitivity Study," in 1988 EIA Quartz Crystal  *)
(* Conference Proceedings, Vol. 2, pp. 33-39.           *)
(*                                                      *)
(*  B. Horine, SAWTEK, INC. 1989                        *)
(*                                                      *)
(*                                                      *)
(*  Uses lu.h to perform matrix solution                *)
(*       com.o for file and general utilities           *)
(*       clmath for complex, longreal math              *)

#include '/home/source/libsource/com.h';
#include '/home/source/libsource/clmath.h';

const
  MAXNUM = 1601;
  degtorad = 0.017453293;
  np = 3;
  n = 3;
  SCCSVERSION = '@(#)ckt.p    1.2';

type
  glnpbynp = array [1..np, 1..np] of longreal;    (* gl
arrays are for the lu *)
  glnp = array [1..np] of integer;      (* matrix solution
                                             routine *)
  glindx = array [1..n] of integer;
  glnarray = array [1..n] of longreal;
  vector = array [1..MAXNUM] of elem;
  matrix_array = array [1..2, 1..2] of vector;
  elemrecord = record  r, co, c, l, qu, fs: longreal; end;


var
  dbdata, degreedata, quit, exit1, one_eighty: boolean;
  num, bad, i: integer;
  omega, go, bo, rad, k1, k2, k3, gnorm, sdrm: longreal;
  go1, co1, go2, co2: longreal;
  a: glnarray;
  freq, db, ang, gexpect, bexpect, rexpect: array
                                   [1..MAXNUM] of longreal;
  y, s: matrix_array;          (* y and s parameter data *)
  fname, ans: str64;
  tmpfile: cstring;
  model: char;           (* one or two port extraction *)
  listfile: text;
  indx: glindx;
  d, stddevg, stddevb, stddevr: longreal;
  df, bw, fo, xlst, xfst: longreal;     (* for UCF file
                                             format *)
  main_elements: elemrecord;
  version: packed array[1..25] of char; (* SCCS version
                                        holding string *)

function atan2(x, y: longreal): longreal;
  external c;

#include '/home/source/libsource/comf.h';
#include '/home/source/ckt/lu.h';

procedure configuration;

(* There are several different models available.  Use the
one port model if you have only S11 data.  Use the two port
model if you also have S22, S21 (and/or S12) data.  The two
port model can extract either 0 or 180 degree resonators.
It simply adds 180 degrees to the phase of 180 degree
resonators. *)

begin
  cleartxt;
  writeln;
  write('(O)ne port or (T)wo port model:  ');
  getans(ans, ['O', 'T']);
  model := ans[1];
  if model = 'T' then begin
    writeln;
    write('(1)80 or (0) degree nominal insertion phase ? '
                                                       );
    getans(ans,['1', '0']);
    one_eighty := (ans[1] = '1');
  end;
end; { configuration }

procedure getdata;

(*  Read in data from the disk files *)

var
  f: text;
  cname, fiflag, freqdata: boolean;
  numsamp: integer;

begin
  dbdata := true;
  degreedata := false;
  num := 0;
(* prompt for file name *)
  repeat
    writeln;
    write('Data file');
    filename(fname, 'd', fiflag);
    filechck(false, cname, exit1, fiflag)
  until not cname or exit1;
  writeln;
(* read S parameter data from file *)
  if not exit1 then begin
    write('Use (S)awtek''s format or (U)CF''s format ?  ');
    getans(ans, ['S', 'U']);
    freqdata := ans[1] = 'S';
    reset(f, fname);
    if not freqdata then begin     (* read in UCF header
                                         info *)
      if not eof(f) then
     readln(f);
      if not eof(f) then
     readln(f, fo);
      if not eof(f) then
     readln(f, xfst);
      if not eof(f) then
     readln(f, xlst);
      if not eof(f) then
     readln(f, numsamp);
      df := (xlst - xfst) * 1.0e6 / (numsamp - 1);
      bw := (xlst - xfst) * 1.0e6
    end;
    while not eof(f) and (num < MAXNUM) do begin
      num := num + 1;     (* num is number of data points *)
      if freqdata then
     read(f, freq[num])
      else
     freq[num] := xfst + df * (num - 1) / 1.0e6;
      if model = 'T' then
     read(f, s[1, 1, num, 1], s[1, 1, num, 2]);
      if model = 'T' then
     read(f, s[1, 2, num, 1], s[1, 2, num, 2]);
      read(f, s[2, 1, num, 1], s[2, 1, num, 2]);
      if model = 'T' then
     readln(f, s[2, 2, num, 1], s[2, 2, num, 2])
      else
     readln(f);
      freq[num] := freq[num] * 1.0e6;
      if s[2, 1, num, 1] > 0.1 then
     dbdata := false;
      (* since the device is passive the gain cannot be
          greater than *)
      (* 0 dB.  If the data is > 0 dB then it must be in
          linear mag  *)
      (* add a little for noise *)
      if abs(s[2, 1, num, 2]) > pi + 0.5 then
     degreedata := true
    end
    (* check to see if the phase exceeds PI plus a little to
          determine *)
    (* if data is in degrees or radians  *);
    close(f);
    if freqdata then
      bw := freq[num] - freq[1]
  end;
end; { getdata }

procedure stoy;                 (* convert 1 port S parameter
                                     to Y parameter *)

var
  unity, ctwo, delta: elem;
  i, j, k : integer;

procedure correct_s11;
(* makes sure s11 and s22 are not greater than 1.0 *)
(* The problem can occur with very wide band measurements *)
(* when the calibration is not perfect *)

var i: integer;

begin
  for i := 1 to num do begin
    if dbdata and (s[1, 1, i, 1] > 0.0) then s[1, 1, i, 1]
          := 0.0;
    if dbdata and (s[2, 2, i, 1] > 0.0) then s[2, 2, i, 1]
          := 0.0;
    if (not(dbdata)) and (s[1, 1, i, 1] > 1.0) then
          s[1, 1, i, 1] := 1.0;
    if (not(dbdata)) and (s[2, 2, i, 1] > 1.0) then
          s[2, 2, i, 1] := 1.0;
  end;
end;

begin
  unity := cmplx(1.0, 0.0);
  ctwo := cmplx(-2.0, 0.0);
  for i := 1 to num do begin
    if model = 'O' then             (* one port *)
    begin
      polartorect(s[2,1,i], dbdata, degreedata);
      y[2, 1, i] := compdiv(
        compsub(unity, s[2, 1, i]),
          compadd(unity, s[2, 1, i]));
        y[2,1,i,1] := y[2,1,i,1] / 50.0;
        y[2,1,i,2] := y[2,1,i,2] / 50.0;
    end
    else begin
      if one_eighty then
        if degreedata then begin
          s[2, 1, i, 2] := s[2, 1, i, 2] + 180.0;
          s[1, 2, i, 2] := s[1, 2, i, 2] + 180.0;
        end else begin
          s[2, 1, i, 2] := s[2, 1, i, 2] + PI;
          s[1, 2, i, 2] := s[1, 2, i, 2] + PI;
        end;
      correct_s11;
      for j := 1 to 2 do
        for k := 1 to 2 do
       polartorect(s[j, k, i], dbdata, degreedata);
      delta := compsub(
        compmult(compadd(unity, s[1, 1, i]), compadd(unity,
          s[2, 2, i])),
        compmult(s[1, 2, i], s[2, 1, i]));
      delta := compmult(delta, cmplx(50.0, 0.0));

      y[1, 1, i] := compdiv(compadd(compmult(compadd(unity,
          s[2, 2, i]), compsub(unity,
          s[1, 1, i])), compmult(s[1, 2, i], s[2, 1, i])),
          delta);

      y[1, 2, i] := compdiv(compmult(ctwo, s[1, 2, i]),
          delta);
      y[2, 1, i] := compdiv(compmult(ctwo, s[2, 1, i]),
          delta);
(* average y12 & y21 to kill noise (reciprocal device) *)
      y[2, 1, i] := compdiv(compadd(y[2, 1, i], y[1, 2, i]),
          ctwo);
      y[1, 2, i] := y[2, 1, i];
      y[2, 2, i] := compdiv(compadd(compmult(compadd(unity,
          s[1, 1, i]), compsub(unity,
          s[2, 2, i])), compmult(s[1, 2, i], s[2, 1, i])),
          delta)
    end
  end;
end; { stoy }

procedure extract(ydata:vector; num:integer; var
elements:elemrecord);

procedure normalize;

var i,j : integer;

begin
  (* normalize to the midband conductance in order to *)
  (* minimize round off error and the effects of measurement
     noise *)
    gnorm := ydata[num div 2, 1];
    for j := 1 to num do
      for i := 1 to 2 do
     ydata[j, i] := ydata[j, i] / gnorm;
end;

procedure zeromatrix(var a: glnpbynp);

var
  i, j: integer;

begin
  for i := 1 to np do
    for j := 1 to np do
      a[i, j] := 0.0
end; { zeromatrix }

procedure getgobo;

(* fit addmittance data to a circle and find Go and Bo  *)
(* where the narrow band approximation has been used to *)
(* consider Bo constant.  The series resistance is      *)
(* calculated from Go.  The static capacitance is cal-  *)
(* culated from Bo once the center frequency is known.  *)

var
  b: glnarray;
  a: glnpbynp;
  m: longreal;
  i: integer;
  fil: text;

begin
  zeromatrix(a);
  b[1] := 0.0;
  b[2] := 0.0;
  b[3] := 0.0;
  a[1, 1] := num;
  rewrite(Fil,'gbraw');   (* write data to a file for
                               plotting. optional *)
  for i := 1 to num do begin
     writeln(fil,Ydata[i, 1],' ',Ydata[i, 2]);
    m := sqr(ydata[i, 1]) + sqr(ydata[i, 2]);
    a[2, 1] := a[2, 1] + ydata[i, 1];   (* coefficients *)
    a[2, 2] := a[2, 2] + ydata[i, 1] * ydata[i, 1];
    a[3, 3] := a[3, 3] + ydata[i, 2] * ydata[i, 2];
    a[3, 1] := a[3, 1] + ydata[i, 2];
    a[3, 2] := a[3, 2] + ydata[i, 1] * ydata[i, 2];
    b[1] := b[1] + m;
    b[2] := b[2] + m * ydata[i, 1];  (* solution vectors *)
    b[3] := b[3] + m * ydata[i, 2]
  end;
  close(fil);
  a[1, 2] := a[2, 1];              (* using symmetry *)
  a[1, 3] := a[3, 1];
  a[2, 3] := a[3, 2];
  ludcmp(a, 3, np, indx, d);       (* solve matrix *)
  lubksb(a, 3, np, indx, b);
  (* gaussj(a,3,np,b,1,1); *)
  elements.r := 1.0 / b[2] / gnorm;     (* denormalize and
                                        calc. R1, Go & Bo *)
  go := b[2] * gnorm / 2.0;
  bo := b[3] * gnorm / 2.0;
  k1 := b[1];
  k2 := b[2];
  k3 := b[3];
  k1 := k1 * sqr(gnorm);
  rad := sqrt(k1 + sqr(go) + sqr(bo));
end; { getgobo }

procedure get_expected_values;

var
  b1, g1, theta: longreal;
  i: integer;
  fil: text;

begin
  rewrite(fil,'gbfit');    (* write fitted G,B to file for
                                   plotting.  optional *)
  for i := 1 to num do begin
    g1 := ydata[i, 1] - go / gnorm;
    b1 := ydata[i, 2] - bo / gnorm;
    theta := atan2(b1, g1);
    gexpect[i] := (go + rad * cos(theta)) / gnorm;   (* move
                                             onto circle *)
    bexpect[i] := (bo + rad * sin(theta)) / gnorm;
    rexpect[i] := sqrt(sqr(g1) + sqr(b1));
    writeln(fil,Gexpect[i],' ',Bexpect[i]);
  end;
  close(fil);
end; { get_expected_values }

procedure find_error(var bad: integer; correct: boolean);

var
  i: integer;
  sum2errg, sum2errb, sum2errr, errorg, errorb, errorr:
longreal;

  procedure correct_data;
  var
    sdr1: longreal;

  begin
    sdr1 := abs(errorr / (rad / gnorm));
    if sdr1 > sdrm then begin
      ydata[i, 1] := gexpect[i];
      ydata[i, 2] := bexpect[i];
      bad := bad + 1;
      errorg := 0.0;
      errorb := 0.0;
      errorr := 0.0
    end
  end; { correct_data }


begin (* find_error *)   (* quantifies errors *)
  sum2errg := 0.0;
  sum2errb := 0.0;
  sum2errr := 0.0;
  for i := 1 to num do begin
    errorg := gexpect[i] - ydata[i, 1];
    errorb := bexpect[i] - ydata[i, 2];
    errorr := rad / gnorm - rexpect[i];
    if correct then
      correct_data;
    sum2errg := sum2errg + sqr(errorg / gexpect[i]);
    sum2errb := sum2errb + sqr(errorb / bexpect[i]);
    sum2errr := sum2errr + sqr(errorr / (rad / gnorm))
  end;
  stddevg := sqrt(sum2errg / num);
  stddevb := sqrt(sum2errb / num);
  stddevr := sqrt(sum2errr / num);
  sdrm := 2.7 * stddevr
end; { find_error }

procedure geta;

(* fit series reactance to a quadratic function of frequency
*)
(* after moving measured data to the fitted circle along the
*)
(* radius.  The A coefficients are later used to find the
*)
(* series resonant frequency and the series inductance
*)

var
  x1, fnorm, f, fsquare, fcube, fquad, bdiff: longreal;
  b: glnpbynp;
  i, j: integer;
  fil: text;
```

```
begin
  zeromatrix(b);
  a[1] := 0.0;
  a[2] := 0.0;
  a[3] := 0.0;
  b[1, 1] := num;
  rewrite(fil,'freqraw');  (* write data to a file for
                              plotting.  optional *)
  for i := 1 to num do begin
    bdiff := bexpect[i] - bo/gnorm;  (* subtract static
                                        susceptance *)
    x1 := -bdiff / (sqr(gexpect[i]) + sqr(bdiff));
                         (* calculate reactance *)
    writeln(fil,Freq[i]/1.0e6,' ',X1);
    f := freq[i] - freq[1];
    b[2, 1] := b[2, 1] + f;
    fsquare := sqr(f);
    fcube := f * fsquare;
    fquad := f * fcube;
    b[2, 2] := b[2, 2] + fsquare;  (* coefficients *)
    b[3, 2] := b[3, 2] + fcube;
    b[3, 3] := b[3, 3] + fquad;
    a[1] := a[1] + x1;             (* solution vector *)
    a[2] := a[2] + f * x1;
    a[3] := a[3] + fsquare * x1
  end;
  close(fil);
  b[1, 2] := b[2, 1];              (* use symmetry *)
  b[3, 1] := b[2, 2];
  b[1, 3] := b[2, 2];
  b[2, 3] := b[3, 2];
  fnorm := b[2, 2];
  for i := 1 to 3 do begin
    for j := 1 to 3 do
      b[i, j] := b[i, j] / fnorm;  (* normalize *)
    a[i] := a[i] / fnorm
  end;
  ludcmp(b, 3,np, indx, d);        (* solve matrix *)
  lubksb(b, 3,np, indx, a);
(*  gaussj(b,3,np,a,1,1); *)
  rewrite(fil,'freqfit');     (* write fitted data to file.
                                        optional *)
  for i := 1 to num do begin
    f := freq[i] - freq[1];
    x1 := a[1] + a[2] * f + a[3] * f * f;
    writeln(fil,Freq[i]/1.0e6,' ',X1);
  end;
  close(fil);
end; { geta }

function getfs: longreal;
```

```
(* use quadratic equation to find the series resonant
frequency *)

begin
  getfs := freq[1] + (-a[2] + sqrt(sqr(a[2]) - 4 * a[3] *
a[1])) / (2 * a[3])
end; { getfs }

function getl: longreal;

(* use the derivative of the series expansion of X1 to find
*)
(* the series inductance *)

begin
  with elements do
    getl := (a[2] + 2 * a[3] * (fs - freq[1])) / (4 * PI *
          gnorm)
end; { getl }

begin { extract }
  normalize;
  getgobo;
  bad := 0;
  get_expected_values;
  find_error(bad, false);
  find_error(bad, true);
  if bad > 0 then begin
    getgobo;
    get_expected_values;
    find_error(bad, false)
  end;
  geta;
  with elements do begin
    fs := getfs;
    l := getl;
    omega := 2 * PI * fs;
    co := bo / omega;
    c := 1.0 / (sqr(omega) * l);
  end;
end; { extract }

procedure twoport;  (* solve for C10, C20, G10, G20 *)

var
  i, j, k: integer;
  w: longreal;
  sum_real, sum_imag: array [1..2] of longreal;

begin
  sum_real[1] := 0.0; sum_real[2] := 0.0;
  sum_imag[1] := 0.0; sum_imag[2] := 0.0;
  for i := 1 to num do begin
    w := 2 * PI * freq[i];
    for j := 1 to 2 do begin
      k := 2 - j div 2;
      sum_real[j] := sum_real[j] + y[j, j, i, 1] -
                    y[j, k, i, 1];
      sum_imag[j] := sum_imag[j] + (y[j, j, i, 2] -
                    y[j, k, i, 2]) / w
    end;
  end;
  go1 := sum_real[1] / num;
  co1 := sum_imag[1] / num;
  go2 := sum_real[2] / num;
  co2 := sum_imag[2] / num
end; { twoport }


procedure printresults(var ptr: text);

var
  day, tim: packed array [1..10] of char;

begin
  writeln(ptr);
  date(day);
  time(tim);
  writeln(ptr, day, ' ', tim);
  writeln(ptr);
  writeln(ptr,version);
  writeln(ptr);
  writeln(ptr, ' File:  ', fname);
  writeln(ptr);
  with main_elements do begin
    if one_eighty then begin
      writeln(ptr,'180 degree phase shift');
      writeln(ptr);
    end;
    writeln(ptr, ' Fs = ', fs * 1.0e-6: 12: 6, ' MHz');
    writeln(ptr, ' R1 = ', r: 4: 1, ' ohms');
    if l >= 1.0e-3 then
      if l < 10.0e-3 then
        writeln(ptr, ' L1 = ', l * 1.0e3: 9: 6, ' mH')
      else
        writeln(ptr, ' L1 = ', l * 1.0e3: 9: 6, ' mH')
    else if l >= 1.0e-6 then
      writeln(ptr, ' L1 = ', l * 1.0e6: 9: 6, ' uH')
    else
      writeln(ptr, ' L1 = ', l * 1.0e9: 9: 6, ' nH');
    if c < 1.0e-15 then
      writeln(ptr, ' C1 = ', c * 1.0e18: 9: 6, ' aF (E-18)')
    else
      writeln(ptr, ' C1 = ', c * 1.0e15: 9: 6,
                    ' fF (E-15)');
    writeln(ptr, ' Qu = ', omega * l / r: 6:0);
    if model = 'O' then
      writeln(ptr, ' Co = ', co * 1.0e12: 4: 3, ' pF')
    else begin
      writeln(ptr, ' Co1 = ', co1 * 1.0e12: 4: 3, ' pF');
      writeln(ptr, ' Go1 = ', go1 * 1.0e3: 6: 3, ' mS');
      writeln(ptr, ' Co2 = ', co2 * 1.0e12: 4: 3, ' pF');
      writeln(ptr, ' Go2 = ', go2 * 1.0e3: 6: 3, ' mS');
      writeln(ptr, ' C12 = ', co * 1.0e12: 4: 3, ' pF')
    end;
  end; { with main_elements }
  writeln(ptr);
  writeln(ptr, '   std dev G = ', stddevg);
  writeln(ptr, '   std dev B = ', stddevb);
  writeln(ptr, '   std dev R = ', stddevr);

  writeln(ptr, '   # bad points = ', bad: 4);
  writeln(ptr, '   Total number of points = ', num: 4);
end; { printresults }

begin (* main *)
  quit := false;
  version := SCCSVERSION;
  for i := 1 to 21 do version[i] := version[i+4];
  for i := 22 to 25 do version[i] := ' ';
  repeat
    configuration;
    getdata;
    if not exit1 then begin
      stoy;
      extract(y[2, 1], num, main_elements);
      if ( model = 'T' ) then twoport;
      printresults(output);
      writeln;
      write('print results (Y/N) ?  ');
      getans(ans, ['Y', 'N']);
      if ans[1] = 'Y' then begin
     tmpfile := gettempfile('ckt', listfile);
     printresults(listfile);
     printtempfile(listfile, tmpfile)
      end;
      writeln;
      write('Do another (Y/N) ?  ');
      getans(ans, ['Y', 'N']);
      if ans[1] = 'N' then
     quit := true
    end else
      quit := true
   until quit
 end. { ckt }
```

APPENDIX D

SOURCE CODE FOR PROGRAM SUBPAR1 (SUBTRACTION OF ELEMENTS FOR SINGLE PORT DEVICES)

```
program subpar(input, output);

#include '/home/source/libsource/com.h';
#include '/home/source/libsource/clmath.h';

const
  MAXNUM = 1601;
  degtorad = 0.017453293;
  SCCSVERSION = '@(#)subpar1.p     1.1';

type
  vector = array [1..MAXNUM] of elem;

var
  dbdata, degreedata, exit1, forcedb, fiflag, cname:
boolean;
  num, i: integer;
  freq: array [1..MAXNUM] of longreal;
  y, s: vector;
  fname, ans: str64;
  f: text;
  df, bw, fo, xlst, xfst, lb, rb, cp, ct: longreal;


#include '/home/source/libsource/comf.h';

procedure getdata;

var
  f: text;
  cname, fiflag, freqdata: boolean;
  numsamp: integer;

begin
  dbdata := true;
  degreedata := false;
  num := 0;
(* prompt for file name *)
  repeat
    writeln;
    write('Data file');
    filename(fname, 'd', fiflag);
    filechck(false, cname, exit1, fiflag)
  until not cname or exit1;
  writeln;
(* read S parameter data from file *)
  if not exit1 then begin
    write('Use (S)awtek''s format or (U)CF''s format ?  ');
    getans(ans, ['S', 'U']);
    freqdata := ans[1] = 'S';
    reset(f, fname);
    if not freqdata then begin
```

```
     if not eof(f) then
    readln(f);
     if not eof(f) then
    readln(f, fo);
     if not eof(f) then
    readln(f, xfst);
     if not eof(f) then
    readln(f, xlst);
     if not eof(f) then
    readln(f, numsamp);
     df := (xlst - xfst) * 1.0e6 / (numsamp - 1);
     bw := (xlst - xfst) * 1.0e6
   end;
   while not eof(f) and (num < MAXNUM) do begin
     num := num + 1;      (* num is number of data points *)
     if freqdata then
    read(f, freq[num])
     else
    freq[num] := xfst + df * (num - 1) / 1.0e6;
     readln(f, s[num, 1], s[num, 2]);
     freq[num] := freq[num] * 1.0e6;
     if s[num, 1] > 0.01 then
    dbdata := false;
      (* since the device is passive the gain cannot be
          greater than *)
      (* 0 dB.  If the data is > 0 dB then it must be in
          linear mag  *)
      (* add a little for noise *)
      if abs(s[num, 2]) > pi + 0.5 then
     degreedata := true
    end;
    (* check to see if the phase exceeds PI plus a little to
          determine *)
    (* if data is in degrees or radians  *)
    close(f);
    if freqdata then
      bw := freq[num] - freq[1]
  end;
  if forcedb then
    dbdata := true
end; { getdata }

procedure stoy;                          (* convert 1 port S
parameter to Y parameter *)

var
  unity: elem;
  i: integer;

  procedure correct_s11;

  var
```

```
    i: integer;

  begin
    for i := 1 to num do begin
      if dbdata and (s[i, 1] > 0.0) then
     s[i, 1] := 0.0;
      if dbdata and (s[i, 1] > 0.0) then
     s[i, 1] := 0.0;
      if not dbdata and (s[i, 1] > 1.0) then
     s[i, 1] := 1.0;
      if not dbdata and (s[i, 1] > 1.0) then
     s[i, 1] := 1.0
    end
  end; { correct_s11 }


begin
  unity := cmplx(1.0, 0.0);
  for i := 1 to num do begin
    correct_s11;
    polartorect(s[i], dbdata, degreedata);
    y[i] := compdiv(
        compsub(unity, s[i]), compadd(unity, s[i]));
    y[i, 1] := y[i, 1] / 50.0;
    y[i, 2] := y[i, 2] / 50.0;
  end
end; { stoy }

procedure ytos;

var
  unity: elem;
  i: integer;

begin
  unity := cmplx(1.0, 0.0);
  for i := 1 to num do begin
    y[i, 1] := y[i, 1] * 50.0;
    y[i, 2] := y[i, 2] * 50.0;
    s[i] := compdiv(
        compsub(unity, y[i]), compadd(unity, y[i]));
  end
end; { ytos }

function ytoz(y:elem):elem;

begin
  ytoz := compdiv(cmplx(1.0, 0), y);
end;

procedure getrl(var rb, lb: longreal);
```

```
begin
  writeln;
  write(’Enter the bond wire inductance (nH) :  ’);
  readln(lb);
  lb := lb * 1.0e-9;
  writeln;
  write(’Enter the bond wire resistance (ohm) :  ’);
  readln(rb);
  writeln;
  write(’Enter the package capacitance (pf) :  ’);
  readln(cp);
  cp := cp * 1.0e-12;
  writeln;
  write(’Enter the transducer capacitance (pf) :  ’);
  readln(ct);
  ct := ct * 1.0e-12;
end; { getrl }

procedure subparasitics;

var
  i: integer;
  zb, z: elem;
  bp, bt, omega: longreal;

begin
  for i := 1 to num do begin
    omega := 2 * PI * freq[i];
    zb := cmplx(rb, omega * lb);
    bp := omega * cp;
    bt := omega * ct;
    y[i, 2] := y[i, 2] - bp;
    z := ytoz(y[i]);
    z := compsub(z, zb);
    y[i] := ytoz(z); (* ytoz is the same as ztoy *)
    y[i, 2] := y[i, 2] - bt;
  end
end; { subparasitics }

begin {main}
  writeln(SCCSVERSION);
  getdata;
  if not exit1 then begin
    stoy;
    getrl(rb, lb);
    subparasitics;
    ytos;
    dbdata := true;
    repeat
      writeln;
      write(’Output file’);
      filename(fname, ’d’, fiflag);
      filechck(true, cname, exit1, fiflag)
    until not cname or exit1;
    rewrite(f, fname);
    for i := 1 to num do begin
      recttopolar(s[i], dbdata, true);
      writeln(f,freq[i]/1.0e6: 11: 6, s[i, 1]: 9: 3,
          s[i, 2]: 9: 3);
    end;
    close(f)
  end
end. { subpar }
```

APPENDIX E

SOURCE CODE FOR PROGRAM SUBPAR (SUBTRACTION OF ELEMENTS FOR DUAL PORT DEVICES)

```
program subpar(input, output);

#include '/home/source/libsource/com.h';
#include '/home/source/libsource/clmath.h';

const
  MAXNUM = 1601;
  degtorad = 0.017453293;
  SCCSVERSION = '@(#)subpar.p 1.2';

type
  vector = array [1..MAXNUM] of elem;
  matrix = array [1..2, 1..2] of elem;
  matrix_array = array [1..MAXNUM] of matrix;


var
  dbdata, degreedata, exit1, forcedb, fiflag, cname:
boolean;
  num, i: integer;
  freq: array [1..MAXNUM] of longreal;
  y, s: matrix_array;
  fname, ans: str64;
  f: text;
  df, bw, fo, xlst, xfst, lb1, lb2, rb1, rb2, cp1, ct1, cp2,
ct2, c12: longreal;


#include '/home/source/libsource/comf.h';

procedure getdata;

var
  f: text;
  cname, fiflag, freqdata: boolean;
  numsamp: integer;

begin
  dbdata := true;
  degreedata := false;
  num := 0;
(* prompt for file name *)
  repeat
    writeln;
    write('Data file');
    filename(fname, 'd', fiflag);
    filechck(false, cname, exit1, fiflag)
  until not cname or exit1;
  writeln;
(* read S parameter data from file *)
  if not exit1 then begin
    write('Use (S)awtek''s format or (U)CF''s format ?  ');
  getans(ans, ['S', 'U']);
  freqdata := ans[1] = 'S';
  reset(f, fname);
  if not freqdata then begin
    if not eof(f) then
   readln(f);
    if not eof(f) then
   readln(f, fo);
    if not eof(f) then
   readln(f, xfst);
    if not eof(f) then
   readln(f, xlst);
    if not eof(f) then
   readln(f, numsamp);
    df := (xlst - xfst) * 1.0e6 / (numsamp - 1);
    bw := (xlst - xfst) * 1.0e6
   end;
   while not eof(f) and (num < MAXNUM) do begin
     num := num + 1;    (* num is number of data points *)
     if freqdata then
    read(f, freq[num])
     else
    freq[num] := xfst + df * (num - 1) / 1.0e6;
     read(f, s[num, 1, 1,  1], s[num, 1, 1,  2]);
     read(f, s[num, 1, 2,  1], s[num, 1, 2,  2]);
     read(f, s[num, 2, 1,  1], s[num, 2, 1,  2]);
     readln(f, s[num, 2, 2,  1], s[num, 2, 2,  2]);
     freq[num] := freq[num] * 1.0e6;
     if s[num, 2, 1,  1] > 0.01 then
    dbdata := false;
     (* since the device is passive the gain cannot be
         greater than *)
     (* 0 dB.  If the data is > 0 dB then it must be in
         linear mag  *)
     (* add a little for noise *)
     if abs(s[num, 2, 1,  2]) > pi + 0.5 then
    degreedata := true
   end;
   (* check to see if the phase exceeds PI plus a little to
         determine *)
   (* if data is in degrees or radians  *)
   close(f);
   if freqdata then
     bw := freq[num] - freq[1]
  end;
  if forcedb then
    dbdata := true
end; { getdata }

procedure stoy;                    (* convert 1 port S
parameter to Y parameter *)
```

```
var
  unity, ctwo, delta: elem;
  i, j, k : integer;

  procedure correct_s11;

  var
    i: integer;

  begin
    for i := 1 to num do begin
      if dbdata and (s[i, 1, 1, 1] > 0.0) then
     s[i, 1, 1, 1] := 0.0;
      if dbdata and (s[i, 2, 2, 1] > 0.0) then
     s[i, 2, 2, 1] := 0.0;
      if not dbdata and (s[i, 1, 1, 1] > 1.0) then
     s[i, 1, 1, 1] := 1.0;
      if not dbdata and (s[i, 2, 2, 1] > 1.0) then
     s[i, 2, 2, 1] := 1.0
    end
  end; { correct_s11 }


 begin
   unity := cmplx(1.0, 0.0);
   ctwo := cmplx(-2.0, 0.0);
   for i := 1 to num do begin
     correct_s11;
     for j := 1 to 2 do
       for k := 1 to 2 do
      polartorect(s[i, j, k], dbdata, degreedata);
     delta := compsub(compmult(compadd(unity, s[i, 1, 1]),
       compadd(unity, s[i, 2, 2])), compmult(s[i, 1, 2],
           s[i, 2, 1]));
     delta := compmult(delta, cmplx(50.0, 0.0));

     y[i, 1, 1] := compdiv(compadd(compmult(compadd(unity,
           s[i, 2, 2]),
     compsub(unity, s[i, 1, 1])), compmult(s[i, 1, 2],
           s[i, 2, 1])), delta);

     y[i, 1, 2] := compdiv(compmult(ctwo, s[i, 1, 2]),
           delta);

     y[i, 2, 1] := compdiv(compmult(ctwo, s[i, 2, 1]),
           delta);

     y[i, 2, 2] := compdiv(compadd(compmult(compadd(unity,
           s[i, 1, 1]),
     compsub(unity, s[i, 2, 2])), compmult(s[i, 1, 2],
           s[i, 2, 1])), delta);
   end
```

```
end; { stoy }

procedure ytos;

var
  unity, ctwo, delta: elem;
  i, j, k : integer;

begin
  unity := cmplx(1.0, 0.0);
  ctwo := cmplx(-2.0, 0.0);
  for i := 1 to num do begin
    for j := 1 to 2 do begin
      for k := 1 to 2 do begin
        y[i, j, k, 1] := y[i, j, k, 1] * 50.0;
        y[i, j, k, 2] := y[i, j, k, 2] * 50.0;
      end;
    end;

    delta := compsub(compmult(compadd(unity, y[i, 1, 1]),
      compadd(unity, y[i, 2, 2])), compmult(y[i, 1, 2],
          y[i, 2, 1]));

    s[i, 1, 1] :=
          compdiv(compadd(compmult(compsub(unity,y[i,1,1]),
          compadd(unity,y[i,2,2])),
          compmult(y[i,1,2],y[i,2,1])),delta);

    s[i, 1, 2] := compdiv(compmult(ctwo, y[i, 1, 2]),
          delta);

    s[i, 2, 1] := compdiv(compmult(ctwo, y[i, 2, 1]),
          delta);

    s[i, 2, 2] := compdiv(compadd(compmult(
          compadd(unity,y[i,1,1]),
          compsub(unity,y[i,2,2])),
          compmult(y[i,1,2],y[i,2,1])),delta);

  end
end; { ytos }

function ytoz(y:matrix):matrix;

var
  delta, minus: elem;
  z: matrix;

begin
  minus := cmplx(-1.0,0);
  delta :=
compsub(compmult(y[1,1],y[2,2]),compmult(y[1,2],y[2,1]));
```

```
  z[1,1] := compdiv(y[2,2],delta);
  z[1,2] := compdiv(compmult(minus,y[1,2]),delta);
  z[2,1] := compdiv(compmult(minus,y[2,1]),delta);
  z[2,2] := compdiv(y[1,1],delta);
  ytoz := z;
end;

function ztoy(z:matrix):matrix;

var
  delta, minus: elem;
  y: matrix;

begin
  minus := cmplx(-1.0,0);
  delta := compsub(compmult(z[1,1],z[2,2]),compmult(z[1,2],
          z[2,1]));
  y[1,1] := compdiv(z[2,2],delta);
  y[1,2] := compdiv(compmult(minus,z[1,2]),delta);
  y[2,1] := compdiv(compmult(minus,z[2,1]),delta);
  y[2,2] := compdiv(z[1,1],delta);
  ztoy := y;
end;

procedure getrl;

begin
  writeln;
  write('Enter the bond wire inductance (nH) (PORT 1) :  ');
  readln(lb1);
  lb1 := lb1 * 1.0e-9;
  writeln;
  write('Enter the bond wire inductance (nH) (PORT 2) :  ');
  readln(lb2);
  lb2 := lb2 * 1.0e-9;
  writeln;
  write('Enter the bond wire resistance (ohm) (PORT 1) :  '
        );
  readln(rb1);
  writeln;
  write('Enter the bond wire resistance (ohm) (PORT 2) :  '
        );
  readln(rb2);
  writeln;
  write('Enter the package capacitance (pf) (PORT 1) :  ');
  readln(cp1);
  cp1 := cp1 * 1.0e-12;
  writeln;
  write('Enter the transducer capacitance (pf) (PORT 1) :  '
        );
  readln(ct1);
  ct1 := ct1 * 1.0e-12;
  writeln;
  write(’Enter the package capacitance (pf) (PORT 2) :  ’);
  readln(cp2);
  cp2 := cp2 * 1.0e-12;
  writeln;
  write(’Enter the transducer capacitance (pf) (PORT 2) :  ’
          );
  readln(ct2);
  ct2 := ct2 * 1.0e-12;
  writeln;
  writeln(’Enter the bridging capacitance (pf) (C12) :  ’);
  readln(c12);
  c12 := c12 * 1.0e-12;
end; { getrl }

procedure subparasitics;

var
  i: integer;
  zb1, zb2: elem;
  bp1, bt1, bp2, bt2, b12, omega: longreal;
  z: matrix;

begin
  for i := 1 to num do begin
    omega := 2 * PI * freq[i];
    zb1 := cmplx(rb1, omega * lb1);
    zb2 := cmplx(rb2, omega * lb2);
    bp1 := omega * cp1;
    bp2 := omega * cp2;
    bt1 := omega * ct1;
    bt2 := omega * ct2;
    b12 := omega * c12;
    y[i, 1, 1, 2] := y[i, 1, 1, 2] - bp1;
    y[i, 2, 2, 2] := y[i, 2, 2, 2] - bp2;
    z := ytoz(y[i]);
    z[1, 1] := compsub(z[1, 1], zb1);
    z[2, 2] := compsub(z[2, 2], zb2);
    y[i] := ztoy(z);
    y[i, 1, 1, 2] := y[i, 1, 1, 2] - bt1;
    y[i, 2, 2, 2] := y[i, 2, 2, 2] - bt2;
    { subtract out the bridging capacitance c12 }
    y[i, 1, 1, 2] := y[i, 1, 1, 2] - b12;
    y[i, 1, 2, 2] := y[i, 1, 2, 2] + b12;
    y[i, 2, 1, 2] := y[i, 2, 1, 2] + b12;
    y[i, 2, 2, 2] := y[i, 2, 2, 2] - b12;
   end
 end; { subparasitics }

 begin {main}
   writeln(SCCSVERSION);
   getdata;
  if not exit1 then begin
    stoy;
    getrl;
    subparasitics;
    ytos;
    dbdata := true;
    repeat
      writeln;
      write('Output file');
      filename(fname, 'd', fiflag);
      filechck(true, cname, exit1, fiflag)
    until not cname or exit1;
    rewrite(f, fname);
    for i := 1 to num do begin
      recttopolar(s[i, 1, 1], dbdata, true);
      recttopolar(s[i, 1, 2], dbdata, true);
      recttopolar(s[i, 2, 1], dbdata, true);
      recttopolar(s[i, 2, 2], dbdata, true);
      writeln(f,freq[i]/1.0e6: 11: 6, s[i, 1, 1, 1]: 9: 3,
          s[i, 1, 1, 2]: 9: 3, s[i, 1, 2, 1]: 9: 3,
          s[i, 1, 2, 2]: 9: 3, s[i, 2, 1, 1]: 9: 3,
          s[i, 2, 1, 2]: 9: 3, s[i, 2, 2, 1]: 9: 3,
          s[i, 2, 2, 2]: 9: 3);
    end;
    close(f)
  end
end. { subpar }
```

REFERENCES

[1] EIA. 1985. "Standard Methods for Measurement of the Equivalent electrical Parameters of Quartz Crystal Units, 1 kHz to 1 GHz." ANSI/EIA-512-1985.

[2] Vendelin, G. D. 1982. Design of Amplifiers and Oscillators by the S-Parameter Method. John Wiley & Sons: New York.

[3] For example: Compact Software, Inc: Paterson, NJ. Eagleware/Circuit Busters, Inc: Stone Mountain, GA. EESOF, Inc: Westlake Village, CA.

[4] Shreve, W. R. 1975. "Surface-Wave Two-Port Resonator Equivalent Circuit." Proc. 1975 Ultrasonics Symposium : 295-298.

[5] Schmidt, R. V. and L. A. Coldren. 1975. "Thin Film Acoustic Surface Waveguides on Anisotripic Media." IEEE Transactions on Sonics and Ultrasonics SU-22 (2). (March): 115-122.

[6] Hafner, E. 1969. "The Piezoelectric Crystal Unit - Definitions and Methods of Measurement." Proc. of the IEEE, 57(2) (February) : 179-201.

[7] Schoenwald, J. S., R. C. Rosenfeld, and E. J. Staples. 1974. "Surface Wave Cavity and Resonator Characteristics-VHF to L-Band." Proc. 1974 IEEE Ultrasonics symposium : 253-256.

[8] Shreve, W. R. 1975. "Surface-Wave Two-Port Resonator Equivalent Circuit." Proc. 1975 IEEE Ultrsonics Symposium : 295-298.

[9] Gunes, D. 1978. "A Measurement System for VHF S.A.W. Resonator Equivalent Parameters and Its Use in Aging." Master's Thesis, Northwestern University, Evanston, IL.

[10] Horine, B. H. 1990. "Narrow Band SAW Filters for IF Applications." Proc. of the 44th Annual Symposium on Frequency Control : 316-322.

[11] Semones, T. 1986. "A Software Implementation of EIA-512." 1986 EIA Quartz Crystal Conference Proceedings.

[12] Electronic Industries Association, "EIA-512" pg 1.

[13] Malocha, D. C., H. Ng, and M. Fletcher. 1988. "Quartz Resonator Model Measurement and Sensitivity Study." 1988 EIA Quartz Crystal Conference Proceedings. Vol. 2: 31-39.

[14] Hewlett-Packard: Palo Alto, CA