Toolbox for SAW Filter Analysis in the Quasi-Static Approximation

For Use in MatLab and
Stand-Alone Applications

Author:

Alexander S. Rukhlenko, Dr.

User's Guide
Version 1.2

Contacting the Author

(*Pricing, sales, consulting, and technical support*)

Author: Alexander S. Rukhlenko, Dr.

Phone: +49 89 976 00 973

Mobile: +49 173 654 7820

Address: Gustav-Heinemann-Ring, 61
81739 Munich
Germany

Web: www.intrasaw.com

E-mail: alexander.rukhlenko@gmail.com
rukhlenko@intrasaw.com

Toolbox for SAW Filter Analysis in the Quasi-Static Approximation

User's Guide

Printing History: May 2000, January 2001, May 2001

CONTENTS

INTRODUCTION

About the Author

Dr. Alexander S. Rukhlenko received his degree in Radiophysics and Electronics (with distinction) from the Belarusian State University, Minsk in 1978. He received the Ph. D. degree from the Minsk Radio Engineering Institute in 1989. From 1980 to 1993, he worked in the laboratory of Acousto- and Optoelectronics at the Minsk Radio Engineering Institute, where he was responsible for research and development of SAW devices and their applications to signal processing. In 1993, he joined Semiconductor Physics Department at the Belarusian State University to continue his SAW research. Currently, he is a Principal Researcher at the Belarusian State University where he is a head of the SAW devices group.

Dr. A.S.Rukhlenko is one of the recognized world specialists in the field of SAW devices computer-aided design. His research work is known and appreciated abroad. Since 1987, he has contributed a lot to the theory and practice of the computer-aided design of SAW filters based on the IBM PC compatible platform. His state-of-the-art computer-aided design of SAW filters was demonstrated to and highly appreciated by the world-recognized SAW experts. Some parts of his software were purchased and used by the industrial SAW companies.

In 1995, he became a winner of the SAW filter design competition organized among SAW research groups and specialists of the former USSR by the American company SAWTEK, Inc. which is one of the global leaders in SAW design and production. In 1995-1996, he worked with SAWTEK as a SAW consultant for 18 months including 6 months stay in the USA. He stayed for 3 months at the Technical University of Munich, Department of Microwave Engineering as a visiting scientist in 1995. He worked as an invited SAW designer for LG Company, Seoul, South Korea in 1997-1998 where he has been engaged in designing SAW IF filters for mobile communications (CDMA, PCS, etc.). He visited with lectures on the computer-aided design of SAW filters number of the American companies (SAWTEK, Toko of America, Andersen Labs.). In 1999-2000, he took part as an independent contractor and consultant in several cooperative projects with foreign SAW companies.

His present scientific interests comprise computer-aided design of SAW devices, problems of analysis, synthesis, and simulation of SAW filters, design techniques of low-loss IF and RF filters, etc. He has more than 40 publications on acoustoelectronics. Some principal results were presented at the IEEE Frequency Control Symposium (1992, 1993), IEEE Ultrasonics Symposium (1994, 1995), World Congress on Ultrasonics (1995), and in the international periodicals.

What is the SAW Filter Analysis Toolbox

The SAW Filter Analysis Toolbox (SAWFAT) is a collection of the software tools for comprehensive analysis of the in-line or dual-track SAW filters in the quasi-static approximation. An accurate MSC modeling can be included in the analysis of the dual-track SAW filters if necessary (optional).

Basic modeling assumption is that bidirectional SAW interdigital transducers (IDT) are supposed to be *periodic* and *non-reflective* (SAWFAT 1.0, 1.1). However, the advanced analysis tools of SAWFAT 1.2 may optionally include analysis of bidirectional SAW filters comprising two reflective periodic SAW transducers.

Two-mode approach (expansion into symmetric and antisymmetric first order rectangular modes) is applied to MSC modeling that provides a sufficient accuracy in the most practical applications.

SAW filters to be analyzed consist of two SAW transducers cascaded in the following combinations:

1) two unapodized (regular or withdrawal-weighted) SAW transducers (in-line);
2) one unapodized and another apodized SAW transducers (in-line);
3) two identical or different apodized SAW transducers coupled through a multistrip coupler (dual-track).

The toolbox provides the following capabilities:

- calculation of the mixed scattering matrix (*P*-matrix) of the unapodized or apodized periodic SAW transducers comprising the acoustoelectric conversion function *m* and transducer admittance $Y(\omega)=G(\omega)+jB(\omega)+j\omega C$ that includes the radiation conductance $G(\omega)$, radiation susceptance $B(\omega)$, and the static capacitance C. The reflection coefficent of a short-circuit transducer is identically equal to zero in the quasi-static approximation. However, the transducer reflection and transduction properties can be accurately modeled in the full version of SAWFAT 1.2. Other pertinent elements of the complete *P*-matrix are found by reciprocity and power conservation. For any arbitrary metallization ratio, the baseband and harmonic frequency characteristics can be calculated;

- calculation of the scattering matrix of a regular multistrip coupler based on the two-mode approach. Four different MSC models are used (coupling-of-modes, reflective array model, closed-form field approach, and quasi-static approximation). MSC scattering matrix is calculated by expansion of the propagating waves in two orthogonal rectangular modes (symmetric and antisymmetric), each of the modes characterized by the open-/short-circuit strip reflection coefficient and SAW phase velocity in the periodic grating;

- analysis of SAW filters comprising two bidirectional periodic SAW transducers coupled acoustically either directly (in-line) or through the multistrip coupler (dual-track). A SAW filter is modeled as the reciprocal lossless two-port network that can be described either in terms of the admittance matrix (**Y**-matrix) or scattering matrix (**S**-matrix). Software to calculate **Y**- and **S**-matrices of a SAW filter based on
  1) the closed-form **Y**-matrix calculation in the quasi-static approximation;
  2) direct cascading of the augmented scattering matrices for reflective SAW transducers

  is included in the toolbox. It is worthy to note that **Y**-matrix is convenient for SAW filter analysis including the external circuitry as the standard electrical network analysis techniques may be applied. On the other hand, the calculated **S**-matrix allows comparing the simulation results with experimental *S*-parameters measured on a standard network analyzer.

There are no constraints on the polarity sequence or apodization shape imposed. The transducers can be uniform (unapodized), regular or polarity-weighted (withdrawal-weighted), or aperture-weighted (apodized) having a constant period (pitch) and metallization ratio (duty factor) throughout the structure. To provide good simulation accuracy in the quasi-static approximation, a transducer central frequency f_0 must be far away from the synchronous frequency f_π. This limitation is removed in SAWFAT 1.2 where tools for analysis of the reflective SAW transducers can be optionally supplied. No further restrictions on the synchronous-to-central frequency ratio f_π/ f_0 are imposed. Therefore, SAW transducers may contain any integer number of fingers $M>2$ per wavelength in the quasi-static approximation $M\geq2$ if analysis of the reflective SAW transducers is included in the full version SAWFAT 1.2 where $f_\pi =M/2\, f_0$. This normally includes a regular split-finger design $M=4$ ($f_\pi =2f_0$) as well as other known periodic structures including M=3 (f_π =1.5f_0) or M=6 (f_π =3f_0) fingers per wavelength.

State-of-the-art computational algorithms of maximum performance and flexibility are implemented in the toolbox based mostly on the closed-form SAW transducer theory developed by the author in the quasi-static approximation [1-4] and generalized later to the case of the reflective SAW transducers (unpublished). A universal computational algorithm based on the weighted summation of the elemental admittances and capacitors is applied to the admittance calculation of the unapodized and apodized SAW transducers, with the unapodized SAW transducers treated as a particular case (uniform aperture-weighting) of the apodized transducers.

The package contains excellent modeling tools, which are useful both for the experienced SAW specialists as well as for the beginners in this field. No previous SAW filter experience is required as the complete set of the toolbox documentation contains the comprehensive theoretical part and the detailed algorithms and software description. Tutorial modeling examples are included which illustrates the basic software features.

Most of the computational modules are implemented as the MATLAB MEX-files created from C and Fortran computational subroutines interfaced with MATLAB. C and Fortran source codes of the computational subroutines as well as of the gateway programs are included to the full software version that makes it possible to use the software in the stand-alone applications written in C or Fortran. The tools for generating MEX-files by the user are also included in the toolbox. This makes the toolbox flexible and extendible as it can be tailored (modified or added) to the user's needs if necessary. The software can be easily integrated in the computer-aided design of SAW filters.

New Features

New features were implemented in MATLAB Toolbox SAWFAT, version 1.2. If compared to the previous version 1.0, the software package includes both M-script files and GUI version of the SAW filter analysis program (in the quasi-static approximation). In addition to the quasi-static approximation, analysis of the reflective periodic SAW transducers can be included (optional). The GUI implementation gives additional power and flexibility to SAWFAT. The program is very useful both for experts and novices in the field. The features of the new version are:

1) friendly graphical user interface (GUI) with convenient control panel and pop-up menus
2) convenient presentation format of the analysis results (*S*- and *Y*-parameters) in the form of the tiled subplots (compact form) or full scale figures
3) analysis of SAW filters with different input/output acoustic apertures
4) analysis of SAW filters with different input/output transducer periodicity
5) analysis of SAW filters for arbitrary source/load resistance
6) analysis of SAW filters comprising the reflective SAW transdcucers that takes into account interelectrode reflections as well as multi-path reflections between SAW transducers (optional)
7) acoustoelectric conversion functions of the input/output SAW transducers can be calculated and plotted
8) Cartesian and Smith chart presentation of *S*-parameters
9) time response calculation using Fast Fourier Transform (FFT)
10) tap weights of the input and output SAW transducers can be read from the separate text files
11) analysis results (*S*- or *Y*-parameters) can be exported onto disc in the ASCII text format
12) external *S*- or *Y*-parameters can be imported from a formatted text file to be visualized in the program

13) input data can be conveniently modified and saved while analyzing a SAW filter
14) help file with useful information can be accessed at any time
15) color palette to draw the plots can be adjusted by the user.

How This Book Is Organized

Chapter 1 discusses compiling, linking and building MEX-files, which enables to call C and/or Fortran computational subroutines directly from MATLAB. This chapter also provides the necessary information to get up and run so that one can configure his system to build MEX-functions from the supplied source codes of the gateway programs and computational subroutines.

Chapter 2 contains the detailed description of the software (directory structure and organization) including purpose, synopsis, argument description, and use of each subroutine. The M-script files, MEX-files, and C/Fortran computational subroutines are separately discussed. The GUI version of the software (SAWFAT, version 1.2) is separately considered.

Chapter 3 discusses tutorial examples and test results. The sample data files are given that allows the user to effectively adopt these examples to the user's needs or compose the own data files for analysis of the customized SAW filters. Instructions for running software are given, with the test results presented in the convenient graphical form.

1. TOOLBOX ORGANIZATION AND USING MEX-FILES

1.1. Introducing MEX-Files

Although MATLAB is a complete, self-contained environment for programming and manipulating data, it is often useful to interact with the programs external to the MATLAB environment. MATLAB provides an Application Program Interface (API) to support these external interfaces to call C and/or Fortran programs from MATLAB. So, the user can call his own C or Fortran subroutines from MATLAB as if they were built-in MATLAB functions. MATLAB callable C and Fortran programs are referred to as MEX-files. Therefore, MEX-files are dynamically linked subroutines (DLLs) created from C or Fortran source codes that the MATLAB interpreter can automatically load and execute.

MEX-files have following applications:

- Pre-existing C and Fortran programs can be called from MATLAB without having to be rewritten as M-files.
- Bottleneck computations (usually for-loops) that do not run fast enough in MATLAB can be recoded in C or Fortran for efficiency.

Such customized C or Fortran MEX-files may considerably reduce computational time if compared to the MATLAB M-script files or MEX-files generated by the built-in MATLAB compiler. These powerful MATLAB options are effectively exploited in the present toolbox where both C and Fortran source codes of all computational subroutines are presented. This also gives an option of using these subroutines in stand-alone applications when necessary.

On Windows, MATLAB 5.x MEX-files have an extension *.DLL*. One can call MEX-files exactly as if they were M-functions. For example, a MEX-file called *Admit.DLL* performs calculation of the admittance of a bidirectional SAW transducer, in the quasi-static approximation. If one invokes the function *Admit* from inside MATLAB, the interpreter searches for this function through the list of directories on the MATLAB's search path. It scans each directory in the path looking for the first occurrence of a file named *Admit* with the corresponding filename extension *.DLL* or *.m*. When it finds one, it loads the file and executes it. Please note that MEX-files take precedence over M-files when like-named files exist in the same directory. One should keep this in mind as there are few toolbox functions implemented both as MEX-files (with an extension *.DLL*) and M-script files (with an extension *.m*), particularly: *Fleg* (Legendre function calculation), *Elemfact* (element-factor calculation) and others.

1.2. Parts of the MEX-File

The source code for a MEX-file consists of two separate parts:

• a computational routine that contains the codes for performing the computations that you want implemented in the MEX-file. As a rule, computations are numerical ones. Both C and Fortran versions of the computational subroutines are presented in the package, with file extensions *.c* and *.f*, respectively.

• a gateway routine that interfaces the computational routine with MATLAB by the entry point *mexFunction* with parameters *prhs, nrhs, plhs, nlhs,*

where *prhs* is an array of MATLAB right-hand input arguments;
nrhs is the number of right-hand input arguments;
plhs is an array of left-hand MATLAB output arguments, and
nlhs is the number of left-hand output arguments.
The gateway routine calls the computational subroutine(s) written in C or Fortran. In the gateway routine, you can access the data in the *mxArray* structure and then pass and manipulate this data in your C or FORTRAN computational subroutine.

The gateway routines can be written in C or Fortran. In this package, all the gateway routines are implemented both in C and Fortran. They call either C or Fortran computational subroutines with the corresponding names. The names of the gateway routines correspond to the names of the generated MEX-files followed by the underscore character (for example, *Admit_.c* or *Admit_.f* is the gateway routine calling the computational subroutine *Admit.c* or *Admit.f* and generating the MEX-file *Admit.DLL*.

Each MEX-file of the package is supplied by a separate like-named MATLAB Help-file with the extension *.m*. Help-file is invoked by the MATLAB command

help *function_name*

where *function_name* is the name of the pertinent MEX-function.

Example: **help** *admit*

ADMIT SAW transducer admittance calculation

[yidt,wp]=ADMIT (ne,y,wmcm,km,ep,k2,fpi,fmhz,[npt])
Calculates the dynamic admittance of the unapodized or
apodized SAW transducer including radiation conductance
and susceptance

Input arguments:

ne - number of electrodes
y - transversal gap positions (array)
wmcm - acoustic aperture, mcm
km - metallization ratio
ep - effective substrate permittivity
k2 - coupling factor, %
fpi - synchronous frequency, MHz
fmhz - frequency, MHz (scalar or array)
npt - number of frequency points (optional)

Output arguments:

yidt - transducer admittance, mho (scalar or array)
wp - effective apertures

1.3. Directory Organization and File Name Conventions

The SAWFAT file names are subjected to the naming conventions as listed in Table 1.1.

The SAWFAT directory structure containing comprehensive software for modeling SAW interdigital transducers (IDT), multistrip couplers (MSC), and in-line or dual-line SAW filters is listed in the alphabetical order in Table 1.2.:

The comprehensive catalog of SAWFAT subdirectories and files is given in Tables 1.2-1.9.

Table 1.1

File Naming Conventions

File name	Extension	To indicate	Example
Low/upper case letters	*.m*	MATLAB M-file	*Elemfact.m*
	.m	MATLAB Help file	*Admit.m*
	.c	C computational subroutine	*Admit.c*
	.f	Fortran computational subroutine	*Admit.f*
	.DLL	MATLAB MEX-file	*Admit.DLL*
	.ini	Initialization file	*SAWFAT.ini*
	.col	Color palette file (color settings)	*SAWFAT.col*
	.hlp	Help file (for GUI SAWFAT)	*SAWFAT.hlp*
Followed by underscore	*.c*	C gateway program	*Admit_.c*
Upper case *TEST_X* or *TEST_XX* or *TEST_XXX*	*.m*	MATLAB test M-file	*TEST_C.m* *TEST_EF.m* *TEST_QSF.m*
Filter_1…6	*.dat*	Data files for SAW filter analysis in the quasi-static approximation	*Filter_1.dat*
MSC_1…4	*.dat*	Data files for MSC analysis	*MSC_1.dat*
Filter_1…6	*.top*	Data files containing topological data of SAW transducers	*Filter_1.top*
Filter_1…6	*.t1*	Text file containing y-coordinates of the finger transversal gaps for the input transducer	*Filter_1.t1*
Filter_1…6	*.t2*	Text file containing y-coordinates of the finger transversal gaps for the output transducer	*Filter_1.t2*
Filter_1…6	*.s*	Text files containing the modeled results (S-parameters) of SAW filters in the tutorial examples	*Filter_1.s*

Table 1.2

SAWFAT Root Directory

Subdirectories	
Directory	Purpose
C	C source codes for generating MEX-files (optional)
DOC	Documentation containing the theory (models, algorithms) and software manual
Examples	Tutorial examples and sample data files
FORTRAN	Fortran source codes for generating MEX-files (optional)
GUI x_x	SAWFAT GUI version for analysis of bidirectional SAW filters in the quasi-static approximation
IDT	Software for *P*-matrix calculation (M-files)
MEX	User-generated MEX-files
MEX.5_3	supplied MEX-files for MATLAB 5.3
MEX.6_1	supplied MEX-files for MATLAB 6.1
MSC	M-files for analysis of the dual-track SAW filters including MSC modeling (optional)
Miscellaneous Files	
File	Purpose
ACAD.ini	AutoCAD initialization file in the DXF-format (read-only)
MSC.ini	Initialization file for analysis of the multistrip couplers (MSC)
MSF.ini	Initialization file for analysis of MSC dual-track SAW filters
QSF.ini	Initialization file for analysis of SAW filters in the quasi-static approximation
SAWFAT.ini	SAWFAT initialization file (to be written automatically in the current directory)
SAWFAT.col	SAWFAT color palette initialization file
SAWFAT.hlp	SAWFAT help file
SAWMEX.m	Program to generate MEX-files

Table 1.3

Subdirectories C and Fortran (Source Codes)

…/GATE Gateway Routines (optional)	
File	Purpose
Admit_.c, Admit_.f	Admittance calculation
Apeff_.c, Apeff_.f	Calculation of the effective apertures for static capacitance
Apeffd_.c, Apeffd_.f	Calculation of the effective apertures for dynamic admittance
Capac_.c, Capac_.f	Static capacitance calculation for a given set of effective apertures
Ekpk_.c, Ekpk_.f	Calculation of the capacitive and potential coefficients
Elemfac_.c, Elemfac_.f	Element function calculation
Fleg_.c, Fleg_.f	Legendre function calculation
…/COMPUTE Computational Subroutines (optional)	
Admit.c, Admit.f	Admittance calculation
Apeff.c, Apeff.f	Calculation of the effective apertures for static capacitance
Apeffd.c, Apeffd.f	Calculation of the effective apertures for dynamic admittance
Capac.c, Capac.f	Static capacitance calculation for a given set of effective apertures
Ekpk.c, Ekpk.f	Calculation of the capacitive and potential coefficients
Elemfact.c, Elemfact.f	Element function calculation
Fleg.c, Fleg.f	Legendre function calculation
Maxmin.c, Maxmin.f	Searching for the minimum/maximum values in an array

Table 1.4

Subdirectory DOC

…/THEORY (optional)	
File	Purpose
Theory_0.doc	Contents, introduction
Theory_1…7.doc	Chapters 1-7.
…/MANUAL (User's Guide)	
Manual_0.doc	Contents, introduction
Manual_1…4.doc	Chapters 1-4.

Table 1.5

Subdirectory **IDT**

MATLAB M-Files (Functions)	
*Admit_g.m**	Admittance calculation for reflective SAW transducers
Analyze.m	Analysis of bidirectional SAW filter in the quasi-static approximation
Capacity.m	Static capacitance calculation of the interdigital SAW transducer
dB.m	Log magnitude calculation in decibels
Elemfact.m	Element factor calculation
Fleg.m	Legendre function calculation
Filt_QS.m	SAW filter analysis in the quasi-static approximation
*Gap.m**	Array factor calculation in terms of the finger overlaps (gaps) for reflective SAW transducers
*IDT_cf.m**	Closed-form calculation of the mixed scattering matrix (*P*-matrix) for reflective SAW transducers
*IDT_IDT.m**	Cascading *P*-matrices in SAW filter with reflective SAW transducers
Idealfrd.m	Ideal frequency response in terms of the gap (overlap) taps
IDT_QS.m	Mixed scattering matrix (*P*-matrix) calculation of the interdigital SAW tranducer
Read_f.m	Reading data file in the format SAWFAT 1.1
Read_f2.m	Reading data file in the format SAWFAT 1.2
MtoT.m	Conversion of the IDT mixed scattering matrix to the transmission matrix
MtoY.m	Conversion of the IDT mixed scattering matrix to the admittance matrix
StoT.m	Conversion of the wave scattering matrix to the transmission matrix
Y_to_S.m	Conversion of the admittance matrix to the wave scattering matrix
YtoS2.m	Conversion of the admittance two-port matrix Y(2,2) to the closed-form wave scattering matrix S(2,2)
Y2toS2.m	Conversion of the vectorized two port Y-matrix with elements Y11, Y12, Y21, Y22 to the vectorized S-matrix with elements S11, S12, S21, S22 for arbitrary input/output characteristic admittance
*Wavenum.m**	Solution of the dispersion equation
MATLAB M-Files (Scripts)	
TEST_C.M	Test program for static capacitance calculation
TEST_EF.M	Test program for element factor calculation
TEST_QSF.M	Test program for analysis of SAW filters in the quasi-static approximation (new version)
CONVERT.M	Data file converter from the format of SAWFAT 1.0- 1.1 to the new format of SAWFAT 1.2
PLOTDATA.M	Program to plot *S*-parameters from a formatted text file
FULLPLOT.m	Program to plot the analysis results (*P*-matrix elements and *S*-parameters) from the MATLAB workspace in the separate large-scale figures
SUB_PLOT.m	Program to plot the analysis results (*P*-matrix elements and *S*-parameters) from the MATLAB workspace in the compact form (subplots in one figure)

MATLAB M-Files (Help)	
Admit.m	MATLAB Help file for MEX-file *Admit.DLL*
Apeff.m	MATLAB Help file for MEX-file *Apeff.DLL*
Apeffd.m	MATLAB Help file for MEX-file *Apeffd.DLL*
Capac.m	MATLAB Help file for MEX-file *Capac.DLL*
Ekpk.m	MATLAB Help file for MEX-file *Ekpk.DLL*

Table 1.6

Subdirectories **MEX, MEX.5_3, MEX.6_1**

MATLAB MEX-Files (DLLs)	
Admit.DLL	Admittance calculation
Apeff.DLL	Calculation of the effective apertures for static capacitance
Apeffd.DLL	Calculation of the effective apertures for dynamic admittance
Capac.DLL	Static capacitance calculation for a given set of effective apertures
Ekpk.DLL	Calculation of the capacitive and potential coefficients
Elemfact.DLL	Element function calculation
Fleg.DLL	Legendre function calculation

Table 1.7

Subdirectory **GUI x.x**

MATLAB M-Files (Functions)	
File	Purpose
Draw_IDT.m	Drawing of SAW transducer topology
Drawtaps.m	Schematical drawing of the transducer tap weignts
Eval1.m, *Eval2.m*, *Eval3.m*	Functions containing the *eval* constructions removed from the M-functions to make them compilable by the built-in MATLAB compiler
Export.m	Export of the *S*- or *Y*-parameters onto disk in the ASCII text file
Figure_.m	Plot of the analysis results in the large scale format (figures)
Fun_help.m	Utility function to display help text in the convenient format
Inp_dlg.m	Input dialog box (analog of the INPUTDLG MATLAB standard function with a bug fixed)
Rd_help.m	Function to read help from a text help file
Read_ini.m	Reading a SAWFAT initialization file from disk

Read_top.m	Reading SAW filter topological data from a separate text file
SAWFAT.m	Main recursive subroutine to implement various computational and GUI functions in SAW filter analysis program
Smith_ch.m	Smith chart representation of the *S*-parameters
Subplot_.m	Plot of the analysis results in the compact (tiled) format (subplots)
Top_DXF.m	Converting SAW filter topology to the AutoCAD DXF-format
Wr_ini.m	Backup SAWFAT initialization data onto disk
Wr_mask.m	Writing SAW filter topology (photomask pattern) in the text file
Write_f2.m	Backup SAW filter data in the text file (format SAWFAT 1.2)
Writef.m	Formatted column-wise data output in the text file (general purpose)

Note: * - optional, for analysis of reflective SAW transducers (supplied separately).

Table 1.8

Subdirectory **MSC** (optional)

MATLAB M-Files (Functions)	
Disp_COM.m	Solution of the dispersion equation using COM analysis
Disp_FLD.m	Solution of the dispersion equation using field approach
gr_COM.m	Short/open circuit grating modeling by applying COM-analysis
gr_FLD.m	Short/open circuit grating modeling using field approach
gr_RAM.m	Short/open circuit grating modeling using reflective array model (RAM)
gr_QS.m	Short/open circuit grating modeling in the quasi-static approximation
MSC.m	Calculation of the MSC scattering matrix using different modeling techniques
MSCN.m	Evaluation of the optimum number of the MSC strips
v_SAW.m	Calculation of the SAW velocity in the short/open-circuit grating
r_SAW.m	Calculation of the reflection coefficient in the short/open-circuit grating
Read_MSC.m	Reading MSC data from the text file
T_MSC_T.m	Calculation of the **Y**-matrix of a SAW filter comprising two identical apodized SAW transducers coupled through a MSC
MATLAB M-Files (Scripts)	
TEST_MSC.m	Test program for MSC wave scattering matrix calculation
TEST_MSF.m	Test program for modeling MSC SAW filter
FULL_MSC.m	Program to plot MSC analysis results (*S*-parameters) from the MATLAB workspace in the separate large-scale figures
SUB_MSC.m	Program to plot MSC analysis results (*S*-parameters) from the MATLAB workspace in the compact form (subplots in one figure)

Table 1.9

Subdirectory **EXAMPLES**

…/FILTER_1…6 Data Files for SAW Filter Modeling	
Filter_1.dat	Data file for SAW filter analysis comprising input unapodized and output apodized (centro-symmetric) SAW transducers on the 128 *YX* lithium niobate substrate (Example # 1)
Filter_2.dat	Data file for SAW filter analysis comprising input unapodized and output apodized (V-shape biased) SAW transducers on the 128 *YX* lithium niobate substrate (Example # 2)
Filter_3.dat	Data file for SAW filter analysis comprising input polarity-weighted and output apodized SAW transducers on the 128 *YX* lithium niobate substrate (Example # 3)
Filter_4.dat	Data file for SAW filter analysis comprising input polarity-weighted and output apodized SAW transducers on the 112 *YX* lithium tantalate substrate (Example # 4)
Filter_5.dat	Data file for SAW filter analysis comprising polarity-weighted input and output SAW transducers on the ST-quartz substrate (Example # 5)
Filter_6.dat	Data file for dual-track SAW filter analysis comprising two apodized SAW transducers and MSC on the 128 *YX* lithium niobate substrate (Example # 6)
…/MSC Data Files for MSC Simulation	
MSC_1.dat	Data file for MSC analysis on the 128 *YX* lithium niobate (MSC Example # 1)
MSC_2.dat	Data file for MSC analysis on the *YZ* lithium niobate (N=120, η=0.375) for comparison with the experimental results (MSC Example # 2)
MSC_3.dat	Data file for MSC analysis on the *YZ* lithium niobate (N=100, η=0.45) for comparison with the experimental results (MSC Example # 3)
MSC_4.dat	Data file for MSC analysis on the *YZ* lithium niobate (N=120, η=0.45) for comparison with the experimental results (MSC Example # 4)
Topological Data	
Filter_1.top	Sample of the topological data file (Example # 1)
Tap Weights	
Filter_1…6.t1	Text files containing y-coordinates for the transversal gaps of the input SAW transducer
Filter_1…6.t2	Text files containing y-coordinates for the transversal gaps of the output SAW transducer

Modeled Results	
Filter_1…6.s	Text files containing modeled *S*-parameters of SAW filters *Filter_1…6.dat*

1.4. Building MEX-Files

MATLAB 5.x or 6.x contains all the tools you need to work with the Application Program Interface (API), except C and/or Fortran compilers (note that MATLAB 6.x contains built-in C/C++ compiler). To generate toolbox MEX-files, WATCOM C/C++ Compiler, Version 10 (or later), and/or Microsoft Visual C, Version 5.0 (or later), and/or DIGITAL Visual Fortran (Compaq Visual Fortran), Version 5.0 (or later) must be properly pre-installed on your computer. The external compilers must be able to create 32-bit windows dynamically linked libraries (DLLs). If you are not going to use the Fortran computational subroutines, you may install C compiler(s) only or use the built-in compiler. You may have to do some preliminary work before you can create MEX-files.

Stages of building MEX-files

There are three basic stages to MEX-file building for both C and Fortran compilers:1) compiling, 2) prelinking, and 3) linking.

1) The compile stage must:
   - Set up paths to the compiler and linker files using the **PATH**, **INCLUDE**, **LIB**, and environment variables in the system file *AUTOEXEC.bat*.
   - Define the **WATCOM** environment variable in order for the WATCOM Linker to locate WATCOM Fortran 32-bit library files when the application is linked.
   - Define the compiler switches in the corresponding environment variables (say, **WCL386**) or in the command line, particularly:
     a) the switch to create a DLL is required for MEX-files.
     b) the –c switch (compile only; do not link) is recommended.
     c) the switch to specify 8-byte alignment.
     d) set up optimizer switches and/or debug switches if needed;
     e) other switches specific to the environment if necessary.

2) The prelink stage dynamically creates import libraries for importing the required function(s) into the MEX-file. All MEX-files link against MATLAB only. MATLAB and each DLL have corresponding *.def* files of the same names located in the **<MATLAB>\EXTERN\INCLUDE** directory. The path should be set up in the **INCLUDE** environment variable.

3) Finally, the link stage is to define the WATCOM Linker directives which must contain
   a) the directive **SYSTEM** to create a DLL for MEX-files;
   b) the directive **NAME** to provide a name of the MEX-file, if necessary;
   c) the directive **EXPORT** for exporting the entry point to the MEX-file as *mexFunction* for WATCOM C and/or Fortran;
   d) the directive **LIBRARY** to specify the library file(s) including import library (s) to be searched when unresolved symbols remain after processing object files, if needed;
   e) the directive **FILE** to specify the object files and library modules that the WATCOM Linker is to process;
   f) other linker directive(s) specific to the environment that can be used.

MATLAB include an option (mex – setup) that generates the file *mexopts.bat* to automatically implement these three stages. This procedure simplifies greatly the setup for particular compiler.

Step-by-Step MEX-Files Generating

To compile, link, generate, and run MEX-files on your computer, you should do the following simple steps.

1. Backup the contents of the supplied toolbox.

2. Include the toolbox subdirectories into MATLABPATH variable in the startup.*m* file using the external text editor or the MATLAB built-in path editor EDITPATH.

List of the subdirectories to be included in the path:

D:\SAWFAT;		
*D:\SAWFAT\FORTRAN\GATE;	*or*	*D:\SAWFAT\C\GATE;
*D:\SAWFAT\FORTRAN\COMPUTE;	*or*	*D:\SAWFAT\C\COMPUTE;
D:\SAWFAT\MEX;		
D:\SAWFAT\GUIx_x;		
D:\SAWFAT\IDT;		
D:\SAWFAT\MSC;		
D:\SAWFAT\EXAMPLES\FILTER_1;		
D:\SAWFAT\EXAMPLES\FILTER_2;		
D:\SAWFAT\EXAMPLES\FILTER_3;		
D:\SAWFAT\EXAMPLES\FILTER_4;		
D:\SAWFAT\EXAMPLES\FILTER_5;		
D:\SAWFAT\EXAMPLES\FILTER_6;		
D:\SAWFAT\EXAMPLES\FILTER_7;		
D:\SAWFAT\EXAMPLES\MSC;		

where **SAWFAT** is the name of the root directory of the SAW Filter Analysis Toolbox.

For user convenience, an example of the startup MATLAB file *startup.m* with the full SAWFAT path is included in the package.

NOTE:

1) The subdirectory *MEX* containing MEX-files must be set on the path before the subdirectory *IDT* that contains like-name MATLAB script files with help.
2) The subdirectories marked with the asterisk are necessary to generate MEX-files from Fortran or C source codes. They can be excluded from the MATLAB path afterwards.

3. Configure MEX using MATLAB **mex** command with the *setup* switch at the MATLAB prompt to generate the options file for the selected pre-installed C or Fortran compiler

mex –setup

The options file controls which compiler to use, the compiler and link command options, and the runtime libraries to link against while building MEX-files (see MATLAB Compiler User’s Guide for details).

4. Make the IDT directory be a current working directory using **CD** (change current working directory) command at the MATLAB prompt.

5. Run the M-script file *SAWMEX.m* with the switch *-c* or *-f* to generate C or Fortran language MEX-files, respectively. The switch *–f* can be omitted as specified by default.

Example:

SAWMEX	build Fortran MEX-files
SAWMEX –f	build Fortran MEX-files
SAWMEX –c	build C MEX-files

6. The normal MEX-files generating process must be followed by the messages:

C/C++ Compiler (*or* FORTRAN Compiler)

1. ADMIT [size] Ok
2. APEFF [size] Ok
3. APEFFD [size] Ok
4. CAPAC [size] Ok
5. EKPK [size] Ok
6. ELEMFACT [size] Ok
7. FLEG [size] Ok

The size of the generated MEX-file depends on the used compiler.

7. The MEX-files are located in the directory **MEX** to be included in the MATLAB path afterwards.

8. Run the test programs SAWFAT, TEST_QSF.*m*, TEST_EF.*m*, TEST_C.*m*, TEST_MSC.*m*, TEST_MSF.*m*.

9. Compare the results of the test examples with those given in this Manual (Chapter 3).

NOTES:

1. MEX-files are generated automatically and located by default in the subdirectory **../SAWFAT/MEX**.

2. The **MEX** subdirectory must be put in the MATLAB path before **IDT** subdirectory (to avoid conflicting between like-name DLL and help M-files (*admit*, *apeff, capac,* etc.).

3. Supplied MEX-files can be copied directly from the subdirectories **MEX.5_3** or **MEX.6_1** to MEX subdirectory if the user does not have any C or Fortran compiler at his disposal.

1.4.3. Troubleshooting

1.4.3.1. Compiler Limitations

1) The toolbox MEX-files were generated and tested using the following compilers:

WATCOM C/C++ Compiler, Version 10.5 (or later)
Microsoft Visual C, Version 5.0 (or later)
DIGITAL Visual Fortran, Version 5.0 (or later).

2) All the he compilers must be able to create 32-bit windows dynamically linked libraries (DLLs).
3) If several different compilers have been pre-installed they must not conflict to each other.
4) The Fortran compiler must support the %val construct.

1.4.3.2. Regenarating MEX-files

Once the MEX-file has been already invoked in MATLAB by running the program (say, TEST_QSF), any attempt of regenerating the DLL fails followed by the WATCOM LINKER error message:

I/O error processing {file name.DLL}: access denied.

Any of the following MATLAB commands can be run to break (deactivate) the active link to the MEX-file(s):

CLEAR FUN clears the function specified by the name FUN.
CLEAR MEX removes all links to MEX-files.
CLEAR ALL removes all variables, globals, functions, and MEX links.

After all the toolbox MEX-files have been generated, you can call them as if they were the conventional M-files.

2. TOOLBOX SOFTWARE

2.1. SAW Filter Analysis in the Quasi-Static Approximation (IDT-Directory)

2.1.1. MATLAB MEX-Files

Admit.dll

Purpose SAW transducer admittance calculation.

Synopsis [*yidt*,*wp*]=ADMIT (*ne*,*y*,*wmcm*,*km*,*ep*,*k2*,*fpi*,*fmhz*)
[*yidt*,*wp*]=ADMIT (*ne*,*y*,*wmcm*,*km*,*ep*,*k2*,*fpi*,*fmhz*,*npt*)

Description Calculates the dynamic admittance of the unapodized or apodized SAW transducer including radiation conductance and susceptance.

Algorithm The computational algorithm is based on the concept of the nodal admittance matrix of a periodic SAW transducer and comprises three major steps [1, Chapter 3], [2, 3]:

1) calculation of a set of the effective apertures in terms of the finger overlaps;
2) calculation of the nodal admittance matrix of the periodic SAW transducer;
3) weighted summation of the elemental addmittances, with the weights given by the effective apertures.

Arguments
ne – number of electrodes
y – transversal gap positions (array)
wmcm – acoustic aperture, m
km – metallization ratio
ep – substrate effective permittivity
k2 – coupling factor, %
fpi – synchronous frequency, MHz
fmhz – frequency, MHz (scalar or vector)
npt – number of frequency points (optional)

Returns
yidt – transducer admittance, $^{-1}$
wp – effective apertures (array)

See also *Apeffd*.

Apeff.dll

Purpose Calculation of the partial effective apertures for capacitance calculation.

Synopsis *wp*=APEFF (*ne*,*y*)

Description Calculates a set of the effective apertures in terms of the finger taps.

Algorithm Partial effective apertures in the static capacitance calculation are defined as the total overlaps of all the neighbor fingers, next neighbor ones, etc. [1, Chapter 4], [2, 4].

Arguments *ne* – number of electrodes *y* – *y*-coordinates of the transversal gaps (array)

Returns *wp* – finger effective apertures (array)

Apeffd.dll

Purpose Calculation of the partial effective apertures for dynamic admittance calculation.

Synopsis *wp*=APEFFD (*ne*,*y*)

Description Calculates a set of the effective apertures in terms of the finger overlaps (gap taps).

Algorithm Partial effective apertures in the dynamic admittance calculation are defined as the total overlaps of all the neighbor gaps, next neighbor ones, etc. [1, Chapter 3], [2, 3].

Arguments *ne* – number of electrodes *y* – *y*-coordinates of the transversal gaps (array)

Returns *wp* – gap effective apertures (array)

Capac.dll

Purpose Static capacitance calculation.

Synopsis [*c*,*cpf*]=CAPAC (*ne*,*wp*,*wmcm*,*nc*,*cp*,*ep*)

Description Given a set of the partial apertures *wp* and capacitive coefficients *cp*, calculates the static capacitance of an unapodized/apodized SAW transducer.

Algorithm The computational algorithm is based on the concept of the interelectrode capacitors of a periodic SAW transducer and comprises three major steps [1, Chapter 4], [2, 4]:

1) calculation of a set of the partial effective apertures in terms of the finger taps;
2) calculation of the capacitive coefficients of the periodic SAW transducer;
3) weighted summation of the interelectrode capacitors, with the weights given by the effective apertures.

This function implements the third step, supposed for the effective apertures and capacitive coefficients found at the earlier steps.

Arguments
ne – number of electrodes
wp – set of the partial apertures (array)
wmcm – acoustic aperture, m
nc – number of the retained capacitive coefficients
c*p* – capacitive coefficients (array)
ep – substrate effective permittivity
y – *y*-coordinates of the transversals gaps

Returns *c* – normalized capacitance (in terms of the capacitance of one finger pair in the solid-finger periodic transducer) *cpf* – static capacitance, *p*F

See also *Apeff, Ekpk.*

Ekpk.dll

Purpose Calculation of the capacitive (or potential) coefficients.

Synopsis [*g*,*gs*]=EKPK (*ne*,*km*,*eps*)

Description Calculates capacitive (or potential) coefficients of a periodic SAW transducer for the arbitrary metallization ratio.

Algorithm The capacitive (or potential) coefficients are found for one generalized period of the periodically replicated SAW transducer as the superposition of the phased array elemental transducers [1, Chapter 4], [2, 4].

Arguments *ne* – number of electrodes
km – metallization ratio (duty factor) *eps* – relative error for calculation of the coefficients

Returns *gs* – capacitive or potential coefficients of the elemental phased transducers *g* – capacitive or potentials coefficients of a SAW transducer

Examples

1) Calculate the capacitive coefficients of a SAW transducer (*N*=11, *km*=0.5, *eps*=1.e-5):

```
[g]=ekpk(11,0.5,1e-5)
g =
    1.2646
   -0.4332
   -0.0941
   -0.0465
   -0.0317
   -0.0267
```

Please note that only [N/2]+1=6 independent coefficients are calculated, with the others found by symmetry *g*(*n*)=*g*(*N*-*n*+1), *n*=1,2, …*N*, if necessary.

2) Check the charge conservation law by summation of the of the capacitive coefficients within one generalized period:

```
sum(2*g)-g(1)
ans = 4.8850e-015
```

Note To calculate the potential coefficients (not used in this software), one should set the value of the internal variable *inv* to –1 in the C or Fortran subroutines *Ekpk.c* or *Ekpk.f* and then generate a new MEX-file.

See also *Apeff, Capac.*

Elemfact.dll

Purpose Element factor calculation in the quasi-static approximation.

Synopsis *xi*=elemfact (*w,k2,ep,v0,p,km,f,type*);

Description Calculates the finger or gap (overlap) element factor.

Algorithm The finger (or gap) element factor is defined as the closed-form Fourier transform of the elemental charge density in the periodic structure with the only finger (or inter-electrode gap) activated [5, Chapter 4].

Arguments

w – acoustic aperture, m
ep – effective permittivity of the substrate
k2 – piezoelectric coupling factor, %
v0 – free-surface SAW velocity, m/s
p – strip period, m
km – metallization ratio
f – frequency, MHz (scalar or array)
type – acoustic source type ('f' – finger; 'g' - gap)

Returns *xi* – form-factor function

See also *Fleg*

Note The results for the gap element factor must be multiplied by a factor *2j* to agree with the Morgan's definition of the element factor [5, Chapter 4].

Fleg.dll

Purpose Legendre function calculation.

Synopsis *p*=Fleg(*nu,x*)

Description Calculates the Legendre function.

Algorithm The Legendre function is calculated using the following series expansion [6]

$$P_{\nu}(x) = \sum_{n=0}^{\infty} a_n \, , a_n = \frac{(n-1-\nu)(n+\nu)(1-x)}{2n^2} a_{n-1} \, , \;\; a_0 = 1, \quad |\, x \,| \leq 1$$

Arguments

nu – index (degree) of the Legendre function (integer or non-integer)
x – argument of the Legendre function

Returns *p* – Legendre function value

Examples

```
1) fleg(-0.5,cos(0.25*pi))
     ans = 1.0400
2) fleg(-0.25,cos(0.5*pi))
   ans = 1.1339
3) fleg(-0.5,cos(0.75*pi))
   ans = 1.5279
```

See also *Ekpk, Elemfact.*

2.1.2. MATLAB M-Functions

*Admit_g.m**

Purpose	SAW transducer admittance calculation for reflective SAW transducers.
Synopsis	[*yidt*,*wp*]=ADMIT_G (*ne*,*y*,*wmcm*,*km*,*ep*,*k2*,*ve*,*p*,*g*,*fmhz*) [*yidt*,*wp*]=ADMIT_G (*ne*,*y*,*wmcm*,*km*,*ep*,*k2*, *ve*,*p*,*g*,*fmhz*,*npt*)
Description	Calculates the dynamic admittance of the unapodized or apodized SAW transducer including radiation conductance and susceptance for reflective SAW transducers.
Algorithm	Unpublished.
Arguments	*ne* – number of electrodes *y* – transversal gap positions (array) *wmcm* – acoustic aperture, m *km* – metallization ratio *ep* – substrate effective permittivity *k2* – coupling factor, % *ve* – effective SAW velocity, m/s p – transducer period (pitch), m *g* – wavenumber found from the solution of the dispersion equation *fmhz* – frequency, MHz (scalar or vector) *npt* – number of frequency points (optional)
Returns	*yidt* – transducer admittance, $^{-1}$ *wp* – effective apertures (array)
See also	*Admit*, *Apeffd*.

Analyze.m

Purpose	Analysis of bidirectional SAW filters in the quasi-static approximation
Synopsis	[*S11*,*S12*,*S21*,*S22*,*Y11*,*Y12*,*Y21*,*Y22*,*C1*,*C2*,*M23_1*,*M13_2*]=analyze(*k2*,*ep*,*v0*, *ve*,*p*,*w*,*km*,*re*, *L*,*N1*,*N2*,*y1*,*y2*,*f*);
Description	Calculates *S*- and *Y*-parameters of a bidirectional SAW filter, static capacitance and acoustoelectric conversion functions of the input/output SAW transducers.
Algorithm	See *Filt_QS.m.*
Arguments	*k2* – piezoelectric coupling factor, % *ep* – substrate effective permittivity *v0* – free surface SAW velocity, m/s *ve* – effective SAW velocity, m/s *p* – input/output transducer finger period (pitch), um *w* – input/output transducer acoustic aperture, um *km* – input/output transducer metallization ratio *re* – input/output transducer finger reflection coefficient (re=0 in the quasi-static approximation)

	L – port-to-port separation of the input/output SAW transducers, um *N1* – finger number (input transducer) *N2* – finger number (output transducer) *y1* – *y*-coordinates of the transversal gaps or electrode voltages (input transducer) *y2* – *y*-coordinates of the transversal gaps or electrode voltages (output transducer) *f* – frequency (ies), MHz
Returns	*Sik* - *S*-parameters of a SAW filter in the 50-Ohm system *(i,k*=1,2*)* *Yik* - *Y*-parameters of a SAW filter (*i,k*=1,2) *C1* – static capacitance of the input transducer, *p*F *C2* – static capacitance of the output transducer, *p*F *M23_1* – acoustoelectric conversion function of the input transducer (m_{23}-element of the mixed scattering matrix) *M13_2* – acoustoelectric conversion function of the output transducer (m_{13}-element of the mixed scattering matrix)
Note	Transducer left (right) acoustic port is located at the distance *p*/2 from the center of the first (last) electrode that corresponds to the beginning of the first (end of the last) elemental cell where *p* is the transducer pitch (period).

Convert.m

Purpose	Conversion of the data files from the SAWFAT 1.1 format to SAWFAT 1.2 format
Synopsis	convert(*in_file*, *out_file*);
Description	Data file is converted from the old format (SAWFAT 1.1) to the new one (SAWFAT 1.2).
Arguments	*in_file* – name of the input (source) file (in the SAWFAT 1.1 format) *out_file* – name of the output (target) file (in the SAWFAT 1.2 format)
Returns	*No.*
Example	Convert*('filter_1.dat','filter1.dat')* – file *filter_1.dat* in the format of the SAWFAT 1.1 is converted to the file *filter1.dat* in the format SAWFAT 1.2.
Note	By default, a new file in the format SAWFAT 1.2 is placed in the same directory with the source file in the format SAWFAT 1.1 if the directory path is not specified explicitly in the file name.

Capacity.m

Purpose	Calculation of the static capacitance of a SAW transducer.
Synopsis	[*cpf,c*]=CAPACITY (*ne,y,w,km,ep*)
Description	Calculates static capacitance of an unapodized/apodized SAW transducer for arbitrary metallization ratio and apodization pattern.
Algorithm	The computational algorithm is based on the concept of the interelectrode capacitors of a periodic SAW transducer and comprises three major steps[1,

Chapter 4], [2, 4]:

1) calculation of a set of the partial effective apertures in terms of the finger taps;
2) calculation of the capacitive coefficients of a periodic SAW transducer;
3) weighted summation of the interelectrode capacitors, with the weights given by the effective apertures.

All three steps are implemented in this function.

Arguments *ne* - number of electrodes *y* - *y*-coordinates of the transversals gaps *w* - acoustic aperture, m *km* - metallization ratio (duty factor) *ep* - substrate effective permittivity

Returns *cpf* - static capacitance, *p*F
c - normalized capacitance (in terms of the capacitance of one finger pair in the solid-finger periodic transducer)

Examples Calculation of the static capacitance for a SAW transducer with *N=2*M* fingers where *M*=2,3,4 is the number of fingers in the period (acoustic aperture *w*=1000 m, metallization ratio *km*=0.5, substrate permittivity *ep*=50):

1) *M*=2 (+–+–)

```
[c2,cn2]=capacity(4,[1 -1 1 –1],1000,0.5,50)
 c2 = 0.9031
 cn2 =   2.0000
```

2) *M*=3 (++–++–)

```
[c3,cn3]=capacity(6,[1 1 –1 1 1 -1],1000,0.5,50)
 c3 = 1.0428
 cn3 =    2.3094
```

3) *M*=4 (++––++––)

```
[c4,cn4]=capacity(8,[1 1 –1 –1 1 1 -1 -1],1000,0.5,50)
 c4 = 1.2772
 cn4 =    2.8284
```

The ratios c3/c2=$1/\sqrt{2}$ and c4/c2=$2/\sqrt{3}$ agree with theoretical results [1, Chapter 4], [4, 5, 7].

See also *Apeff, Ekpk, Capac*

dB.m

Purpose Calculation of the logarithmic magnitude (in decibels).

Synopsis *f_db*=dB(*f*)

Description Computes logarithmic magnitude of the function in decibels.

Algorithm f, dB=20 log10(|f|)

Arguments *f* – the function value (real or complex)

Returns *f_db* – Log magnitude in decibels

Elemfact.m

Purpose
Synopsis
Description
Algorithm
Arguments
Returns

See MEX-function *Elemfact.DLL*.

Fleg.m

Purpose
Synopsis
Description
Algorithm
Arguments
Returns

See MEX-function *Fleg.DLL*

Filt_QS.m

Purpose SAW filter analysis in the quasi-static approximation.

Synopsis [*y11*,*y12*,*y21*,*y22*]=FILT_QS(*M1*,*M2*,*y1*,*y2*,*v*,*f*,*L*)
[*y11*,*y12*,*y21*,*y22*,*s11*,*s12*,*s21*,*s22*]=FILT_QS(*M1*,*M2*,*y1*,*y2*,*v*,*f*,*L*)

Description Calculates *Y*- and *S*-parameters of a bidirectional SAW filter

Algorithm Supposed for a short-circuit SAW transducer to be lossless and non-reflective in the quasi-static approximation, *Y*-parameters of a SAW filter are found in the closed-form [1, Chapter 6], [8]. *S*-parameters are expressed in terms of the *Y*-parameters [1, Chapter 1].

Arguments
M1 - acoustoelectric conversion function of the input IDT in the right direction (M23-element)
M2 - acoustoelectric conversion function of the output IDT in the left direction (M13-element)
Y1 - admittance of the input IDT (*Y1*=*G1*+*jB1*+*jwC1*)
Y2 - admittance of the output IDT (*Y2*=*G2*+*jB2*+*jwC2*)
v - free surface SAW velocity, m/s
f - frequency, MHz (scalar or array)
L - port-to-port separation of the input/output SAW transducers, mum

Returns
Yik - *Y*-parameters of a SAW filter (*i*,*k*=1,2)
Sik - *S*-parameters of a SAW filter in the 50-Ohm system (*i*,*k*=1,2)

Note Transducer left (right) acoustic port is located at the distance *p*/2 from the center of the first (last) electrode that corresponds to the beginning of the first (end of the last) elemental cell where *p* is the transducer pitch (period).

*Gap.m**

Purpose Calculation of the ideal frequency response (array factor) of a SAW transducer in terms of the gap voltages.

Synopsis [*m13*,*m23*]=gap(*N*,*V*,*p*,*g*);
[*m13*,*m23*]=gap(*N*,*V*,*p*,*g*,*ng*);

Description Given the SAW wavenumber in a periodic SAW transducer found from the solution of the dispersion equation, calculates the ideal frequency response (array factor) of an unapodized/apodized SAW transducer using gap (overlap) taps.

Algorithm Unpublished.

Arguments
N – number of fingers
V – vector of the finger potentials (coordinates of the finger breaks for apodized SAW transducer)
p – transducer period (pitch), m
g – wavenumber (scalar or vector) *ng* – number of the wavenumber points (optional)

Returns
m13 – acoustoelectric conversion function in the left direction (m_{13}-element)
m23 – acoustoelectric conversion function in the right direction (m_{23}-element)

Note Phase reference is located at the distance *p*/2 apart from the center of the first (last) finger for *m13* (*m23*) function.

Idealfrd.m

Purpose Calculation of the ideal frequency response (array factor) of a SAW transducer.

Synopsis fr=idealFRD(*fpi*,*f*,*v*)
fr=idealFRD(*fpi*,*f*,*v*,*m*)

Description Calculates the ideal frequency response (array factor) of an unapodized/apodized SAW transducer using gap (overlap) taps.

Algorithm The ideal frequency response of a SAW transducer is calculated as the Fourier transform of the gap voltages (or finger overlaps for the apodized SAW transducers) [1, Chapter 2], [5, Chapter 4].

Arguments
fpi – synchronous frequency, MHz *f* – frequency, MHz (scalar or vector) *v* – electrode voltages (*y*-coordinates of the transversal gaps for apodized SAW transducers) *m* – string variable to specify electroacoustic function ('m13' or 'm23') (optional, mm='m23' by default if omitted)

Returns
fr – ideal frequency response

Note Phase of the frequency response is referenced to a transducer center.

*IDT_CF.m**

Purpose Calculation of the mixed scattering parameters of an unapodized/apodized reflective periodic SAW transducer.

Synopsis [*m11,m12,m13,m23,m33*]=IDT_CF (*k2,ep,v0,p,km,w,n,v,re,ve,alpha,f*)
[*m11,m12,m13,m23,m33,c*]=IDT_CF (*k2,ep,v0,p,km,w,n,v,re,ve,alpha,f*)

Description Calculates the independent elements of the mixed scattering matrix (*P*-matrix) of the reflective periodic SAW transducer.

Algorithm Unpublished.

Arguments
k2 – piezoelectric coupling factor, %
ep – substrate effective permittivity
v0 – free-surface SAW velocity, m/s
p – finger period (pitch), m
km – metallization ratio (duty factor)
w – acoustic aperture, m
n – finger number
v – electrode voltages (or coordinates of the transversal gaps)
re – finger reflection coefficient
ve – effective SAW velocity in the transducer, m/s
alpha – propagation attenuation constant per period, Np *f* – frequency, MHz (scalar or vector)

Returns
m11 – reflection coefficient (dimensionless)
m12 – transmission coefficient (dimensionless)
m13 – electroacoustic transfer function (to the left), sqrt($^{-1}$)
m23 – electroacoustic transfer function (to the right), sqrt($^{-1}$)
m33 – IDT admittance, $^{-1}$
c – static capacitance, *p*F

Note
1. Phase of the variables at the left (right) acoustic ports is referenced to the beginning of the first (end of the last) elemental cell.
2. The reflection coefficient *m11* of a periodic SAW transducer is identically equal to zero, in the quasi-static approximation.

See also *Elemfact, Idealfrd, Admit, Capacity*.

IDT_IDT.m

Purpose Cascading two SAW transducers (in-line).

Synopsis [*y11,y12,y21,y22,s11,s12,s21,s22*]=IDT_IDT
(*m11_1,m12_1,m13_1,m23_1,m33_1,*
m11_2,m12_2,m13_2,m23_2,m33_2,f,v0,l);

Description Calculates *Y*- and *S*-parameters of a bidirectional SAW filter comprising two reflective SAW transducers

Algorithm Supposed for a mixed scattering matrix of each transducer to be known a priori, the cascading algorithm follows the guidelines described in [1, Chapter 6] using the augmented transmission matrices of the input/output SAW transducer. *S*-parameters are expressed in terms of the *Y*-parameters [1, Chapter 1].

Arguments *mik_1* – mixed scattering matrix elements of the input SAW transducer, *i*,*k*=1,2,3
mik_2 – mixed scattering matrix elements of the output SAW transducer, *i*,*k*=1,2,3
v0 - free surface SAW velocity, m/s
f - frequency, MHz (scalar or array)
l - port-to-port separation of the input/output SAW transducers, m

Returns *Yik* - *Y*-parameters of a SAW filter (*i*,*k*=1,2)
Sik - *S*-parameters of a SAW filter in the 50-Ohm system *(i,k*=1,2*)*

Note Transducer left (right) acoustic port is located at the distance *p*/2 from the center of the first (last) electrode that corresponds to the beginning of the first (end of the last) elemental cell where *p* is the transducer pitch (period).

Note Phase of the variables at the left (right) acoustic ports is referenced to the beginning of the first (end of the last) elemental cell which are located at the distance *p*/2 apart from the center of the first (last) transducer finger.

MtoT.m

Purpose Transformation of the mixed scattering matrix **M** into the mixed transmission matrix **T.**

Synopsis *T*=MtoT(*M*)

Description Transforms in the closed-form the mixed scattering matrix **M=**[m_{ik}], *i*,*k*=1,2,3, relating the amplitudes of the reflected waves at the acoustic ports and terminal current at the electric port with the amplitudes of the incident waves and the terminal voltage to the mixed transmission matrix **T** relating the acoustic wave amplitudes at the left acoustic port and the terminal current with the acoustic waves at the right acoustic port and the terminal voltage at the electric port.

Algorithm The closed-form equations are deduced from the solution of the linear system of equations in terms of the mixed scattering matrix with respect to a new set of the unknown variables which are the amplitudes of the acoustic wave at the left acoustic port and the terminal current at the electric port [1, Chapter 1].

Arguments *M* – mixed scattering matrix of a SAW transducer

Returns *T* – mixed transmission matrix of a SAW transducer

MtoY.m

Purpose Transformation of the mixed scattering matrix **M** into the admittance matrix **Y**.

Synopsis *Y*=MtoY (*M*,*n*)

Description Transforms the mixed scattering matrix **M** with an arbitrary number of the electric and acoustic ports (in general case) to the admittance matrix **Y**.

Algorithm The transformation is based on the relationship between acoustic and electric variables in the multi-port mixed units network [1, Chapter 1].

Arguments *M* – mixed scattering matrix of the multi-port network
n – number of the acoustic ports

Returns *Y* – admittance matrix of the multi-port network

Note The unit characteristic admittance is arbitrarily attributed to all the acoustic ports.

PlotData.m

Purpose Plot of *S*-parameters (in decibels) versus frequency (in MHz) from a formatted data file.

Synopsis plotdata(*filename*,*parameter*);
plotdata(*filename*,*parameter*,*style*);
[*f*,*s11*,*s12*,*s21*,*s22*]=plotdata(*filename*,*parameter*,*style*);

Description Reads and plots the specified *S*-parameters from the formatted data file using a specified color and line style.

Arguments *filename* – name of the data file including extension *parameter* – character string to specify the name of the S-parameter to be plotted ('S11' | 'S12' | 'S21' | 'S22') *style* - MATLAB character string to specify the line style (see MATLAB definition for the standard built-in function PLOT) (optional)

Returns *f* - frequency, MHz (scalar or vector) *sij, i,j =1,2* - *S*-parameters

Example 1) To plot *S12* versus frequency *f* from the data file *Filter_1.s* (the line color is red, the line type is solid), type the following MATLAB command:

plotdata(‘*Filter_1.s’*,*’S12’,’r*-‘).

2) The same example, with *S*-parameters plotted and exported from the text file *Filter_1.s* into MATLAB workspace:

[f,S11,S12,S21,S22]=plotdata(‘*Filter_1.s’*,*’S12’,’r*-‘).

Note Format of the text file containing S-parameters is as follows:

f, MHz Re{S_{11}} Im{S_{11}} Re{S_{12}} Im{S_{12}} Re{S_{21}} Im{S_{21}} Re{S_{22}} Im{S_{22}}

where any fixed or floating point format providing a sufficient accuracy can be used.

Example:

40.00 0.92411 -0.35864 0.00011 -0.00017 0.00011 –0.00017 0.82782 -0.53714
40.05 0.92419 -0.35898 0.00005 -0.00015 0.00005 –0.00015 0.82742 -0.53805

Read_F.m

Purpose Input of the SAW filter data from the text file to be used with *TEST_QSF*, *TEST_MSF* (SAWFAT 1.1).

Synopsis [*substrate*,*k2*,*ep*,*v0*,*ve*,*re*,*km*,*f0*,*fpi*,*w*,*l*,*fs*,*fe*,*df*,*y1*,*y2*]=read_F(*data_file*) [*substrate,k2,ep,v0,ve,re,km,f0,fpi,w,l,fs,fe,df,y1,y2,inp,out*]=read_F(*data_file*)

Description Reads the SAW filter data including transducer topologies from the text file.

Arguments *data_file* – name of the data file

Returns
Substrate – substrate material (string variable)
k2 – piezoelectric coupling factor, %
ep – substrate effective permittivity
v0 - free-surface SAW velocity, m/s
ve – effective SAW velocity, m/s
re - finger reflection coefficient (*re*=0 in the quasi-static approximation)
km – metallization ratio
f0 - SAW filter central frequency, MHz
fpi – synchronous frequency of a SAW transducer, MHz
w - acoustic aperture, m
l – input/output transducer port-to-port separation, m
fs - start frequency of the analysis range, MHz
fe - end frequency of the analysis range, MHz
df – discretization interval, MHz
y1 - *y*-coordinates of the transversal gaps of the input SAW transducer
y2 - *y*-coordinates of the transversal gaps of the output SAW transducer
inp – name of the file containing input tap weights
out – name of the file containing output tap weights

Note Arguments *ve,re,km,f0,fpi,w* are scalar variables assumed to be the same for both SAW transducers in SAWFAT 1.1, for simplicity.

Read_F2.m

Purpose Input of the SAW filter data from the text file to be used with *SAWFAT*, *TEST_QSF* (SAWFAT 1.2).

Synopsis
[*substrate,comment,k2,ep,v0,fpi,ve,p,w,km,re,Lp,Lc,f0,fs,fe,df,y1,y2*]= read_F(*data_file*)
[*substrate,comment,k2,ep,v0,fpi,ve,p,w,km,re,Lp,Lc,f0,fs,fe,df,y1,y2,inp,out*]= read_F(*data_file*)

Description Reads the SAW filter data including transducer topologies from the text file.

Arguments *data_file* – name of the data file

Returns
substrate – substrate material (string variable)
comment - commentary text (two first lines)
k2 – piezoelectric coupling factor, %
ep – substrate effective permittivity
v0 – free-surface SAW velocity, m/s
fpi – synchronous frequencies of SAW transducers, MHz
ve – effective SAW velocities under SAW transducers, m/s
p – period (pitch) of SAW transducers, m
w - acoustic apertures of SAW transducers, m
km – transducer metallization ratios
re - finger reflection coefficient for each transducer

(*re*=0 in the quasi-static approximation)
Lp – transducer port-to-port separation, m
Lc – transducer center-to-center separation, m
f0 – SAW filter central frequency, MHz
fs - start frequency of the analysis range, MHz
fe - end frequency of the analysis range, MHz
df – discretization interval, MHz
y1 - *y*-coordinates of the transversal gaps of the input SAW transducer
y2 - *y*-coordinates of the transversal gaps of the output SAW transducer
inp – name of the file containing input tap weights
out – name of the file containing output tap weights

Note Arguments *fpi,ve,p,w,km,re* are two-element vectors containing parameters for the input and output SAW transducer, respectively, in SAWFAT 1.2.

StoT.m

Purpose Transformation of a wave scattering matrix **S** to the transmission matrix **T**.

Synopsis *T*=StoT(*S*)

Description Converts a square wave scattering matrix **S** of the multi-port network to the transmission matrix **T**.

Algorithm The transformation is based on the relationship between acoustic and electric variables in a multi-port network [1, Chapter 1].

Arguments *S* - scattering matrix (*N* by *N*)

Returns *T* – transmission matrix (*N* by *N*)

*Wavenum.m**

Purpose Closed-form solution of the dispersion equation.

Synopsis *g*=Wavenum (*f, ve,alpha,re,p*)
[*g,r,t*]=Wavenum (*f, ve,alpha,re,p*)

Description Solves in the closed-form the dispersion equation for short-circuit grating to find SAW propagation constant

Algorithm Unpublished.

Arguments *f* - frequency, MHz (scalar or vector)
ve – effective SAW velocity, m/s
alpha – attenuation constant per period, Np
re - finger reflection coefficient (dimensionless)
p - transducer period (pitch), m

Returns g - propagation constant (complex wavenumber) *r, t* - reflection and transmission coefficients of the elemental cell (optional)

Y_to_S.m

Purpose Transformation of the admittance matrix **Y** into the wave scattering matrix **S** for an arbitrary multi-port network.

Synopsis *S*=Y_to_S (*Y*)

Description Transforms the admittance matrix **Y** into the wave scattering matrix **S** for an arbitrary multi-port network.

Algorithm The transformation is based on the relationship between acoustic and electric variables in a multi-port network [1, Chapter 1].

Arguments *Y* – admittance matrix of the multi-port network

Returns *S* – scattering matrix of the multi-port network

Note The unit characteristic admittance is arbitrarily attributed to all the ports.

YtoS2.m

Purpose Transformation of the admittance matrix **Y** into the wave scattering matrix **S** for a two-port network.

Synopsis *S*=Y_to_S (*Y*)

Description Transforms in the closed-form the admittance matrix **Y** into the wave scattering matrix **S** for an arbitrary two-port network.

Algorithm The transformation is based on the closed-form relationship between acoustic and electric variables in a two-port network [1, Chapter 1].

Arguments *Y* – admittance matrix of the two-port network (2 by 2)

Returns *S* – scattering matrix of the two-port network (2 by 2)

Note The unit characteristic admittance is arbitrarily attributed to all the ports.

Y2toS2.m

Purpose Transformation of the vectorized admittance matrix **Y** into the vectorized wave scattering matrix **S** for a two-port network with arbitrary characteristic impedance.

Synopsis *[s11,s12,s21,s22]=Y2toS2(y11,y12,y21,y22,yg,yl)*

Description Converts the vectorized two port Y-matrix with elements *y11*, *y12*, *y21*, *y22* to the vectorized S-matrix with elements *s11*, *s12*, *s21*, *s22* for arbitrary real input/output characteristic admittance.

Algorithm The transformation is based on the closed-form relationship between acoustic and electric variables in a two-port network [1, Chapter 1].

Arguments *y11, y12, y21, y22* – vectorized elements of the admittance matrix of the two-port network

yg – input (source) characteristic admittance
yl – output (load) characteristic admittance

Returns — *s11*, *s12*, *s21*, *s22* – vectorized elements of the admittance matrix of the two-port network

2.1.3. MATLAB M-files

TEST_C.m

Purpose — Test program for the static capacitance calculation.

Synopsis — TEST_C

Description — Calculates the static capacitance of the periodic unapodized SAW transducer for three different metallization ratios η=0.25,0.5,0.75. SAW transducer parameters:

number of fingers N=100;
acoustic aperture W=1000   m;
substrate effective permittivity Ep=55;
metallization ratio Km=[0.25 0.5 0.75];
polarity sequence + Ĕ + Ĕ + Ĕ…

Algorithm — The test program calls the subroutine *capacity* to calculate the static capacitance of the solid-finger SAW transducer for different metallization ratios.

Arguments — No

Returns — No

Test results — TEST_C

Metallization ratio = 0.25
Transducer capacitance = 16.9 pF
Normalized capacitance = 34.0

Metallization ratio = 0.50
Transducer capacitance = 24.8 pF
Normalized capacitance = 50.0 (=N/2)

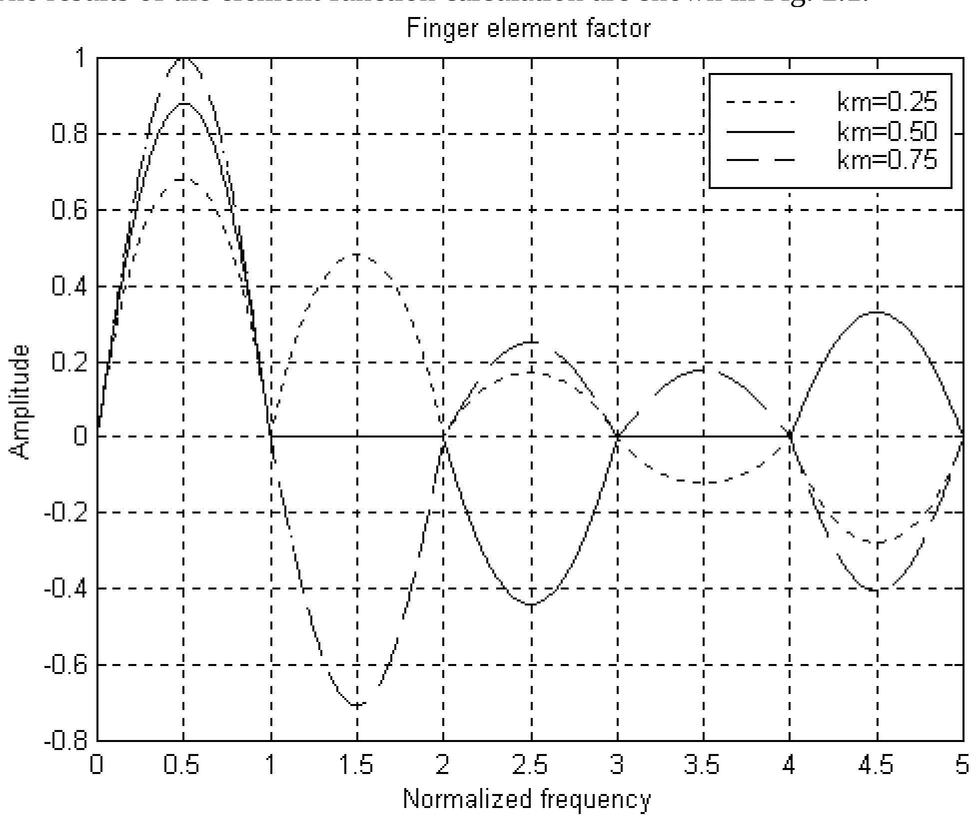
Metallization ratio = 0.75
Transducer capacitance = 36.4 pF
Normalized capacitance = 73.5

For the metallization ratio η=0.5, the normalized SAW transducer capacitance is equal to the number of finger pairs. This agrees with the theory [1, Chapter 3], [4, 5, 7] where the static capacitance of one finger pair per unit length is equal to $C_0=\varepsilon_0+\varepsilon_p$ for η=0.5 and hence the normalized capacitance of one finger pair C/C_0 =1.

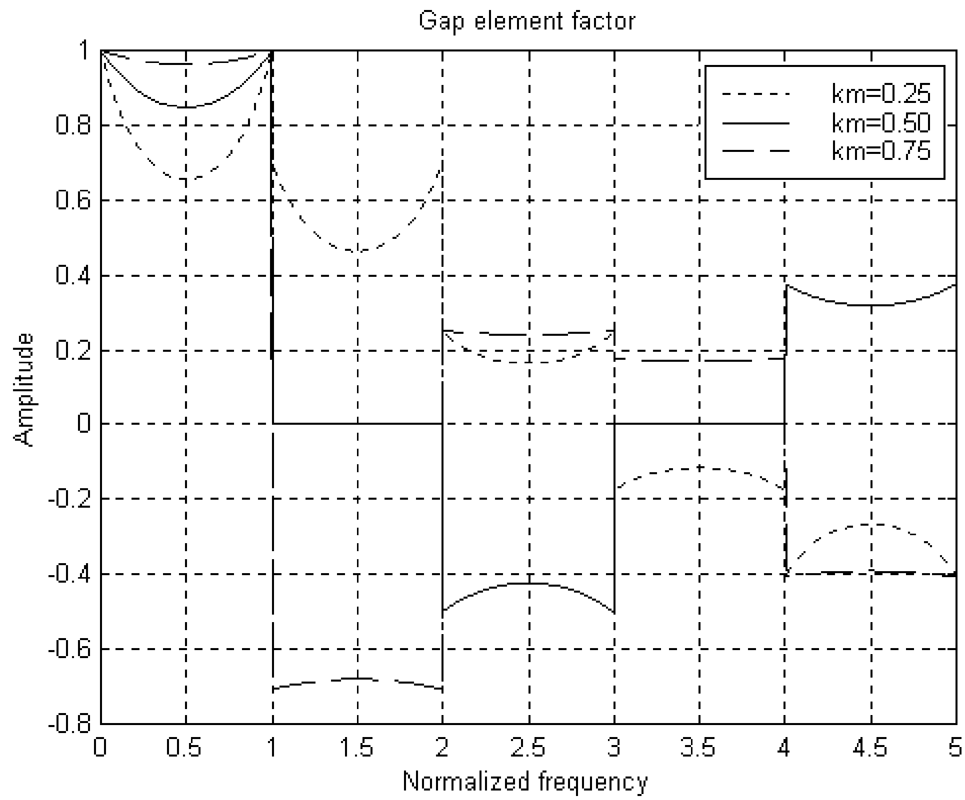
See also — *Capacity*.

TEST_EF.m

Purpose	Test program for the element factor calculation.
Synopsis	TEST_EF
Description	Calculates the normalized finger and gap element factors versus normalized frequency =*f*/2*f* for three different metallization ratios η=0.25,0.5,0.75 at first five space harmonics (0 ≤ ≤ 5).
Algorithm	The finger (or gap) element factor is defined as the closed-form Fourier transform of the elemental charge density in the periodic structure with the only finger (or inter-electrode gap) activated [1, Chapter 2], [5, Chapter 4].
Arguments	No
Returns	No
See also	*Elemfact.*
Test results	The results of the element function calculation are shown in Fig. 2.1.



a) Finger element factor



b) Gap element factor

Fig. 2.1. Normalized element factor for metallization ratios η=0.25,0.5,0.75

TEST_QSF.m

Purpose	Test program for SAW filter analysis in the quasi-static approximation based on the mixed scattering matrix (*P*-matrix) calculation of SAW transducers.
Synopsis	TEST_QSF
Description	Calculates the independent elements of the mixed scattering matrix (acoustoelectric conversion function and SAW transducer admittance including the radiation conductance, susceptance, and static capacitance). Performs SAW filter analysis based on the mixed scattering matrices of the constituent input/output SAW transducers. The initial data are read from a formatted data file. The output results are presented in terms of the *S*- and/or *Y*-parameters of a SAW filter.
Algorithm	See [1, Chapters 1-4, 6] for the detailed theory and closed-form equations.
Arguments	No
Returns	No
Test Results	See Chapter 3 of this Manual.
Note	Synchronous frequencies of the input/output SAW transducers are assumed to be the same, for simplicity. Minor modifications are required to generalize the program to the case of the different synchronous frequencies.

2.1.4. C and Fortran Computational Subroutines (Optional) ***Supplied Separately***

Admit.c, Admit.f

Purpose

SAW transducer admittance calculation.

C Syntax

void admit (long int **ne*, double **y*, double **wmcm*,
double **km*, double **ep*, double * *k2*,
double * *fp*i, double * *fmhz*, long int **npt*,
complex **yidt*, double **wp*)

Fortran Syntax

Subroutine ADMIT (*ne,y,wmcm,km,ep,k2,fpi,fmhz,npt*, *yidt,wp*)
Implicit Integer *4 (I-N)
Implicit Real *8 (A-H, O-Z)
Real *8 *km, k2*
Real *8 *fmhz(*), wp(*), y(*)*
Complex *16 *yidt(*)*

Description

Calculates the dynamic admittance of an unapodized or apodized SAW transducer including radiation conductance and susceptance.

Algorithm

The computational algorithm is based on the concept of the nodal admittance matrix of a periodic SAW transducer and comprises three major steps [1, Chapter 3], [2, 3]:

4) Calculation of a set of the effective apertures in terms of the finger overlaps;
5) calculation of the nodal admittance matrix of the periodic SAW transducer;
6) weighted summation of the elemental admittances, with the weights given by the effective apertures.

Arguments

ne – number of electrodes
y – transversal gap positions (array)
wmcm – acoustic aperture, m
km – metallization ratio
ep – substrate effective permittivity
k2 – coupling factor, %
fpi – synchronous frequency, MHz
fmhz – frequency, MHz (scalar or vector)
npt – number of frequency points (optional)
yidt – transducer admittance, Ω^{-1} *wp* – effective apertures

See also

Apeffd

Apeff.c, Apeff.f

Purpose Calculation of the partial effective apertures for capacitance calculation.

C Syntax void apeff (long int **n*, double **y*, double **wp*)

Fortran Syntax Subroutine Apeff (*n*,*y*,*wp*) Implicit Integer *4 (I-N) Implicit Real *8 (A-H,O-Z) Real *8 *y(*), wp(*)*

Description Calculates a set of the effective apertures in terms of the finger taps.

Algorithm Partial effective apertures for static capacitance calculation are defined as the total overlaps of all the neighbor fingers, next neighbor ones, etc. [1, Chapter 4], [2, 4].

Arguments
ne – number of electrodes
y – *y*-coordinates of the transversal gaps
wp – finger effective apertures

Apeffd.c, Apeffd.f

Purpose Calculation of the partial effective apertures for dynamic admittance calculation.

C Syntax void apeffd (long int **n*, double **y*, double **wp*)

Fortran Syntax Subroutine Apeffd (*n*,*y*,*wp*)
Integer *4 *n*
Real *8 *y(*), wp(*)*

Description Calculates a set of the effective apertures in terms of the finger overlaps (overlap taps).

Algorithm Partial effective apertures for the dynamic admittance calculation are defined as the total overlaps of all the neighbor gaps, next neighbor ones, etc. [1, Chapter 3], [2, 3].

Arguments
ne – number of electrodes
y – *y*-coordinates of the transversal gaps
wp – gap effective apertures

Capac.c, Capac.f

Purpose Static capacitance calculation.

C Syntax void capac (long int **n*, double **wp*, double **wmcm*, long int **nc*,
double **cp*, double * *ep*, double **c*, double **cpf*)

Fortran Syntax Subroutine Capac (*n*,*wp*,*wmcm*,*nc*,*cp*,*ep*, *c*, *cpf*) Implicit Real *8 (A-H, O-Z) Integer *4 *n*, *nc* Real *8 *wp(*)*,*cp(*)*

Description Given a set of the partial apertures *wp* and capacitive coefficients *cp*, calculates the static capacitance of an unapodized/apodized SAW transducer.

Algorithm

The computational algorithm is based on the concept of the interelectrode capacitors of a periodic SAW transducer and comprises three major steps [1, Chapter 4], [2, 4]:

4) calculation of a set of the partial effective apertures in terms of the finger taps;
5) calculation of the capacitive coefficients of the periodic SAW transducer;
6) weighted summation of the interelectrode capacitors, with the weights given by the effective apertures.

This function implements the third step, supposed for the effective apertures and capacitive coefficients found at the earlier steps.

Arguments

ne – number of electrodes
wp – set of the partial apertures
wmcm – acoustic aperture,   m
nc – number of the retained capacitive coefficients
c*p* – capacitive coefficients
ep – substrate effective permittivity
y – y-coordinates of the transversals gaps
c – normalized capacitance (to capacitance of one finger pair in a solid-finger periodic transducer)
cpf – static capacitance, *p*F

See also

Apeff.

Ekpk.dll

Purpose

Calculation of the capacitive (or potential) coefficients.

C Syntax

void ekpk (long int **n*, double * *km*, long int **ngs*, double **gs*, long int **ng* , double **g*, double **eps*)

Fortran Syntax

Subroutine EKPK (*n,km,ngs,gs,ng,g,eps*)   Real *8 *km, eps, gs(*), g(*)*   Integer *4 *n, ng, ngs*

Description

Calculates capacitive (or potential) coefficients of a periodic SAW transducer for the arbitrary metallization ratio.

Algorithm

The capacitive (or potential) coefficients are found for one generalized period of the periodically replicated SAW transducer as the superposition of the elemental phased transducers [1, Chapter 4], [2, 4].

Arguments

ne – number of electrodes
km – metallization ratio (duty factor)
eps – relative error for calculation of the coefficients
gs – capacitive or potential coefficients of the elemental phased transducers   *g* – capacitive or potentials coefficients of a SAW transducer

Note

To calculate the potential coefficients (not used in this software), one should set the value of the internal variable *Inv* to –1 in the C or Fortran subroutines *Ekpk.c* or *Ekpk.f* and then generate a new MEX-file.

See also — *Apeff, Capac*.

Elemfact.c, Elemfact.f

Purpose — Element factor calculation in the quasi-static approximation.

C Syntax — void elemfact (double **w*, double **k2*, double **ep*, double **v*, double **p*, double **km*, double **f*, long int **npt*, char ***type*, double **xi*)

Fortran Syntax — Subroutine Elemfact (*w,k2,ep,v,p,km,f,npt,type, xi*) Implicit Integer *4 (I-N) Implicit Real *8 (A-H, O-Z) Real *8 *f(*), xi(*)* Real *8 *k2, km* Character *type*

Description — Calculates the finger or gap (overlap) element factor.

Algorithm — The finger (or gap) element factor is defined as the closed-form Fourier transform of the elemental charge density in the periodic structure with the only finger (or inter-electrode gap) activated [1, Chapter 1], [5, Chapter 4].

Arguments

w – acoustic aperture, m
ep – effective permittivity of the substrate
k2 – piezoelectric coupling factor, %
v0 – free-surface SAW velocity, m/s
p – strip period, m
km – metallization ratio
f – frequency, MHz (scalar or vector)
type – acoustic source type ('f' – finger; 'g' – gap)
xi – form-factor function

See also — *Fleg*.

Note — The results for the gap element factor must be multiplied by a factor *2j* to agree with the Morgan’s definition of the element factor [5, Chapter 4].

Fleg.c, Fleg.f

Purpose — Legendre function calculation.

C Syntax — void fleg (double **s*, double **z*, double **f*, long int **ns*)

Fortran Syntax — Subroutine FLeg (*s,z,f,ns*)
Integer *4 *ns*
Real *8 *z*
Real *8 *s(*), f(*)*

Description — Calculates the Legendre function.

Algorithm — The Legendre function is calculated using the following series expansion [6]:

$$P_\nu(x) = \sum_{n=0}^{\infty} a_n \, , \, a_n = \frac{(n-1-\nu)(n+\nu)(1-x)}{2n^2} a_{n-1} \, , \; a_0 = 1, \quad | x | \le 1$$

Arguments — *nu* – index (degree) of the Legendre function (integer or non-integer) *x* – argument of the Legendre function

Returns *p* – Legendre function value

See also *Ekpk, Elemfact*.

Maxmin.c, Maxmin.f

Purpose Calculation of the minimum and maximum values in an array.

C Syntax void maxmin (long int **n*, double **a*, long int **imax*, long int **imin*)

Fortran Syntax
SUBROUTINE MAXMIN (*n*,*a*,*imax*,*imin*)
Integer *4 *n*,*imax*,*imin*
REAL *8 *a(*)*

Description Calculates the minimum and maximum values in an array and passes their indexes to the calling program.

Algorithm Direct search of the minimum and maximum values.

Arguments
n – number of the elements in the array
a – array
imax – index of the maximum element of the array
imin – index of the minimum element of the array

2.2. SAW Filter Analysis (GUI-directory)

2.2.1. MATLAB M-Functions

Analyze.m

Purpose	Analysis of bidirectional SAW filters in the quasi-static approximation
Synopsis	[*s11*,*s12*,*s21*,*s22*]=analyze(k2,ep,v0,ve,re,p,km,w,L,N1,N2,y1,y2,f) [*s11*,*s12*,*s21*,*s22*,*y11*,*y12*,*y21*,*y22*]=analyze(*k2,ep,v0,ve,re,p,km,w,L,…* *N1*,*N2*,*y1*,*y2*,*f*) [*s11*,*s12*,*s21*,*s22*,*y11*,*y12*,*y21*,*y22*,*c1*,*c2*]=analyze(*k2,ep,v0,ve,re,p,km,w,…* *L*,*N1*,*N2*,*y1*,*y2*,*f*)
Description	Calculates *Y*- and *S*-parameters of a bidirectional SAW filter
Algorithm	Supposed for a short-circuit SAW transducer to be lossless and non-reflective in the quasi-static approximation, the *Y*-parameters of a SAW filter are found in the closed-form [1, Chapter 6], [8]. *Y*-parameters are converted into *S*-parameters [1, Chapter 1].
Arguments	*k2* - piezoelectric coupling factor, % *ep* - substrate effective permittivity *v0* - free surface SAW velocity, m/s *ve* - effective SAW velocity, m/s *re* - finger reflection coefficient (*re*=0 in the quasi-static approximation) *p* - finger period (pitch), mum *km* - metallization ratio *w* - acoustic aperture, mum *L* - port-to-port separation of the input/output SAW transducers, mum *N1* - finger number (input transducer) *N2* - finger number (output transducer) *y1* - *y*-coordinates of the transversal gaps or electrode voltages (input transducer) *y2* - *y*-coordinates of the transversal gaps or electrode voltages (output transducer) *f* - *frequency (ies), MHz*
Returns	*Yik* - *Y*-parameters of a SAW filter (*i*,*k*=1,2) *Sik* - *S*-parameters of a SAW filter in the 50-Ohm system (*i*,*k*=1,2) *C1* - input transducer static capacitance, *p*F *C2* - output transducer static capacitance, *p*F
See also	*Filt_QS, IDT_QS*
Note	Transducer left (right) acoustic port is located at the distance *p*/2 from the center of the first (last) electrode that corresponds to the beginning of the first (end of the last) elemental cell where *p* is the transducer pitch (period).

Draw_IDT.m

Purpose Drawing of SAW transducer topology

Synopsis draw_IDT (*nidt,xo,yo,x,y,d,dc,dl,w,color,options,d,de*)

Description Draws SAW transducer topology as appears on the photomask.

Arguments
nidt - transducer number (*nidt*=1 – input transducer, *nidt*=2 - output transducer)
xo,yo - coordinates of the transducer center, um
x - vector containing *X*-coordinates of the finger centers, um
y - vector containing *Y*-coordinates of the transversal gap (finger break) centers, um
d - vector containing finger widths, um
dc - two-element vector containing the width of the upper and lower rectangular contact pads, um
dl - width of the transversal gap (finger break), um
color - two-character string with colors of the upper and lower parts of a SAW transducer to be drawn
shape – two-character string to control upper/lower pad primitive shape, with the character values: 'l'-left,'r'-right,'b'-both,'n'-none (rectangular)
delta - two-element vector with delta for the shaped pad (upper/lower), um
de - etching tolerance (to be added to the actual finger width), um

Returns No.

Note The meaning of the input arguments and basic topological elements will be explained in Chapter 3.

Drawtaps.m

Purpose Schematical drawing of the transducer tap weights.

Synopsis drawtaps(*y1*,*y2*)

Description Draws positions (*y*-coordinates) of the transversal gaps (finger breaks) in a periodic SAW transducer.

Algorithm None

Arguments
y1 - *y*-coordinates of the transversal gaps (input transducer)
y2 - *y*-coordinates of the transversal gaps (input transducer)

Returns No

Note Tap weights are given by the overlaps of the adjacent fingers.

Export.m

Purpose Export of *S*- or *Y*- parameters onto disk.

Synopsis [*err*]=export(*filename*,*f*,*S11*,*S12*,*S21*,*S22*,[*R0*])

Description Writes *S*- or *Y*-parameters in the text ASCII file.

Algorithm	Real and imaginary parts are saved separately for each parameter *Sik*, *i*,*k*=1,2
Arguments	*filename* - name of the text file *f* - frequency, MHz (scalar or vector) *Sik* – complex-valued parameters to be exported (*S*- or *Y*-parameters) *R0* - source/load characteristic resistance (obsolete)
Returns	*err* – error code (*err*=0 - normal exit)
See also	*Write.m*

Figure_.m

Purpose	Plot of the analysis results in the large scale format (figures).
Synopsis	figure_(*f*,*S11*,*S12*,*S21*,*S22*,*Y11*,*Y12*,*Y21*,*Y22*,*C1*,*C2*,*IL*)
Description	Plots the analysis results (*S*- or *Y*-parameters) superimposed on the same large scale figire in the active window.
Algorithm	None
Arguments	*Sik* - *S*-parameters *Yik* - *Y*-parameters *C1* - input transducer static capacitance, *p*F *C2* - output transducer static capacitance, *p*F *IL* - insertion loss, dB
Returns	None
See also	*Subplot_.m*

Fun_Help.m

Purpose	Utility function to display help text in the convenient format.
Synopsis	FUN_HELP (*titleStr*,*varargin*)
Description	Shows the help text.
Algorithm	None
Arguments	*titleStr* – help title (heading) *varargin* – variable number arguments containing help pages as the cell array elements
Returns	None
See also	*RD_help.m*

Inp_Dlg.m

Purpose	Utility function to display help text in the convenient format.
Synopsis	[*Answer*]=inp_dlg(*Prompt*, *Title*, *NumLines*, *DefAns*,*Resize*)
Description	Creates a modal dialog box that returns user input for multiple prompts in the cell array Answer.
Algorithm	The same as MATLAB standard dialog function *Inputdlg.m*, with a bug fixed.

Arguments See *Inputdlg.m* in MATLAB
Returns See *Inputdlg.m* in MATLAB
See also *Inputdlg.m*

Rd_Help.m

Purpose Reading a help text file from disk.
Synopsis [*help*]=RD_help(*help_file*)
Description Reads SAWFAT HELP from a text file on disk.
Algorithm Page-wise text reading to memory.
Arguments *help_file* - name of the text file containing HELP
Returns *help* – cell array containing help pages in the separate cells
Note Pages in the text file must be separated by a separator line as follows
----

Read_Ini.m

Purpose SAWFAT initialization from disk.
Synopsis [*SUBSTRATE,k2,ep,v0,ve,re,options,fileList*]=read_ini(*data_file*)
Description Reads SAWFAT initialization data (list of substrate materials, default parameter values, options, etc.) from the text file.
Algorithm None
Arguments *data_file* - name of the SAWFAT initialization file
Returns *SUBSTRATE* - list of the supported substrates
k2 – substrate piezoelectric coupling factors, %
ep – substrate effective permittivities
v0 - free-surface SAW velocities, m/s
ve – effective SAW velocities, m/s
re - finger reflection coefficients (re=0 in the quasi-static approximation)
options – options data
fileList – current list of the recent data files
See also *Wr_Ini.m*

Read_Top.m

Purpose Reading SAW filter topological data from a separate text file.
Synopsis [*dc1,dc2,dl,lmin,as,bs,xo,yo,de*]=read_top(*top_file*)
Description Reads SAW filter topological data (bonding pad width, transversal gap, etc.) from the text file with an extension *.*top*, by default.
Arguments *top_file* - name of the data file (*.top)
Returns *dc1* - contact pad width (upper/lower) for the input transducer

dc2 - contact pad width (upper/lower) for the output transducer
dl - transversal gap of input/output SAW transducers
lmin - minimum length of dummy fingers for input/output SAW transducers
as - chip (substrate) length, μm
bs - chip (substrate) width, μm
xo,yo - coordinates of the chip center, μm
de - etching tolerance, um

SAWFAT.m

Purpose Main recursive subroutine to implement various computational and GUI functions in SAW filter analysis program.

Synopsis sawfat (*action*,*s*)

Description Implements MATLAB GUI program functions (initialization, analysis, data control, results presentation and export, etc.) in the analysis program for bidirectional SAW filters in the quasi-static approximation.

Algorithm Closed-form theory of the periodic SAW transducers in the quasi-static approximation ([1-5]).

Arguments *action* – action to be performed
s - optional string variable to be passed if necessary

Returns No

See also *Analyze.m*

Smith_Ch.m

Purpose Smith chart representation of *S*-parameters.

Synopsis smith_ch (*S11*,*S22*,*color1*,*color2*)

Description Plots the reflection coefficients *S11* and *S22* on the Smith chart.

Algorithm Conversion from the reflection coefficient (polar form) to normalized impedances (or admittances) using the impedance (or admittance) circles on the Smith chart.

Arguments *S11, S22* – input/output *S*-parameters (reflection coefficients)
color1, *color2* - colors to plot the input/output *S*-parameters

Returns No

Subplot_.m

Purpose Plot of the analysis results in the compact (tiled) format (subplots)

Synopsis subplot_(*f*,*S11*,*S12*,*S21*,*S22*,*Y11*,*Y12*,*Y21*,*Y22*,*C1*,*C2*,*IL*)

Description Plots the analysis results (*S*- or *Y*-parameters) in the separate tiled subplots in the active window

Algorithm None

Arguments
Sik - *S*-parameters
Yik - *Y*-parameters
C1 - input transducer static capacitance, *p*F
C2 - output transducer static capacitance, *p*F
IL - insertion loss, dB

Returns None

See also *Figure_.m*

Top_DXF.m

Purpose Converting SAW filter topology (photomask pattern) to AutoCAD DXF-format.

Synopsis [*nc*,*np*]=top_dxf(*file_in*, *file_out*)
top_dxf(*file_in*, *file_out*)

Description Reads SAW filter topology (photomask pattern) from a text file (*.*msk* by default) and converts photomask pattern data to the file in the AutoCAD DXF-format (*.*DXF*).

Arguments *File_in* – name of the input (source) text file containing the description of a SAW filter topology (photomask pattern) in terms of contours of the metallized regions
File_out – name of the output (target) file containing the description of a SAW filter topology (photomask pattern) in terms of the closed polylines in the AutoCAD DXF-format

Returns *nc* - total number of contours (polylines) (optional)
np – total number of the points (vertices) in all polylines (optional)

See also *Wr_mask.m*

Note AutoCAD DXF-format initialization read-only file *ACAD.ini* is used that must be included in the MATLAB path.

Wr_ini.m

Purpose Backup SAWFAT initialization data on disk.

Synopsis WR_ini(*file_ini*,*SUBSTRATE*,*k2*,*ep*,*v0*,*ve*,*re*,*options*,*fileList*)

Description Saves SAWFAT initialization data (list of substrate materials, default parameter values, options, etc.) in the text file in a current directory.

Arguments
Data_file - name of the SAWFAT initialization file
SUBSTRATE - list of the supported substrates
k2 – substrate piezoelectric coupling factors, %
ep – substrate effective permittivities
v0 – free-surface SAW velocities, m/s
ve – effective SAW velocities, m/s
re – finger reflection coefficients (re=0 in the quasi-static approximation)
options – options data
fileList – current list of the recent data files

Returns No.

See also *Read_Ini.m*

Wr_mask.m

Purpose Writing SAW filter topology (photomask pattern) in the text file

Synopsis Wr_mask (*filename*, *x1,y1,x2,y2*, ... *xn,yn,dir*)

Description Writes up to 20 topological contours specified by the *XY*-duples *xn,yn*

Arguments
Filename - name of the output file
x1,y1, ... x20,y20 - vectors of X,Y-coordinates of the contour vertices in microns
dir - variable that specifies the contour directivity
dir=-1 - clock-wise (negative) direction
dir=+1 - anticlock-wise (positive) direction

Returns No

Note
1) Variable number (<=20) of input arguments can be used, with the ending parameter being the contour directivity
2) Contours are separated by a character string as follows

Write_F.m

Purpose Output of the SAW filter data in the text file.

Synopsis Write_f(*data_file, substrate,k2,ep,v0,ve,re,km,f0,fpi,w,L,fs,fe,df, ...*
file_taps1, file_taps2)

Description Writes the current SAW filter data including in the text file.

Arguments
Substrate – substrate material (string variable)
k2 – piezoelectric coupling factor, %
ep – substrate effective permittivity
v0 - free-surface SAW velocity, m/s
ve – effective SAW velocity, m/s
re - finger reflection coefficient (*re*=0 in the quasi-static approximation)
km – metallization ratio
f0 – SAW filter central frequency, MHz
fpi – synchronous frequency of SAW transducers, MHz
w - acoustic aperture, mum
L – input/output transducer port-to-port separation, mum
fs - start frequency of the analysis range, MHz
fe - end frequency of the analysis range, MHz
df – discretization interval, MHz
file_taps1 – name of the file containing tap weights of the input SAW transducer
file_taps2 – name of the file containing tap weights of the output SAW transducer

Returns No.

Note Files *file_taps1* and *file_taps2* contain *y*-coordinates of the transversal gaps.

See also *Read_F.m*

Writef.m

Purpose General-purpose function for formatted column-wise data output to the text file

Synopsis [*err*]=Writef (*filename, access, format, x1,x2,...,x20*)

Writef (*filename, access, format, x1,x2,...,x20*)

Description Writes column-wise data to a text file at a fixed format.

Arguments

filename - file name for writing data

access – file access mode which is one of the strings:

'w' - write to a new file

'a' - append to an existing file

format - string variable containing the numerical format

x1,x2, ...x20 - column-wise entries with the data to be saved

Returns *err* – error code (*err*=0 – normal exit)

Note

1) The numeric format is the same for all data columns.
2) Variable number of the column-wise entries is possible (<=20).

2.3. MSC Modeling (MSC-Directory) ***Optional***

2.3.1. MATLAB M-Functions

Disp_COM.m

Purpose Solution of the coupling-of-modes dispersion equation.

Synopsis [*g*]=DISP_COM (*v0,v,r,p,f*)
[*g,p*]=DISP_COM (*v0,v,r,p,f*)

Description Solves a COM-dispersion equation to find SAW propagation constant.

Algorithm COM dispersion equation has the form $g = \sqrt{d^2 - \kappa^2}$ where *d* is the detuning parameter and κ is the coupling coefficient [1, Chapter 5], [9-11].

Arguments *v0* - free-surface SAW velocity, m/s
v - effective SAW velocity in the grating, m/s *r* - reflection coefficient referenced to the strip center (dimensionless) *p* - period, m *f* - frequency, MHz (scalar or vector)

Returns g - SAW propagation constant (wavenumber) *p* - COM-parameter (harmonic coupling factor) $p = \frac{\kappa}{d + g}$ (optional)

Disp_FLD.m

Purpose Solution of the dispersion equation in the field approach.

Synopsis [*g,p*]=DISP_FLD (*v0,v,r,p,f*)

Description Solves iteratively the dispersion equation for short- and open- circuit grating to find SAW propagation constant using filed approach.

Algorithm Iterative solution of the transcendental dispersion equation [1, Chapter 5], [12].

Arguments *k2* - coupling factor, %
v0 - free-surface SAW velocity, m/s
v - effective SAW velocity in the grating, m/s
r - reflection coefficient (dimensionless)
p - period, m
km – metallization ratio
f - frequency, MHz (scalar or vector)
gr_type - grating type (‘oc’ or ‘sc’) (optional)

Returns g - propagation constant (wavenumber) *p* - harmonic coupling factor

Gr_COM.m

Purpose COM-analysis of a finite length periodic grating.

Synopsis [*r,t*]=GR_COM (*p,n,re,ve,v0,f*)

Description Calculates scattering parameters (reflection and transmission coefficients) of a finite length periodic grating using COM-analysis.

Algorithm Closed-form solutions of the COM-equations is deduced, with the propagation constant and harmonic coupling factor found a priori [1, Chapter 1], [9-11].

Arguments
p - grating period, m
n - number of electrodes
re – strip reflection coefficient(s) (scalar or vector) referenced to the electrode center
ve – effective SAW velocity (ies), m/s (scalar or vector)
v0 - free-surface SAW velocity, m/s
f - frequency, MHz (scalar or vector)

Returns *r* - reflection coefficient of the grating (versus frequency) *t* - transmission coefficient of the grating (versus frequency)

Note Reflection and transmission phases are referenced to the beginning of the first elemental cell.

See also *Disp_COM*.

Gr_FLD.m

Purpose Calculation of the scattering coefficients of a finite length periodic grating using field approach.

Synopsis [*r*,*t*]=GR_FLD (*p*,*n*,*re*,*ve*,*v0*,*k2*,*km*,*f*,*gr_type*)

Description Calculates the scattering parameters (reflection and transmission coefficients) of a finite length periodic grating using Ingebrigtsen's field approach.

Algorithm Closed-form equations deduced from the boundary conditions at the ends of the grating using the harmonic coupling factor [1, Chapter 5], [12].

Arguments
p - grating period, m
n - number of the electrodes
re – strip reflection coefficient(s) (scalar or vector) referenced to the electrode center
ve – effective SAW velocity (ies), m/s (scalar or vector)
v0 - free-surface SAW velocity, m/s
k2 – piezoelectric coupling factor, %
km – metallization ratio
f - frequency, MHz (scalar or vector)
gr_type - grating type (‘oc’ or ‘sc’)

Returns *r* - reflection coefficient of the grating (versus frequency) *t* - transmission coefficient of the grating (versus frequency)

Note Reflection and transmission phases are referenced to the beginning of the first elemental cell.

See also *Disp_FLD*

Gr_QS.m

Purpose Analysis of a finite length reflectionless periodic grating in the quasi-static approximation.

Synopsis [*r*,*t*]=GR_QS (*p*,*n*,*v0*,*k2*,*km*,*f*,*gr_type*)

Description Calculates the scattering parameters (reflection and transmission coefficients) of a short- and open-circuit periodic grating in the quasi-static approximation.

Algorithm Closed-form equations for short- and open-circuit grating wavenumbers are deduced in the quasi-static approximation [1, Chapter 5], [5].

Arguments *p* - grating period, m *n* - number of the electrodes *v0* - free-surface SAW velocity, m/s *k2* – piezoelectric coupling factor, % *km* – metallization ratio *f* - frequency, MHz (scalar or vector) *gr_type* - grating type (‘oc’ or ‘sc’)

Returns *r* - reflection coefficient of the grating (versus frequency), zero in the quasi-static approximation *t* - transmission coefficient of the grating (versus frequency)

Note Reflection and transmission phases are referenced to the beginning of the first elemental cell.

Gr_RAM.m

Purpose Calculation of the scattering coefficients of a finite length periodic grating using the Morgan’s reflective array model (RAM).

Synopsis [*r*,*t*]=GR_RAM (*p*,*n*,*re*,*ve*,*f*)

Description Calculates in the closed-form the scattering parameters (reflection and transmission coefficients) of a finite length periodic grating using Morgan's reflective array model (RAM).

Algorithm Closed-form equations are deduced, supposed for the scattering coefficients of the elemental cell to be given a priori [1, Chapter 5], [5, Appendix D].

Arguments *p* - grating period, m
n - number of the electrodes
re – strip reflection coefficient(s) (scalar or vector) referenced to the electrode center
ve – effective SAW velocity (ies), m/s (scalar or vector)
f - frequency, MHz (scalar or vector)

Returns *r* - reflection coefficient of the grating (versus frequency), zero in the quasi-static approximation *t* - transmission coefficient of the grating (versus frequency)

Note Reflection and transmission phases are referenced to the beginning of the first elemental cell.

MSC.m

Purpose Calculation of the MSC scattering parameters by different methods using two-mode approach.

Synopsis [*s11*,*s12*,*s13*,*s14*]=MSC(*k2*,*ep*,*v0*,*p*,*km*,*w*,*n*,*r*,*v*,*f*, *MSC_model*)

Description Calculates MSC scattering parameters using two-mode approach (expansion into symmetric and antisymmetric normal modes [1, Chapter 5], [12-14] which corresponds to the normal modes of the open- and short-circuit gratings, respectively).

Algorithm The MSC scattering matrix is defined in terms of the scattering coefficients of the normal modes [1, Chapter 5], [11, 12]. Four different methods are applied to characterize the symmetric and antisymmetric modes: 1) COM-analysis [9-11]; 2) Morgan's reflective array model (RAM) [5, Appendix E]; 3) Ingebrigtsen's field approach [12]; 4) quasi-static approximation [5, Chapter 5].

Arguments
k2 – piezoelectric coupling factor, %
ep - substrate effective permittivity
v0 - free-surface SAW velocity, m/s
p - strip period, m
km – metallization ratio
w - acoustic aperture, m (obsolete in two-mode assumption)
f - frequency, MHz (scalar or vector)
n - number of MSC strips
r(1,:) - short-circuit strip reflection factor(s) (scalar or vector)
r(2,:) - open-circuit strip reflection factor(s) (scalar or vector)
v(1,:) - short-circuit SAW velocity(ies), m/s (scalar or vector)
v(2,:) - open-circuit SAW velocity(ies), m/s (scalar or vector)
MSC_model – character string to specify MSC model:
- 'QS' – quasi-static approximation;
- 'RAM' – reflective array model (RAM);
- 'COM' – COM-analysis;
- 'FLD' – field approach

Returns *s1j* - independent elements of the MSC scattering matrix, *j*=1, 4

Note Phase is referenced to the beginning of the first elemental cell.

See also *Gr_QS*, *Gr_RAM*, *Gr_COM*, *Gr_FLD*.

MSCN.m

Purpose Evaluation of the optimum number of the MSC strips.

Synopsis *n*=MSCN(*k2*,*km*,*fo*,*fpi*)

Description Calculates the optimum number of the MSC strips for complete transfer of the acoustic power from one track to another.

Algorithm The optimum number of the MSC strips is found in the closed-form in the quasi-static approximation from the condition of the complete acoustic power transfer from track to another (S_{14}=1) [5, Chapter 5].

Arguments
k2 – piezoelectric coupling factor, %
km – metallization ratio
fo - MSC working ("central") frequency, MHz *fpi*- MSC synchronous frequency

Returns *N* - optimum number of the MSC strips.

R_SAW.m

Purpose Closed-form calculation of the strip reflection coefficient.

Synopsis *r*=R_SAW (*k2*,*v0*,*p*,*km*,*f*)
r=R_SAW (*k2*,*v0*,*p*,*km*,*f*,*gr_type*)

Description Calculates strip reflection coefficient in a short- and/or open-circuit periodic grating using Morgan's approximation (mass-loading neglected).

Algorithm Floquet analysis is applied to a periodic grating, with the closed-form solution in terms of the Legendre functions [1, Chapter 5], [5, Appendix E].

Arguments *k2* – piezoelectric coupling factor, % *v0* - free-surface SAW velocity, m/s *p* - strip period, m *km* - metallization ratio *f* - frequency, MHz *gr_type* - grating type (‘oc’ or ‘sc’) (optional)

Returns *r* - reflection coefficient

Note Both open- and short-circuit strip reflection coefficients are calculated by default for the input argument *gr_type* omitted.

Read_MSC.m

Purpose Input of the MSC data from the text file.

Synopsis [*substrate*,*k2*,*ep*,*v0*,*ve*,*km*,*f0*,*fpi*,*w*,*n*,*fs*,*fe*,*df*]=read_msc(*data_file*)

Description Reads the MSC data from the text file.

Arguments *data_file* – name of the data file

Returns
substrate – substrate material (string variable)
k2 – piezoelectric coupling factor, %
ep – substrate effective permittivity
v0 – free-surface SAW velocity, m/s
ve – effective SAW velocity, m/s
km - metallization ratio
f0 – MSC working frequency, MHz
fpi – MSC synchronous frequency, MHz
w - acoustic aperture, m (obsolete in two-mode approach)
n - number of MSC strips
fs – start frequency of the analysis range, MHz
fe – end frequency of the analysis range, MHz
df - discretization interval, MHz

T_MSC_T.m

Purpose SAW filter analysis comprising input and output SAW transducers coupled through a multistrip coupler

Synopsis [*s11*,*s12*,*s21*,*s22*,*y11*,*y12*,*y21*,*y22*]=T_MSC_T
(*m11_1*,*m12_1*,*m13_1*,*m23_1*,*m33_1*, ...
m11_2,*m12_2*,*m13_2*,*m23_2*,*m33_2*, …
s11,*s12*,*s13*,*s14*, *f*,*v*,*l*)

Description Cascades input IDT → MSC → output IDT to generate *S*- and/or *Y*-parameters of a SAW filter as the two-port lossless reciprocal network.

Algorithm Algorithm is based on the conversion of the mixed scattering matrices of the constituent SAW transducers and MSC wave scattering matrix to the augmented mixed transmission matrices which are cascaded, with the **Y**-matrix (admittance matrix) found from the overall transmission matrix [1, Chapter 6]. **Y**-matrix is converted to **S**-matrix (scattering matrix) of the SAW filter [1, Chapter 1].

Arguments *m11_1,m12_1,m13_1,m23_1,m33_1* – independent mixed scattering matrix elements of the input IDT;
m11_2,m12_2,m13_2,m23_2,m33_2 – independent mixed scattering matrix elements of the output IDT;
s11,s12,s13,s14 – independent wave scattering elements of the multistrip coupler;
v – free-surface SAW velocity, m/s
l – port-to-port separation (IDT-MSC-IDT), m (two-element vector)
f - frequency, MHz (scalar or vector)

Returns sij - SAW filter *S*-parameters (*i, j*=1,2) yij – SAW filter *Y*-parameters (*i, j*=1,2)

See also *MtoT, StoT, Y_to_S.*

V_SAW.m

Purpose Closed-form calculation of a SAW velocity in the grating.

Synopsis *v*=V_SAW (*k2,v0,p,km,f*)
v=V_SAW (*k2,v0,p,km,f,gr_type*)

Description Calculates SAW velocity in a short- and/or open-circuit periodic grating using Morgan's approximation (mass-loading neglected).

Algorithm Floquet analysis is applied to a periodic grating, with the closed-form solution in terms of the Legendre functions [1, Chapter 5], [5, Appendix D].

Arguments *k2* – piezoelectric coupling factor, %
v0 - free-surface SAW velocity, m/s
p - strip period, m
km – metallization ratio
f - frequency, MHz
gr_type - grating type (‘oc’ or ‘sc’) (optional)

Returns *v* - effective SAW velocity, m/s

Note Both open- and short-circuit SAW velocity values are calculated by default, for the input argument *gr_type* omitted.

2.3.2. MATLAB M-Files

TEST_MSC.m

Purpose Test program for MSC analysis in the two-mode assumption using different methods of modeling the symmetric and antisymmetric modes.

Synopsis TEST_MSC

Description Calculates and plots MSC scattering parameters using two-mode approach [1, Chapter 5], [12-14] with the modes modeled by different techniques. The initial data are read from a formatted data files. The output results are presented in terms of the MSC scattering parameters.

Algorithm The MSC scattering matrix is determined in terms of the scattering coefficients of the normal modes [1, Chapter 5], [11, 12]. Four different methods are applied to characterize the symmetric and antisymmetric modes: 1) COM-analysis [9-11]; 2) Morgan’s reflective array model (RAM) [5, Appendix E]; 3) Ingebrigtsen’s field approach [12]; 4) quasi-static approximation ([5, Chapter 5].

Test results See Chapter 3 in this Manual.

See also *MSC.*

TEST_MSF.m

Purpose Test program for analysis of dual-track SAW filters comprising two apodized transducers coupled through a multistrip coupler

Synopsis TEST_MSC

Description Calculates and plots *S*-parameters of a dual-track SAW filter composed of two SAW transducers coupled through a MSC with full or partial acoustic power transfer. The initial data are read from a formatted data file. The output results are presented in terms of the *S*- and/or *Y*-parameters of a SAW filter.

Algorithm 1) Scattering matrices of the input and output SAW transducers and MSC wave scattering matrix are calculated. 2) After converting to the augmented transmission matrices, they are cascaded to the overall SAW filter transmission matrix. 3) *Y*-parameters of a SAW filter are found from the overall mixed transmission matrix. 4) *S*-parameters are calculated by conversion of the **Y**-matrix into **S**-matrix [1, Chapter 1].

Test results See Chapter 3 in this Manual.

2.4. GUI Reference

2.4.1. New Features

The main program *SAWFAT.m* is a graphical user interface (GUI) that allows analyzing bidirectional SAW filters in the quasi-static approximation. The same program is used for analysis both non-reflective and reflective SAW transducers in the full version of the SAWFAT 1.2. Basically, this program corresponds to the older program *TEST_QSF* with much more convenient capabilities to manipulate design data, plot the results, change presentation graphical format, export the results, etc. Besides, in addition to the GUI implementation of the previous functions, the program also includes several new features, particularly

- Interactive input data editing and saving on disk while analyzing a SAW filter.
- Each transducer is specified separately.
- Spacing between SAW transducers specified in terms of port-to-port separation and/or center-to-center separation.

- Acoustoelectric conversion functions of SAW transducers calculated and superimposed on the figures to evaluate contribution of each transducer to the overall SAW filter response.
- Analysis of reflective SAW transducers (optional).
- Smith-chart representation of the scattering parameters.
- Time response calculation using Fast Fourier Transform (FFT).
- Export of the analysis results (*S*- or *Y*-parameters) onto disk in the ASCII text format.
- Generating SAW filter photomask pattern that can be saved onto disk in the AutoCAD DXF-format.

2.4.2. Control Panel Elements

To get started and open the SAWFAT control panel, just type at the MATLAB prompt

sawfat.

The SAW Filter Analysis Tool opens in the default design mode. The initialization file *SAWFAT.ini* contains settings from the last filter analysis. The general view of the control panel and its constitutional elements are shown in Fig. 2.2.

Note: Please be sure that all SAWFAT subdirectories as well as root directory are included in the MATLAB path before getting started.

2.4.2.1. Menu Bar

Four new menu items are added to the standard MATLAB Window Menu Bar: **Read**, **Write**, **Plot**, **Setup**, **About**. The meaning of these menu items is explained as follows.

Reading Data

Use the pop-up menu **Read** which is added to the standard Menu Bar to perform the following file operations:

- Reading SAW filter data from the main data file with an extension *.dat* by default (**Data)** and from the auxiliary likename topological data file with an extension *.top*. The name of the main data can be selected from the directory list.
- Reading tap weights (acoustic sources) for the input and output SAW transducers from the text files with extensions *.t1* and *.t2*, respectively (**Input/Output Taps**).
- Reading (importing) *S*-parameters from a text file, with an extension *.s* by default
- (**S-parameters**).
- Reading (importing) *Y*-parameters from a text file, with an extension *.y* by default
- (**Y-parameters**).
- Quit the program without saving data, results, and settings (**Quit**).
- Reading filter data by selection of the file name from the list of the recent files (up to 6 the most recent files).

Format of the main data file to be prepared for each analyzed SAW filter and format of the topological data file to generate filter photomask will be described in details in Chapter 3 of this Manual.

Note: 1) When browsing or selecting a file from the list of the recent files, one click on the file name loads the SAW filter data from disk. The further analysis is invoked following the instructions in the active window.

2) Double click on the file name loads the data and launches automatically analysis in the default design mode.

Writing Data

The user can save data and analysis results onto disk by using the pop-up menu **Write** with the following items:

- Saving current design data in the formatted text file (**Data**).
- Writing (exporting) *S*-parameters in the ASCII text file (**S-parameters**).
- Writing (exporting) *Y*-parameters in the ASCII text file (**Y-parameters**).
- Writing (exporting) SAW filter topology in the ASCII text file using the internal SAWFAT format (**Photomask, Photomask As**).
- Writing (exporting) SAW filter topology (photomask pattern) in the AutoCAD DXF-format (**AutoCAD, AutoCAD As**).
- Quit the program without saving data, results, and settings (**Quit**).

Note: 1) It is recommended to use internal compact SAWFAT format to store SAW filter topologies on the disk. These files can be converted to the AutoCAD DXF-format using the supplied utility function *top_DXF.m*, at any time (see the program *top_DXF.m* in Chapter 3).

2) Some of the menu items are disabled unless the actual computations are performed to generate the relevant data.

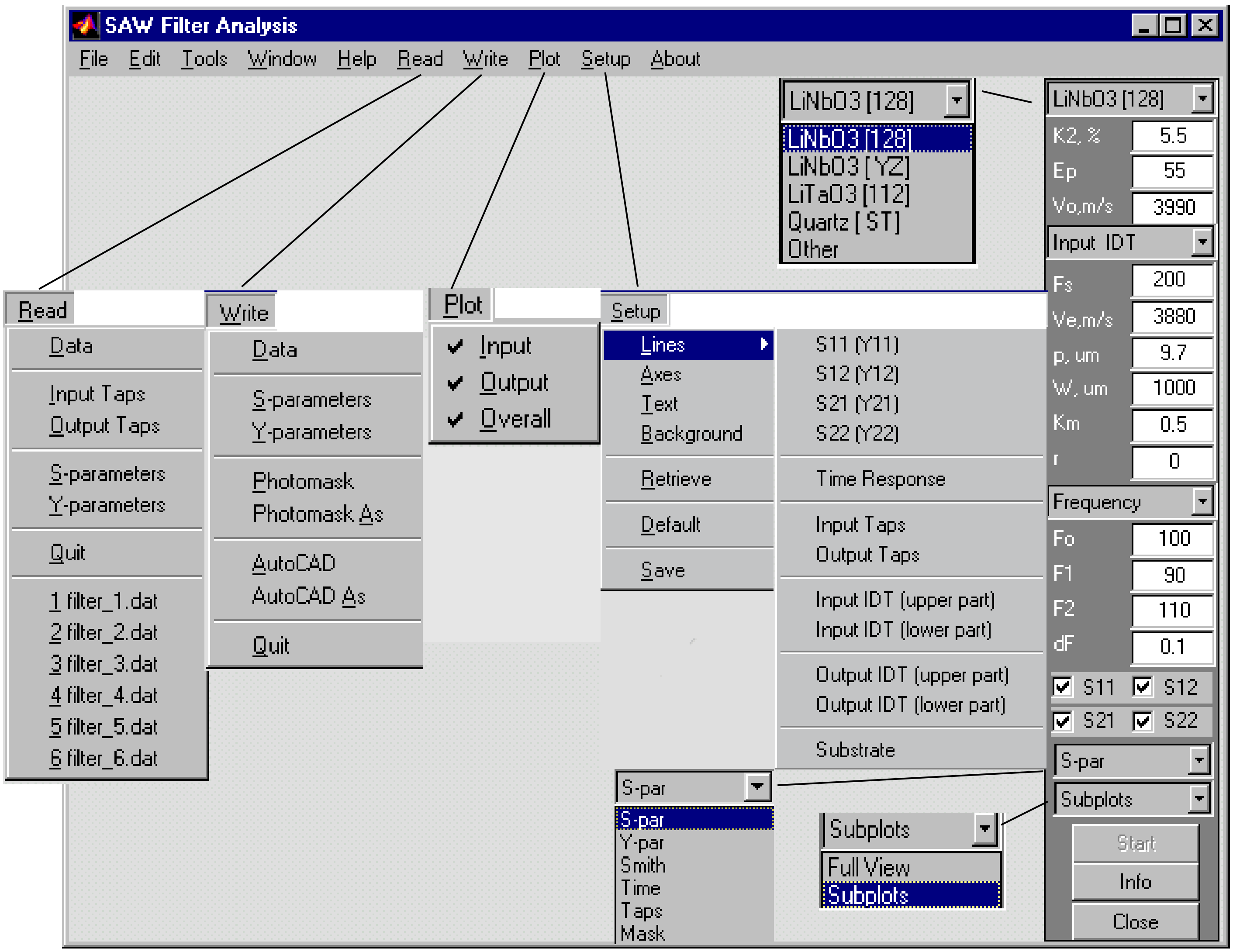


Fig. 2.2. General view and basic elements of SAWFAT control panel

Plotting

This new item has been introduced to plot individual SAW transducer responses (normalized acoustoelectric conversion function) to evaluate its contribution to the overall SAW filter response. These functions are superimposed on the plot of the S_{12}-parameter. You can select any combination of three following options by checking:

Input – plot the response of the input SAW transducer
Output – plot the response of the output SAW transducer
Overall – plot overall SAW filter response.

Note: This option affects plotting S_{12} parameter only by superimposing on the plot the response of the input/output SAW transducer, if checked.

Setup

This menu option is to specify and save a color palette for drawing the plots. The colors can be individually attributed to the following graphical elements:

- Separate lines on the plots of the calculated characteristics, taps, transducers, etc. (**Lines**).
- Axes in figures or subplots (**Axes**).
- Text (titles, legends, etc.) (**Text**).
- Figure background (**Background**).

Colors can be individually assigned to different lines on the same or different plots to visualize *S*- or *Y*-parameters, time response and input/output taps (apodization shape), topological elements (upper/lower parts of each transducer, chip boundaries). The standard color selection dialog (see MATLAB UISETCOLOR command) appears after selecting a target to change the color. This dialog returns a standard or customized color selected by the user.

The initial palette loaded from the color initialization file *SAWFAT.col* can be retrieved at any time by using the menu item **Retrieve**. The palette can be redefined to the default colors (black lines on the white background) by using the option **Default**. The color palette can be saved without quitting the program using the menu item **Save**.

About

This is a text box containing information about an author and the the current SAWFAT version (not shown in Fig. 2.2).

2.4.2.2. Data Control

The control elements are grouped in the right-hand side of the window. They include (from top to bottom):

- lists of the substrate materials (**Substrates)**,
- list to select input or output SAW transducer to be specified (**Input | Output**),

- list to select a group of SAW filter parameters including a frequency span for analysis, spacing between SAW transducers, and termination (source and load resistance) (**Frequency | Spacing | Termination**),
- fields with editable numerical data on substrate, transducer, and SAW filter parameters,
- checked boxes to pick up *S*- or *Y*-parameters to be plotted,
- list of SAW filter characteristics to be analyzed that includes *S*- or *Y*-parameters, Smith chart, time response, tap weights, and photomask pattern (**S-par | Y-par | Smith | Time | Taps|Mask**),
- list of the presentation formats (**Full | Subplots),**
- **Start** button to perform analysis,
- **Info** button to show the helpful information,
- **Close** button to exit the program and close the control panel.

Substrate Data

One can select a substrate material (128*YX* or *YZ* lithium niobate, 112 *YX* lithium tantalate, *ST*-quartz are specified by default) or customize his own substrate by specifying

- piezoelectric coupling factor (K2, %),
- effective permittivity (Ep, dimensionless),
- free-surface SAW velocity (Vo, m/sec),

A customized list of the substrate materials can be specified in the initialization file S*AWFAT.ini*. The format of the initialization file is explained in Fig. 2.3.

Option values contain internal integer parameters governing the control panel default settings.

The list of the substrate materials is an editable database that can be edited in the program or by an external text editor. The substrate material is specified by

- the piezoelectric coupling coefficient,
- substrate permittivity,
- free-surface velocity,

The numeric parameters must start from the >30 position.

A substrate identificator should include the name of the material and the cut (optional). The cut must be embraced by square brackets. The length of the substrate identificator should not exceed 30 characters.

Example: LiNbO3 [128] or

Lithium Niobate [128 YX]

Please note that two separator lines must bracket the substrate list.

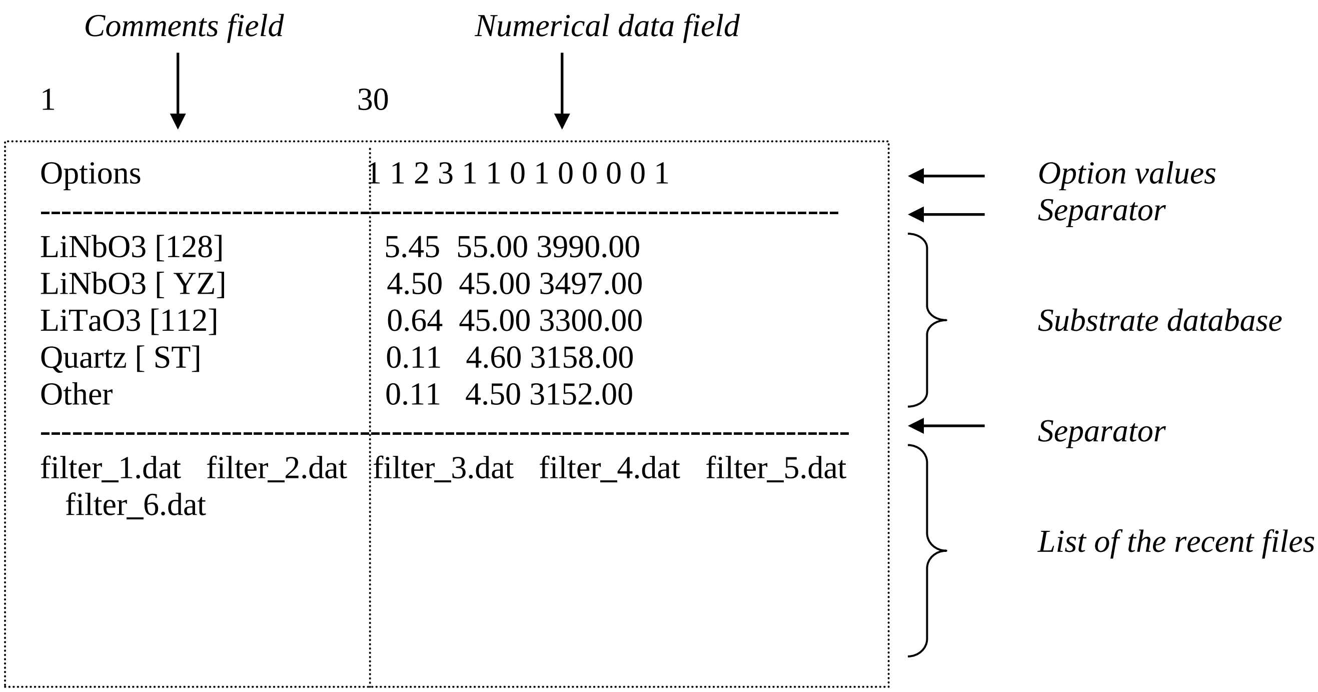


Fig. 2.3. Format of the initialization file *SAWFAT.ini*

Transducer Data

Transducer data are specified separately for the input and output SAW transducer whichever is selected from the list. SAW transducer parameters are entered by using the controls on the right of the figure panel. They include:

Fs - transducer synchronous frequency (MHz)
Ve – effective SAW velocity under SAW transducer (m/s)
p - SAW transducer period (micrometers)
W - acoustic aperture (micrometers)
Km - metallization ratio (duty factor)
r – finger reflection coefficient (zero in the quasi-static approximation)

If you want to enter parameter value other than specified by default, just enter its numerical value in the pertinent parameter field. You can restore the default values of the substrate parameters (K2,Ep,V0,Ve) by entering the "meaningless" zero value in the proper field, at any moment.

Note: The values of the synchronous frequency Fs and transducer period p can be specified independently each other. You can get the default values of the period p calculated from the specified synchronous frequency Fs and the effective SAW velocity Ve by entering a “meaningless” zero value of the period.

SAW Filter Parameters

SAW filter parameters are separated into three groups to be selected from the list. The first group (**Frequency**) includes frequency parameters, particularly:

Fo - SAW filter central frequency (MHz)
F1 - start span frequency (MHz)
F2 - end span frequency (MHz)
dF - discretization interval (MHz)

The second group (**Spacing**) includes spacings between SAW transducers

Lp - port-to-port separation between SAW transducers (micrometers)
Lc - center-to-center separation between SAW transducers (micrometers)

Both spacing parameters are interrelated: if you change one of them, another is automatically recalculated.

The last group (**Termination**) allows changing the electrical termination of a SAW filter by specifying separately the values

Rg – generator (source) resistance (Ohm)
Rl – load resistance (Ohm)

In the program *SAWFAT.m* *S*-parameters are calculated both for the standard 50-Ohm system and for the input/output characteristic impedances Rg and Rl which may differ from the standard values.

Note: Only the set of the standard 50-Ohm S-parameters can be written in the text file using the option **Write → S-parameters**.

2.4.2.3. Visualization Control

You can change between visualization of *S*-parameters, *Y*-parameters, time response, and Smith chart, draw schematically a set of the input/output SAW filter, or generate SAW filter topology (photomask pattern) taps by selecting an item from the parameter list in the popup menu.

Another popup menu just above the **Start** button lets you select a graphical format for presentation of the modeled results. If you choose the **Full** view format, all the multiple calculated characteristics (Sik or Yik) are superimposed on the same full-size figure (Fig. 2.4). If you select **Subplot** format, the characteristics are presented as the separate tiled subplots (Fig. 2.5).

You can control the number of the characteristics to be plotted (up to four) by checking the required parameters in the check boxes (say, you may check S12 only if you do not need S11, S22, and S12, at the moment). Additionally, two acoestoelectric conversion functions can be superimposed by selecting options from the menu **Plot**.

You can change the presentation format and number of the characteristics to be plotted without recalculating the results, just by selection of another form from the popup menu and/or checking the required boxes.

2.4.2.4. Action Buttons

There are three alternative ways to start the SAW filter analysis after reading and/or correcting the data:

1) click on the **Start** button or
2) click a mouse button in the Figure window or
3) press **Enter** or **Space Bar** on the keyboard.

Besides, you can start the analysis immediately by double click on the file name while browsing or selecting from the list of the recent files.

Every time you click the **Start** button the results are recalculated if the substrate or filter data have been changed or read from disk. Otherwise, the calculated data are retrieved from memory.

The **Info** button activates multi-page SAWFAT Help window with a set of numbered buttons assigned to the numbered pages.

The **Close** button closes an active SAWFAT window and exit the program. Program settings and color palette are automatically saved on the disk, with the list of the recent files updated. These updated values are used as default values when the SAWFAT is invoked next time.

2.4.3. Test Example

Test examples of SAW filter analysis using different substrates will be considered in details in Chapter 3 of this Manual for the program TEST_QSF. As the test results and the plots generated by the GUI program SAWFAT are the same they are not considered separately. However, the analysis results for the Example # 1 (data *filter_1.dat*) are given in Figs. 2.4-2.10 for GUI version of the software to complete the present discussion. Please refer to Chapter 3 for more details on the test examples.

In Fig. 2.4 (**S-Par** and **Full** selected), the results of *S12*-parameter calculation are shown, with the acoustoelectric functions M1 and M2 of the input and output SAW transducers superimposed. Please note that in the menu **Plot** the items **Input**, **Output**, and **Overall** are checked and the **S11**, **S21**, **S22** fields on the control panel are unchecked.

Fig. 2.5 (**S-Par** and **Subplot** selected) shows the tiled view of all four *S*-parameters (S11, S12, S21, S22). Here the items **Input**, **Output** of the menu **Plot** are unchecked while all the fields **S11**, **S12**, **S21**, **S22** are checked.

Fig. 2.6 (**Y-Par** and **Full** selected) shows *Y*-parameters superimposed on the same plot. All the fields **Y11**, **Y12**, **Y21**, **Y22** are checked. Please note that the state of the menu **Plot** doesn't affect on plotting *Y*-parameters.

The tiled (subplot) version of the same set of *Y*-parameters is shown in Fig. 2.7 (**Y-Par** and **Subplot** selected).

Smith chart for this example is shown in Fig. 2.8 (**Smith** selected) based on the reflection coefficients **S11** and **S22** calculation. Please note that the **Full | Subplot** menu is disabled here as this menu doesn't affect plotting in this mode.

The calculated time response is shown in Fig. 2. 9 (**Time** selected). The triple transit echo signal which has three times delay with respect to the main signal can be easily identified in this figure.

The taps weights are shown schematically in Fig. 2.10 (**Taps** selected), for this example.

Finally, the photomask pattern is shown in Fig. 2.11 (**Mask** selected). You can view the pattern details using the standard MATLAB option Zoom.

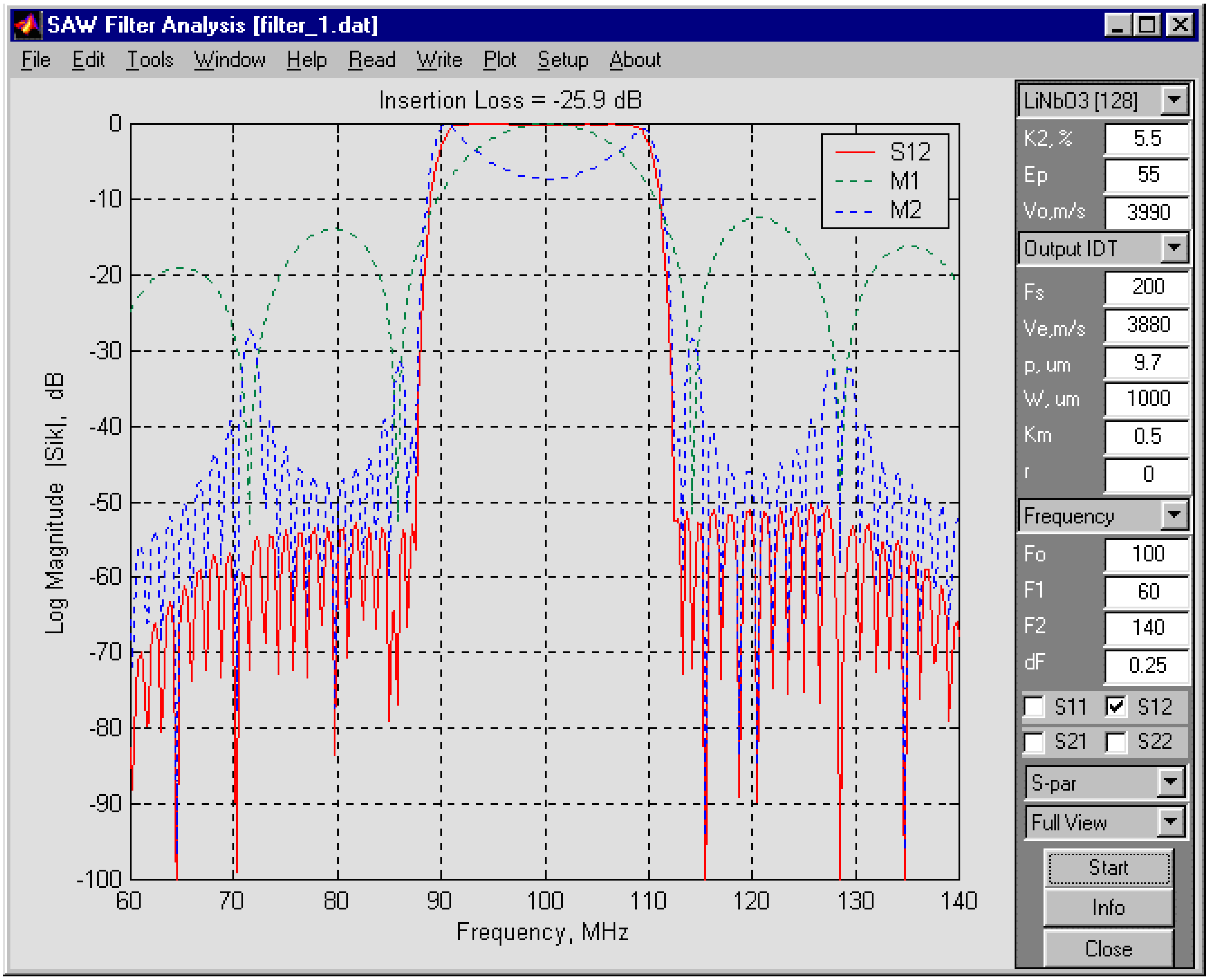


Fig. 2.4. **Full** view of *S*-parameters (Example # 1)

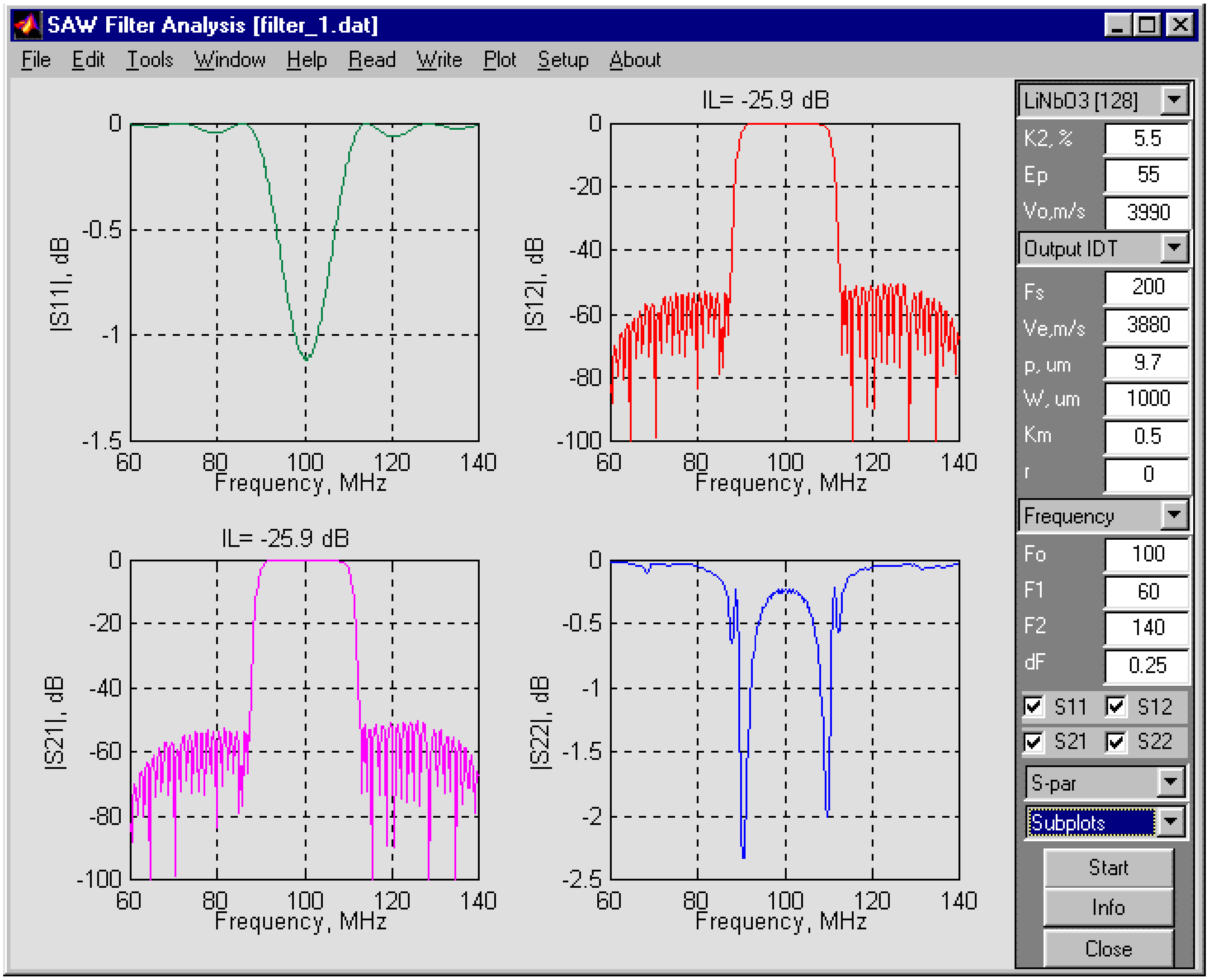


Fig. 2.5. **Subplots** view of *S*-parameters (Example # 1)

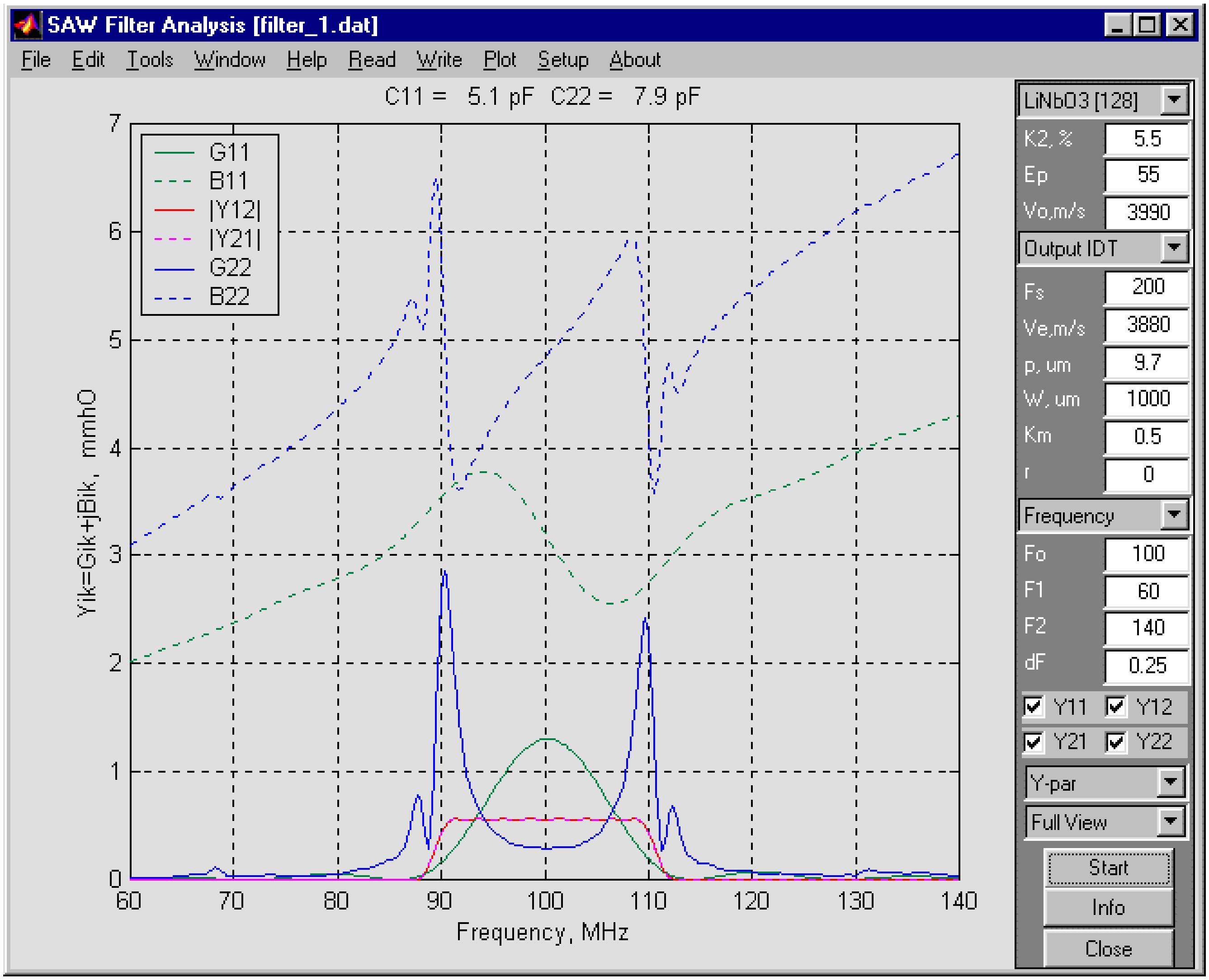


Fig. 2.6. **Full** view of *Y*-parameters (Example # 1)

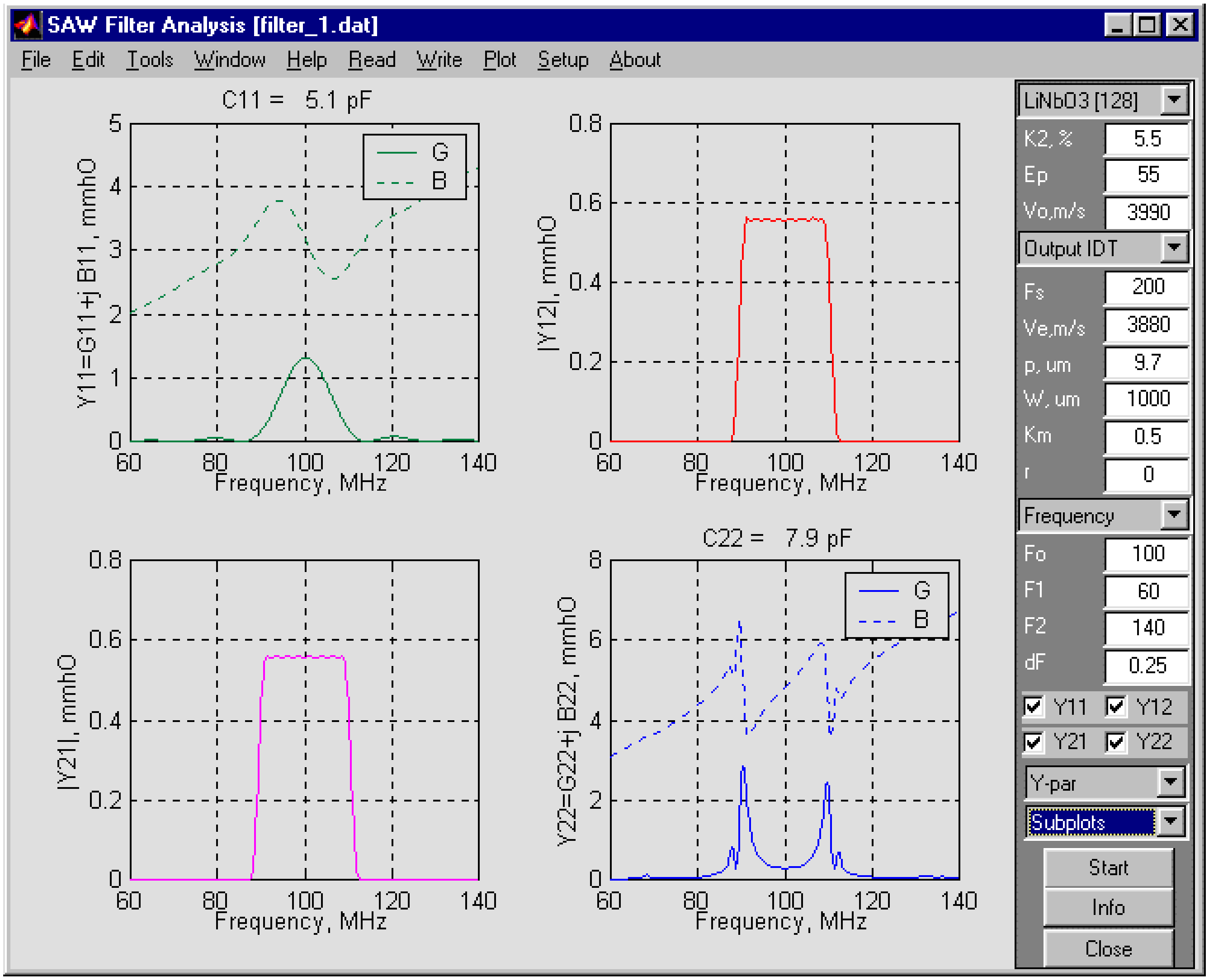


Fig. 2.7. **Subplots** view of *Y*-parameters (Example # 1)

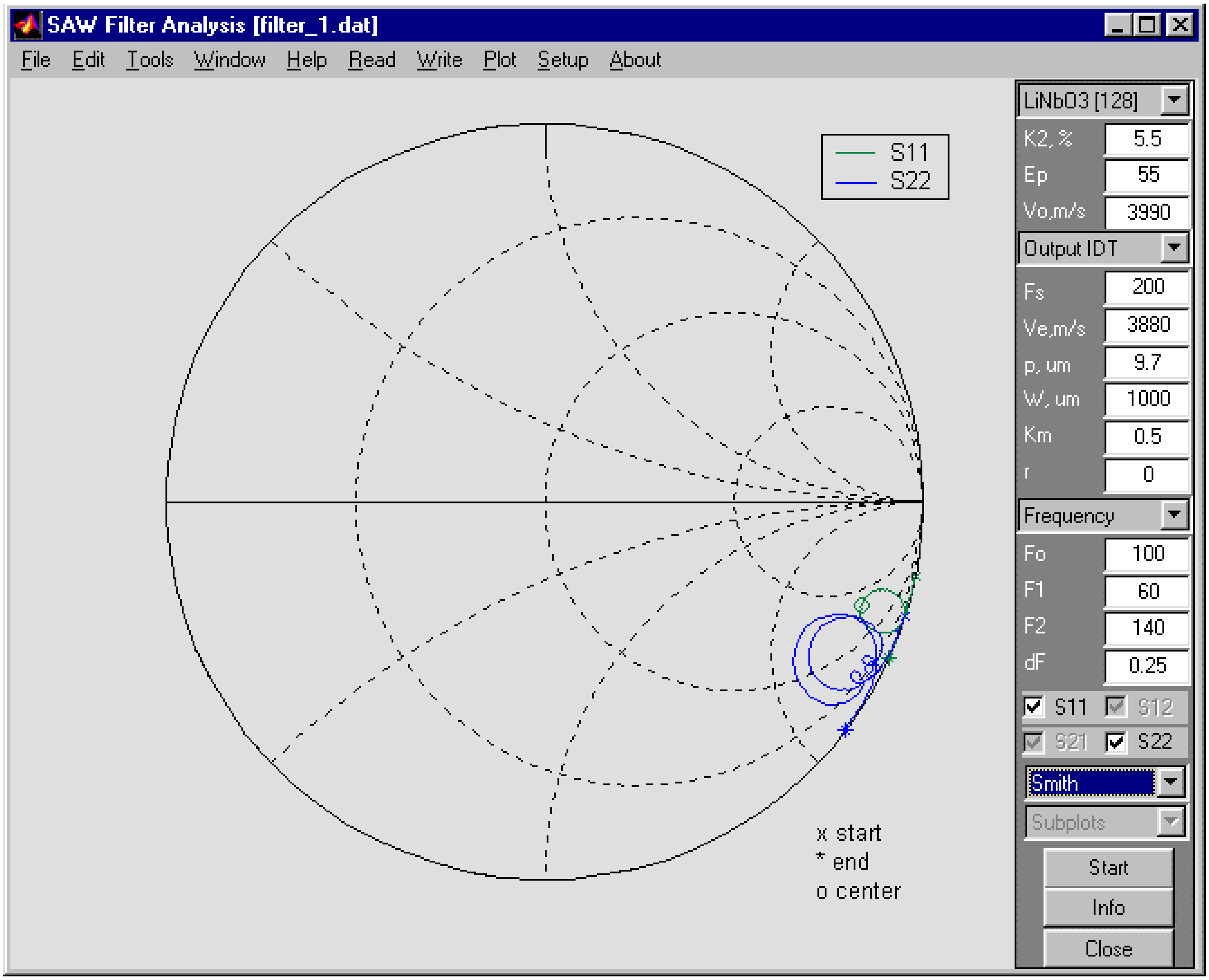
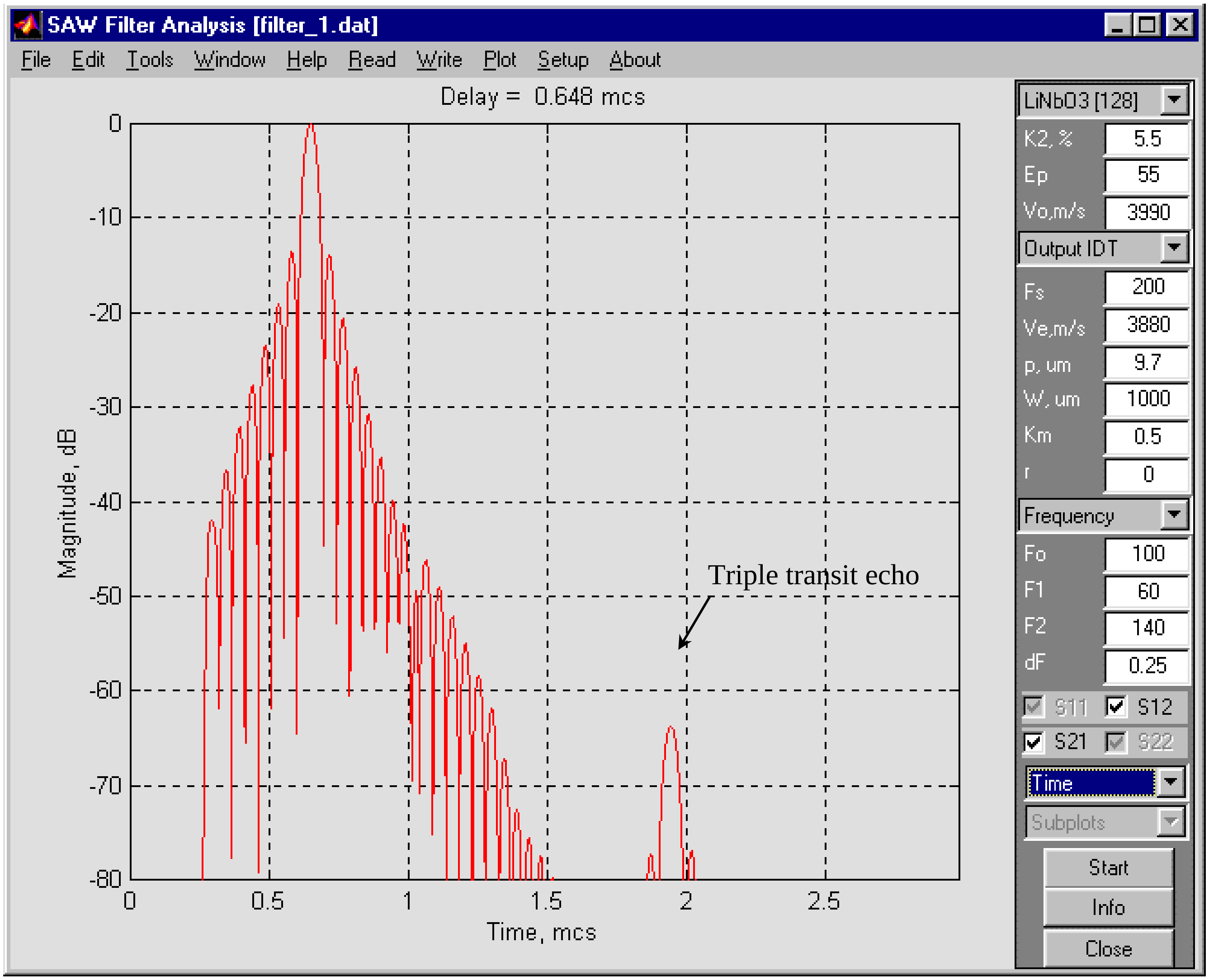


Fig. 2.8. **Smith** chart (Example # 1)



ig. 2.9. SAW filter time response (Example # 1)

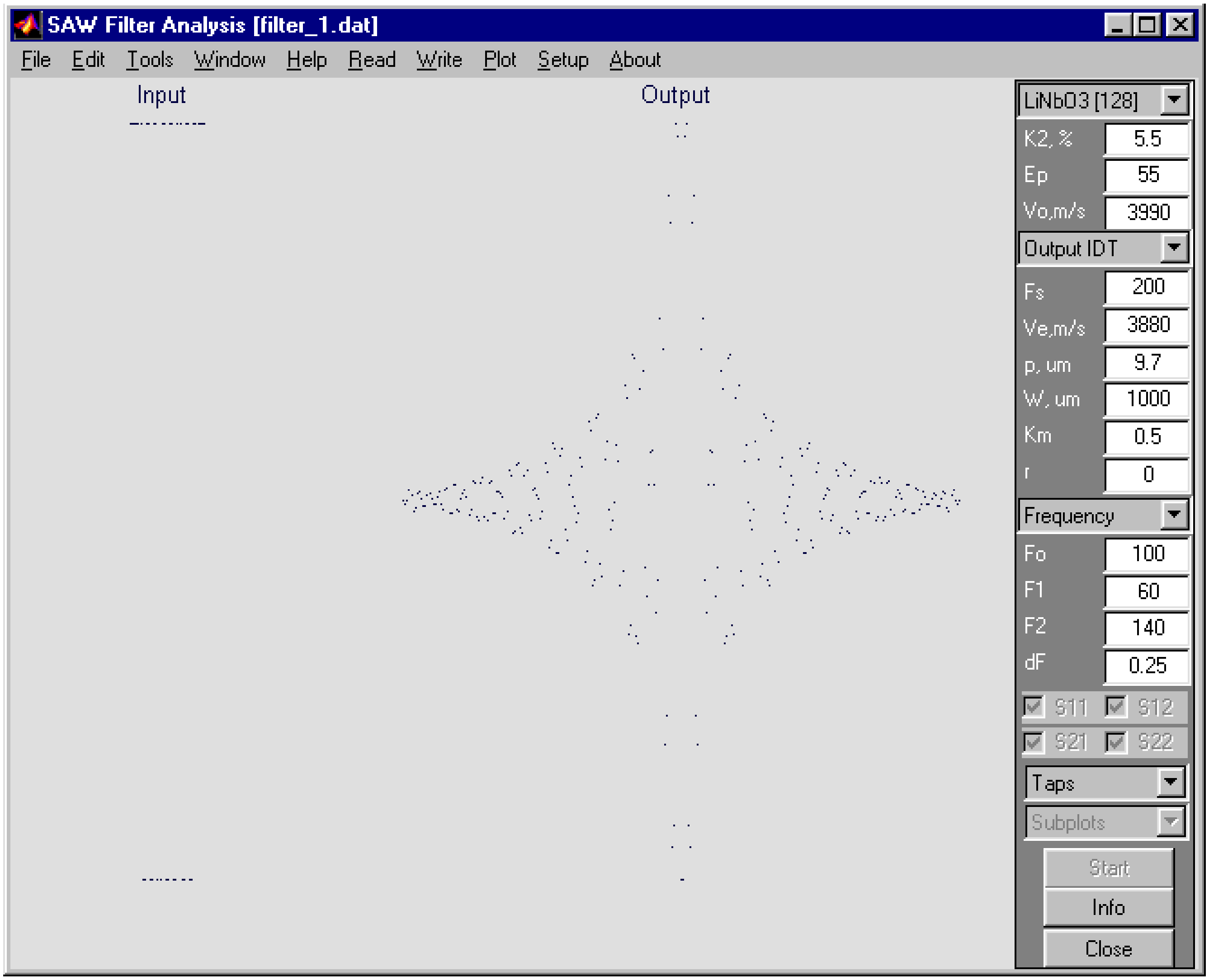


Fig. 2.10. Tap weights of the input unapodized and output apodized SAW transducers (Example # 1)

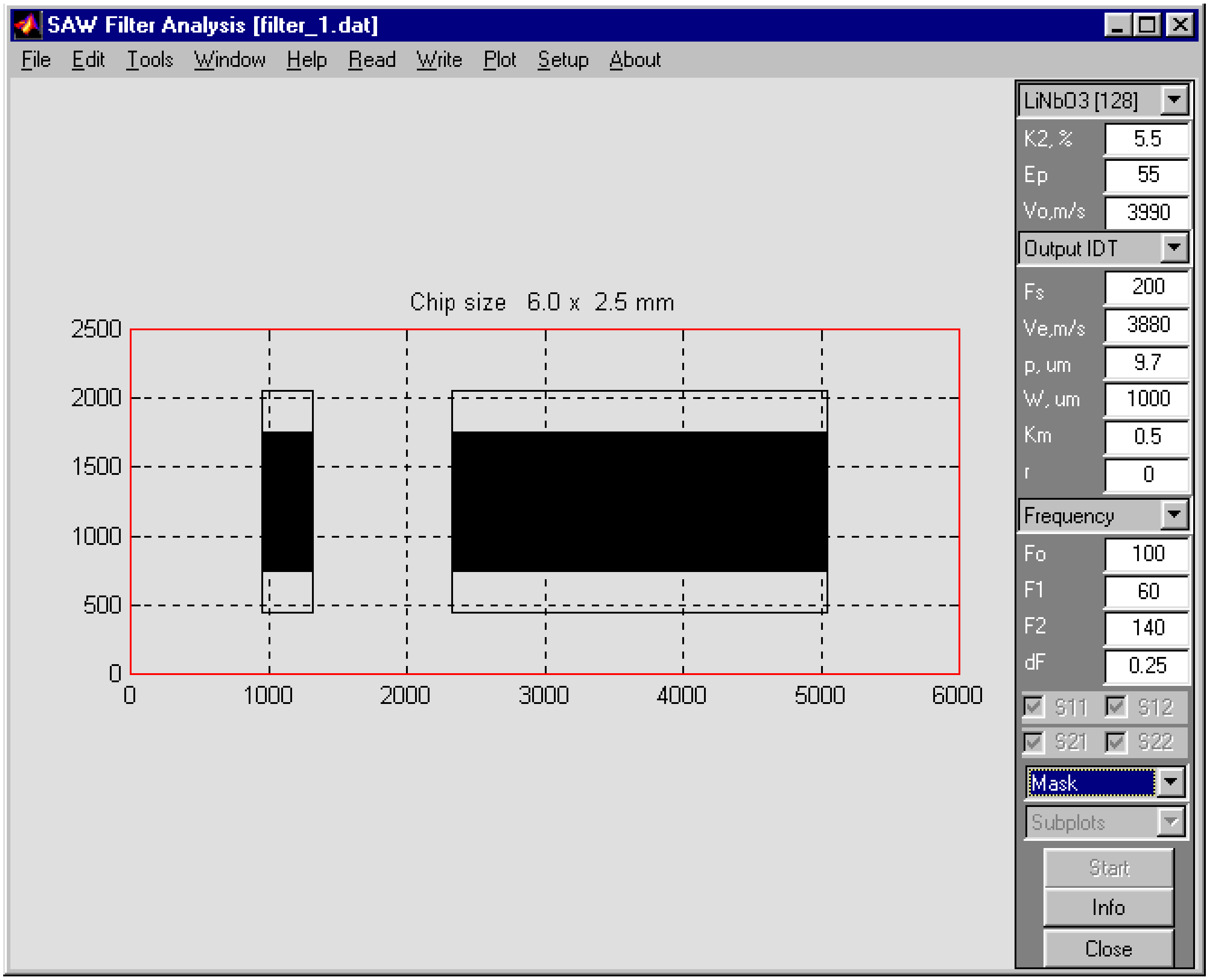


Fig. 2.11. SAW filter topology (photomask pattern) (Example # 1)

3. TUTORIAL EXAMPLES AND TEST RESULTS

3.1. SAW Filter Modeling in the Quasi-Static Approximation

3.1.1. Data Format (Main Data File)

The main test program is called *TEST_QSF*. This M-script file is intended for modeling in-line SAW filters comprising input unapodized (uniform or withdrawal-weighted) and output apodized SAW transducer. The same program can be used for modeling dual-track SAW filters comprising two identical or different apodized SAW transducers coupled through a MSC, supposed for the MSC to be ideal (track-to-track coupling at any frequency is identically equal to 1). For more accurate modeling of the dual-track SAW filters, the test program *TEST_MSF* can be used to account for the MSC transfer function.

The calculations are based on the computation and cascading of the mixed scattering matrices (*P*-matrices) of the constituent SAW transducers using the closed-form technique [1, Chapter 6], [8]. Transducer topologies are specified in terms of the normalized positions of the transversal gaps which separate the active (overlapped) and dummy (non-overlapped) parts of fingers.

The output results are presented in the convenient graphical form and contain the basic *P*-matrix elements (electroacoustic conversion function and transducer admittance) and *S*-parameters of a SAW filter.

Six tutorial examples of modeling SAW filters are included. Data files and tap weights for each example are located in the separate subdirectories **FILTER_1…6** in the directory **EXAMPLES**.

The basic features of these examples are listed in Table 3.1.

Table 3.1

List of Tutorial Examples on SAW Filter Modeling

Data file	**Input transducer**	**Output transducer**	**Cascading**	**Substrate**		**SAW Filter parameters**		
				Material	**Cut**	**Central frequency, MHz**	**Passband width, %**	**Shape factor (-3/-40)**
Filter_1.dat	uniform	apodized (symmetric)	in-line	$LiNbO_3$	128° YX	100	20	1.23
Filter_2.dat	uniform	apodized (V-biased)	in-line	$LiNbO_3$	128° YX	100	20	1.23
Filter_3.dat	withdrawal-weighted	apodized (V-biased)	in-line	$LiNbO_3$	128° YX	90	12	1.31
Filter_4.dat	withdrawal-weighted	apodized (V-biased)	in-line	$LiTaO_3$	112° YX	160	3.3	1.30
Filter_5.dat	withdrawal-weighted	withdrawal-weighted	in-line	Quartz	ST	210.38	0.6	1.98
Filter_6.dat	apodized (symmetric)	apodized (symmetric)	dual-track	$LiNbO_3$	128° YX	60	17	1.2

Data files have an extension *.dat* and contain the structural, material, and frequency data that are necessary for analysis. They must satisfy to the format shown in Table 3.2.

Table 3.2

Example of the Formatted Data File for SAW Filter Modeling

TEST EXAMPLE # 1 (in-line SAW filter, unapodized + apodized, 128YX LiNbO3)

Substrate material		LiNbO3 [128 YX]	
Coupling factor, %	k2 =	5.5	
Substrate permittivity	ep =	55	
Free-surface SAW velocity, m/s	v0 =	3990	
Synchronous frequency, MHz	fpi=	200.000	200.000
Effective SAW velocity, m/s	ve =	3880.000	3880.000
Period, um	p =	9.700	9.700
Acoustic aperture, um	w =	1000.000	1000.000
Metallization ratio	km =	0.500	0.500
Reflection coefficient	re =	0.000	0.000
Port separation, um	L =	1000.000	
Center separation, um	Lo=	2542.300	
Central frequency, MHz	f0 =	100.000	
Frequency range, MHz	Fs =	60	
	Fe =	140	
	dF =	0.25	
Levels, dB	Levels	-1 -30	
Input taps file name:		'Filter_1.t1'	
Output taps file name:		'Filter_1.t2'	
MSC file name		‘Filter_1.MSC’	

The first two lines of the data file may contain arbitrary comments (about SAW filter structure, substrate material, etc.). They should be skipped (left blank) if there are no comments. The next line contains a string variable with the name of a substrate material (LiNbO3, LiTaO3, Quartz, etc.). The cut orientation may be included in the square brackets (optional) (LiNbO3 [128 YX] in this example).

The next 12 lines contain the commented numerical data on the piezoelectric coupling factor (per cent), substrate permittivity (dimensionless), free-surface SAW velocity (m/sec), effective SAW velocity under the transducer (m/sec), metallization ratio or duty factor (dimensionless), central frequency (MHz), synchronous frequency (MHz), acoustic aperture (micrometers), transducer lateral port-to-port separation (micrometers), frequency range for the analysis (MHz) including start and stop span frequency and discretization interval. The last two lines include the string variables with the file names for the input and output tap weights which are to be placed in two different files.

The numerical data at each line should start from the 39 position, with the first 38 characters ignored and optionally used as comments. The file names (last two lines) in the data file (‘*Filter_1.t1*’ and ‘*Filter_1.t2*’ in this example) must be single or double quoted. By default, they have the same file name as the data file, with the extensions *.t1, .t2* for the input and output SAW transducers, respectively. These files contain the tap weights for the input and output SAW transducers, for example:

```
 0.0045207
-0.0079656
 0.0004041
 0.0322613
 0.0155191
-0.0252426
-0.0148819
```

⋮

It is recommended to normalize the tap weights to the unit value where appropriate. However, arbitrary normalization (or units) can be used in general case, at the most convenience as the tap weights are renormalized to the maximum value in the program.

The material constants for different substrates used in the tutorial examples are listed in Table 3.3.

Table 3.3

Material Constants

Material	**Cut**	**Free surface SAW velocity, m/s**	**Electromechanical coupling factor, %**	**Dielectric permittivity**
$LiNbO_3$	128°YX	3990	5.5	55
$LiNbO_3$	YZ	3497	4.5	45
$LiTaO_3$	112°YX	3300	0.64	45
Quartz	ST	3158	0.11	4.6

It is worthy noting that the numerical values in Table 3.3 are estimated and the author is not responsible for their accuracy. However, they provide a sufficient accuracy for the most practical cases of interest. The user may use more accurate values based on the experimental data or the latest published results if necessary.

The tutorial examples in this manual are subjected to some limitations, particularly:

1) The wideband unapodized SAW transducers contain 5 guard fingers at each side to suppress electrostatic end effects.

2) Synchronous frequencies of the input/output SAW transducers are assumed to be the same, for simplicity.

3) All the transducers in the tutorial examples have split (double) fingers (M=4 fingers per wavelength).

In general case, this software can be used for the analysis of SAW transducers containing $M \geq 3$ fingers per wavelength without any modification. In these cases, a synchronous frequency f_π should be set as $f_\pi = M/2\ f_0$, where f_0 is the central frequency. Please note that the tap weights should be properly sampled in this case.

Minor modifications are required if the input and output SAW transducers may have different synchronous frequencies (say, the input transducer contains M=4 and the output one M=3 fingers per period). The user must specify the synchronous frequencies separately at the input and then call the function IDT_QS for the input and output SAW transducers with the appropriate synchronous frequency specified as the input argument.

Following the aforementioned rules and data file format, the users can easily compose their own data files for the analysis of the customized SAW filters.

3.1.2. Compatibility with Earlier Versions (Data Format Conversion)

The format of the data files in SAWFAT 1.2 is different from the previous versions. The program *Convert.m* converts from the old format (SAWFAT 1.1) to the new one (SAWFAT 1.2) to provide upward data compatibility with the previous versions.

Example:

Convert (“filter_1.dat”,”filter1.dat”).

Source data file with the name filter_1.dat (old format) will be converted to the target data file "filter1.dat" (new format).

3.1.3. Topological Data Format

Topological data are used for drawing SAW filter layout on the screen and generating photomask pattern in the AutoCAD DXF format. They must be supplied in the separate data file that is the likename with the major data file and gas an extension *.top*.The topological data file is optional. If it is not found on the MATLAB path, the internal default values are used to generate SAW filter layout (photomask pattern). An example of the topological data file is shown in Table 3.4.

Table 3.4

Example of Formatted Topological Data File

SAWFAT 1.2
TEST EXAMPLE # 1 (in-line SAW filter, unapodized + apodized, 128YX LiNbO3)

Contact pad width (In),	um	dC1	300	300
Contact pad width (Out),	um	dC2	300	300
Transversal gap,	um	dL	1	1
Dummy finger length,	um	Lmin	0	0
Chip size,	um	A x B	8000	3000
Chip center,	um	X0,Y0	4000	1500
Etching tolerance,	um	de	0	0
Rectangular pad fraction,	um	Delta	0	0
Input reference		RefIn		
Input increment		DeltaIn	0	0
Input slope rate,	%	SlopeIn		
Output reference		RefOut		
Output increment		DeltaOut		
Output slope rate,	%	SlopeOu	100	100
Bias type		Biast	N	V
Polarity		Polarity	+	+
Shape of input contact pads		pad_in	rr	
Shape of output contact pads		pad_out	ll	
Pattern type R | P		PATTERN	P	

Again, first two lines can be used for arbitrary comments. Each next line comprises a commentary field (first 38 columns) and a numeric data field. The numeric fields must contain the following data

- contact (bonding) pad width for the input transducer (upper/lower) dC1, μm
- contact (bonding) pad width for the output transducer (upper/lower) dC2, μm
- transversal gap width (input/output) dL, μm
- dummy finger length (input/output) Lmin, μm
- chip size (As x Bs), μm
- chip center coordinates X0, Y0, μm
- etching tolerance (input/output) de, μm
- slanted pad parameter (rectangular pad fraction)delta, μm
- input and output reference finger numbers RefIn, RefOut which are used to impose the bias

- input/output increment (parameter to control the bias rate in the left and right parts of the transducer) SlopeIn, SlopeOut
- input/output slope rate, %
- bias type (string)
- input /output transducer polarity (string)
- shape of the input/output contact pads (string)
- pattern type (different presentation of the transducer topology for making mask)

The first three parameters (dC1, dC2, dL) are obligatory while the others (Lmin, As,Bs,X0,Y0,de,delta) are optional and can be omitted in the topological data file.

The meaning of the topological parameters is explained in Figs 3.1-3.3.

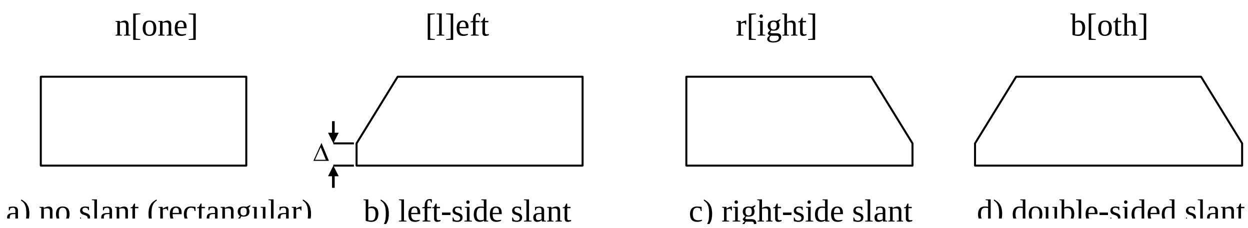


Fig. 3.1. Contact (bonding) pad primitives

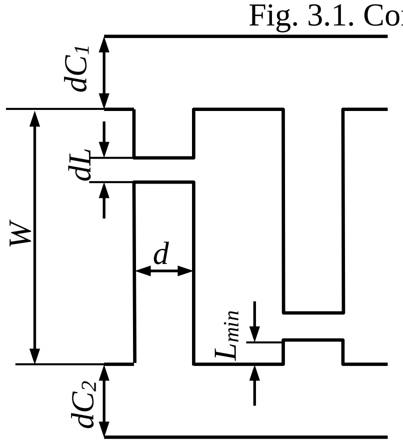


Fig. 3.2. Basic topological elements of a SAW transducer

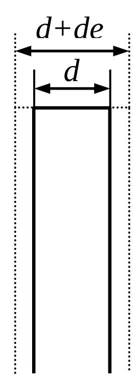


Fig. 3.3. Etching tolerance

The rectangular shape of the upper/lower bonding pads is used in SAWFAT by default. This shape can be controlled in the subroutine *draw_IDT.m* by changing the value of the two-character argument *shape* where the characters can take one of the following values: ‘n’, ‘l’,’r’,’b’ (see fig. 3.1).

The bias type variable takes one of the following values:

‘N’ – no bias

‘V’ – V-type bias

‘S’ – skewed

Polarity controls the polarity of each transducer and takes just character values ‘+’ or ‘-‘.

The last parameter PATTERN is to set the format of the transducer topology for generating photomask. It takes the following character values:

‘R’ – [R}ectangle when each finger is presented by a separate rectangle

‘P’ – [P]olyline when the whole lower and upper finger of the transducer (including dummy foingers) is presented by the separate polyline.

Please note that after converting to AutoCAD DXF format all the contours (rectangles or polylines) are described in terms of the AutoCAD polyline elements. It is obvious that the rectangle format is redundant and results in the longer mask files.

Bas-bars are always presented as the separate contours, for convenience.

Note: 1) If the likename topological data file is not found or doesn't exist the default values are specified to the topological parameters:

```
dC1=[250 250];
dC2=[250 250];
dL=[1 1];
Lmin=[0 0];
As=0;
Bs=0;
X0=0;
Y0=0;
de=[0 0];
delta=[5 5]
Bias='NN  (no bias)
Polarity='++' (positive polarity )
pad_in='nn' (rectangular pads)
pad_out='nn' (rectangular pads)
PATTERN='P' (Polyline)
```

2) If the chip (substrate) size is set to zero values, the actual chip size is calculated that is the actual transducer region length including spacing between transducers plus 500 microns for acoustic absorber at each end (total 1000 microns) and the actual transducer region width including bonding pads plus 500 microns.

3.1.4. Running Test Examples

The test program is invoked by typing *TEST_QSF*. At the prompt "Data file name:", one of the following file names *Filter_1.dat, Filter_2.dat, ... Filter_6.dat* should be entered without quote characters. An extension *.dat* is assumed by default and can be omitted where appropriate. Thefore, both forms are valid in the following example:

Data file name ["filter_1.dat"]: filter_2.dat
Data file name ["filter_1.dat"]: filter_2

The file name by default is shown in the square brackets. This name is accepted after pressing the "Enter" key. For convenience, the name of the last executed data file is written in the file *QSF.INI* to be created and placed in the current directory.

Test results are presented in the graphical form. Two different formats of the plots can be selected while executing the program *TEST_QSF* or *TEST_MSF*:

1) compact form where the results are presented in the form of the small-size figures (subplots) when the internal variable kplot=1;
2) comprehensive form where the results are presented in the form of separate large-scale figures when kplot=2.

For user's convenience, the basic MATLAB workspace variables are listed in Table 3.5.

Table 3.5

Basic MATLAB Workspace Variables for the Programs
SAWFAT, TEST_QSF, TEST_MSF

Variable name	Type	Program	Meaning
F	Real	*SAWFAT, TEST_QSF, TEST_MSF*	Frequency, MHz
M11_1	Real	*SAWFAT, TEST_QSF, TEST_MSF*	P-matrix elements for the input SAW transducer
M12_1	Complex		
M13_1	Complex		
M23_1	Complex		
M33_1	Complex		
M11_2	Real	*SAWFAT, TEST_QSF, TEST_MSF*	P-matrix elements for the output SAW transducer
M12_2	Complex		
M13_2	Complex		
M23_2	Complex		
M33_2	Complex		
Y_1	Complex	*SAWFAT, TEST_QSF, TEST_MSF*	Input transducer admittance including radiation conductance and susceptance with static capacitance
Y_2	Complex	*SAWFAT, TEST_QSF, TEST_MSF*	Output transducer admittance including radiation conductance and susceptance with static capacitance
C1	Real	*SAWFAT, TEST_QSF, TEST_MSF*	Input transducer static capacitance
C2	Real	*SAWFAT, TEST_QSF, TEST_MSF*	Output transducer static capacitance
s11	Complex	*SAWFAT*	SAW filter *S*-parameters (arbitrary system)
s12	Complex		
s21	Complex		
s22	Complex		
S11	Complex	*SAWFAT, TEST_QSF, TEST_MSF*	SAW filter *S*-parameters (50-Ohm)
S12	Complex		
S21	Complex		
S22	Complex		
Y11	Complex	*SAWFAT, TEST_QSF, TEST_MSF*	SAW filter *Y*-parameters
Y12	Complex		
Y21	Complex		
Y22	Complex		
S11_MSC	Complex	*TEST_MSF*	MSC *S*-parameters
S12_MSC	Complex		
S13_MSC	Complex		
S14_MSC	Complex		

The sample test results (*S*-parameters) are included in the files *Filter_1...6.s*. The user can plot the attached *S*-parameters using *plotdata* command (see **help** for *plotdata.m*) to compare with those in the MatLab workspace (arrays f, S11, S12, S21, S22) where f is the frequency array, Sij are *S*-parameters.

3.1.5. Example # 1 (Filter_1)

The SAW filter comprises a uniform unapodized SAW transducer and apodized centro-symmetric SAW transducer (Fig. 3.4a). The substrate material is 128° YX $LiNbO_3$. Number of active fingers in the input (unapodized) and output (apodized) SAW transducers are N_1=28 and N_2=280, respectively. The transducers have the central frequency f_0=100 MHz and synchronous frequency f_π=2 f_0=200 MHz that corresponds to the split (double) finger transducer structures. Acoustic aperture is W=1 mm, port-to-port separation L=1 mm.

The test results (compact form) for this example are given in Fig. 3.5. They include basic *P*-matrix elements for the input and output SAW transducers (Fig. 3.5a), and *S*-parameters of the SAW filter as a two-port (Fig. 3.5b). The static capacitance values for the input and output SAW transducers (Cin=5.1 pF and Cout=7.9 pF) are also printed in Fig. 3.5a. For convenience, the S_{12} and S_{21} plots in Fig. 3.5b are normalized, with the insertion loss values printed above the plots.

As the comprehensive form contains virtually the same information, the enlarged space-consuming figures are not shown for this and other examples.

3.1.6. Example # 2 (Filter_2)

This example differs from Example # 1 by apodization pattern of the output SAW transducer as the *V*-shaped bias has been imposed on the centro-symmetric prototype taps to minimize “hot” finger area and hence the electrical coupling to the case ground (Fig. 3.4b). The modeled results for this example are given in Fig. 3.6. As can be seen, the imposed V-shaped bias slightly changes the transducer characteristics. In particular, amplitude of the spikes in the radiation conductance of the output apodized SAW transducer is reduced (compare Figs. 3.5a and 3.6a).

3.1.7. Example # 3 (Filter_3)

The SAW filter in this example has the fractional passband width of 12 %. The central frequency is f_0=90 MHz. The substrate material is 128° YX $LiNbO_3$. Number of active fingers in the input withdrawal-weighted and output apodized SAW transducers are N_1=86 and N_2=380, respectively. Acoustic aperture is W=2 mm, port-to-port separation L=0.5 mm. The calculated results are shown in Fig. 3.7.

3.1.8. Example # 4 (Filter_4)

This SAW filter has the narrower fractional passband width of 3.3 % with the central frequency f_0=160 MHz. The substrate material is 112° YX $LiTaO_3$. Number of active fingers in the input withdrawal-weighted and output apodized SAW transducers are N_1=142 and N_2=800, respectively. Acoustic aperture is W=2 mm, port-to-port separation L=0.5 mm. The results are shown in Fig. 3.8.

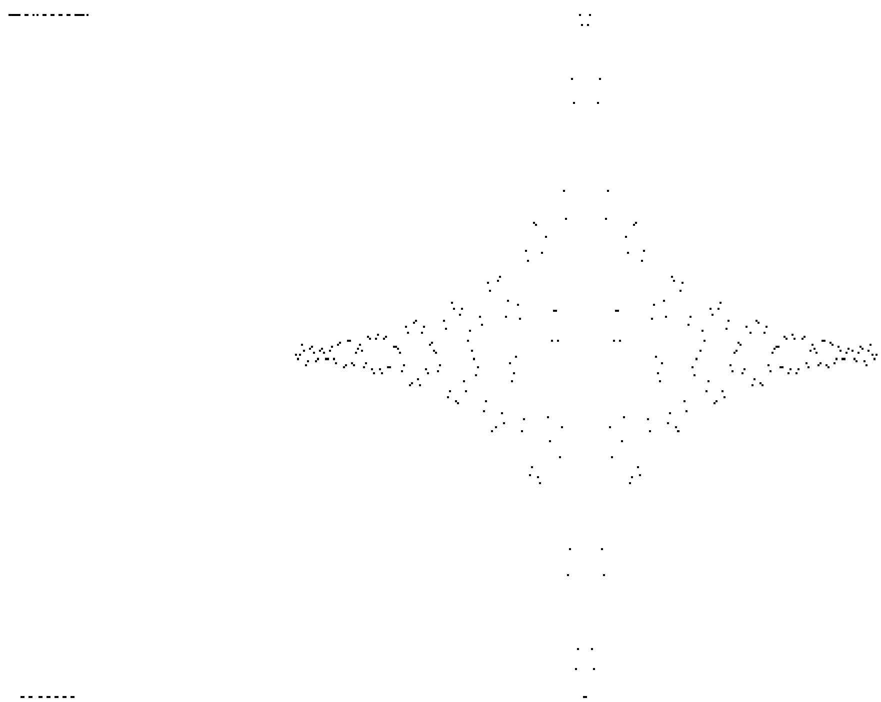
3.1.9. Example # 5 (Filter_5)

This example is to illustrate modeling of a narrowband SAW filter using *ST*-quartz substrate. The central frequency is f_0=210.38 MHz, the fractional passband width is about 0.6 %. Both SAW transducers are withdrawal-weighted with the number of fingers N_1=1262 and N_2=1178, respectively. Acoustic aperture is W=1.5 mm, port-to-port separation L=0.6 mm. Metallization ratio is η=0.4. The modeled results are given in Fig. 3.9.

3.1.10. Example # 6 (Filter_6)

The last example illustrates dual-track SAW filter modeling. The SAW filter comprises two identical apodized split-finger SAW transducers coupled through a MSC. The central frequency is f_0=60 MHz. The number of fingers is N_1=N_2=160. Substrate material is 128° YX $LiNbO_3$. Acoustic aperture is W=2.5 mm, port-to-port separation L=1 mm. The schematical layout and topology (apodization pattern) of the dual-track SAW filter is shown in Fig. 3.10.

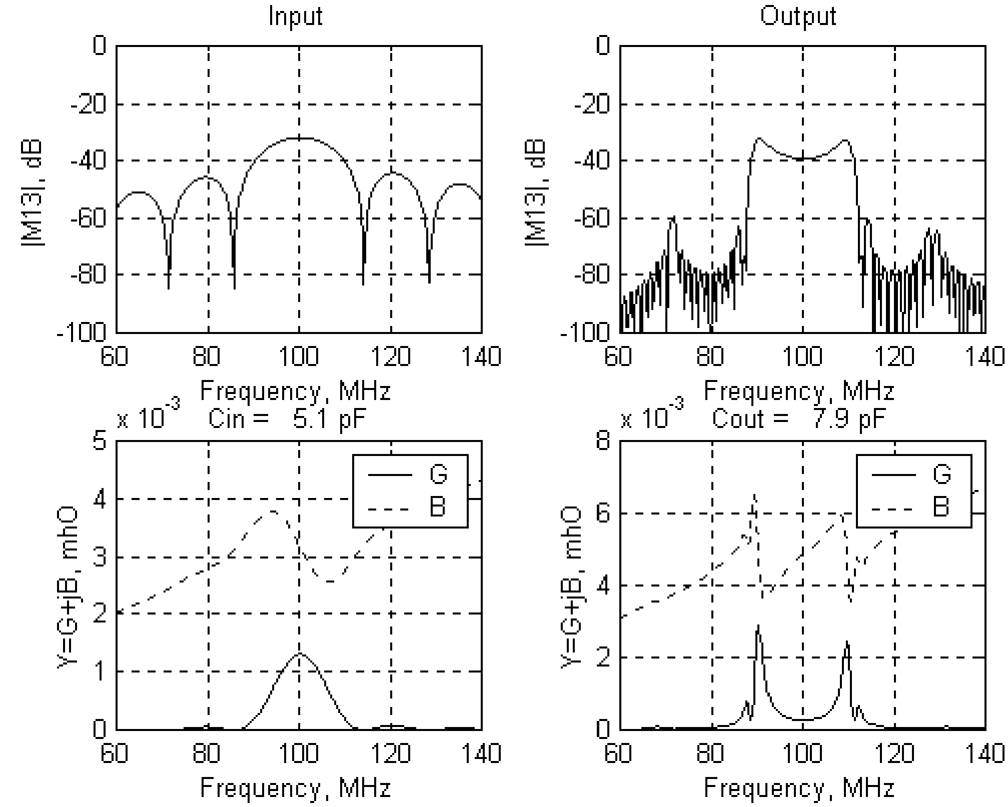
The modeling results are shown in Figs. 3.11 and 3.12, respectively. In Fig. 3.11, the SAW filter characteristics were calculated using the program *TEST_QSF* where the MSC transfer function is implicitly supposed to be identically equal to one (ideal MSC). The plots in Fig. 3.12 were calculated using the program *TEST_MSF* which takes into account the simulated MSC transfer function. COM-model was used in this particular example. MSC model is selected by the value of the last parameter in the subroutine *MSC* which may take one of the following string values: "QS", "RAM", "COM", "FLD". MSC track-to-track coupling function is also shown by the dotted line in Fig. 3.12b. MSC parameters are the following: the synchronous frequency f_0==1.4f_π, number of the strips N_{MSC}=90, metallization ratio η=0.5. Port-to-port separation of the MSC and SAW transducer is L=1 mm at each side. As can be seen in Fig. 3.12, the MSC contribution to the overall SAW filter response is small resulting mostly in the slight roll-off of the magnitude response in the passband.



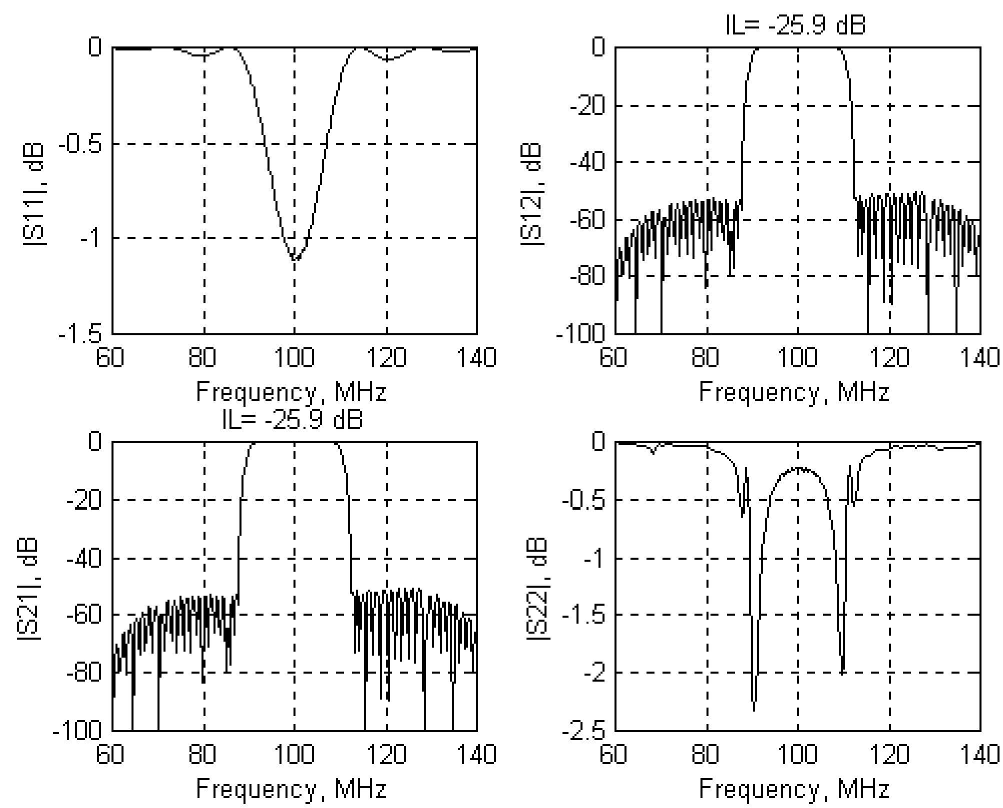
a) centro-symmetric apodized SAW transducer (Example # 1)

b) V-biased apodized SAW transducer (Example # 2)

Fig. 3.4. Input/output SAW transducer apodization patterns

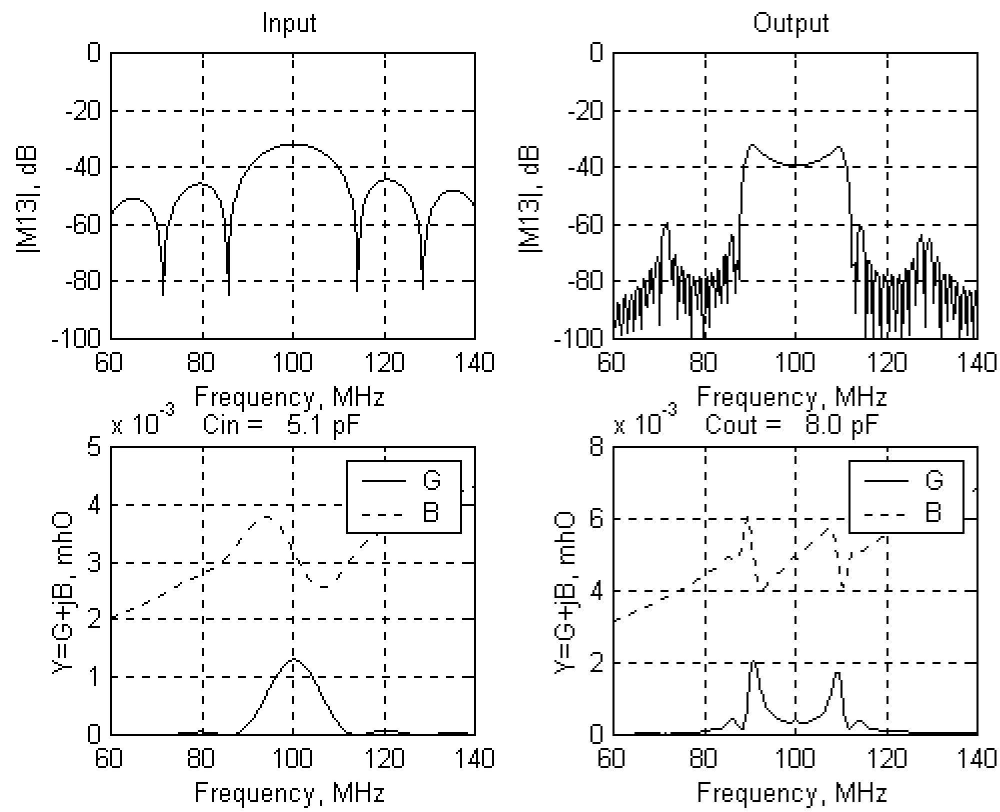


a) P-matrix elements

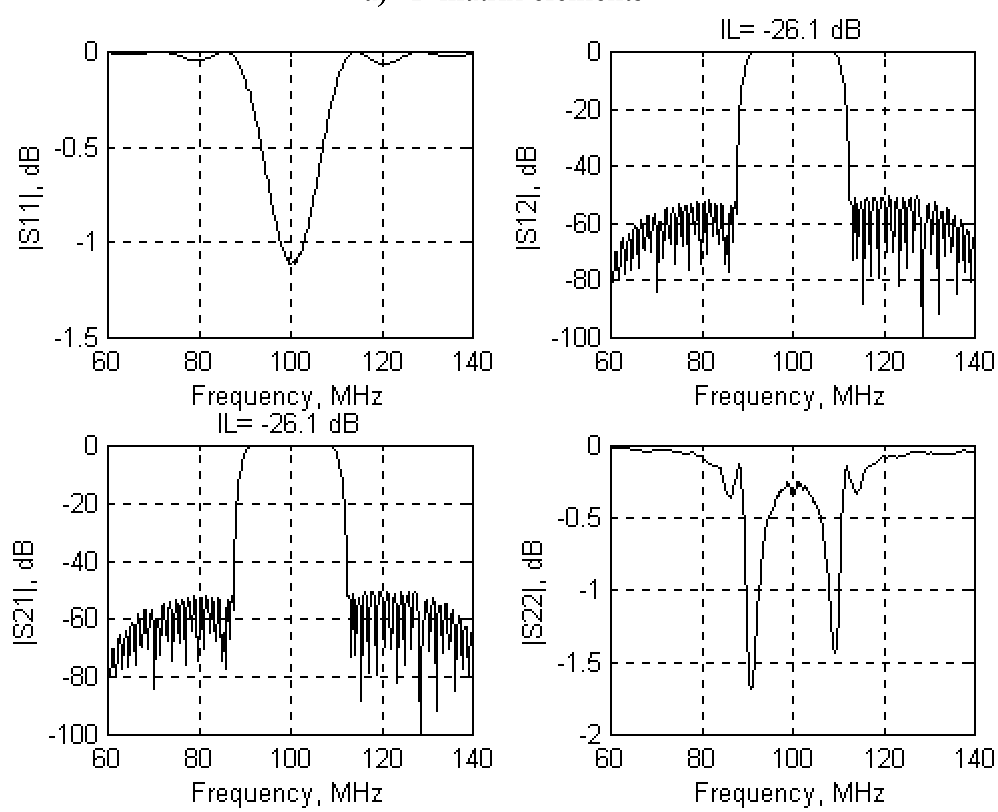


b) S-parameters

Fig. 3.5. Example # 1: calculation results (128° YX $LiNbO_3$)

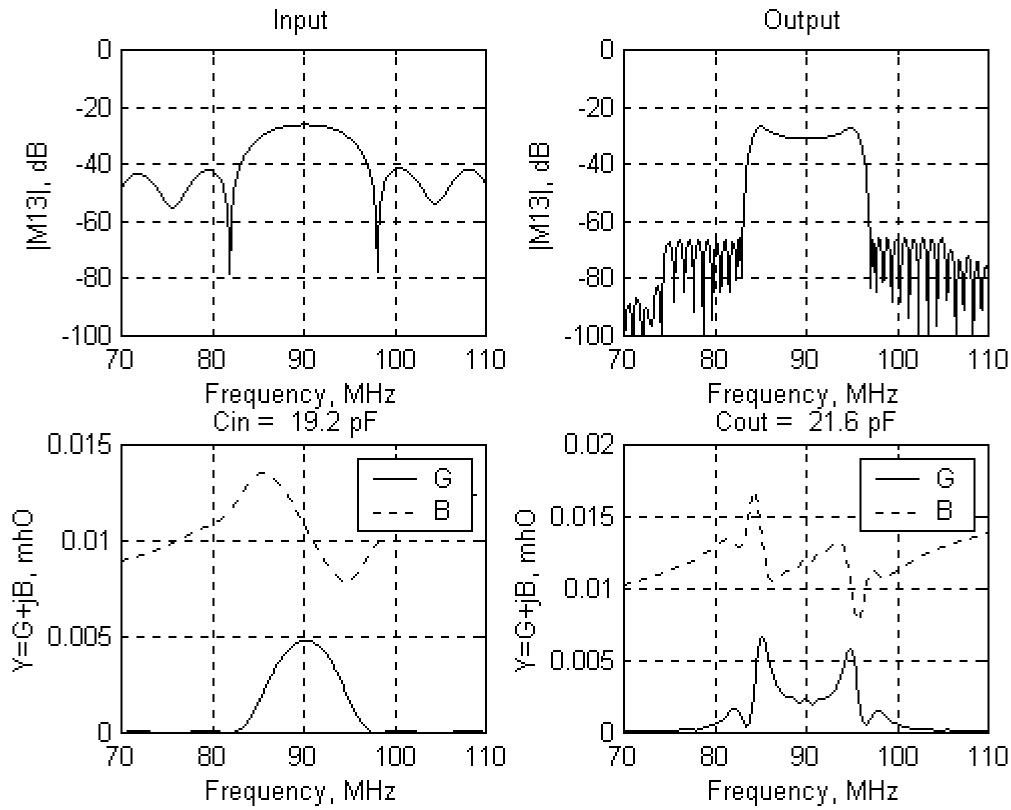


a) P-matrix elements

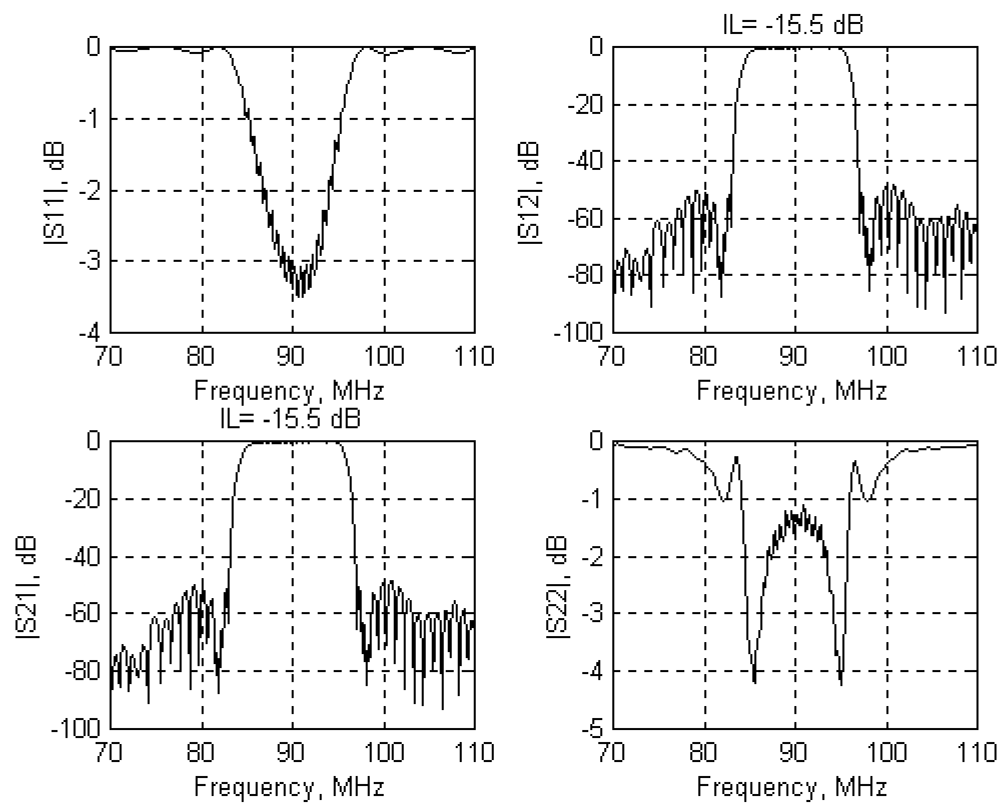


b) S-parameters

Fig. 3.6. Example # 2: calculation results (128° YX $LiNbO_3$)

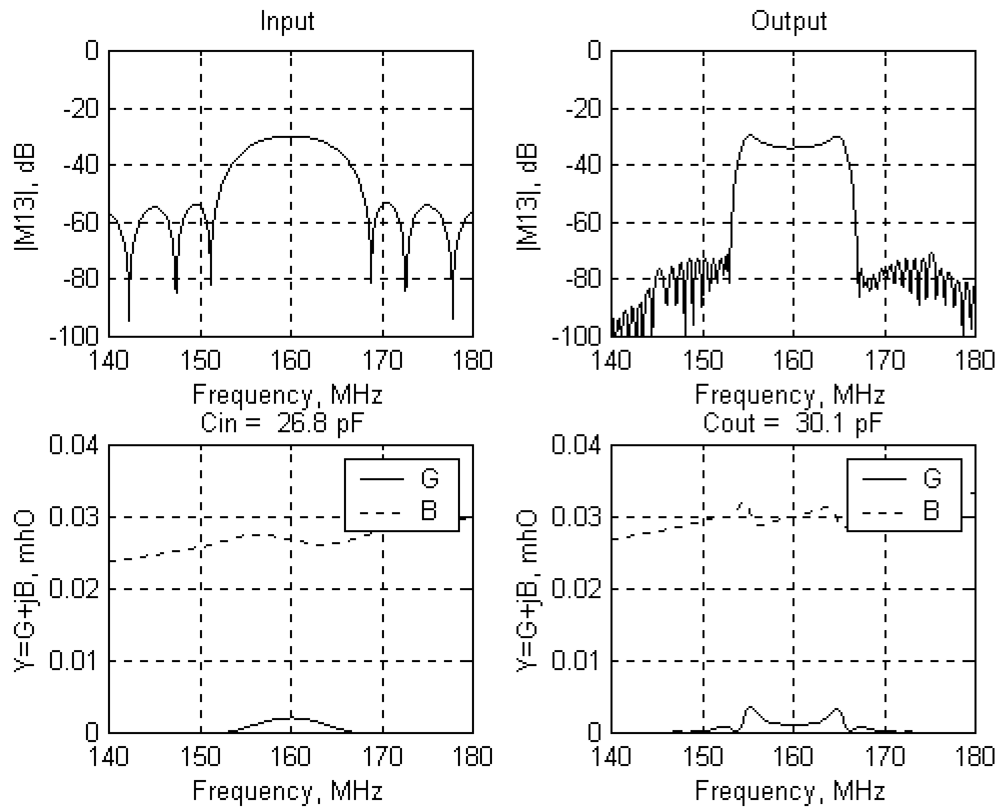


a) P-matrix elements

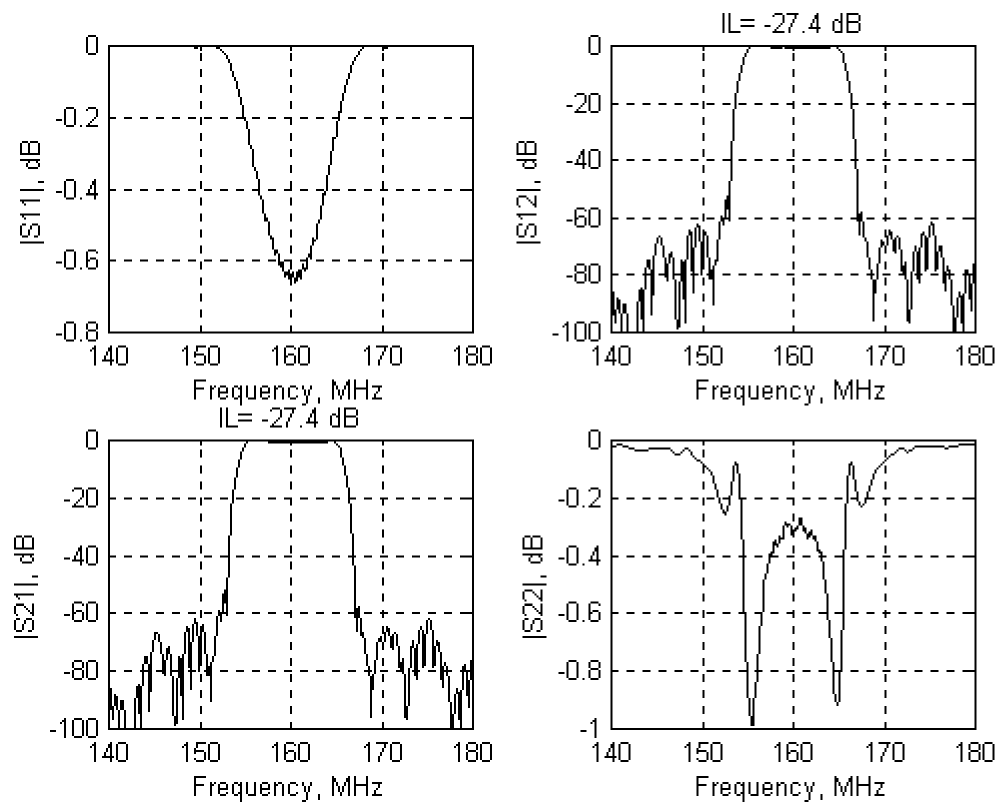


b) S-parameters

Fig. 3.7. Example # 3: calculation results (128° YX $LiNbO_3$)

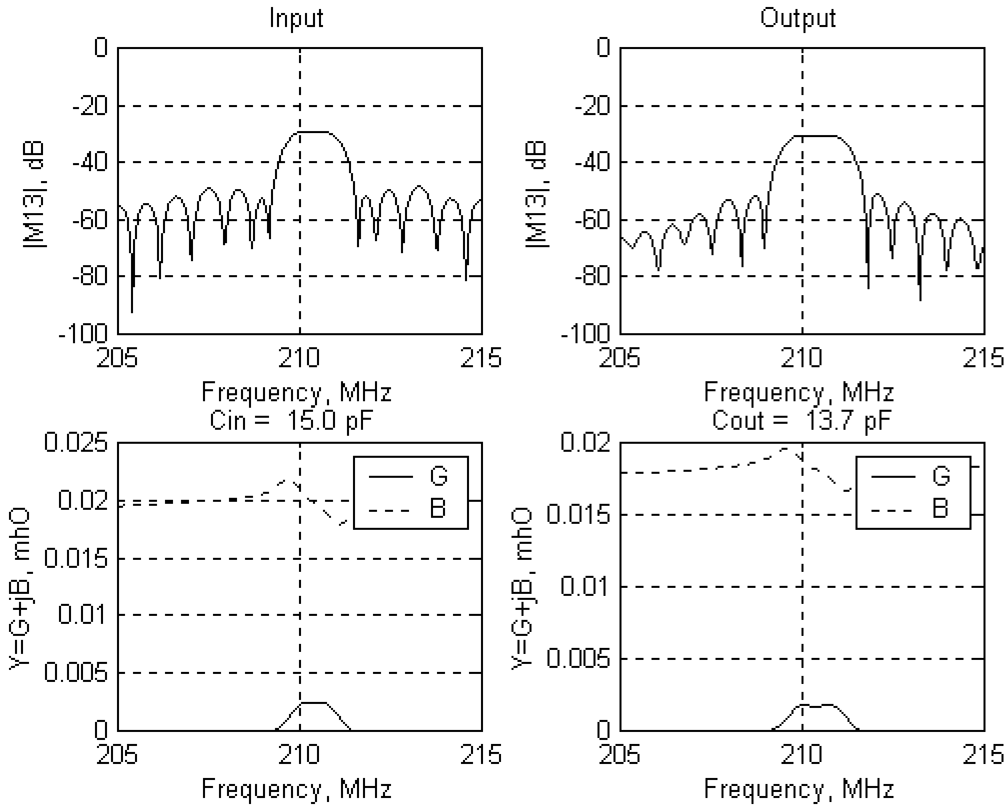


a) P-matrix elements

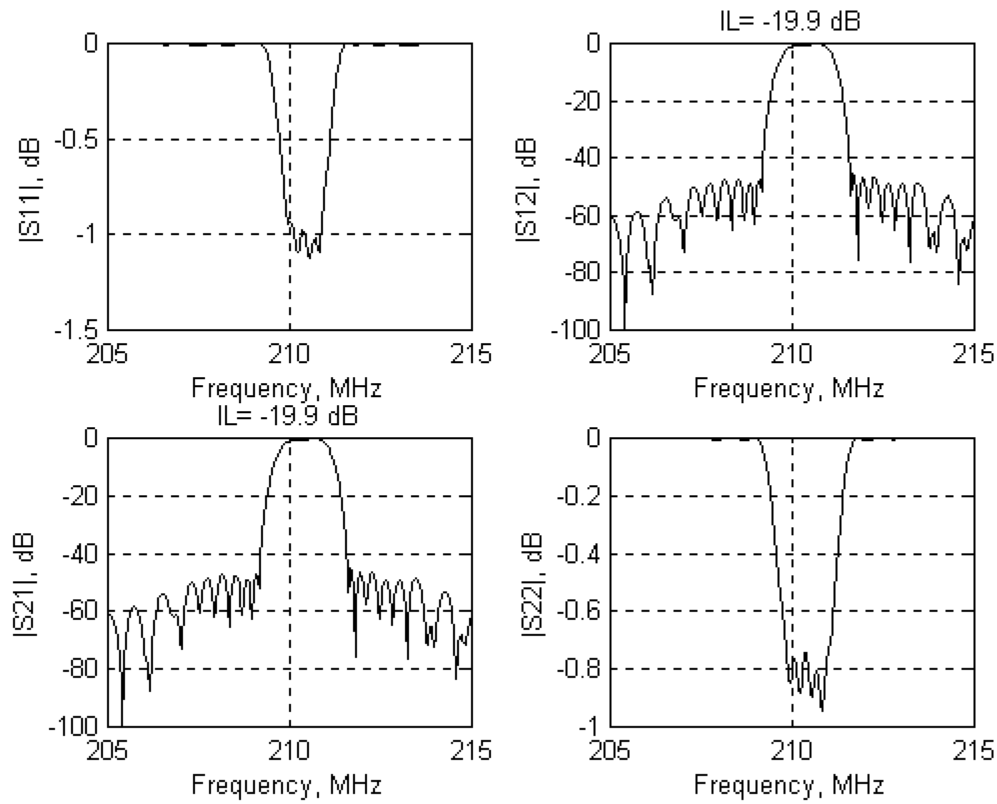


b) S-parameters

Fig. 3.8. Example # 4: calculation results (112° YX $LiTaO_3$)



a) P-matrix elements



b) S-parameters

Fig. 3.9. Example # 5: calculation results (*ST*-quartz)

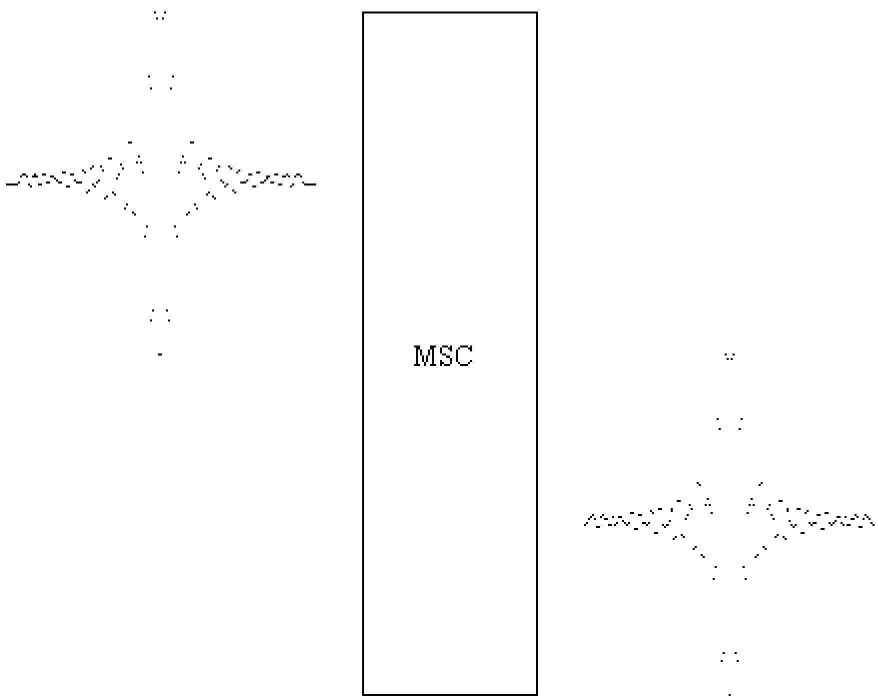
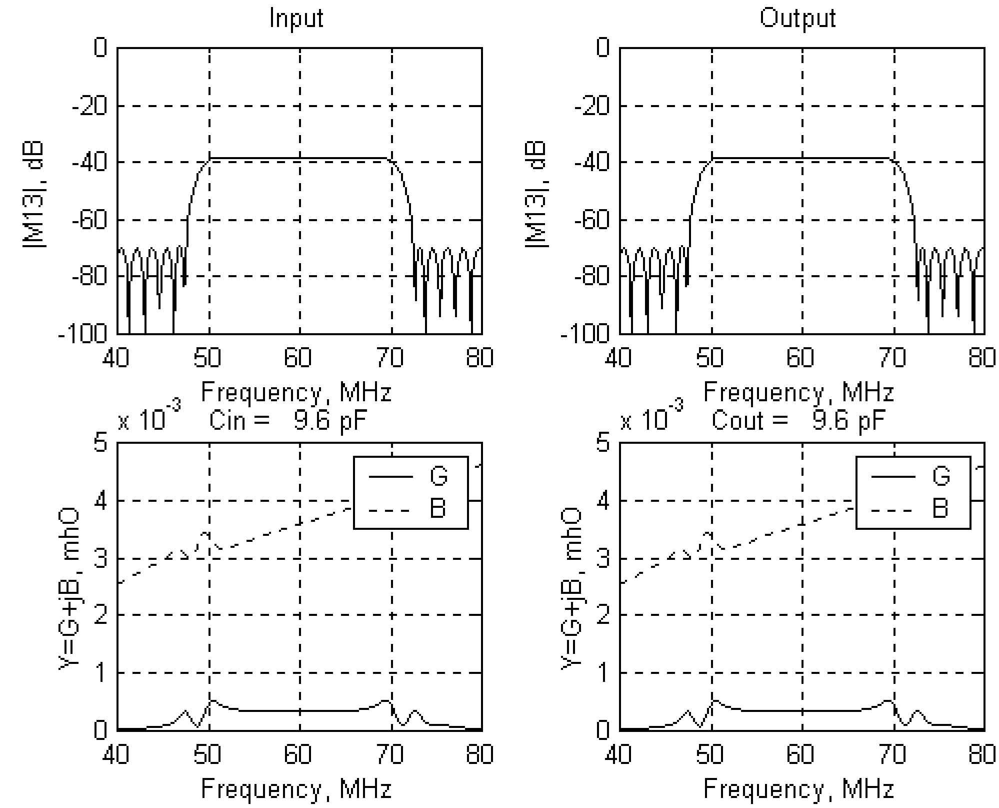
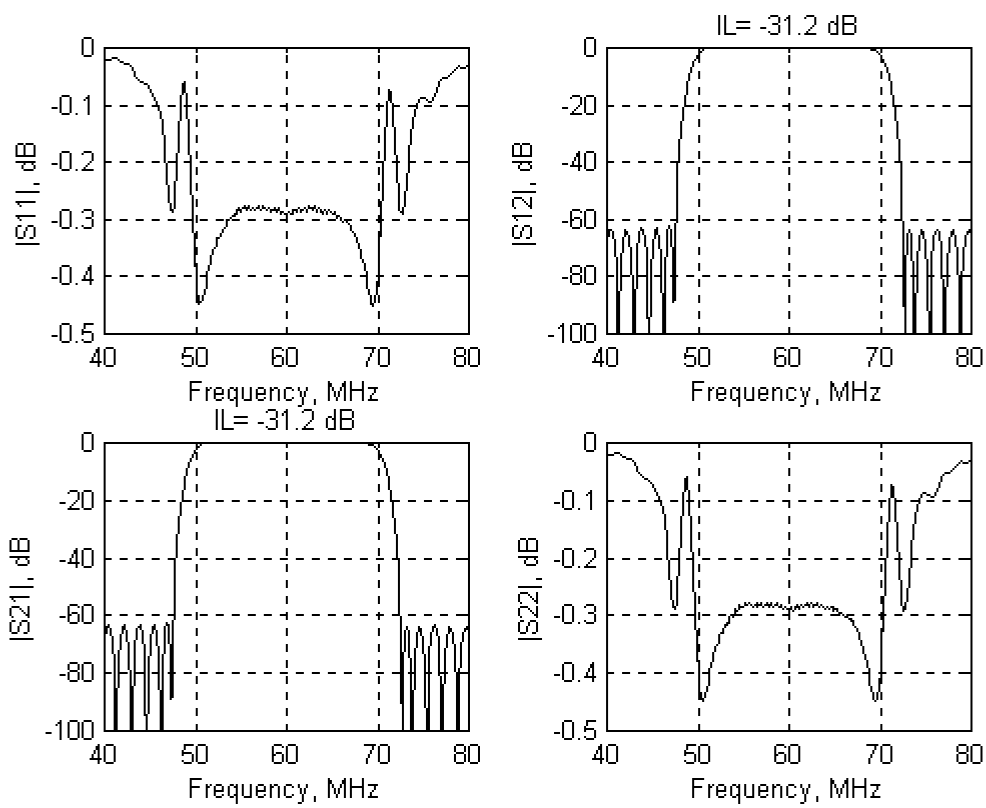


Fig. 3.10. Schematical layout of the dual-track SAW filter (Example # 6)

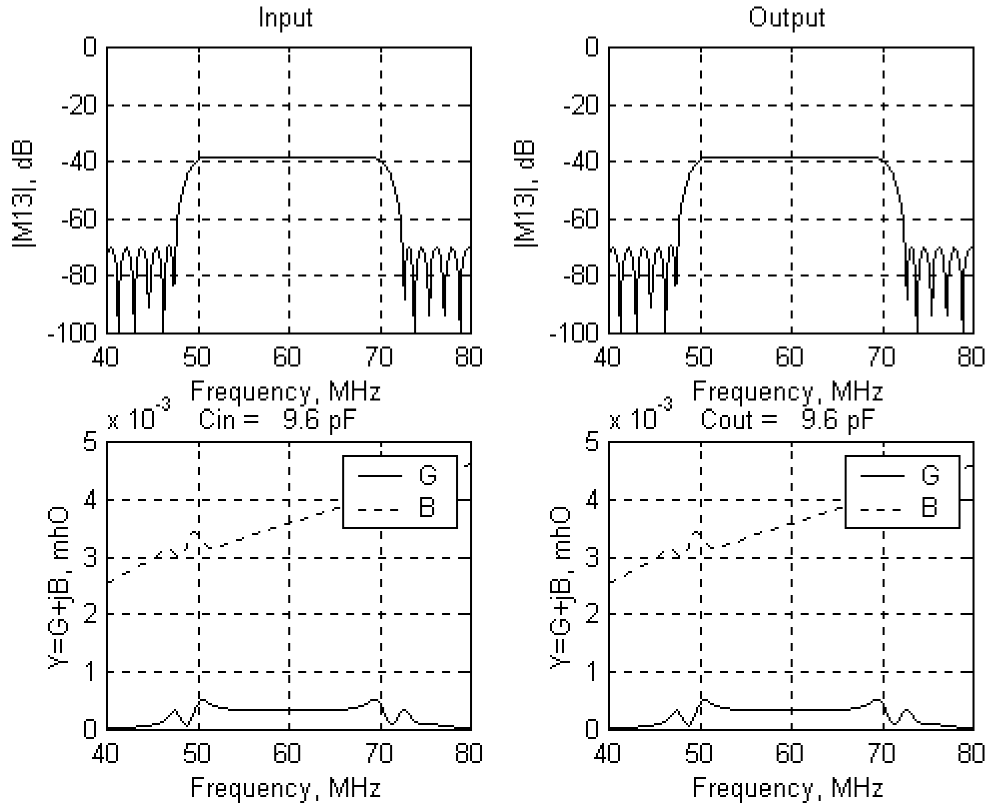


a) P-matrix elements

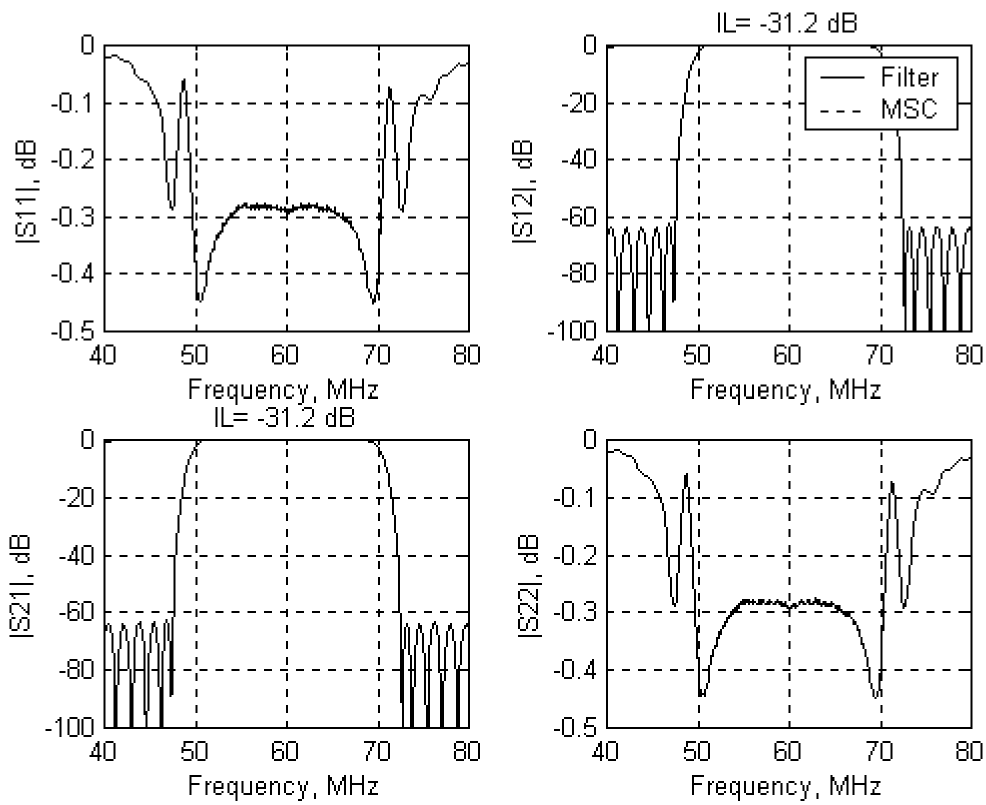


b) S-parameters

Fig. 3.11. Example # 6: calculation results for the dual-track SAW filter (128° YX $LiNbO_3$, MSC neglected)



a) P-matrix elements



b) S-parameters

Fig. 3.12. Example # 6: calculation results for the dual-track SAW filter
(128° YX $LiNbO_3$, MSC included)

3.2. MSC Modeling

3.2.1. Data Format

The test program for MSC modeling is called *TEST_MSC*. The basic MSC modeling assumption is the two-mode approach where two lowest order orthogonal modes (symmetric and antisymmetric) are taken into account [1, Chapter 5], [12-14]. Different modeling technique are applied for closed-form description of the symmetric and antisymmetric modes, particularly: 1) quasi-static approximation [5, Chapter 5]; 2) reflective array model (RAM) [5, Appendix E]; 3) coupling-of-modes (COM) model [9-11]; 4) field approach [12]. MSC characteristics are presented in the form of the MSC scattering parameters as the four-port acoustical network.

Four tutorial examples on MSC modeling are included. All these examples are located in the subdirectory **MSC** of the directory **EXAMPLES**. The basic features of these examples are listed in Table 3.6.

Table 3.6

List of Tutorial Examples on MSC Modeling

Data file	**Substrate**		**MSC parameters**			
	Material	**Cut**	**Number of strips**	**Central frequency, MHz**	**Synchronous frequency, MHz**	**Metallization ratio**
MSC_1.dat	$LiNbO_3$	128^o YX	80	34.2	42.7	0.5
MSC_2.dat	$LiNbO_3$	YZ	120	30.5	43	0.375
MSC_3.dat	$LiNbO_3$	YZ	100	68.8	86	0.45
MSC_4.dat	$LiTaO_3$	YZ	125	68.8	86	0.45

MSC data files have an extension *.dat* and contain the structural, material, and frequency data which are necessary for MSC analysis. They must satisfy to the format shown in Table 3.7.

Table 3.7

Example of the Formatted Data File for MSC Modeling

```
MSC TEST EXAMPLE # 1 (Y128X LiNbO3)

Substrate material                          LiNbO3   [128 YX]
Coupling factor, %                 k2 =     5.5
Substrate permittivity             ep =     55
Free-surface SAW velocity, m/s     v0 =     3990
Effective SAW velocity, m/s        ve =     3882
Metallization ratio                km =     0.5
Central frequency, MHz             f0 =     34.16
Synchronous frequency, MHz         fpi=     42.7
Acoustic aperture, mcm             w  =     1000
Number of strips                   N  =     80
Frequency range, MHz               Fs =     0
                                   Fe =     85.5
                                   dF =     0.1
```

Again, the first two lines of the data file may contain arbitrary comments (paper title, substrate material, etc.). They must be skipped (left blank) if there are no comments. The next line contains a string variable with the name of a substrate material, the cut information should be included in the square brackets (LiNbO3 [128 YX] in this example).

The next 12 lines contain the commented numerical data on the piezoelectric coupling factor (per cent), substrate permittivity (dimensionless), free-surface SAW velocity (m/sec), effective SAW velocity under the MSC (m/sec), metallization ratio or duty factor (dimensionless), MSC central frequency and synchronous frequency (MHz), acoustic aperture (micrometers), number of the MSC strips, frequency range for the analysis (MHz) including start and stop frequency and discretization interval.

The material constants used for different substrates used in the MSC tutorial examples have been listed in Table 3.4.

The users can compose their own data files for the MSC modeling following the format of the sample data file given in Table 3.7.

3.2.2. Running MSC Test Examples

The test program is invoked by typing *TEST_MSC*. At the prompt "Data file name:" one of the following file names *MSC_1.dat, MSC_2.dat, ... MSC_4.dat* should be entered without quote characters. An extension *.dat* is assumed by default and can be omitted. Thefore, both forms are valid in the following example:

Data file name ["MSC_1.dat"]: MSC_2.dat
Data file name ["MSC_1.dat"]: MSC_2

The file name by default is shown in the square brackets. This name is accepted after pressing the "Enter" key. For convenience, the name of the last executed data file is written in the file MSC.INI to be created and located in the current directory.

Test results are presented in the graphical form. Two different formats of the plots can be selected while executing the program *TEST_MSC:*

1) compact form where the results are presented in the form of the small-size figures (subplots) when the internal variable kplot=1;
2) comprehensive form where the results are presented in the form of separate large-scale figures when kplot=2.

The basic MATLAB workspace variables are listed in Table 3.8.

Table 3.8

Basic MATLAB Workspace Variables for the Program *TEST_MSC*

Variable name	Type	Meaning
f	Real	Frequency, MHz
S11_QS	Complex	MSC S-parameters (in the quasi-static approximation)
S12_QS	Complex	
S13_QS	Complex	
S14_QS	Complex	
S11_RAM	Complex	MSC S-parameters (reflective array model)
S12_RAM	Complex	
S13_RAM	Complex	
S14_RAM	Complex	

S11_COM	Complex	MSC S-parameters (coupling-of-modes)
S12_COM	Complex	
S13_COM	Complex	
S14_COM	Complex	
S11_FLD	Complex	MSC S-parameters (field approach)
S12_FLD	Complex	
S13_FLD	Complex	
S14_FLD	Complex	

3.2.3. MSC Example # 1 (MSC_1)

Test results (MSC scattering parameters) for this example calculated by four different models are presented in Figs. 3.13, 3.14 for the data file *MSC_1.dat*. The substrate material is 128° YX lithium niobate, number of the MSC strips N=N_0=80 (optimum value), metallization ratio η=0.5, working frequency f_0=34.16 MHz, synchronous frequency f_π=42.7 MHz (the ratio f_π/f_0=1.25). For convenience, the narrowband scattering parameters S_{11} and S_{12} are shown in the frequency range from 30 to 55 MHz (Fig. 3.13). The wideband parameters S_{13} and S_{14} are shown in the wider frequency span from 0 to 85 MHz (Fig. 3.14).

As can be seen, all the techniques give similar results within the MSC working frequency range (far away from the stopband frequency). As expected, while giving the correct results within the frequency range of interest, the quasi-static approximation fails to predict stopband behavior. For other three models which take into account the strip reflections, some discrepancy is observed in the stopband that can be explained by the different assumptions and approximations used in these modeling techniques.

3.2.4. MSC Example # 2 (MSC_2)

The substrate material in this example is YZ lithium niobate. The MSC parameters are as follows: number of the MSC strips N=120 that is close to the optimum value N_0=122 for the metallization ratio η=0.375, working frequency f_0=30.5 MHz, synchronous frequency f_π=43 MHz (the ratio f_π/f_0=1.4). These MSC parameters correspond to the MSC structure investigated theoretically and experimentally in [12].

The modeled results for this example calculated by different modeling techniques are shown in Fig. 3.15 (compact form). Again, within the model constraints applied good correspondence between different modeling techniques is observed.

3.2.5. MSC Example # 3 (MSC_3)

The substrate material is YZ lithium niobate. Number of the strips N=100 that is slightly lower than the optimum value N_0=105, the metallization ratio η=0.45, working frequency f_0=68.8 MHz, synchronous frequency f_π=86 MHz (the ratio f_π/ f_0=1.25). These data correspond to the MSC modeled and measured in [14].

The modeled results are shown in Fig. 3.16 (compact form).

3.2.6. MSC Example # 4 (MSC_4)

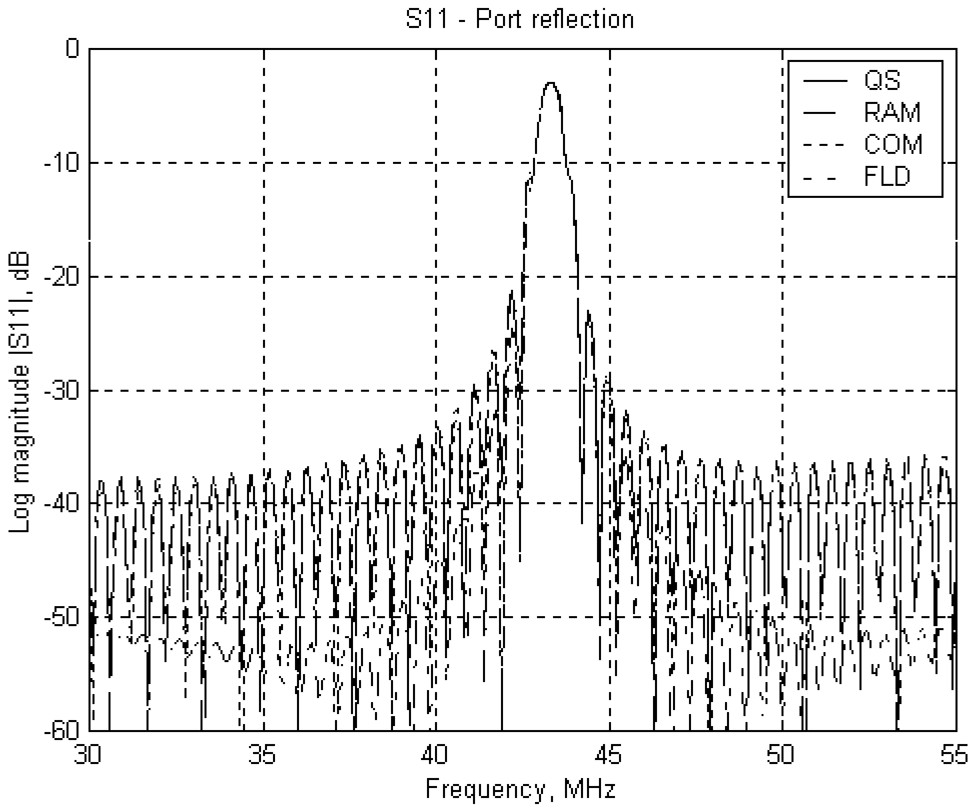
The MSC parameters are virtually the same as in the MSC example # 3, except for the number of the MSC strips N=125 that is higher than the optimum value N_0=105. The modeling results are shown in Fig. 3.17 (compact form).

3.2.7. Comparison with Experimental Data

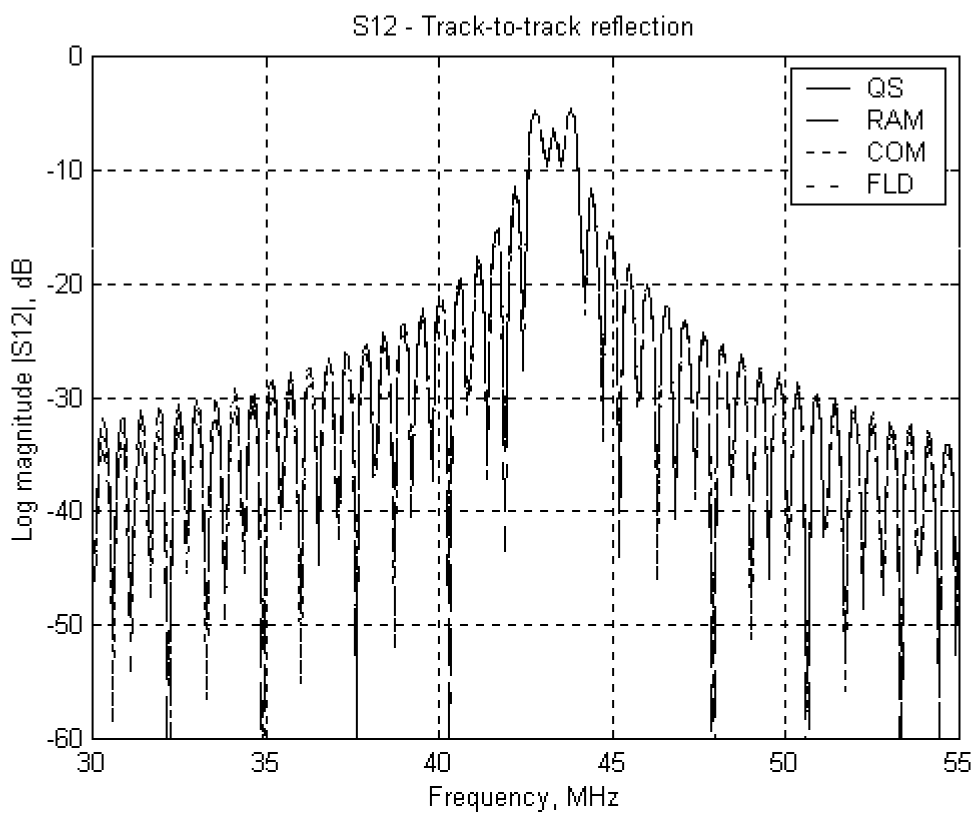
The detailed comparison of the modeled (field approach, MSC Example #2) and measured data [12] is presented in Fig. 3.18 for S_{13}, S_{13} and S_{14} parameters, in the stopband region. There is good correspondence between the modeled and measured results.

There is also comparison of the modeled (field approach, MSC Example #3) and experimental data [14] shown in Fig. 3.19 for S_{13} and S_{14} parameters. Again, good correspondence between theory and experiment can be observed, with the slight discrepancy attributed to the MSC resistive loss that is not accounted for in the model.

It is worthy noting that in conventional MSC applications the frequency range of interest lies far away from the stopband region. To the first order, all four modeling techniques give essentially the same results in the working frequency range. The stopband MSC analysis has rather theoretical importance and has been used as the second-order effect for estimating the accuracy of the different models.

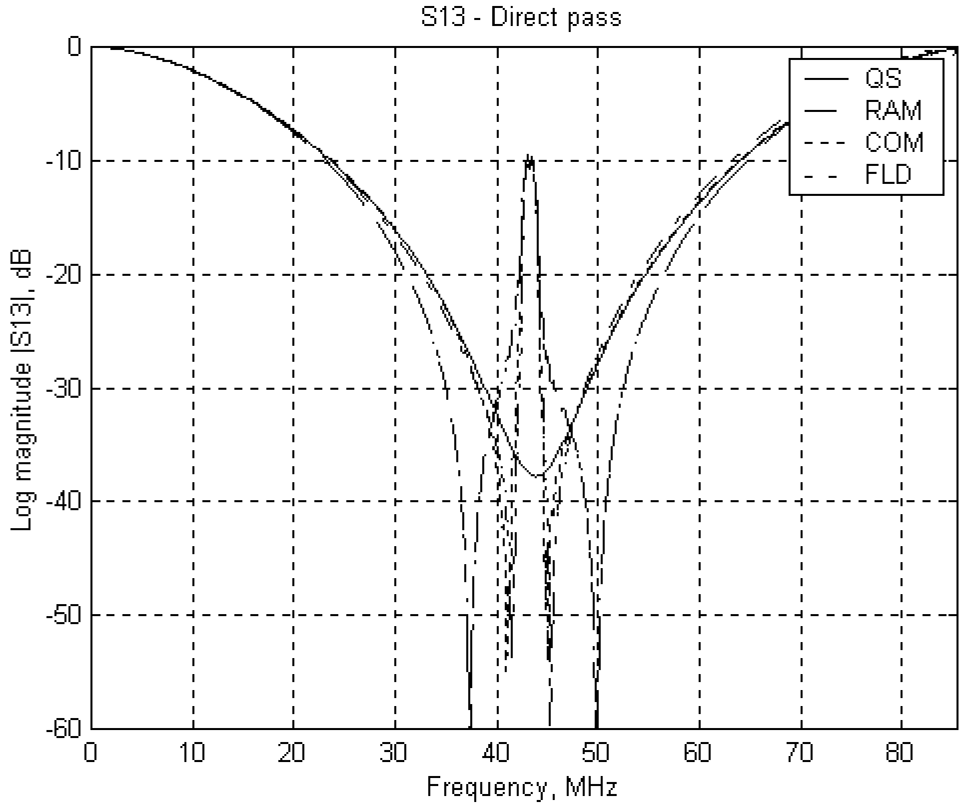


a) Port reflection S_{11}

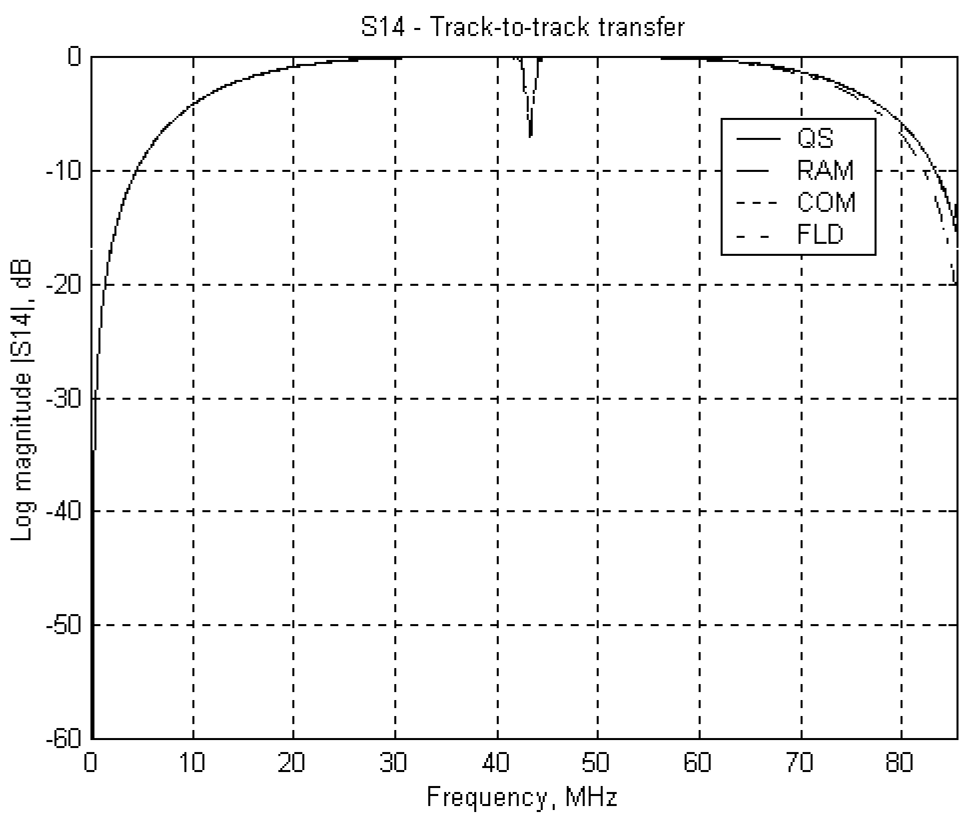


b) Track-to-track reflection S_{12}

Fig. 3.13. MSC Example # 1: narrowband MSC scattering parameters S_{11} and S_{12}



a) Direct pass S_{13}



b) Track-to-track coupling S_{14}

Fig. 3.14. MSC Example # 1: wideband MSC scattering parameters S_{13} and S_{14}

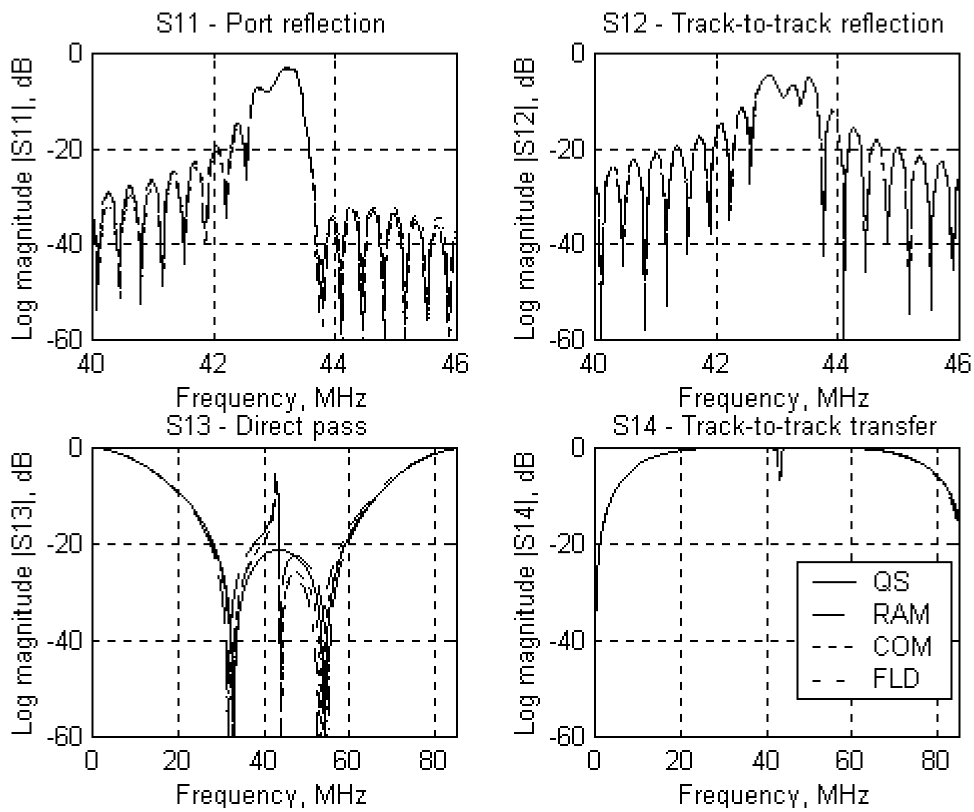


Fig. 3.15. MSC Example # 2: MSC scattering parameters

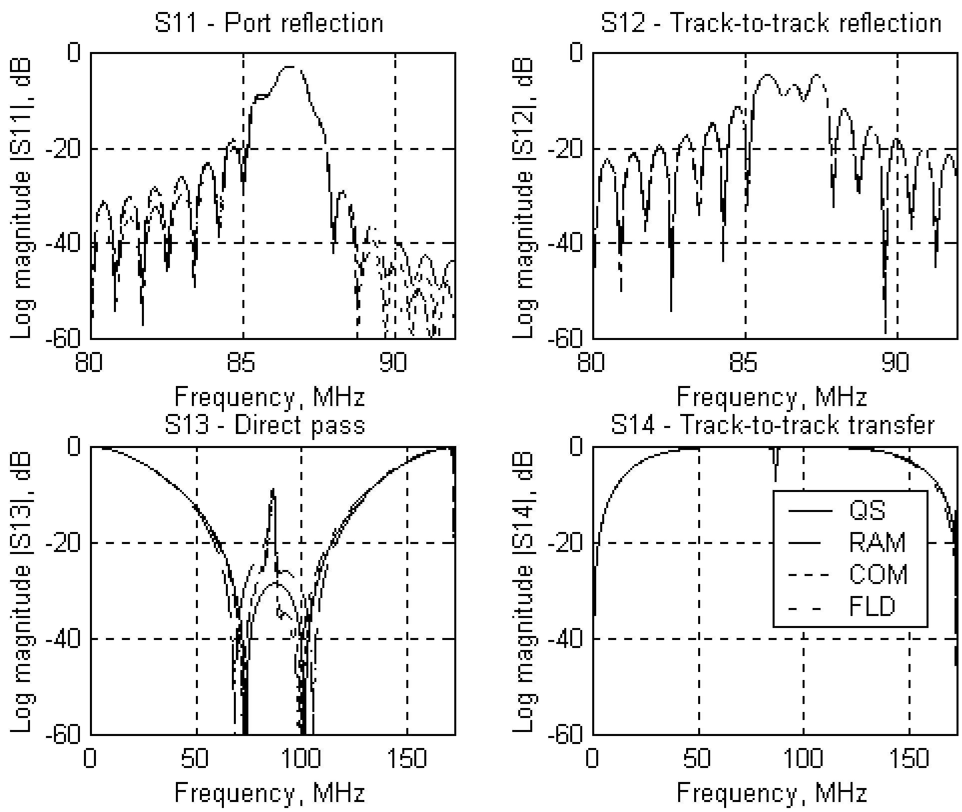


Fig. 3.16. MSC Example # 3: MSC scattering parameters

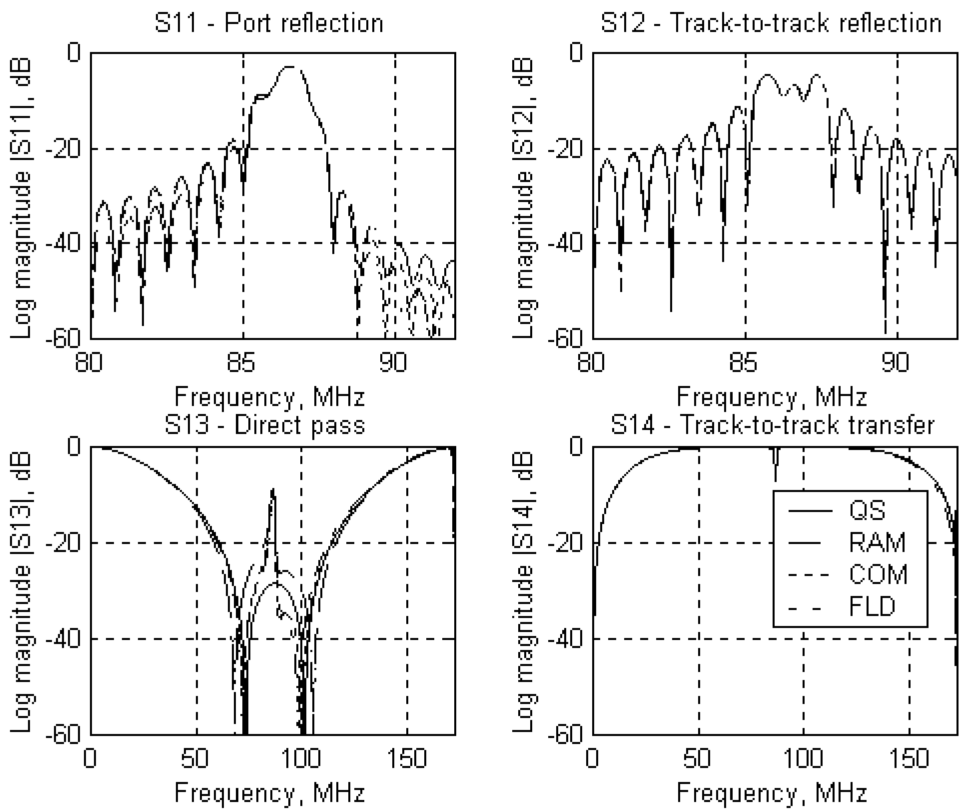


Fig. 3.17. MSC Example # 4: MSC scattering parameters

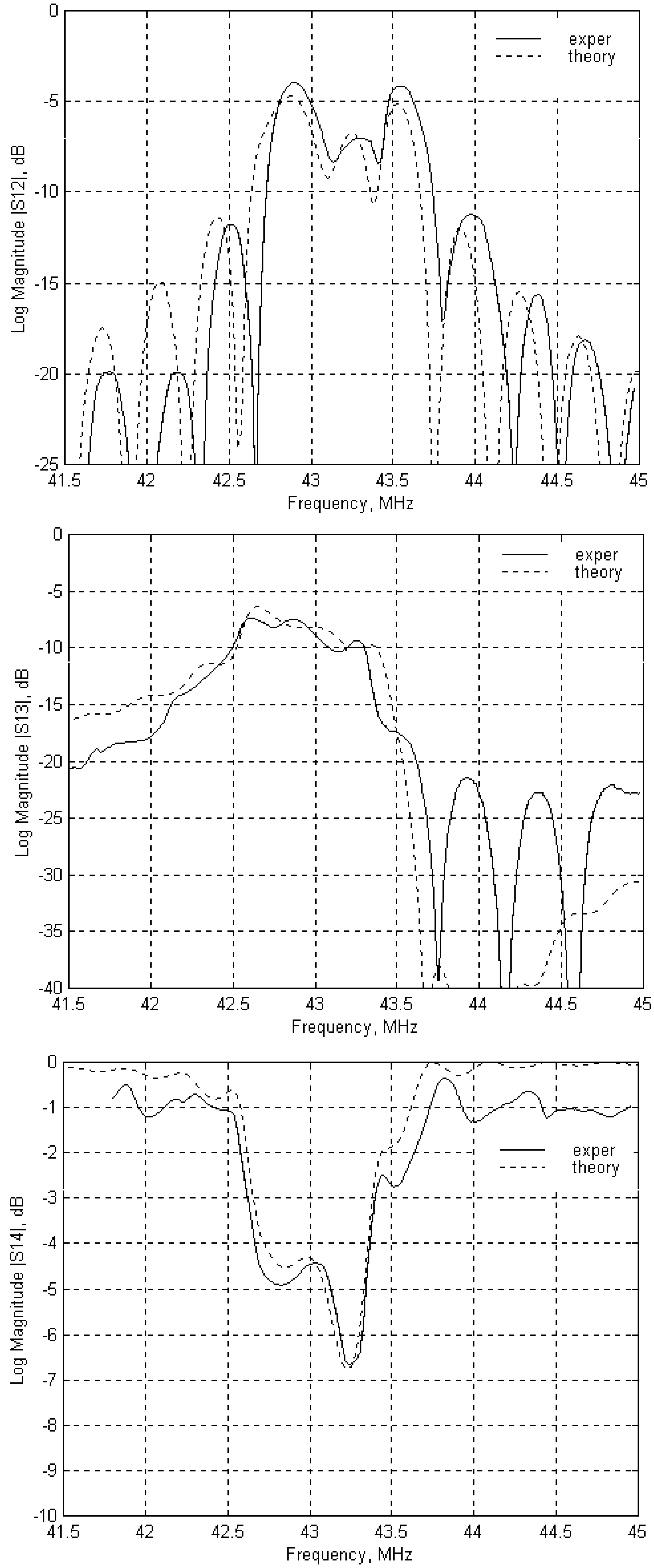
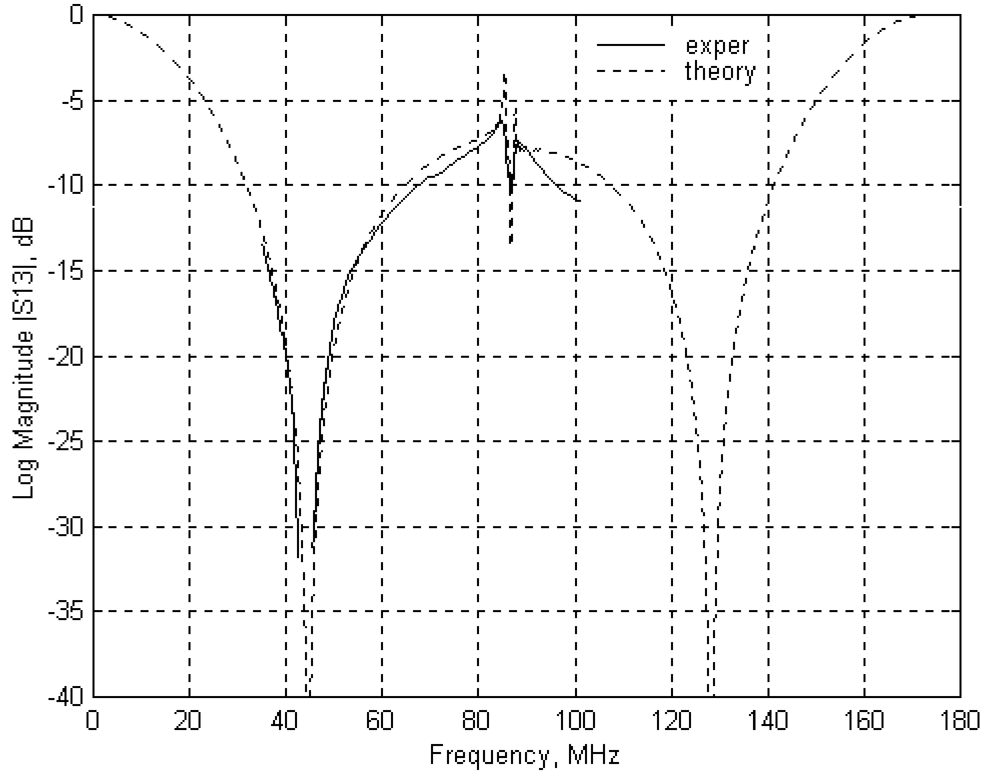


Fig. 3.18. Comparison of the modeled data (data file *MSC_2.dat*, field approach) with the experimental data [12]



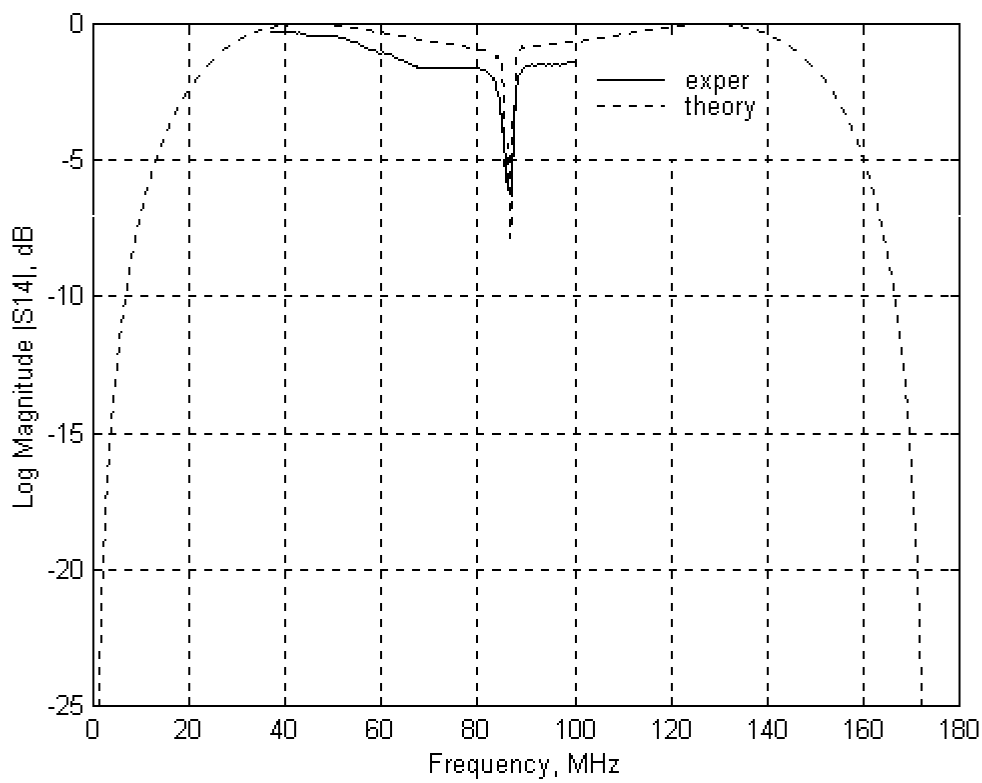


Fig. 3.19. Comparison of the modeled data (data file *MSC_4.dat*, field approach) with the experimental data [14]

REFERENCES

1. A. S. Rukhlenko, “SAW filter analysis in the quasi-static approximation: theoretical background (theory and algorithms)”, MatLab Toolbox documentation, Version 1.1, 2000.

2. A. S. Rukhlenko, *Analysis and approximational synthesis of the surface acoustic wave filters*, Ph. D. Dissertation, 1989.

3. A. S. Rukhlenko, “Closed-form admittance calculation for generalized periodic SAW transducers,” Proc. 1995 World Congress Ultrason., 1995, Berlin, pp. 379-382.

4. A. S. Rukhlenko, “Charge distribution and capacitance calculation for generalized periodic SAW transducers using Floquet's technique,” *Proc. 1994 IEEE Ultrason. Symp*., pp. 325-329.

5. D. P. Morgan, *Surface-wave devices for signal processing*, Amsterdam: Elsevier, 1985.

6. A. Erdelyi, *Higher transcendental functions*, McGraw-Hill, 1953.

7. H. Engan, “Surface acoustic wave multielectrode transducers,” *IEEE Trans. Sonics and Ultrason.*, vol. SU-22, pp. 395-401, Nov. 1975.

8. A. S. Rukhlenko, “Nodal analysis of multitransducer SAW devices,” *Proc. 1995 IEEE Ultrason. Symp*., pp. 297-300.

9. Y. Koyamada, S.Yoshikawa, “Coupled mode analysis of a long IDT”, *Rev. Electr. Commun. Lab.*, vol. 27, No 5-6, pp. 432-444, 1979.

10. D.-P. Chen, H.A.Haus, “Analysis of metal-strip SAW gratings and transducers”, *IEEE Trans. Sonics and Ultrason.*, vol. SU-32, No 3, pp. 395-408, 1985.

11. G. Sholl et al., “Efficient analysis tool for coupled-SAW-resonator filters,” *IEEE Trans. UltraSon., Ferroelectrics, and Freq. Control*, vol. UFFC-38, No 3, pp. 243-251, May 1991.

12. J. Bording, K.A. Ingebrigtsen, “Scattering parameters of surface-wave multistrip directional couplers: a field approach”, *Electronics Letters*, vol. 9, No 3, pp. 63-64, Feb. 1973.

13. K.A.Ingebrigtsen, “Normal mode representation of surface wave multistrip couplers,” *Proc. 1973 IEEE Ultrason. Symp*., pp. 163-167.

14. F.G. Marshall, C.O.Newton, and E.G.S. Paige, “Theory and design of the surface acoustic wave multistrip coupler,” *IEEE Trans. Microwave Theory and Techn.*, vol. MTT-21, No 4, pp. 206-215, Apr. 1973.