Imaging and Data Interpretation
Department
Stieltjesweg 1
P.O. Box 155
2600 AD DELFT
The Netherlands

www.tno.nl

T +31 15 269 2000
F +31 15 269 2111

TNO report

DII-RPT-040031

MATLAB implementation of the KLM model for piezoelectric transducer simulation
A user's manual

Date	September 13, 2004
Author(s)	Erik Droog
Reviewer:	Uilke Stelwagen

All rights reserved.
No part of this publication may be reproduced and/or published by print, photoprint, microfilm or any other means without the previous written consent of TNO.

In case this report was drafted on instructions, the rights and obligations of contracting parties are subject to either the Standard Conditions for Research Instructions given to TNO, or the relevant agreement concluded between the contracting parties. Submitting the report for inspection to parties who have a direct interest is permitted.

© 2004 TNO

Summary

The KLM model is a one-dimensional electrical equivalent model developed in the seventees by Krimholtz, Leedom and Matthaei [1], which describes the electromechanical behaviour of piezoelectric transducers. This report shortly describes the model itself, but focusses on a Matlab implementation of the model and the graphical user interface (GUI) around it.

This report as well as all Matlab routines can be found on the KLM CD-ROM accompanying this report (see section 3.3).

Imaging and Data Interpretation
Department
Stieltjesweg 1
P.O. Box 155
2600 AD DELFT
The Netherlands

www.tno.nl

T +31 15 269 2000
F +31 15 269 2111

Contents

1 The KLM model 4

2 The KLM Toolbox 5
2.1 Transducer structure 5
2.2 The user interface 7
2.3 Fitting to impedance data 9
2.4 Exporting data 14

3 References, M-Files and KLM CD-ROM 15
3.1 References 15
3.2 M-files 15
3.3 KLM CD-ROM 16

1 The KLM model

For the explanation of the KLM model, we start with a discussion of the most common configuration of a piezo-electric transducer, see Figure 1. The element is layered with different materials, and the surfaces are smooth. This means, that vibrations will have destructive or constructive interference due to the reflections. The piezo-electric material is sandwiched between electrodes, and will act as a capacitance at frequencies where no resonances occur. The DC electrical resistance is very high. A matching layer optimizes the energy transfer from the piezo to the medium, and the backing will absorb vibrations going in the opposite direction. This forces the element to look ahead. All the material characteristics and their dimensions will have a considerable effect on the frequency behaviour.

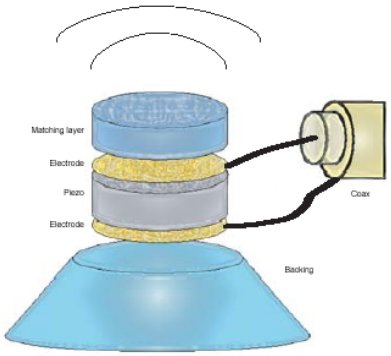


Figure 1. Exploded view of a simple transducer [from 2].

The KLM model is a fully electrical analogy of the transducer, and models the layers as electrical transmission lines. The piezoelectric material between the electrodes acts as a capacitance C_0, and the moving mass gives rise to an inductance L_1. The electro-mechanical conversion is modeled as a perfect transformer, with a winding ratio 1:Ω. The medium and backing are modeled as load resistances, absorbing all energy that is radiated into it. The values of L_1 and Ω depend on the frequency, so the model is easiest calculated in the frequency domain. The connections of the electrical components in the model are depicted in Figure 2. For a detailed discussion on the KLM model, we refer to [1,2 and 3].

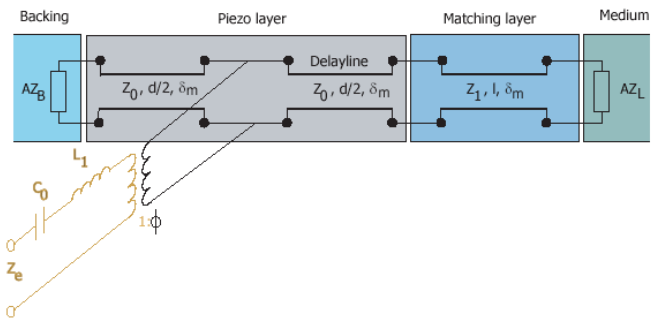


Figure 2. KLM model representation of a transducer with a backing and a single matching layer [from 2].

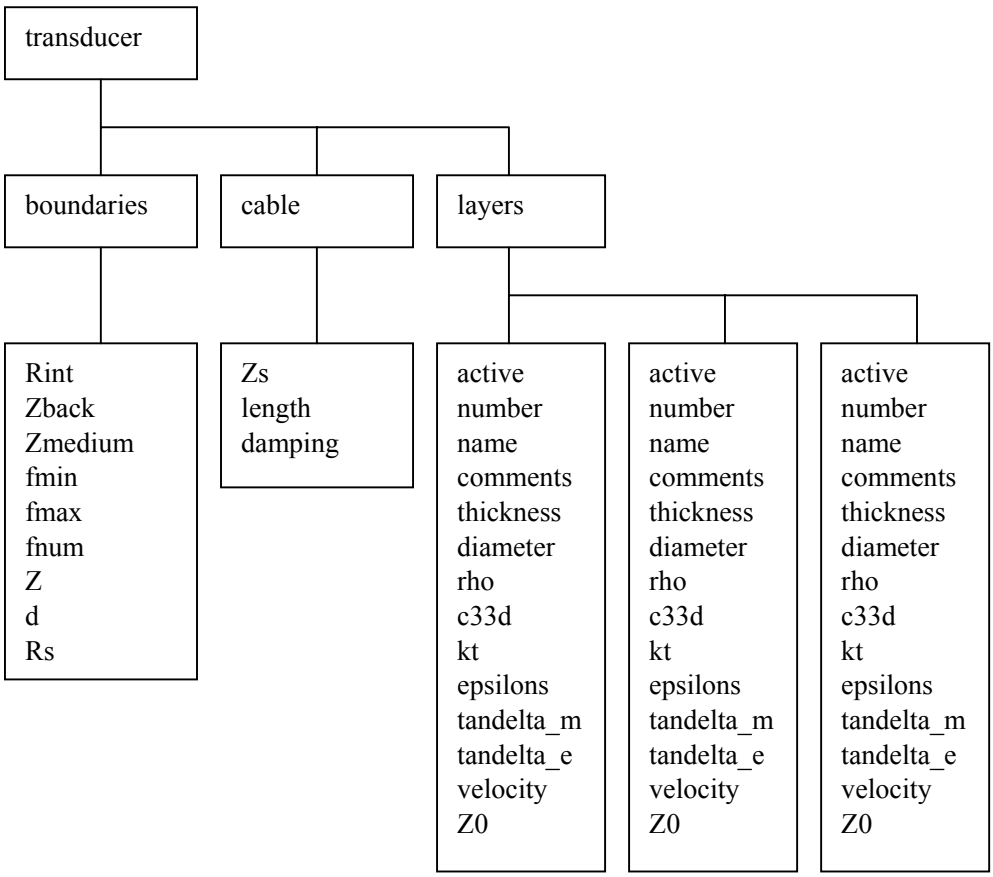
2 The KLM Toolbox

In the Matlab implementation of the KLM model, the user can define a transducer structure, consisting of a single piezoelectric layers and several passive layers. It is then possible to interactively change the material properties of the different layers and see the effect on the imput impedance and the transfer functions.

The implementation has been fully tested for Matlab R13 (version 6.x), and requires Matlab and the Signal Processing Toolbox.
In Matlab R14 (version 7.0) things work well too, only the routine fitgui.m doesn't work entirely properly though there is a viable workaround (see page 11 and following).
With Matlab R12 the GUI does not work too well at all.
For installation of the software from the KLM CD-ROM see section 3.3.

2.1 Transducer structure

The transducer is defined by means of a structured cell array in Matlab. In this array the material properties of the layers, as well as the boundary conditions and the cable are specified. The cell array is used as an input to most functions in the KLM toolbox. The array 'transducer' has the following structure:



Boundaries
Boundaries is a cell that contains the boundary conditions of the transducer:

Rint: Internal impedance of the power amplifier (Ohm)
Zback: Backing acoustic impedance (Rayl)
Zmedium: Medium acoustic impedance (Rayl)
fmin: lower frequency limit (Hz)
fmax: upper frequency limit (Hz)
fnum: number of frequency points
Z: acoustic impedance of the layer between backing and piezoelectric layer (Rayl) [optional].
d: thickness of the layer between backing and piezoelectric layer (m) [optional].
Rs: serial resistance (Ohm) [optional]

The fields Z and *d* have been added afterwards and enable the special situation of an extra passive layer between the backing and the piezoelectric layer. The velocity and mechanical damping of this layer can be specified in the file *make_deadbranch*.

Cable
The cell *Cable* contains the cable parameters:

Zc: Characteristic impedance (Ohm)
Length: Length of the cable (m)
Damping: Damping in terms of loss angle (*tan* δ_e)

Layers
Layers is a 1 x n cell array which contains the properties of the n layers of the transducer. For every layer, the following fields exist:

active 1 activates the layer, 0 disables the layer
number Defines the order of the layers
name Name of the layer [optional]
comments Comments [optional]
thickness Thickness of the layer (m)
diameter Diameter of the circular layer (m)
rho density (kg/m^3) [only used for first layer]
c33d bulk modulus (Pa) [only used for first layer]
kt coupling coefficient [only used for first layer]
epsilons dielectric constant [only used for first layer]
tandelta_m Mechanical loss angle
tandelta_e Electrical loss angle [only used for first layer]
velocity Sound velocity (m/s) [only used for passive layers]
Z0 Acoustic impedance (Rayl) [only used for passive layers]

The model assumes a circular transducer, of which the diameter can be specified in the layer fields *diameter*. The first layer has to be the piezoelectric layer and should have the *number* field set to 1. The subsequent layers have to be passive layers and should have the *number* field set to 2,3…etc.

2.2 The user interface

A GUI has been built with which the user can interactively change the properties of the transducer and see the effect on the input impedance and transfer functions. The GUI can be started wit the command:

klm(transducer)

where transducer is a transducer structure as described in the previous paragraph. A number of transducer structures are located in the *data* subfolder as .mat files. They can be loaded using the command:

load <structure_name>

where structure_name is the filename. For example, the command:

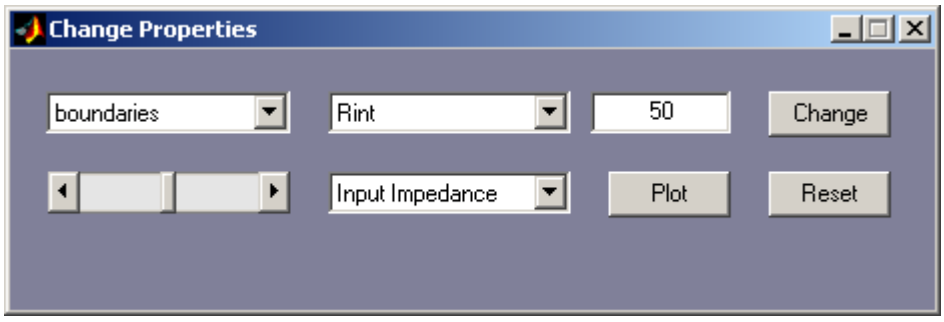
load pz27.mat

loads the transducer structure *pz27* representing a piezoelectric disk of the material PZ27 from Ferroperm piezoceramics, in air.

The GUI can now be started by typing for example[1]:

klm(pz27)

The following window will be opened



This is the *Change Properties* window of the GUI in which the user can change all the properties of the structure. Again, note that in the active layer *velocity* and *Z0* cannot be changed and that in the passive layers *rho, c33d, kt, epsilons* and *tandelta_e* cannot be changed.

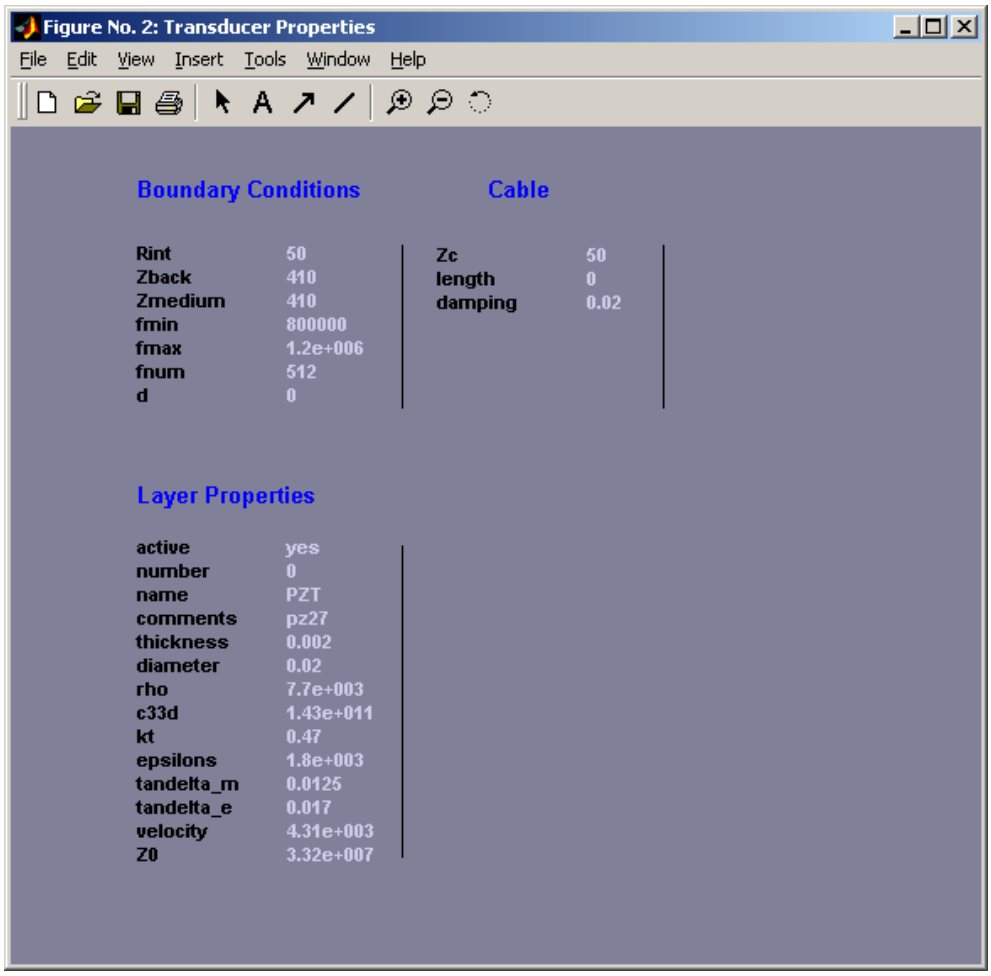
The layer can be selected in the top left popup menu, which shows the boundaries, cable and names of the layers. After selecting the layer (or cable or boundaries) the property can be selected in the top right popup menu. A new value can be entered in the top right box, or interactively changed using the slider. Use the *Change* button to confirm changed values. The type of plot can be selected using the bottom popup menu.

[1] The examples given in the following can be conveniently tried using the routine klmdemo.m.

The options are:

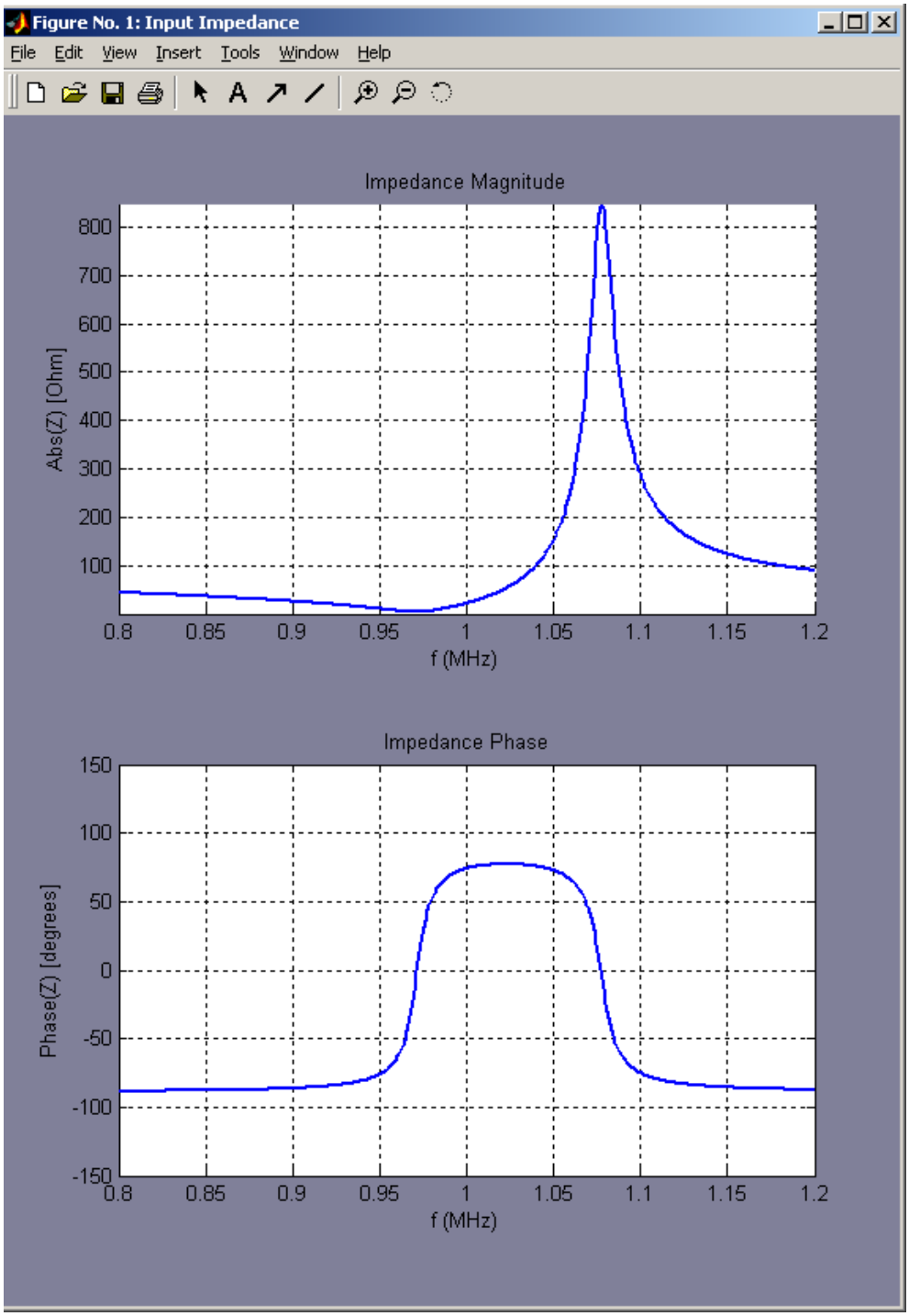
Input Impedance
Voltage to Pressure
Pressure to Voltage
Power transfer
Insertion Loss

At the same time another window, the *Transducer Properties* window, is opened which shows the current transducer properties:



In this window the transducer structure is visualised and it can be seen that this structure consists of a single piezoelectric layer in air. The *Plot* button in the *Change Properties* window can be used to plot the selected function. The *Reset* button reloads the original structure. Let's plot the input impedance, which should result in something like shown in the next figure[2].

[2] Due to auto-scaling, the actual axis limits in graphs in this report may differ somewhat from what is shown on screen when running the programs. However, axis limits can be easily adapted to the user's liking by first clicking the 'edit plot' button (the pointerlike arrow), selecting the axis with a left mouse click and next invoking a pop-up menu with a right mouse click from which the property editor is invoked by selecting 'properties' from that menu.



Anytime a plot is drawn or the *Change* button is pressed, the GUI outputs the variable *transducer_out* to the workspace. This variable contains the current transducer structure and can be used for fitting.

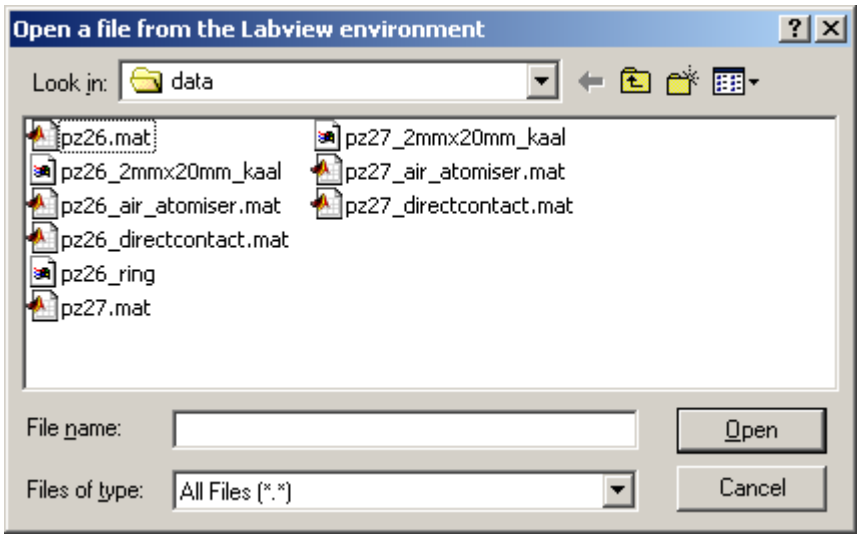
2.3 Fitting to impedance data

A useful method to estimate transducer properties is fitting the model to measured impedance data. The impedance measurements can be performed using the HP Vector

Analysor in Gerrit van Dijk's room at the TU Delft. The measurements can be stored using the LabView program TDHPC.VI. The stored files can be loaded into the GUI using the command[3]:

impdata = load_data

This will open a dialog in which the user can select a datafile containing the impedance data:, e.g.



In the folder *data* there are some impedance datafiles. These files have a binary format (see HP Vector Analysor manual and/or load_data.m) and do not necessarily have an extension.

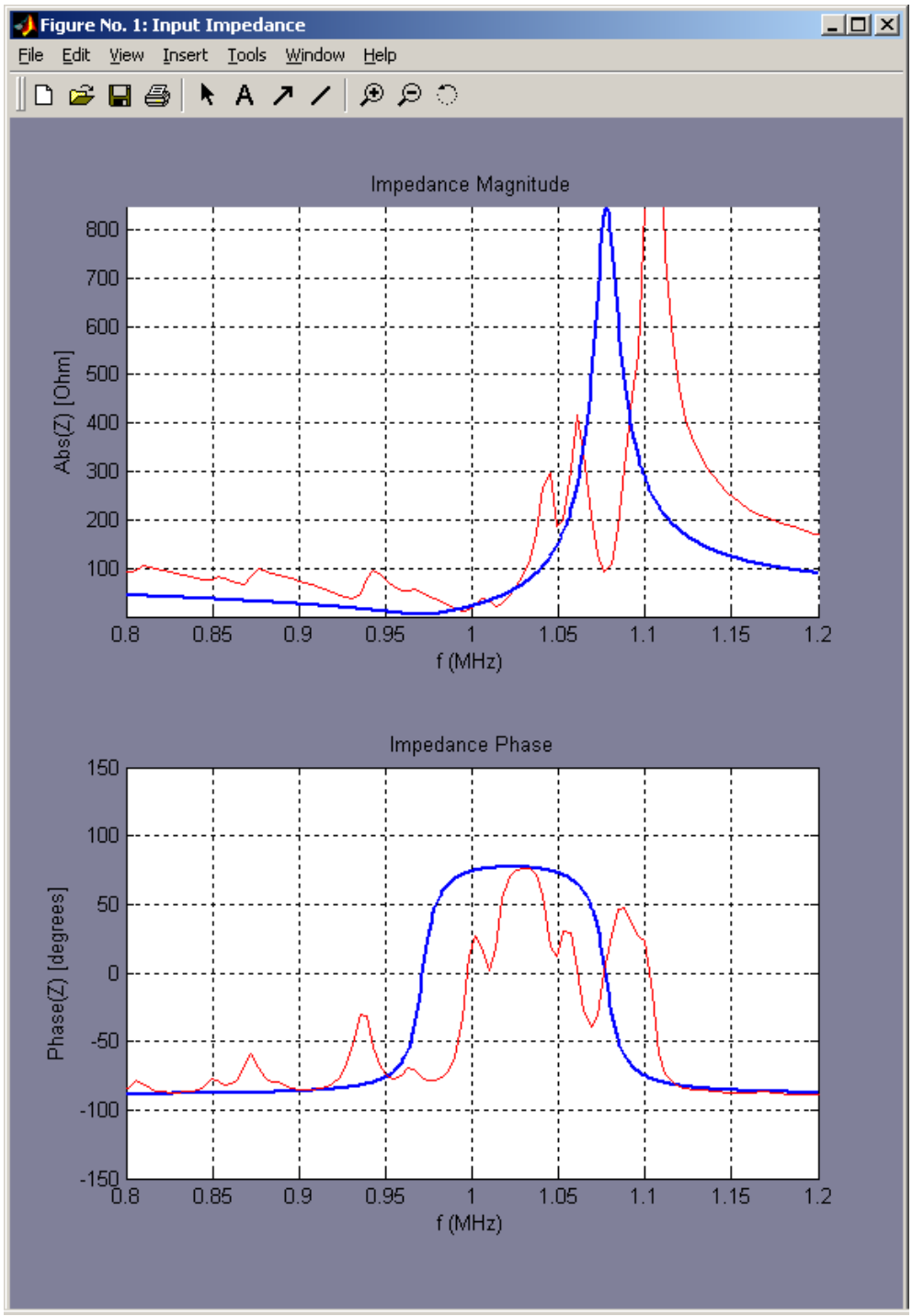
Let's load the file *pz27_2mmx20mm_kaal* from the *data* folder. The user has to select the compensation for the probe with which the measurement is performed. In this case, it is the *Copper probe*:

Now the measurement data is loaded into the workspace in the variable *impdata*.

When the *plot* button in the GUI is pressed again, the impedance data is shown together with the model data.

[3] Provided the directory where the data is located is either in the Matlab path or is the current directory.

For example,

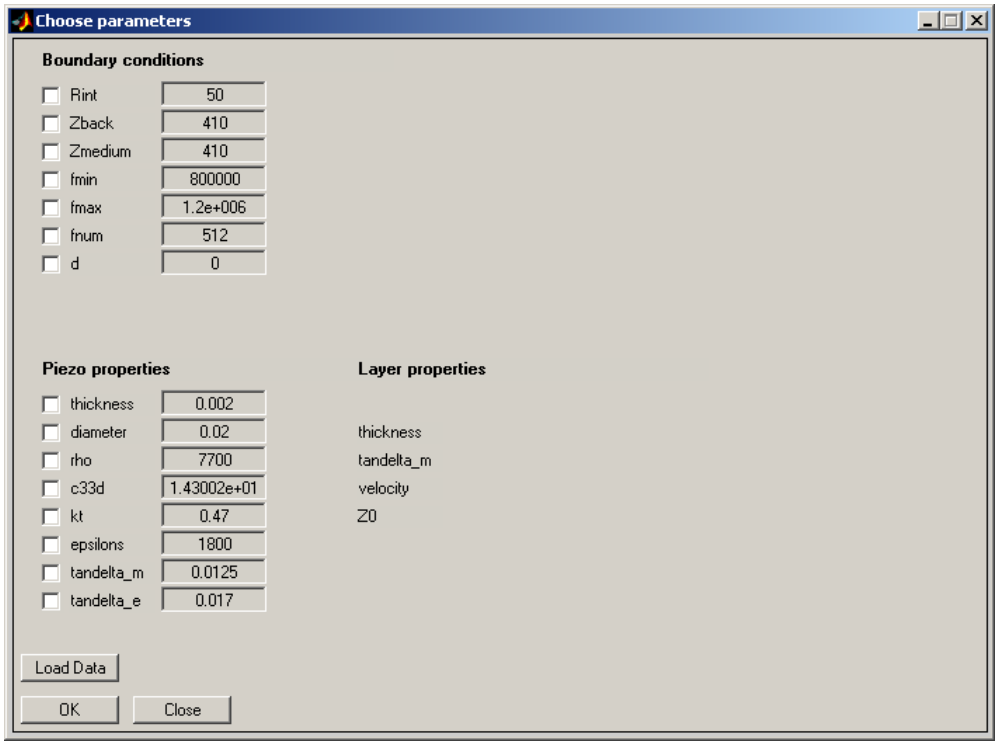


The user can now change the transducer properties by hand to fit the model to the data, or use the fitting GUI to do this for him. In this case, it seems that the thickness of the piezo is less in the measured data and that the mechanical loss is lower in the measured data. It should be noted that in this case the measured data is quite noisy, so automatic fitting will possibly be not so adequate.

Nevertheless, let's do the fitting[4]:

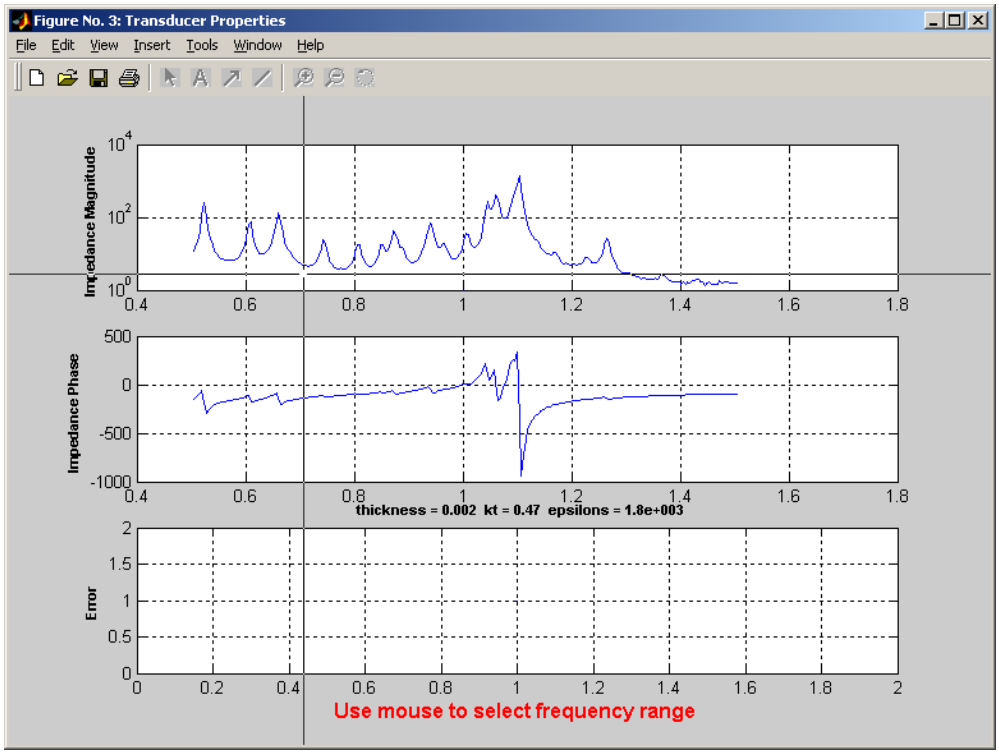
fitgui(transducer_out)

This opens the main fitting GUI window, where the user can select the properties he wants to vary during the fitting procedure. The values next to the selected properties can be altered and represent the initial values used in the fitting algorithm. The fitting algorithmn is based on the function *fminsearch* and uses the squared difference of the fitted and modeled data points.



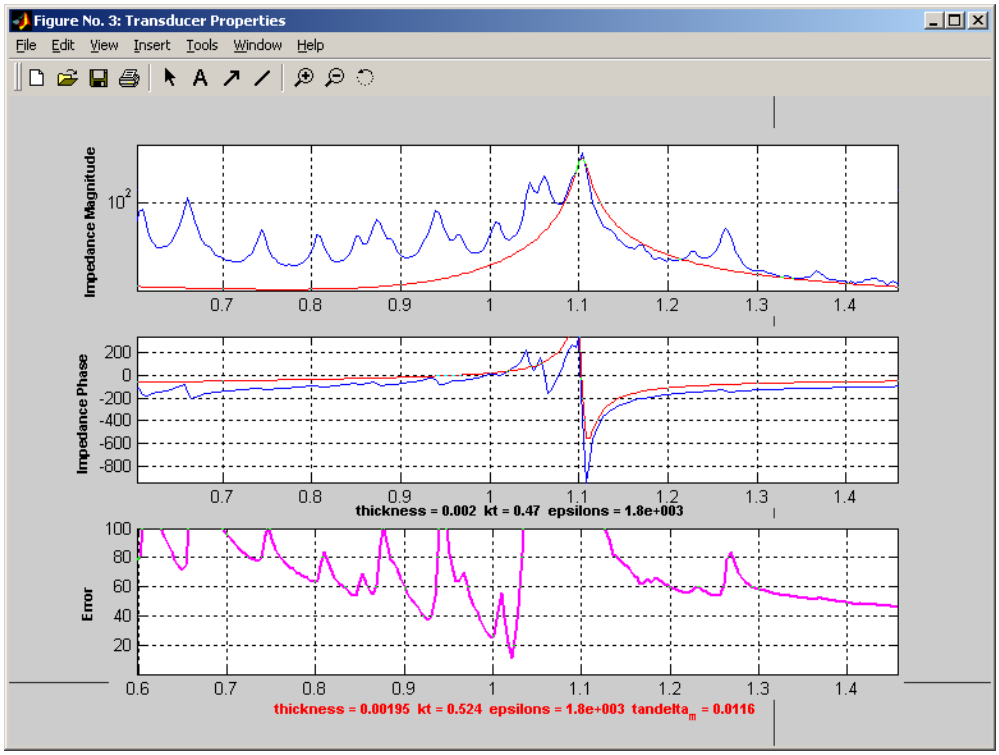
We could load the measurement data using the *Load Data* button, but the default used data is the *impdata* in the workspace, if present. If we select the thickness, kt, epsilons and tandelta_m, and press *OK*, we will see the following:

[4] In Matlab R13 this works properly. In Matlab R14 selecting a lower and upper frequency bound with the cross hair and the mouse may go wrong. The reasons (and remedy) are currently unclear. A viable workaround however is the following procedure. First move the cross hair to the desired lower frequency bound., then carefully press the cross hair (single mouse click) and repeat this (not too fast) until the cross hair disappears (usually two well seperated clicks suffice). This disappearing indicates the lower frequency bound has been read from the cross hair position in the plot. Next, by moving the mouse the cross hair will reappear and now the upper frequency bound may be selected and read (this usually requires only one mouse click).



We can click twice to select the frequency range to be used for the fitting algorithm. The first click selects the lower bound and the second click selects the higher bound.

Now the fitting will start and the updated properties are shown below the plots:



The fitting can be aborted at any time by pressing CTRL-C. The updated transducer structure is continuously written to the workspace in the variable *transducer_fit*. This variable can be loaded into the GUI by the command:

klm(transducer_fit)

Now we have fitted the transducer model to the measured data and estimated the material properties.

2.4 Exporting data

It is possible to write the input impedance and transfer functions to a variable using the function *xducer_plots*:

plots =xducer_plots(transducer_out)

writes the functions to a struct array *plots*, which exists of all the plots that are possible in the GUI:

plots =

freq: [1x1 struct] [frequency scale]
II: [1x512 double] [input impedance]
VP: [1x512 double] [voltage to pressure]
PV: [1x512 double] [pressure to voltage]
PT: [1x512 double] [power transfer]
IL: [1x512 double] [2-way insertion loss]
WF: [1x1 struct] [time pressure waveform]

To plot the input impedance, for example, possible syntaxes would be:

plot(plots.freq.scale , real (plots.II))

or

plot(plots.freq.scale , abs (plots.II))

3 References, M-Files and KLM CD-ROM

3.1 References

1. R. Krimholtz, D. A. L., G.L. Matthaei (1970). "New equivalent circuits for elementary piezoelectric transducers." Electronic letters **6**(13): 389-399.

2. Vos, H.J., 2004. The modelling, design and prototyping of a 20MHz – 40MHz harmonic intravascular ultrasound transducer. Msc. Thesis, TU Delft.

3. G.S. Kino, Acoustic waves: Devices, imaging and analog signal processing. Prentice-Hall Inc, New Jersey, 1987.

3.2 M-files

The main m-files are the following:

fitgui
syntax: fitgui(transducer)

description:
Opens the GUI that can be used to fit model to data

klm
syntax: klm(transducer)

description:
Opens the GUI that can be used to change transducer parameters and plot transfer functions and input impedance.

load_data
syntax: impdata = load_data

description:
loads a impedance measurement datafile from the HP vector analyzer created by the Labview program at Gerrit van Dijk's lab. The data is written in the variable *impdata*.

xducer_plots
syntax: plots = xducer_plots(transducer)

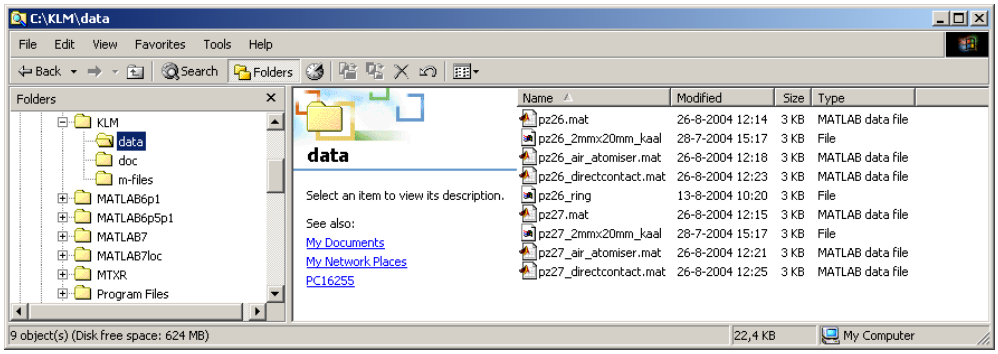
description:
writes the input impedance and transfer functions into the strucutre *plots.* This structure contains the frequency scale, the input impedance, and the transfer functions:

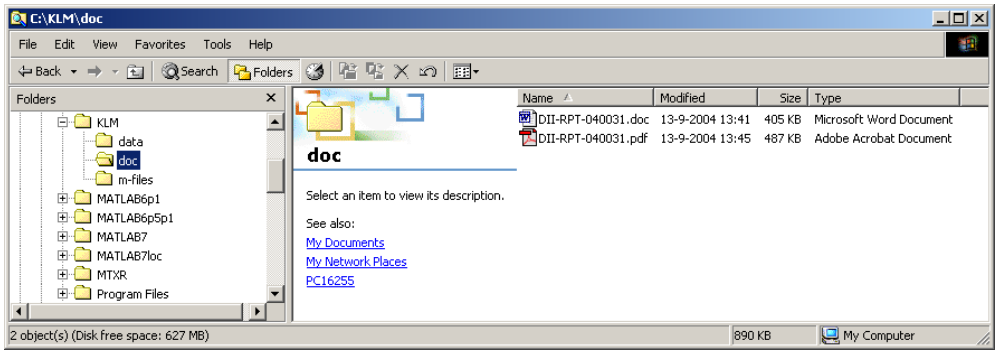
plots.freq.scale	frequency scale [real]
plots.II	input impedance [complex]

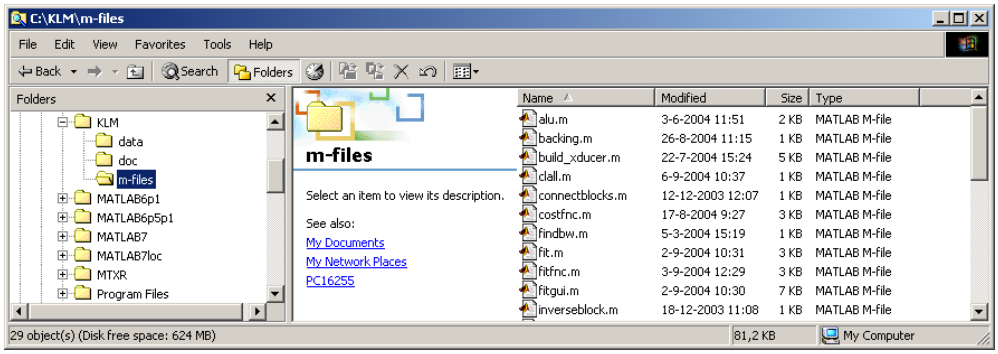
plots.VP	voltage to pressure transfer [complex]
plots.PV	pressure to voltage transfer [complex]
plots.PT	power transfer [complex]
plots.IL	insertion loss [complex]

3.3 KLM CD-ROM

All m-files needed to run the KLM GUI and the associated programs can be found on the CD-ROM accompanying this report. The installation is very straightforward, just copy the directories on the disk to a convenient location, e.g. to C:\KLM. This directory then should look something like this:







Then start Matlab, go to the directory C:\KLM\m-files, run klmstartup.m located there and everything is set to go. Next, a demonstration with comments (as discussed in sections 2.2 and 2.3) can be easily run with the script klmdemo.m.