

# FDBFIT device

FDBFIT device.....	1
1 Background.....	1
2 Fitting process .....	3
2.1 How FDBFIT works.....	3
2.2 Non-Default Situations.....	7
3 Running FDBFIT from EMTWork.....	8
3.1 FDBFIT device data .....	8
3.1.1 Fitting Data.....	8
3.1.2 Measured data file names .....	9
3.1.3 Magnetizing branch .....	10
3.2 Running a case.....	10
3.3 FDBFIT model data file format.....	12

## 1 Background

The FDBFIT device is used to generate High Frequency Transformer (HFT) models. The HFT model is one module of the more comprehensive Frequency-dependent Transformer Model, which consists of several components (see Figure 1–1) that can be combined to obtain the desired simulation capabilities. These components or building blocks are:

- ❑ Power frequency module (i.e, BCTAN, TRELEG, TOPMAG)
- ❑ Hysteresis/Saturation module
- ❑ Eddy currents module
- ❑ High Frequency Transformer module

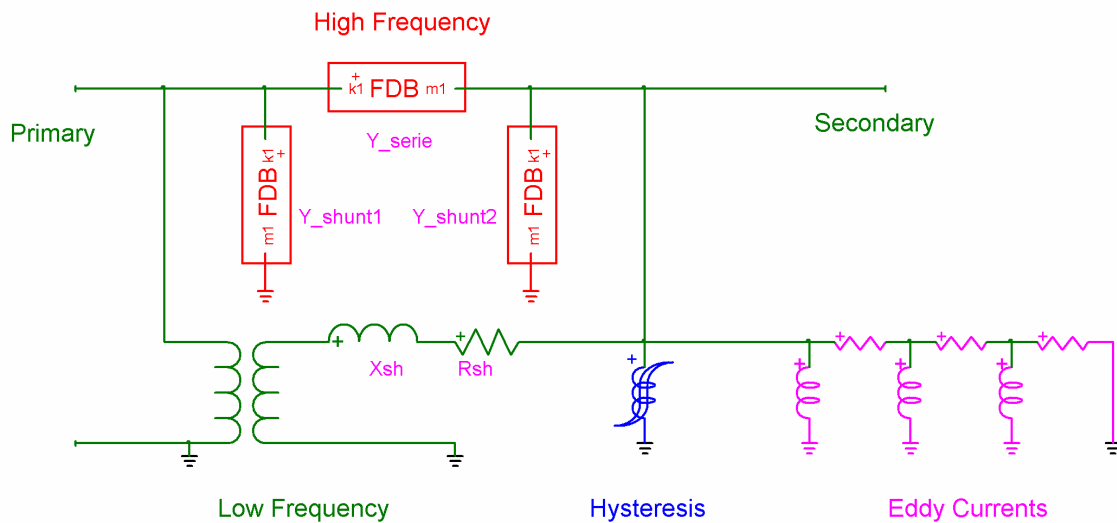


Figure 1–1 Frequency-Dependent Transformer Model

The HFT module reproduces the behaviour of a transformer over a wide frequency range. Raw input data for the HFT model can be in the form of field measurements of  $Y(\omega)$ ; that is, frequency scan measurements of the nodal admittance matrix of the transformer. Alternatively, raw data can also be supplied by some

transformer parameter calculation program, possibly based on the physical characteristics and construction details of the transformer. Whether the raw data is synthesized or measured, it must be manipulated and approximated with rational functions, which can later be represented as FDB branches in the EMTP. The FDBFIT Model Data Calculation Function handles the data manipulation and approximation process. On output, FDBFIT produces model data files that contain a multi-port, multi-phase pi-circuit. Each branch in the pi-circuit contains a multi-phase FDB model (thus the name FDBFIT).

For example, a two-winding, three-phase transformer would be represented with a two-port three-phase pi-circuit. The nodal admittance matrix  $\mathbf{Y}$  for this transformer can be partitioned in 3x3 blocks.

$$\mathbf{Y} = \begin{bmatrix} \mathbf{Y}_{11} & \mathbf{Y}_{12} \\ \mathbf{Y}_{21} & \mathbf{Y}_{22} \end{bmatrix}$$

where  $\mathbf{Y}_{ij}$  is an 3x3 block matrix of the form

$$\mathbf{Y}_{ij} = \begin{bmatrix} Y_{11} & Y_{12} & Y_{13} \\ Y_{21} & Y_{22} & Y_{23} \\ Y_{31} & Y_{32} & Y_{33} \end{bmatrix}$$

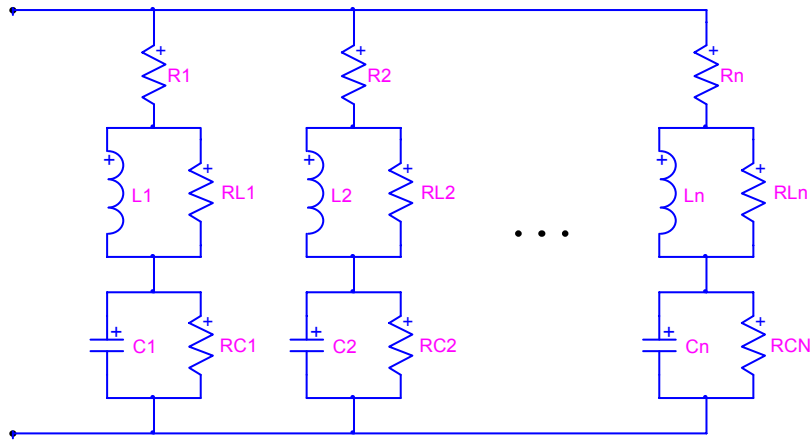
If  $\mathbf{Y}_{ij}$  is assumed to be balanced (or if all the diagonal elements are averaged to produce  $y_s$  and all the off-diagonal elements are averaged to produce  $y_m$ ) then,

$$\mathbf{Y}_{ij} = \begin{bmatrix} y_s & y_m & y_m \\ y_m & y_s & y_m \\ y_m & y_m & y_s \end{bmatrix}$$

Since the elements of the pi-circuit are described by balanced matrices, they can be modelled using zero and positive sequence parameters. The FDB branches produced by FDBFIT represent the approximations by rational functions of  $\mathbf{Y}_{shunt}$  and  $\mathbf{Y}_{series}$  (of the pi-circuit) in zero and positive sequence.

When the frequency response of the transformer is obtained from field measurements, it is not uncommon to find that due to instrumentation limitations, measured data that is accurate at high frequencies (e.g., 1 kHz to 500 kHz) loses accuracy in the power frequency range, especially the real part of the elements of  $\mathbf{Y}$ . Therefore, it is good practice to rely on the power frequency module for low frequency response and on the HFT module for high frequency response. This is achieved by subtracting the frequency behaviour of the power frequency module from the measured data, and then approximating the resulting difference function using FDB models. This is default option assumed in FDBFIT. The model data files obtained after running FDBFIT contain data for both the power frequency module and the HFT module.

The elements of  $\mathbf{Y}$  are approximated with rational functions which contain real as well as complex conjugate poles and zeros. The rational functions can then be realized with RLC networks (see Figure 1–2) which can be combined to produce the parameters of the equivalent pi-circuit.



**Figure 1–2 Generic structure of an FDB branch**

The general structure of these RLC networks reflects the known frequency characteristics of the admittance functions of a transformer:

1. Inductive behaviour at low frequencies which includes frequency dependent effects due to skin effect in the windings and iron core eddy current losses. These are simulated the power frequency transformer model, if present, and by the RL branches.
2. Series and parallel resonances from mid to high frequencies caused by winding-to-winding and winding-to-ground stray capacitances. These are reproduced by the RLC branches.
3. Predominantly capacitive behaviour at high frequencies, represented by a single RC series branch.

## 2 Fitting process

Although FDBFIT has been designed to require minimal user intervention, the nature of the problem of fitting measured transformer admittance functions with closed-form rational functions depends on a number of factors that cannot be fully accounted for in all cases, such as, the quality of the measurements (e.g., proper grounding techniques, leads, bandwidth of the equipment, etc.), the amount of noise present, and the self-consistency of the measurements. Self-consistency of measurements is important to guarantee the numerical stability of the final HFT model.

### 2.1 How FDBFIT works

This Section provides a simplified overview of the fitting process in order to give a better understanding of the optional features of FDBFIT that may be required to obtain a successful HFT model.

To create the FDB branches required to produce an HFT model, FDBFIT automatically goes through the following steps:

- > **Read measured nodal admittance matrix of the transformer as a function of frequency.** One of the simplest ways to measure the nodal admittance matrix of a transformer is to connect a variable-frequency voltage source to one of the terminals, while all other terminals (to be retained in the model) are shorted. The currents flowing into these terminals are proportional the elements of one column of  $\mathbf{Y}$ .

$$\begin{bmatrix} y_{11} & y_{12} & y_{13} \\ y_{21} & y_{22} & y_{23} \\ y_{31} & y_{32} & y_{33} \end{bmatrix} \begin{bmatrix} V_0 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} I_1 \\ I_2 \\ I_3 \end{bmatrix}$$

For example, if  $V_0 = 1$ , then  $I_1$ ,  $I_2$ , and  $I_3$  give  $y_{11}$ ,  $y_{21}$ , and  $y_{31}$  directly. By connecting the voltage source to terminals 2 and 3, the complete Y matrix as a function of frequency can be obtained. Since Y is symmetric, only  $N(N+1)/2$  currents have to be measured (where N is the product of the number of phases times the number of windings to be modelled).

The preferred sampling scale is logarithmic. 50 points per decade is probably sufficient, but more points allow a finer resolution of high frequency resonance peaks. The lowest scan frequency is generally determined by the characteristics of the instrumentation, but it should not be lower than the first resonant peak (typically below 5 kHz for large HV transformers). The maximum frequency in the scan is determined by the highest significant frequency (fmax) response required in the EMTP simulations. The higher fmax is, the larger the order of the fitted functions. Therefore, there are computational advantages to keeping fmax reasonably low (e.g., 200-300 kHz is probably sufficient for most transient simulations). Note even if fmax in the measured data is not very high, the asymptotic behaviour of the transformer at high frequencies is predominantly capacitive, and it would still be properly reproduced by the HFT model at frequencies beyond fmax. This will be discussed in more detail later on.

- > **Clean-up noisy data and possible scale mismatches.** Although this type of manipulation is best done before data is given to FDBFIT, the program provides some simple tools for frequency scale synchronization and stray peak removal.
- > **Build the raw-data nodal admittance matrix.** By default, FDBFIT builds the nodal admittance matrix assuming block-symmetry, in order to create zero and positive sequence multi-port pi-representations of the transformer as seen at its terminals. The current version of FDBFIT cannot describe a delta-connected winding from its terminals unless the measurements are made on an open delta, and the delta connection is made implicitly in the EMTP via node name assignment.

**Read power frequency model and subtract low frequency response.** Experience in the measurement and fitting of transformer measured data suggests that it is very difficult to obtain accurate measurements at high frequencies (hundreds of kHz) and at 50 or 60Hz with the same instrumentation. For example, at low frequencies, Y is almost purely imaginary and small errors in the measurements can result in large errors in the discrimination of the real and imaginary parts of Y. It is then advantageous to generate an HFT model that "rides on top" of one of the standard low frequency models such as BCTRAN. FDBFIT can read any of the EMTP power frequency transformer models that can be represented as a series impedance matrix, such as TRELEG, or as R and  $L^{-1}$  matrices, such as BCTRAN, and TOPMAG. FDBFIT will then fit the frequency response of the transformer after the response of the power frequency model has been subtracted (see Figure 2-1 and Figure 2-2).

- > **Selecting the fitting method.** There are three fitting options:
  - Fit each element of Y (not supported in this release of the EMTP).
  - Fit the branches of the equivalent multi-phase pi-circuit in zero and positive sequence (can have accuracy problems)
  - Fit the elements of Y and then combine the approximations to produce the branches of the equivalent multi-phase pi-circuit in zero and positive sequence (default). Fitting by elements is the default option since it appears to produce better results even if the raw measurements show some inconsistencies.
- > **Evaluate RC branch for high frequency asymptotic behaviour.** This process is fairly accurate and the optimization process seldom changes the initial estimate significantly.
- > **Determine fcut, assign complex conjugate poles and optimise.** As illustrated in Figure 2-3, the functions to be fitted are divided in two zones (from power frequency to fcut, and from fcut to fmax).

The transition frequency  $f_{cut}$  is normally determined automatically by FDBFIT. Typical values of  $f_{cut}$  for large hv transformers vary from 1 to 20 kHz. Ideally,  $f_{cut}$  should be equal to the power frequency (50 or 60 Hz). In practice there is always a certain amount of disagreement between the frequency scan measurements and the power frequency model obtained from nameplate data, and  $f_{cut}$  takes place at a higher frequency. Current thinking is that these discrepancies are mainly due to instrumentation accuracy. In any event, this leaves a region between power frequency and  $f_{cut}$  where the accuracy of the fitted function is not very high after the complex poles and zeros have been assigned and optimised. The selection criterion for  $f_{cut}$  is the frequency at which the first minimum in the real part of  $y(\omega)$  takes place. In some functions it has been found that the automatic selection criterion can choose a value of  $f_{cut}$  that is too low because of small measurement glitches. In such cases the user has the option to override the automatic selection and to provide the correct  $f_{cut}$ . Whether or not the region between  $f_{min}$  (minimum frequency) and  $f_{cut}$  should be fitted with extra real poles and zeros in order to improve accuracy will be discussed next.

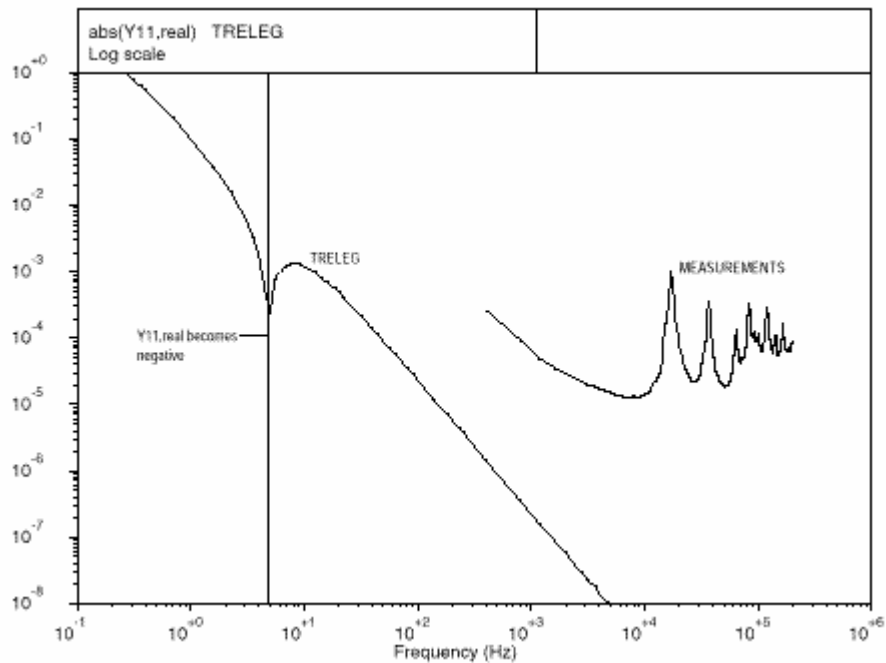


Figure 2-1 : Real part of  $|y_{11}|$  zero sequence. TRELEG and measured response

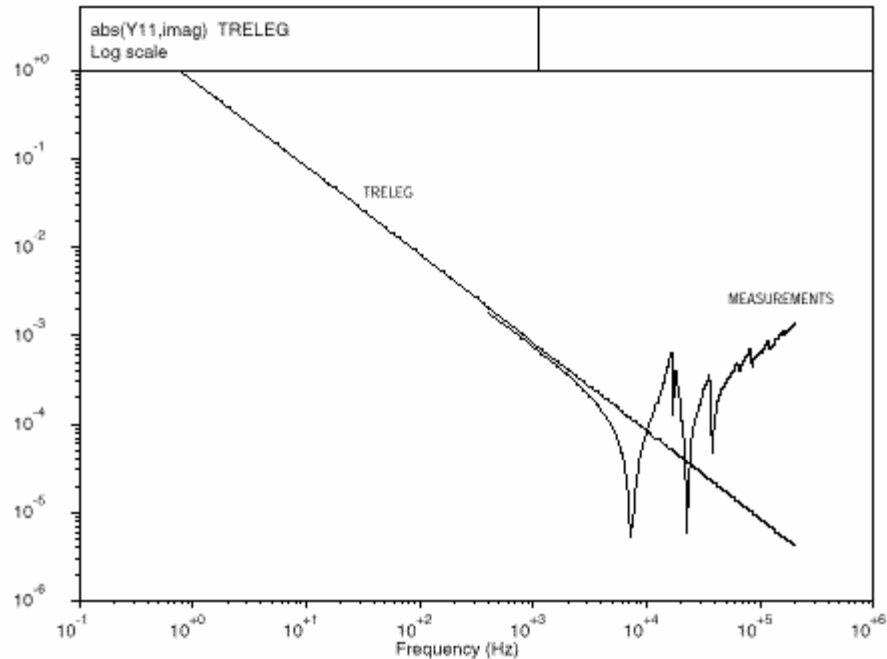


Figure 2–2 : Imaginary part of  $|y_{11}|$  zero sequence. TRELEG and measured response

- > **Assignment of real poles at low frequencies.** To improve the fitting between power frequency and  $f_{min}$ , it is possible to add extra real poles and zeros in this region. The addition of the extra poles and zeros is left as a user-defined option. In some instances, the addition of the extra poles result in very high accuracy at both high and low frequencies. On the other hand, there are instances where the addition of real poles can actually degrade overall accuracy. If the option to add real poles is selected, FDBFIT will automatically discard these real poles if they degrade the accuracy of the final function. The number of real poles to be added can also be controlled by the user. In any event, the addition of real poles increases the order of the fitted functions and slows down the optimisation process. Note that when real poles are assigned, the entire function including complex conjugate poles is optimised from  $f_{min}$  to  $f_{max}$ .
- > **Combine fitted functions and store in model data file.** The model data file can be referenced in the EMTP-EMTPWorks FDB device. External circuits, such as the magnetization branch must be assembled and connected separately.

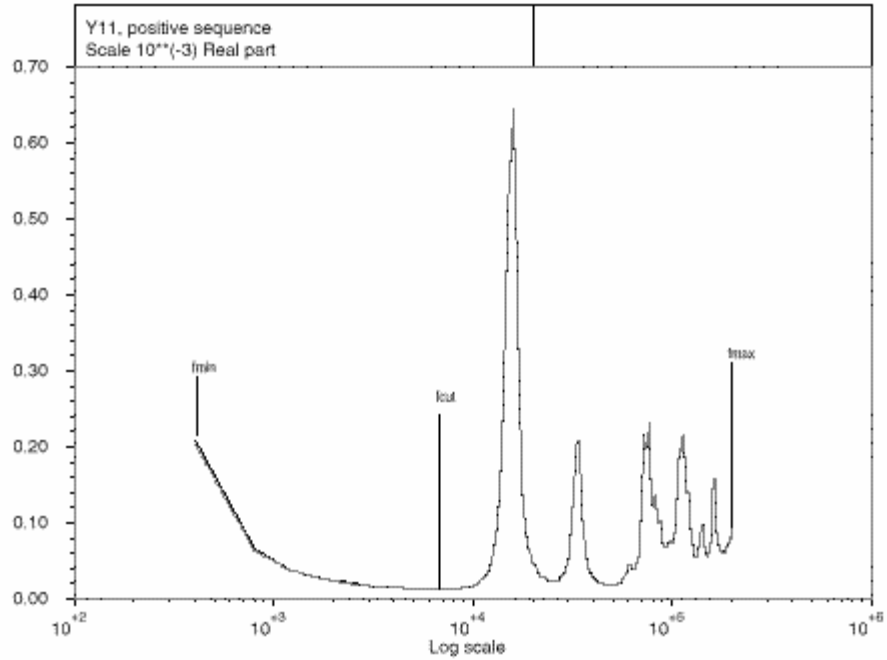


Figure 2-3 : Real part of  $y_{11}$  positive sequence. Raw data and difference function

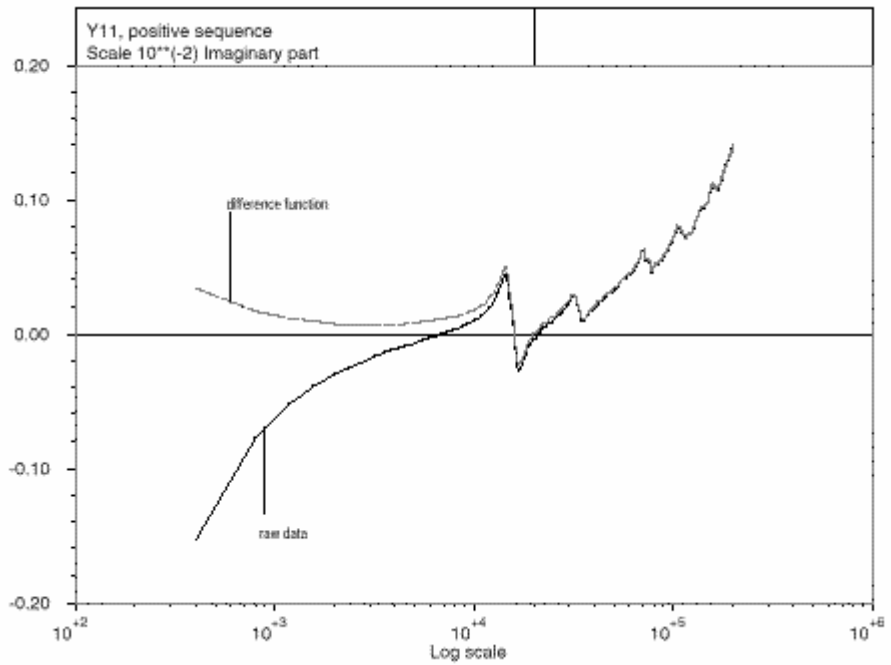


Figure 2-4 : Imaginary part of  $y_{11}$  positive sequence. Raw data and difference function

## 2.2 Non-Default Situations

The default settings and options of FDBFIT have been selected to produce stable and accurate fits of measured transformer data in most of the cases tested. It is possible, however, that a particular transformer

may require a certain amount of user intervention to produce a stable and accurate model. The following are a number of suggested courses of action to be tried if something does not work as expected:

- ❑ Check the validity and consistency of the raw data. If, for instance, the negative of an admittance function is recorded, the resulting HFT model would be inconsistent and unstable. Use a known set of admittance functions as reference to give an idea of what these functions should look like.
- ❑ Use a reasonable value of  $f_{max}$  based on dominant resonance peaks and the bandwidth required for the transient simulation. Past a certain frequency, resonance peaks have very little influence in most transient simulations, and it is the asymptotic behaviour of the transformer due to stray capacitances that dominates the behaviour at very high frequencies. In some cases, non-dominant high frequency resonance peaks beyond 200 kHz only result in a slower higher order model without a significant improvement in accuracy. Note that FDBFIT correctly models this asymptotic behaviour even if relatively low values of  $f_{max}$  are used. This provides some leeway in the selection of  $f_{max}$ .
- ❑ Given a choice, request high accuracy measurements in the lower frequency ranges near power frequency. Never try to splice measured functions from two different instruments.
- ❑ Use the option to override  $f_{cut}$  as a tool to improve the quality of the fit before the addition of real poles.
- ❑ The fewer the accessible terminals to be modelled, the better. For example, if the transformer neutrals are known to be solidly grounded during simulations, perform the measurements with the transformer grounded, rather than increasing the order of the model to be fitted and grounding the neutral by connecting it to a ground pin.

### 3 Running FDBFIT from EMTWorks

#### 3.1 FDBFIT device data

The FDBFIT device (“Data from FDBFIT”), Figure 3–1, serves to select data and options for FDBFIT and to run it.



**Figure 3–1 FDBFIT in EMTWorks**

Double-clicking on the above device opens data input tabs. The first tab allows the definition of the basic parameters for FDBFIT. The required data fields are self-explanatory and extra explanations are available in tool tips.

#### 3.1.1 Fitting Data

The second tab defines the control parameters of the fitting process. The “Fitting Data” table define the quality of the fitting for each element of the equivalent pi-circuit. Table 3-1 gives a detailed explanation for each data entry. None of the data fields can be left blank.

FCUT	This is the frequency that marks the beginning of the region to be fitted with complex conjugate poles of the $k$ th admittance function to be approximated. The program generally chooses it automatically, and it often coincides with the first as the first minimum of the real part of $y_{ij}$ . The
------	--



	user can override the automatic calculation of fcut by entering a non-zero value in this field.
RJECT	Tolerance used in the rejection criteria of measurement noise of the kth function to be approximated. There are instances where RJECT can be used to eliminate small spurious peaks that are probably not legitimate resonance peaks. Enter 0 to get the default for RJECT of 0.03.
FLAST	Maximum frequency to be considered in the fitting process of the kth admittance function to be approximated. Can be given the value 0 if it is assumed to be the frequency of the last measured data point.
TOLTER	Tolerance for one of the termination criteria of the optimization process of the kth admittance function to be approximated. TOLTER is the quotient between the least squares errors calculated between consecutive iteration steps. A larger value of TOLTER (e.g., 1.015) will speed up the fitting process substantially. Default recommended value is 1.001.
NRP	Number of real poles to be added to the kth fitted function. These real poles are in addition to the real pole determined by the RC branch. Default recommended value is 4.
ADDRP	Flag to control the addition of real poles to the of the kth fitted function. This will add real poles in addition to the complex conjugate ones. Select "Yes" to add real poles.
OPTRP	If not checked ("Yes"), the real poles of the kth fitted function will not be optimized. Check this box to optimize the function including real poles.
CHOOSE	Valid only if ADDRP is checked. If choose is checked, FDFIT will optimize the kth fitted function with added real poles, but it will only retain the solution with the lowest least squares error. The solution retained may not even include real poles. If choose is not checked FDBFIT will do exactly as the user asks with the different combinations of control flags

**Table 3-1 : Fitting Control parameters**

The node names for the terminals of the transformer are specified in the "Node Names" table. If the low frequency response is to be subtracted from the data, the node names should follow the ones specified in the power frequency data file. See also the tool tip associated to this table for more info on node names.

### 3.1.2 Measured data file names

The data files of measured data are specified in the "Measured data file names" table. A total of  $m$  file names are required, where  $m = nphases * nwind * (nwind * nphases + 1) / 2$ , with  $nphases$  being the number of phases and  $nwind$  being the number of windings. Each file name corresponds to the measured data of one element of  $\mathbf{Y}$ . The following rules apply:

- ❑ Only enter the file name, not its extension. For example, if file "file1.dat" contains the desired data, only enter "file1". File extension for measurement data files is always assumed to be ".dat".
- ❑ The files must be located below the current design directory.
- ❑ File names must be entered in the same order as the elements of  $\mathbf{Y}$  they represent, by columns, and assuming that  $\mathbf{Y}$  is upper triangular. For example, for a 4x4  $\mathbf{Y}$  matrix

file-Y11  
file-Y12  
file-Y22  
file-Y13  
file-Y23  
file-Y33  
file-Y14  
file-Y24  
file-Y34  
file-Y44

- ❑ The same file name may be used for more than one matrix element. For example, the following file list would represent a 4x4 balanced matrix

file-ys  
file-ym  
file-ys  
file-ym  
file-ym  
file-ys  
file-ym  
file-ym  
file-ym  
file-ys

- Comment cards are allowed inside the measurement data file.

Data is read in free format. Expected data structure is the following:

Frequency	Real{Y <sub>i,j</sub> }	Imaginary{Y <sub>i,j</sub> }
-----------	-------------------------	------------------------------

For example, the file corresponding to the element y<sub>11</sub> might look as shown below:

```
C FREQUENCY RESPONSE TEST ON TS 215.5/44KV T8 TRANSFORMER
C TEST ON TERMINAL H1H1
C
Frequency          Real{Yi, j}        Imaginary{Yi, j}
0.4000000000E+03  0.2278124593E-03  -0.1620969735E-02
0.8000000000E+03  0.7271661889E-04  -0.8228833321E-03
0.1200000000E+04  0.4186973456E-04  -0.5416742060E-03
0.1600000000E+04  0.3082307740E-04  -0.3978578316E-03
0.2000000000E+04  0.2523938929E-04  -0.3123318602E-03
0.2400000000E+04  0.2202727774E-04  -0.2522802097E-03
0.2800000000E+04  0.1981685637E-04  -0.2065615990E-03
...
```

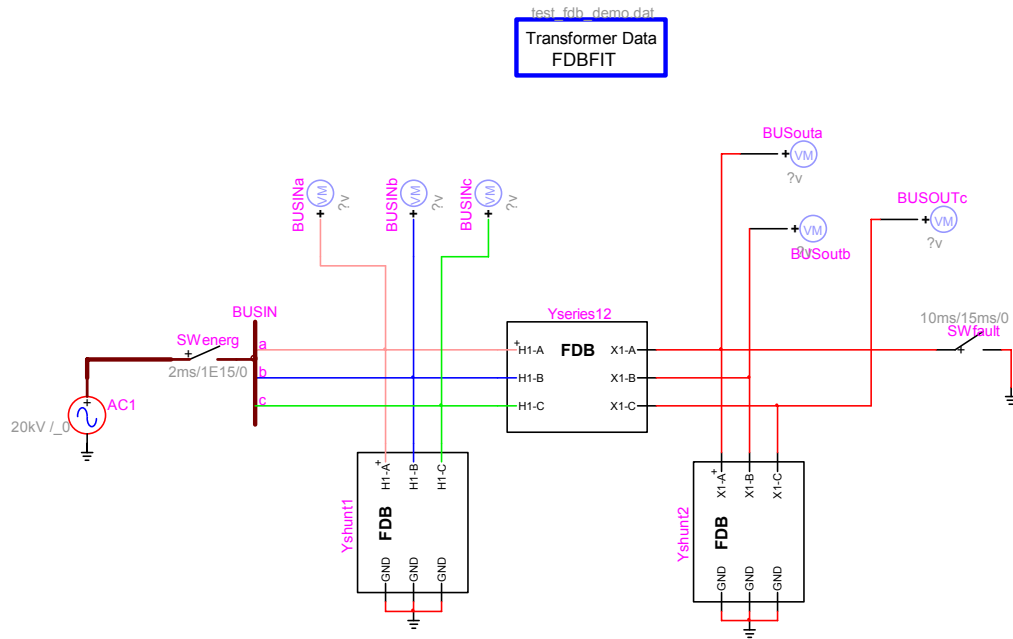
When the HFT model is modeled as an add-on module of the power frequency transformer model (default option), it is necessary to subtract the frequency response of the power frequency model (e.g., BCTAN, TRELEG, or TOPMAG) from the measured data. This is specified with the model data file obtained from running cases created in these devices.

### 3.1.3 Magnetizing branch

The “Magnetizing Branch” table allows to consider the optional magnetizing branch cards that are often added to the power frequency transformer model. See also the tool tip associated to this table for more info on name rules. If no data entered in this table, it will be ignored by FDBFIT. Arbitrary connections are achieved through left (k) and right (m) nodes names.

### 3.2 Running a case

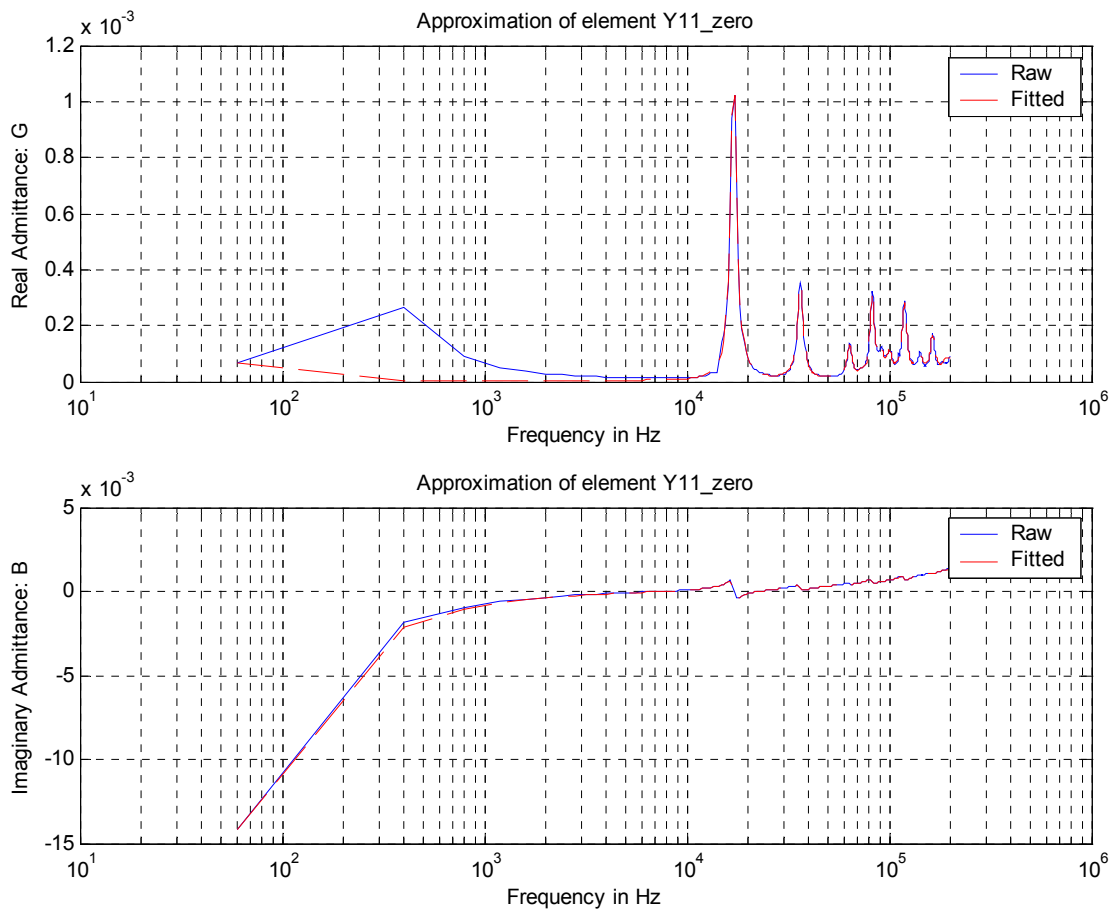
The example presented in this section is available in the Examples\fdbfit directory of EMTWork in the design file test\_fdb.ecf. The picture of this design is shown in Figure 3–2.



**Figure 3–2: Test design test\_fdb.ecf**

When all required data has been specified in the previous data panels it is possible to run the case using the panel “Run this case”. After choosing to run this case, the user must provide a valid Case Data File name. The Case Data File has the name *anymyfile.dat* and its contents are related to the data format required by the FDBFIT Calculation Function. According to the above test\_fdb.ecf design, the data used in the FDB branches is generated by FDBFIT in the files *anymyfile\_shunt(i).pun* and *anymyfile\_serie(ij).pun*. In this case, the FDB device named Yshunt1 is using the model data file test\_fdb\_demo\_shunt1.pun, Yshunt2 is using the file test\_fdb\_demo\_shunt2.pun and Yseries12 is using the file test\_fdb\_demo\_serie12.pun.

A plot data file for the visualization of the fitting results is saved in the file *anymyfile.m*. To plot results from a previously available run, it is necessary to right-click on the device and select the menu “Extras”. Figure 3–3 shows an example of the plotted data.



**Figure 3–3 : Fitted function compared to measured data**

### 3.3 FDBFIT model data file format

The generated files *anymyfile\_shunt(i).pun* and *anymyfile\_serie(ij).pun*, respect the self-explanatory format shown in Figure 3–4. Comment lines are started with the “C” character in the first column.

It is also feasible to enter a comment line for automatically identifying pin (node) names in the FDB device symbol, such as:

```
C BUS= H1-A ,X1-A ,H1-B ,X1-B ,H1-C ,X1-C ,
```

The code `BUS=` is followed by comma separated bus (node) names.

```

C FDB model file
n_phase=      3
balanced=     1
C Number of parallel branches for the zero sequence
  15
C Parameters for zero sequence
C values for each parallel branch of zero sequence are in order:
C           R           > <           L           ><           RL           >
C           C           > <           RC           >
0.1656071092168434E+04  0.3326502937989810E+00  0.0000000000000000E+00
0.1469841810029435E+07  0.1469841810029435E+07
0.3731597760747834E+05  0.1353565729213579E+01  0.0000000000000000E+00
-.1854947345732028E+07  -.1854947345732028E+07
...                [rest of data for zero sequence omitted]
C Number of parallel branches for the positive sequence follow
  12
C Parameters for positive sequence
-.1605980249544658E+05  0.4864506377100327E+00  0.0000000000000000E+00
0.1340221448823828E+06  0.1340221448823828E+06
0.2657516226627203E+04  0.8526136550421219E+00  0.0000000000000000E+00
...                [rest of data for positive sequence omitted]
C end of file

```

**Figure 3-4 : FDFIT output file format**