Program IFFT

By Antonio Carlos M. de Queiroz
Titular Professor
Universidade Federal do Rio de Janeiro

Documentation for version 1.7

The IFFT program analyses linear circuits, computing transfer functions in Laplace transform. From the transfer functions, in a ratio-of-polynomials form, the program can compute poles and zeros an plot frequency response graphs.

The program was developed as a tool for fast verification of continuous time filter designs. It is also useful for other purposes, as:

- Stability analysis by direct poles and zeros calculation.
- Studies about frequency compensation of amplifiers.
- Studies about compensation of operational amplifier GB and input and output impedance effects in filters.
- Studies about the effects of parasitic capacitances and resistances in filters.
- Monte Carlo analysis and study of the effect of component variations.

User Interface:

The present version uses a standard Windows interface.

Keyboard commands in the frequency response graph:

A/R - Reduces/increases the maximum frequency.
</> - Moves down/up the frequency range.
+/- - Reduces/increases the maximum gain.
G - Toggles the drawing of a grid.
F - Toggles the plotting of phase curves.
T - Toggles the plotting of group delay curves.
L - Frequency scale logarithmic/linear.
Vertical arrows - Move the gain curve.
Horizontal arrows - Move the cursor.
PgUp, PgDn - Move the cursor faster.
Return, Backspace - Change the selected gain curve.
Z, Space - Zoom out.

Keyboard commands in the poles and zeros graph:

Arrows- Move the plot.
+/- - zoom in/out.
Z, Space – Zoom out.

Mouse commands:

In the two graphs, the mouse moves a cursor with the right button (or Shift-Left button) and controls a zoom function.

Input:

The program begins by opening the frequency response window and the directory window. In the directory window, a netlist file name can be selected.

The program accepts as input a circuit description file in the format generated by the EdFil editor program. It is a text file with the following format. All the element values in Ohms, Farads, Henrys, Amperes, and Volts:

First line: Number of nodes, excluding the reference (node 0).
Following lines: One element per line:

- Resistor: R<name> <node1> <node2> <R>
- Inductor: L<name> <node1> <node2> <L>
- Capacitor: C<name> <node1> <node2> <C>
- Transformer with inductances L11, L22, and coupling k12: K<name> <node1+> <node1-> <node2+> <node2-> <L11> <L22> <k12>
- Gyrator, with gyration resistance Rg: Y<name> <node1+> <node1-> <node2+> <node2-> <Rg>
- Voltage controlled voltage source (voltage amplifier): E<name> <nodeV+> <nodeV-> <nodev+> <nodev-> <Av>
- Current controlled current source (current amplifier): F<name> <nodeI+> <nodeI-> <nodei+> <nodei-> <Ai>
- Voltage controlled current source (transconductor): G<name> <nodeI+> <nodeI-> <nodeV+> <nodeV-> <Gm> [<gc>].

- Current controlled voltage source (transresistor):

- H<name> <nodeV+> <nodeV-> <nodei+> <nodei->
- Operational amplifier, with gain GB/s (GB in rad/s) and output resistance Ro: A<name> <nodev-> <nodev+> <nodeV+> <GB> <Ro>
- Or: A<name> <nodeV+> <nodeV-> <nodev-> <nodev+> <GB> <Ro>
- Ideal operational amplifier: nullator-norator model: O<name> <nodev-> <nodev+> <nodeV+>
- Or: O<name> <nodeV+> <nodeV-> <nodev-> <nodev+>
- Bipolar transistor: hybrid h model Q<name> <nodeC> <nodeB> <nodeE> <hfe> <hie> <hre> <hoe>
- MOS transistor: M<name> <nodeD> <nodeG> <nodeS> <Gm> <Gds>
- Current source: I<name> <node+> <node-> <I>
- Voltage source: V<name> <node+> <node-> <V>
- Voltage source with series resistor (V=1 is the default): Z<name> <node-> <node+> <Rs> [<V>]
- Comment: *<comment>

Other elements can be built with combinations of these elements.
Note that the transistor models do not include capacitances, but they can be added explicitly. Note also that the current-controlled sources have the current in a short-circuit branch as input.

Two forms for the operational amplifiers are accepted. The forms with 4 nodes have floating input and output. The forms with 3 terminals have grounded output. The order of the nodes is different to keep compatibility with older versions.
<gc> for the transconductor is an optional common-mode transconductance. It produces identical currents at both outputs, coming from the ground, with value gc*(V+ + V-)/2, added to the normal current. This is useful for simulation of balanced transconductors.

Analysis method:

The program operates with a version of the modified nodal analysis method, so the currents in voltage sources and in controlling short circuits are computed. Their values can be accessed as "nodal voltages" in nodes above the last node of the circuit.
The program lists while reading the netlist what are the node numbers assigned to each current. Ideal operational amplifiers are treated as nullator-norator pairs in a more efficient way, and their output currents are not computed.

When the Analyze button in the analysis parameters window is pressed, the circuit is analyzed in a number of frequencies equally spaced over the unit circle. From the obtained values, the numerator and denominator polynomials in "s" of the transfer functions in Laplace transform are recovered by Fast Fourier Transforms (FFT), and listed in the messages window. The complete analysis is made only when changes in the analysis parameters invalidate anterior analysis results.

The method is precise, but can present numerical problems if the circuit is not adequately normalized. In the analysis parameters window there are some fields that can be initialized with values that control the precision of the analysis. The default values are usually convenient for normalized circuits. The meanings of the items in this window are:

- Frequency normalization factor: Factor used to normalize the circuit in frequency. The ideal value is the geometrical mean of the circuit poles and zeros. After the analysis, the program lists what would be the ideal values for the numerator and denominator, if applicable. For usual circuits, the used value must be within a decade of the ideal values for reliable results. Some variations on the factor to verify if the program insists in the same ideal values is recommended when in the analysis of high order circuits with poles and zeros spread over several magnitude orders.
- Impedance normalization factor: Factor used to normalize the circuit in impedance. The same observations apply.
- Estimated numerator and denominator degrees: The program computes the maximum possible denominator degree for voltage transfer functions in the circuit based on a normal tree, listed after the netlist in the messages window. This value is correct for non-pathological circuits, and the program will not accept as valid any analysis resulting in a different value. The estimated numerator degree must be set by the user if the option of forcing degrees is used. In circuits where the numerator degree is higher than the denominator degree, the numerator degree must be set to the correct value, or the analysis may result wrong.
- Force degrees option, and numerator and denominator dispersion factors: The analysis method produces polynomials of degrees that can be higher than the correct ones, and so must be cleaned of numerical residues. The

dispersion factors can be used for this purpose. They define the maximum allowable ratio among polynomial coefficients (normalized), and are used to eliminate too small coefficients from the polynomials. The default settings are usually enough for normalized circuits. In some cases, it is more practical to directly force the numerator and denominator degrees (to the estimated values, that must be known), as in the analysis of complex amplifiers with many natural frequencies widely spaced, or circuits with nonideal operational amplifiers.

References:

Frequency response plots, and poles and zeros plots, can be compared with saved references. The references can be obtained from the present analysis or from saved files. It is possible to analyze a circuit, save the results as a reference, edit some of the component values, reanalyze, and plot the new results compared with the saved ones. The references are enumerated from 1 to a maximum. The references are also used in the value sweep and Monte Carlo analysis.

Edition:

The circuit values can be edited in the edit window. The edition causes the invalidation of the last analysis, and the circuit must be reanalyzed. The edited circuit cannot be saved.

Sweep:

There is an option of sweeping a value over a specified range (in the edit window). The results are saved as references, and subsequent plotting commands produce families of curves showing the results. The sweep range can be specified by absolute limits or percentages of the normal value. The sweep can be linear or logarithmic. The original circuit is retained, but must be reanalyzed if its curves are to appear in the plots. All the references are cleared before a sweep.

Monte Carlo analysis:

A set of circuits with element values within specified tolerances is automatically generated and analyzed. All the results are stored as references, as in the sweep analysis. The variabilities of certain groups of elements can be specified independently, and uniform or Gaussian distributions can be used. In the Gaussian distribution, the specified variabilities correspond to three standard deviations. There is a menu item in the references submenu that can be used to save all the frequency responses, poles, and zeros resulting from the analysis to files. The files can be used for comparison with the results obtained by sensitivity analysis with the SENSI program, by the use of the CPSENSI and CPSPZ programs. (See the SENSI package at the site http://www.coe.ufrj.br/~acmq.)

Output files:

All the results can be saved, in separate text files according to the type of result. The names are given automatically. It is also possible to save all the messages in the message window as a file. The polynomials and roots are saved in normalized form.
The normalization factor used is also saved at the end of the files. The reference poles and zeros are saved unnormalized.

Observations:

The program can have problems in finding poles and zeros in some high-order circuits. In these cases it will lower the tolerance of the roots until it can solve for them, and the final results may be inaccurate. It is important to turn on the automatic pole and zero computation (in the analysis window) only when the frequency normalization factor is correctly adjusted, otherwise an error may occur. For comparison of results, the program can use two different methods for computing polynomial roots (set in the root parameters window). The default method is the most reliable, and slower, one. The poles and zeros option in the main menu causes the computation of poles and zeros of the present function and of all the references, if they were not already computed. The changing of a parameter in the roots parameters window causes the invalidation of the poles and zeros of the present function, but not the ones of the references. This is to allow easy comparison of results.

The IFFT program is in continuous evolution since its first version, written in BASIC for an HP-75 computer in 1984. The program was ported to a PC, rewritten in Turbo Pascal, and in 1992 rewritten with the new graphical interface, also in Turbo Pascal. The Windows version was written in 2012.

The EdFil editor program:

The EdFil editor can be used to generate the input net-list file for the IFFT program, and also for other programs, even SPICE. It was written after the conclusion that it is virtually impossible to write a textual netlist for a non-trivial circuit without errors. Its use is recommended for any circuit analysis.

Contacting the author:

Comments and doubts about the programs shall be sent to the author:

Dr. Antonio Carlos M. de Queiroz
Programa de Engenharia Elétrica - COPPE
Universidade Federal do Rio de Janeiro
CP 68504
21945-970 Rio de Janeiro, RJ, Brazil
E-mail: acmq@ufrj.br
http://www.coe.ufrj.br/~acmq

Files on the distribution compressed file:

IFFTW.EXE: Windows IFFT program.
EDFILW.EXE: Windows version of the Edfil editor.
*.CIR: Circuit files em EdFil format.

*.NET: NetList files.
*.VAL: Value list files for EdFil.
*.PDF: Documentation.
The ANATRAN program is included too. It computes transient responses from the transfer functions calculated by the program, and others.

Licensing

Users are encouraged to send to the author an e-mail message telling where the program is being used, and for what purpose.

Version information

Version 1.7: Windows version.

Version 1.6: Transformer with letter "K".
Version 1.5b: The nonideal operational amplifier may have four nodes too.
Version 1.5a: The transconductor has a common-mode transconductance, the operational amplifier may have 4 nodes, and comments are accepted.
Version 1.4: The transformer is treated by the coupling coefficient.
Version 1.3b: A bug in the inclusion of amplifiers in the normal tree was corrected.
Version 1.3a: A bug in Gds in the MOS transistor was corrected.
Version 1.3: Complexity order estimation improved by a more general algorithm for the generation of the normal tree. The program will not anymore underestimate the order. Monte Carlo analysis introduced.

ACMQ - 7 March 2001
Last update: 13/05/2019