Loop Gain Simulation

Loop Gain or Return Ratio?

When I talk about loop gain on this page, it means the same as the term return ratio that some other people prefer to use. Return ratio is the original term defined by Hendrik W. Bode in his book "Network Analysis and Feedback Amplifier Design" published in 1945, but it is not used by very many people today. I use the term loop gain instead, like R. David Middlebrook and probably most electronic engineers.

Some people make a difference between return ratio and loop gain and define the term loop gain in a different way. For example, Michael Tian and his colleagues define loop gain as the negative return ratio in their article "Striving for Small-Signal Stability". Paul J. Hurst uses the term loop gain for the result of a two-port analysis in his articles "Exact Simulation of Feedback Circuit Parameters" and "A Comparison of Two Approaches to Feedback Circuit Analysis" and in the book "Analysis and Design of Analog Integrated Circuits".

Analysis of Linearized Circuit

The methods presented on this page use the small-signal ac analysis of SPICE and similar circuit simulators. It is important to remember that for this analysis, the circuit is linearized around its operating point. This means that the results are only valid if the linearized circuit is a good approximation of the actual circuit. It is always a good idea to verify the results of a loop gain simulation with a transient analysis of the step response. Both up and down steps should be examined, the step at the input should have the maximum amplitude and slew rate that the circuit might encounter in its application, and the circuit response should show no unwanted oscillation or ringing.

For circuits with several different states like switched-capacitor circuits or switched-mode power supplies, ac analysis is not valid. A method based on transient analysis that can be used for these circuits is presented in the FRA directory of the educational examples in the LTspice distribution. SIMPLIS has a special ac analysis for switched-mode power supplies. The examples for these two simulators show how to simulate voltage loop gain. A third possibility for simulating the loop gain of these circuits is the pstb analysis of SpectreRF from Cadence Design Systems, which uses an extended version of Tian's Method.

Why Opening the Loop is a Bad Idea

If you open the loop in order to simulate loop gain, you create two problems:

- The dc operating points on both sides of the opening are usually different.
- The small-signal ac impedances seen on both sides are different from the closed-loop case.

The first problem is often solved by closing the loop again with a large inductor and by injecting the signal with a large capacitor. The second problem can be approached by adding a replica of the circuit on the other side of the opening. However, in many cases this only provides an approximation to the actual closed-loop impedance. In general, opening the loop in order to simulate loop gain is a rather inexact and error-prone method.

Because of these problems, alternative methods have been developed for loop gain simulation. With these methods, ideal ac voltage and current sources are inserted into the circuit in such a way that neither the dc operating points nor the small-signal ac impedances are changed. In order to obtain the final result, these methods combine the results of two or three simulation runs in which different sources are active. I will present these methods in the following sections. (Under certain conditions, voltage loop gain, which only requires a single simulation run, will also give very similar results.)

Middlebrook's Method

The method generally known as "Middlebrook's Method" was published by R. David Middlebrook in 1975 in his article "Measurement of Loop Gain in Feedback Systems", which appe

International Journal of Electronics (volume 38, no. 4, pages 485-512, April 1975).

Middlebrook developed this method using a simplified model which did not take into account backward transmission through the loop. This limitation was later removed by the other two methods. Nevertheless, the method usually provides pretty accurate results and, like the other two methods, will always tell you correctly whether a circuit is stable for small pertubations or not (use the Nyquist stability criterion in case of doubt).

The best description of this method currently available on the web is in a Newsletter from Spectrum Software. The method is also presented in the LoopGain educational example that comes with the free LTspice circuit simulator.

Tian's Method

This method was developed by Michael Tian and his colleagues from Cadence Design Systems and was published in 2001 in their article "Striving for Small-Signal Stability", which appeared in the *IEEE Circuits and Devices Magazine* (volume 17, no. 1, pages 31-41, January 2001).

The method is used by Cadence (with a slightly unusual sign convention) in the stability analysis (stb) of their Spectre circuit simulator and, by extension, in the periodic stability analysis (pstb) of SpectreRF. Its advantage over the other two methods presented here is that it is symmetrical, so that the orientation of the probe components with respect to the loop does not matter.

My implementation of this method for LTspice is available in the archive LoopGain_Probe.zip. It also contains an example that shows how to measure phase margin and gain margin. The documentation is included as a comment in the circuit schematics. Some additional discussion about this implementation can be found in the thread starting at http://tech.groups.yahoo.com/group/LTspice/message/2482 (free registration is required for access to Yahoo Groups). The LoopGain2 educational example from the LTspice distribution presents a simplified version of this implementation.

Loop Gain of Differential Circuits

Like the other two methods presented here, Tian's method can only be applied to circuits with exactly one single-ended loop (see Tian's article mentioned above for details). In order to extend this method to differential circuits, connect two ideal baluns back-to-back with their differential-mode and common-mode ports and insert this combination into your circuit. Now, you can insert the probe components into the path between the differential-mode ports in order to simulate the differential-mode loop gain or between the common-mode ports in order to simulate the loop gain of the common-mode regulation. In the Spectre circuit simulator from Cadence, the diffstbprobe cell from the analogLib library simplifies this setup.

Middlebrook's General Feedback Theorem

The General Feedback Theorem (GFT) developed by R. David Middlebrook allows a more complete analysis of the feedback circuit that also includes the closed-loop gain. His article "The General Feedback Theorem: A Final Solution for Feedback Systems" was published in 2006 by the *IEEE Microwave Magazine* (volume 7, no. 2, pages 50-63, April 2006). A more detailed description is available in the "GFT Template User's Manual", which he wrote for an implementation of the method for the ICAP/4 circuit simulator from Intusoft. The basis for the development of the General Feedback Theorem was the Two Extra Element Theorem (2EET). More extensive background information can be found in Middlebrook's Design-Oriented Analysis Rules and Tools.

The GFT shows that for a general feedback circuit, there usually exists an additional path H_0 in the equivalent circuit that bypasses the loop and goes directly from the input to the output. For some circuits, this bypass path can modify the closed-loop transfer function quite considerably. For example, it can cause peaking in the closed-loop transfer function although the loop gain has a very large phase margin. One such case is presented in example 1 of the "GFT Template User's Manual".

The GFT is very closely related to the model from the Wikipedia article about the Asymptotic Gain Model. The new aspects of the GFT with respect to this model are described in R. David Middlebrook's answer to my question about this relationship.

My implementation of the GFT for LTspice is available in the archive GFT_LTspice.zip. It is much less complex than the one for ICAP/4 and, like my implementation of Tian's method, can be adapted for any SPICE simulator that supports parameter stepping with subsequent calculations using results from different steps. If the simulator does not support this, you can still use these methods by making two or three identical copies of your circuit (one for each of the steps). Extending my GFT implementation, the INTEC_design group at Ghent University has realized an implementation of the General Network Theorem in Cadence Virtuoso.

The documentation for my GFT implementation is in the readme.txt file from the archive, which for your convenience is also presented in the following section of this page. You can find some additional information about the GFT and loop gain in general in my discussion with R. David Middlebrook that starts at http://groups.yahoo.com/group/Design-Oriented_Analysis_D-OA/message/40.

readme.txt file from GFT_LTspice.zip

What do these examples show?

These examples show how the quantities of the General Feedback Theorem (GFT) developed by R. David Middlebrook can be simulated in LTspice. The GFT is explained in the GFT manual that is available at http://www.intusoft.com/gft.htm and in the article "The general feedback theorem: a final solution for feedback systems" that was published in the IEEE Microwave Magazine and is available at http://resolver.caltech.edu/CaltechAUTHORS:MIDieeemm06. The examples in this archive were taken from these two documents. Additional information about the GFT can also be found at R. David Middlebrook's website at http://www.rdmiddlebrook.com.

What are the advantages of the General Feedback Theorem?

Unlike other methods for calculating loop gain (see below), the GFT also gives you information about the closed-loop gain of the circuit. This can especially be useful when there is a significant direct transmission from the input to the output of the circuit, bypassing the loop. In this case, the loop gain alone often cannot explain the behavior of the circuit.

Such a situation can be found in example 1 of the GFT manual (see the corresponding circuit in the file manual1.asc). Here, the closed-loop gain H shows significant peaking although the loop gain T has a very large phase margin of 87 degrees and so cannot cause this behavior. In such a case, calculating the GFT transfer functions can help you better understand the circuit and also show you possible ways for improving it.

How do you run these examples?

First of all, install LTspice if you have not already done so. LTspice (also called SwitcherCAD III) can be downloaded for free from http://www.linear.com/designtools/software/. It is a full-featured SPICE simulator with very good performance and no arbitrary limitations. There is also a very active and helpful users' group at http://tech.groups.yahoo.com/group/LTspice/.

After you have installed LTspice, copy the file plot.defs from this archive to the directory where the LTspice executable scad3.exe was installed. (If there is already a plot.defs file in that directory, add its contents to it.) The plot.defs file contains the formulas used for evaluating the simulation results.

Next, open one of the .asc files and click the "Run" button. After the simulation has completed, the plot window will open automatically. The plot configuration is stored in the .plt file with the same base name as the .asc file. For some examples, there are additional .plt files that correspond to the figures in the GFT manual or the paper. They can be loaded with "Plot Settings -> Open Plot Settings File". (The plot window has to be active for the Plot Settings menu to be visible.)

How do these examples work?

Three simulations are run with different sources being active. This is controlled by the variable z, which takes the values 1, 0, and -1. (Parameter stepping in LTspice has to be monotonous, so an order like 0, 1, and -1 is not possible.) For z=1, only the voltage injection source Vz is active, for z=0, only the input source Vi (or Ii) is active, and for z=-1, only the current injection source Iz is active. (If you only want to do normal closed-loop simulations without the GFT, you can comment out the .step command and set z=0.)

The results from the three simulation runs are combined by the equations defined in the plot.defs file.

You can look at the plot.defs file with "Plot Settings -> Edit Plot Defs File" or by opening it with any text editor. The suffix @1 in the equations refers to the first run with z=1 (Vz active), the suffix @2 refers to the second run with z=0 (Vi or Ii active), and the suffix @3 refers to the third run with z=-1 (Iz active).

How were these equations obtained?

The basic idea for deriving these equations was to explicitly determine the relative amplitudes of the sources necessary to satisfy the different nulling conditions and then sum up their contributions to the results. Of course, this is a completely wrong approach if you are doing symbolic calculations on a circuit in order to obtain low-entropy expressions. Here, however, this approach has led to remarkably compact formulas for numerical evaluation that, while not really being low-entropy, also have a certain aesthetic value. The detailed derivation is available in the file GFT_SPICE.doc in this archive, thanks to Alberto Petrini who took my handwritten calculations and transformed them into a clear presentation.

What are the advantages of this method?

The biggest advantage of this method is the fact that you only need three simulations in order to obtain the results. It can be implemented in any SPICE simulator that supports parameter stepping with subsequent calculations using results from different steps. If the simulator does not support this, you can still use this method by making three identical copies of your circuit (one for each of the sources).

How do you set up your own simulations?

There are a number of points that have to be observed when you are setting up your own GFT simulations. The voltage-controlled voltage source Ey that senses the voltage Vy and the voltage source Viy that senses the current Iy have to be placed backwards from the injection point relative to the direction of loop transmission. The orientation of the injection sources Vz and Iz must be such that a positive voltage Vz causes a positive voltage V(y) and that a positive current Iz causes a positive current into the "+" terminal of the voltage source Viy (corresponding to a positive current I(Viy)).

The names of the voltage source Viy, of the node y for Vy and of the output node o are used in the equations and must not be changed (unless you also change the equations accordingly). In order to simplify the equations, I have used the fact that the amplitudes of the sources are exactly 1 when they are active, so this must not be changed either.

Of course, you also have to choose the correct injection point so that the results are meaningful. You should always check Hinf() first to see if it has the expected value. For further details, please refer to the GFT manual and the article.

The available quantities are defined in the plot.defs file. They are H(), Hinf(), H0(), T(), Tn(), Tp(), Dd(), Dn(), D0(), and Dp(). I could not use D() for the discrepancy factor D because the function D() is already used for the derivative by LTspice, so I chose Dd() instead. The quantities ending in _1 are for a configuration with a single injection, like the examples manual3a.asc and manual3b.asc.

If you have an output voltage that does not have the global ground as reference, you can use an ideal voltage-controlled voltage source (E element) like for Vy to measure the voltage. If you have an output current, you can use an ideal current-controlled voltage source (H element) to measure it so that you do not have to modify the equations. More exotic injection configurations like dual voltage injection or dual current injection can be realized in a similar way.

What about other definitions for loop gain?

A different method for simulating loop gain has been described in the article "Striving for Small-Signal Stability" by Michael Tian and others, which was published in the IEEE Circuits & Devices Magazine and is available at http://www.kenkundert.com/docs/cd2001-01.pdf. The differences between the GFT loop gain and Tian's loop gain can be most easily explained by looking at figure 7 and equation (3) from that article.

Figure 7 shows a feedback loop with voltage and current injection. You might recognize that the component parameters correspond to the Y parameters of the circuit if it is opened at the injection point, with Ye=Y11, k3=Y12, k1=Y21, and Yf=Y22. For greater clarity, I will use the Y parameters in the following discussion.

The loop gain T (Tian calls it return ratio) is given by equation (20) as (Y12+Y21)/(Y11+Y22). If you calculate the GFT loop gain for the circuit of figure 7, you will find that it is Y21/(Y11+Y22+Y12). So, who is wrong? As it turns out, neither one is really wrong. This can be seen by looking at equation (3). The determinant Delta for this circuit is Y11+Y12+Y21+Y22. This means that both definitions satisfy equation (3), with x=Y21 for the GFT loop gain and x=Y21+Y12 for Tian's loop gain.

The variable x refers to the controlled source of the loop. This means that the GFT loop gain is more physical and is directly related to the closed-loop gain by the GFT equations. On the other hand, Tian's loop gain is symmetrical which means that it is independent of the orientation of the loop. Because both definitions satisfy equation (3), both will tell you correctly if a circuit is stable or not.

In practice, the results of both definitions are generally very similar. I have defined the function Tt() for Tian's loop gain and the .plt files ...t.plt for comparison. A real difference can only be observed for the article2.asc circuit. Even here, the differences only appear at very high frequencies where the loop gain is far below 0 dB.

In an article published in 1975, R. David Middlebrook presented an early method for measuring loop gain. For its development, he was using a circuit model with Y12=0. Although this method has been superseded by the GFT and by Tian's method, it is still widely known as "Middlebrook's method" among engineers (most of whom probably have never heard about the later methods) and is being regarded as an advanced method for simulating loop gain.

If one applies this method to the circuit of figure 7, one obtains (Y21-Y12)/(Y11+Y22+2*Y12) for the loop gain. Comparing this expression with equation (3), one finds that it also satisfies this equation, with x=Y21-Y12. So, the good news is that even the older method will give the correct result for the stability of the circuit. In general, the loop gain calculated by this method is also very similar to the GFT loop gain.

It can be shown that the loop gain is independent of the choice of the injection point for all three methods discussed so far (as long as there is only one loop and opening the circuit at the injection point breaks the loop completely, which means that there are no parallel paths). If Y12=0 (case with no backward transmission), all three methods give the same loop gain Y21/(Y11+Y22).

Some simple simulation setups only use the voltage loop gain Tv=Vy/Vx for voltage injection or the current loop gain Ti=Iy/Ix for current injection. It is interesting to note that these simple definitions also satisfy equation (3), with x=Y21+Y11 for Tv and x=Y21+Y22 for Ti. So, they will also tell you correctly if the circuit is stable or not. However, both these definitions are dependent on the injection point and can be very different from the GFT loop gain, so great care has to be taken when one wants to evaluate properties like phase margin from these definitions.

The derivation of the well-known relationships between phase margin and frequency response or step response uses the equation H=Hinf*T/(1+T) as its starting point. Additional conditions are that Hinf is constant and that T has two widely separated poles and no zeros. The equation is always satisfied by the pseudo loop gain Tp, because by definition H=Hinf*Tp/(1+Tp). The pseudo loop gain Tp equals the GFT loop gain T if there is no transmission through the direct path that bypasses the loop, which is equivalent to H0=0, Dn=1, or Tn=infinity.

Some time ago, I implemented Tian's loop gain in an LTspice example that also contains Middlebrook's older method. It is available at http://tech.groups.yahoo.com/group/LTspice/files/%20Examples/Educational/LoopGain_Probe/LoopGa
Some discussions regarding this circuit can be found in the LTspice users' group in the thread starting with message 2482 at http://tech.groups.yahoo.com/group/LTspice/message/2482. A stripped-down version of the example is available as the LoopGain2 educational example from the LTspice distribution.

How can you contact me?

If you have questions, remarks, suggestions for improvement etc., you can contact me (Frank Wiedmann) by using the contact form at http://sites.google.com/site/frankwiedmann/contact. I apologize for not giving you my email address here, but I already receive enough spam messages and do not want this number to increase any further.