Prelab 5 - Open-Loop Gain (G_{ol})

Introduction

In this lab exercise, you will write and test the code to perform the open-loop gain test. We will be using the nulling amp to perform this measurement since it is much easier to predict the output voltage of the DUT. We will be using the QMS for our voltmeter again.

Prelab Assignment

a) Print out a copy of the DIB schematic for each site. Show all paths necessary to perform this test. In this manner, you will determine which relays must be closed and which resources you will use. Label the resistors in the same manner that we did in class (e.g. R1, RF, RA, and RB).
b) Calculate the expected worst case gained-up offset voltage (e.g. the QMS measurement) so that you know what range to set your QMS at in lab.
c) Will you need to add any more instrument groups or #defines? Explain your answer.
d) Add to your test plan the Gol test. (Note: in a real design solution, you would choose either the nulling amp or the false summing junction to perform Gol and Vos. As a result, the DIB would be significantly simplified and thereby cheaper. However, in this learning environment, I am trying to show you multiple methods.)

Lab 5 - G_{ol} Test

Write the test code for the G_{ol} test

Reminders of the key coding elements are shown below. Please refer to labs 1, 2, and 4 for details.

1. Set up any additional instrument groups. Some functions will require no additions.
2. Create a new function.
3. Set up the datasheet file.
4. Write the G_{ol} test. Recall that you must save the data into a datastructure immediately after the measurement or the measurement will be lost. Perform the proper math and save in a new array. Make sure you scale your results to match the values in your datasheet file.
5. You may want to write a debug routine to see your measurement values. This will help you in debugging your routine on the ATE.
6. Don't forget to modify TestCompletion and FailSite if necessary.
7. Compile, build, and run your program. Fix any errors.
8. In this lab we will test for site-to-site compatibility.
9. Backup your program onto your memory stick and let the professor know that you are ready to go to the tester.
10. First, run your test empty socket and debug any final errors. Then, test both sites simultaneously. Debug any errors. Once both tests pass, play with the accuracy/time tradeoff

parameters. Swap your chips. How consistent are your measurements from site to site? Record your results.

11. If you made any necessary changes to your code, make sure you back up the new version and restore it onto your computer for next week.

Write-up your results

Record your test results and your test time. Turn in your final code, test results, and a discussion of the time/accuracy trade-off and site-to-site reproducibility.