

# ECEN 474/704 Lab 9: Two-Stage Miller Operational Amplifier

## Introduction

Operational amplifiers (op-amp) are essential components of analog system design. Integrated circuit design, as well as board level design, often uses operational amplifiers. This component is basically a high gain voltage amplifier used in many analog systems such as filters, regulators and function generators. This rudimentary device is also used to create buffers, logarithmic amplifiers and instrumentation amplifiers. Op-amps can also function as simple comparators. Knowledge of operational amplifier functionality and design is important in analog design.

The symbol for an operational amplifier is shown in Figure 9-1. The basic device has two inputs and a single output. A fully differential version of the op-amp has two outputs and is often used in high performance integrated circuit designs.

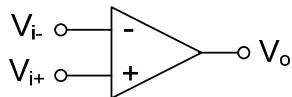


Figure 9-1: Operational Amplifier Symbol

The operational amplifier functions as a voltage amplifier. The relationship between the input and output voltage is given by:

$$V_o = A_{v0}(V_i^+ - V_i^-)$$

The amplifier has a high voltage gain ( $A_{v0} > 1000$  for CMOS op-amps). Due to the high gain, the linear region of an op-amp is very narrow, so the op-amp is commonly used in a negative feedback loop. Figure 9-2 illustrates the typical input-output characteristic for an operational amplifier used with and without feedback. The open loop (without feedback) plot shows the linear region is only a few millivolts wide. From Figure 9-2, the open loop input-output characteristic is clearly nonlinear. Notice the closed loop linear region consists of almost the entire input voltage range. The application of feedback reduces the non-linearity, but also reduces the voltage gain.

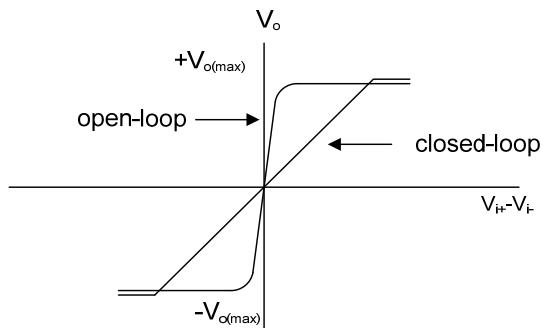
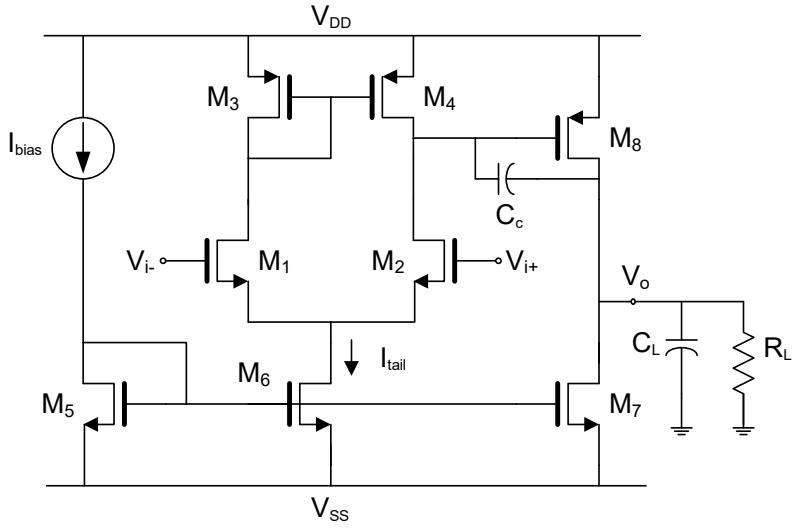


Figure 9-2: Input-Output Characteristic for an Op-amp

The simplest operational amplifier is the simple differential amplifier studied earlier. This amplifier can be improved by adding a second stage as an inverting amplifier with a current source load. The two-stage amplifier shown in Figure 9-3 is referred to as a Miller op-amp. This op-amp topology is presented for instructional purposes. In practice, various factors will determine what kind of amplifier topology is most appropriate for a given application.



**Figure 9-3:** Two-Stage Miller op-amp

The Miller op-amp has a low frequency gain of:

$$A_{v0} = G_m(R_{out} \parallel R_L)$$

The transconductance is given by:

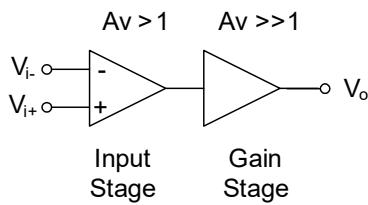
$$G_m = \frac{g_{m1,2}}{g_{o2} + g_{o4}} g_{m8}$$

The output resistance is given by:

$$R_{out} = r_{o7} \parallel r_{o8}$$

### Design Description

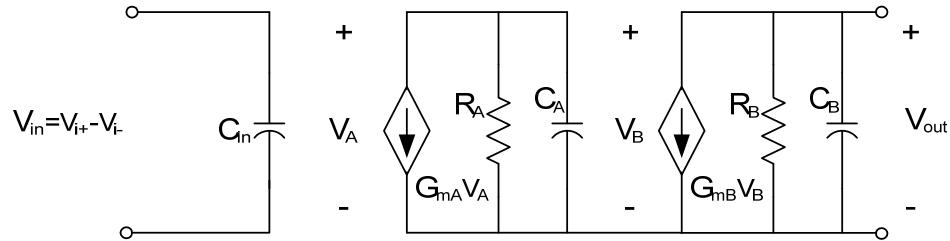
The two-stage amplifier can be modeled as a cascade of two amplifiers, as illustrated in Figure 9-4. The first stage is a differential amplifier, which produces an amplified version of the difference in input signals. This stage determines the CMRR, slew rate and other performance specifications determined by the differential amplifier.



**Figure 9-4:** The Two Stage Operational Amplifier Model

The second stage is an inverting amplifier. The purpose of this stage is to provide a large voltage gain. The gain stage and the input stage create two poles, which affect the stability of the feedback system. Usually, some form of compensation is required to assure the amplifier is stable at unity gain. Additional gain stages can be employed to increase the gain, but this degrades stability and requires complex compensation techniques.

The frequency response of an operational amplifier will be analyzed using the macro-model of the op-amp shown in Figure 9-5. The capacitor  $C_{in}$  models the input capacitance of the op-amp, which is mostly gate to source capacitance. The sub-circuit consisting of  $G_{mA}$ ,  $R_A$  and  $C_A$  model the gain and frequency response of the input stage. The capacitance  $C_A$  includes the input capacitance of the second stage and the output capacitance of the first stage. The components  $G_{mB}$ ,  $R_B$  and  $C_B$  model the second stage. The load capacitor and resistor are also included in  $R_B$  and  $C_B$ .

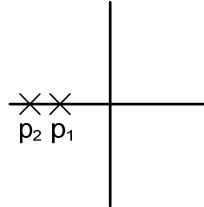


**Figure 9-5:** Operational Amplifier Macro-Model

The transfer function of the macro-model is given by:

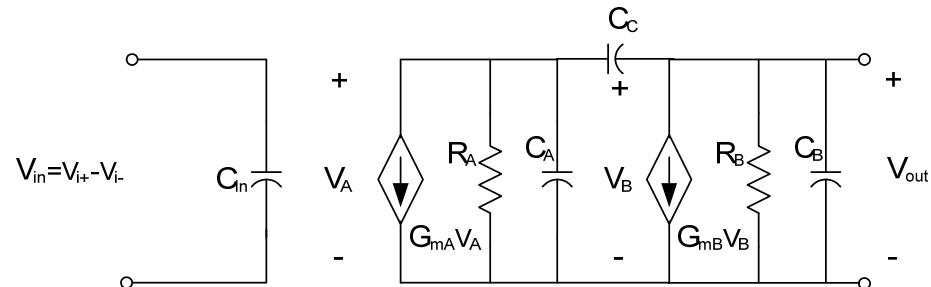
$$H(s) = \frac{A_{v0}}{\left(1 + \frac{s}{\omega_{p1}}\right)\left(1 + \frac{s}{\omega_{p2}}\right)} = \frac{G_{mA}G_{mB}R_AR_B}{(1 + sR_AC_A)(1 + sR_BC_B)}$$

This transfer function assumes zero source resistance. Notice the two poles are approximately equal. The capacitors  $C_A$  and  $C_B$  are dominated by gate to source capacitances, and  $R_A$  and  $R_B$  are the parallel connected small-signal drain to source resistances. The pole-zero plot of this transfer function is illustrated in Figure 9-6.



**Figure 9-6:** Pole-Zero Diagram for Uncompensated Op-amp

Due to the poles being located close together and the large DC gain, the system is unlikely to be stable in unity-gain feedback configuration, therefore some form of compensation is required. The modified macro-model shown in Figure 9-7 uses capacitor  $C_C$  to compensate the frequency response of the op-amp by splitting the two poles.



**Figure 9-7:** Operational Amplifier Macro-Model with Compensation Capacitor  $C_C$

Assuming  $R_A$  is large ( $R_A \approx R_B$  and  $R_A \gg 1/G_{mB}$ ) and  $C_A$  is small ( $C_B, C_C \gg C_A$ ), and using the results obtained from the inverting amplifier lab, the transfer function for the operational amplifier with the compensation capacitor is:

$$H_C(s) = \frac{A_{v0} \left(1 - \frac{s}{\omega_z}\right)}{\left(1 + \frac{s}{\omega_{p1}}\right) \left(1 + \frac{s}{\omega_{p2}}\right)} = \frac{G_{mA} G_{mB} R_A R_B \left(1 - s \frac{C_C}{G_{mB}}\right)}{(1 + s G_{mB} R_A R_B C_C) \left(1 + s \frac{C_B}{G_{mB}}\right)}$$

These simplifying assumptions hold because capacitance  $C_B$  will include the capacitance of the load, and the compensation capacitance  $C_C$  will often be much larger than the gate-source capacitance of M8 which makes up most of  $C_A$ .

With the transfer function in factored form, we can find the open-loop DC gain, poles, and zero of the compensated op-amp. They are given by:

$$A_{v0} = G_{mA} G_{mB} R_A R_B$$

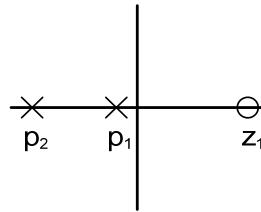
$$\omega_{p1} = \frac{-1}{G_{mB} R_B C_C R_A} = \frac{-1}{|A_{v2}| C_C R_A}$$

$$\omega_{p2} = \frac{-G_{mB}}{C_B}$$

$$\omega_z = \frac{G_{mB}}{C_C}$$

$$\omega_{GBW} = \frac{G_{mA}}{C_C}$$

Note that the addition of the compensation capacitor  $C_C$  caused the poles to split. One pole moved closer to the origin by a factor of  $A_{v2} = G_{mB} R_B$ , while the other pole moved away from the origin by a factor of  $A_{v2}$ . This compensation technique is called "pole splitting". The pole-zero plot of this transfer function is illustrated in Figure 9-8. Also, notice the creation of a zero as a result of the path created by the compensation capacitor.



**Figure 9-8:** Pole-Zero Plot for a Compensated Op-amp

Using the compensated op-amp in a feedback loop produces the following transfer function:

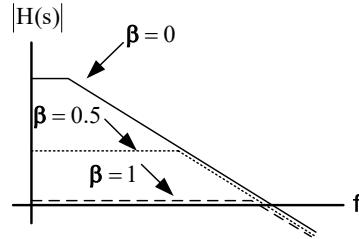
$$A_C(s) = \frac{H_C(s)}{1 + H_C(s)\beta} = \frac{A_{v0} \left(1 - \frac{s}{\omega_z}\right)}{\left(1 + \frac{s}{\omega_{p1}}\right) \left(1 + \frac{s}{\omega_{p2}}\right) + A_{v0}\beta \left(1 - \frac{s}{\omega_z}\right)}$$

$$A_C(s) = \frac{A_{v0} \left(1 - \frac{s}{\omega_z}\right)}{1 + \beta A_{v0} + s \left(\frac{1}{\omega_{p1}} + \frac{1}{\omega_{p2}} - \frac{\beta A_{v0}}{\omega_z}\right) + \frac{s^2}{\omega_{p1}\omega_{p2}}}$$

We see that when  $\beta = 0$  or is very small, the expression will more closely resemble the original open-loop transfer function. This expression could also be rewritten in the following way:

$$A_C(s) = \frac{\left(1 - \frac{s}{\omega_z}\right)}{\beta \left(1 + \frac{1}{\beta A_{v0}} + s \left(\frac{1}{\beta A_{v0}\omega_{p1}} + \frac{1}{\beta A_{v0}\omega_{p2}} - \frac{1}{\omega_z}\right) + \frac{s^2}{\beta A_{v0}\omega_{p1}\omega_{p2}}\right)}$$

If  $\omega_{p2} > A_{v0}\omega_{p1}$  and  $\omega_z > \omega_{p2}$ , then as  $\beta$  gets closer to 1 the above expression can be approximated as having a dominant pole located at  $\beta A_{v0}\omega_{p1}$ . Therefore, we can see that the effect of varying  $\beta$  from 0 to 1 is to shift the location of the closed-loop system's dominant pole from the open-loop dominant pole location up to the gain-bandwidth product (GBW). Figure 9-9 illustrates the effect of feedback on the frequency response.



**Figure 9-9:** Open and Closed Loop Frequency Response

To ensure the feedback system is stable at unity gain ( $\beta = 1$ ), the phase margin must be examined. The phase margin is the amount of phase before phase inversion ( $180^\circ$ ) at the unity-gain frequency (UGF). If the UGF is approximately equal to the GBW, then the expression for the phase margin is given by:

$$PM = 180^\circ - \tan^{-1} \left( \frac{\omega_{UGF}}{\omega_{p1}} \right) - \tan^{-1} \left( \frac{\omega_{UGF}}{\omega_{p2}} \right) - \tan^{-1} \left( \frac{\omega_{UGF}}{\omega_z} \right)$$

$$PM = 180^\circ - \tan^{-1} \left( \frac{\omega_{GBW}}{\omega_{p1}} \right) - \tan^{-1} \left( \frac{\omega_{GBW}}{\omega_{p2}} \right) - \tan^{-1} \left( \frac{\omega_{GBW}}{\omega_z} \right)$$

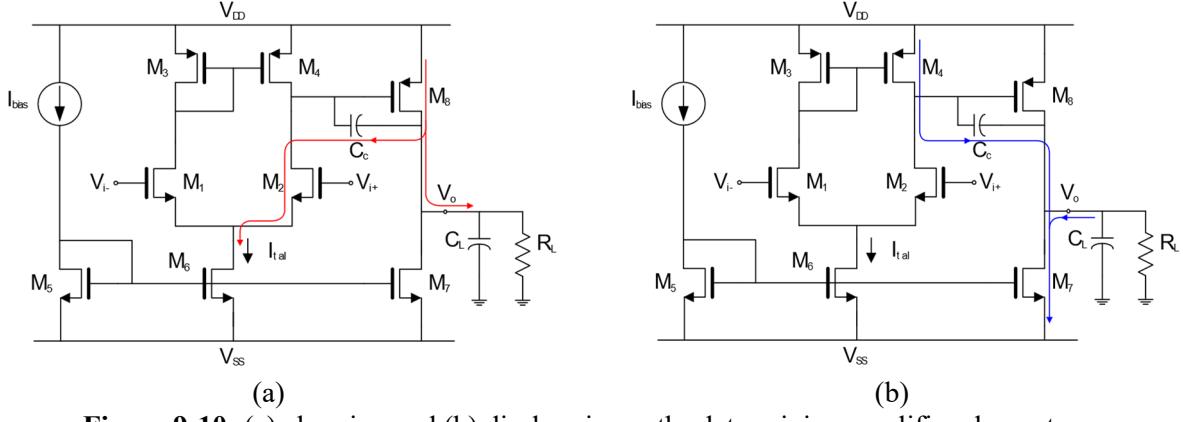
$$PM = 180^\circ - \tan^{-1}(A_{v0}) - \tan^{-1} \left( \frac{\omega_{GBW}}{\omega_{p2}} \right) - \tan^{-1} \left( \frac{\omega_{GBW}}{\omega_z} \right)$$

$$PM \approx 90^\circ - \tan^{-1} \left( \frac{\omega_{GBW}}{\omega_{p2}} \right) - \tan^{-1} \left( \frac{\omega_{GBW}}{\omega_z} \right)$$

The phase margin is improved by moving the non-dominant pole and zero to higher frequencies away from the unity-gain frequency. The phase margin can also be improved by using compensation techniques which place the zero in the left half plane.

The slew rate of the amplifier is determined by the path needed to charge or discharge both the compensation capacitor,  $C_C$ , as well as the load capacitor,  $C_L$ . The different charging paths for the two-

stage Miller amplifier are shown in Figure 9-10. The path needed to charge  $C_C$  and  $C_L$  for increasing the output voltage is shown in Figure 9-10(a) while Figure 9-10(b) shows the path for discharging these capacitors to decrease the output voltage. These different paths will lead to different rising and falling slew rates.



**Figure 9-10:** (a) charging and (b) discharging paths determining amplifier slew rates

When the output voltage is increasing, the two-stage amplifier is usually designed such that the current that is provided by M8 is capable of sourcing the current needed to charge  $C_L$  while also providing  $I_{tail}$  which is needed to charge  $C_C$  (Figure 9-10a). In this instance, charging  $C_C$  imposes the dominant limit on the output voltage rate of change, and therefore the rising slew rate is determined by

$$SR^+ = \frac{I_{tail}}{C_C}$$

When the output voltage is decreasing, now the current sunk by M7 must sink  $I_{tail}$  which is discharging  $C_C$ , but it must also sink the current needed to discharge  $C_L$  (Figure 9-10b). Because the gate voltage of M7 is fixed, it can only sink a fixed amount of current,  $I_{D7}$ . If the load capacitor is similar in value to the compensation capacitor, then the charging of  $C_C$  will still impose the dominant limit on the slew rate. This means that the falling slew rate will still be determined by  $I_{tail}$  and  $C_C$ , but in the opposite direction, yielding

$$SR^- = \frac{-I_{tail}}{C_C}$$

However, if  $I_{D7}$  is significantly less than  $I_{tail}$  or if  $C_L$  is significantly larger than  $C_C$ , then  $I_{D7}$  discharging  $C_L$  will create the dominant limit affecting the slew rate, and thus the falling slew rate will be given approximately by

$$SR^- = \frac{-I_{D7}}{C_L}$$

The performance characteristics of the two-stage amplifier are summarized below:

$$A_{v0} = g_{m1,2}g_{m8}(r_{02}||r_{04})(r_{07}||r_{08}||R_L)$$

$$\omega_{p1} = \frac{-1}{g_{m8}(r_{07}||r_{08}||R_L)C_C(r_{02}||r_{04})} \quad \omega_{p2} = \frac{-g_{m8}}{C_L} \quad \omega_z = \frac{g_{m8}}{C_C}$$

$$\omega_{GBW} = \frac{g_{m1,2}}{C_C} \quad SR^+ = \frac{I_{tail}}{C_C} \quad SR^- = \frac{-I_{tail}}{C_C} \text{ or } \frac{-I_{D7}}{C_L}$$

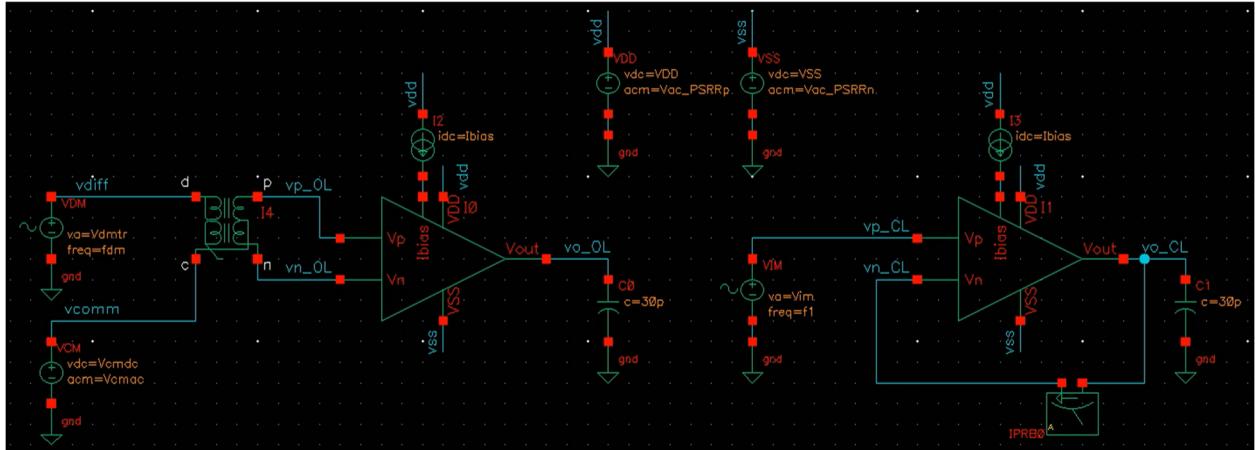
$$PM \approx 90^\circ - \tan^{-1}\left(\frac{\omega_{GBW}}{\omega_{p2}}\right) - \tan^{-1}\left(\frac{\omega_{GBW}}{\omega_z}\right)$$

Note that in the above expression for phase margin, the sign of the poles and zeros is already accounted for by the signs associated with the inverse tangent functions. In other words, you can consider the arguments to the inverse tangent functions to utilize the absolute value of the pole/zero frequencies in this instance.

## Op-amp Test Bench Setup

For this lab, we will mostly be using ADE XL as the simulation management tool rather than ADE L. A single setup of ADE XL is stored as a cell view and has additional functionality compared with ADE L. Each ADE XL cell view can be composed of multiple “tests,” each of which is an independent ADE L state with its own set of analyses, outputs, design variables, and default values for those variables. Additionally, each test within ADE XL can be linked to a different schematic cell view if needed. This functionality can help to simplify an individual test bench schematic so that it contains only what is needed for a specific set of analyses while enabling multiple test bench schematics to be used together for a complete characterization of a circuit.

To fully characterize the op-amp designed in this lab, a single test bench schematic with multiple amplifier instances can be created. One of the setups in the schematic will be for open-loop tests while the other will be for closed-loop tests. An example of this is shown in Figure 9-11. In the open-loop setup, we can characterize the power dissipation, common-mode input voltage range, differential gain, common-mode gain, PSRR, slew rate, and input-referred noise.



**Figure 9-11:** schematic with open- and closed-loop amplifier test setups

The component that looks like a center-tapped transformer in the test bench is the **ideal\_balun** component from the **analogLib** library. This component is used to convert a single-ended signal into two differential signals while also allowing another voltage to provide the common-mode level. This kind of setup with one source that is used to create a pair of differential signals will be needed for a few of the tests described below. There are other ways to implement a similar setup, but the **ideal\_balun** component provides one of the simpler ways of doing this. The terminal names of the **ideal\_balun** component are also displayed next to the respective terminal in Figure 9-11. The “d” terminal will take a voltage signal and will convert it to a pair of differential signals at the “p” (positive) and “n” (negative) outputs. The “c” terminal of the

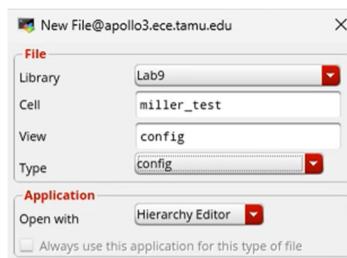
**ideal\_balun** takes in a voltage and sets this as the common-mode level of the two differential signals. Refer to Lab 6 for more information on the use of the **ideal\_balun** component.

The VDD and VSS voltage sources each have a design variable given for their AC magnitude parameters, which are, respectively, “Vac\_PSRRp” and “Vac\_PSRRn” in order to test PSRR<sup>+</sup> and PSRR<sup>-</sup>, as described later. The source labeled “VDM” is a **vsin** source from the **analogLib** library that provides the differential inputs to the open-loop amplifier. This source creates a sinusoidal signal for testing the rising and falling slew rates of the op-amp. The VDM source will also be used to evaluate the op-amp’s output swing. The DC voltage and AC magnitude of this source are also given as design variables for finding the linear output swing range and the differential gain magnitude. The source labeled “VCM” provides the DC common-mode voltage for the open-loop amplifier. The DC voltage and AC magnitude of this source are also set as separate design variables for sweeping the common-mode input voltage range (to properly set the DC bias point of the amplifier) and for testing the common-mode gain to calculate CMRR. In the closed-loop setup, the source labeled “VIM” is a **vsin** source from **analogLib** that uses the same design variables for the DC voltage and AC magnitude as the VCM source. This makes it easy to set the DC bias point of both setups to a value within the common-mode input voltage range of the op-amp. The VIM source will be used for the linearity (intermodulation) tests, but the setup of this source will be described in more detail later.

The test bench in Figure 9-11 is given as an example test setup for characterizing the op-amp. For the purposes of this lab manual, this test bench also provides an example of how to utilize more of the functionality of ADE XL for performing different types of simulations on the same circuit, which is described in more detail below. In this way, the multi-test functionality of ADE XL can be utilized to simplify a test bench schematic. However, if preferred, the op-amp can be characterized using a test bench schematic with a greater number of op-amp configurations, each of which covers different types of simulations. For more complex circuit blocks, however, it can often be preferable to keep the test bench schematic simpler and utilize different test configurations in ADE XL in order to reduce the computational resources required for a given set of simulations.

### Configuration (Config) View Setup

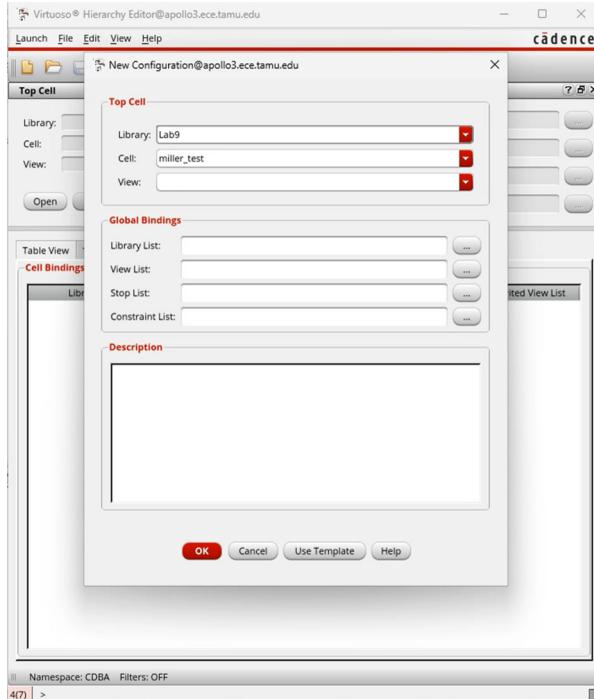
Once the test bench schematic is set up, there is one more cell view that needs to be created before creating the ADE XL cell view. This additional cell view is known as a *config* view, and although it is not required for performing the tests, it will notably simplify the process of comparing pre- and post-layout simulations while utilizing multiple tests in the ADE XL cell view. In addition to this, the *config* view can also be used to easily switch between different versions of a schematic, providing a method of version control for a cell, or to swap between idealized models of a cell or a transistor-level schematic. Because of this, the *config* view is often used when creating larger, hierarchical designs to more easily configure various test setups.



**Figure 9-12:** Creation of a new “config” cell view

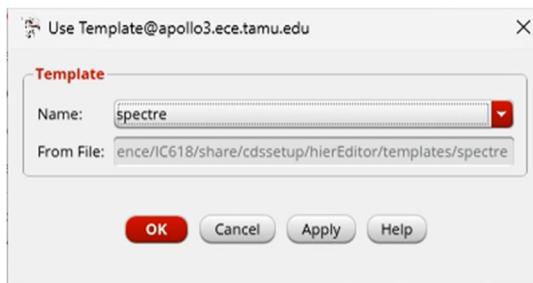
First, close the test bench schematic (it will be reopened later after it is linked to the config view). Then, to create a *config* view, from the Library Manager window go to *File* → *New* → *Cell View*. The new file creation window should appear, like what is shown in Figure 9-12, but the “Cell” field should be replaced

with the name you gave to the test bench schematic cell. Change the “View” and “Type” fields to “config” and make sure the “Open with” application changes to “Hierarchy Editor.” When all fields are changed appropriately, click “OK” at the bottom of the window. A pair of new windows should appear. One of them is the hierarchy editor (the *config* view) and the other should be a setup window for the new configuration view, as shown in Figure 9-13.



**Figure 9-13:** Setup of new configuration view

Within the configuration setup window, click the “Use Template” button at the bottom of the window. A new window will appear like what is shown in Figure 9-14. Various templates for the config view setup are pre-made for different simulator types or other conditions. Since we are using the “spectre” simulator, choose the “spectre” option from the “name” dropdown menu in the template selection window, and click “OK.”

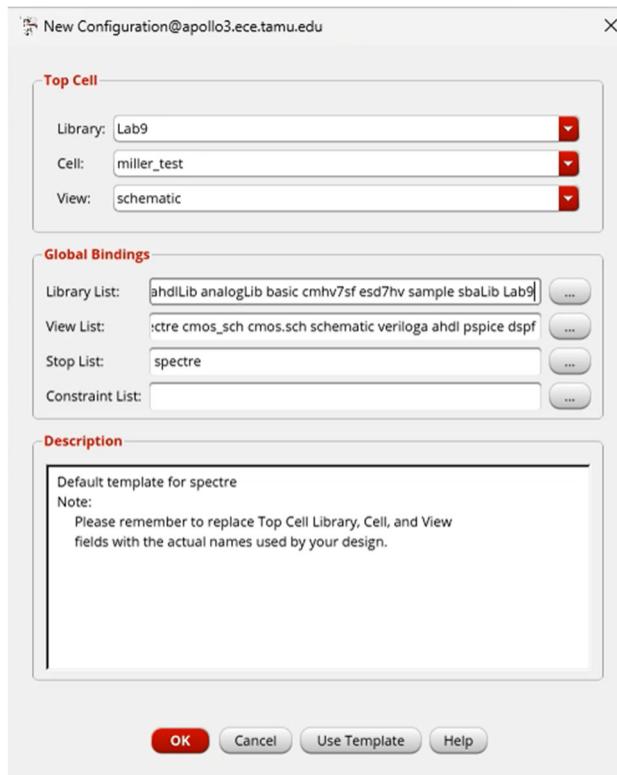


**Figure 9-14:** Template selection window

After selecting the spectre template, several of the fields in the original config setup window should be filled in with default values. The “View” option near the top of the window (in the “Top Cell” section) will need to be changed to “schematic” so that the config view is associated with the test bench schematic and will thus have control over which view types are used for the components in the test bench.

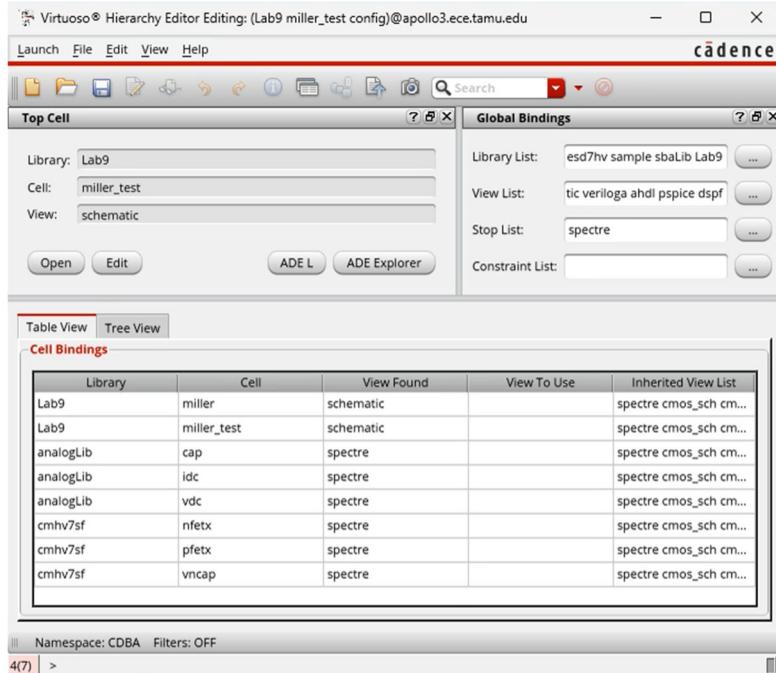
The “Library List” field provides the default list of libraries that the *config* view will use to find the cell views associated with the components in the schematic. This list should include the default libraries associated with Cadence and with the PDK (like **analogLib** and **cmhv7sf**, for example, along with a few other default libraries), as well as the current library which contains your op-amp cell. Clicking the ellipsis (“...”) button will allow you to select the libraries you would like to include in the *config* view’s search path. The default library option of “myLib” that is created after the template selection can be removed from the list since it is not a valid library.

The “View List” field provides the list of cell view types that the *config* view will look for when it identifies a component in the schematic. Other cell view types can still be chosen manually even if they are not specified in this list, so this list does not need to be modified for the purposes of this lab. The “Stop List” provides a list of cell view types for which the *config* view will no longer search in the schematic hierarchy. Similarly to the View List, this list also does not need to be modified from the default that is provided by the template. Once complete, the *config* setup window should look similarly to what is shown in Figure 9-15. These fields can also be modified later if needed, so when you have finished the initial setup, click “OK” at the bottom of the window.



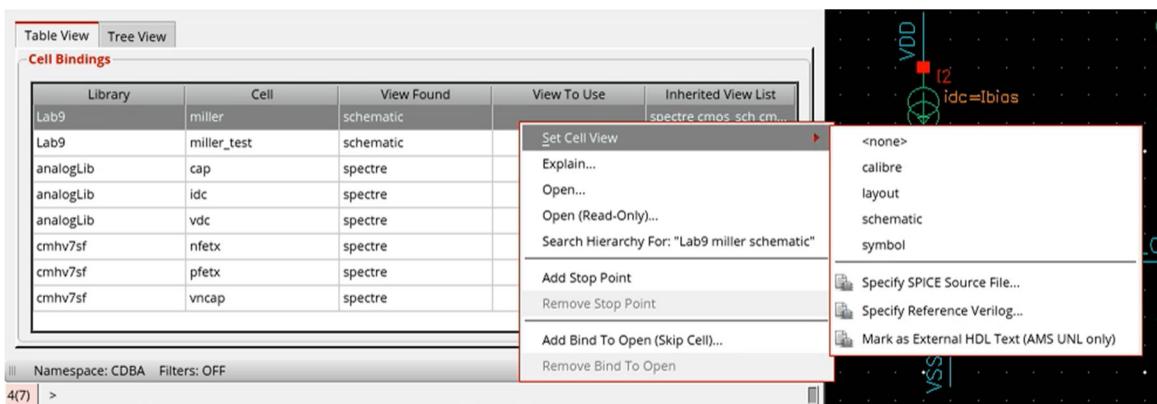
**Figure 9-15:** Finished configuration setup window

After confirming the configuration setup, the finished *config* view should look like what is shown in Figure 9-16. Save the new *config* view so that is added to the list of cell views for the op-amp test bench. The “Global Bindings” sub-window in the top right contains the various lists that were edited in the previous steps. If changes need to be made to any of these lists, they can be made here rather than re-creating the *config* view. The “Top Cell” sub-window should point to the test bench cell and the corresponding schematic view that will be used for the test bench. The schematic can also be opened from the *config* view by clicking “Open” in the bottom-left of the “Top Cell” sub-window.



**Figure 9-16:** Initialized config view for the test bench schematic. The “miller” cell is the op-amp in the test bench.

The “Cell Bindings” sub-window in the bottom portion of the config view controls which cell view type is used for a given component in the associated schematic. **This sub-window can be used to switch between schematic and calibre view types for easily switching between pre- and post-layout simulations.** To switch the cell view type, find the row corresponding to the component you want to change, and right-click over the “View To Use” field. A menu should appear like what is shown in Figure 9-17. Within this first menu, hover over the “Set Cell View” option, and from the second menu that appears you can choose the appropriate cell view for the tests that you are performing. After the cell view type is selected, save the *config* view to apply the changes. Additionally, **whenever changes are made to the schematic associated with the *config* view, you should save the *config* view after you “check and save” the schematic.**



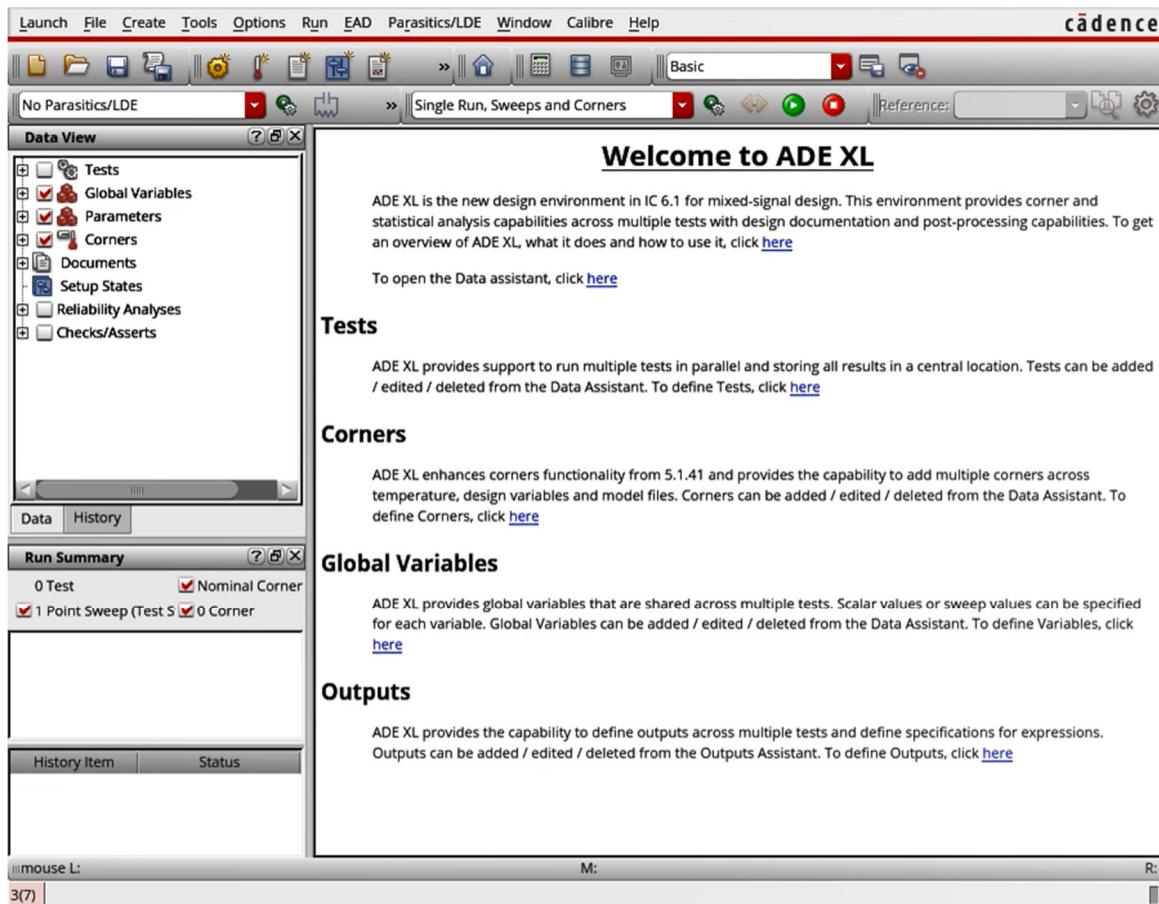
**Figure 9-17:** Cell view type selection in the config window

After a *config* view is created for a cell, you can open the *config* view to open the schematic as well. When the *config* view is opened, a menu will appear asking if you want to open both the *config* view as well as the schematic. The default choice for the schematic that will be opened is the one that is currently linked to the *config* view. If it is the correct view, then click “OK” in this window to open both cell views. However,

when copying a cell with a *config* view, the copied *config* view may be linked to the original schematic (i.e. a schematic view belonging to a different cell). If this is the case, edit the Top Cell parameters to change which schematic cell view is linked to the copied *config* view, and then save the *config* view. The next time you open the *config* view, you will be prompted again with whether to open the associated schematic view. If the changes were made correctly, the schematic cell view and *config* view you are opening should belong to the same cell.

## ADE XL Initialization

To open ADE XL, from the test bench schematic window go to *Launch* → *ADE XL*. A window should appear with two options in which you should select “Create new view” and then click “OK.” A new window will appear with options for how you would like to open the ADE XL view. You can open the view in a tab of the current schematic window, or you may choose to open it in a new, separate window. Select which of these options you prefer, and then click “OK.” The opening screen for the ADE XL view should look like what is shown in Figure 9-18.



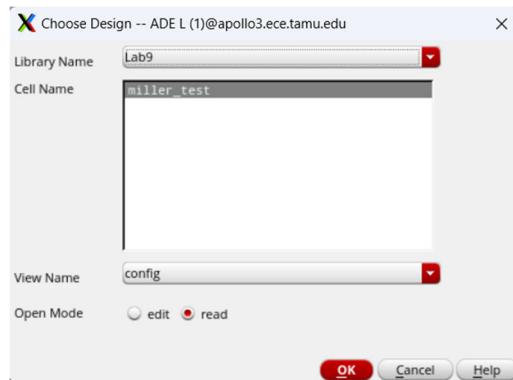
**Figure 9-18:** Initial screen for ADE XL view

Within the ADE XL window, much of the simulation setup occurs within the “Data View” sub-window. The *Tests* list contains independent simulation states, each of which can have their own set of analyses, design variables, and default values of those design variables. The *Global Variables* list contains a copy of all design variables across every test. The variable values in the *Global Variables* list can be used to overwrite the default value of the same variable across multiple tests. The list of *Global Variables* can also be used for performing parametric sweeps of multiple design variables. The *Parameters* list contains any

circuit component parameters (that are set as constant values in the schematic) that are varied for individual simulations (like the parameter sweep setup described in Lab 1) or as part of a parametric sweep. The *Corners* list is used for storing specific design variable, model parameter, and/or other simulator conditions as a “corner case” for the test bench. Along with saving specific design variable conditions, the *Corners* list can also be used for process corner tests (explained below) and for testing a circuit in different modes of operation (assuming these modes can be selected with a design variable).

### Setting up the Common-mode Test

To create a new test, click the “+” next to *Tests* and then select the “Click to add test” option. A window for a new ADE L state will appear along with another window that looks like what is shown in Figure 9-19. In this window, you will choose the “design” (i.e. which cell view) you want the test setup to be associated with. The “View Name” option is the view type that the test will be associated with. Normally, the test would be associated with a “schematic” cell view by default. However, we will change the view to the *config* view that was created earlier. This means that the *config* view will control the setup of the test bench rather than the schematic itself or the environment setup of the individual test. **This step will be needed when setting up every test in the ADE XL cell view.** By linking each test in the ADE XL cell view to the *config* view, we can switch between pre- and post-layout simulations by changing one setting in the *config* view rather than multiple settings across each of the tests. **Note:** the window in Figure 9-19 can also be accessed by going to *Setup → Design...* in the ADE L test setup window.



**Figure 9-19:** Design selection window for ADE XL test

When a test is initialized, there may be default simulations that appear in the “Analyses” sub-window of the ADE L window that configures the test. For the first test, configure a DC sweep of the common-mode input voltage (the VCM source in Figure 9-11) on the open-loop op-amp. The DC sweep results will be used to identify the minimum and maximum values of the common-mode input voltage and to evaluate the power consumption of the op-amp across the common-mode input voltage range. Recall from Lab 6 that the common-mode input voltage range is determined by the voltage over which the differential pair devices are all in the saturation region. In addition to the DC sweep, configure an AC simulation of the common-mode voltage to plot the common-mode gain magnitude. For this test, the design variables controlling the DC and AC components of the VDM source should be set to zero, and the design variables controlling the AC magnitude of the VDD and VSS sources should also be set to zero. The configuration of the VIM source will be addressed in a later section, but if any design variables have been added to this component, then they can be set to arbitrary values for the initial configuration of the test. Use the *Outputs → To Be Saved...* menu in the ADE L window for the test to add the necessary transistor “region” parameters and test bench signals for evaluating the common-mode input voltage range, the power consumption, and the common-mode gain. Recall that the “X” or “Shift-X” hotkeys can be used to descend into the op-amp symbol to select the DC operating point parameters of the transistors, and the “B” hotkey can be used to ascend back to the test bench level.

Once the initial setup of the test is complete, the ADE L window can be closed. Changes to the test can be made from the main ADE XL window as well. Expanding the list of tests in the Data View allows the analyses to be edited as well as the default values of the design variables. The test is assigned a default value after it is created, but this can also be changed to something more meaningful. For this test, click the test name and then tap the F2 key to edit the name. Since the simulations of this test are primarily for common-mode characteristics, rename the test to “*test\_comm*” to make this clearer. **Note:** apart from editing the name of the test, all other changes to a test setup can be made in the ADE L window associated with the test state, so this window can be used for configuring each test if it is more familiar. Additional details regarding how to edit the tests from the main ADE XL window are provided for instructional purposes.

The outputs of the various tests can also be edited in the main ADE XL window. After the initial setup of *test\_comm*, the “Outputs Setup” tab of the main ADE XL window will be populated with any signals or expressions that were created for the test through the ADE L setup window. The outputs for a test can also be edited within the Outputs Setup tab directly. As shown in Figure 9-20, the leftmost button under the tab names can be used to add an output to a specific test. When an expression is created as an output, double-clicking the “Name” field of the expression (the second column in the Outputs Setup tab) will allow you to define your own name for the expression, much like for output expressions created in an ADE L state. This output name can be referenced in other expressions to cascade various calculations together.

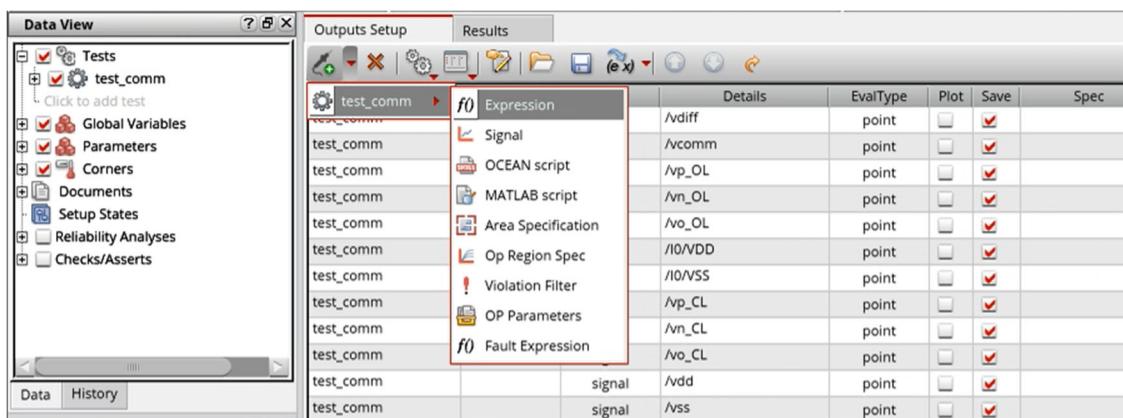


Figure 9-20: Creating new test outputs directly in the ADE XL Outputs Setup tab

Additionally, when an expression is created, double-clicking the “Details” field will allow you to type in the expression you want. Otherwise, you can click the ellipsis button to bring up the calculator and create an expression there. Once an expression is completed in the calculator, the gear icon button in the calculator window can be used to send the expression to the ADE XL list of outputs. Alternatively, the “<” button (which also appears after double-clicking in the Details field of an output expression) can be used to import the current calculator expression into the Details field of the ADE XL output.

Once the *test\_comm* test is configured with the DC and AC simulations, close the ADE L window for this test and return to the main ADE XL window. When a test is created, the design variables associated with that test are added to the *Global Variables* list and are enabled by default. This means that the values in the *Global Variables* list will overwrite the values of the variables in other tests. This behavior will interfere with the setup of the remaining tests, so the *Global Variables* must be disabled for the time being. To do this, right-click on *Global Variables* and click the “Deselect All” option. **Note:** if new design variables are added to a test, they will be added to the *Global Variables* list and enabled by default, so they will need to be disabled in the same way. Variables in the *Global Variables* list which have a check mark next to them are enabled. Alternatively, the use of global variables can be disabled by clicking the box next to *Global*

*Variables* to toggle the check mark for the entire list. However, if a design variable is used to control the test bench conditions, like the bias current of the op-amps in the test bench, for example, enabling this global variable can make it easier to adjust the test condition across all tests if needed.

## Setting up the Differential-mode Test

Now we will configure the differential-mode tests. Create a new test in the Data View. This test will be used primarily for measuring the differential characteristics of the amplifier, so the test can be renamed to *test\_diff* to indicate this purpose. **Remember to associate the design selection of this test with the config view rather than the schematic view of the test bench, just as was done for *test\_comm*.** If similar signals and/or expressions will be used as outputs between multiple tests, they can be copied between tests within the Outputs Setup tab. For example, to copy the voltage signals from *test\_comm* into the *test\_diff* setup, first select all of the signals in the *test\_comm* outputs list (hold either the Shift or Ctrl key while left-clicking on the output row to select multiple outputs at one time). With the outputs selected, right-click on one of them, and go to *Copy to Test* in the menu that appears. Then, select which test you want the outputs added to. This can help to significantly speed up the process of building a new list of test outputs.

For the setup of *test\_diff*, the design variable associated with the DC voltage of the VCM source should be kept within the common-mode input voltage range of the op-amp. The design variable controlling the AC magnitude of the VCM source should be set to zero. Initially, the DC voltage of the VDM source can be set to zero, but the default value will be changed to the input offset voltage after an initial DC sweep of the differential input voltage. This input offset voltage will be important for later tests because this will set the output of the open-loop amplifier to zero, allowing the open-loop amplifier to operate with its maximum gain and maximum voltage swing before distorting. The variable for the AC magnitude of the VDM source should be set to 1 so that we can evaluate the differential gain of the op-amp from the open-loop setup in the test bench schematic. For this test, configure a DC sweep of the differential DC voltage as well as an AC sweep of the AC differential voltage. The DC sweep can be used for the output swing measurement described later while the AC simulation will be used for determining the open-loop differential gain of the op-amp.

We can also add the stability analysis to the setup of *test\_diff*. See the Lab 8 manual for more information on how to setup a stability analysis. Once the stability analysis is created in the state, there are two ways that we can retrieve the plot data for the loop gain magnitude and phase responses from the *Direct Plot* menu.

1. One way is to use the ADE L window associated with the test. With the *test\_diff* ADE L window open, a simulation of only this test with its current analyses can be performed like a standalone ADE L state. Once the simulations complete, the loop gain magnitude and phase responses can be retrieved by going to *Results* → *Direct Plot* → *Main Form...* as was done for Lab 8.
2. A second method is to perform a simulation of all tests in the ADE XL state using the  button near the top of the ADE XL window. This will begin to populate the “Results” tab with the outputs from the simulations as each test completes. Once the tests are completed, right click on the  button for one of the outputs from *test\_diff* that appears in the Results tab and go to *Direct Plot* → *Main Form...* in the menu that appears.

In order to plot the results from the various tests, multiple options are also available. To plot all waveforms for which plotting is enabled, click the  button at the top of the Results tab in the ADE XL window. Plotting of a signal or expression can be enabled or disabled in the Outputs Setup tab. Alternatively, a single waveform can be plotted by double-clicking the  symbol in the row for a specified output. Multiple output waveforms can be plotted at once by selecting multiple waveforms (as before, holding Shift or Ctrl

while left-clicking allows you to select multiple output waveforms), then right-clicking on one of the selected outputs, and choosing the *Plot* option from the menu. There is a plotting option at the top of the Results tab in the ADE XL main window, next to the  button, that controls whether additional plots will overwrite previous ones, if they will be added to windows, or displayed in a new tab of the waveform viewer window. Additionally, the output expressions can be reevaluated using the same simulation data (for example, if an output expression definition is changed) by clicking the  button at the top of the Results tab. This reevaluation functionality is useful for verifying that a test's output expressions work properly before comparing the results of new simulations. **Once the setup for a test is completed, remember to save the ADE XL cell view.**

Once *test\_diff* is configured with the DC, AC, and stability simulations, construct output expressions to provide the value of the following quantities: input offset voltage,  $A_{v0}$  (op-amp DC gain),  $f_{p1}$  (first pole frequency), GBW, and phase margin. These quantities will be needed for additional simulations later in the lab.

### Common-mode Rejection Ratio (CMRR) Measurement

Like the common-mode and differential-mode gains, CMRR is a frequency dependent characteristic, as discussed in Lab 6. To determine the CMRR, we need both the common-mode and differential-mode gain magnitude responses. Since we have set up *test\_comm* and *test\_diff*, the data for both the common-mode and differential-mode gain responses will be available after simulating all the tests in the ADE XL state. To use the data from *test\_comm* as part of a calculation performed in the *test\_diff* test, we will use the *calcVal* function. First, you must name the output expression(s) that you want to call with the *calcVal* function. **Note:** when naming expressions and tests, use underscores (“\_”) instead of spaces to make the expressions easier to reference in other calculations. For example, you may choose to name the common-mode gain result from *test\_comm* as “gain\_common” in the list of outputs. Similarly, you may choose to name the differential-mode gain from *test\_diff* as “gain\_diff” in the list of outputs.

After *test\_comm* and *test\_diff* have been performed and the needed output expressions have been named, we can form the CMRR expression. To use the *gain\_common* output in the *test\_diff* outputs, the *calcVal* function should be called with the following syntax:

```
calcVal(“<output name>” “<test name>”)
```

Using the above test and expression names as an example, the common-mode gain from *test\_comm* would be called in one of the *test\_diff* output expressions as:

```
calcVal(“gain_common” “test_comm”)
```

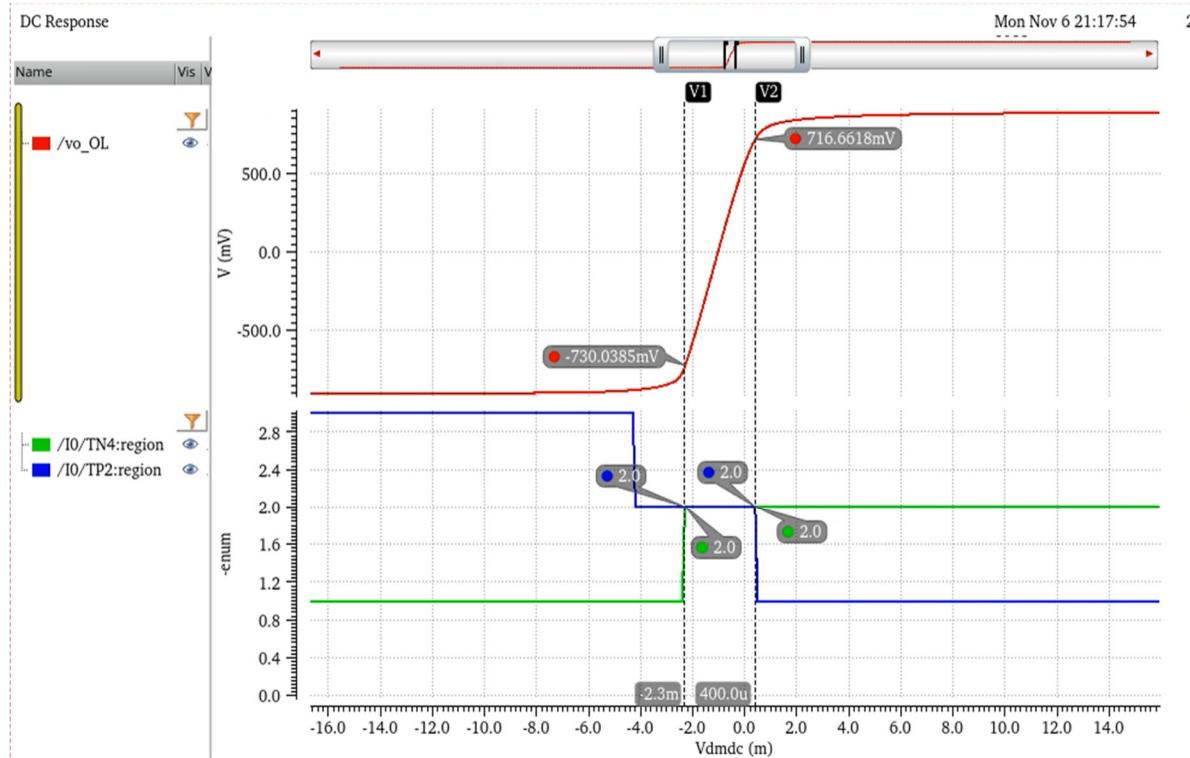
The *calcVal* function is not needed for outputs that already exist within a test. It is only needed when an expression requires an output from a different test in the ADE XL state. Use the above syntax of the *calcVal* function to calculate the CMRR in dB. **Note:** the *calcVal* function can also be used to define the value for a design variable, thus allowing the result(s) from one test to be used in setting the conditions of a following test.

As mentioned in Lab 6, additional op-amp setups in the test bench schematic could be used as an alternative method for calculating the CMRR. One of the setups in the test bench would be used for common-mode measurements while the other would be used for differential-mode measurements. In this way, both the differential-mode and common-mode gain responses would exist from a single simulation and could thus be combined to calculate the CMRR. The above method is introduced for instructional purposes to demonstrate the additional functionality that is available in ADE XL.

### Output Swing Measurement

The output swing of an amplifier is limited by the output voltage at which any of the output transistors are no longer operating in the saturation region. Through simulations, there are multiple ways that the output swing range can be evaluated. However, a more precise evaluation of the output swing will often require a distortion specification, typically based on total harmonic distortion (THD). THD is a measure of the power contained in the harmonic components of a signal relative to the power of the fundamental tone.

One method of determining the output swing from simulations is to use a DC sweep on the open-loop amplifier, as in Lab 8. The “region” operating point parameter of the output transistors can be used to determine when the devices transition from the saturation region to the triode region (refer to lab 5 for more information on the region parameter). The output voltages at which one of the devices is no longer in saturation can define the output swing of the op-amp. An example of what these simulation results may look like is shown in Figure 9-21. The vertical markers (placed with the “V” hotkey) are placed where either of the output devices transition between the saturation and triode regions. The output swing would be the difference between the output voltage values indicated at these two points, which is about 1.45V in this case. **Note:** one disadvantage of this method for evaluating the output swing is that the simulation must be updated for post-layout simulations. To plot the region parameter of the output transistors, the devices must be re-selected from the calibre view of the cell after the post-layout extraction (refer to Lab 5). Because this method was used in Lab 8, we will explore another method of determining the output swing in this lab.

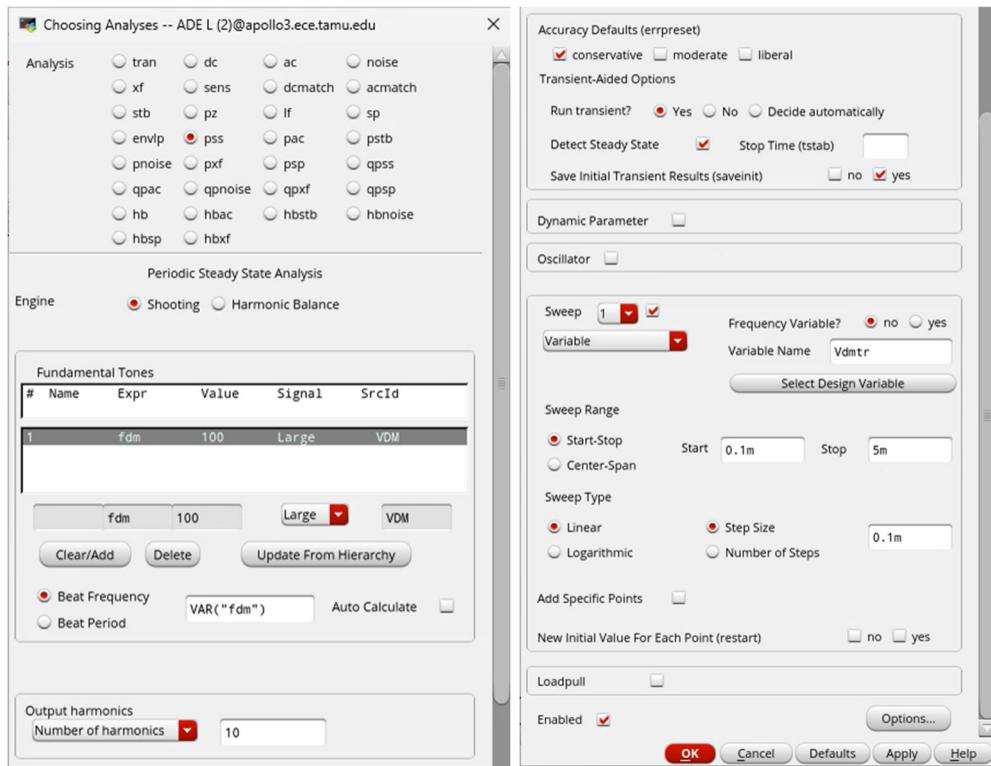


**Figure 9-21:** Output swing measurement based on transistor region parameter

Another option for determining the output swing is based on a distortion requirement. For this method, we will use a “periodic steady state,” or “PSS,” analysis. The PSS analysis performs a short transient simulation until periodic behavior occurs in the circuit being analyzed. Once periodic behavior occurs, the simulator analyzes the frequency components that are present in the signals of interest. Because the PSS analysis inherently includes a transient simulation, any transient parameters of the sources must be properly set for them to be active in the analysis (e.g. “AC magnitude” parameters of sources will not activate them, but “Amplitude” parameters will).

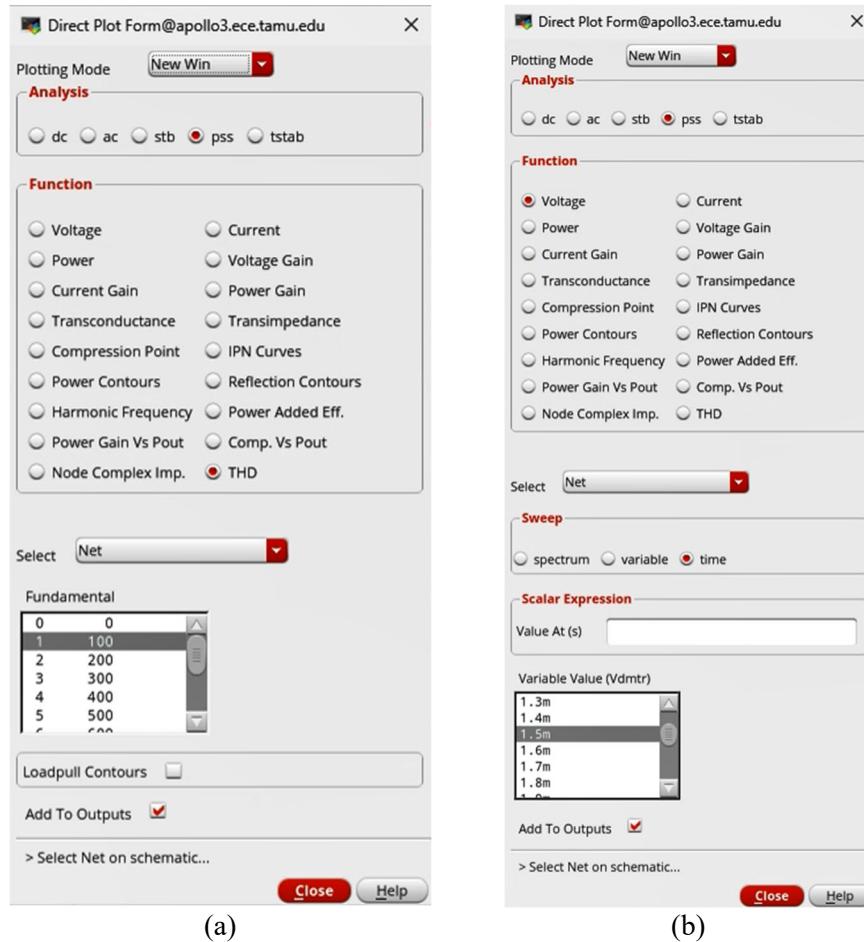
We will set up the PSS analysis as part of the *test\_diff* setup. Before configuring the PSS simulation, make sure that a design variable has been added, and a default value for it has been given, to control the transient amplitude and frequency of the VDM source in the test bench shown in Figure 9-11. In the figure, the design variable for the transient amplitude has been called “Vdmtr” and the frequency of this signal has been called “fdm.” For now, set the default value of Vdmtr as 1mV and fdm to 10kHz. In addition to these default values, make sure to change the default value of the differential DC voltage of the VDM source to the input offset voltage found from an earlier test. **The input offset voltage is needed to center open-loop amplifier’s output voltage around 0V.** This improves the symmetry of the output which simplifies the evaluation of the output swing. With the design variables added to the test bench, expand the settings for *test\_diff* in the Data View by clicking the “+” next to the test name, and make sure that the design variables are added to the list of design variables for the test as well. Then, expand the “analyses” list and click the space which says “Click to add analysis” to open the analysis setup window. Alternatively, open the ADE L state associated with the test, and open the analysis setup window from there. In the analysis setup window, select “pss” from the options at the top.

The settings for the PSS simulation are shown in Figure 9-22. The *VAR* function allows the simulator to utilize the value of a design variable in an output expression or a condition of the simulator setup (the design variable name given in the *VAR* function must be surrounded by double quotes). In this case, the design variable controlling the frequency of the transient sine wave produced by the VDM source in the test bench is chosen as the fundamental frequency for the analysis. Defining it in this way mitigates the need to update the PSS simulation setup when the design variable value is changed. **The default value for the input frequency needs to be significantly lower than the bandwidth of the op-amp to get an accurate assessment of the output swing while using the open-loop amplifier.** In this case, 100Hz is chosen as the frequency of the input sine wave. **Note:** the *VAR* function can also be used in the definition of output expressions to make an output calculation dependent on the design variable values in a test configuration.



**Figure 9-22:** Settings for PSS simulation setup

The “number of harmonics” parameter can affect the accuracy of some calculations, like THD. A larger number of harmonics can notably increase simulation time for more complicated circuits, but for the purposes of this lab, analyzing 10 harmonic components should not present any issues. The simulator should be able to detect the steady state of the circuit automatically, but if convergence issues occur then you may need to disable the “Detect Steady State” option and instead provide a stop time for the simulation. If this is needed, it can be helpful if the stop time is a multiple of the period for the fundamental frequency chosen earlier in the analysis settings. Finally, because we would like to assess the output swing, we will need to vary the differential input voltage amplitude. We do this by enabling the sweep option, selecting the transient signal amplitude in the VDM source, and configuring the sweep similarly to a DC sweep, as shown in Figure 9-22. **Note:** depending on your amplifier’s gain and the anticipated output swing range, the step size and upper limit of the sweep may need to be adjusted.



**Figure 9-23:** Direct plot menu for PSS spectrum results (a) to display the THD as a function of the fundamental amplitude and (b) to display the time-domain waveform of a specific pss sweep point

Once the PSS analysis has been configured and saved to the *test\_diff* setup, run the simulation (either by running the test independently or by running all the simulations in the ADE XL view). We will analyze the output swing by identifying the point at which the THD reaches 5%. To do this, the THD as a function of the differential input voltage first needs to be plotted. Open the Direct Plot menu for the PSS simulation results, and plot the THD vs. differential input voltage by choosing the settings according to what is shown in Figure 9-23(a). After configuring the Direct Plot settings, click on the output voltage net of the open-loop amplifier in the test bench schematic to plot the THD. Make sure that this plot is added to the list of

outputs for the test to make it easier to recreate the plot for the post-layout simulations. The THD plot should look similar to what is shown in Figure 9-24. On this plot, find the differential input voltage at which the THD reaches 5%. Then, we will need to plot the transient waveform from the PSS simulation that corresponds to this input amplitude.

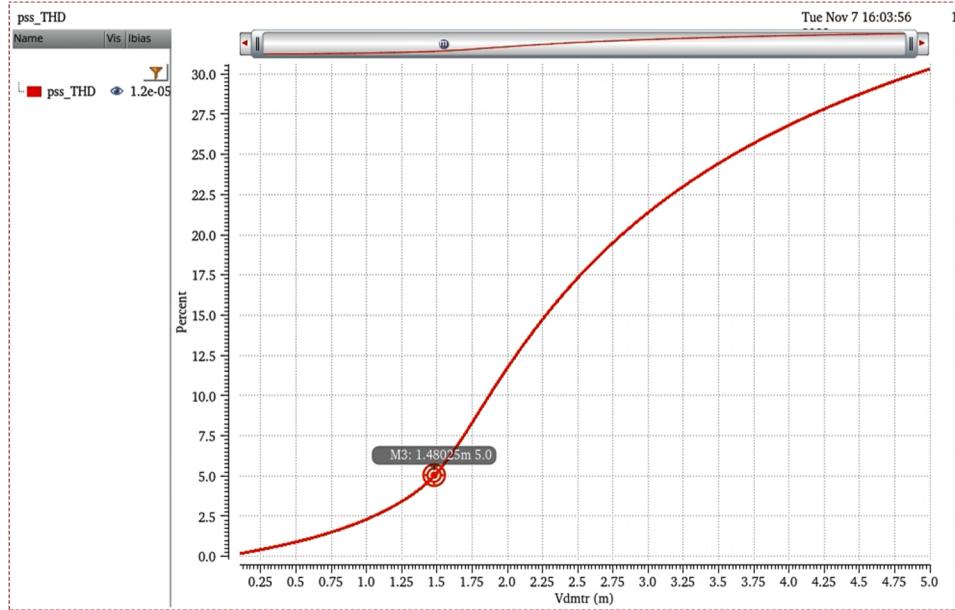


Figure 9-24: Plot of THD vs. Differential input amplitude

Because the sweep of differential input amplitude was performed in discrete steps, we must use the transient waveform associated with the input amplitude that is closest to yielding the 5% THD plot (unless a new PSS simulation is performed with the more precise input amplitude). To plot the transient data from the PSS simulation, return to or re-open the Direct Plot menu and change the options to those shown in Figure 9-23(b). Make sure that the variable value chosen for the input amplitude is changed according to what you find from your own THD plot. Additionally, make sure the waveform will be added to the list of outputs for the test to allow for easier editing of the expression for the post-layout simulations.

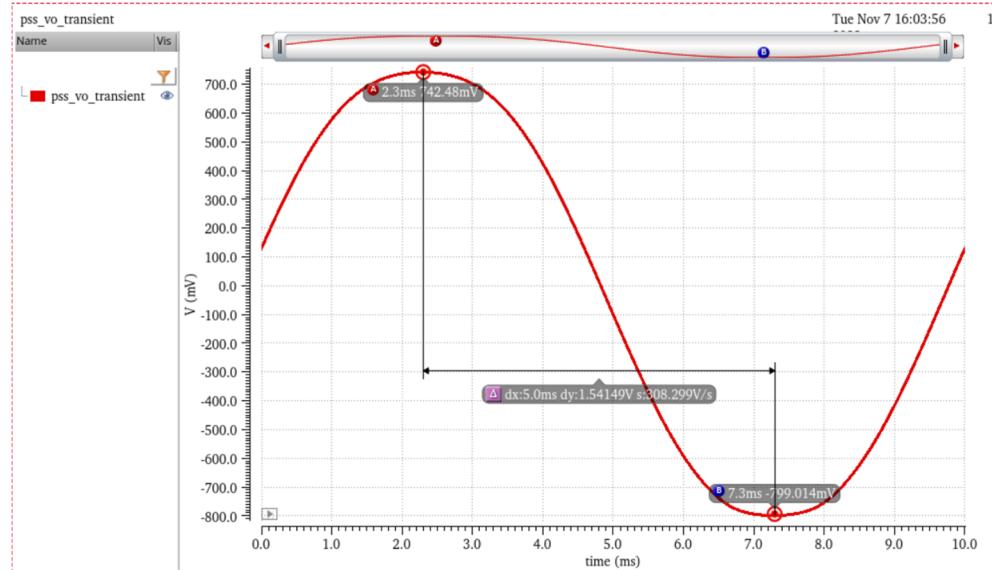


Figure 9-25: Transient output from PSS simulation for output voltage swing measurement

Once the Direct Plot settings are configured, then click on the output voltage net of the open-loop amplifier in the test bench, as previously. A plot that looks like Figure 9-25 should be generated. Place a pair of delta markers to indicate the output swing. **Note:** although additional plots are needed, one advantage to this method of determining the output swings is that no information about the transistors inside the op-amp are needed, allowing the same simulation setup to be used for both pre- and post-layout simulations and only requiring potential changes to default values of design variables.

### Plotting Power Supply Rejection Ratio (PSRR)

PSRR is a measure of the effect of power supply variation on the output voltage, relative to the effect of the amplifier's gain. Because of this, the differential-mode gain of the amplifier is needed to calculate PSRR. For this reason, two new tests will be created, one for evaluating  $\text{PSRR}^+$  and one for  $\text{PSRR}^-$ . Beginning with the former, create a new test in the ADE XL state called *test\_psrrp*.

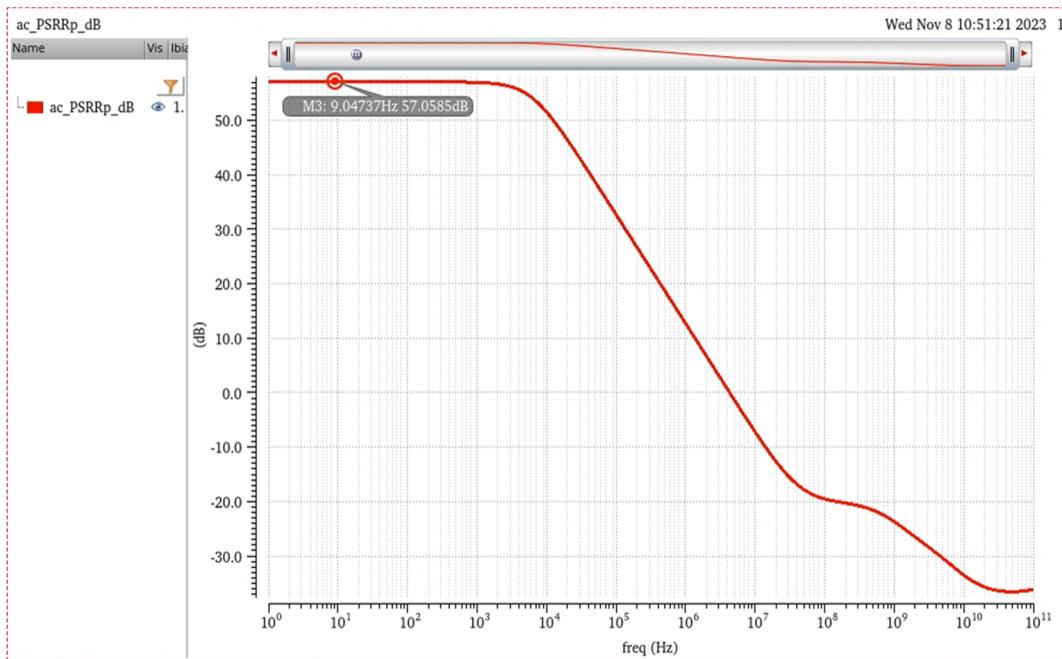
PSRR is calculated as the ratio of the differential gain magnitude to the “supply gain,” or the ratio of  $V_{\text{out}}$  to  $V_{\text{DD}}$  for  $\text{PSRR}^+$  (or  $V_{\text{SS}}$  in the case of  $\text{PSRR}^-$ ). Therefore, we can define these gain responses (in V/V) as

$$A_{dm} = \frac{V_{\text{out}}}{V_{\text{in}}}, \quad A_{VDD} = \frac{V_{\text{out}}}{V_{\text{DD}}}$$

Then  $\text{PSRR}^+$  (in dB) will be calculated as

$$\text{PSRR}^+ = 20 \log \left( \frac{A_{dm}}{A_{VDD}} \right)$$

To plot  $\text{PSRR}^+$ , we will need the differential gain magnitude versus frequency. Because we have already set up *test\_diff* to calculate this response, we can use the *calcVal* function again to incorporate that output from *test\_diff* into our PSRR calculation, as was done for the CMRR calculation. For the *test\_psrrp* setup, we will then need to set the AC magnitude of the differential input to the amplifier equal to zero, and the AC magnitude of the VDD source will be set to 1. Then, run the AC simulation. A plot of  $\text{PSRR}^+$  is shown in Figure 9-26, mark the maximum value for the best-case scenario.

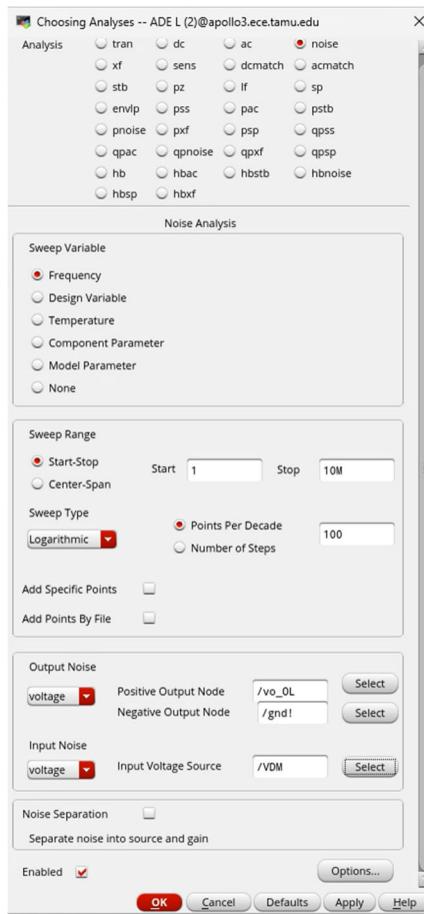


**Figure 9-26:** Plot of  $\text{PSRR}^+$

Now that the setup for *test\_psrrp* is complete, the setup can be copied to create an additional test for evaluating PSRR<sup>-</sup>. Right-click on *test\_psrrp* in the Data View and select “Create Test Copy” from the menu. Once the test copy is created, rename the copied test to *test\_psrrn*. The analysis used in the test will be the same, and most of the design variables and test outputs will be the same as well. However, the AC magnitude of the VDD source will be set to 0 and while the VSS source will be set to 1 in the test setup. The output expression for the supply-to-output gain and PSRR calculations will also need to be changed to use the AC signal at the VSS supply rather than the VDD supply. With the new test setup completed, run the test and plot PSRR<sup>-</sup>, and again mark the maximum value of the plot to indicate the best-case scenario.

## Noise Analysis

A noise analysis allows us to calculate the total amount of noise at the output of a circuit as a result of the components in the circuit. The resulting noise calculation can also be referenced back to the input of the circuit (“input-referred noise”) to more easily compare the circuit noise to the amplitude of an input signal. We will add the noise analysis to the *test\_diff* setup. For this test, open the analysis setup window and select “noise” at the top of the window. Configure the noise analysis settings as shown in Figure 9-27.



**Figure 9-27:** Noise simulation settings

Note the large number of points per decade used in the analysis setup. This will directly influence the accuracy of the calculation as well as the simulation time. For this circuit, 100 points per decade in the logarithmic sweep will not significantly slow down the simulations due to the relative simplicity of the schematics. However, in practice, this number may need to be chosen to reach a compromise in simulation

accuracy versus simulation time depending on the complexity of the circuit being tested. The default options for the output and input noise settings are typically set to utilize “port” sources. These kinds of sources are often used in RF circuit simulations for analyzing the noise figure of RF circuits. For the purposes of this lab, we primarily want to look at the input-referred noise rather than the noise figure metric, so the output and input sources can be set to voltages. For the output noise, the voltage will be the output voltage of the op-amp measured with respect to ground in the circuit. If the op-amp were fully differential (i.e. with differential outputs), then the “Output Noise” nets would be the positive and negative outputs of the op-amp. Only one input source is allowed for noise measurements, so this makes the use of the **ideal\_balun** component (or another means of single-ended-to-differential conversion) necessary in the test bench to get an accurate input-referred noise measurement. After configuring the simulation settings, perform a noise analysis. When the simulation completes, open the Direct Plot menu again and choose the settings shown in Figure 9-28 to plot the input-referred noise power (given in units of  $V^2/\text{Hz}$ ). This option allows for easily converting the plot data into a single root-mean-square (RMS) value of the input-referred noise.

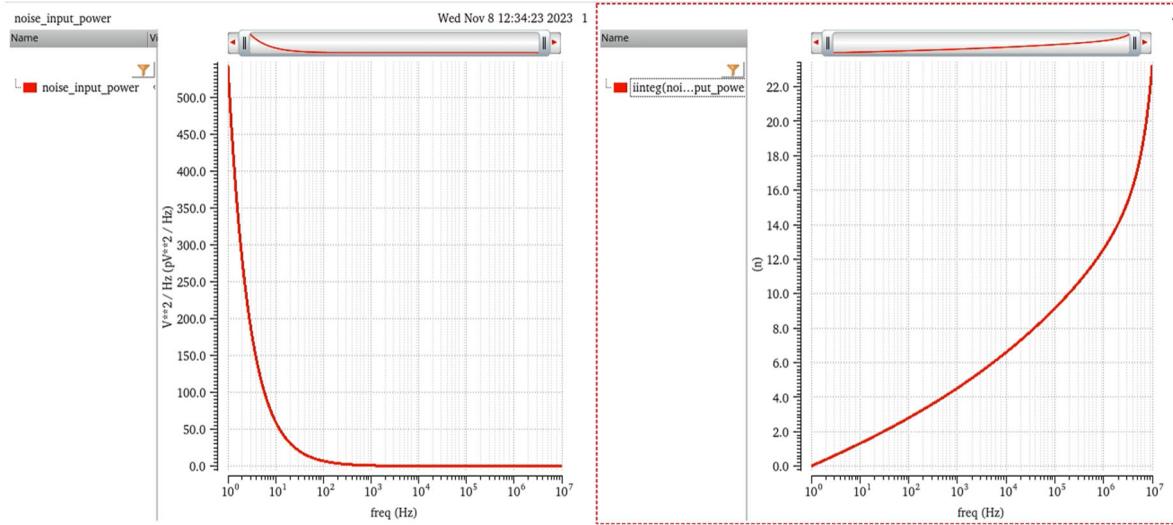


**Figure 9-28:** Direct Plot settings for displaying input noise power vs. frequency

Because we are calculating the noise voltage using the noise power spectral density rather than a time-domain waveform, the RMS noise voltage is calculated according the following formula:

$$V_{n,rms} = \sqrt{\int_{F_1}^{F_2} V_n^2(f) df}$$

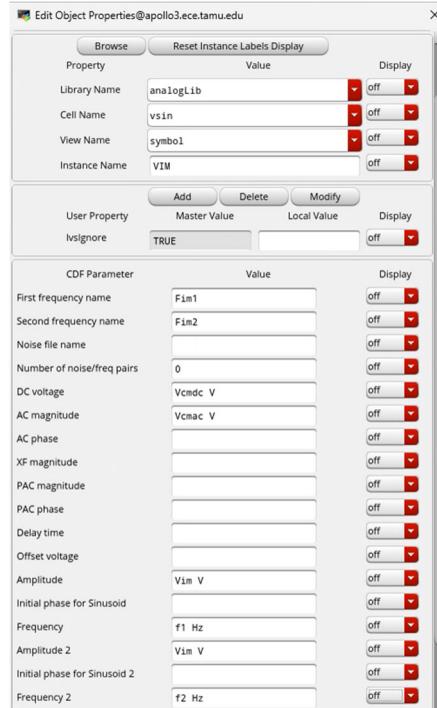
In the above expression,  $F_1$  and  $F_2$  are the minimum and maximum frequencies in the noise analysis and  $V_n^2$  is the noise power as a function of frequency (the plot that appears from the Direct Plot settings shown in Figure 9-28). The *integ* function in the Cadence calculator can be used to calculate a definite integral of a waveform to return a single value as the result of the calculation while the *iinteg* function will plot the indefinite integral, yielding a new waveform. As an example, plots of the noise power spectral density and integrated noise power are shown in Figure 9-29. The RMS noise power can be created as one of the test outputs so that the result is displayed once the simulation is completed.



**Figure 9-29:** Plots of noise power spectral density and integrated noise power vs. frequency

### Nonlinearity Analysis (Intermodulation Test)

The nonlinearity, or intermodulation, test also utilizes a PSS analysis. However, this test will utilize the closed-loop op-amp setup and will apply two tones to the input of the amplifier rather than only a single tone. Because a PSS simulation is already set up for *test\_diff*, we will configure the new PSS simulation as part of the *test\_comm* setup.

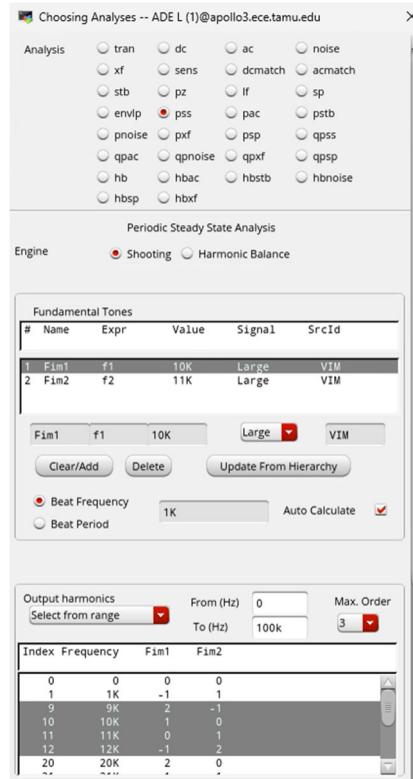


**Figure 9-30:** Parameters of **vsin** source connected to closed-loop op-amp

Before configuring the PSS analysis, there are a few parameters that may need to be added to the test bench schematic. In particular, a few parameters are required in the VIM source connected to the closed-loop op-amp configuration, as shown in Figure 9-30. The “frequency name” parameters at the top of the parameter

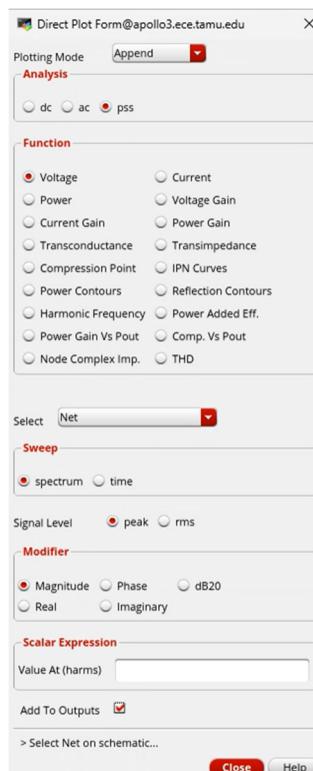
list are used by the PSS simulation for constructing the different intermodulation products (i.e. different frequency components based on the original two input frequencies). The DC voltage is set by the same parameter as the VCM source for the open-loop amplifier setup. In the test configuration, this voltage needs to be set in the middle of the valid common-mode input voltage range of the amplifier. If this DC voltage is skewed, then the output voltage of the closed-loop amplifier will be limited by the common-mode input voltage range limits which can cause greater asymmetric distortion and adversely affect the results for this particular test setup. The AC magnitude also utilizes the same design variable as the VCM source, but we are not examining the output of the closed-loop setup after AC simulations.

The two frequency components of the VIM source produce the two tones needed for the intermodulation test. The two tones should have the same amplitude, so they share the same amplitude parameter, called “Vim” in Figure 9-30. The frequencies of the two tones need to be different, so they are set by the “f1” and “f2” parameters.



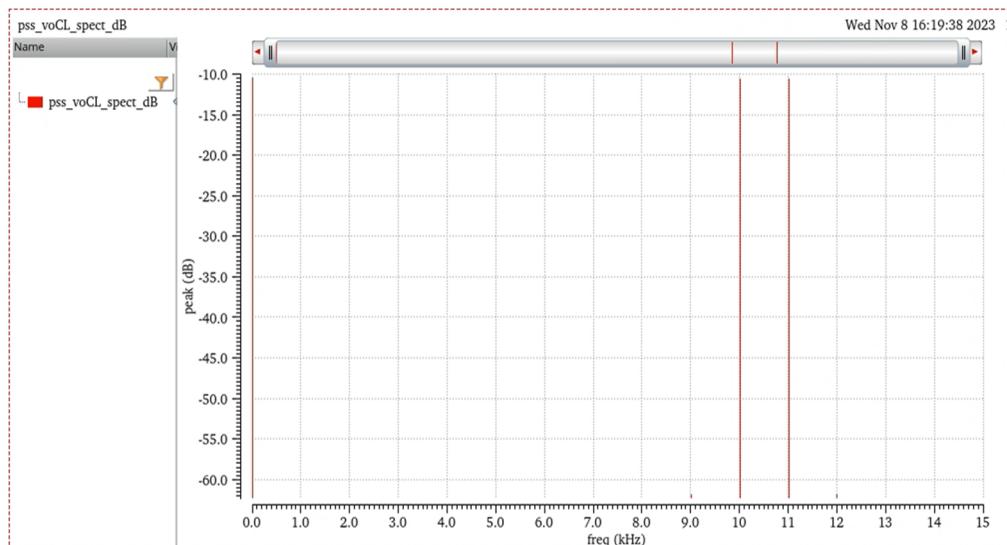
**Figure 9-31:** PSS simulation setup for intermodulation test

Once the source parameters are configured, begin setting up the PSS analysis as part of the *test\_comm* setup. The settings for this PSS analysis are shown in Figure 9-31. They are similar to the PSS simulation configuration shown previously, but now there is a list of harmonic components that must be selected as part of the simulation setup. These harmonics that are selected include the fundamental tones as well as two of the intermodulation products (the two which are nearest to the original two tones). Only a single amplitude is needed for the intermodulation test, so the “Sweep” option of the PSS simulation setup can remain disabled. Once the PSS analysis is configured, click “OK” to add the analysis to *test\_comm* and perform a simulation. **Note:** the PSS simulation may need to be opened and updated if the frequency design variables (f1 and f2) are changed so that the simulator can recalculate the beat frequency. If issues arise with this simulation after changing the variable values, check that the beat frequency is properly updated.



**Figure 9-32:** Direct Plot window for displaying spectrum for intermodulation test

After running a PSS simulation, the frequency spectrum for the output voltage of the closed-loop op-amp setup can be displayed from the Direct Plot menu. The Direct Plot settings shown in Figure 9-32 can be used to plot the frequency spectrum of the output voltage from the intermodulation test. The magnitude can be plotted either in Volts or in dB. Figure 9-33 shows an example plot of the spectrum with the amplitude given in dB.



**Figure 9-33:** Plot of PSS simulation results for calculating IM3

If we allow  $\Delta f = |f_1 - f_2|$ , then the intermodulation tones are the tones in the spectrum at  $f_1 - \Delta f$  and at  $f_2 + \Delta f$  (if  $f_1$  is the lower of the two frequencies). If the amplitudes for the spectrum are given in Volts (linear scale), the third-order intermodulation product can be calculated from the spectrum as:

$$IM3 = \frac{V_{out}(f_1 - \Delta f)}{V_{out}(f_1)}$$

In the above expression,  $V_{out}(f)$  is the value of the output voltage spectrum evaluated at the specified frequency. Because the result of the IM3 calculation can be quite small, depending on the test conditions and the linearity of the circuit, the IM3 result may instead be given in dB.

## Process Corner Simulations

The value of the device parameters, like threshold voltage or mobility, will generally change when manufacturing a circuit. These variations result in changes to the circuit's characteristics, like gain, phase margin, input offset, etc. A more comprehensive method of quantifying the amount of variation that might be expected from these process variations is to use Monte Carlo simulations (described below). However, there is often a smaller subset of conditions that serve as "corner" cases for testing a circuit. These process corner simulations change the device parameter values away from those of the nominal simulation environment to provide a snapshot of how the circuit's operation may change as a result of some of these variations. The results from a Monte Carlo simulation are more rigorous and will generally cover a much wider range of variations, but the process corner simulations are faster to perform since there are fewer cases to look through.

The PDK for the lab contains definitions for specific process corners based on different "skews" from the nominal parameters for the NMOS and PMOS devices. In this case, there are 2 skew cases for each device: fast or slow (not including the "typical" case for each device). Since the skew for each device is independent, there are 4 possible corners for the transistors (additional corners exist to separately account for variations in the parameters of other devices, like resistors or diodes). The corners are also often referred to according to the pair of skews that is used. For example, "fast-slow" would be used to refer to the process corner with faster NMOS devices and slower PMOS devices. However, the skews may not always be entirely independent from each other, e.g. the variations causing a slower PMOS device may still be correlated with a reduced speed of the NMOS devices as well, though to a lesser extent.

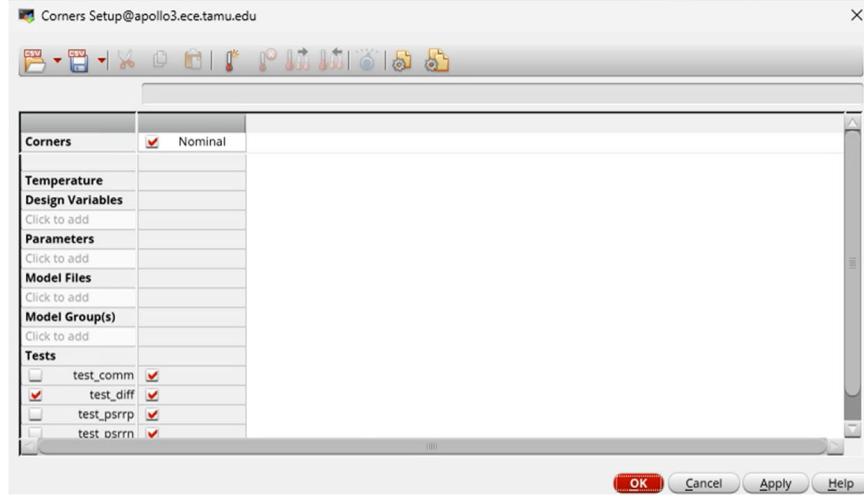
In this lab, we will look at 4 transistor corners and compare the variations with the results seen in a Monte Carlo analysis. The 4 corners we will look at are: **fast-fast, fast-slow, slow-fast, and slow-slow**. We also limit the focus of their effects to the following op-amp characteristics:  **$A_{v0}$ , dominant pole, gain-bandwidth product, and phase margin**. This should limit the simulations to requiring only the *test\_diff* setup. **Note:** for the corner simulations, the noise and PSS analyses can be disabled to reduce the simulation time.

To access and utilize the corner definitions, model files need to be imported into the simulation setup and specific options enabled. To prevent these corner related options from interfering with the default test setups in ADE XL, and to make it easier to select specific corner definitions, the process corner setups are more easily managed with the *Corners* list in the Data View.

To set up the process corners, expand the *Corners* list and add a corner setup with the "Click to add corner" space. The Corners Setup window should appear, as shown in Figure 9-34. To add a new corner to the

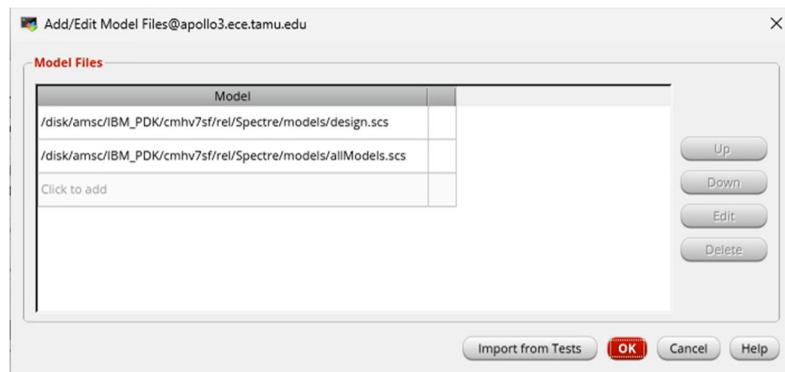
setup, click the  button at the top of the window. A new column should appear in the window with a default name for the corner in the "Corners" row at the top. You can rename the corner by clicking the corner name in its associated column and then clicking the F2 key on the keyboard to edit the field. The corner name should also appear in the space immediately above the corner settings ("Corner Name" should

appear to the left of this space when the corner name is selected) and can be edited in this space as well. Rename the corner to “FastFast” to indicate the process skew that will be used for this setup.



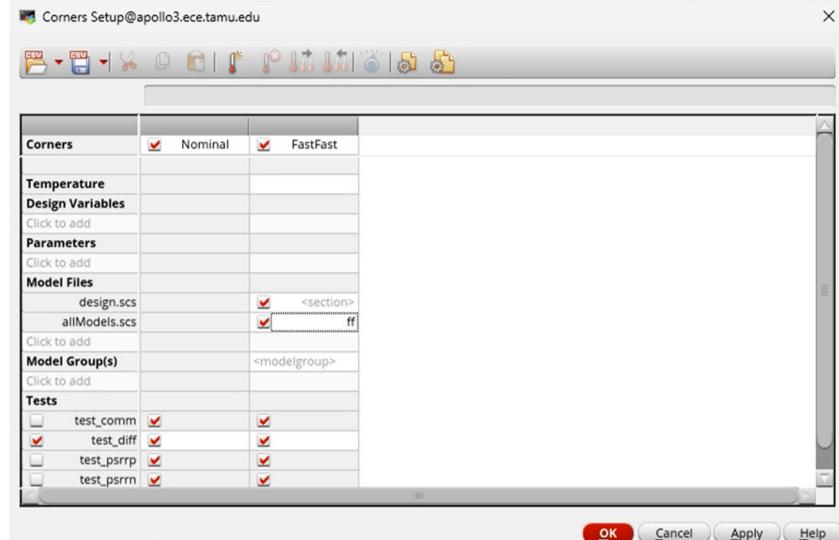
**Figure 9-34:** Corners Setup window

After renaming the corner, click on the “Click to add” space immediately below the “Model Files” heading. A new window will appear in which model files can be imported which are related to the simulation environment or to the devices. Click the “Import from Tests” button at the bottom of this window, as this imports the model files already being used in the default simulation setups, thereby mitigating the need to look for the specific directory containing the files with the process corner definitions. Two rows should be populated with directories pointing to two files for this PDK, one named “design.scs” and the other named “allModels.scs” as shown in Figure 9-35. If this is the case, then click “OK” at the bottom of this window. There should now be two additional rows, one for each of these model files, appearing under the “Model Files” heading in the Corners Setup window.



**Figure 9-35:** Addition of model files to corner setups

For the PDK used in these labs, the “allModels.scs” file contains the process corner definitions. Double-click on the field associated with the allModels.scs file within the FastFast corner settings. An arrow should appear for a drop-down menu. Expand the drop-down menu and select the “ff” option to choose the fast-fast corner. Then, enable both settings in these rows associated with the two model files by clicking the box in the associated fields of the FastFast corner column. The settings for the “design.scs” row can be left empty, but the field must still be enabled. The resulting Corners Setup window should now look something like Figure 9-36.



**Figure 9-36:** Corners Setup window with fast-fast corner added

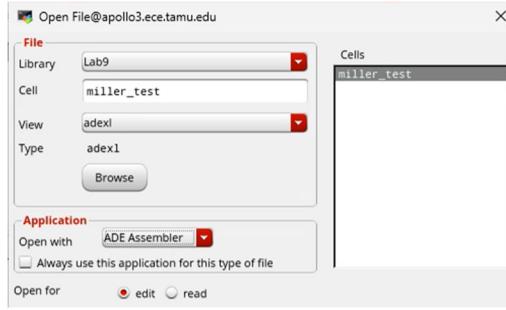
Add three more corners to the Corners Setup window named “FastSlow,” “SlowFast,” and “SlowSlow.” These corners will use the “fs,” “sf,” and “ss” options from the allModels.scs file, respectively. After setting up each of these corners in a manner similar to the FastFast corner, click “OK” in the Corners Setup window to save all of the corner setups to the ADE XL state. Once finished, the *Corners* list should show the Nominal corner (a default that cannot be removed) as well as the 4 new corner definitions from the previous steps. **Remember to save the ADE XL setup after the corner setups are completed.** Make sure that all corners are enabled, and then run a simulation with the *test\_diff* setup. The Results tab should populate with multiple columns, one for each corner, that will be populated with results as each simulation finishes.

### Monte Carlo Analysis

Monte Carlo analysis provides an accurate and powerful method for parametric yield estimation. The principle of Monte Carlo analysis can be defined as the generation of circuit figure-of-merit distributions as a function of statistically varying device model parameters that accurately reflect manufacturing process variations.

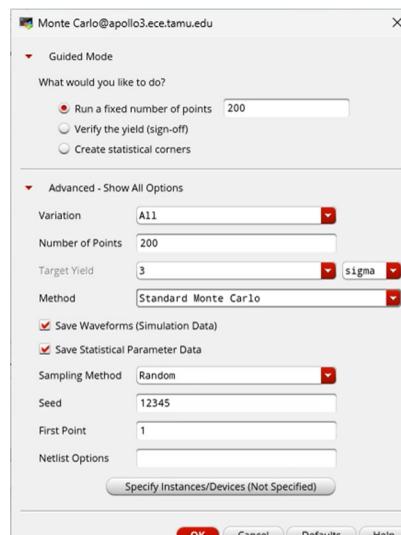
With Monte Carlo analysis, you can generate and save statistical information about a circuit's temperature and geometry dependent performance characteristics. The mathematics supporting Monte Carlo method proves that the probability distribution of the simulated results will be statistically the same as the actual measurements of a real circuit that has been fabricated.

To perform Monte Carlo simulations, the ADE XL setup must be converted to an ADE Assembler setup (a “maestro” cell view). To do this, find the ADE XL cell view in the Library Manager, right-click on the cell view, and select “Open With...” from the menu. A new window will appear where you can select which program to use when opening the ADE XL state. As shown in Figure 9-37, change the “Application” setting to “ADE Assembler” and then click “OK” at the bottom of this window. A new window should appear with a similar user interface to ADE XL. However, the method of setting up an individual test in this window does not resemble an ADE-L-like configuration. But the license associated with ADE Assembler also includes a few additional tools, like Monte Carlo simulations.



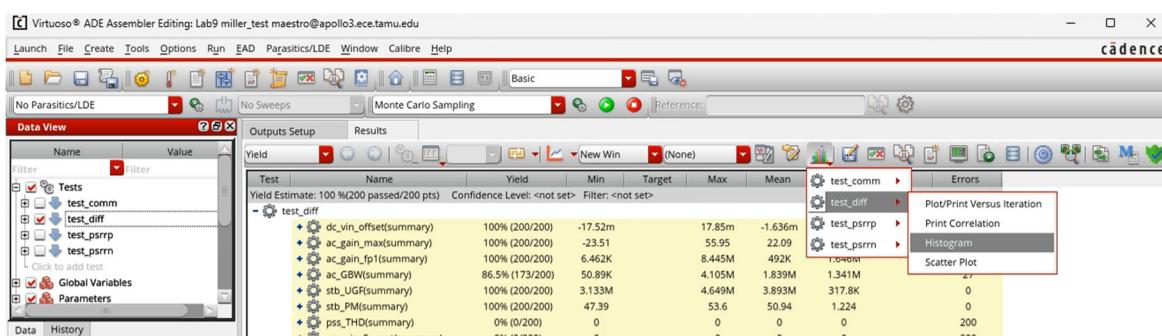
**Figure 9-37:** Opening ADE XL with ADE Assembler

We will perform a Monte Carlo analysis to evaluate the variations in the following parameters: **A<sub>v0</sub>, dominant pole, gain-bandwidth product, and phase margin**. These outputs should already be available from the *test\_diff* setup that has now been imported into the ADE Assembler view. The noise and PSS simulations should be disabled for this test to speed up the Monte Carlo simulations. **Additionally, disable the process corners before beginning the Monte Carlo analysis.** At the top of the ADE Assembler window, go to *Run* → *Monte Carlo Sampling* → *Set options as shown in Figure 9-38* → *OK*. The Monte Carlo options can also be accessed by clicking the button near the top of the ADE Assembler window.

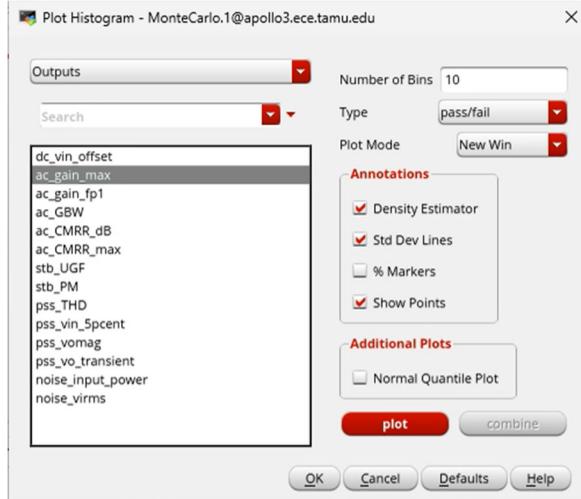


**Figure 9-38:** Monte Carlo settings

Wait until all 200 points of the analysis finish. Once completed, click on the button at the top of the Results tab → *test\_diff* → *Histogram* → select the test output → plot (see Figure 9-39 and 9-40).



**Figure 9-39:** Create Histogram from Monte Carlo simulation results



**Figure 9-40:** Histogram plot selection

## Prelab

Design an operational amplifier of Figure 9-3 to obtain the following specifications:

$A_{v0}$	> 50 dB
CMRR	> 60 dB
GBW	> 2 MHz
PM	> 45°
Output Swing (peak-to-peak)	> 1 V
Load Capacitor	30 pF
Load Resistor	$\infty$ (open circuit)
Power Dissipated	< 500 $\mu$ W (including bias current source)
Power Supply	VDD = -VSS = 0.9 V

## Lab Report

1. Simulate the design from the prelab. Adjust the transistor sizes until all specifications are met. Provide plots of:
  - Open-loop differential gain vs. frequency
  - CMRR vs. frequency (using open-loop amplifier)
  - Loop gain magnitude and phase vs. frequency
  - PSRR<sup>+</sup> vs. frequency (using open-loop amplifier)
  - PSRR<sup>-</sup> vs. frequency (using open-loop amplifier)
  - Output swing (using PSS simulation)
  - Transient response (either open or closed-loop amplifier)
  - Input noise power spectral density from 1Hz to 10MHz (using open-loop amplifier)
  - Intermodulation Test (using closed-loop amplifier)

For the intermodulation test, use two tones, one at 10kHz and one at 11kHz, both with an amplitude of 300mV. For the single-tone PSS simulation to determine the output swing, use a signal frequency of 100 Hz to remain well inside the bandwidth of the open-loop amplifier.

Using an appropriate plot, mark the following measurements and summarize the results in a table. Recall the "M" hotkey for adding markers and the use of “delta markers” (refer to Lab 1) for measuring slope:

- $A_{v0}$
  - Maximum CMRR
  - Gain-bandwidth product
  - Phase margin
  - Power Dissipation
  - Slew rate (rising and falling)
  - Output swing
  - RMS Noise Voltage
  - Third-order Intermodulation product (IM3)
2. Layout your final design using good layout techniques. Remember to include the Miller capacitor as part of the amplifier layout. Include the LVS report (with your NetID and time stamp). Be sure to include parasitic capacitances in your extraction. Perform post-layout simulations and plot and annotate the simulations from part 1. Include an image of the DRC RVE window results filtered to “show unresolved” in the lab report.
3. Perform process corner simulations on the op-amp design. For the nominal simulation (typical corner) and the 4 process corners, tabulate the results of the following parameters:
- a)  $A_{v0}$
  - b) Dominant pole
  - c) Gain-bandwidth product
  - d) Phase margin
4. Run a Monte Carlo simulation on the op-amp design. Be sure to run this simulation with process variation and mismatch. Generate histograms of the following parameters:
- a)  $A_{v0}$
  - b) Dominant pole
  - c) Gain-bandwidth product
  - d) Phase margin

Parts 3 and 4 only need to be completed for the post-layout circuit. Additionally, the outputs for parts 3 and 4 can be found from the stability analysis results (since the feedback factor in the closed-loop setup is unity). Using the closed-loop setup to determine these quantities will mitigate the need to cancel the offset as the input offset voltage changes across the corner and Monte Carlo simulations.

In your discussion, make note of any significant changes to your amplifier design compared with your prelab calculations and justify the modifications. Comment on the impact of process variations and mismatch on each parameter. Additionally, a two-stage Miller op-amp may not be an optimal choice for applications with a large load capacitance, as we have here. Can you think of any alternative amplifier topologies that might be better-suited for such a use case? What is your reasoning for the alternative(s)?