

2011

Supélec

SUPELEC - Department
of Signal Processing and
Electronic Systems

Philippe BENABES and
Catalin-Adrian TUGUI

[SIMECT]

A tool for SImulation and Macromodel Extraction of analog CT functions

Table of Contents

1.	SIMECT Overview	3
1.1.	Tool presentation	3
1.2.	Supported topologies	3
2.	SIMECT configuration	4
2.1.	Prerequisites.....	4
2.2.	Package contents.....	4
2.3.	Environment configuration	5
3.	Design Steps	5
3.1.	Circuit schematic entry and parameterization.....	5
3.2.	Creation of the testbench schematic	6
3.3.	Simulation and macro-model extraction using SIMECT_GUI.....	9
3.3.1.	Interface overview.....	9
3.3.2.	Setting up the SIMECT_GUI simulations	13
3.3.3.	Setting up the multi-machine simulations	13
3.3.4.	Saving and loading SIMECT settings.....	14
3.3.5.	Automatic generation of the simulation sources.....	15
3.3.6.	Simulation of subcircuits and selection of the Netlist/View type	16
3.3.7.	Running analog simulations	17
3.3.8.	Macro-models extraction	18
3.4.	Batch-mode simulations using SIMECT	19
3.5.	Performing optimizations using SIMECT_GUI or SIMECT_PAR	21
4.	Design examples.....	22
4.1.	Unipolar voltage to unipolar voltage source	23
4.2.	Differential voltage to unipolar current source	28
4.3.	Unipolar current to differential voltage source	34
4.4.	Differential current to differential current source	41
5.	Optimization example	50

1. SIMECT Overview

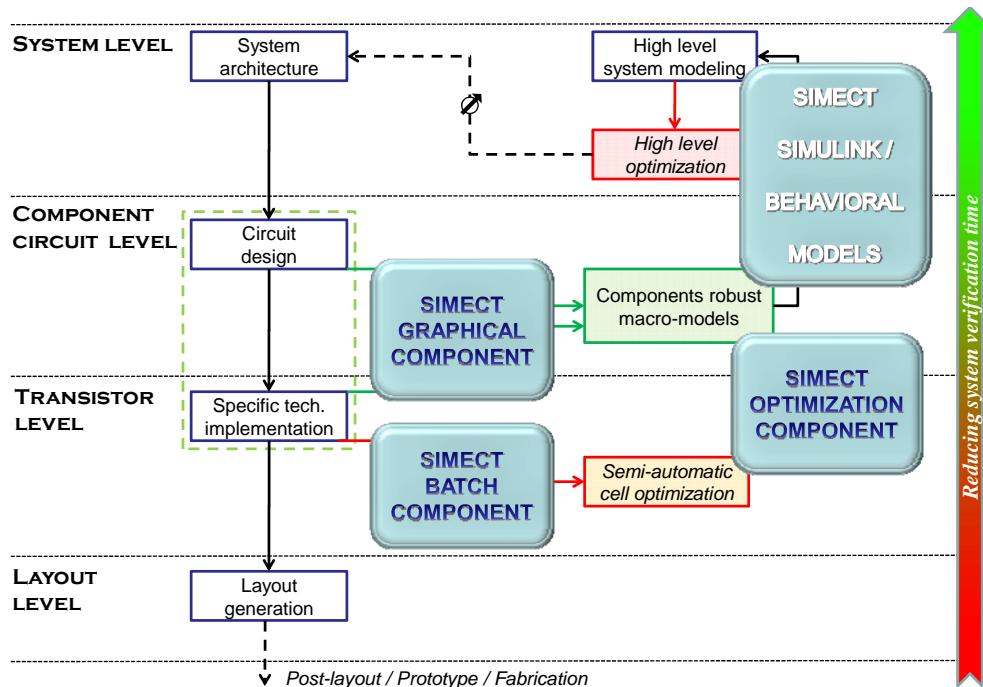
1.1 Tool presentation

SIMECT is a tool intended for the design, simulation and optimization of complex CT functions by using an analog macro-modeling technique.

SIMECT involves a suite of algorithms and programming paradigms including CADENCE Virtuoso Analog Design Environment, MATLAB, SIMULINK, VHDL-AMS, Verilog-A in order to create a very flexible analog design framework.

The tool consists of three major applications, assisting the designer through the entire conception process:

- a user-friendly graphical component (**SIMECT_GUI.m**) intended for initial circuit design, analysis, parametric simulations, high-level exploration and macro-models extraction;
- a batch component (**SIMECT.m**) which can be used for rapid analyses, scripting, further algorithmic development and regressions;
- an optimization component (**SIMECT_PAR.m**) implementing a semi-automatic optimization method for transistor-level components.



1.2 Supported topologies

SIMECT supports any type of unipolar/differential design with the inputs/outputs either as voltages or currents. The circuit topologies are defined starting from the 4 general types of controlled sources:

- Voltage-controlled voltage source (VCVS);
- Voltage-controlled current source (VCCS);
- Current-controlled voltage source (CCVS);
- Current-controlled current source (CCCS);

Each of the 4 types can be implemented with a unipolar/differential input and a unipolar/differential output. The tool supports simple-input-multiple-output designs (up to 3 full differential output ports in **SIMECT_GUI** and an unlimited number in **SIMECT**). The multiple-input-multiple-output designs can be simulated by a convenient decomposition.

2. SIMECT configuration

2.1.Prerequisites

It is recommended to use SIMECT on a Linux Red Hat distribution or SUN Solaris.

In order to function correctly, the following tools are required:

- MATLAB R2010a or higher
- SIMULINK 7.4 or higher
- CADENCE ANALOG DESIGN TOOLS (IC5 or IC6)
 - o Schematic entry
 - o Analog Design Environment
 - o OCEAN (Open Command Environment for Analysis) interface tool
 - o Virtuoso Multi-Mode Simulation (MMSIM) Spectre/RF toolbox
- VHDL-AMS SIMULATOR (for the simulation of VHDL-AMS models)
- VerilogA SIMULATOR (for the simulation of VerilogA models)

2.2.Package contents

A. The SIMECT tool:

- ❖ /SIMECT root folder:
 - the three main scripts used to lunch the components: *SIMECT.m*, *SIMECT_GUI.m* and *SIMECT_PAR.m*;
 - a *pathdef.m* example file;
- ❖ /SIMECT /GUI_SIMECT folder: the routines for the graphical interface of SIMECT
- ❖ /SIMECT /SIMECT_APP folder: the analysis and extraction routines
- ❖ /SIMECT /SIMULINK folder: the functions and sources for SIMULINK models extraction
- ❖ /SIMECT /VHDL folder: the functions and sources for VHDL-AMS and VerilogA models extraction

B. Cadence design examples:

- ❖ /CADENCE_EXP folder: contains four design examples each with the following structure
 - circuit schematic
 - schematic cellview
 - symbol cellview
 - testbench schematic
 - /Sim repository - includes the Spectre netlist of the testbench and the /variables repository including a variant file which can be used in conjunction with the provided schematic

C. SIMECT Documentation: /DOCUMENTATION folder

2.3.Environment configuration

The following file [.profile](#) gives as an example the variables to be set in order to use correctly all the required tools, especially the following path to the .dll libraries used by the Matlab importation routines (replace '/home/mmsim' by the MMSIM Spectre/RF tool location):

```
export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:/home/mmsim/tools/lib/64bit  
for 64-bit systems,
```

```
export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:/home/mmsim/tools/lib  
for 32-bit systems
```

The OCEAN (Open Command Environment for Analysis) interface tool resides into the dfII Cadence repository, which can be included as:

```
export  
PATH=$PATH:$CDS_ROOT/tools/bin:$CDS_ROOT/tools/dfII/bin:$CDS_ROOT/tools/plo  
t/bin
```

The Matlab path should include the following directories (the *pathdef.m* example can be edited):

```
/home/mmsim/tools/lib/64bit:  
/home/mmsim/tools/spectre/matlab/64bit:
```

for 64-bit systems or

```
/home/mmsim/tools/lib:  
/home/mmsim/tools/spectre/matlab:
```

for 32-bit ones.

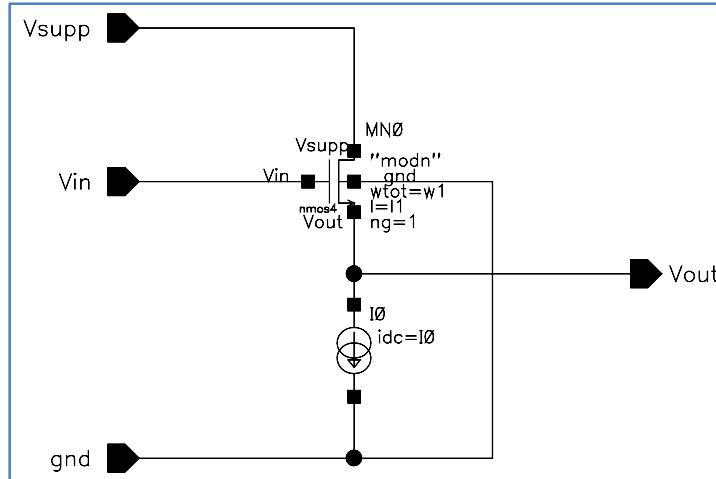
The Matlab path should include also the SIMECT components:

```
.../SIMECT:  
.../SIMECT/GUI_SIMECT:  
.../SIMECT/SIMECT_APP:  
.../SIMECT/SIMULINK:  
.../SIMECT/VHDL:
```

3. Design Steps

3.1.Circuit schematic entry and parameterization

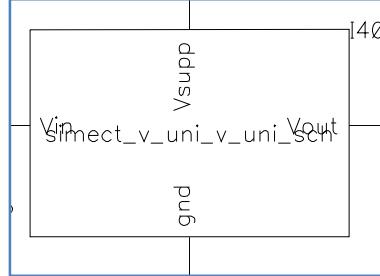
In order to use the SIMECT framework, the first step is the creation of the circuit schematic using the CADENCE Analog Design Schematic editor. The schematic is created directly into the required technology and includes only the circuit components, the inputs and outputs as IN/OUT ports. No excitation sources are required during this step. A schematic example for a simple voltage follower is presented:



Here, the input is a voltage (V_{in}) and the output a voltage (V_{out}), while V_{supp} is used to connect the supply source and gnd the circuit ground.

The schematic design variables will be parameterized with comprehensive names which will be used by SIMECT afterwards. In our example, the transistor width is edited as $w1$, its length $l1$ and the polarization source $I0$.

Once the schematic is designed and parameterized, a symbol cellview can be generated by using *Design->Create Cellview->From Cellview...*:



3.2. Creation of the testbench schematic

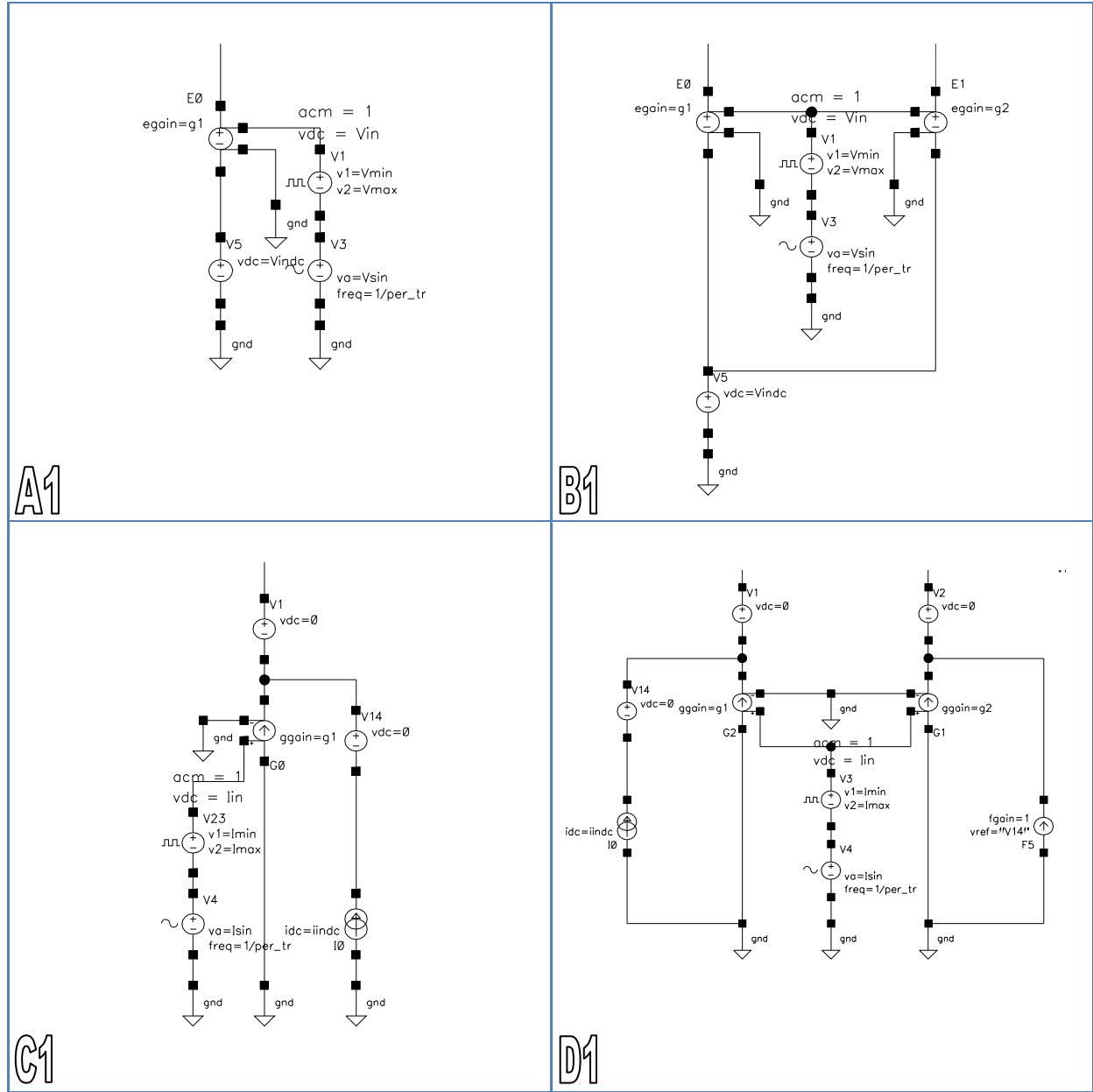
During this step, a test schematic is created starting from the circuit schematic.

The symbol cellview of the circuit will be used. The signal sources needed to simulate the circuit are added on each input and output of the circuit. The simulation sources can be manually added or automatically generated. This section presents the manual process while the automatic generation option is presented in 3.3.5.

Depending on the circuit topology, the input and output simulation sources should be chosen from 4 different types of composite sources:

- ❖ on input:
 - if the input is a unipolar voltage, a composite unipolar voltage source is used (A1)
 - if the input is a differential voltage, a composite differential voltage source is used (B1)

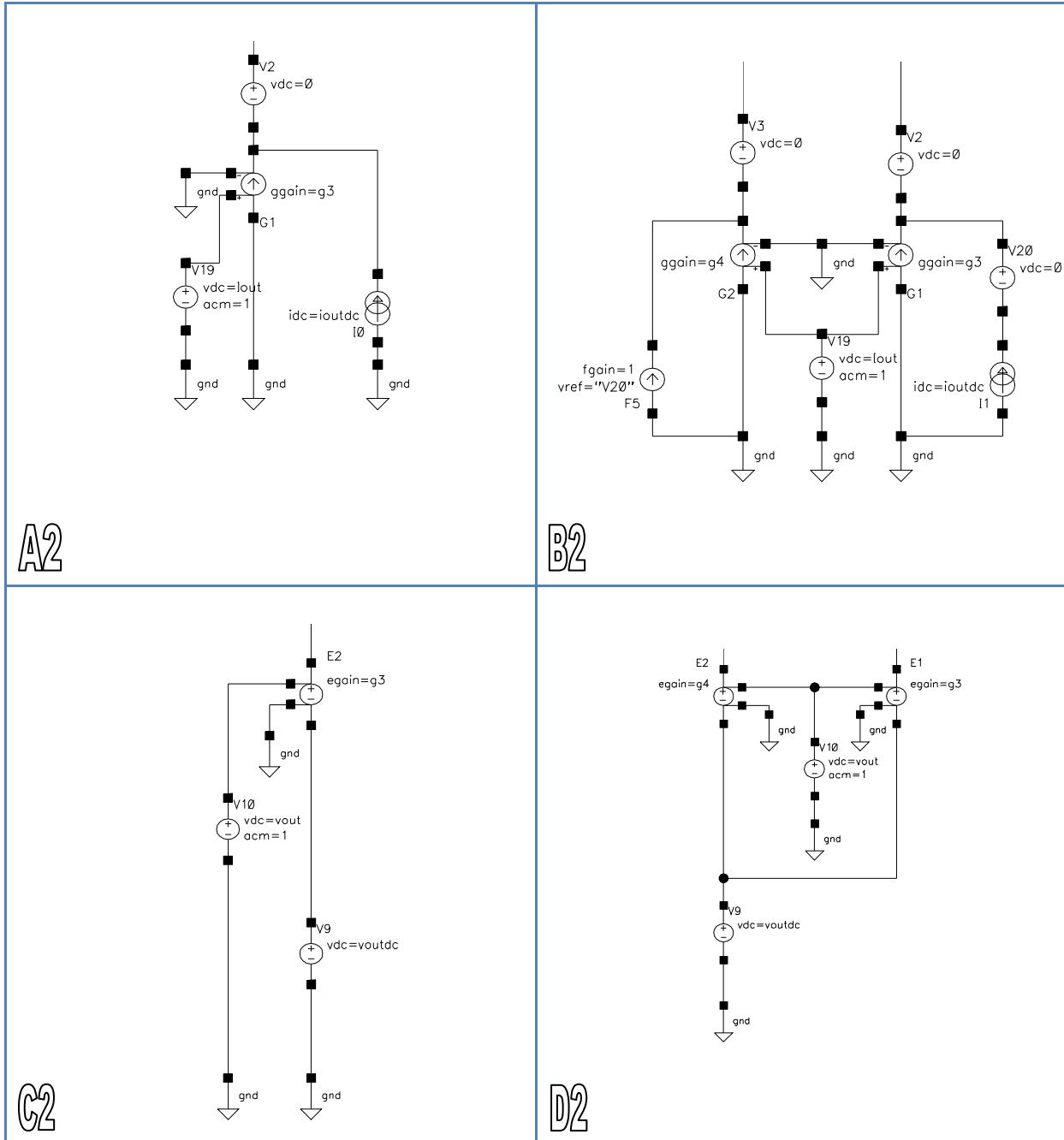
- if the input is a unipolar current, a composite unipolar current source is used (C1)
- if the input is a differential current, a composite differential current source is used (D1)



- ❖ on output:
- if the output is a unipolar voltage, a composite unipolar current source is used (A2)
 - if the output is a differential voltage, a composite differential current source is used (B2)
 - if the output is a unipolar current, a composite unipolar voltage source is used (C2)
 - if the output is a differential current, a composite differential voltage source is used (D2)

This configuration of sources is imposed by the specific analyses to be performed for macro-models extraction and will be further discussed in the next sections. The main idea is to use one or two controlled sources on each side, function of the unipolar / differential topology which can be commanded by the gains gi .

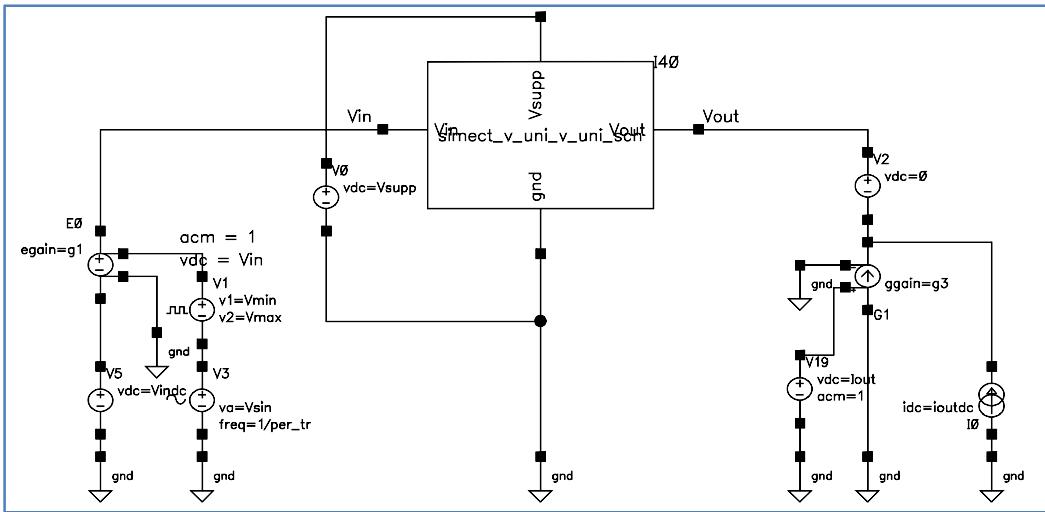
The sources schematics should be also parameterized with variables names which will be used by SIMECT.



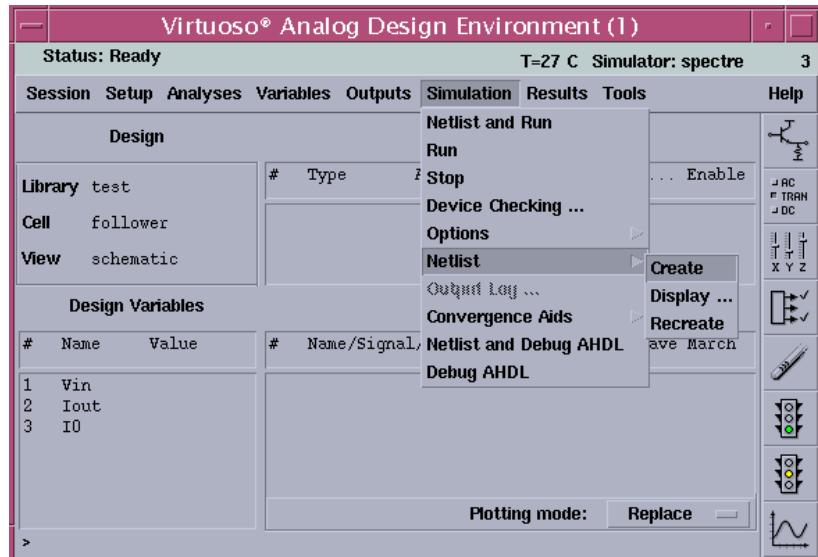
Reconsidering the voltage follower example, we can identify on the schematic the unipolar voltage input requiring the usage of A1 source and the unipolar voltage output requiring the A2 source.

As supply source, a VDC source is used and parameterized with the DC value $Vsupp$.

The testbench schematic is created by incorporating the symbol cellview and the three types of sources:



Once the testbench schematic is obtained, its [netlist](#) should be generated from the Virtuoso Analog Design Environment using Netlist -> Create:



Once the netlist created, the Cadence Environment is no more needed and can be closed.

3.3.Simulation and macro-model extraction using SIMECT_GUI

3.3.1. Interface overview

The graphical interface of SIMECT is launched in MATLAB by running the SIMECT_GUI.m script.

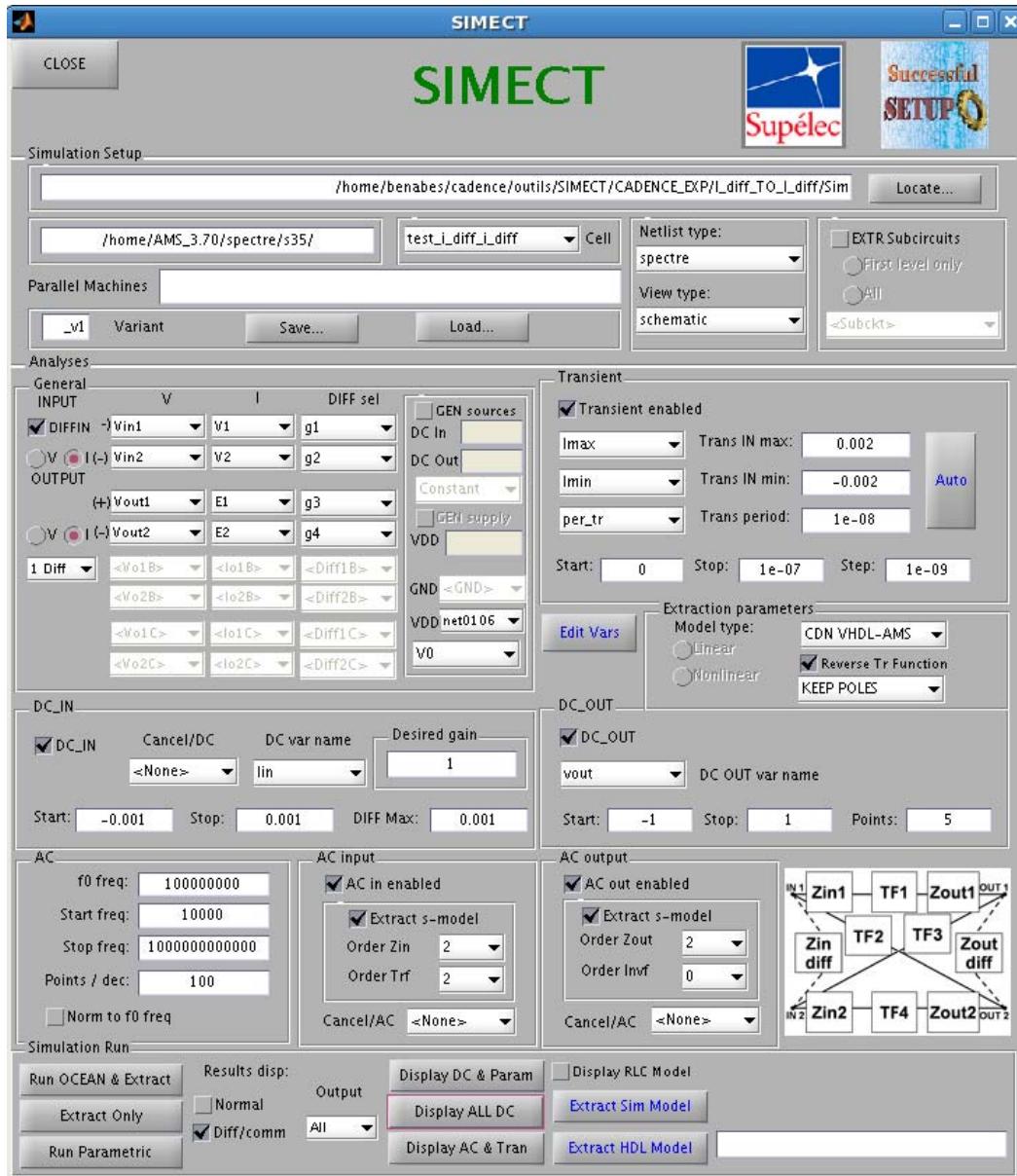
The main window is presented bellow and its three sections should be used in descendent order:

- *Simulation Setup* permits the following operations:
 - selection of the Cadence simulation repository: [Locate...];
 - selection of the cell to be simulated: [Cell];
 - display of the technology components repository: [Technology];
 - definition of the machines for parallel simulations: [Parallel machines];
 - saving and loading of the SIMECT settings for a specific circuit: [Save...], [Load...];
 - definition of different variants for the save/load operation: [Variant];

- selecting the netlist type and view type for the circuit to be simulated: [*Netlist type*], [*View type*];
- selecting to extract subcircuits, the subcircuit level and name of subcircuit: [*EXTR Subcircuits*], [*Level*], [*Subckt*];
- *Analyses* contains 8 subsections allowing the following operations:
 - *General*
 - definition of the input and output type: [*V*, *I*];
 - definition of the voltage nets representing the input(s) and output(s): [*V(+)*, *V(-)*];
 - definition of the sources for currents measurement representing the input(s) and output(s): [*I(+)*, *I(-)*];
 - definition of the unipolar/differential behavior on the input and output: [*DIFF IN*], [*DIFF OUT*] selected when the design is differential on input/output and [*DIFF sel (+), (-)*] for the selection of the differential variables of the simulation sources; The popup menu associated to the differential select on output will select up to 3 unipolar/differential outputs;
 - editing the design variables previously defined in Cadence: [*Edit Vars*];
 - selection to automatically generate the simulation sources: [*GEN Sources*]; when the sources generation is active:
 - the input and output DC values should be specified: [*DC In*], [*DC Out*];
 - selection of the transient source signal: [*Constant*], [*Square*], [*Sinus*];
 - the supply source can also automatically be generated by [*GEN Supply*];
 - the supply source voltage will be defined as [*VDD Val*];
 - the ground net of the circuit will be defined as [*GND*];
 - selection of the supply source voltage net and current measurement: [*VDD*];
 - *Extraction parameters*
 - editing the design variables previously defined in Cadence: [*Edit Vars*];
 - selecting the model behavior template (linear/nonlinear) – not yet available;
 - selecting the behavioral model type: VHDL-AMS for Cadence AMS (CDN VHDL-AMS), VerilogA for Cadence AMS (CDN VerilogA), VHDL-AMS for Dolphin Integration SMASH (SMASH VHDL-AMS);
 - enabling/disabling the reverse transfer functions for the models: [*Reverse Tr Function*];
 - remove or keep the out of band singularities (singularly or in pairs);
 - *Transient*
 - enabling/disabling the transient analysis: [*Transient enabled*];
 - definition of the design variables representing the minimum/maximum and the period of the transient signal: [*Trans IN max*], [*Trans IN min*], [*Trans period*];
 - definition of the start, stop and step time for transient simulation: [*Start*], [*Stop*], [*Step*];
 - automatic generation of the *Transient* values starting from DC, AC or design variables [*Auto*];
 - *DC_IN*
 - enabling/disabling the DC input analysis: [*DC_in enabled*];

- definition of the design variable for which to perform the DC analysis: *[DC var name]*;
 - specification of a component to be disabled when performing the DC input analysis: *[Cancel/DC]*;
 - definition of the start value and stop value for DC analysis: *[Start], [Stop]*;
 - definition of the maximum value for DC when in differential mode: *[DIFF Max]* (the analysis will be performed between $-DIFF\ Max$ and $DIFF\ Max$);
 - definition of DC desired gain: *[Desired gain]*;
- *DC_OUT*
 - enabling/disabling the DC analysis on output: *[DC_OUT enabled]*;
 - definition of the design variable for which to perform the DC output analysis: *[DC OUT var]*;
 - definition of the start value, stop value and number of points for DC output analysis: *[Start], [Stop], [Points]*;
- *AC*
 - definition of an f_0 frequency of interest: *[f0 freq]*;
 - definition of the start frequency, the stop frequency and the number of points per decade when running an AC analysis: *[Start freq], [Stop freq], [Points/dec]*;
 - specification if the s-models characteristics are normalized to the f_0 frequency: *[Norm to f0 freq]*;
- *AC input*
 - enabling/disabling the AC analysis for the input side (for direct transfer function and input impedance extraction): *[AC in enabled]*;
 - enabling/disabling the s-models extraction for the direct transfer function and the input impedance): *[Extract s-model]*;
 - when the s-models extraction is enabled, specification of a maximum model order for the direct transfer function *[Trf]* and the input impedance *[Zin]*;
 - specification of a component to be disabled when performing the AC input analysis (e.g. a capacitor, when applicable): *[Cancel/AC]*;
- *AC output*
 - enabling/disabling the AC analysis for the output side (for inverse transfer function and output impedance extraction): *[AC out enabled]*;
 - enabling/disabling the s-models extraction for the inverse transfer function and the output impedance: *[Extract s-model]*;
 - when the s-models extraction is enabled, specification of a maximum model order for the inverse transfer function *[Invf]* and the output impedance *[Zout]*;
 - specification of a component to be disabled when performing the AC output analysis(e.g. a capacitor, when applicable): *[Cancel/AC]*;
- *Simulation Run* permits the following operations:
 - *[Run OCEAN & Extract]* will run the analog simulator and extract the analyses results;
 - *[Extract Only]* will extract the analyses results saved from a previous simulation (the results should be present in the simulation repository);
 - *[Run Parametric]* is used to run macro-parametric analyses used for semi-automatic optimizations;

- [Display DC & Param] will display the extracted and computed DC and Parametric DC results;
- [Display AC & Tran] will display the extracted and computed AC input/output and transient results;
- [Display ALL DC] will display all the intermediary DC voltages and currents in the circuit;
- [Extract Sim Model] will extract the SIMULINK macro-model of the cell;
- [Extract HDL Model] will extract the VHDL-AMS or VerilogA macro-model of the cell;
- selection if the results are displayed for each separate output or in common/differential mode: [Results disp] and selection of the output [Output];



The main window contains also three indications, displaying useful information during different stages of simulation:

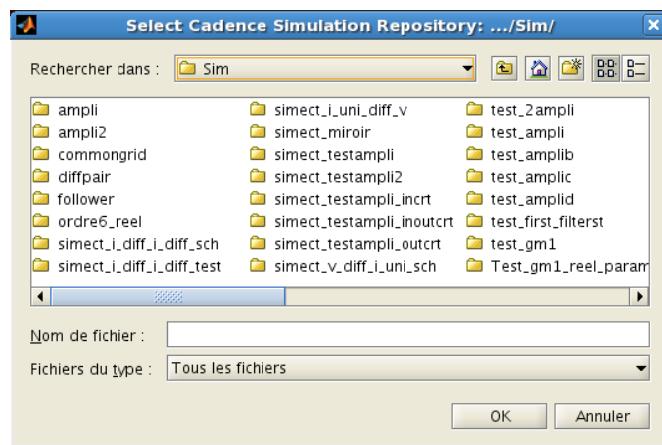
- the setup indication will change from *Simulation SETUP* to *Successful SETUP* when all the steps in the first section of SIMECT are completed;

- the operations indication will display each in-progress operation performed by SIMECT: *Running OCEAN, Extracting data, Extracting VHDL-AMS Model*, etc.;
- Important note:** When the operations indication is active and a specific operation is in-progress, it is not permitted to modify the controls of the main window and/or design variables. Doing so could result in program malfunctions and erroneous results.
- the subsystem structure (bottom-right region of the Analyses section) is displayed after a set of analyses is successfully performed.

3.3.2. Setting up the SIMECT_GUI simulations

The first step for the SIMECT_GUI simulation setup is the localization of the simulation ('/Sim') repository containing the Cadence cells netlists.

Pressing the *Locate* button on the interface will result in a pop-up window where one can browse for the respective repository:



Please note that once a valid simulation repository is selected, it will be displayed in the textbox preceding the button and the available cells will populate the *Cell* menu.

Next, the valid cells are automatically extracted from the Cadence simulation repository and the user has only to select the one to be simulated. By selecting a cell, its technology is displayed in the respective textbox and the *Successful SETUP* indication is presented.

3.3.3. Setting up the multi-machine simulations

SIMECT can use a single machine or multiple machines in parallel in order to accelerate simulations.

By default, all the simulations are performed on the local host. In this case, the *Parallel machines* field will remain empty.

When running in multi-machine mode, SIMECT uses the SSH protocol for the connections on the parallel hosts. In order to be able to run multi-machine simulations, the following operations should be manually performed before using SIMECT:

- 1) generation of authentication keys for ssh authentication. DSA keys can be used, generated with:

```
ssh-keygen -t dsa
```

- 2) copying of the generated public keys under the authorized keys list for ssh:

```
cp .ssh/id_dsa.pub .ssh/authorized_keys
```

- 3) establishing a simple ssh connection on each machine to be used and confirm the host should be added to the list of known hosts:

```
ssh <hostname> ls
```

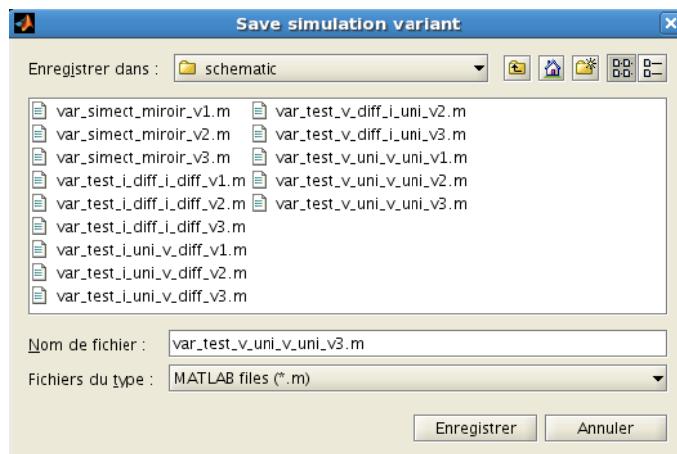
Steps 1) and 2) have to be performed once on the local host where SIMECT is installed. Step 3) has to be performed on the local host for all the other hosts to be used for parallel simulations.

Up to 3 additional machines for parallel simulations can be used on a SIMECT simulation session. The hosts can be changed from one session to another. The names of the hosts should be added in the *Parallel machines* field, separated by ";" and no other symbol should be used before or after the list of hosts.

3.3.4. Saving and loading SIMECT settings

In order to facilitate the recursive simulations during the initial circuit design stages, one can save the SIMECT settings as variants (.m files) and reuse them on successive SIMECT sessions. The field variant in the interface permits to associate multiple files with a single circuit. This can be useful when different design configurations are used in conjunction with the same schematic.

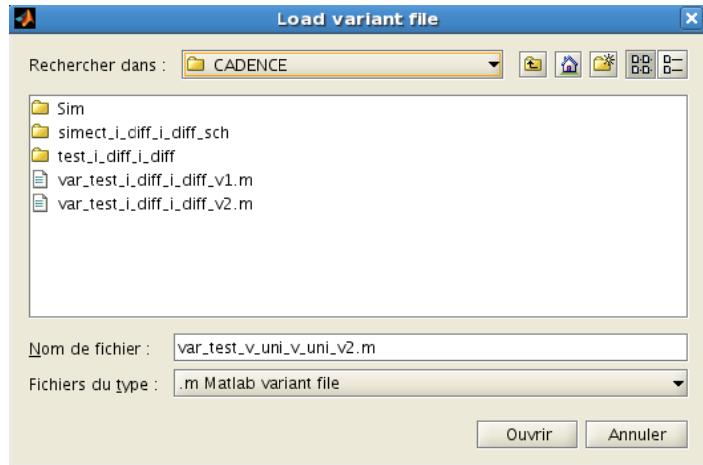
In order to make a Save operation, filling in the variant field and pressing *Save* will activate:



The default name of the variant file will be *var_<cell_name>_<variant>.m* (e.g. [var_test_v_uni_v_uni_v1](#) for the considered voltage follower).

The save operation can be initiated once a specific cell is selected and all the required fields in the activated analyses are filled.

In order to make a Load operation, the *Load* button will activate:



As SIMECT variants are cell-specific, a load operation can be initiated once the Simulation setup is performed and the *Successful SETUP* indication is displayed.

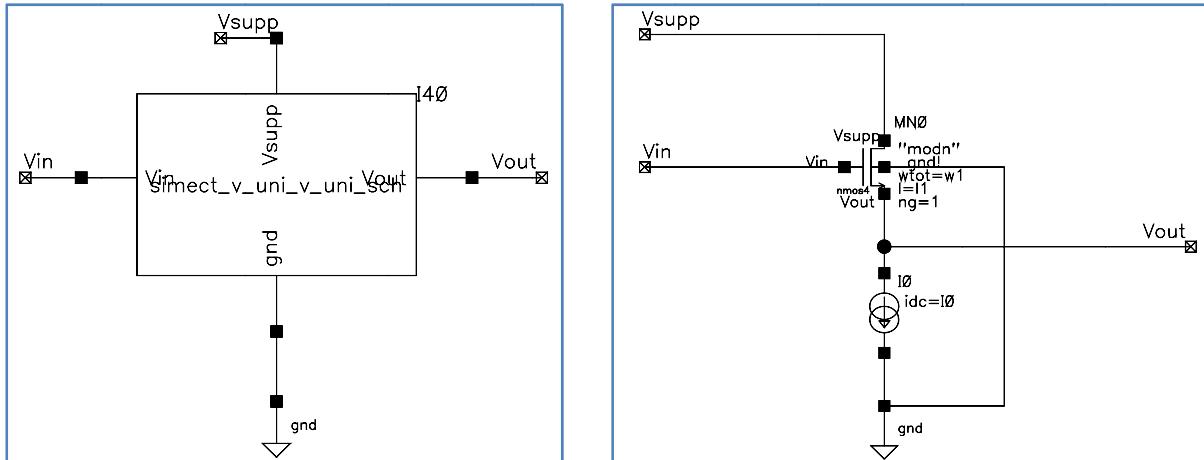
A successful Load operation will result in populating the Graphical Interface with the prior saved settings, while an unsuccessful Load will display an error message and only the available found settings are applied to the interface.

3.3.5. Automatic generation of the simulation sources

The design process can be simplified by using the automatic generation option for the simulation sources. In this case, the A_i and B_i sources types will be automatically selected and added by SIMECT to the design schematic and the user will only select the positions (nets) to connect them.

For this option, no testbench schematic is required and SIMECT will use directly the circuit schematic, which should be edited as specified in 3.1.

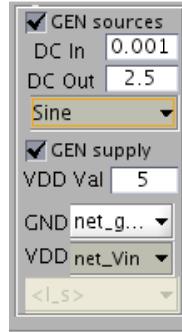
Next, the nets representing the inputs and outputs of the circuit are parameterized with comprehensive names and remain unconnected. The netlist of the schematic will be generated in CADENCE as discussed.



One can use the symbol cellview or directly the schematic cellview of the circuit.

Next, in SIMECT_GUI the simulation repository is located and the cell is selected. Once the *Successful SETUP* indication is presented, the user will select *GEN sources* in the General tab. This will unlock all the controls needed for sources generation.

The DC values on the input (*DC In*) and output (*DC Out*) should be specified. These values can be voltages or currents, depending on the circuit topology and should be filled as actual value without units. When multiple outputs are selected, multiple values for the *DC Out* can be specified as a vector of values.



Next, the user will select the transient source type. Three types of transient signals can be accessed via the pop-up menu:

- *Constant* - the transient signal on the input will be the constant DC input value;
- *Square* - the transient source will be a square generator defined by the parameters in the Transient Tab;
- *Sine* - the transient source will be a sine generator defined by the parameters in the Transient Tab;

The supply source can also be generated by using the *GEN supply* option. Selecting the checkbox will activate the VDD Val field where the voltage value of the supply source should be filled.

A ground net can be specified with *GND*. Leaving it default (<None>) will not add a ground to the circuit.

The aim of the voltage nets and sources for current measurement in the General Tab will change from manual source adding to automatic generation. In the first case, the voltage nets representing the input(s) and output(s)[*V(+)*, *V(-)*]will indicate the actual input(s) and output(s) where the sources are connected by the user, while for automatic generation, the voltage nets will indicate the places where the sources should be added by SIMECT.

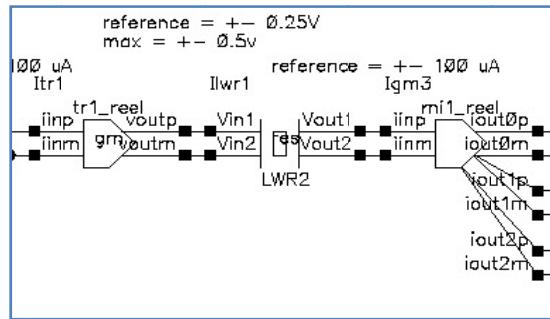
Please note that in the case of generated sources it is not necessary to define the sources for current measurement so the [*I(+)*, *I(-)*] fields are inactive.

Furthermore, the *DIFF sel* indications and the variables for DC, PARAM and Transient are not necessary so the respective fields are inactive.

3.3.6. Simulation of subcircuits and selection of the Netlist/View type

When simulating large designs (multi-stage amplifiers, Sigma-Delta filters, etc), an important feature offered by SIMECT is the possibility of simulation for subcircuits.

Supposing a multi-stage filter like:



It is possible to select the option *EXTR Subcircuits* on the SIMECT GUI and then to select separately one subcell of the design (e.g. tr1_reel, LWR2, mi1_reel) on which we can do all the operations as for a normal circuit (analyses, model extraction, multi-parametric simulation).



The options associated with the subcircuits simulation are:

- *First level only* – allows to extract the first descendent level in the design hierarchy;
- *All* – allows to extract all the subcells which are instantiated in the hierarchy;
- *Subcircuits popupmenu* – will select which subcell to simulate;

Important note: When simulating subcircuits, the sources should be auto-generated as presented in 3.3.5.

SIMECT allows the simulation of different netlist configurations: spectre, spectreVerilog, etc. and also different views of these netlists: schematic, config, etc. The popupmenus on the GUI permit to select between them. Furthermore, these options can be associated with the circuits or subcircuits simulations.

3.3.7. Running analog simulations

The circuit input, output and supply characteristics should be specified in the *General* subsection of the interface. The sources will be manually added or automatically generated.

Afterwards, by pressing the *Edit Vars* button, the design variables can be modified (using only the first column):



Once this step is accomplished, the user will press *Save* and return to the SIMECT main window.

Then he can select the analog simulations to be performed. All the specific fields for an analysis description (e.g. *Start*, *Stop*, *Points*, etc.) should be filled.

Please note that the fields under a specific analysis are only available when the respective analysis is enabled.

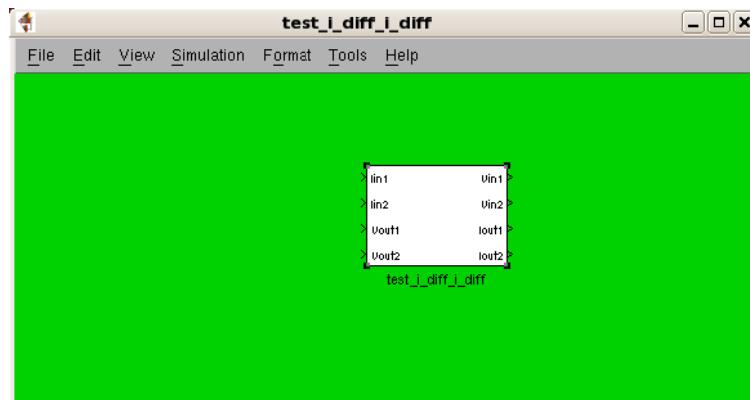
The SIMECT simulation will be run either by *Run OCEAN & Extract* or *Extract Only*, depending if the analog simulator is invoked or previous analog results are used.

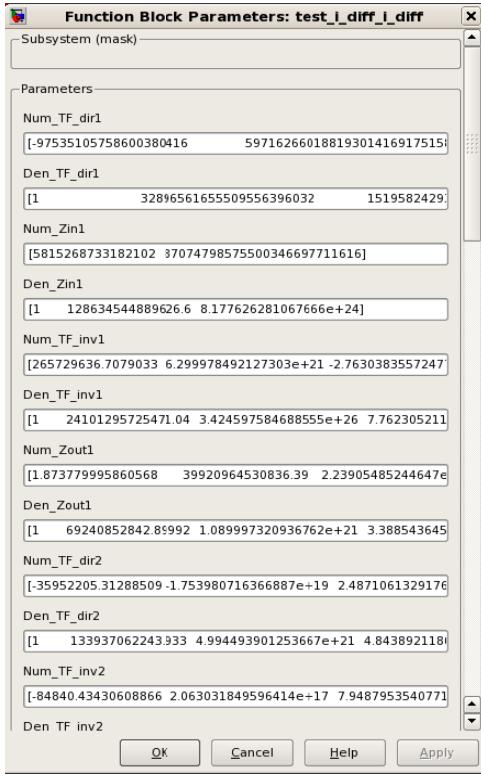
3.3.8. Macro-models extraction

SIMECT allows the extraction of two types of macro-models for the analog designs: as SIMULINK blocks or behavioral modules (type VHDL-AMS or VerilogA).

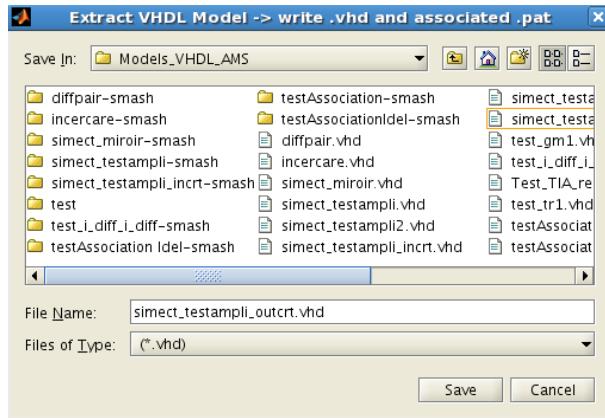
The models can be extracted only if the AC input, AC output, DC and Parametric DC analyses are active.

Pressing the *Extract Sim Model* button results in a SIMULINK window containing the masked model:





Pressing the *Extract HDL Model* button results in an exporting window which allows saving the .vhd and .pat files for the VHDL-AMS model and the .va file for VerilogA:



A textbox associated with the *Extract HDL button* shows the path of the VHDL-AMS/VerilogA file extracted. It allows keeping the same file for successive extractions of models, so the extraction procedure will only overwrite the previous model.

3.4. Batch-mode simulations using SIMECT

Batch mode simulations for a circuit can be launched in MATLAB by running the [SIMECT.m](#) script.

SIMECT.m extends the functionality of *SIMECT_GUI.m* with two important features:

- It can simulate an unlimited number of subcircuits; simulations are performed sequentially for each subcircuit;
- For each design, the number of unipolar/differential outputs is unlimited;

In this case the main program *SIMECT.m* must be manually parameterized to configure and indicate what kind of simulations and extractions must be performed.

- First of all the schematic name, subcircuits, the variant and the simulation directories

```
%Model parameters
model_par.cell='test_gm1';                                % name of schematic
model_par.variant='_v1';                                  % name of variant
model_par.simrep='/home/cadence/ams35/Sim/';           % simulation directory
model_par.netltype='spectre' ;                            % type of netlist
model_par.viewtype='schematic' ;                          % type of view
model_par.cellrep=[];                                    % cell rep for IC6
model_par.sim_subckt=0;                                 % simulate subcircuits
model_par.subcell=[] ;                                  % list of subcircuits
```

.cell: the name of the cell used in Cadence;

.variant: the name of the variant to be simulated associated with the .m variant file;

.simrep: is the path to the simulation repository used by Cadence. This directory was created by the analog simulation environment.

Important note: The SIMECT batch simulator will use a variant file of type *var_<cell>_<variant>.m* from the */variables* repository under the working simulation repository. The user should create the specific file as presented in 3.3.3. by using SIMECT_GUI or manually edit the file.

- Then indicate the verbosity to give which indications will be printed or plotted

```
verb.warning=0;                                         % display warnings or not
verb.ocn_msg=1;                                         % display ocean messages
verb.txt=0;                                            % display debug information in a text file
verb.plot=0;                                           % plot DC results
verb.mesh=0;                                           % plot parametric results
```

- Some information about the run

```
model_par.rnocn=1;                                     % execute ocean
model_par.disp_res=1;                                   % display results
model_par.disp_res_diffcomm=0;                         % display DC/AC in differential mode
model_par.enable_cdn_model=1;                          % will extract a CADENCE compatible vhdl-ams model
model_par.extract_vhdl=0;                             % extract vhdl model
model_par.rempoles=0;                                 % remove/keep out-of-band poles
```

.rnocn : indicates if OCEAN must be run or not. Usually this field should be 1. It can be set to 0 to display the results of a previously run simulation

- DC simulation

```
%DC Analysis Parameters
dc_an.enabled=1;                                       % indicates if a DC analysis is to be performed
```

- Parametric DC simulation

```
%Parametric Analysis
par_an.enabled=1; % indicates in a Parametric DC analysis is to be performed
```

- AC simulation

```
%AC Analysis Parameters
ac_in_an.enabled=1; % indicates if an AC analysis at input is to be performed
ac_in_an.extract_ac_in_pz = 1; % input s-model should be extracted
ac_out_an.enabled=1; % indicates if an AC analysis at output is to be performed
ac_out_an.extract_ac_out_pz = 1; % output s-model should be extracted
ac_an.enable_ac_norm=0; % output s-model should be extracted
```

- Transient simulation

```
%Transient Analysis
trans_an.enabled=1; % indicates if a transient analysis is to be performed
```

Once the SIMECT.m and the variant file are completed, the batch simulation is launched and it will display the selected analyses results at the end of the run.

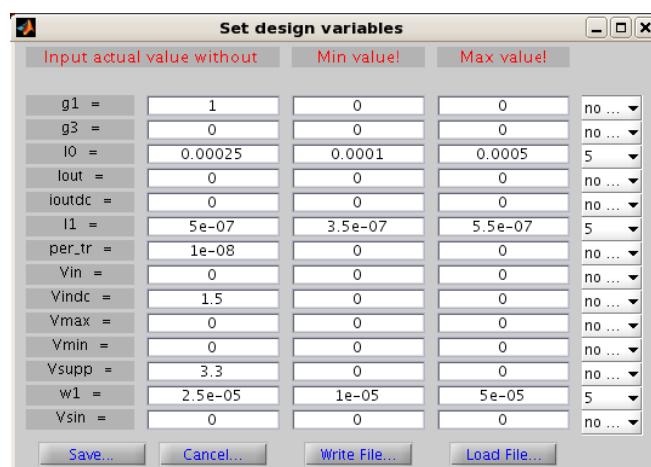
3.5.Performing optimizations using SIMECT_GUI or SIMECT_PAR

When the circuit and transistor-level architectures are designed, the next step is to perform semi-automatic optimizations of the structure.

In this mode the previous parameters extraction can be automatically run by changing some of the schematic design variables. Then the main characteristics (gains, impedances, offsets, transfer functions, etc.) will be plotted as a function of these variables.

The macro-parametric simulations can be performed by using the graphical interface (SIMECT_GUI) or the batch mode parametric component (SIMECT_PAR).

In SIMECT_GUI, the user should define which design variables to parameterize, the start value, the stop value and the number of points of variation. This can be done by pressing the [Edit Vars] button and filling the fields *Min Value*, *Max Value* and *Steps*:



For the other design variables, which are not to be varied, the fields *Min Value*, *Max Value* and *Steps* should be kept 0, 0 and *nopar*;

The button *Save* will retain the values and return the command into the SIMECT main window, *Cancel* will discard the changed values. *Write File* button will save a parameters file of type *param_<cell>_<variant>.m* which can be loaded in other sessions using the *Load File* button.

Once this step done, by pressing the *[Run Parametric]* button, the macro-parametric analyses run. For each step SIMECT will update AC and DC results and it will present for each varied parameter a synthetic window with the results.

Otherwise, the user can perform macro-parametric simulations by using the batch mode parametric component of SIMECT which can be launched by using the [SIMECT_PAR.m](#) MATLAB script.

This file contains the same fields as *SIMECT.m* (schematic name, simulation repository, analyses, etc.) and will be edited by the user accordingly.

Each parametric analysis requires a variant file (like the one used by *SIMECT.m*) and a parameter file. This parameter file of type *param_<cell>_<variant>.m* can be generated in SIMECT_GUI as explained before and must be put in the simulation repository under */variables* as the variant file.

The name of this file is ‘param_’ followed by the name of the schematic, followed by a facultative variant extension e.g. [param_test_v_uni_v_uni_v1.m](#)

Each parameter should be added in the file as:

```
ind=ind+1 ;
param(ind).nam='w1';
param(ind).min=1e-05;
param(ind).max=5e-05;
param(ind).npt=5;
```

A schematic can be associated to several extensions in order to make several simulation variants.

Further details will be presented in the following sections.

4. Design examples

In order to illustrate the possibilities of the tool, we will consider 4 design examples, specifically selected to cover all the combinations of unipolar/differential voltages or currents topologies.

The selected design examples are:

- a unipolar voltage to unipolar voltage source;
- a differential voltage to unipolar current source;
- a unipolar current to differential voltage source;
- a differential current to differential current source;

All others combinations can easily be designed and simulated starting from these basic examples.

We will present all the results obtained using the graphical interface (SIMECT_GUI), but the batch mode can also be employed as presented in 3.4.

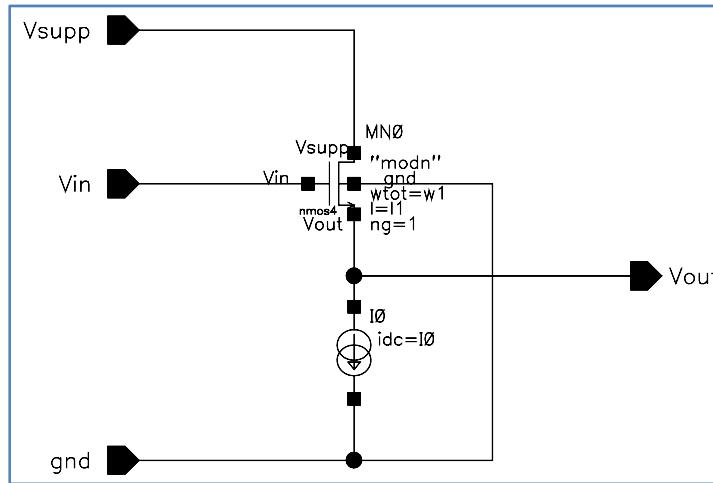
All the possible analyses will be run for these design examples in order to illustrate the possibilities and to extract the macro-models, but separate analyses can also be performed.

The */CADENCE_EXP* folder in the provided archive contains the schematics, testbenches, netlists and SIMECT variants corresponding to the presented designs. The user can directly use them and perform the respective analyses.

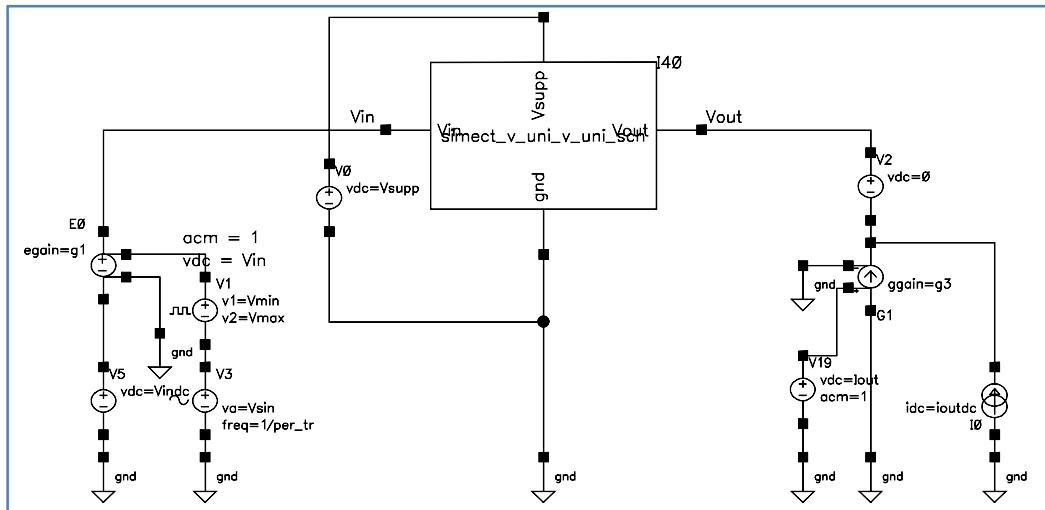
4.1.Unipolar voltage to unipolar voltage source

In this first example, we will reconsider the voltage follower composed with a single MOS transistor polarized by a current source, as presented in 3.1.

Following the presented methodology, its schematic was edited in Cadence (e.g. *simect_v_uni_v_uni_sch* in the */CADENCE_EXP* folder) and a symbol cellview was generated.



As discussed, the design variables are w_1 , l_1 and I_0 . Next, the sources are added and the tesbench is obtained.



It is of main importance to parameterize also the input, output and supply sources with design variables to be used in SIMECT.

The input source is an A1 type source, a unipolar composite source consisting of the voltage sum of a voltage controlled voltage source and a simple VDC source. The static DC component is parameterized as $Vindc$. The AC, dynamic DC and transient components are enabled / disabled by the source E0, its gain being parameterized as $g1$. The source V1 is a composite source supporting AC, DC and transient signal generation. The AC magnitude on the source V1 will be 1, the DC voltage representing the dynamic DC will be Vin and the transient voltage values $Vmin$ and $Vmax$. The period of the pulse will be per_tr . The sinusoidal source V3 delivers a sinusoidal voltage of amplitude $Vsin$ and frequency parameterized as $1/per_tr$.

The input net is denominated Vin .

The output source is an A2 type source, also a unipolar composite source consisting of the current sum of a voltage controlled current source and a simple IDC source. The static DC component on the output is parameterized as $Ioutdc$. The AC and dynamic DC on the output are enabled / disabled by the source G1, its gain being parameterized as $g3$. The source V19 is a composite source supporting AC and DC signal generation. The AC magnitude on the source V19 will be 1, while the DC voltage representing the dynamic DC will be $Iout$. An additional VDC source (V2) is used on the output in order to measure the output current.

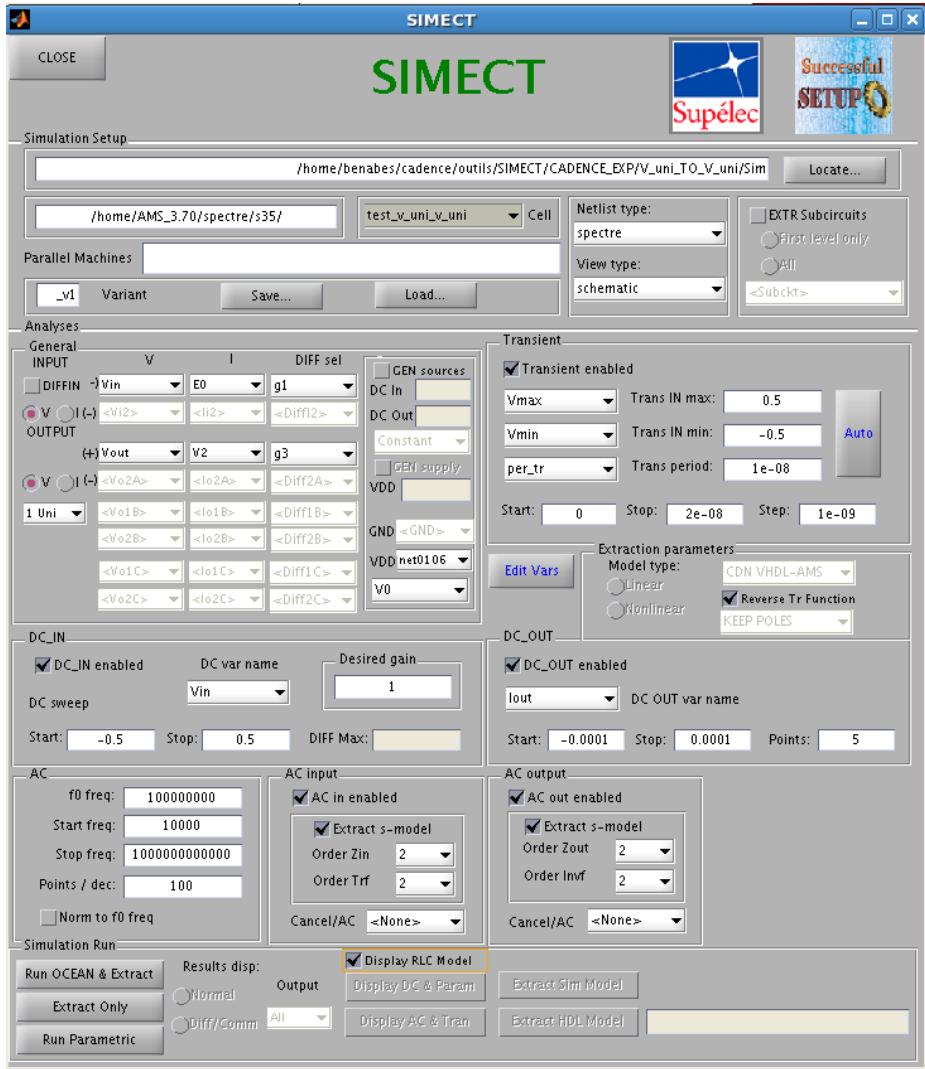
The output net is denominated $Vout$.

The supply source will be a VDC source with the value of the voltage parameterized as $Vsupp$.

Once this step is accomplished, the netlist of the testbench schematic will be generated and Cadence can be closed.

The simulation sources can be also automatically generated by SIMECT, as presented in section 3.3.5.

SIMECT_GUI is launched in MATLAB. The simulation repository is selected and the cell representing the testbench (e.g. `test_v_uni_v_uni` in the `/CADENCE_EXP` folder). The selected variables and the values on the interface will be as example:

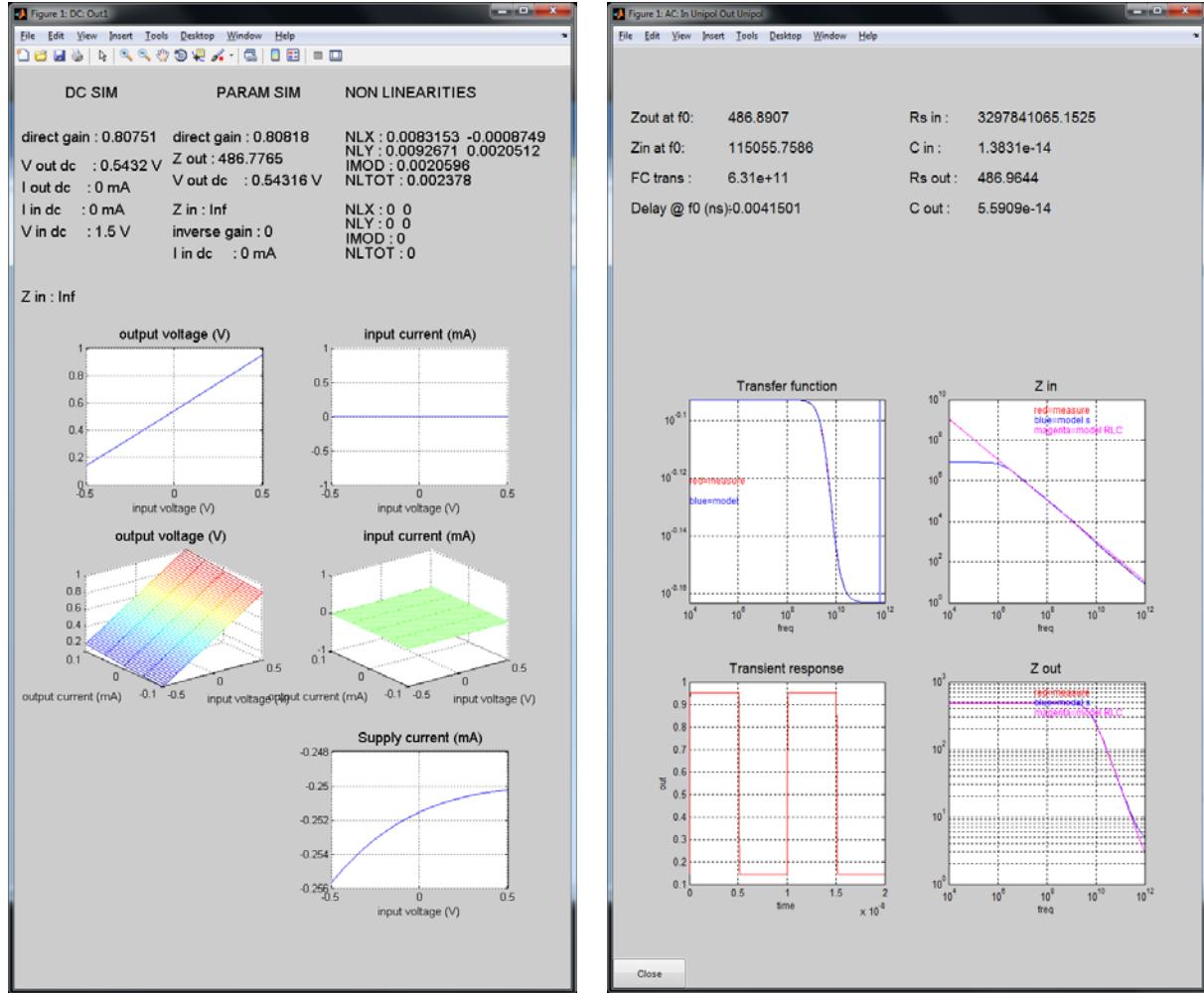


The input net is V_{in} , as parameterized in the schematic and the source to measure the input current is $E0$. The DIFF selection variable on input will be $g1$. Consequently the output and supply are selected.

Next, the design variables are initialized as:

Set design variables				
	Input actual value without units!	Min value!	Max value!	
$g1 =$	1	0	0	no par ▾
$g3 =$	0	0	0	no par ▾
$I0 =$	0.00025	0	0	no par ▾
$ioutdc =$	0	0	0	no par ▾
$I1 =$	5e-07	0	0	no par ▾
$per_tr =$	1e-08	0	0	no par ▾
$V_{in} =$	0	0	0	no par ▾
$V_{indc} =$	1.5	0	0	no par ▾
$V_{max} =$	0	0	0	no par ▾
$V_{min} =$	0	0	0	no par ▾
$V_{supp} =$	3.3	0	0	no par ▾
$w1 =$	2.5e-05	0	0	no par ▾
$V_{sin} =$	0	0	0	no par ▾

We will perform a DC analysis with V_{in} (source V1 in the schematic) varying in [-0.5V; 0.5V], afterwards a parametric analysis over the previous defined DC, with I_{out} (source V19 in the schematic) varying in [-100uA;100uA], AC analyses in the frequency domain [10kHz;1THz] and will extract s-models of order 2 for all the frequency functions. A transient analysis over two periods of f_0 frequency was also enabled. Running the simulation, the results are plotted in two figures, one for the DC and Parametric analyses and one for the AC and transient analyses:



The first figure gives:

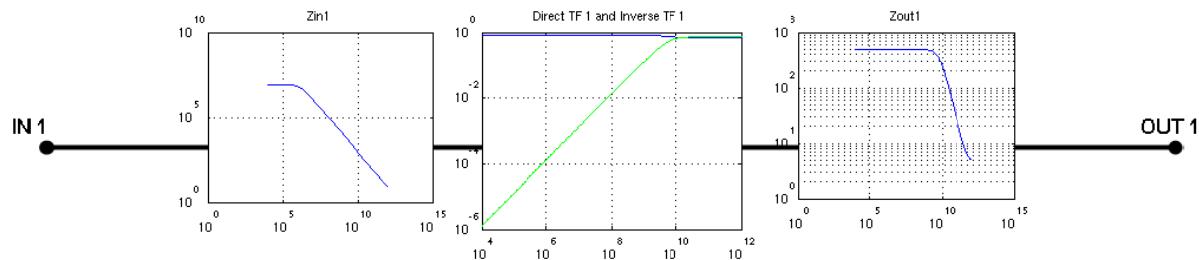
- The amplifier gain extracted from simple DC and parametric DC: 0.8
- The output impedance extracted from parametric DC: 486Ω
- The output offset for $V_{in}=0$ extracted from simple DC and parametric DC: 0.543V
- The input impedance extracted from simple DC and parametric DC: infinite
- The inverse gain extracted from parametric DC: 0
- The input bias current at DC : 0
- All the non-linearity coefficients : NLX(ord2), NLX(ord3), NLY(ord2), NLY(ord3), IMOD
- NLTOT is the maximum non-linearity error for the whole input or output range
- A 3D plot of the output DC voltage function of the input voltage and output current
- A 3D plot of the input DC current function of the input voltage and output current

- A 2D plot of the output voltage function of the input voltage for output current=0
- A 2D plot of the input current function of the input voltage for output current=0
- A 2D plot of the supply current function of the input voltage

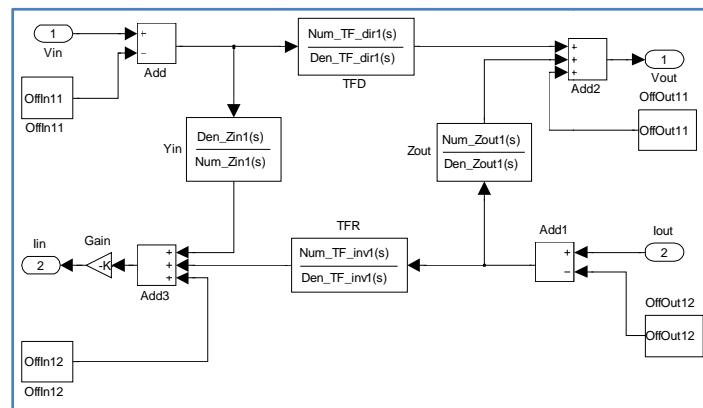
The second figure gives:

- Resistive part of the input and output impedance at frequency f_0 (defined as a parameter)
- The delay at this frequency
- The transfer-function cutoff frequency
- Input and output equivalent model parameters
- AC plots of input impedance, output impedance and transfer function
- AC plots of the extracted s-models and RLC models for the three characteristics
- The transient response.

Choosing the “Normal” type for results display, we can visualize the model topology as s-functions:

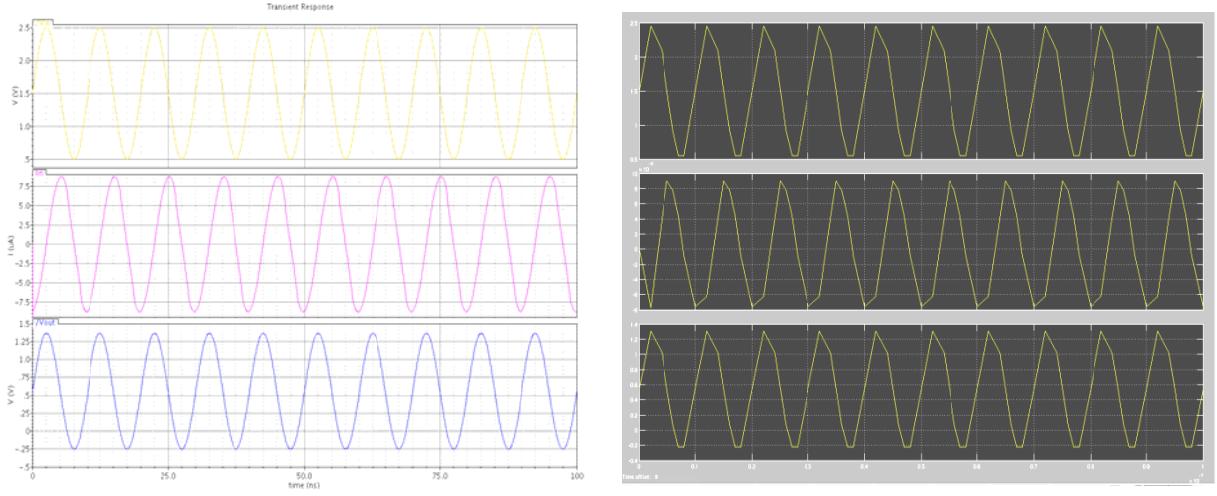


Next, we can extract the SIMULINK and HDL models of the follower. The SIMULINK model will have the structure already presented on the interface and on the AC Normal results:



For the SMASH VHDL-AMS model, two files are extracted: [test_v_uni_v_uni.vhd](#) containing the module description and [test_v_uni_v_uni.pat](#) containing a simple simulation template. For the CADENCE VHDL-AMS model, the extracted file will be: [test_v_uni_v_uni_cdn.vhms](#), while the Verilog model will be: [test_v_uni_v_uni_cdn.va](#).

The models were validated by performing comparative simulations under Cadence Analog Environment and SIMULINK. The results for the transient analysis are presented:

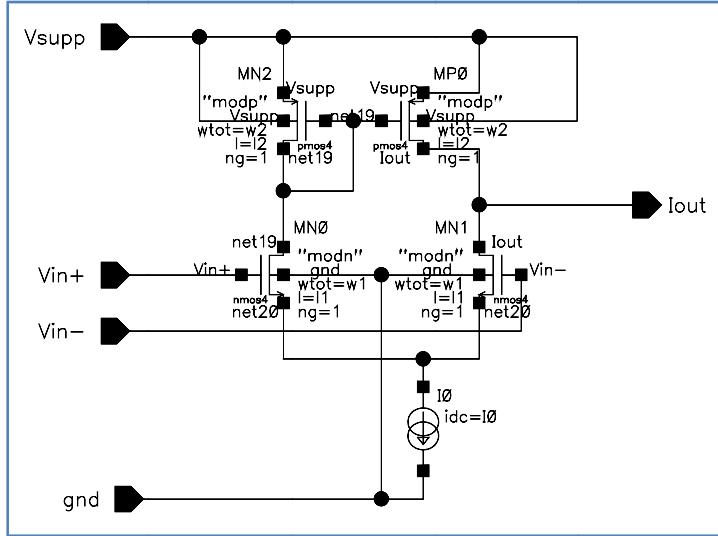


4.2.Differential voltage to unipolar current source

We consider in this example a differential pair: the input is a differential voltage, and the output is a current.

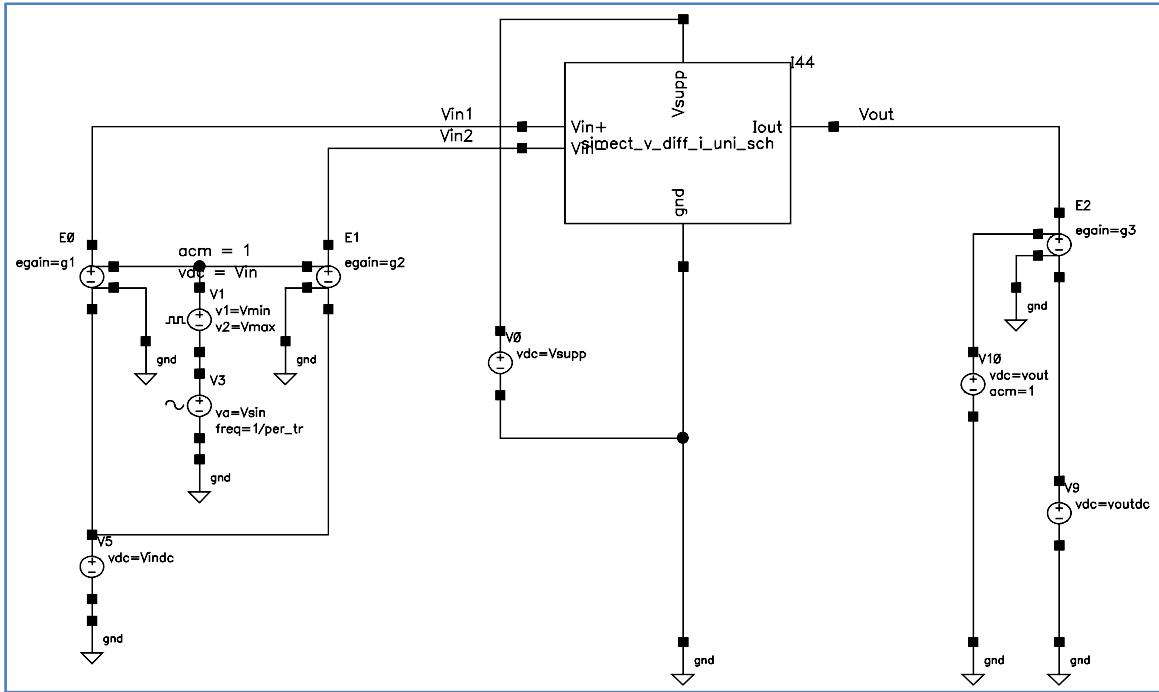
The Cadence schematic was edited (e.g. *simect_v_diff_i_uni_sch* in the */CADENCE_EXP* folder), containing the differential input ports, the output port, a supply port and the ground connection.

The design variables are $w1$, $I1$ – the sizes of the transistors in the input MOS pair, $w2$, $I2$ – the sizes of the MOS transistors in the current mirror and $I0$ the polarization current.



Following the methodology, a symbol cellview is generated. The simulation sources can be manually added or automatically generated by SIMECT, as presented in section 3.3.5. We will consider the manual process in order to show the sources types and parameters.

The testbench schematic (e.g. *test_v_diff_i_uni* in the */CADENCE_EXP* folder) is obtained by adding the required simulation sources (B1 type on input and C2 type on output) and a DC supply source:



The input, output and supply sources are parameterized with design variables to be used in SIMECT.

The input source is a differential composite source consisting on each branch of a voltage sum of a voltage controlled voltage source and a simple VDC source. The common-mode DC component is parameterized as $Vindc$ on source V5. The AC, dynamic DC and transient components are copied on each input through the sources E0 (gain $g1$) and E1 (gain $g2$), used to emulate the differential behavior. The source V1 is a composite source supporting AC, DC and transient signal generation. The AC magnitude on the source V1 will be 1, the DC voltage representing the differential dynamic DC will be Vin and the transient voltage values $Vmin$ and $Vmax$. The period of the pulse will be per_tr . The sinusoidal source V3 delivers a sinusoidal voltage of amplitude $Vsin$ and frequency parameterized as $1/per_tr$.

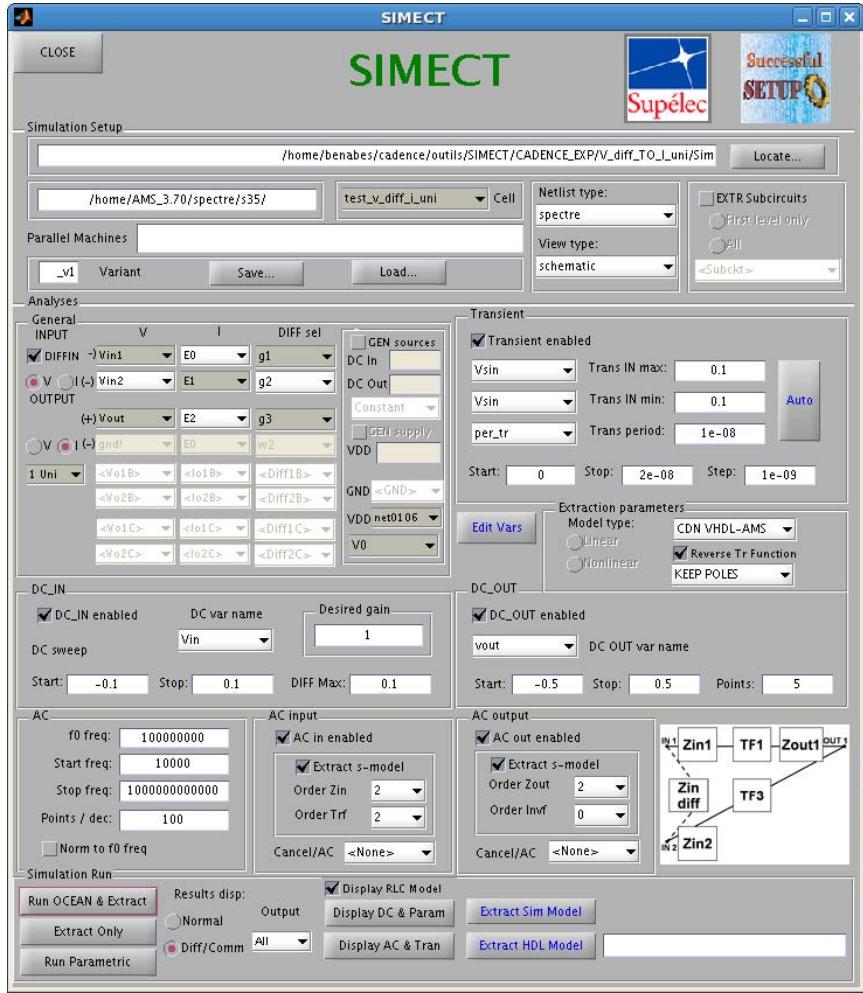
The input nets are parameterized as $Vin1$ for $V+$ and $Vin2$ for $V-$.

The output source is a unipolar composite source consisting of the voltage sum of a voltage controlled voltage source and a simple VDC source. The static DC component on the output is parameterized as $Voutdc$. The AC and dynamic DC components are enabled / disabled by the source E2, its gain being parameterized as $g3$. The source V10 is a composite source supporting AC and DC signal generation. The AC magnitude on the source V10 will be 1, the DC voltage representing the dynamic DC on the output will be $Vout$.

The output net is named $Vout$.

As for the first example, the supply source will be a VDC source with the value of the voltage parameterized as $Vsupp$.

Once this step is accomplished, the netlist of the testbench schematic is generated, Cadence is closed and SIMECT_GUI is launched. The simulation repository is selected and the cell representing the testbench is chosen from the pop-up menu (e.g. *test_v_diff_i_uni* in the */CADENCE_EXP* folder). The selected variables and the values on the interface will be as example:



By using the *Edit Vars* button, the design variables are initialized as:

Set design variables					
Input actual value without units!		Min value!		Max value!	
g1 =	1	0	0	no par	no par
g2 =	-1	0	0	no par	no par
g3 =	0	0	0	no par	no par
I0 =	0.0002	0	0	no par	no par
I1 =	5e-07	0	0	no par	no par
I2 =	5e-07	0	0	no par	no par
per_tr =	1e-08	0	0	no par	no par
Vin =	0	0	0	no par	no par
Vindc =	1	0	0	no par	no par
Vmax =	0	0	0	no par	no par
Vmin =	0	0	0	no par	no par
vout =	0	0	0	no par	no par
voutdc =	2	0	0	no par	no par
Vsin =	0	0	0	no par	no par
Vsupp =	3.3	0	0	no par	no par
w1 =	1e-05	0	0	no par	no par
w2 =	1e-05	0	0	no par	no par

The input nets are selected: *Vin1* and *Vin2*, as parameterized in the schematic and the sources to measure the input currents are *E0* and *E1* respectively. The DIFF selection variable on the (+) input will be *g1*, while on the (-) input will be *g2*. The output and supply variables are also selected.

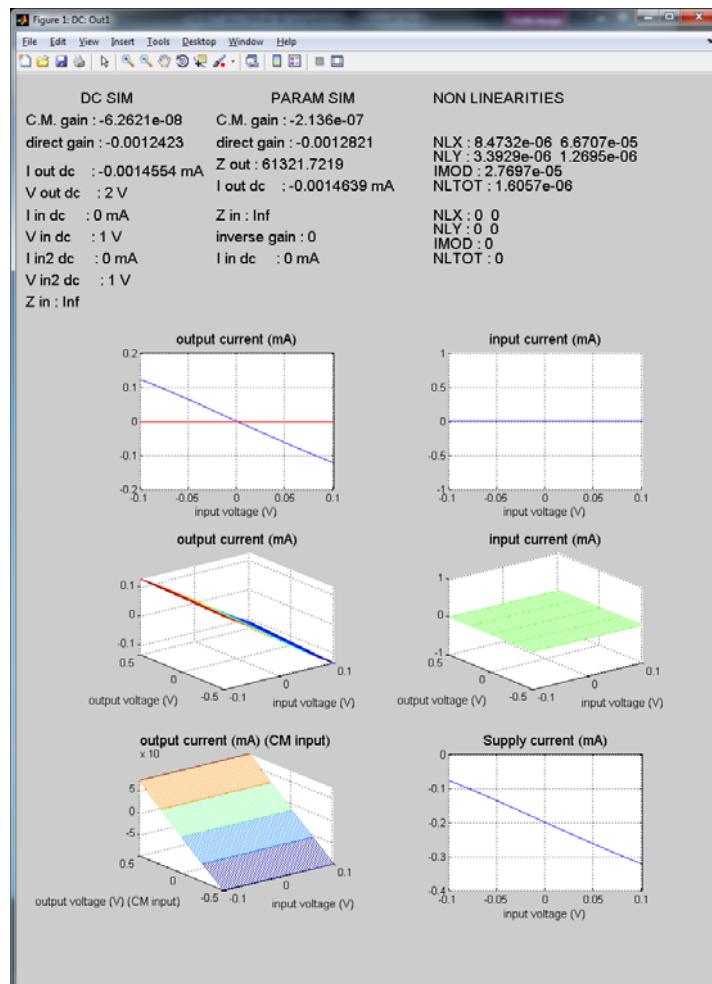
The analyses to be performed:

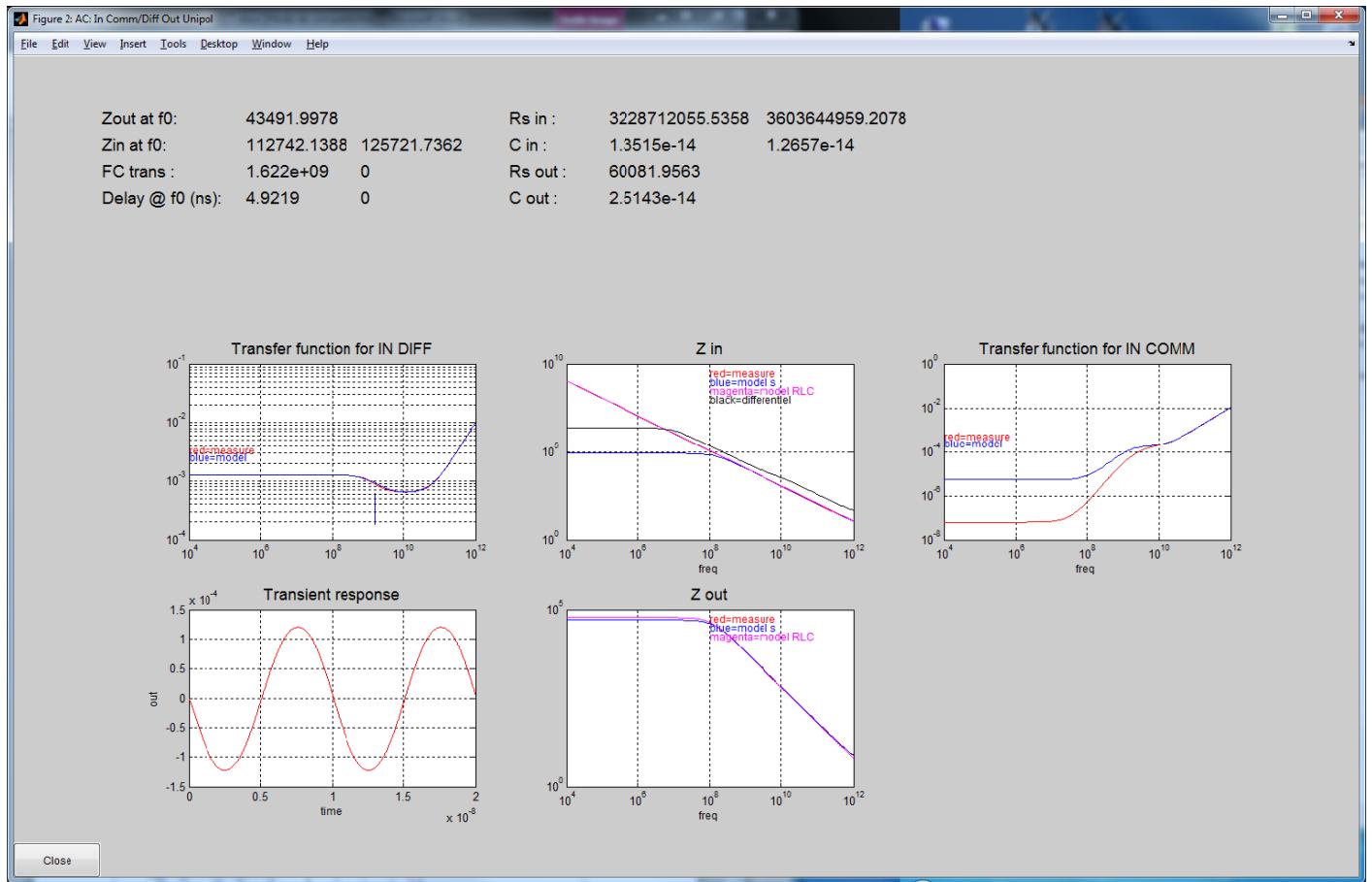
- DC analysis with V_{in} (source V1 in the schematic) varying in [-0.1V; 0.1V] in common mode (fields *Start* and *Stop*) and [-0.1V; 0.1V] in differential mode (field *DIFF Max*, where the differential minimum value is obtained as $-DIFF\ Max$);
- Parametric DC analysis with V_{out} (source V10 in the schematic) varying in [-0.5V; 0.5V], for a number of 10 points of simulation;
- AC analyses in the frequency domain [10kHz; 1THz] resulting in s-models of order 2 for all the frequency functions;
- Transient analysis with a sine signal on input over two periods of f_0 frequency.

Running the simulation and selecting the *Diff/Comm* option for results presentation, they are also plotted in two figures, one for the DC and Parametric analyses and one for the AC and transient analyses.

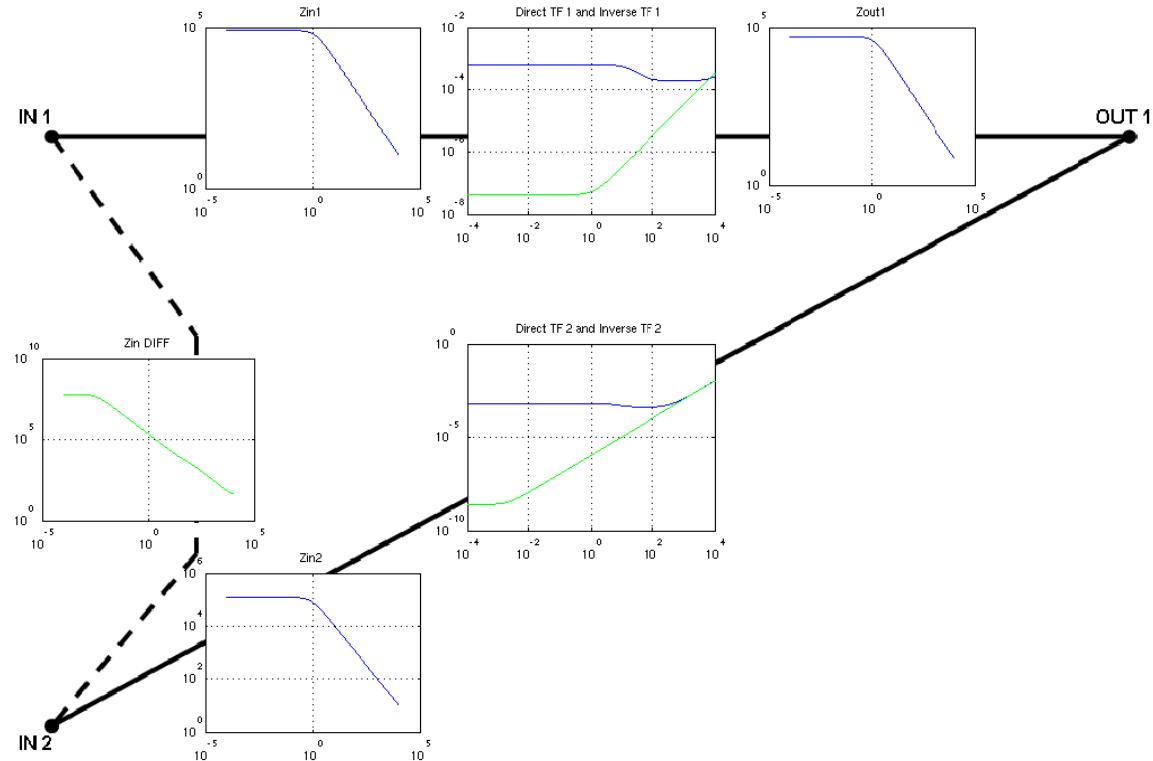
Additional indications are presented in the two figures in order to characterize the common-mode and differential behavior:

- Output function of input in common-mode
- 3D plot of the output function of the input voltage (in common-mode) and output voltage
- The direct transfer function in differential mode and common-mode
- The input impedance in differential mode and common-mode

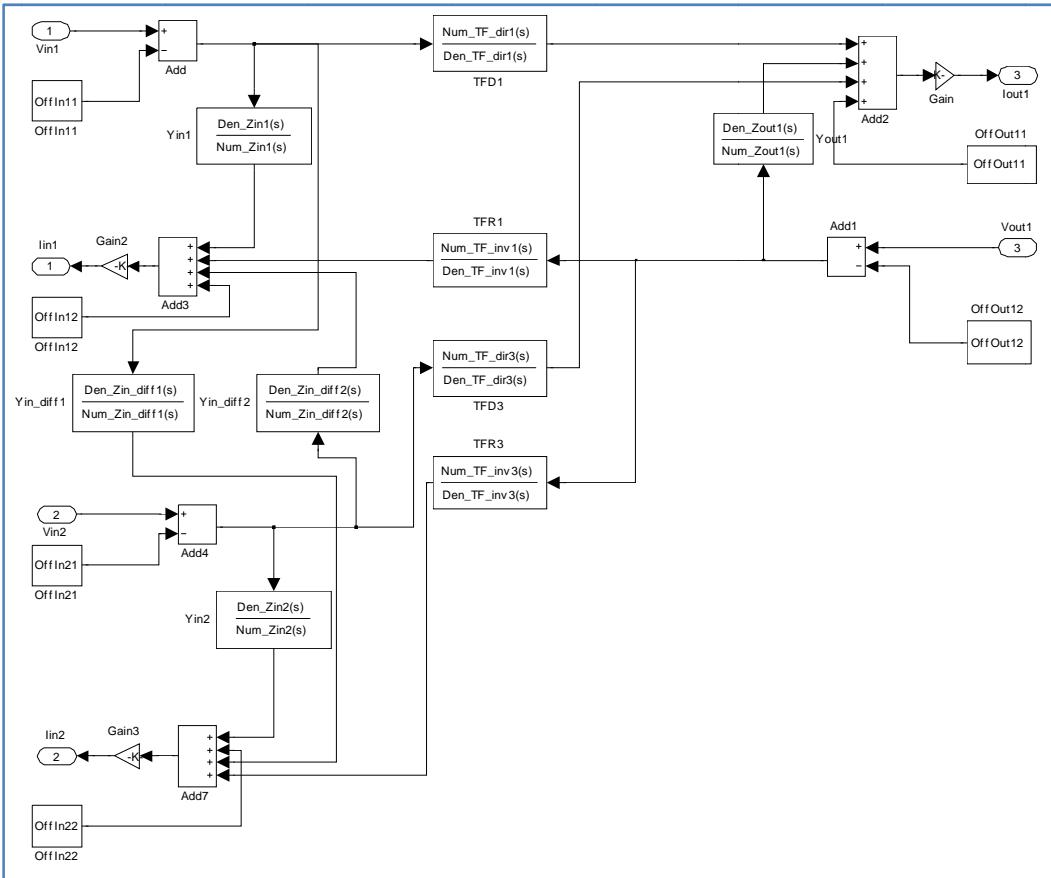




Choosing the “Normal” type for results display, we can visualize the model topology as s-functions:



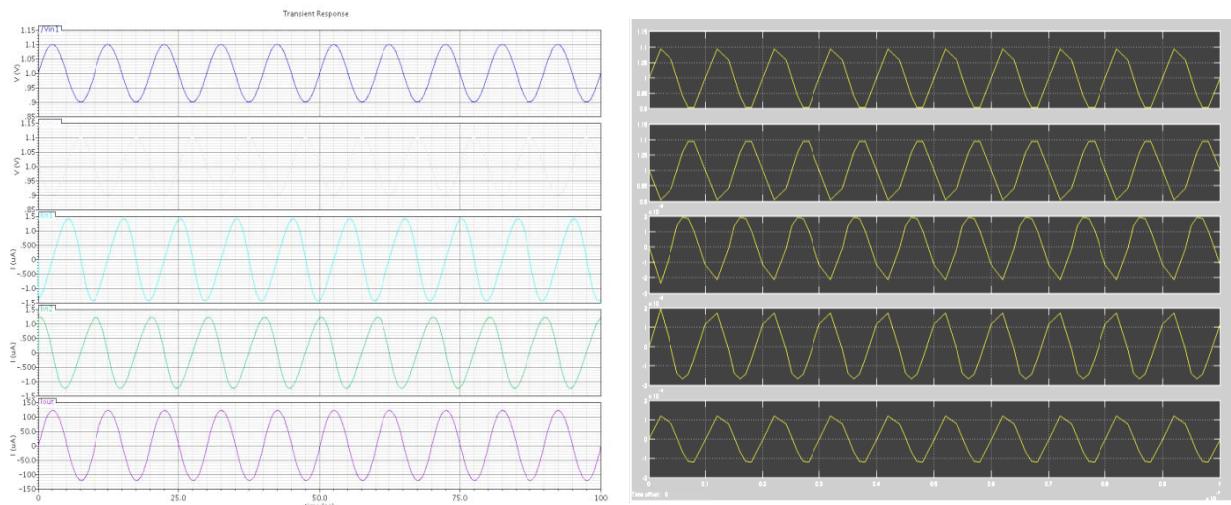
The SIMULINK and HDL models of the differential pair can be extracted. The SIMULINK model will have the following structure:



For the SMASH VHDL-AMS model, the two files are: [test_v_diff_i_uni.vhd](#) and [test_v_diff_i_uni.pat](#).

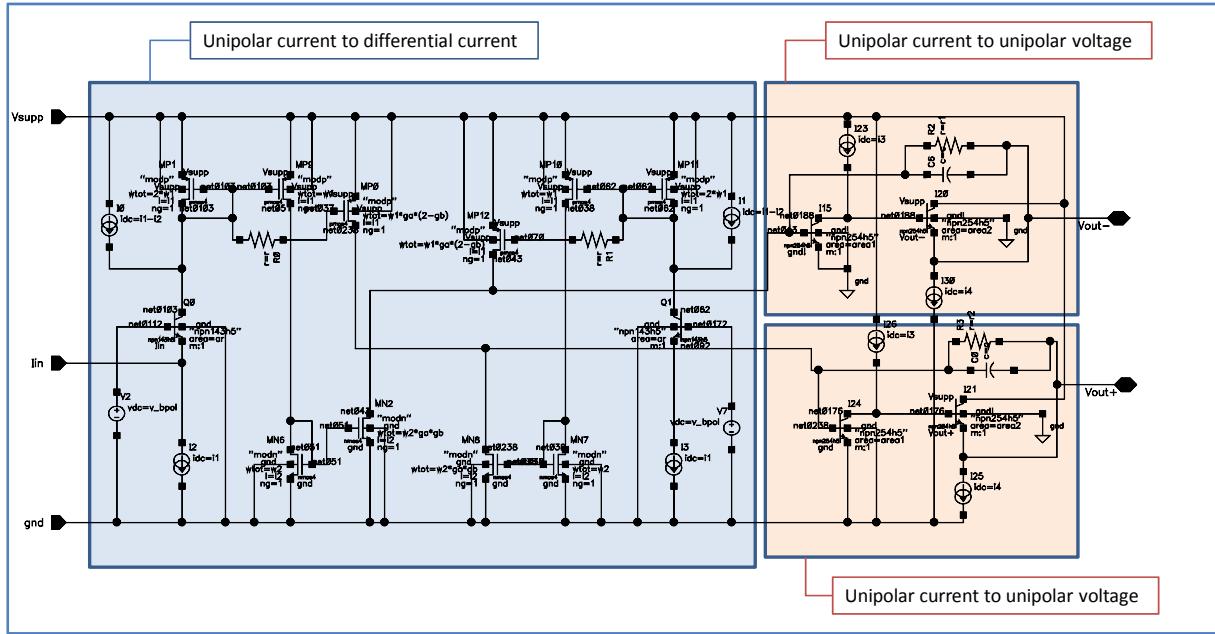
For the CADENCE VHDL-AMS model, the file is: [test_v_diff_i_uni_cdn.vhms](#) while for VerilogA is [test_v_diff_i_uni_cdn.va](#).

Again, the comparative simulations under Cadence Analog Environment and SIMULINK are performed, resulting in coherent results:



4.3.Unipolar current to differential voltage source

The following example employs a multi-stage amplifier of type current-controlled voltage-source. It is composed with one unipolar current to differential current amplifier, followed by two unipolar current to unipolar voltage amplifiers. Its Cadence schematic (e.g. *simect_i_uni_v_diff_sch* in the */CADENCE_EXP* folder) is:



The design variables for the first amplifier are:

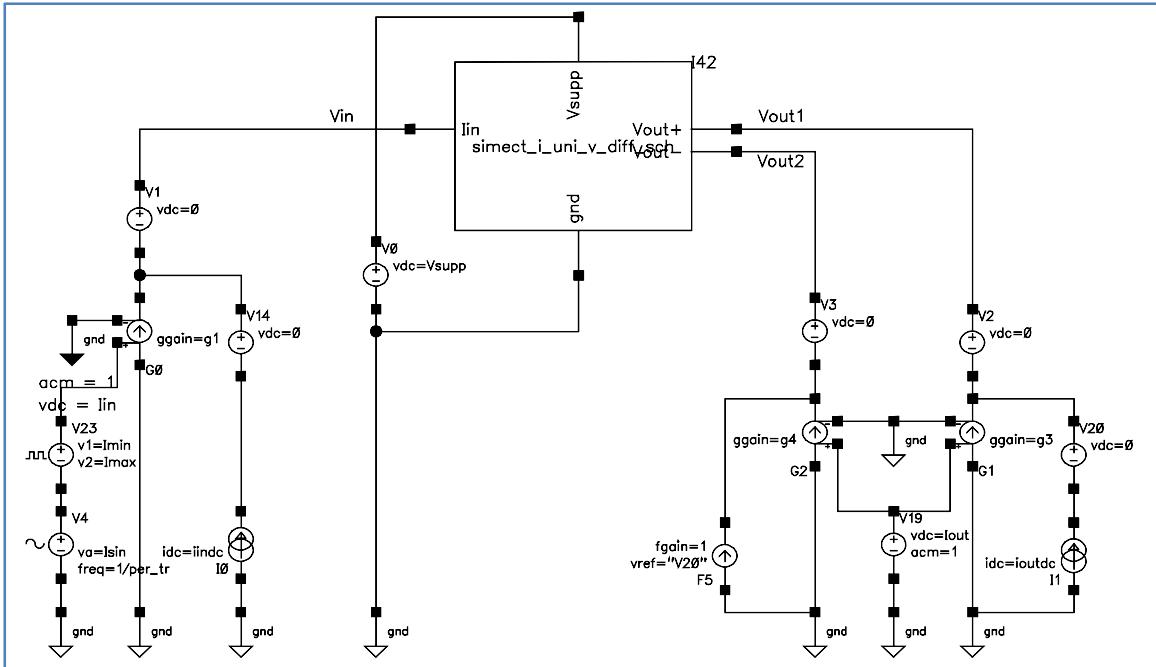
- ar - the size of each bipolar transistor;
- $w1, l1$ - the sizes used for the PMOS current mirrors;
- $w2, l2$ - the sizes used for the NMOS current mirrors;
- $i1, i2$ - the values of the polarization sources;
- v_bpol - the imposed base voltage for each bipolar transistor;
- ga, gb - two gain variables, controlling the differential gain and the common-mode gain on the output;
- r - the value of the two amortization resistances;

The design variables for the second stage amplifiers are:

- $area1$ and $area2$ - the sizes of the two bipolar transistors in both designs
- $i3$ and $i4$ - the values of the two polarization sources
- c - the value of the capacitor on the RC output pair in both designs
- $r1$ or $r2$ - the value of the resistance on the RC pair, customized for each design

A symbol cellview is generated. Next, the simulation sources should be added. As for the previous designs we will consider the manual process for exemplifying purposes.

The testbench schematic (e.g. *test_i_uni_v_diff* in the */CADENCE_EXP* folder) is obtained by adding the required simulation sources (C1 type on input and B2 type on output) and a DC supply source:



The input, output and supply sources are parameterized with design variables to be used in SIMECT.

The input source is a unipolar composite source consisting of the current sum of a voltage controlled current source and a simple IDC source. The static DC component on the input is parameterized as *lindc* on the source I0. The AC, dynamic DC and transient components on the input are enabled / disabled by the source G0, its gain being parameterized as *g1*. The source V23 is a composite source supporting AC, DC and transient signal generation. The AC magnitude on the source V23 will be 1, the DC voltage representing the dynamic DC will be *lin* and the two voltage limits for the square transient signal are *lmin* and *lmax*. The period is *per_tr*. The source V4 is a sinusoidal transient source with *lsin* as amplitude and frequency = *1/per_tr*. An additional VDC source (V1) is used on the input in order to measure the input current.

The input net is parameterized as *Vin*.

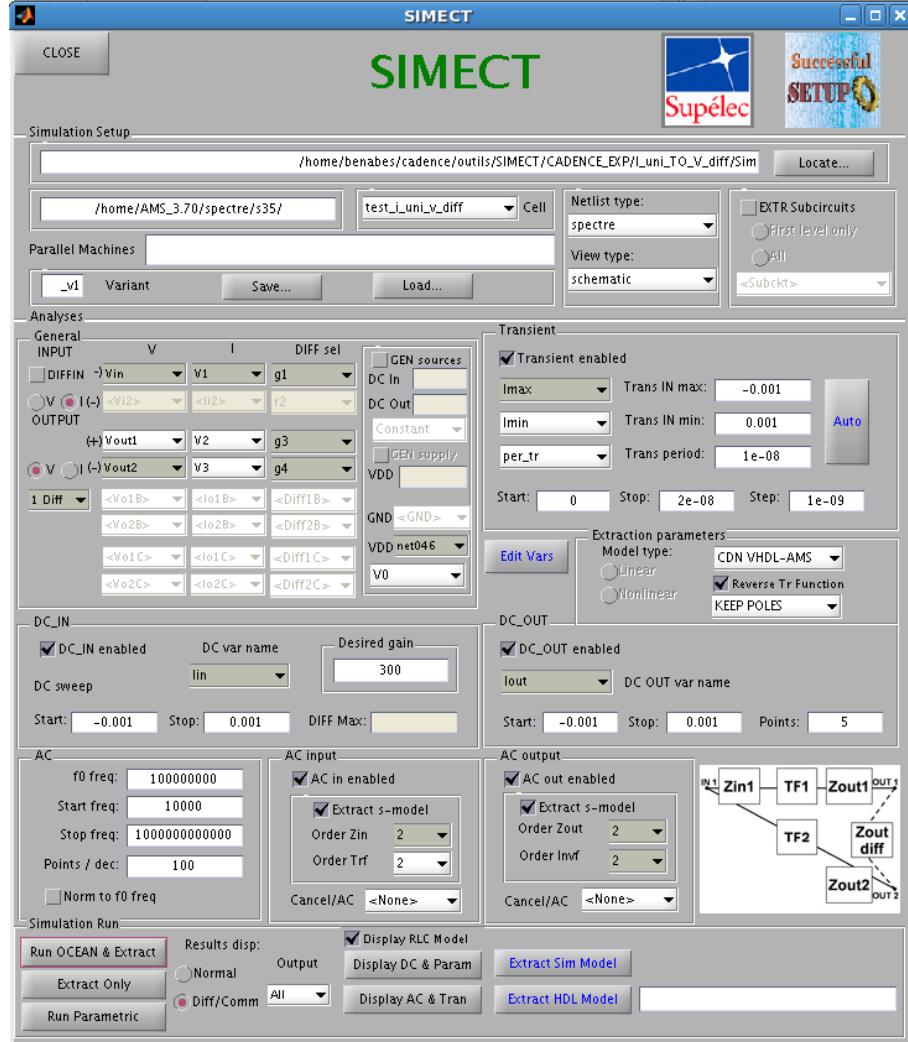
The output source is a differential composite source consisting on each branch of a current sum of a voltage controlled current source and a simple IDC source. The static DC component on the output is parameterized as *loutdc* on the source I1. Its value is copied by the source F5 on the second branch. The AC and dynamic DC components on the output are enabled / disabled on each branch by the sources G1 (gain *g3*) and G2 (gain *g4*), assuring the common-mode and differential output behavior. The source V19 is a composite source supporting AC and DC signal generation. The AC magnitude on the source V19 will be 1, the DC voltage representing the dynamic DC will be *lout*. Two additional VDC source (V2 and V3) are used on each output branch in order to measure the output currents.

The output nets are named *Vout1* and *Vout2*.

The supply source will be parameterized as *Vsupp*.

Once this step is accomplished, the netlist of the testbench schematic is generated, Cadence is closed and SIMECT_GUI is launched.

The simulation repository is selected and the cell representing the testbench is chosen from the pop-up menu (e.g. *test_i_uni_v_diff* in the */CADENCE_EXP* folder). The selected variables and the values on the interface will be:



By using the *Edit Vars* button, the design variables are initialized as:

Set design variables				
	Input actual value without units!	Min value!	Max value!	
ar =	9.6	0	0	no par ▾
g1 =	1	0	0	no par ▾
g2 =	-1	0	0	no par ▾
g3 =	0	0	0	no par ▾
g4 =	0	0	0	no par ▾
i1 =	0.0039	0	0	no par ▾
i2 =	0.0017	0	0	no par ▾
lin =	0	0	0	no par ▾
iindc =	0.0001	0	0	no par ▾
lmax =	0	0	0	no par ▾
lmin =	0	0	0	no par ▾
lsin =	0.001	0	0	no par ▾
I1 =	3.5e-07	0	0	no par ▾
I2 =	3.5e-07	0	0	no par ▾
per_tr =	1e-08	0	0	no par ▾
r =	1510	0	0	no par ▾
v_bpol =	3	0	0	no par ▾
Vsupp =	5	0	0	no par ▾
w1 =	1.5e-05	0	0	no par ▾
w2 =	1e-05	0	0	no par ▾
ga =	0.9986	0	0	no par ▾
gb =	0.9738	0	0	no par ▾
area1 =	4.8	0	0	no par ▾
area2 =	16	0	0	no par ▾
c =	7.5e-14	0	0	no par ▾
i3 =	0.0002	0	0	no par ▾
i4 =	0.004	0	0	no par ▾
lout =	0	0	0	no par ▾
ioutdc =	3.2e-15	0	0	no par ▾
r1 =	500	0	0	no par ▾
r2 =	400	0	0	no par ▾

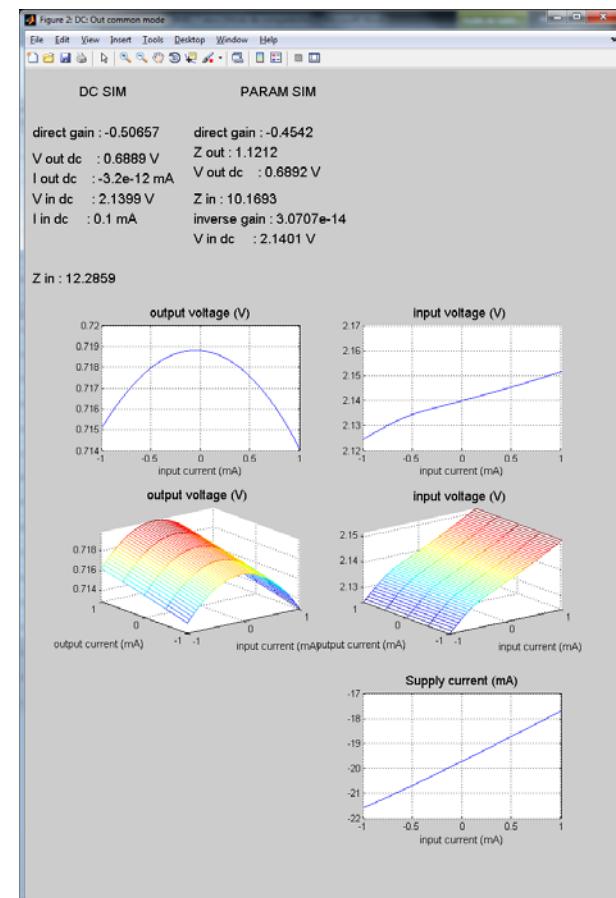
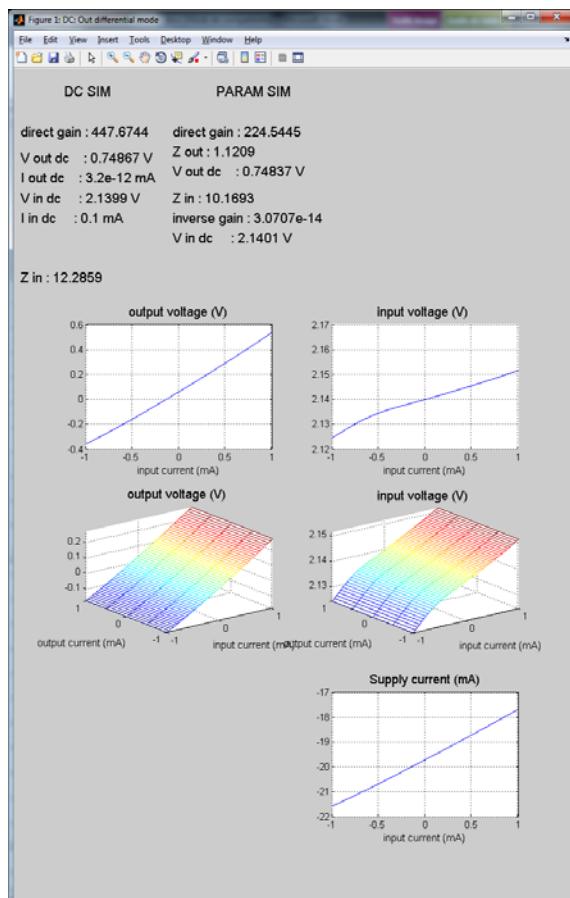
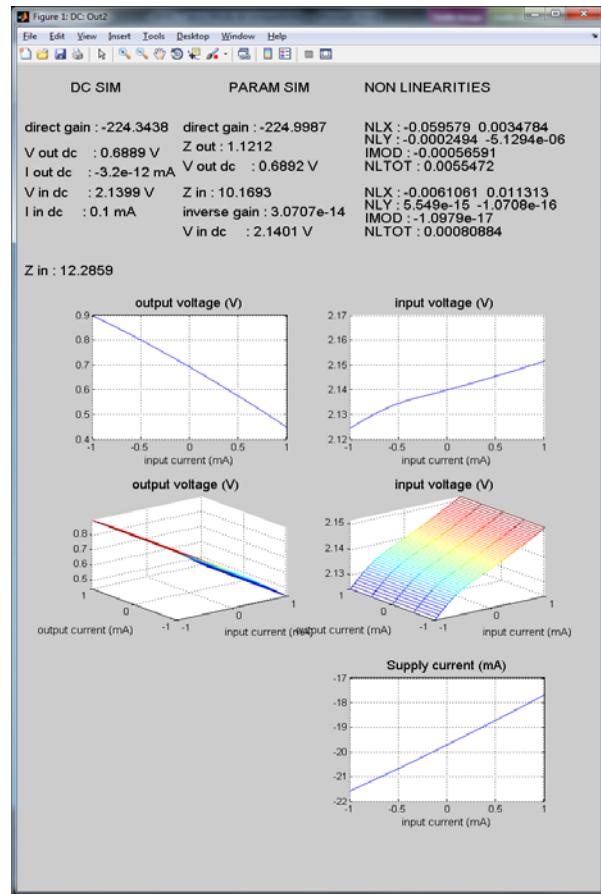
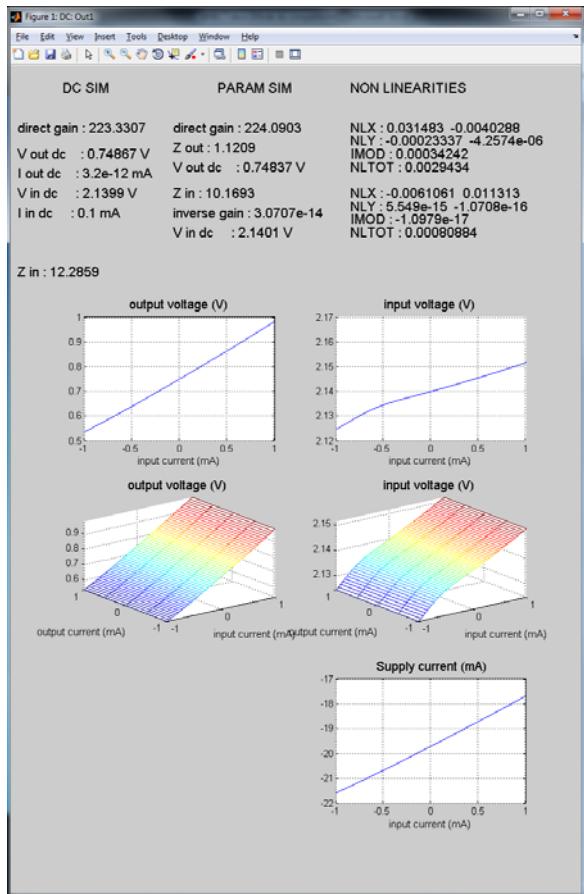
[Save...](#)[Cancel...](#)[Write File...](#)[Load File...](#)

The input net is V_{in} as parameterized in the schematic and the source to measure the input current is $V1$. The DIFF selection variable on the input will be $g1$. The output nets are V_{out1} and V_{out2} , the sources to measure the output currents are $V2$ and $V3$ respectively. The supply source is selected: $V0$.

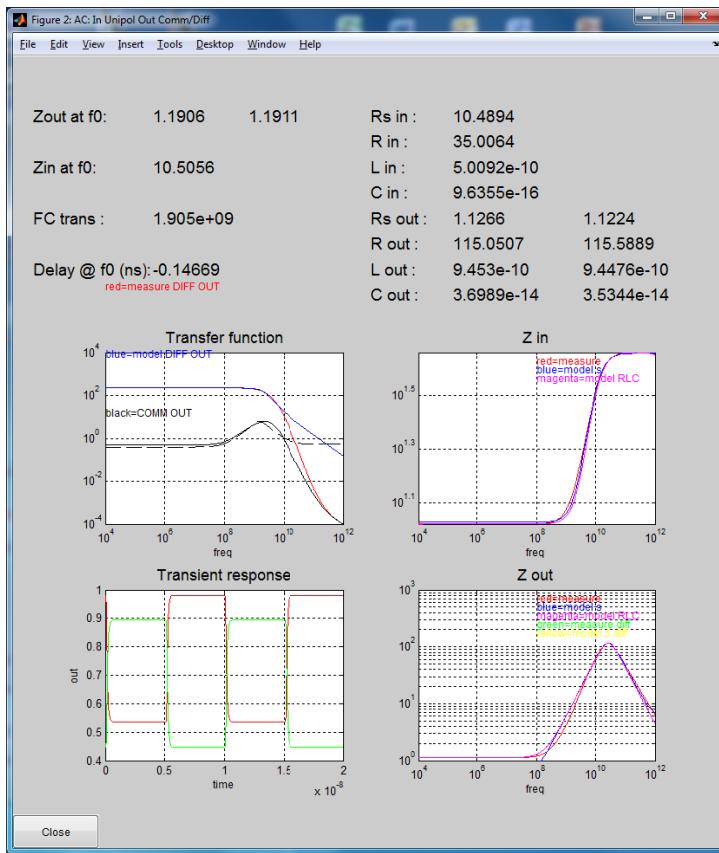
The analyses to be performed:

- DC analysis with lin (source V23 in the schematic) varying in [-1mA; 1mA];
- Parametric DC analysis with $lout$ (source V19 in the schematic) varying in [-1mA; 1mA], for a number of 10 points of simulation;
- AC analyses in the frequency domain [10kHz; 1THz] resulting in s-models of order 2 for all the frequency functions;
- Transient analysis with square input signal performed over two periods of f_0 .

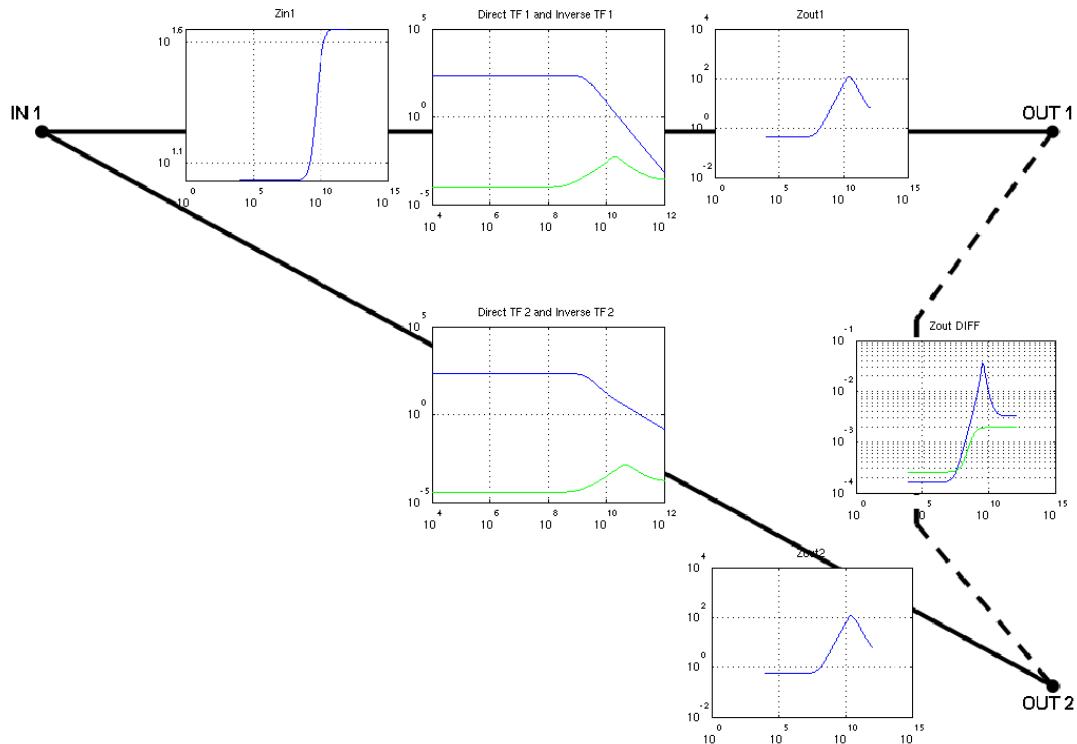
Running the simulation and selecting the *Normal* option for results presentation, the DC results are presented in two windows, one for each circuit output. The *Diff/Comm* option for results presentation will compute and display the common mode DC on output and the differential DC on output, also in two separate windows:



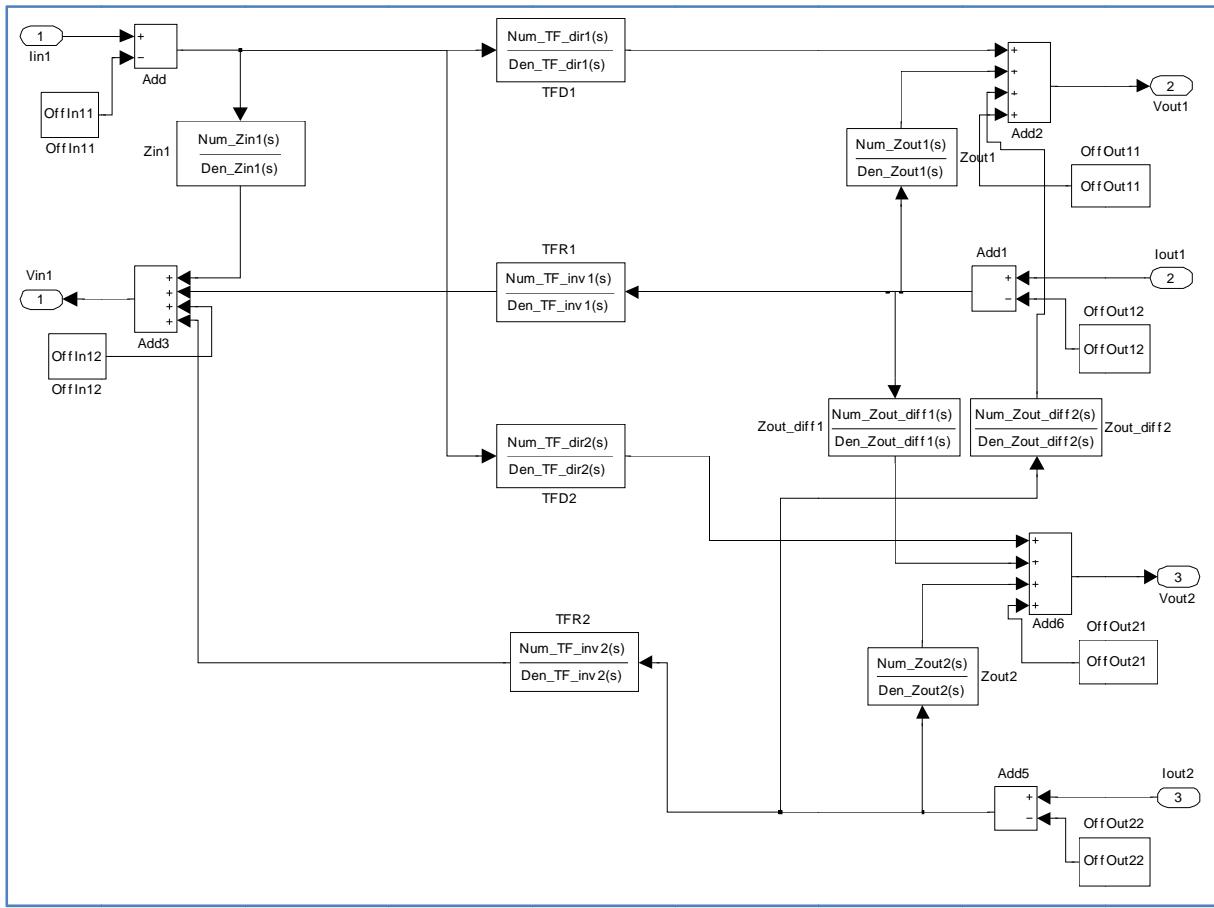
The AC results when selecting the *Diff/Comm* option will be:



Selecting the “Normal” type for results display, the model topology as s-functions will be displayed:



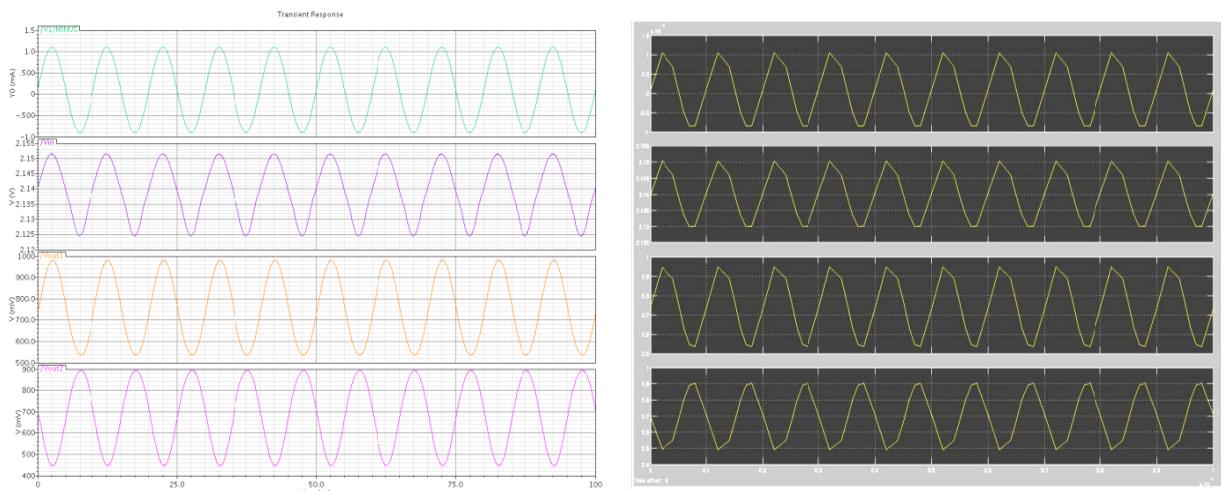
The SIMULINK model of the circuit will have the following structure:



For the SMASH VHDL-AMS model, the two files extracted will be: [test_i_uni_v_diff.vhd](#) and [test_i_uni_v_diff.pat](#).

For the CADENCE VHDL-AMS we will have: [test_i_uni_v_diff_cdn.vhms](#) and for VerilogA [test_i_uni_v_diff_cdn.va](#).

The comparative simulations under Cadence Analog Environment (transistor-level) and SIMULINK (macro-model) are performed:

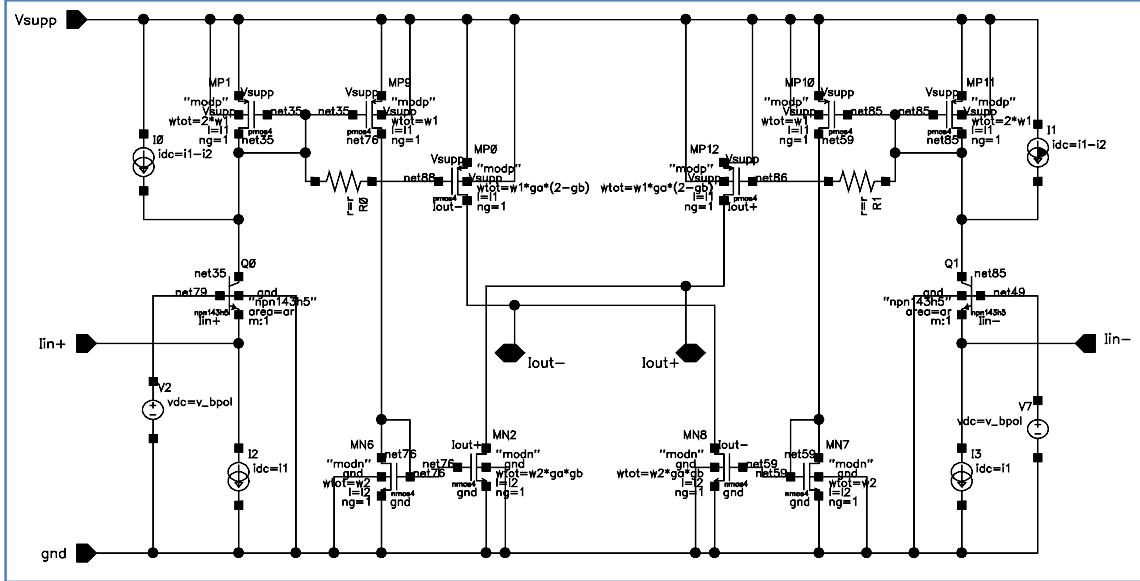


The levels of voltage and current on the input and output are verified for the two types of simulation.

4.4.Differential current to differential current source

The fourth example presents a fully differential amplifier of type current-controlled current-source.

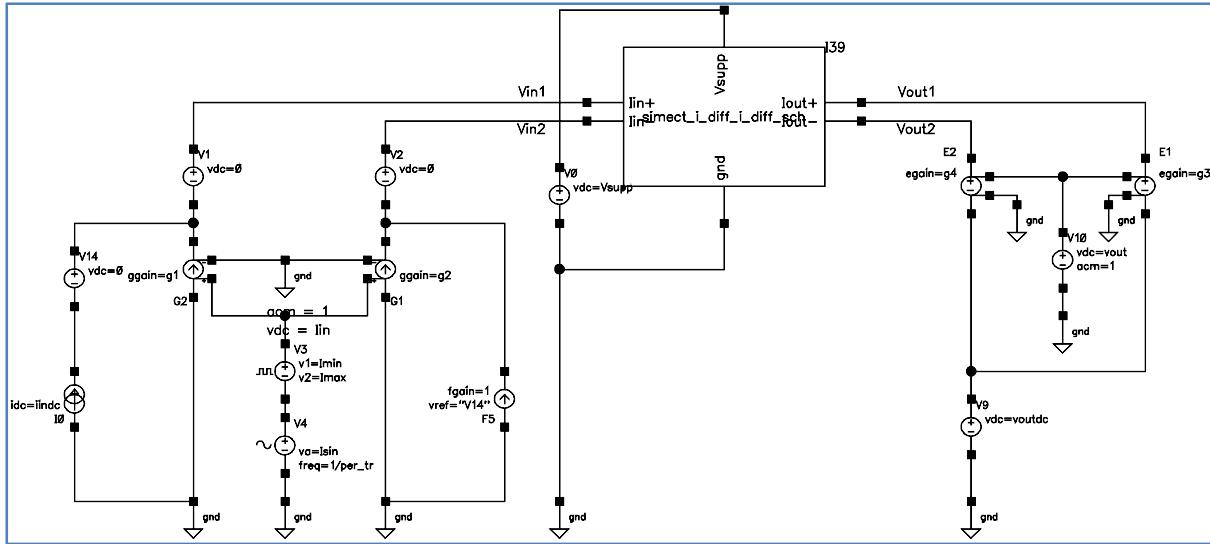
Its Cadence schematic (e.g. *simect_i_diff_i_diff_sch* in the */CADENCE_EXP* folder) is:



The design variables for this amplifier are:

- ar - the size of each bipolar transistor;
- $w1, I1$ - the sizes used for the PMOS current mirrors;
- $w2, I2$ - the sizes used for the NMOS current mirrors;
- $i1, i2$ - the values of the polarization sources;
- v_bpol - the imposed base voltage for each bipolar transistor;
- ga, gb - two gain variables, controlling the differential gain and the common-mode gain on the output;
- r - the value of the two amortization resistances;

A symbol cellview is generated and the testbench schematic (e.g. *test_i_diff_i_diff* in the */CADENCE_EXP* folder) is obtained with the simulation sources, D1 type on input and D2 on output:



The input, output and supply sources are parameterized with design variables to be used in SIMECT.

The input source is a differential composite source consisting on each branch of a current sum of a voltage controlled current source and a simple IDC source. The static DC component on the input is parameterized as *lindc* on the source *I0*. Its value is copied by the source *F5* on the second branch. The AC, dynamic DC and transient components on the input are enabled / disabled on each branch by the controlled sources of gain *g1* and *g2*, assuring the common-mode and differential input behavior. The source *V3* is a composite source supporting AC and DC and transient signal generation. The AC magnitude on the source *V3* is imposed 1, the DC voltage representing the dynamic DC will be *lin*. The two voltage limits for the square transient signal are *lmin* and *lmax* with the period *per_tr*. The source *V4* is a sinusoidal transient source with *lsin* as amplitude and frequency = *1/per_tr*. Two additional VDC source (*V1* and *V2*) are used on each output branch in order to measure the input currents.

The input nets are parameterized as *Vin1* and *Vin2*.

The output source is a differential composite source consisting on each branch of a voltage sum of a voltage controlled voltage source and a simple VDC source. The common-mode DC component on the output is parameterized as *Voutdc* on source *V9*. The AC and dynamic DC components are copied on each output branch through the sources *E1* (gain *g3*) and *E2* (gain *g4*), used to emulate the differential behavior. The source *V10* is a composite source supporting AC and DC signal generation. The AC magnitude on the source *V10* will be 1, the DC voltage representing the differential dynamic DC will be *Vout*.

The output nets are named *Vout1* and *Vout2*.

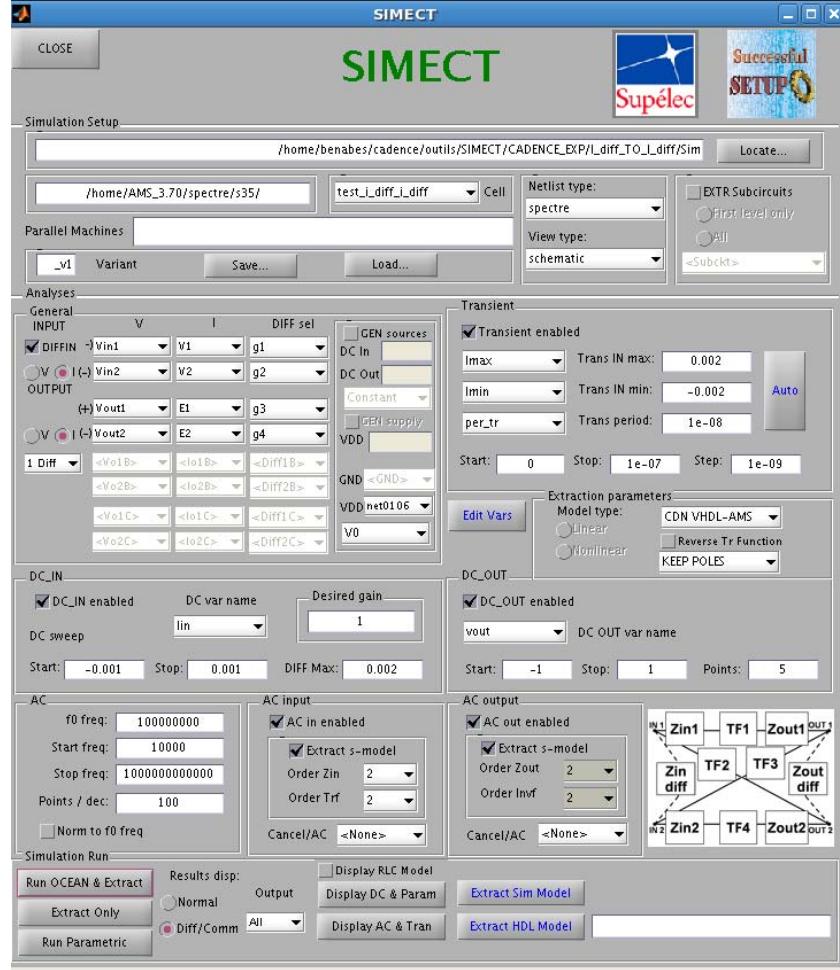
The supply source will be a VDC source parameterized as *Vsupp*.

Again, the netlist of the testbench schematic is generated.

The simulation sources can be also automatically generated by SIMECT, as presented in section 3.3.5, without needing a testbench schematic.

Cadence is closed and SIMECT_GUI is launched.

The simulation repository is selected and the cell representing the testbench is chosen from the pop-up menu (e.g. *test_i_diff_i_diff* in the */CADENCE_EXP* folder). The selected variables and the values on the interface will be:



The design variables will be initialized as:

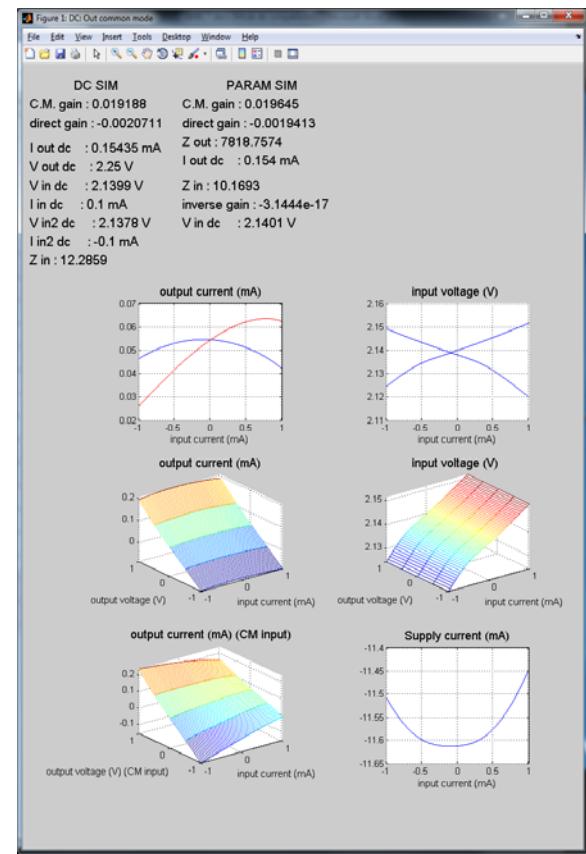
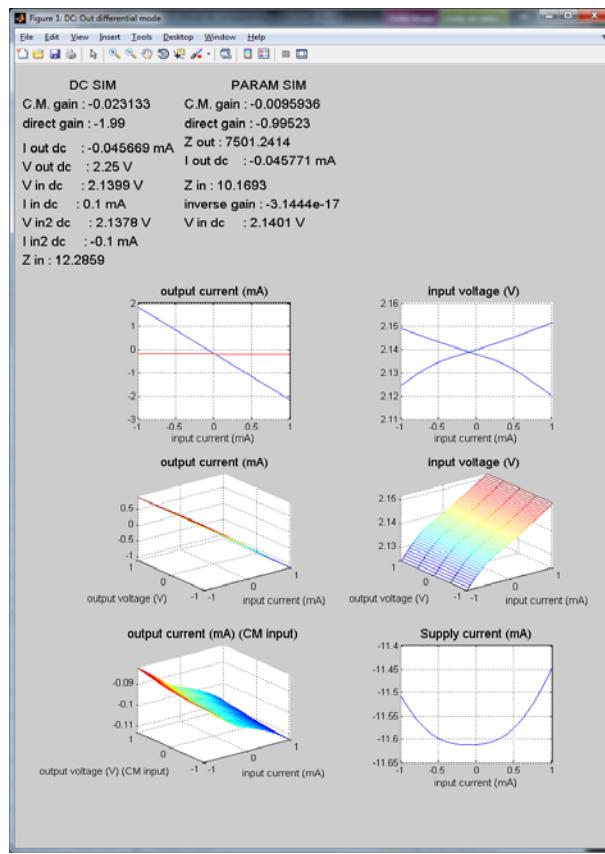
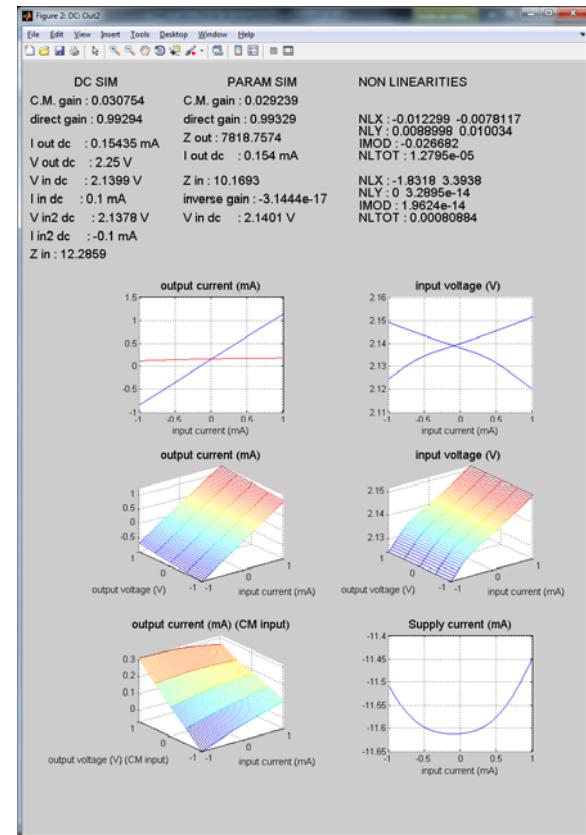
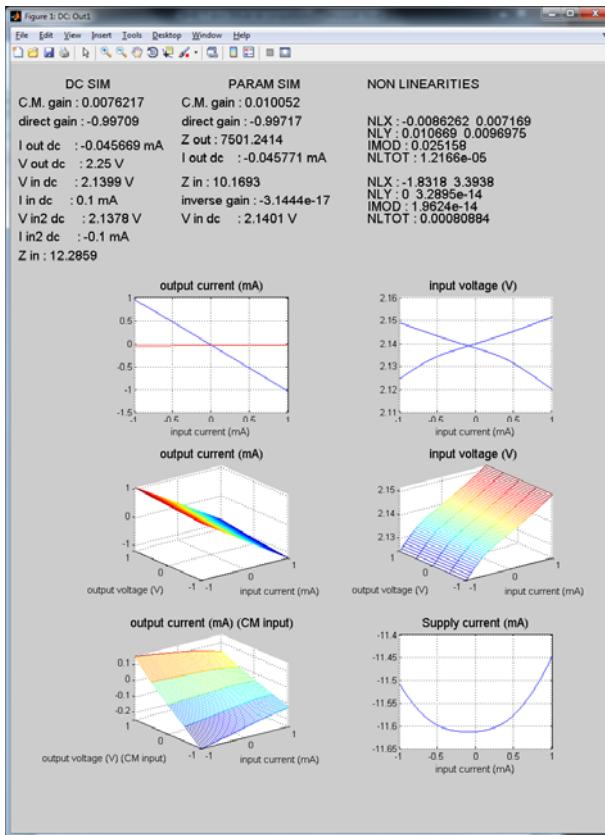


The input nets are selected: *Vin1* and *Vin2* and the sources to measure the input currents: *V1* and *V2*. The DIFF selection variables on the input will be *g1* and *g2*. The output nets are *Vout1* and *Vout2*, the sources to measure the output currents are *E1* and *E2* respectively. The supply source is selected: *V0*.

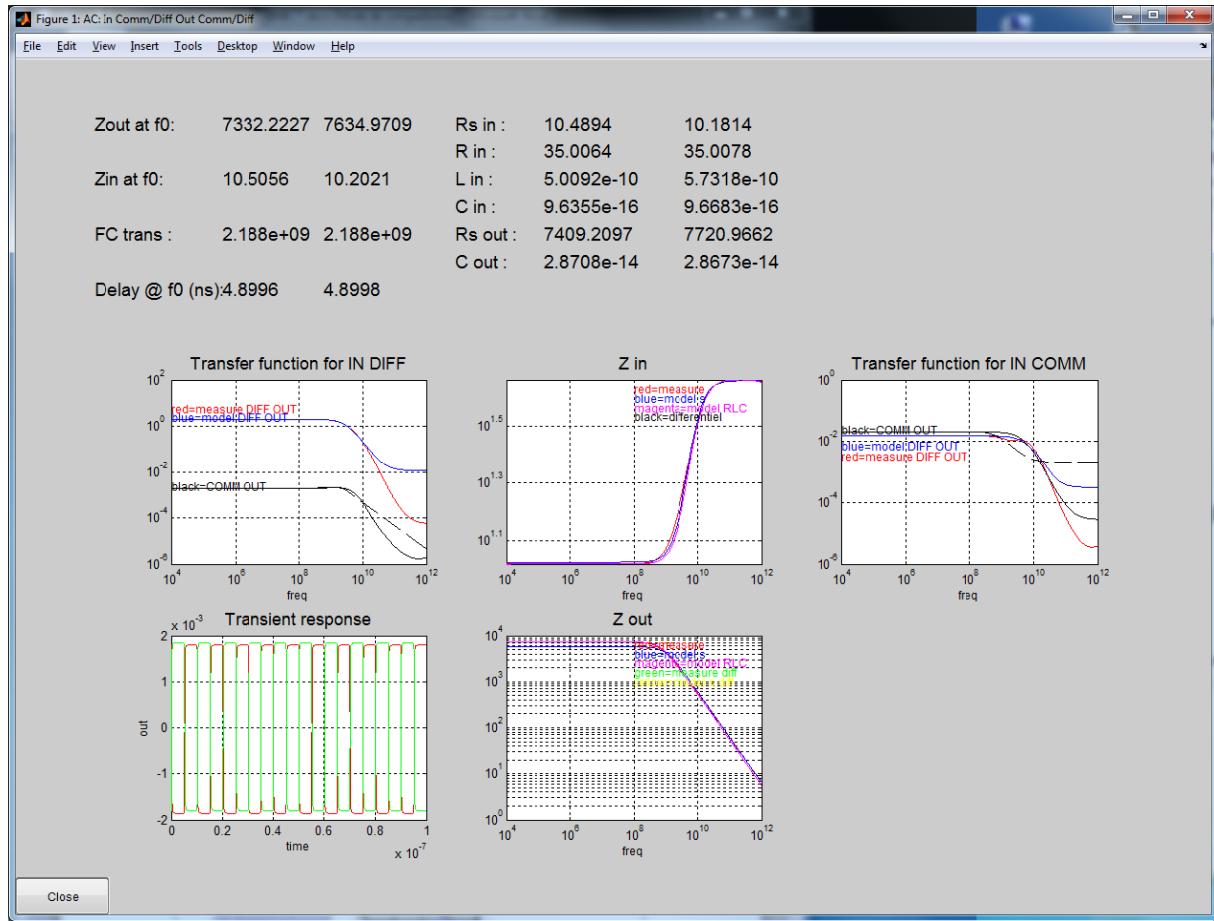
The analyses to be performed:

- DC analysis with *lin* (source V3 in the schematic) varying in [-1mA; 1mA];
- Parametric DC analysis with *Vout* (source V10 in the schematic) varying in [-1V; 1V], for a number of 10 points of simulation;
- AC analyses in the frequency domain [10kHz; 1THz] resulting in s-models of order 2 for all the frequency functions;
- Transient analysis with automatically generated parameters, using *Auto*;

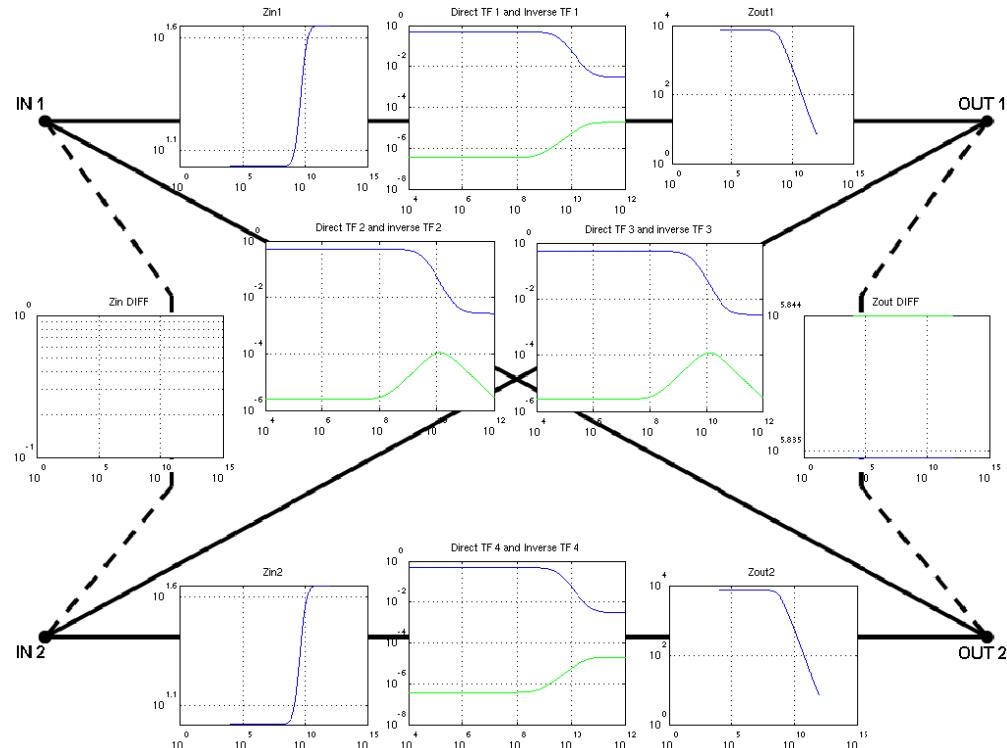
Running the simulation and selecting the *Normal* option for results presentation, the DC results are presented in two windows, one for each circuit output. The *Diff/Comm* option for results presentation will compute and display the common mode DC on output and the differential DC on output, also in two separate windows:



The AC results when selecting the *Diff/Comm* option will be:

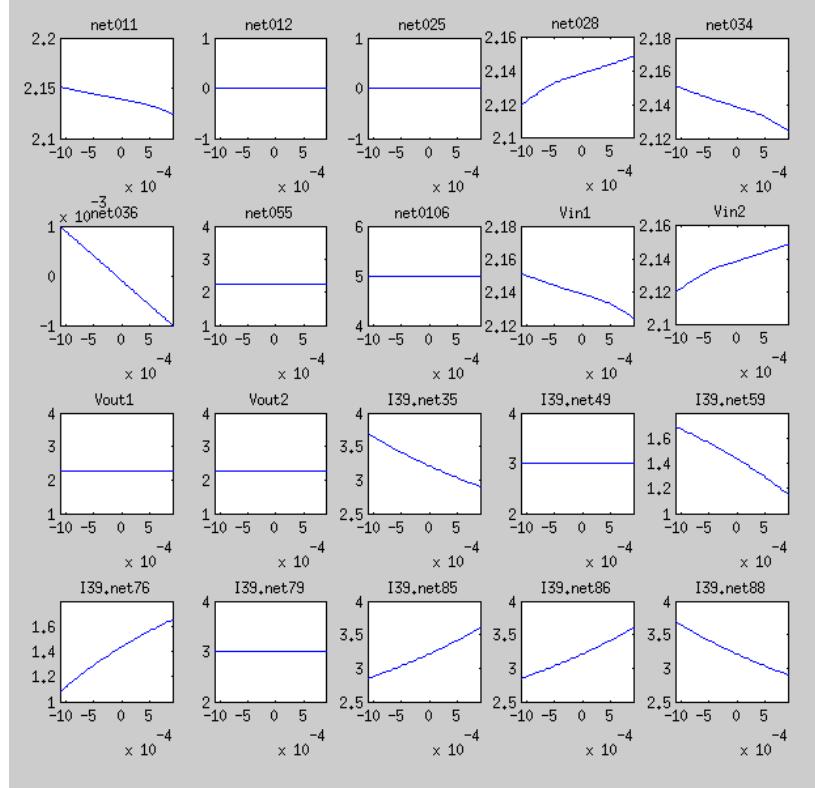


Selecting the “Normal” type for results display, the model topology as s-functions will be displayed:

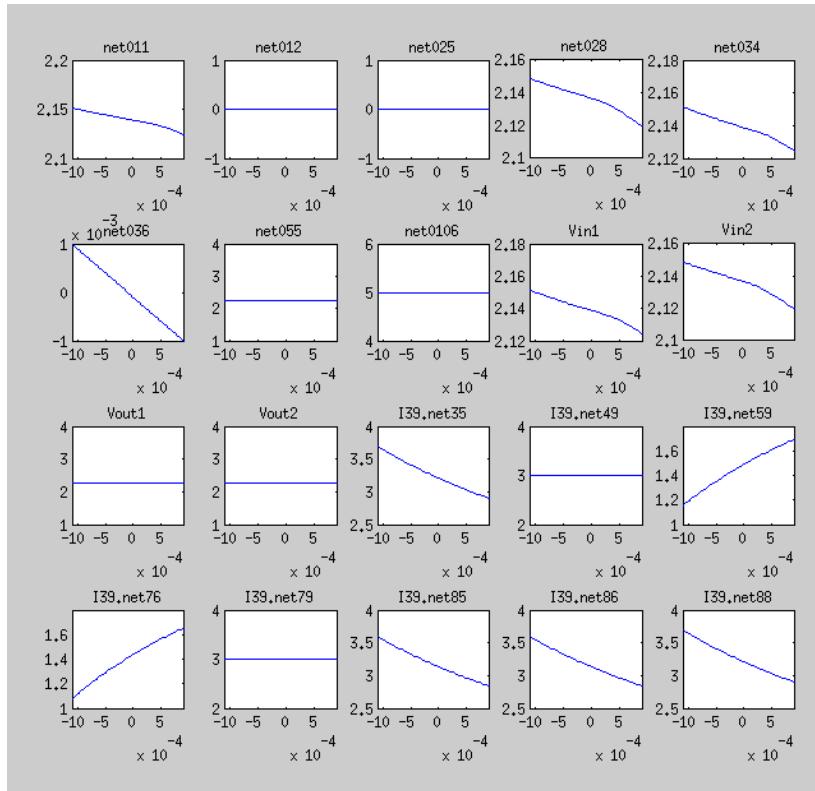


For this circuit, we extracted as an example all the DC intermediary currents and voltages, by using the *Display ALL DC* button.

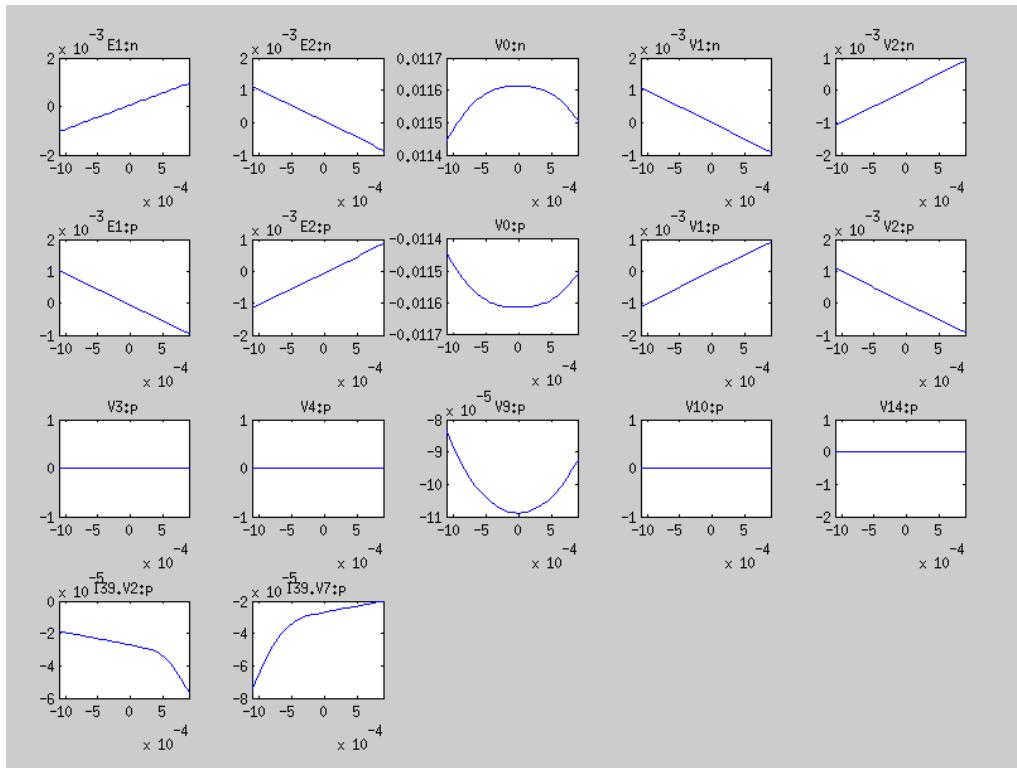
So we have the voltages in all the circuit nets when the input is in differential mode:



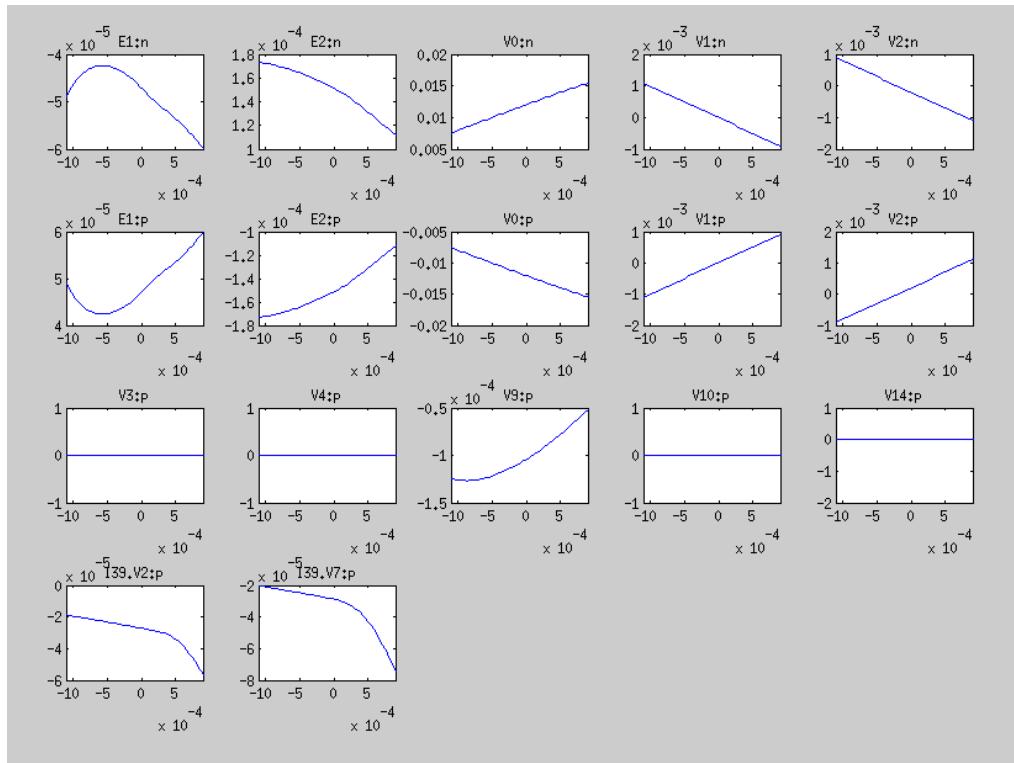
The voltages in all the circuit nets when the input is in common mode:



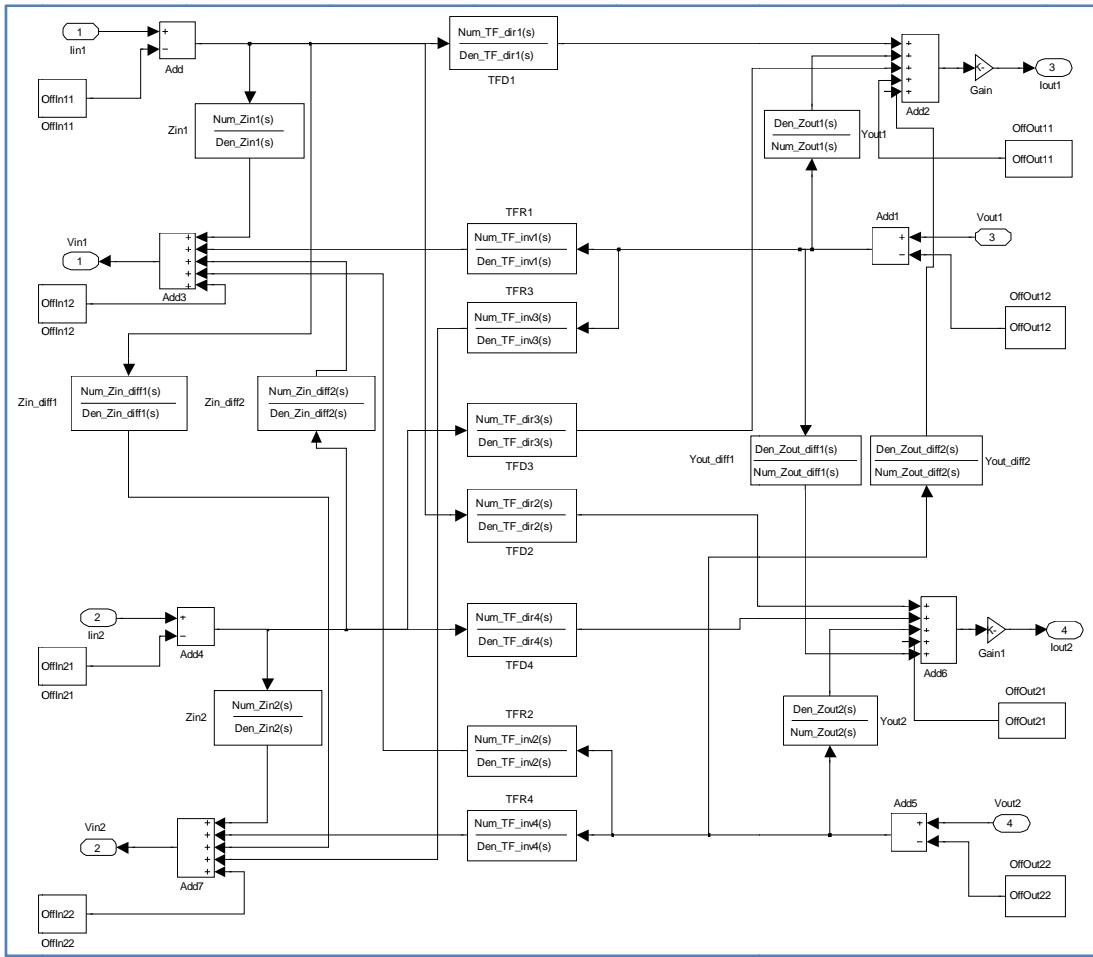
All the intermediary currents when the input is in differential mode:



All the intermediary currents when the input is in common mode:

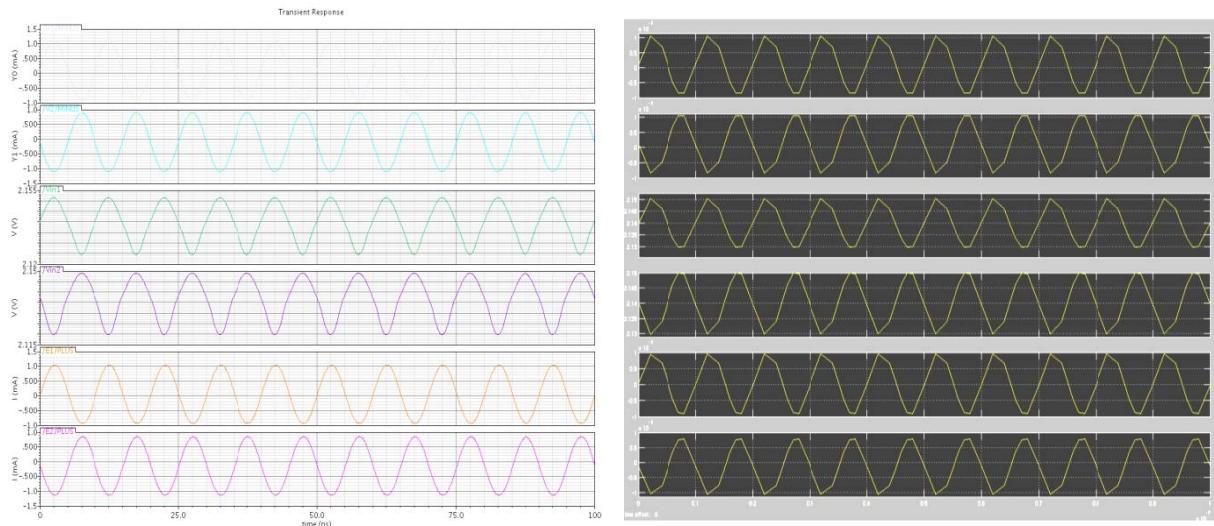


The SIMULINK model of the circuit will have the following structure:



For the VHDL-AMS model, the two files extracted will be: [test_i_diff_i_diff.vhd](#) and [test_i_diff_i_diff.pat](#). For the CADENCE VHDL-AMS we will have: [test_i_diff_i_diff_cdn.vhms](#) and for VerilogA [test_i_diff_i_diff_cdn.va](#).

The comparative simulations under Cadence Analog Environment (transistor-level) and SIMULINK (macro-model) are:

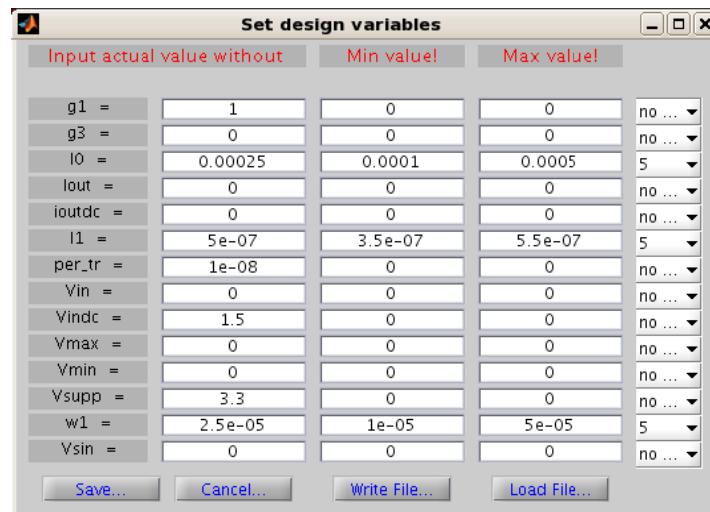


5. Optimization example

We will perform the optimization of the voltage follower presented in 4.1. The name of the circuit schematic is *simect_v_uni_v_uni_sch* and the name of the testbench schematic is *test_v_uni_v_uni*.

The macro-parametric analysis needed for optimization can be started either in SIMECT_GUI or SIMECT_PAR, as explained in Section 3.5.

Launching SIMECT_GUI for running macro-parametric analyses will result in completing the design variables window like:



We can use also the SIMECT_PAR.m script for batch mode parametric simulations.

SIMECT_PAR will use the two types of files:

- a variant file (e.g. [var_test_v_uni_v_uni.m](#)) which defines the circuit characteristics, the design variables and the analyses to be performed. This file is compatible with SIMECT_GUI and SIMECT batch and can be automatically generated by using the *Save* option on the graphical interface;
- a parameters file (e.g. [param_test_v_uni_v_uni.m](#)) which defines the design variables to be varied and the variation domain.

The file contains an index *ind* which is used to add variables to the variation *param* vector.

```
ind=0 ;
param=[];

ind=ind+1 ;
param(ind).nam='I0';
param(ind).min=100e-6 ;
param(ind).max=500e-6 ;
param(ind).npt=5 ;

ind=ind+1 ;
param(ind).nam='w1';
param(ind).min=10e-6 ;
```

```

param(ind).max=50e-6 ;
param(ind).npt=9 ;

ind=ind+1 ;
param(ind).nam='l1';
param(ind).min=0.35e-6 ;
param(ind).max=0.55e-6 ;
param(ind).npt=5 ;

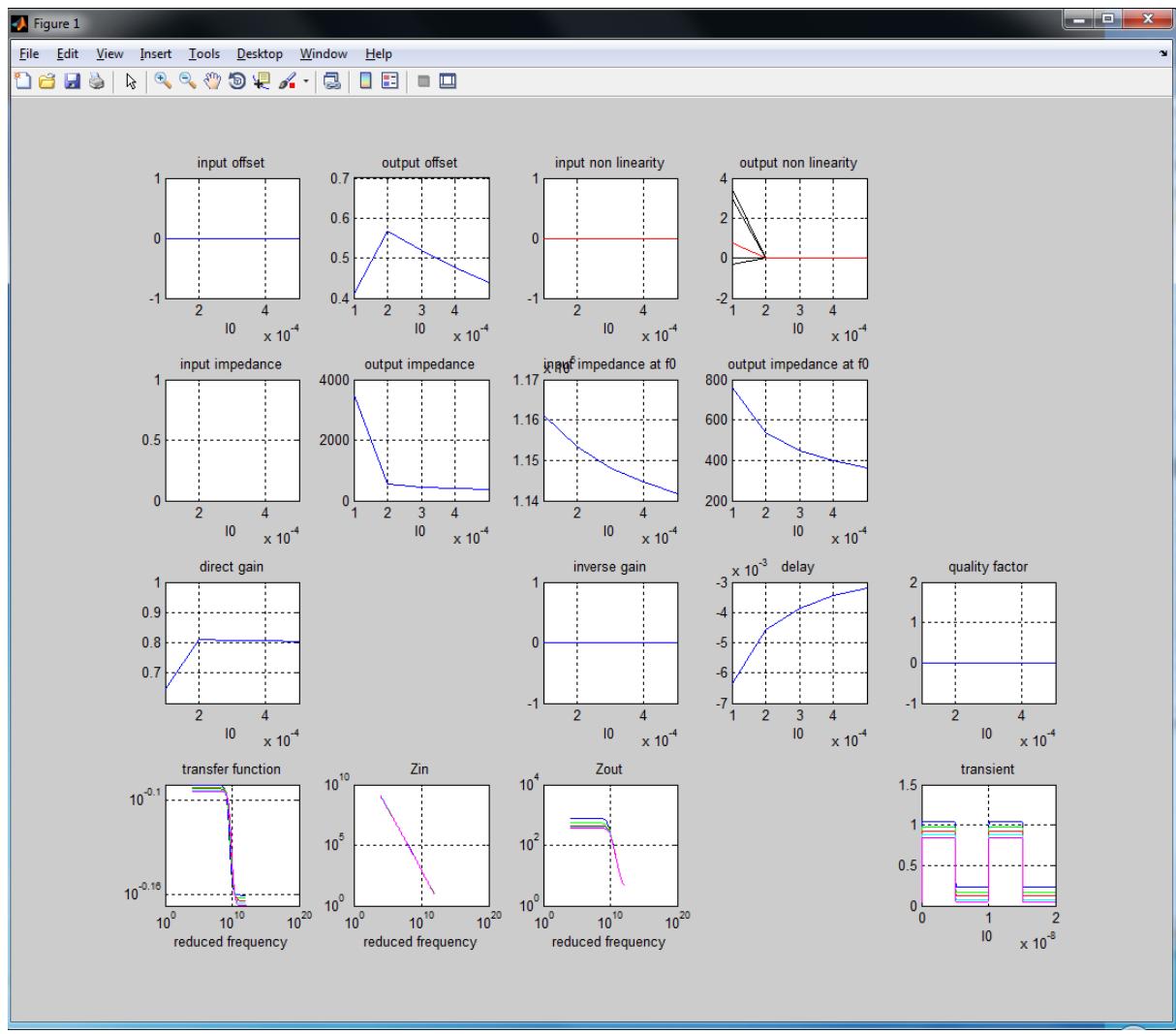
```

In this example we make 3 parametric simulation sets; in the first case the bias current I_0 varies from $100\mu\text{A}$ to $500 \mu\text{A}$, in the second the transistor width varies from $10 \mu\text{m}$ to $50 \mu\text{m}$, and in the third its length varies from 0.35 to $0.55 \mu\text{m}$.

Each parametric simulation set result is gathered in one figure. With this figure, it is possible to deduce which characteristic depends on the parameter and optimize it (manually).

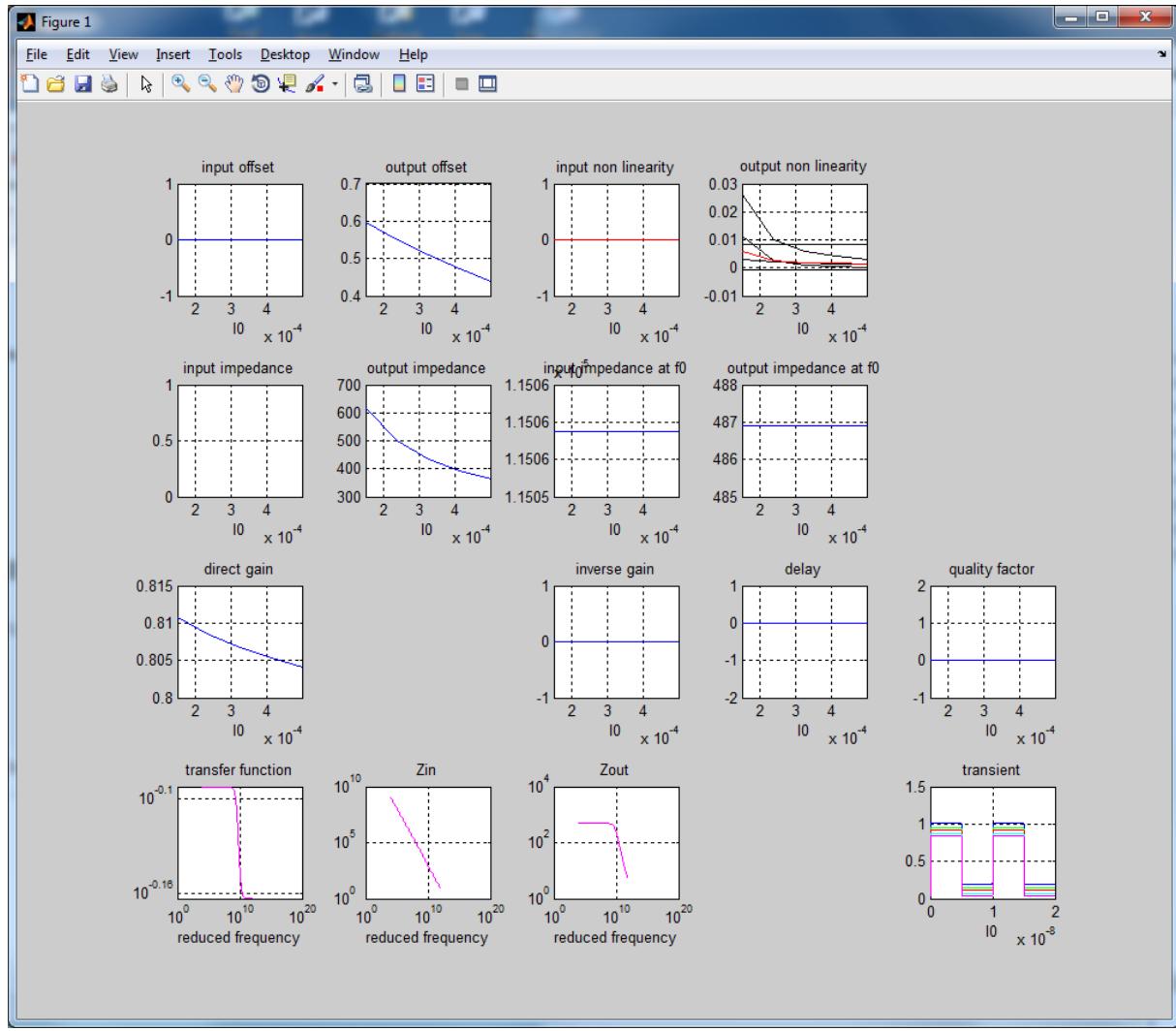
The plotted characteristics are:

- Input and output offset
- Direct and reverse gain
- Transfer function
- Input and output impedance
- Delay at a designated frequency
- Input and output non linearity
- Transient response



It can be seen clearly from the first figure that $I_0=100 \mu\text{A}$ induces a non linearity due to the fact that the transistor gets out of its linear functionality.

So the current source must be changed from $150 \mu\text{A}$ to $500 \mu\text{A}$ to avoid these non-linearities. It results in the second figure.



The two following figures show the effect of the transistor sizes. In order to reduce the output impedance, the transistor length should be as small as possible and its width as high as possible. It can also be seen that the gain can hardly reach 1 showing that this schematic is not a good voltage follower.

