

---

# **OSIM Documentation**

***Release 0.1***

**Tim Maiwald**

**May 25, 2017**



## CONTENTS:

<b>1</b>	<b>Modeling</b>	<b>3</b>
<b>2</b>	<b>Simulation</b>	<b>7</b>
<b>3</b>	<b>Optimization</b>	<b>9</b>
<b>4</b>	<b>Indices and tables</b>	<b>11</b>
	<b>Python Module Index</b>	<b>13</b>



OSIM is a analog circuit simulation tool with the main target to explore new circuit optimization algorithms and methods. It can be separated into three layer:



## MODELING

The Modeling-Layer is the fundamental part of the mathematical description of an analog circuit. Therefor exist two again

### 1.1 Circuit System Equations

```
class Modeling.CircuitSystemEquations.CircuitSystemEquations (components)
    Bases: object

    Class that contains the main data structures for a circuit

    ATYPE_AC = 3
    ATYPE_DC = 2
    ATYPE_EST_DC = 5
    ATYPE_NONE = 1
    ATYPE_TRAN = 4

    getCompByName (name)
    getPreviousSolutionAt (componentName)
    getSolutionAt (componentName)
        method to get a value out of the solution vector x

        Parameters componentName (string) – name of a component or a node
            in the netlist

        Returns solution

        Return type complex

    isVoltage (name)
    printComponents ()
```

```
reset ()  
    method to reset the equations  
setParameterForCompsList (setables)  
    Parameters setables – list([compname,paramname,paramval],[...])  
setParameterForComp (compname, paramname, paramval)  
setSolutionAt (name, val)
```

## 1.2 Components

### 1.2.1 OSIM.Modeling.AbstractComponents.Component module

```
class Modeling.AbstractComponents.Component.Component (nodes, name,  
                                                    value, super-  
                                                    Component,  
                                                    **kwargs)  
  
Bases: object  
Abstract description of a component in a circuit  
  
GMIN = 1e-09  
  
Udiff (twonodes)  
  
assignToSystem (sys)  
  
containsNonlinearity ()  
  
doStep (freq_or_tau)  
    abstract function to use in a iteration of simulations  
    Parameters freq_or_tau (float) – frequency or timestep value de-  
    pending on simulation type  
  
getAdmittance (nodesFromTo, freq_or_tstep)  
  
getMyParameterFromDictionary (param, paramDict, default)  
  
getNodes ()  
  
getValue ()  
  
initialSignIntoSysEquations ()  
  
insertAdmittanceintoSystem (freq)  
  
myBranchCurrent ()  
  
parseArgs (**kwargs)
```



```
performCalculations ()  
printMyOPValues ()  
putA (A, node, myIdx, nm, mn)  
putJ (J, m, node, mn)  
readParamsAndVariables (filename)  
    ACHTUNG: BEIM AUSDRUECKE MIT VARIABLEN MUESSEN FLOATS EN-  
    THALTEN UND KEINE INTEGER; WEIL SONST GEFAHR; DASS NICHT  
    GERUNDET WIRD!!  
registerBranches (compDict, sysIdx)  
registerNodes (compDict, sysIdx)  
reloadParams ()  
setNewParamsAndVariablesDicts (paramDict, variableDict)  
setOPValues ()  
setParameterOrVariableValue (name, value)  
setValue (value)  
signIntoSysDictionary (compDict, sysIdx)  
updateParamsAndVariablesDicts (paramDict, variableDict)  
updateSuperCompositeComponent ()  
voltageOverMe ()
```



## SIMULATION

### 2.1 Simulation.NetToComp

Use the triangle class to represent triangles.

**class** `Simulation.NetToComp.NetToComp` (*filename*)

Bases: `object`

Beispielkommentar

**getCommentsFromNetlist** (*netListFile*)

Create a triangle with sides of lengths *a*, *b*, and *c*.

Raises *ValueError* if the three length values provided cannot actually form a triangle.

**getComponents** ()

Create a triangle with sides of lengths *a*, *b*, and *c*.

Raises *ValueError* if the three length values provided cannot actually form a triangle.

**parseCommentsToArgs** (*args*, *commentList*, *name*)

**stringArrToDict** (*strArr*)

Create a triangle with sides of lengths *a*, *b*, and *c*.

Raises *ValueError* if the three length values provided cannot actually form a triangle.



## OPTIMIZATION

### 3.1 Optimizer

```
class Optimizations.OptimizationComponents.Optimizable.Optimizable (comp_names_li  
                                                                    param-  
                                                                    name,  
                                                                    val-  
                                                                    from,  
                                                                    valto,  
                                                                    **kwargs)
```

Bases: object

**getOptimizableComponentNames** ()

**getParamName** ()

**getRangeBegin** ()

**getRangeEnd** ()

**static getSetableList** (olist)

**getValue** ()

**setValue** (v)

**toString** ()



## INDICES AND TABLES

- genindex
- modindex
- search





## PYTHON MODULE INDEX

### m

`Modeling.AbstractComponents.Component,`  
4

`Modeling.CircuitSystemEquations,`  
3

### O

`Optimizations.OptimizationComponents.Optimizable,`  
9

### S

`Simulation.NetToComp,` 7

# INDEX

assignToSystem()	(Model- ing.AbstractComponents.Component.Component method), 1	initialSignIntoSysEquations()	(Model- ing.AbstractComponents.Component.Component method), 1
Component	(class in Model- ing.AbstractComponents.Component), 1	insertAdmittanceintoSystem()	(Model- ing.AbstractComponents.Component.Component method), 1
containsNonlinearity()	(Model- ing.AbstractComponents.Component.Component method), 1	Modeling.AbstractComponents.Component	(module), 1
doStep()	(Model- ing.AbstractComponents.Component.Component method), 1	myBranchCurrent()	(Model- ing.AbstractComponents.Component.Component method), 1
		NetToComp	(class in Simulation.NetToComp), 3
getAdmittance()	(Model- ing.AbstractComponents.Component.Component method), 1	parseArgs()	(Model- ing.AbstractComponents.Component.Component method), 1
getCommentsFromNetlist()	(Simula- tion.NetToComp.NetToComp method), 3	parseCommentsToArgs()	(Simula- tion.NetToComp.NetToComp method), 3
getComponents()	(Simula- tion.NetToComp.NetToComp method), 3	performCalculations()	(Model- ing.AbstractComponents.Component.Component method), 1
getMyParameterFromDictionary()	(Model- ing.AbstractComponents.Component.Component method), 1	printMyOPValues()	(Model- ing.AbstractComponents.Component.Component method), 1
getNodes()	(Model- ing.AbstractComponents.Component.Component method), 1	putA()	(Model- ing.AbstractComponents.Component.Component method), 1
getValue()	(Model- ing.AbstractComponents.Component.Component method), 1	putJ()	(Model- ing.AbstractComponents.Component.Component method), 1
GMIN	(Model- ing.AbstractComponents.Component.Component attribute), 1	readParams()	(Model- ing.AbstractComponents.Component.Component method), 1

method), [1](#)

`readParams()` (Simulation.NetToComp.NetToComp static method), [3](#)

`registerBranches()` (Modeling.AbstractComponents.Component.Component method), [1](#)

`registerNodes()` (Modeling.AbstractComponents.Component.Component method), [1](#)

`setOPValues()` (Modeling.AbstractComponents.Component.Component method), [1](#)

`setValue()` (Modeling.AbstractComponents.Component.Component method), [1](#)

`signIntoSysDictionary()` (Modeling.AbstractComponents.Component.Component method), [1](#)

`Simulation.NetToComp` (module), [3](#)

`stringArrToDict()` (Simulation.NetToComp.NetToComp method), [3](#)

`Udiff()` (Modeling.AbstractComponents.Component.Component method), [1](#)

`updateSuperCompositeComponent()` (Modeling.AbstractComponents.Component.Component method), [1](#)

`voltageOverMe()` (Modeling.AbstractComponents.Component.Component method), [2](#)