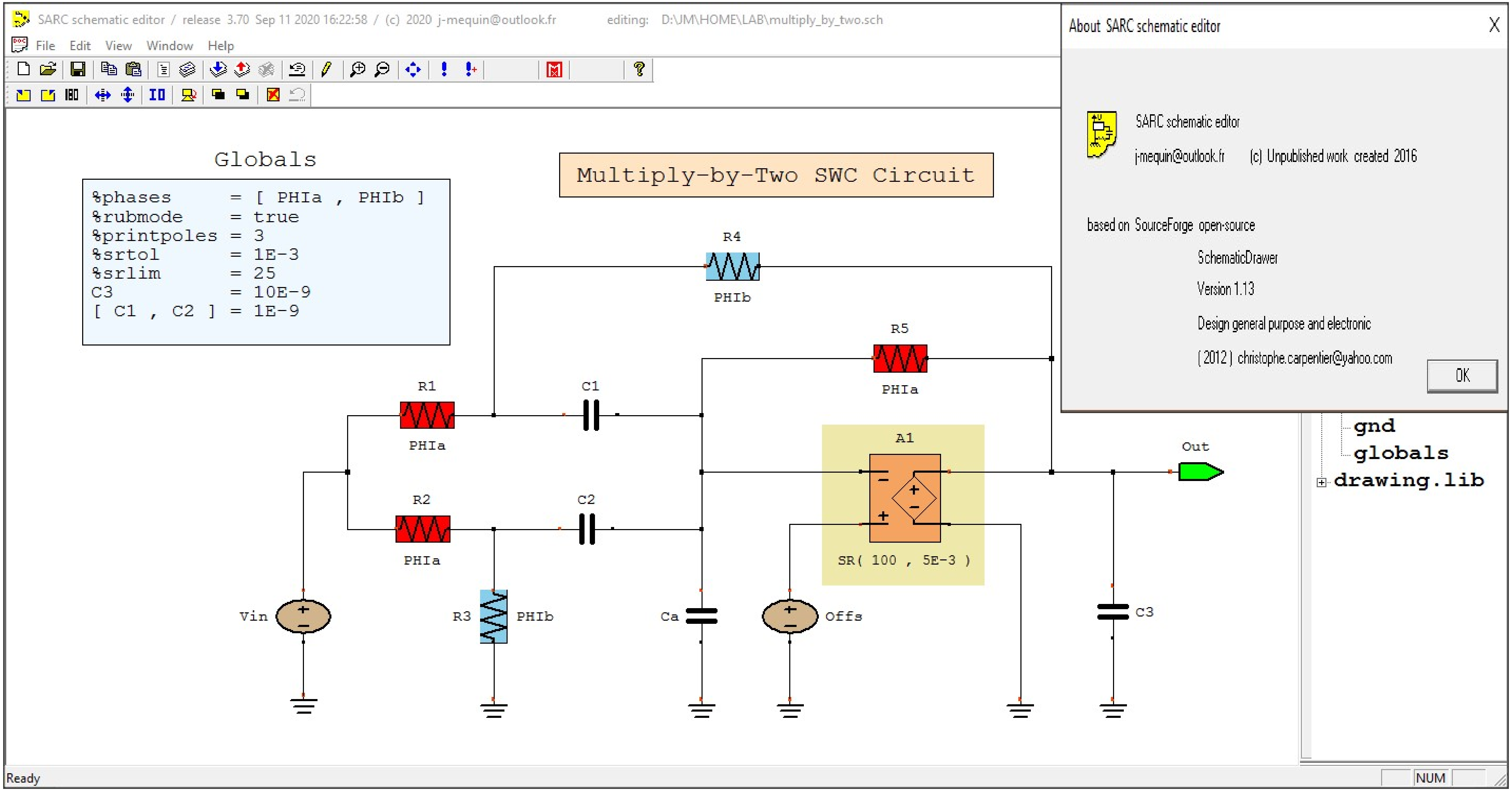


SARC schematic editor / release 3.70 Sep 11 2020 16:22:58 / (c) 2020 j-mequin@outlook.fr editing: D:\JM\HOME\LAB\multiply_by_two.sch

```
%phases     = [ PHIa , PHIb ]
%rubmode    = true
%printpoles = 3
%srtol      = 1E-3
%srlim      = 25
C3          = 10E-9
[ C1 , C2 ] = 1E-9
```

About SARC schematic editor

SARC schematic editor

j-mequin@outlook.fr (c) Unpublished work created 2016

based on SourceForge open-source

SchematicDrawer

Version 1.13

Design general purpose and electronic

(2012) christophe.carpentier@yahoo.com

revision 7.48

WARNING :

Attempting to change the value of some components during a simulation will not interrupt the execution but its outcome may become ***silently misleading***

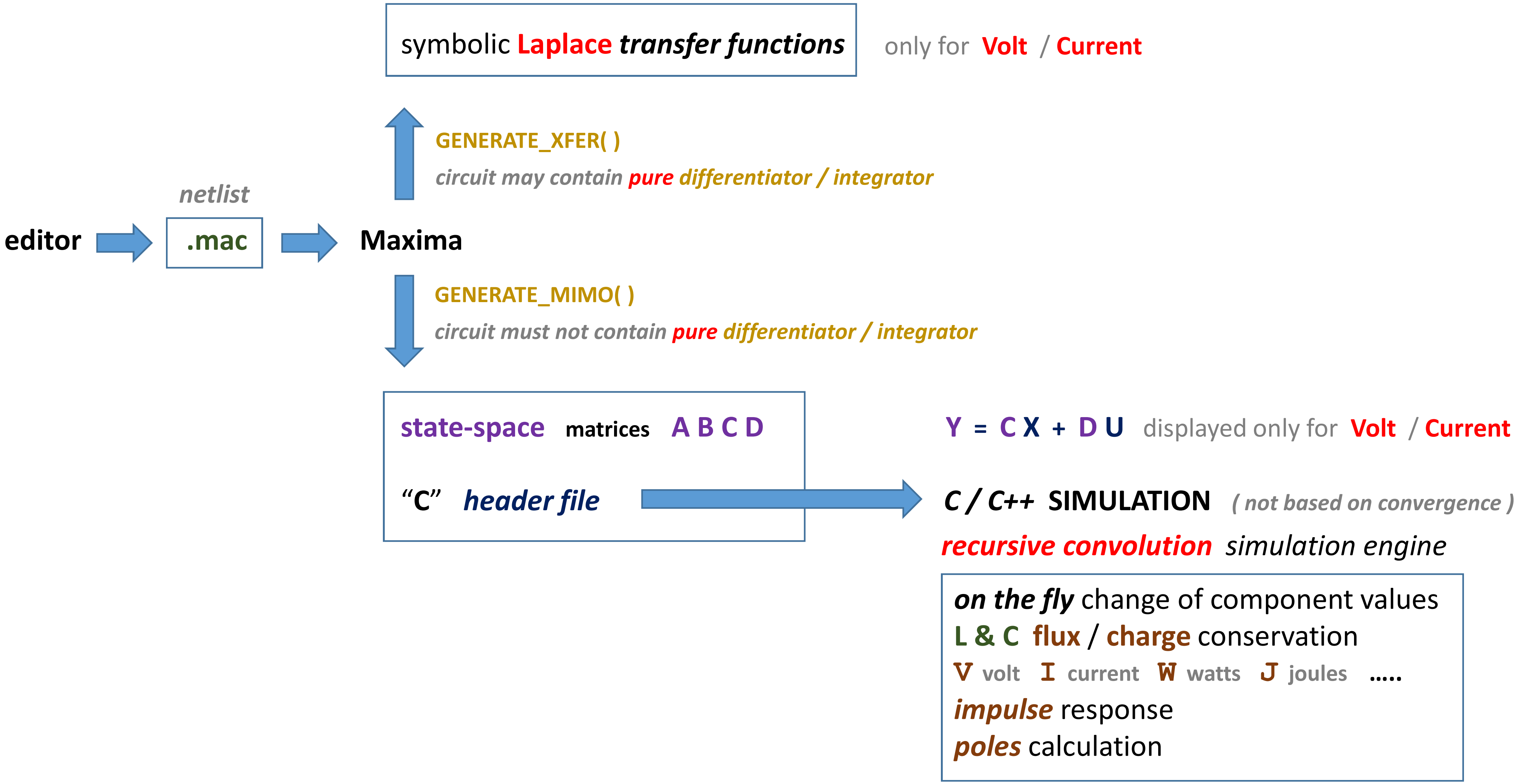
Depending on the* topology *of the circuit, some ***flux*** and/or ***charge*** may have to be partially redistributed on some other components
(trivial example : *changing the value of a capacitor in parallel with another one*)

Therefore, simulating accurately component "***percussions***" requires some additional analysis, equations and processing

For ***Laplace transfer function*** (**XFER / ILT**) and ***Z transform*** (**ZTRANS**) such functionality has not been implemented and simulations may turn to be ***silently misleading***

In summary,

only *STATE model* (MIMO) is able to simulate accurately "*on the fly*" component changes



symbolic **Laplace *transfer functions*** only for **Volt** / **Current**

GENERATE_XFER( )

*circuit may contain **pure differentiator / integrator***

netlist

editor → **.mac** → **Maxima**

GENERATE_MIMO( )

*circuit must not contain **pure differentiator / integrator***

state-space **matrices** **A B C D**

"C" ***header file***

Y = C X + D U displayed only for **Volt** / **Current**

C / C++ **SIMULATION** ***(not based on convergence)***

recursive convolution *simulation engine*

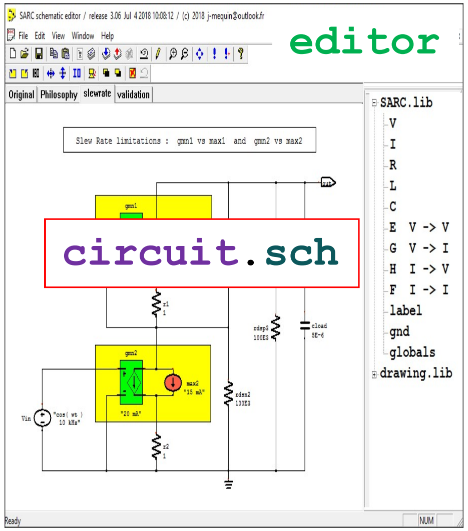
on the fly change of component values

L & C **flux** / **charge** conservation

V volt **I** current **W** watts **J** joules **.....**

impulse response

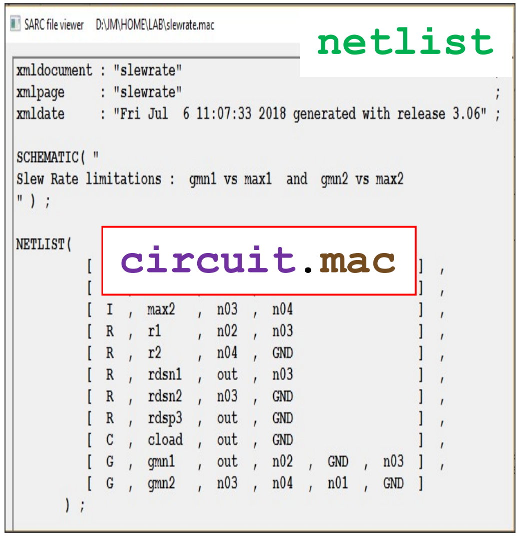
poles calculation



editor

circuit.sch

1

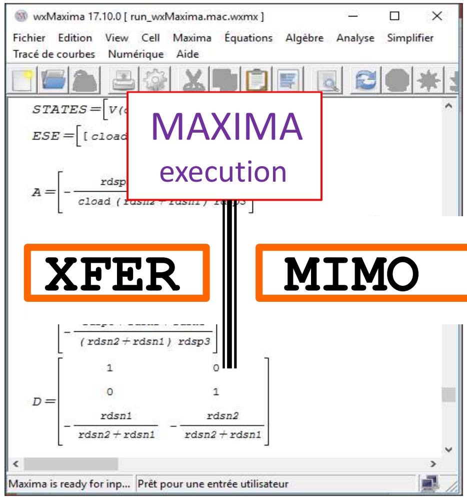

SARC file viewer D:\JM\HOME\LAB\slewrate.mac

```
xmldocument : "slewrate"
xmlpage     : "slewrate"                                          ;
xmldate     : "Fri Jul  6 11:07:33 2018 generated with release 3.06" ;

SCHEMATIC( "
Slew Rate limitations :  gmn1 vs max1  and  gmn2 vs max2
" ) ;

NETLIST(
        [                                               ]  ,
        [                                               ]  ,
        [  I  ,  max2   ,  n03  ,  n04                  ]  ,
        [  R  ,  r1     ,  n02  ,  n03                  ]  ,
        [  R  ,  r2     ,  n04  ,  GND                  ]  ,
        [  R  ,  rdsn1  ,  out  ,  n03                  ]  ,
        [  R  ,  rdsn2  ,  n03  ,  GND                  ]  ,
        [  R  ,  rdsp3  ,  out  ,  GND                  ]  ,
        [  C  ,  cload  ,  out  ,  GND                  ]  ,
        [  G  ,  gmn1   ,  out  ,  n02  ,  GND  ,  n03  ]  ,
        [  G  ,  gmn2   ,  n03  ,  n04  ,  n01  ,  GND  ]
     ) ;
```

netlist

circuit.mac

2


wxMaxima 17.10.0 [run_wxMaxima.mac.wxmx]
Fichier
Edition
View
Cell
Maxima
Équations
Algèbre
Analyse
Simplifier
Tracé de courbes
Numérique
Aide
Maxima is ready for inp...
Prêt pour une entrée utilisateur

STATES=[V(

ESE=[[cload

$$A=\left[-\frac{rdsp}{cload\,(rdsn2+rdsn1)\,rdsp3}\right]$$

$$\left[-\frac{}{(rdsn2+rdsn1)\,rdsp3}\right]$$

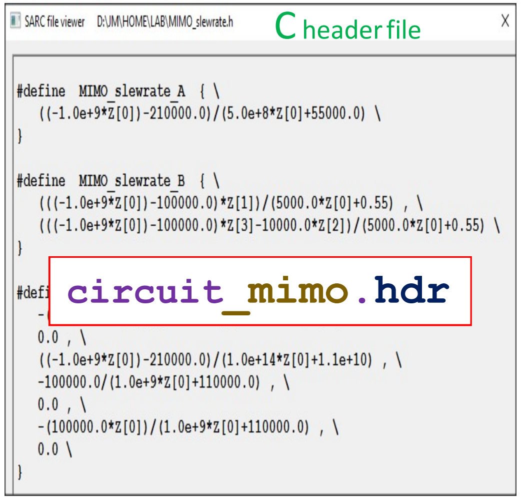
$$D=\begin{bmatrix}1 & 0\\ 0 & 1\\ -\frac{rdsn1}{rdsn2+rdsn1} & -\frac{rdsn2}{rdsn2+rdsn1}\end{bmatrix}$$

MAXIMA
execution

XFER

MIMO

3


SARC file viewer D:\JM\HOME\LAB\MIMO_slewrate.h

```
#define  MIMO_slewrate_A  { \
   ((-1.0e+9*Z[0])-210000.0)/(5.0e+8*Z[0]+55000.0) \
}

#define  MIMO_slewrate_B  { \
   (((-1.0e+9*Z[0])-100000.0)*Z[1])/(5000.0*Z[0]+0.55) , \
   (((-1.0e+9*Z[0])-100000.0)*Z[3]-10000.0*Z[2])/(5000.0*Z[0]+0.55) \
}

#defi
   -(
   0.0 , \
   ((-1.0e+9*Z[0])-210000.0)/(1.0e+14*Z[0]+1.1e+10) , \
   -100000.0/(1.0e+9*Z[0]+110000.0) , \
   0.0 , \
   -(100000.0*Z[0])/(1.0e+9*Z[0]+110000.0) , \
   0.0 \
}
```

C header file

circuit_mimo.hdr

4

NAPA

A **schematic XML** file is called a **“ DOCUMENT ”** that can be

displayed on a resizable window

that may contain several **“ PAGEs ”**

Each **“ PAGE ”** is a ***“ container ”***

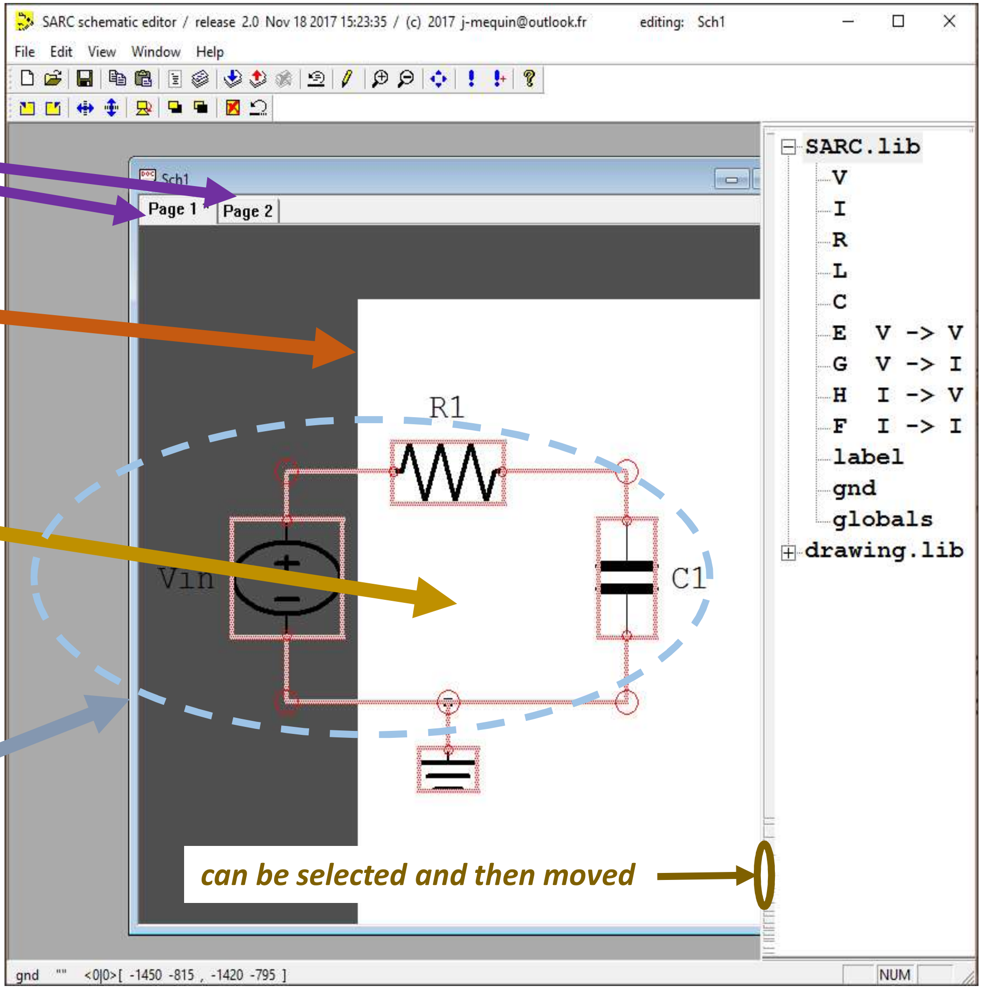
including the

available surface for the drawing

by default **width** = 2970 **height** = 2100

(thus ***landscape***)

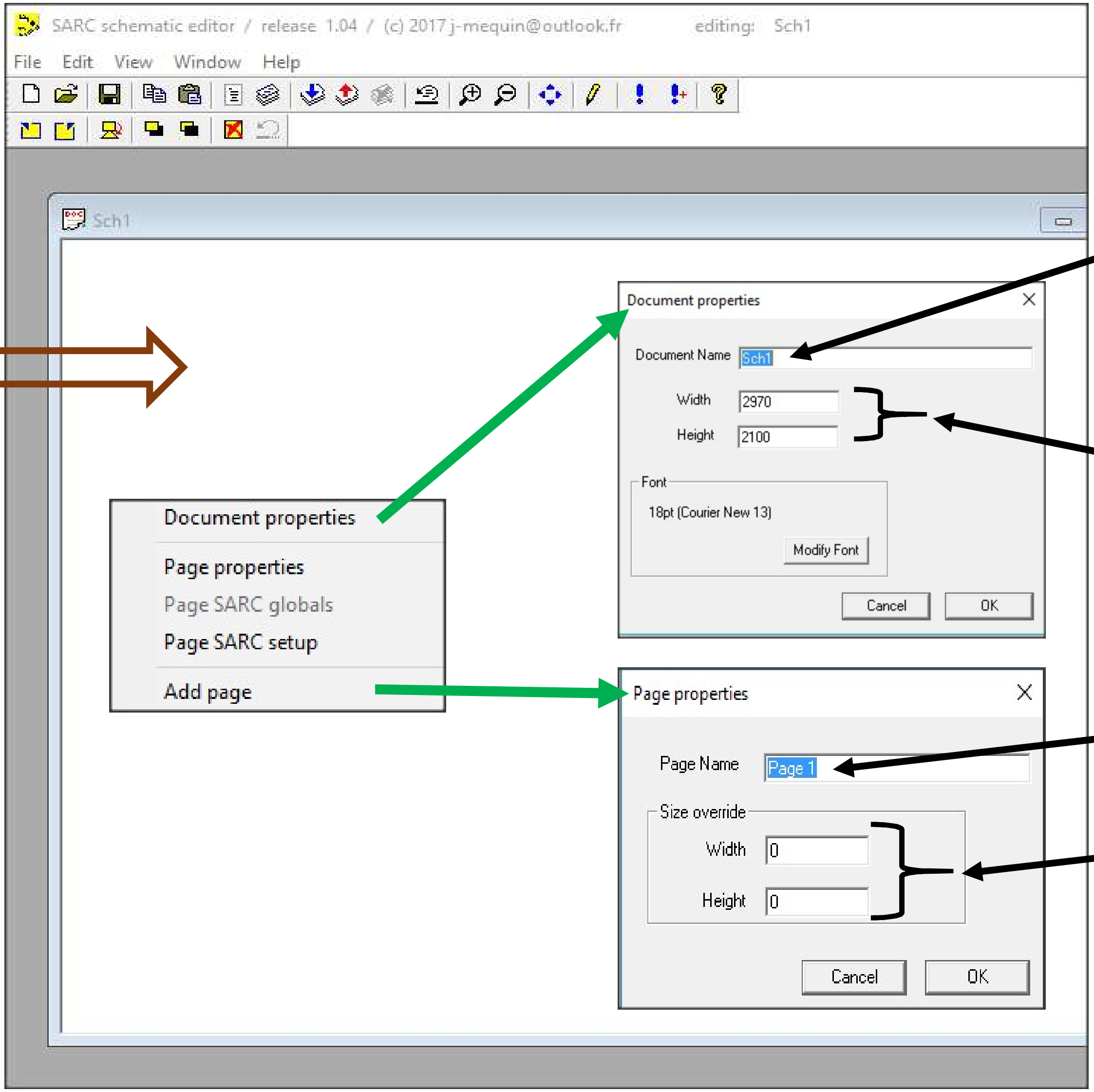
this whole schematic needs to be moved to the ***printable area***


SARC schematic editor / release 2.0 Nov 18 2017 15:23:35 / (c) 2017 j-mequin@outlook.fr
editing: Sch1
File Edit View Window Help
Sch1
Page 1
Page 2
R1
Vin
C1
SARC.lib
V
I
R
L
C
E V -> V
G V -> I
H I -> V
F I -> I
label
gnd
globals
drawing.lib
gnd "" <0|0>[-1450 -815 , -1420 -795]
NUM

can be selected and then moved

click right

on an empty zone



SARC schematic editor / release 1.04 / (c) 2017 j-mequin@outlook.fr editing: Sch1

Document properties

Document Name: Sch1

Width: 2970

Height: 2100

Font: 18pt (Courier New 13)

Page properties

Page Name: Page 1

Size override

Width: 0

Height: 0

document **name**

will remain coherent with

the **XML** and **MAC** ***filenames***

width and ***height***

of the **default** ***printable area***

page **name**

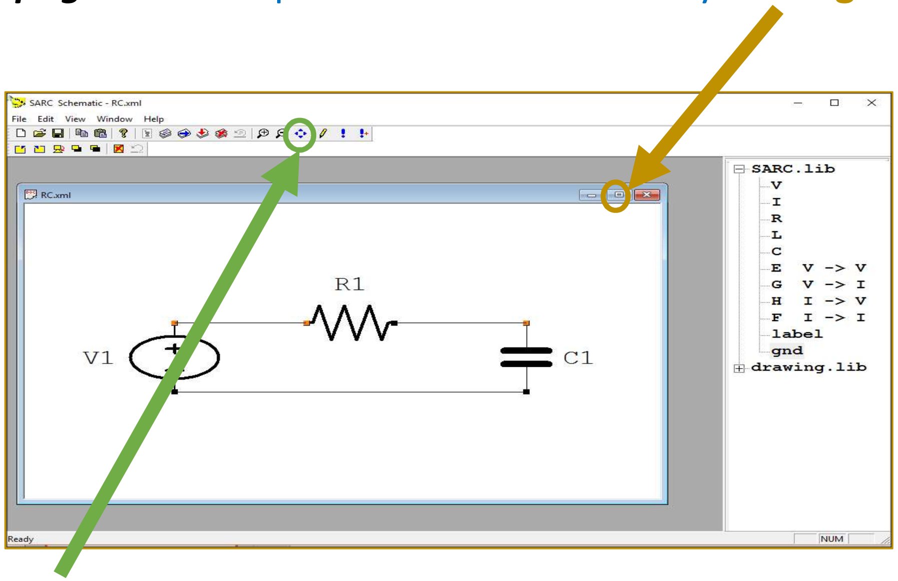
width and ***height***

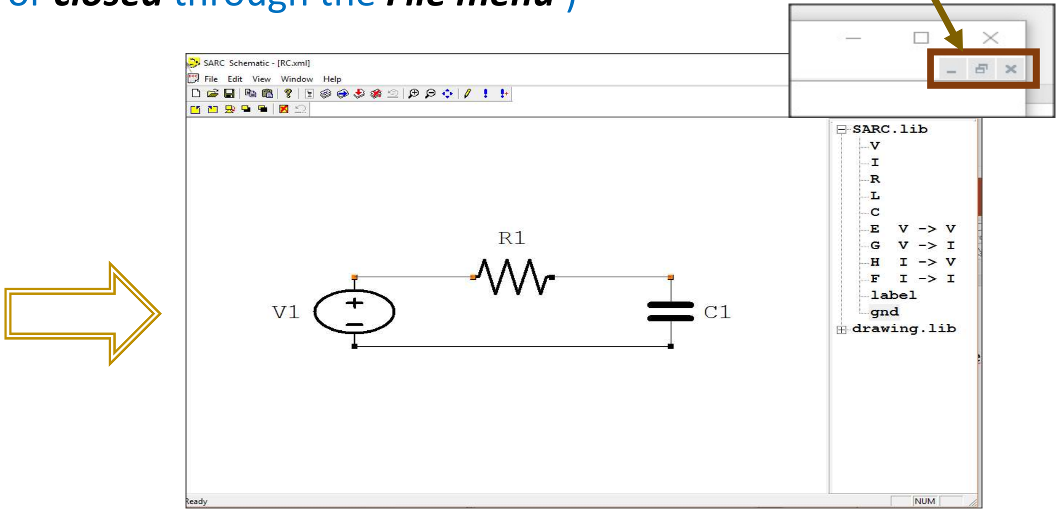
of **this** ***page area***

(***0*** means same as **document**)

Several **XML** files can be active at the same time, allowing to ***cut*** / ***paste*** selected item(s) between them

A ***single page*** can be expanded to the full area by clicking (and then , can still be *minimized* , *restored* , *closed* or ***closed*** through the ***File menu***)





The ✥ icon allows **to rescale** the schematic in order to fulfill the visible window area (= ZOOM to fit)

After positioning the mouse, the ***space bar*** **centers** the view ***without modifying the scale***

After positioning the mouse, **Page UP** increases the zoom by **25%**, **Page Down** decreases the zoom by **1/25%**, **around the mouse**
or Ctrl + **or Ctrl -**

The ***mouse wheel*** can be also used to control the zoom

Any object on the " **Page** " can be **selected** by ***clicking left*** on it

Once selected, the objects can be	***copied***	***deleted***	***pasted***	***moved***
	ctrl-C	***del key** or **ctrl-X***	***ctrl-V***	***click left***

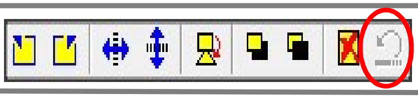
Objects can be ***deselected*** by clicking on an ***empty area*** or with the ***ESCAPE*** key

Holding **Ctrl** **key** allows **to select** ***all the objects*** that are inside a ***rectangular zone*** ***delimited with the mouse clicking left***

Once selected, the ***area*** can be	***copied***	***deleted***	***pasted***	***moved***
	ctrl-C	***del key** or **ctrl-X***	***ctrl-V***	holding ***ctrl*** **key** **+** ***click left***

An ***area*** can be ***deselected*** by clicking ***elsewhere*** or with the ***ESCAPE*** key

Holding **both *Ctrl*** and ***Alt***, and then **moving the mouse clicked left** , allows to move EVERYTHING within the ***page container***

ctrl-Z undo undo last edit operation(**s**) ⟺  **undo** *icon*

ctrl-A	select	***everything*** on the ***page***	**F2**	check page for missing connection diagnostics
ctrl-C	copy	whatever is ***selected*** to the clipboard	**F3**	display net names (after netlist generation)
ctrl-V	paste	the contents of the clipboard	**F4**	playback all LIFO stack undo items

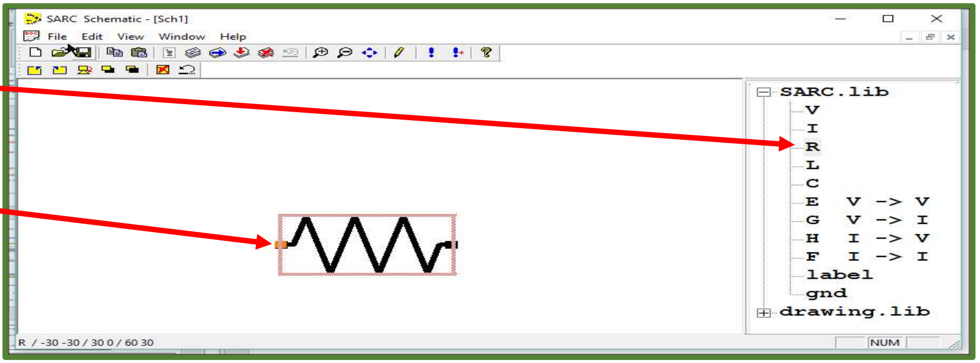
ctrl-F search an ***instance name*** ***(or a** **NET** **name)*** in the schematic

To add a ***component***, select it from the ***SARC library***

and ***then*** ***double click left*** on the ***page***

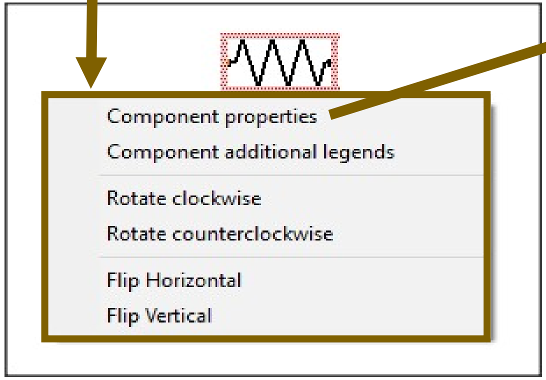
The ***component*** will ***remain selected*** until another one is chosen

Therefore, ***double clicks*** can be used to place ***multiple instances***

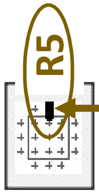


The original software allows to ***GROW*** *or* ***SHRINK*** *components with* ***SHIFT + the 4 arrows***

double click left
or ***click right*** on *a selected component*,
pop-up

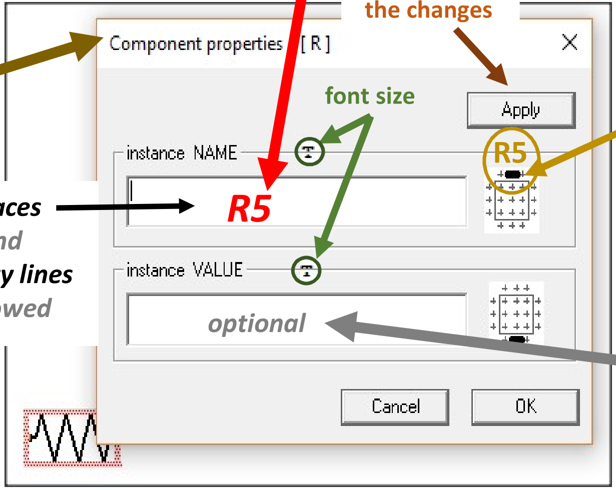


instance name required



click right

allows to see the changes



font size

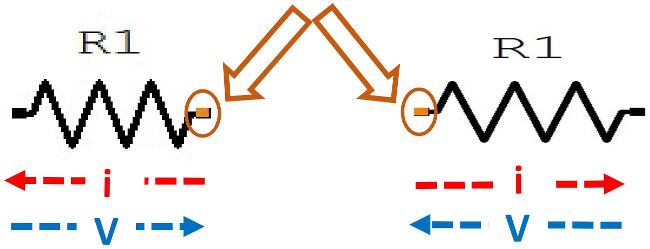
spaces *and* ***empty lines*** *allowed*

optional

Selecting another ***location*** and ***orientation*** of this ***rectangular mark*** changes the ***location*** and ***orientation*** of the ***displayed text***

see the slide about "globals"

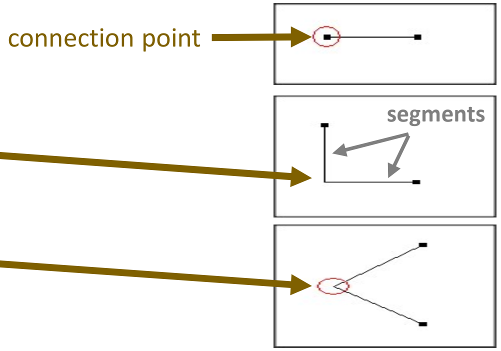
On each ***SARC*** **component**, an ***orange mark*** allows to identify, ***despite possible rotations***, its **SARC** ***netlist pins order***



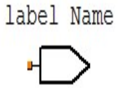
Holding **Alt key** while ***clicking mouse left*** allows to ***draw a*** **WIRE** ***segment***

While holding the **Alt key**, ***toggling*** the ***mouse left*** allows to ***change direction***

Without holding the **Alt key**, ***clicking*** the ***mouse left*** allows to ***drag*** the ***selected location*** of an existing **WIRE**

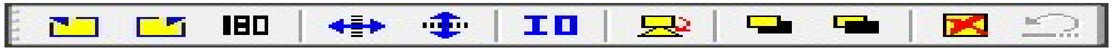


Pressing **F1** allows to make disappear **BOTH** the ***orange marks*** and the ***connection points*** *(example, for nicer plot)*

The ***SARC*** component ***" label "***  can be used to associate a ***name*** to a **NET** *(overriding its default " **nxx** ")*

" Distant ***labels*** *" sharing* ***same name*** *are* ***wirelessly*** *connected !!!*

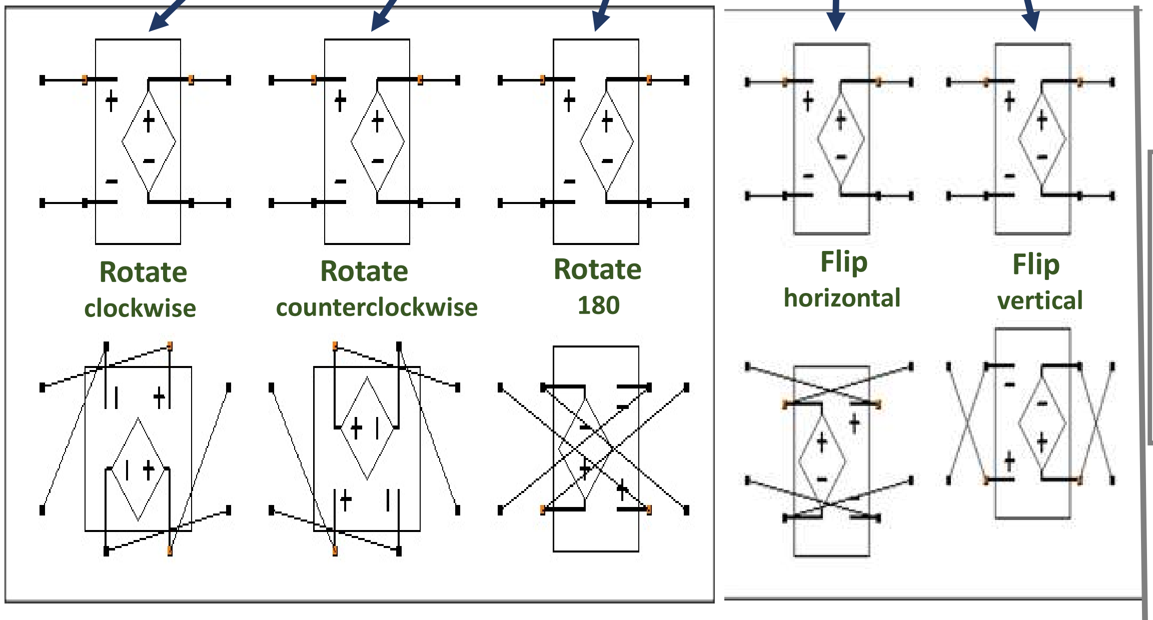
The objects **" Title "** or **" Sticker "** of the ***" drawing.lib "*** can be used to add some ***legends*** to the schematic



“**undo**” **previous** edit / ctrl-Z

from the *history* queue

delete **selected** object(**s**) / ctrl-X



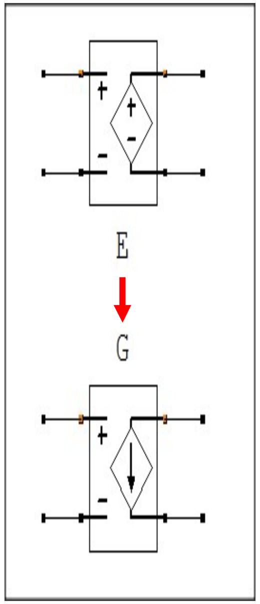
I/O swap can be used

to reverse V I R L C

and only outputs of

E H G F

change component **type**

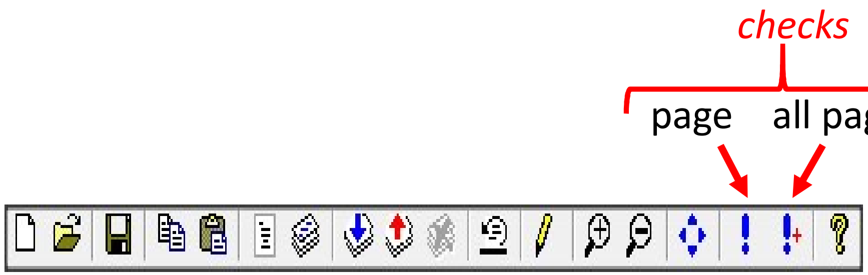


connected nets remain attached

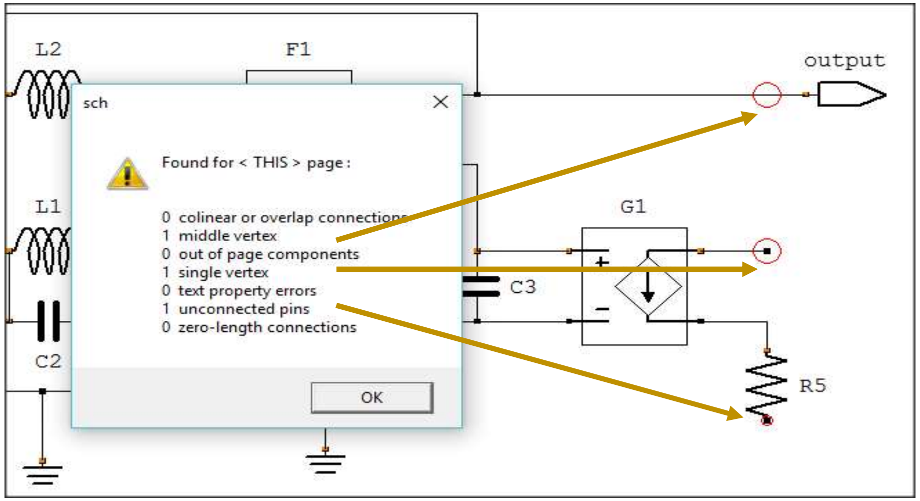
Before the ***netlist generation*** it is **mandatory** to perform some ***sanity checks*** on the schematic(**s**)

Those checks are ***almost the ones of the original software*** and are ***not as exhaustive as*** the ***SARC Maxima*** version

Therefore, even if a netlist is later generated, it can still contains some remaining issues

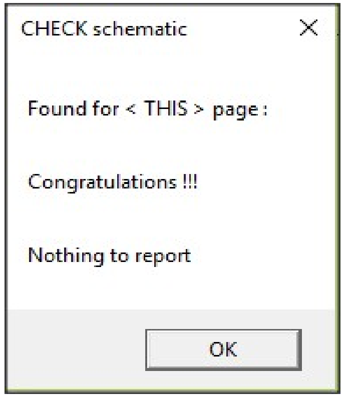


On the current page, it is also possible to display the diagnostics of the checks, ***but without the pop-up window***, by pressing **F2**



Found for < THIS > page :

```
0  colinear or overlap connections
1  middle vertex
0  out of page components
1  single vertex
0  text property errors
1  unconnected pins
0  zero-length connections
```



CHECK schematic

Found for < THIS > page :

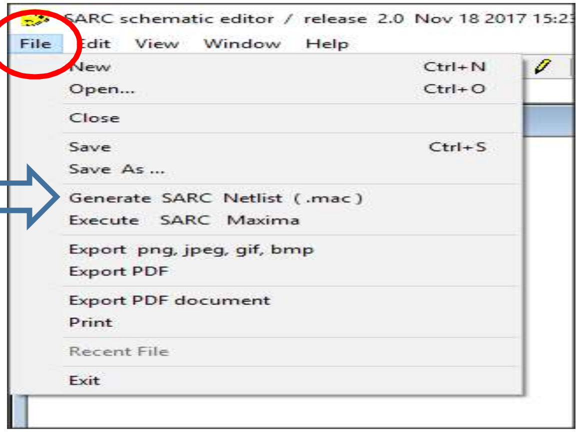
Congratulations !!!

Nothing to report

Netlist generation

same ***file location*** that
the associated ***XML***

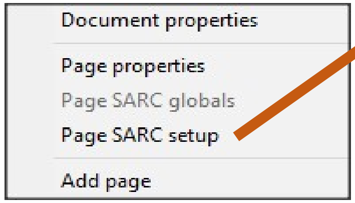
File menu

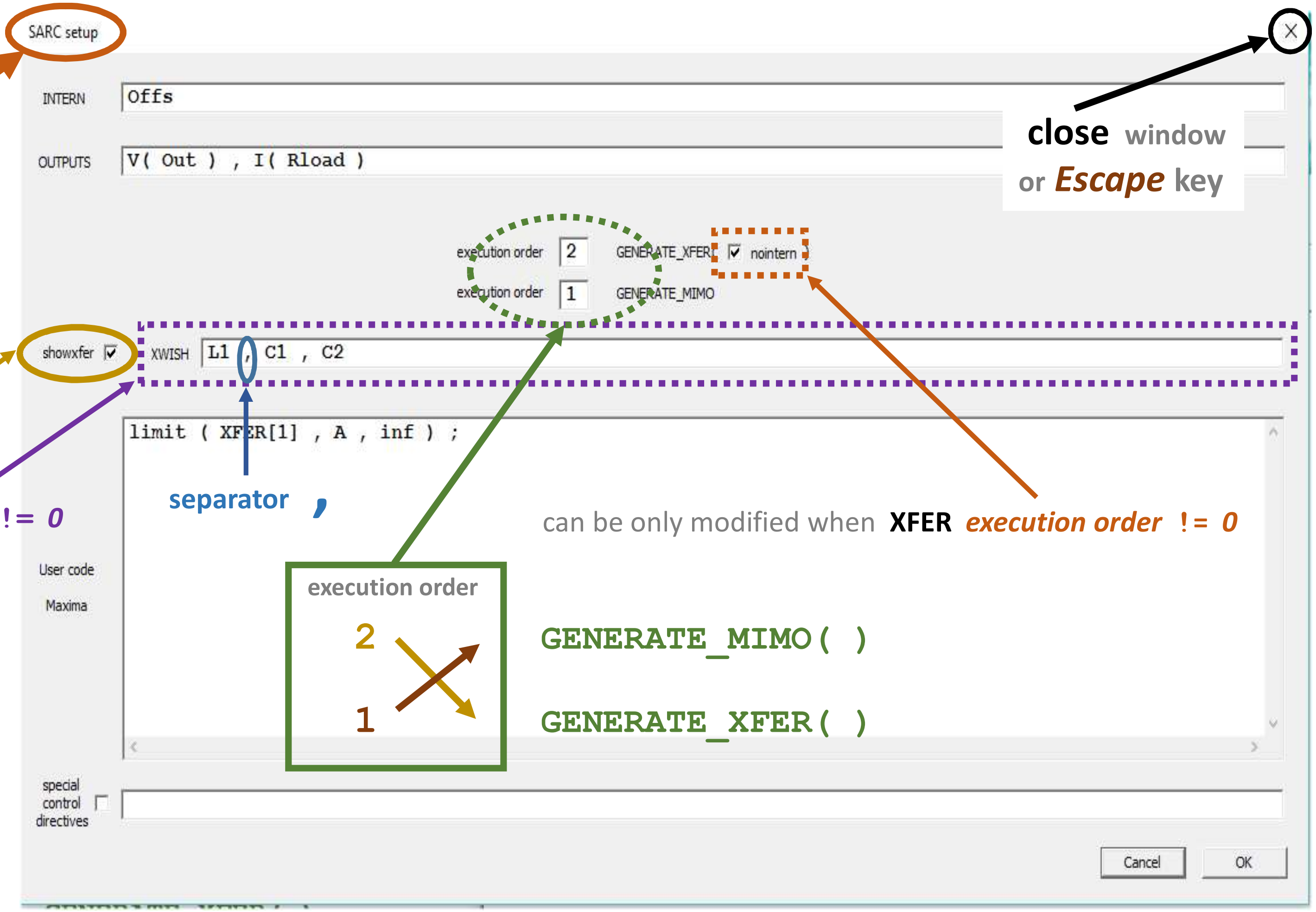


Maxima **SARC parameters** can be

collected and ***saved*** within the **XML**

However, this release ***does not yet check*** them





INTERN `Offs`

OUTPUTS `V( Out ) , I( Rload )`

execution order 2 GENERATE_XFER ☑ nointern

execution order 1 GENERATE_MIMO

showxfer ☑ XWISH `L1 , C1 , C2`

User code Maxima

```
limit ( XFER[1] , A , inf ) ;
```

special control directives □

Cancel OK

close window or ***Escape*** key

separator ,

can be only modified when **XFER** ***execution order*** **!= 0**

execution order

2

1

GENERATE_MIMO( )

GENERATE_XFER( )

generate XFER or MIMO with/without display

can be only modified when **MIMO** ***execution order*** ***!= 0***

document property “ ***name*** ”

(that is related to ***filename***)

page property “ ***name*** ”

generation ***date***

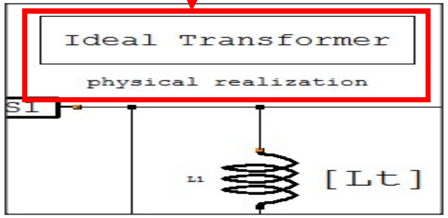
when defined ,

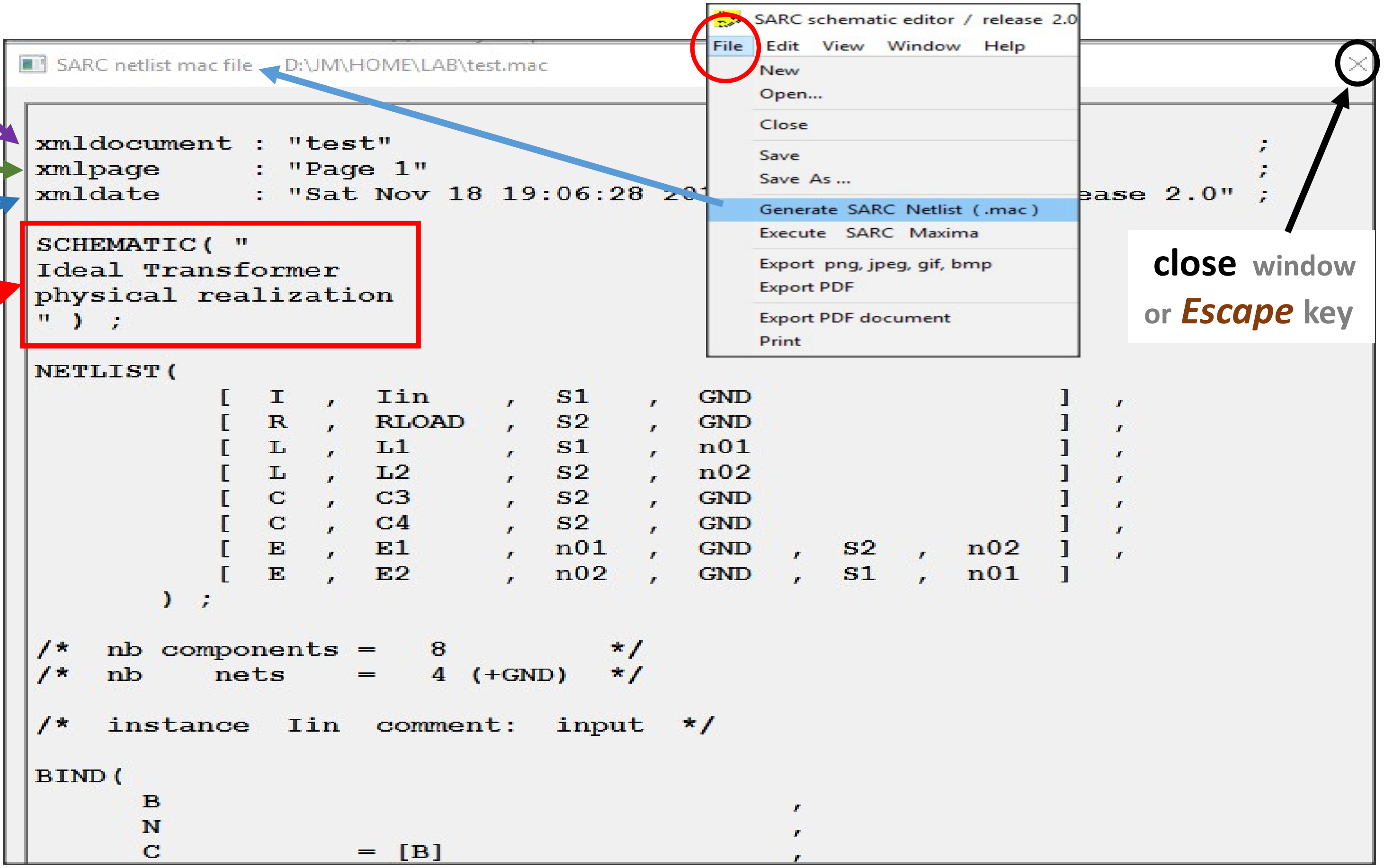
those lines come from

drawing.lib

“ **Title** ” objects

“ ***Component additional legends*** ”





SARC netlist mac file D:\JM\HOME\LAB\test.mac
SARC schematic editor / release 2.0
File
Edit
View
Window
Help
New
Open...
Close
Save
Save As ...
Generate SARC Netlist (.mac)
Execute SARC Maxima
Export png, jpeg, gif, bmp
Export PDF
Export PDF document
Print

```
xmldocument  :  "test"                                                    ;
xmlpage      :  "Page 1"                                                  ;
xmldate      :  "Sat Nov 18 19:06:28 20                        ease 2.0"  ;

SCHEMATIC( "
Ideal Transformer
physical realization
" ) ;

NETLIST(
          [  I  ,  Iin    ,  S1   ,  GND                      ]  ,
          [  R  ,  RLOAD  ,  S2   ,  GND                      ]  ,
          [  L  ,  L1     ,  S1   ,  n01                      ]  ,
          [  L  ,  L2     ,  S2   ,  n02                      ]  ,
          [  C  ,  C3     ,  S2   ,  GND                      ]  ,
          [  C  ,  C4     ,  S2   ,  GND                      ]  ,
          [  E  ,  E1     ,  n01  ,  GND  ,  S2  ,  n02       ]  ,
          [  E  ,  E2     ,  n02  ,  GND  ,  S1  ,  n01       ]
       ) ;

/*  nb components =   8        */
/*  nb    nets    =   4 (+GND) */

/*  instance  Iin  comment:  input  */

BIND(
      B                              ,
      N                              ,
      C          = [B]               ,
```

close window

or ***Escape*** key

In case of several **Title** use « ***Bring to front*** » icon to select one of them

Escape-S can be used to view the ***current*** schematic **“.sch”** file

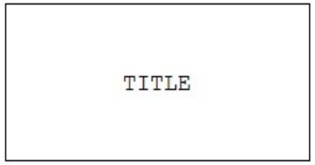
Escape-C can be used to view the **colormap**

Escape-P can be used to view (**AFTER** ***schematic Save / Save as***) the ***latest*** generated **“.pdf”** file

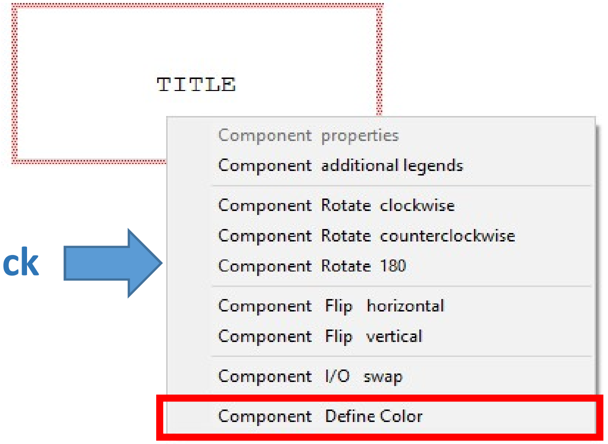
Escape-M can be used to view (**AFTER** ***generate SARC Netlist***) the ***latest*** generated **“.mac”** file

Escape-H can be used to view (**AFTER** ***GENERATE_MIMO( )***) the ***latest*** generated **“.hdr”** file

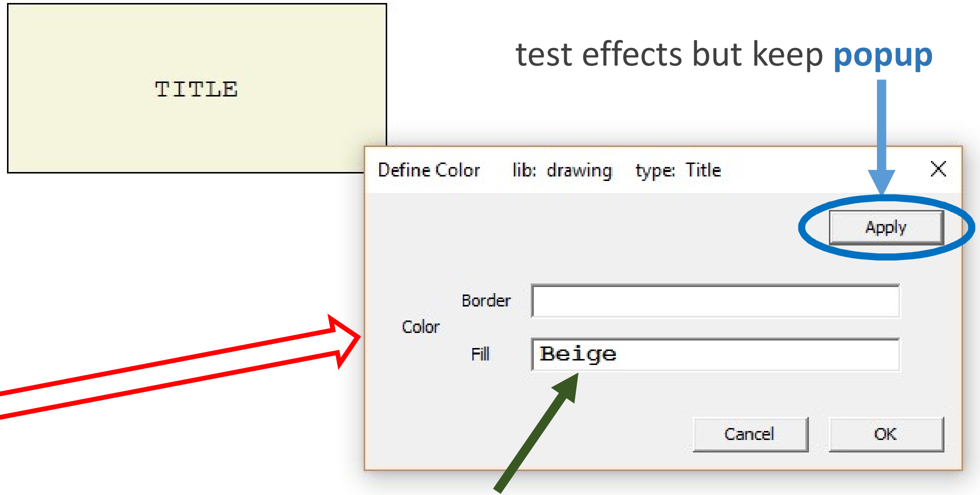
Escape-Z can be used to view (**AFTER** ***GENERATE_Z( )***) the ***latest*** generated **“.hdr”** file



right click


TITLE
Component properties
Component additional legends
Component Rotate clockwise
Component Rotate counterclockwise
Component Rotate 180
Component Flip horizontal
Component Flip vertical
Component I/O swap
Component Define Color

test effects but keep **popup**



Color

Border

Fill Beige

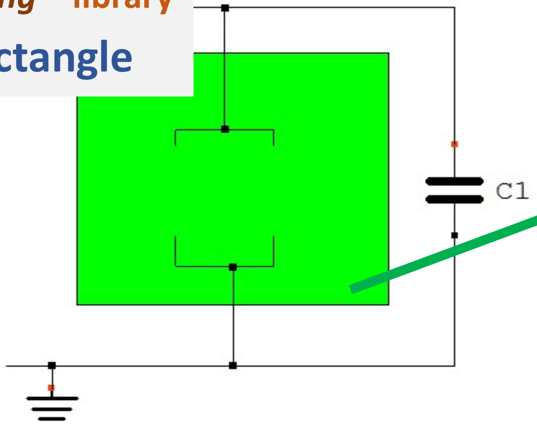
define in file ***colormap*** (***Escape-C***)

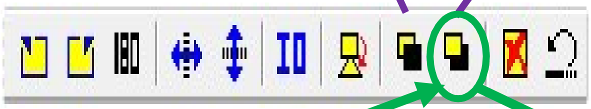
Bring to Front

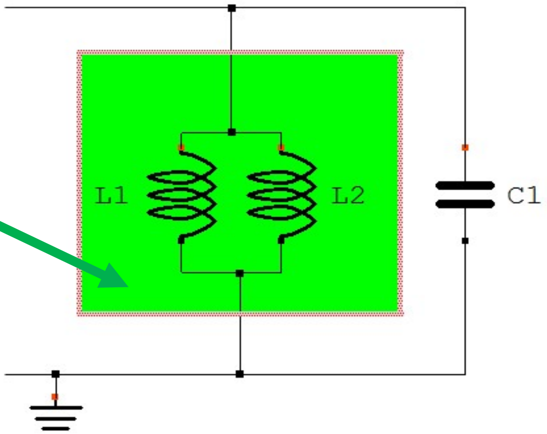
Bring to Back

"drawing" library

Rectangle







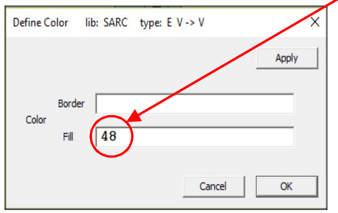
ESC - C

1 Black	2 Navy	3 DarkBlue	4 MediumBlue
5 Blue	6 DarkGreen	7 Green	8 Teal
9 DarkCyan	10 DeepSkyBlue	11 DarkTurquoise	12 MediumSpringGreen
13 Lime	14 SpringGreen	15 Aqua	16 Cyan
17 MidnightBlue	18 DodgerBlue	19 LightSeaGreen	20 ForestGreen
21 SeaGreen	22 DarkSlateGray	23 LimeGreen	24 MediumSeaGreen
25 Turquoise	26 RoyalBlue	27 SteelBlue	28 DarkSlateBlue
29 MediumTurquoise	30 Indigo	31 DarkOliveGreen	32 CadetBlue
33 CornflowerBlue	34 MediumAquamarine	35 DimGray	36 SlateBlue
37 OliveDrab	38 SlateGray	39 LightSlateGray	40 MediumSlateBlue
41 Lawngreen	42 Gray	43 Chartreuse	44 Aquamarine
45 Maroon	46 Purple	47 Olive	48 SkyBlue

49 LightSkyBlue	50 BlueViolet	51 DarkRed	52 DarkMagenta
53 SaddleBrown	54 YellowGreen	55 DarkSeaGreen	56 LightGreen
57 MediumPurple	58 DarkViolet	59 PaleGreen	60 DarkOrchid
61 Navyblue	62 Sienna	63 Brown	64 DarkGray
65 LightBlue	66 GreenYellow	67 PaleTurquoise	68 LightSteelBlue
69 PowderBlue	70 FireBrick	71 DarkGoldenRod	72 MediumOrchid
73 RosyBrown	74 DarkKhaki	75 Silver	76 MediumVioletRed
77 IndianRed	78 Peru	79 Chocolate	80 Tan
81 LightGrey	82 Thistle	83 Orchid	84 GoldenRod
85 PaleVioletRed	86 Crimson	87 Gainsboro	88 Plum
89 Burlywood	90 LightCyan	91 Lavender	92 DarkSalmon
93 Violet	94 PaleGoldenRod	95 LightCoral	96 Khaki

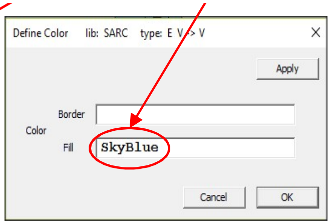
97 AliceBlue	98 HoneyDew	99 Azure	100 SandyBrown
101 Wheat	102 Beige	103 WhiteSmoke	104 MintCream
105 GhostWhite	106 Salmon	107 AntiqueWhite	108 Linen
109 LightGoldenRodYellow	110 OldLace	111 Red	112 Fuchsia
113 Magenta	114 DeepPink	115 OrangeRed	116 Tomato
117 HotPink	118 Coral	119 DarkOrange	120 LightSalmon
121 Orange	122 LightPink	123 Pink	124 Gold
125 PeachPuff	126 NavajoWhite	127 Moccasin	128 Bisque
129 MistyRose	130 PapayaWhip	131 LavenderBlush	132 SeaShell
133 Cornsilk	134 LemonChiffon	135 FloralWhite	136 Snow
137 Yellow	138 BlanchedAlmond	139 LightYellow	140 Ivory
141 White			

COLORS can be defined


Define Color lib: SARC type: E V -> V

Border:

Color Fill: 48

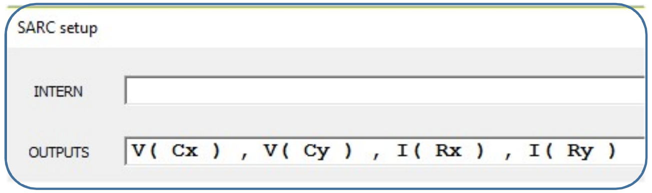

Define Color lib: SARC type: E V -> V

Border:

Color Fill: SkyBlue

by **index** or by **name** (*case insensitive*)

OUTPUT syntax examples :



OUTPUTS `V( Cx ) , V( Cy ) , I( Rx ) , I( Ry )`

V(*label*)	label : netlist node
V(*label1*:*label2*)	voltage between 2 labels (note: GND is a label)
V(*component*)	components : U I R L C and E G H F outputs
I(*component*)	components : U I R L C and E G H F outputs
W(*component*)	components : U I R L C and E G H F outputs
cV(voltage *control*)	control side of E G
cI(current *control*)	control side of H F

. . .

note : *the* Y[ ] *vector* ***displayed*** *by* ***wxMAXIMA*** *only contains* V( ) cV( ) I( ) cI( )

component parameters

string **« hello world »** *comments are ignored by* ***Maxima***

symbolic **3*gain**
k*sqrt(gain)
0.8 / gain

substituted ***at the start*** *of* ***Maxima*** *analysis*

symbolic definitions must reference at least one variable

NOTE : ***special syntax*** **[1] [+1] [-1]** ***(1 is the only accepted « value »)***

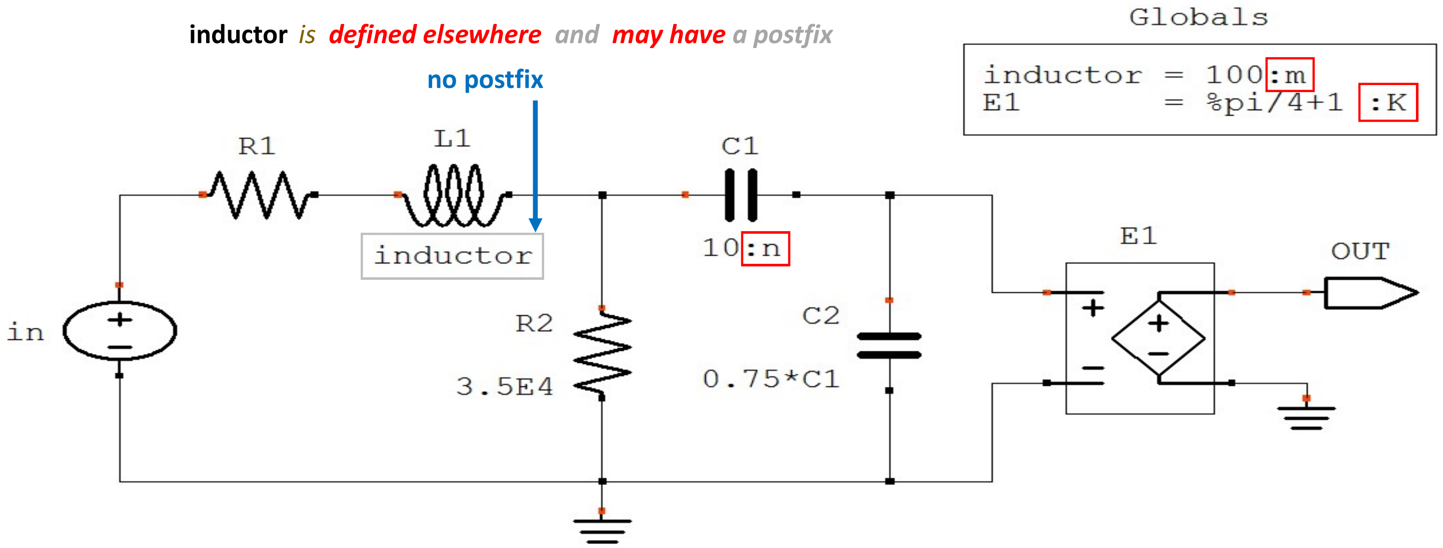
numeric **0.4**
12E-3
sqrt(%pi)

substituted ***at the end*** *of* ***Maxima*** *analysis*

numeric definitions must not reference any variable

inductor *is* ***defined elsewhere*** *and* ***may have*** *a postfix*

no postfix



schematic instance ***numeric values*** can be ***postfixed*** with a multiplicative unit

:P	*peta*	10^+15
:T	*tera*	10^+12
:G	*giga*	10^+9
:M	*mega*	10^+6
:K	*kilo*	10^+3

uppercase

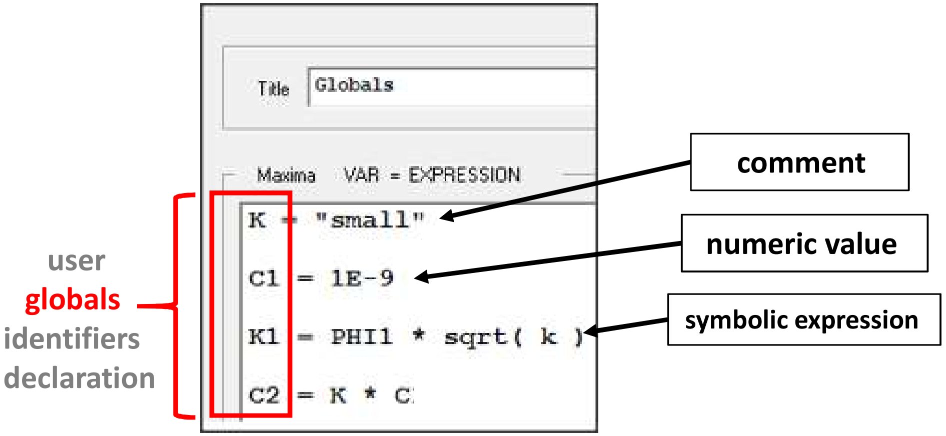
:m	*mili*	10^-3
:u	*micro*	10^-6
:n	*nano*	10^-9
:p	*pico*	10^-12
:f	*femto*	10^-15

lowercase

Globals

A *special SARC component* named « *globals* » can be instantiated on the page

Its purpose is to define *circuit global parameters*
referenced by *instance values* or other *globals*

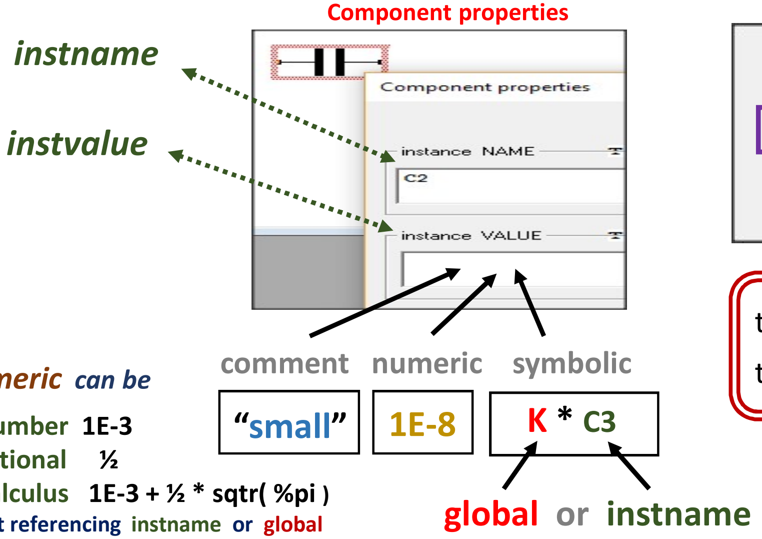


Title: Globals

Maxima VAR = EXPRESSION

```
K  = "small"
C1 = 1E-9
K1 = PHI1 * sqrt( k )
C2 = K * C
```

Component properties



Component properties

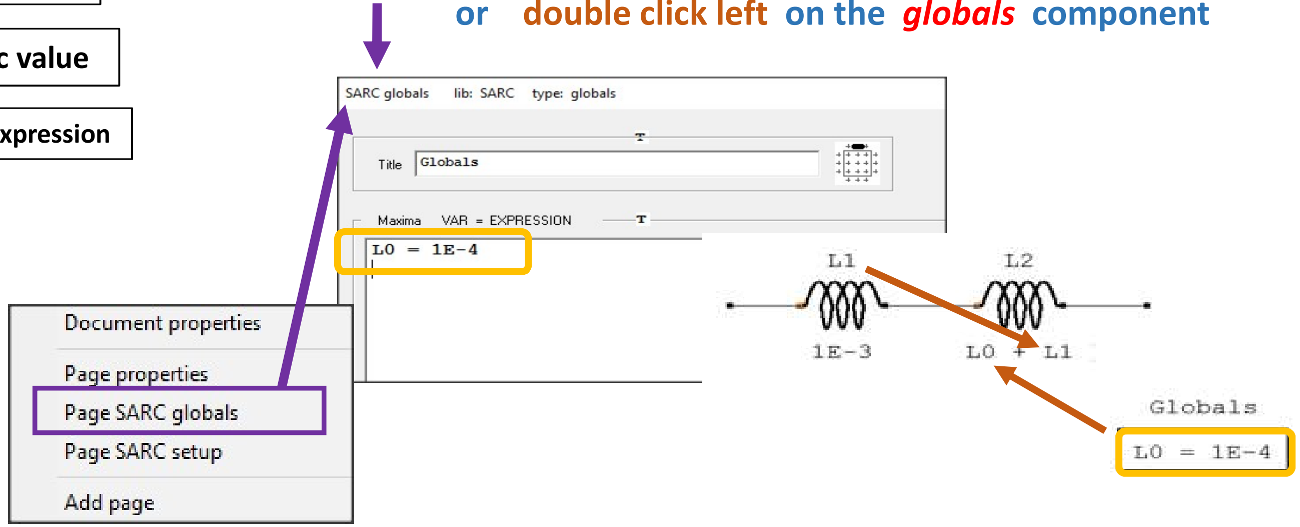
instance NAME: C2

instance VALUE:

a numeric can be

- a number 1E-3
- a rational ½
- a calculus 1E-3 + ½ * sqrt(%pi)
  not referencing instname or global

After placing the *globals* component, the *global menu* can be accessed by right click on an *empty area* of the screen
or double click left on the *globals* component



SARC globals lib: SARC type: globals

Title: Globals

Maxima VAR = EXPRESSION

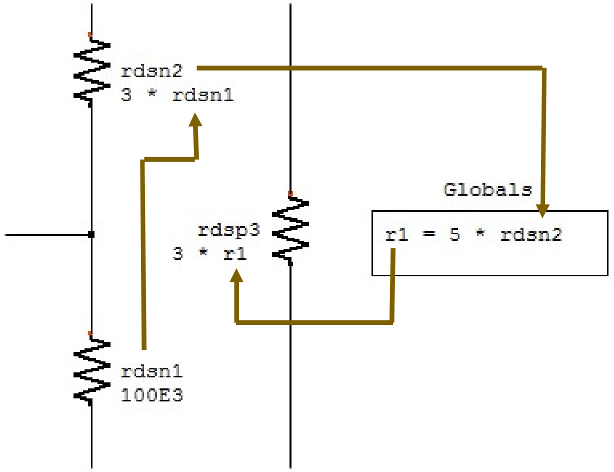
```
L0 = 1E-4
```

the function **XFLOAT(expr)** can be used to replace **all** F%nn with their numerical values

the function **XNUM(expr)** can be used to replace **all** Globals with their numerical values

NOTE : any *symbolic constant* value of the form [1234] are not allowed except [1] [+1] [-1] that are reserved for E , H , G , F components

- **instnames** and **GLOBALS** variables can both be used in equations

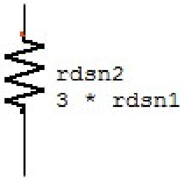


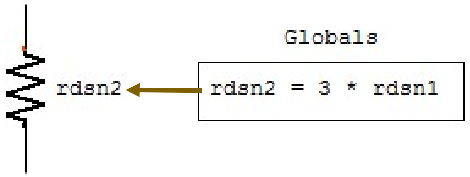
- **instnames** are by definition « **declared** » **and** « **used** »

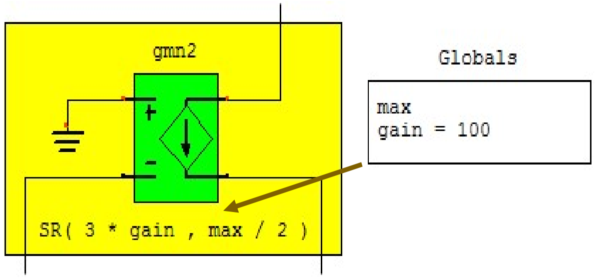
- **GLOBALS** variables must be also « **declared** » **and** « **used** » (in **instvalues** or **GLOBALS**)

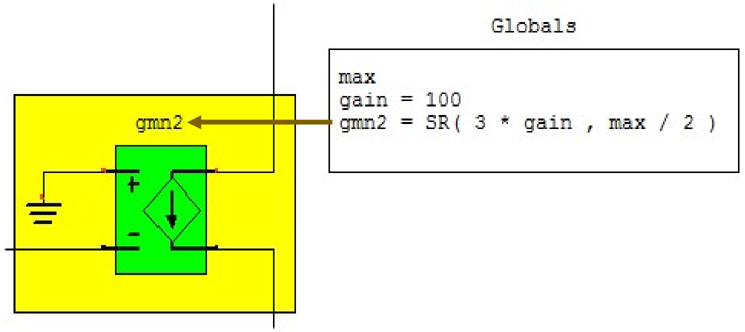
R1 **R1 = « load »** **R1 = 100** **R2 = 2 * R1**

- equations can be defined either in **instvalues** or **GLOBALS**

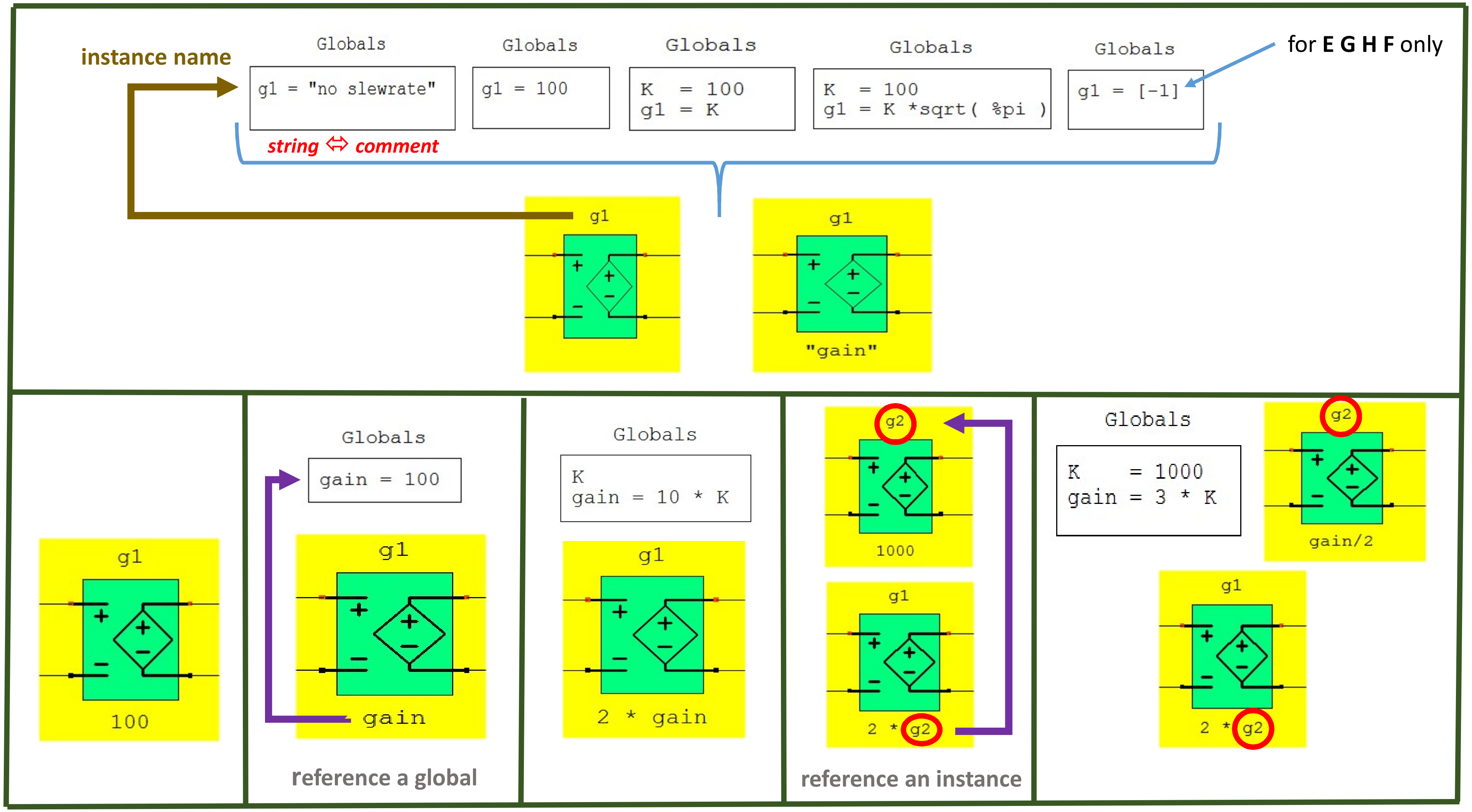








« xxx » **ignored string**	xxxx **symbolic** expressions substituted at **build time**	33.5 or 3E-8 **numeric** substituted at **run time**



instance name

Globals
```
g1 = "no slewrate"
```
string ⇔ comment

Globals
```
g1 = 100
```

Globals
```
K  = 100
g1 = K
```

Globals
```
K  = 100
g1 = K *sqrt( %pi )
```

Globals
```
g1 = [-1]
```
for **E G H F** only

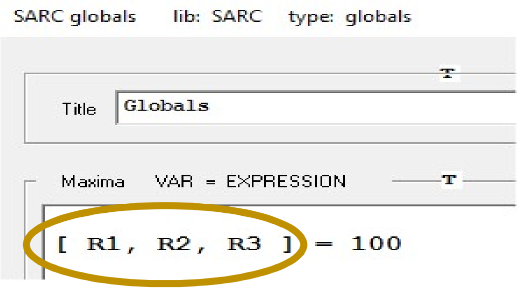
Globals
```
gain = 100
```
reference a global

Globals
```
K
gain = 10 * K
```

reference an instance

Globals
```
K    = 1000
gain = 3 * K
```

declarations can be grouped



SARC globals lib: SARC type: globals

Title: Globals

Maxima VAR = EXPRESSION

```
[ R1, R2, R3 ] = 100
```

runtime directives can also be defined with **GLOBALS**

%rubmode = true	do not plot ***instantaneous*** **percussion spikes** (*simulation remains unchanged*)	(*default* **false**)
%printpoles = 3	report the computations of the poles **due to the first 3 BOM** changes	(*default* **0**)
%srtol = 1E-3	***saturation rendering*** computation ***tolerance*** *roughly, dividing* **%strol** *by 10 increases the number of iterations by +4*	(*default* **1E-4**)
%srlim = 25	***saturation rendering*** computation ***iteration limit***	(*default* **50**)
%phases = [Phi1 , Phi2]	**mandatory** for ***Z transform*** but can be declared for **XFER** / **MIMO**	(*no default*)
%zopen = Gds **%zopen = [Gds1 , Gds2]**	*component* ***'Gds'*** *will be* ***ignored*** *by* ***GENERATE_Z( )***	
%zshort = Rds **%zshort = [Rds1 , Rds2]**	*component* ***'Rds'*** *will be* ***shorted*** *by* ***GENERATE_Z( )***	

In electronic, ***slew rate*** is defined as the *maximum change of* ***current*** *or* ***voltage*** *per unit of time* (***max(di / dt)*** or ***max(dv / dt)***)

but, *rather than this* ***« slope control »*** *,* SARC ***slew rate*** simulates the ***OP amp*** ***« clipping »*** effect on ***current*** sources

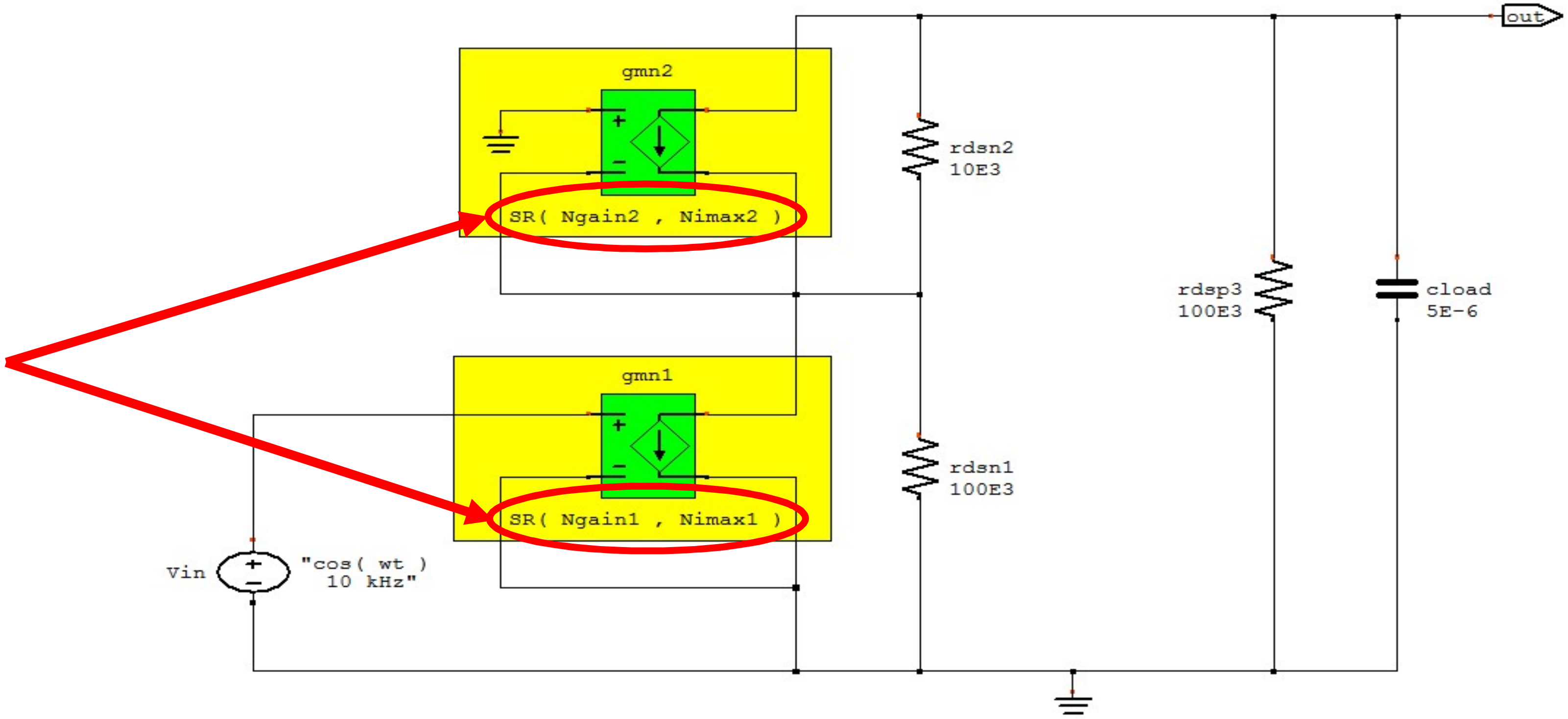
In other words, for **G** and **F** `SR( gain , imax )` ⇔ clipping **current** = **imax**

gmn1 **and** **gmn2**

will both simulate

Saturation Rendering

« OP amp SLEWRATE »



Globals

```
%srtol = 1E-3
%srlim = 25
```

targetted tolerance = **fabs(** **(** maxOUTPUT – currentOUTPUT **)** / maxOUTPUT **)**

max iterations

Norton - Thévenin previous slide

by duality,

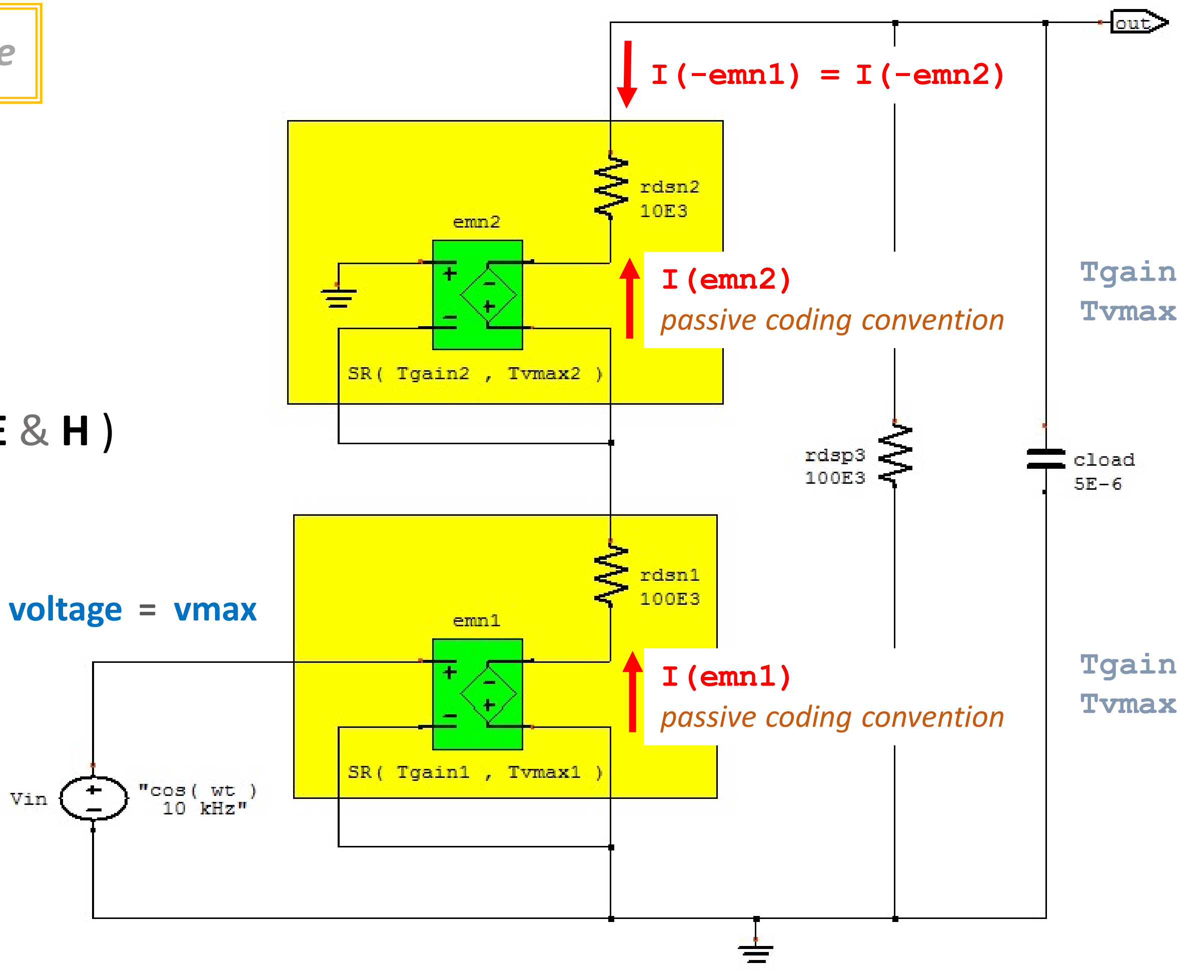
a similar « saturation effect »

can also be defined

for ***controlled voltage source*** (**E** & **H**)

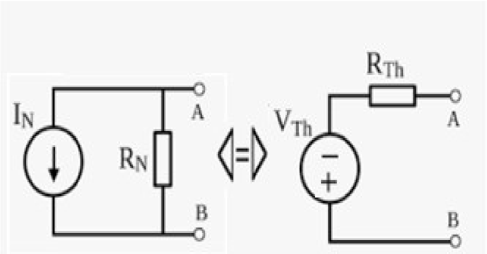
In other words, for E and H

SR(gain , vmax) ⟺ clipping voltage = vmax



Tgain2 = rdsn2 * Ngain2
Tvmax2 = rdsn2 * Nimax2

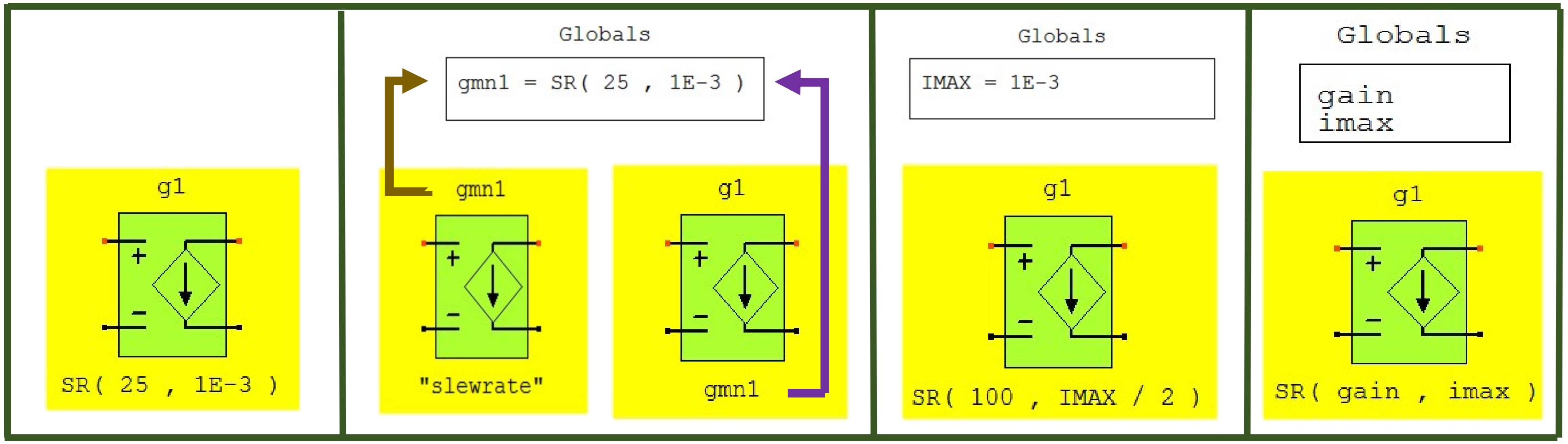
Tgain1 = rdsn1 * Ngain1
Tvmax1 = rdsn1 * Nimax1

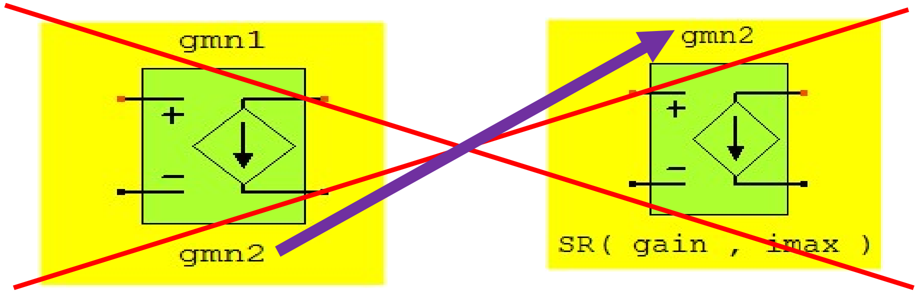


Norton *Thévenin*

SR(gain , maxout) syntax

maxout	Imax for G and F	maxout = 0 ➔ linear mode
	Vmax for E and H	maxout sign ignored





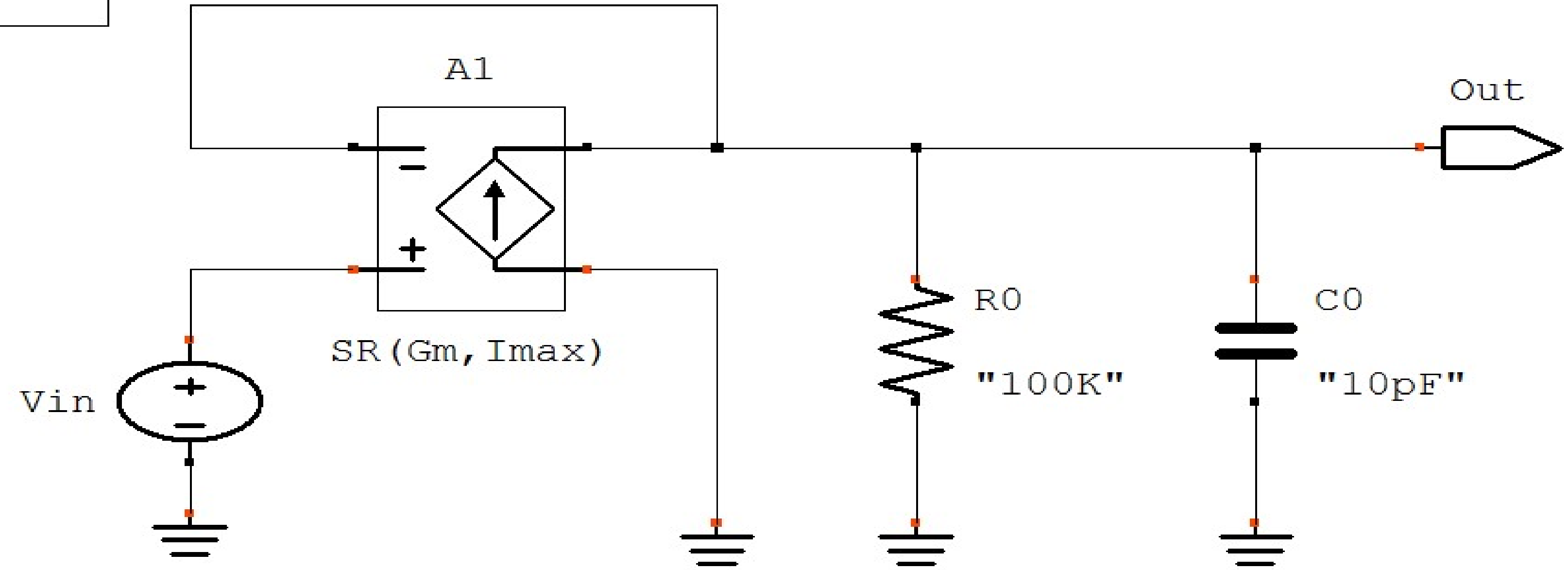
gmn1 gain cannot depend on
SR( ) of gmn2

There is no need to have several schematics : GENERATE_XFER and GENERATE_Z ignore *SR( )* declarations
GENERATE_XFER and GENERATE_MIMO ignore *%phases* declarations
MIMO *SR( )* simulations can be performed LINEAR by setting `maxout` = 0

LP_SK Follower (SR Modeling)

Globals

```
Gm   = "50uA/V"
Imax = "7uA"
```



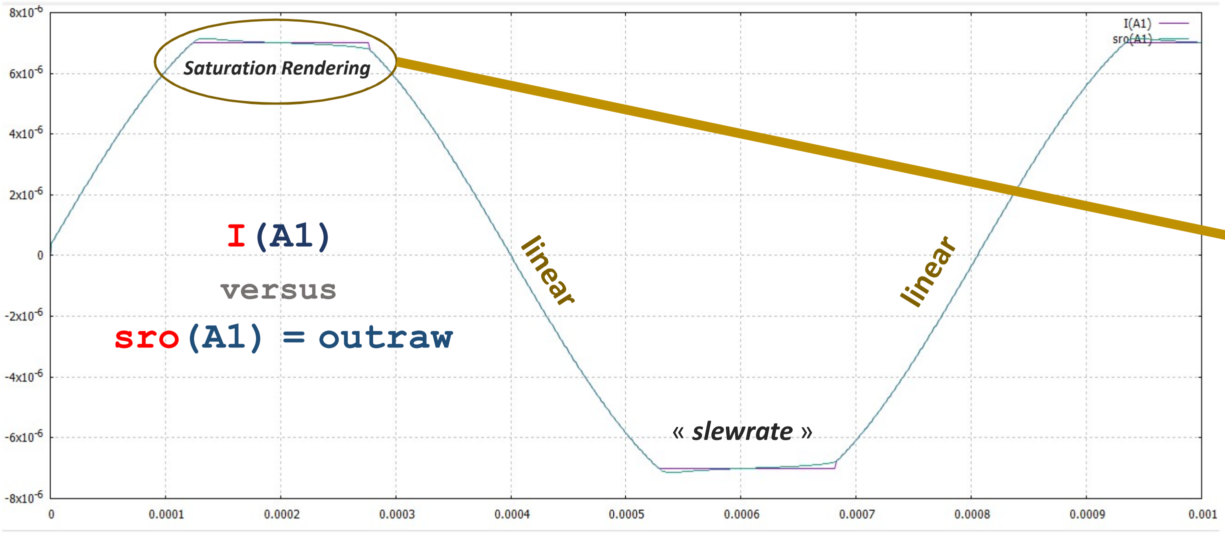
3 special outputs can be plotted to observe the Saturation Rendering convergence

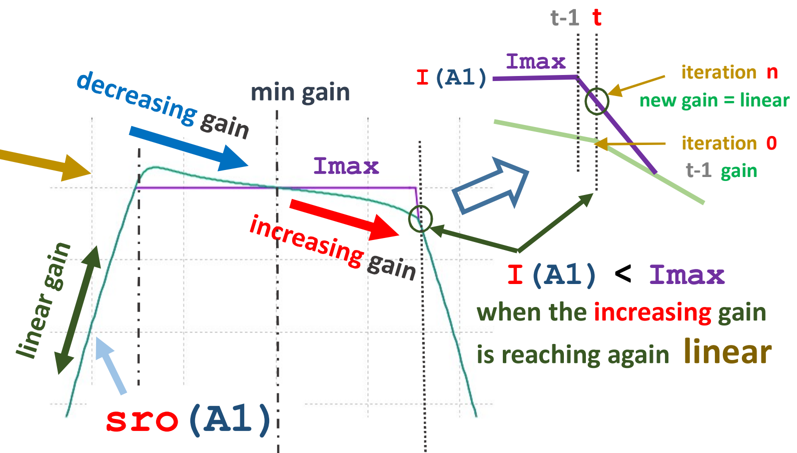
srn(A1) number of recalculations

sro(A1) raw output before relaxation

srg(A1) effective reduced gain necessary to sustain max output

CONVERGENCE



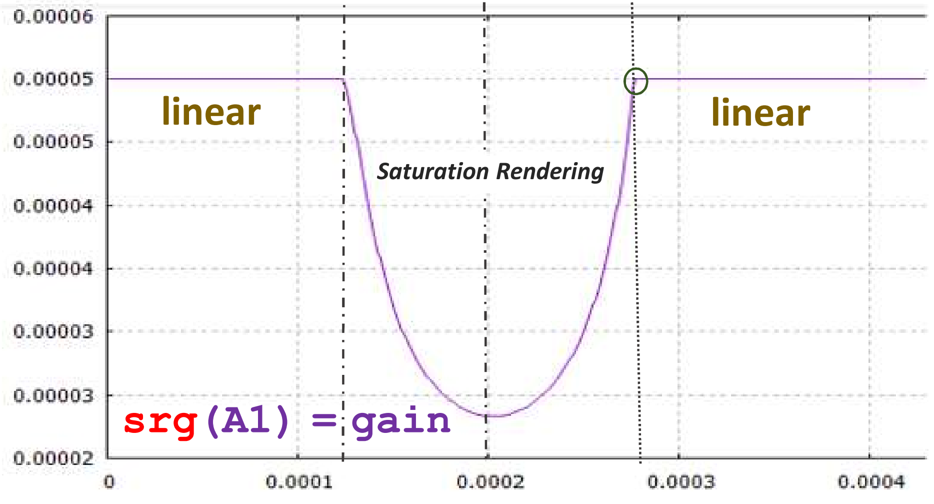


SARC is dedicated to ***Linear Time Invariant*** analog circuits

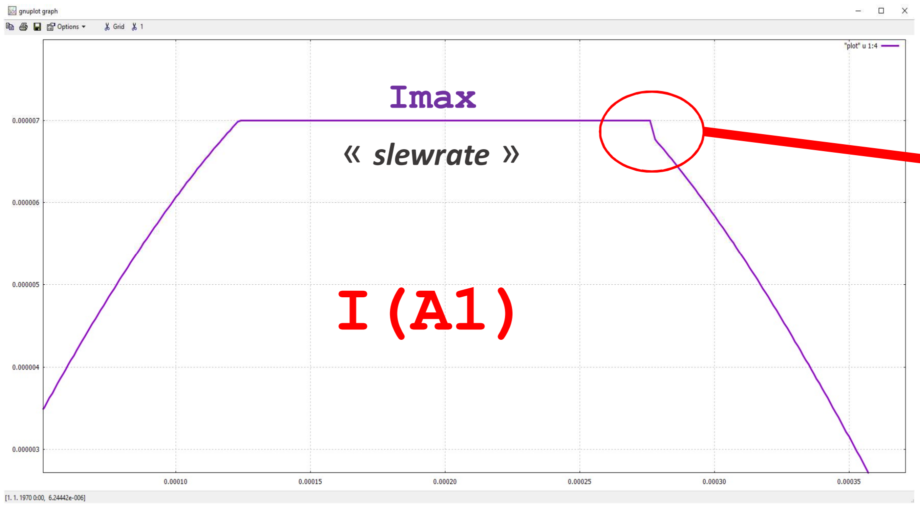
However,

SARC supports « ***Saturation Rendering*** » that is a **nonlinear behavior**

This can be only simulated through a ***convergence process***



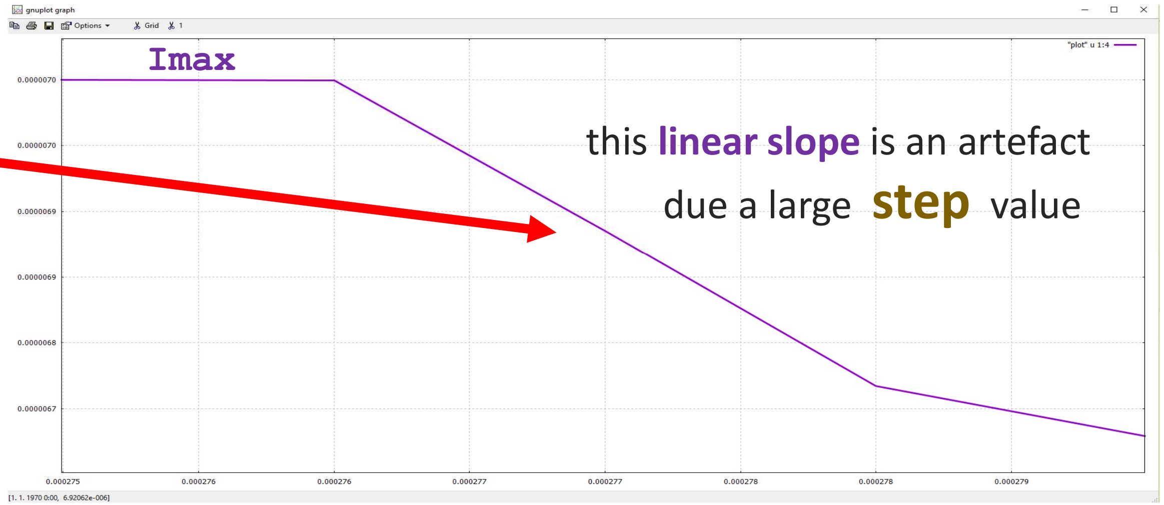
effective reduced gain necessary to sustain max output



Imax

« *slewrate* »

I(A1)

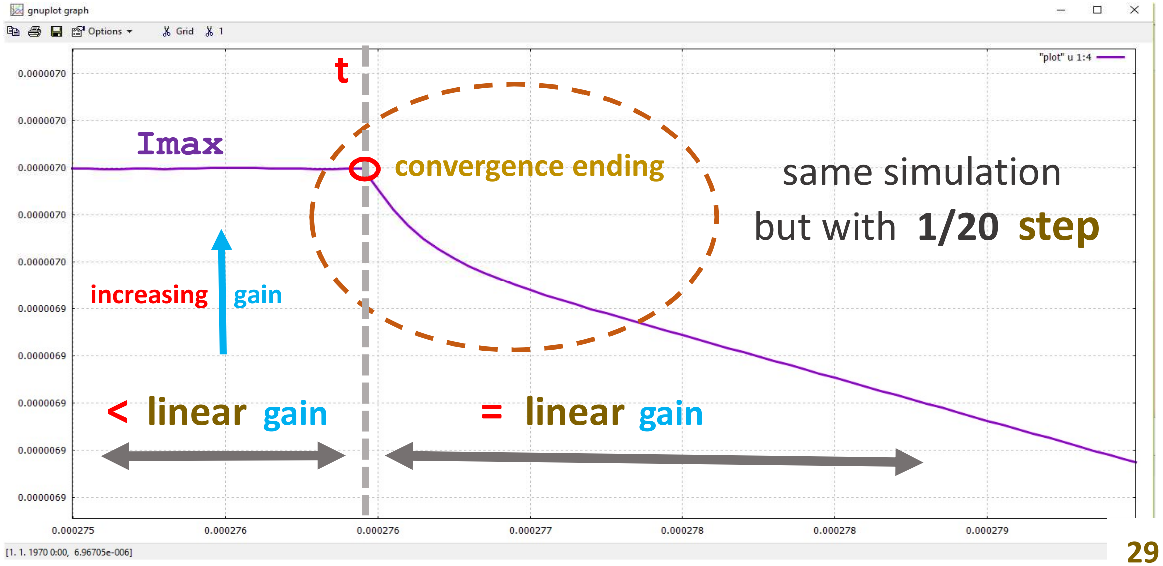


Imax

this **linear slope** is an artefact

due a large **step** value

CONVERGENCE ENDING



t

Imax

convergence ending

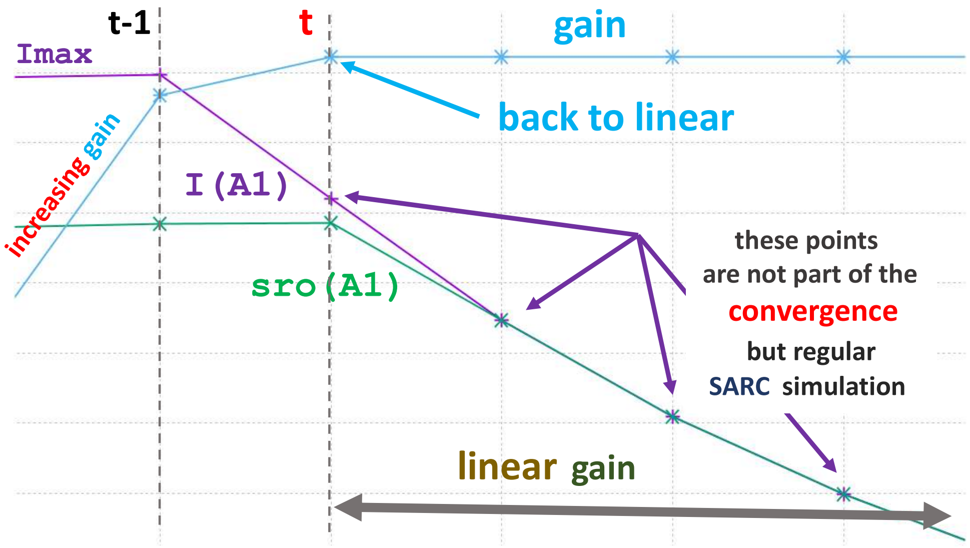
same simulation

but with **1/20 step**

increasing gain

< **linear gain**

= **linear gain**



t-1

t

Imax

gain

back to linear

increasing gain

I(A1)

sro(A1)

these points

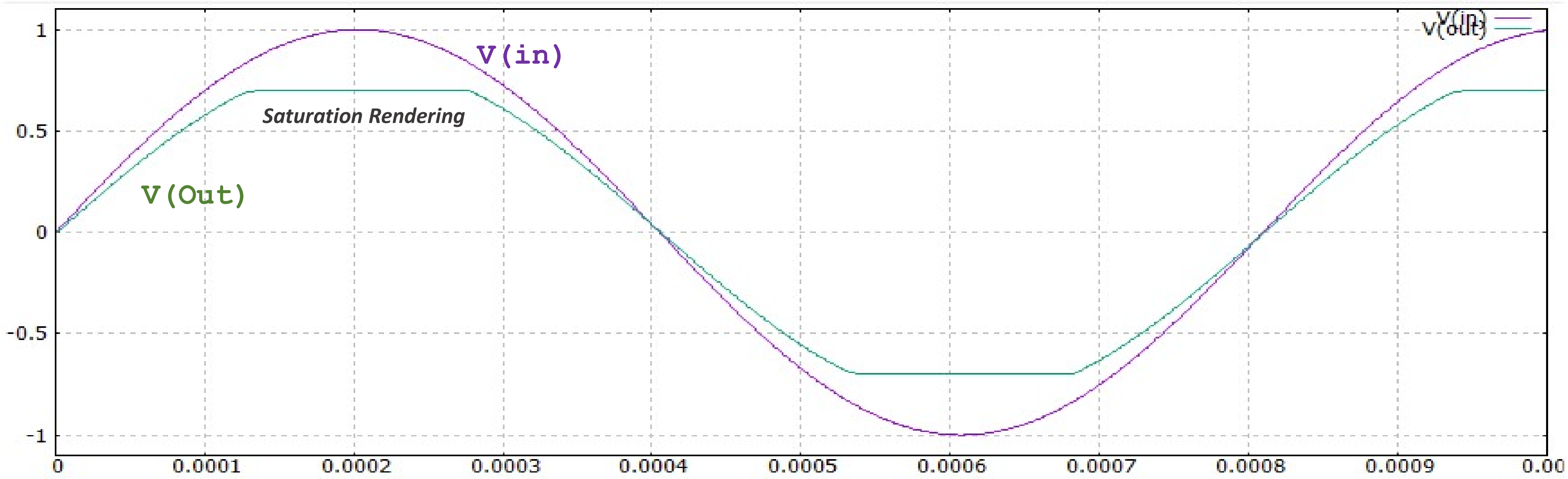
are not part of the

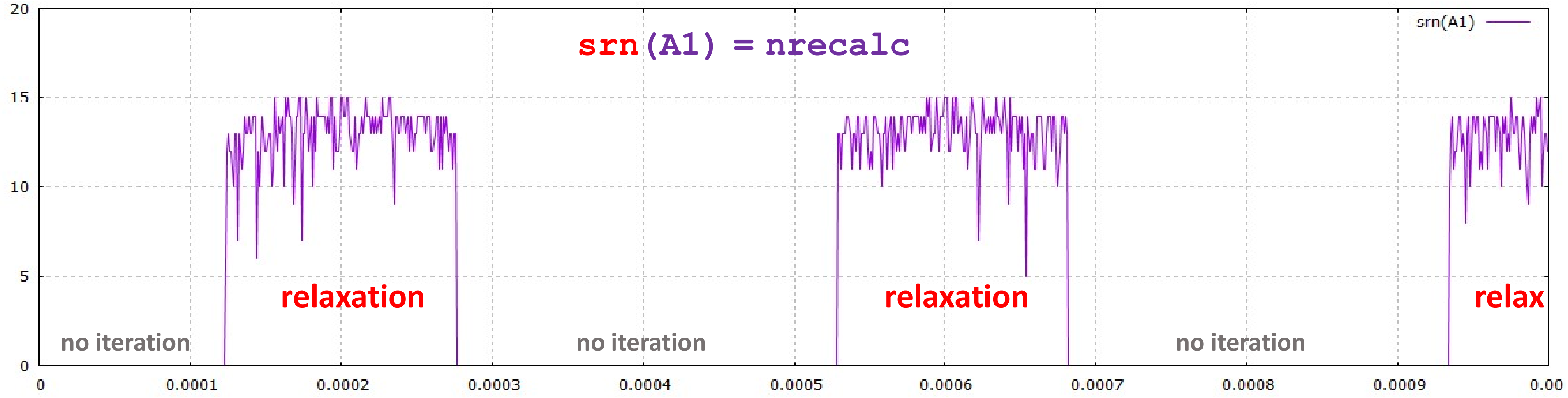
convergence

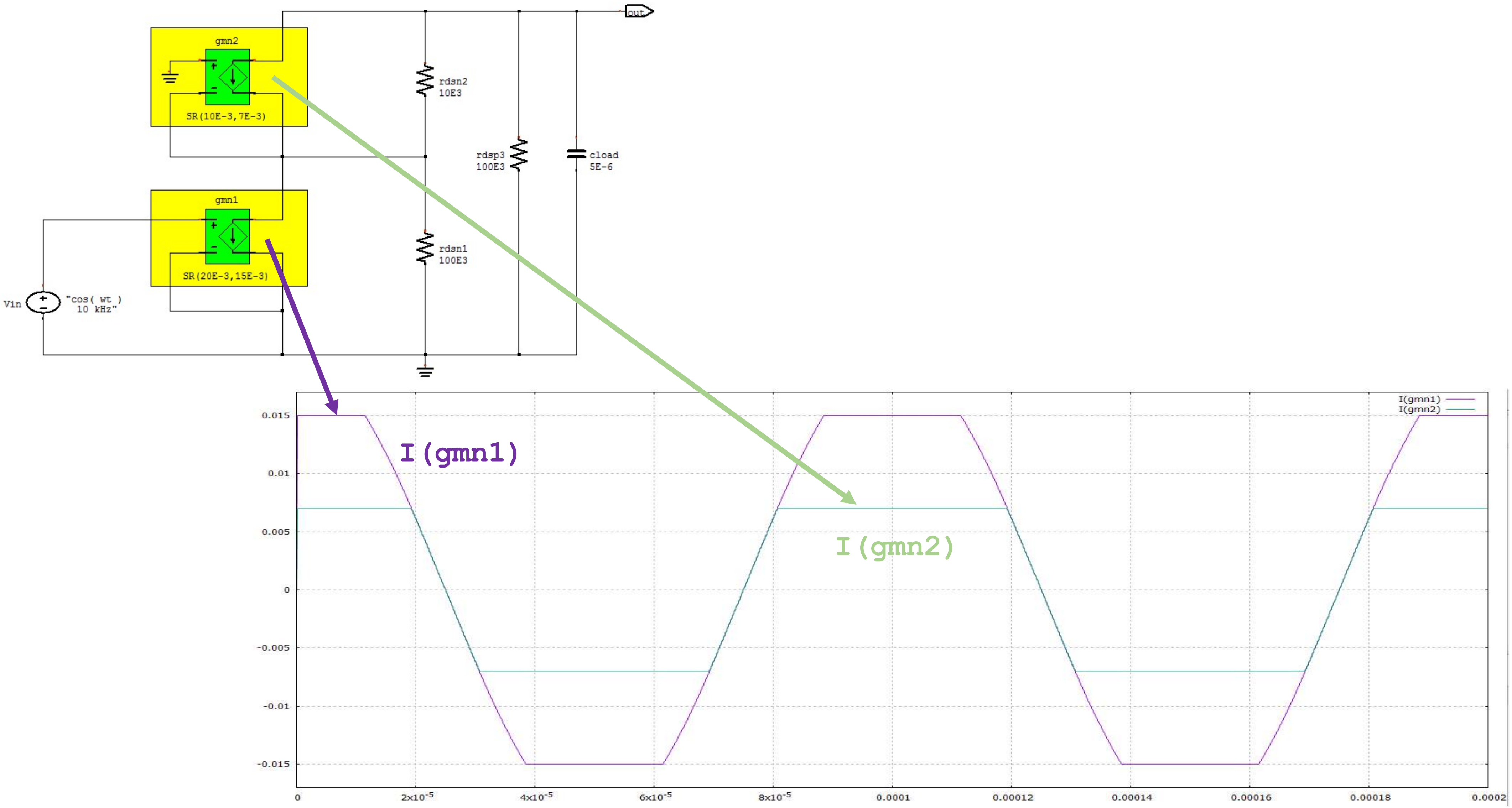
but regular

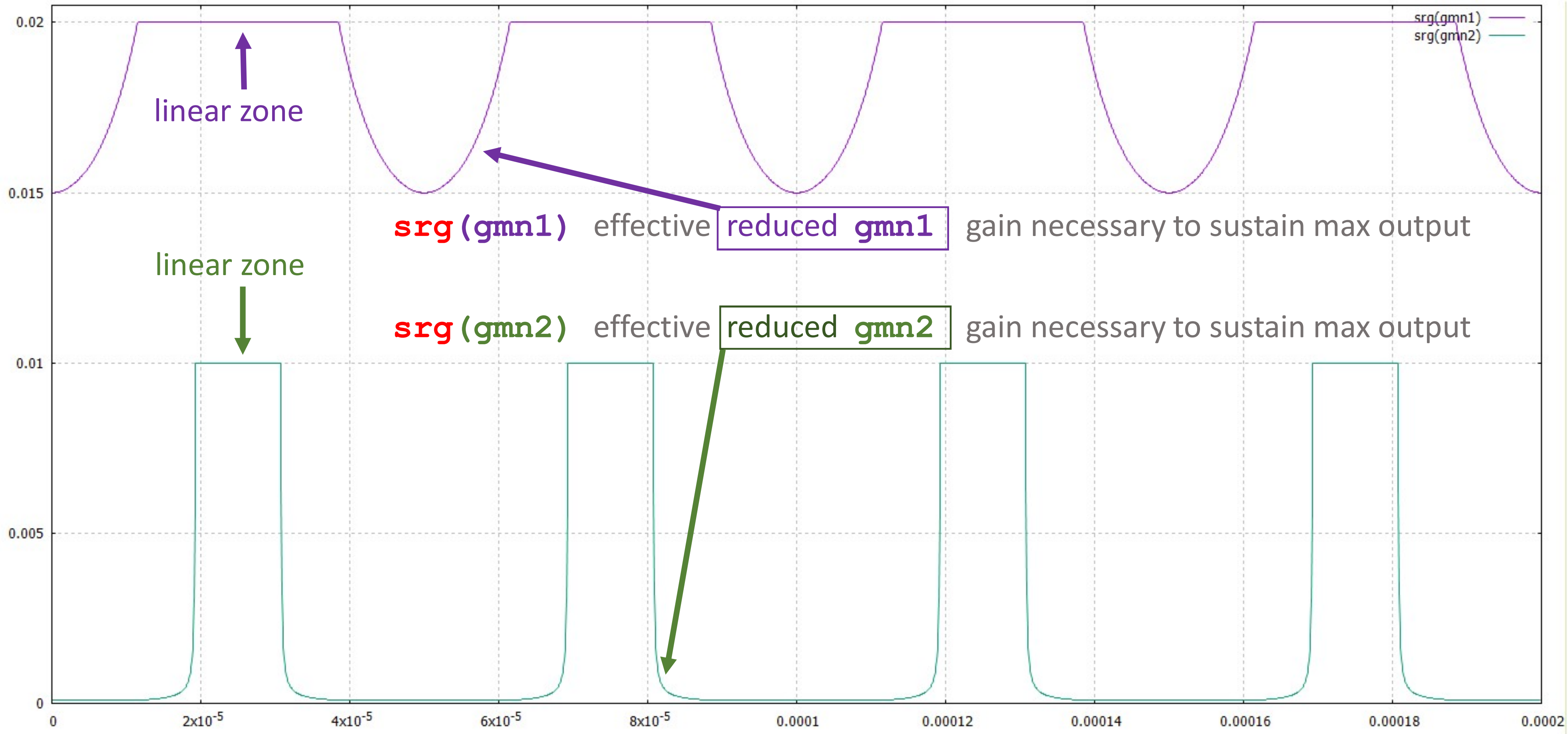
SARC simulation

linear gain


V(in)
V(out)
V(in)
V(Out)
Saturation Rendering
1
0.5
0
-0.5
-1
0
0.0001
0.0002
0.0003
0.0004
0.0005
0.0006
0.0007
0.0008
0.0009
0.00


srn(A1)
srn(A1) = nrecalc
relaxation
relaxation
relax
no iteration
no iteration
no iteration
20
15
10
5
0
0
0.0001
0.0002
0.0003
0.0004
0.0005
0.0006
0.0007
0.0008
0.0009
0.00


out
gmn2
SR(10E-3,7E-3)
gmn1
SR(20E-3,15E-3)
rdsn2
10E3
rdsn1
100E3
rdsp3
100E3
cload
5E-6
Vin
"cos(wt)
10 kHz"
I(gmn1)
I(gmn2)
0.015
0.01
0.005
0
-0.005
-0.01
-0.015
0
$2x10^{-5}$
$4x10^{-5}$
$6x10^{-5}$
$8x10^{-5}$
0.0001
0.00012
0.00014
0.00016
0.00018
0.0002



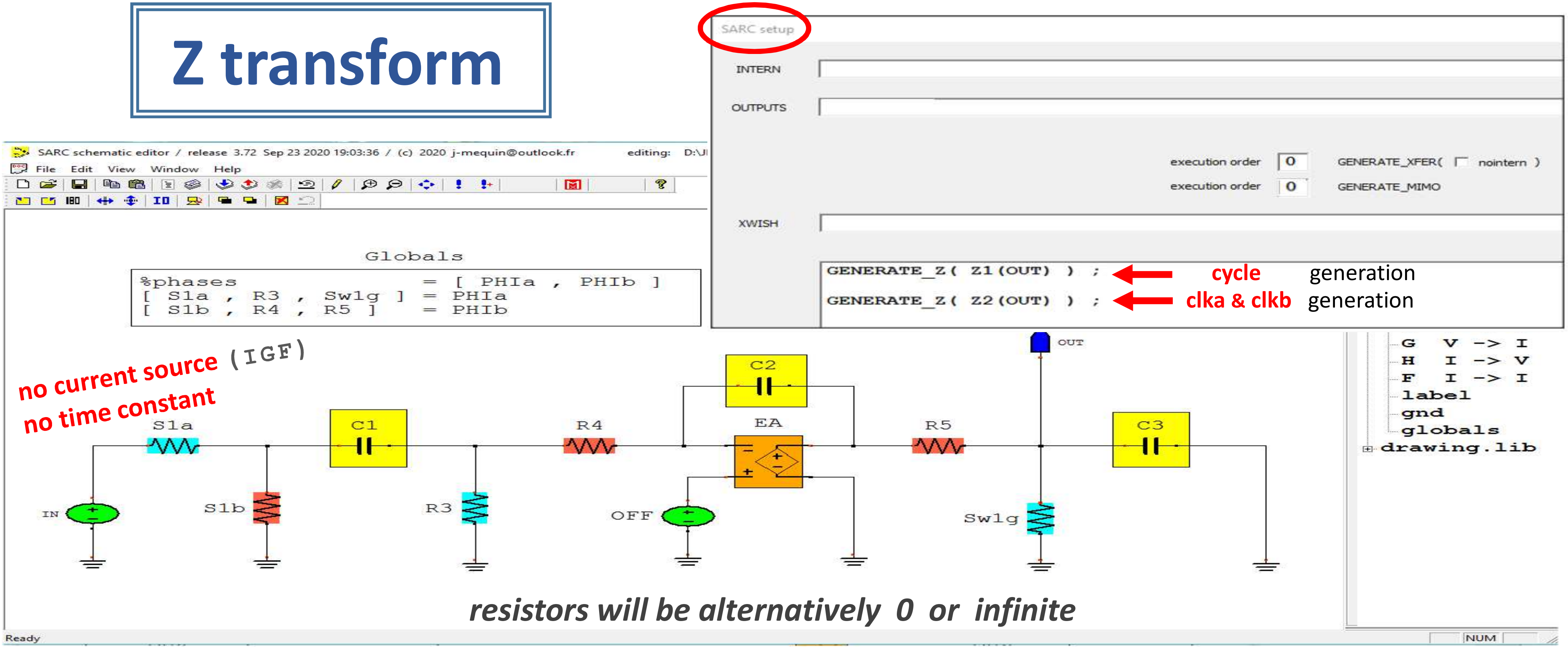
linear zone

srg(gmn1) effective reduced **gmn1** gain necessary to sustain max output

linear zone

srg(gmn2) effective reduced **gmn2** gain necessary to sustain max output

Z transform



SARC schematic editor / release 3.72 Sep 23 2020 19:03:36 / (c) 2020 j-mequin@outlook.fr editing: D:\J

Globals

```
%phases                  = [ PHIa , PHIb ]
[ S1a , R3 , Sw1g ]      = PHIa
[ S1b , R4 , R5 ]        = PHIb
```

SARC setup

INTERN

OUTPUTS

execution order 0 GENERATE_XFER(□ nointern)

execution order 0 GENERATE_MIMO

XWISH

```
GENERATE_Z( Z1(OUT) ) ;
GENERATE_Z( Z2(OUT) ) ;
```

cycle generation

clka & clkb generation

no current source (IGF)

no time constant

resistors will be alternatively 0 or infinite

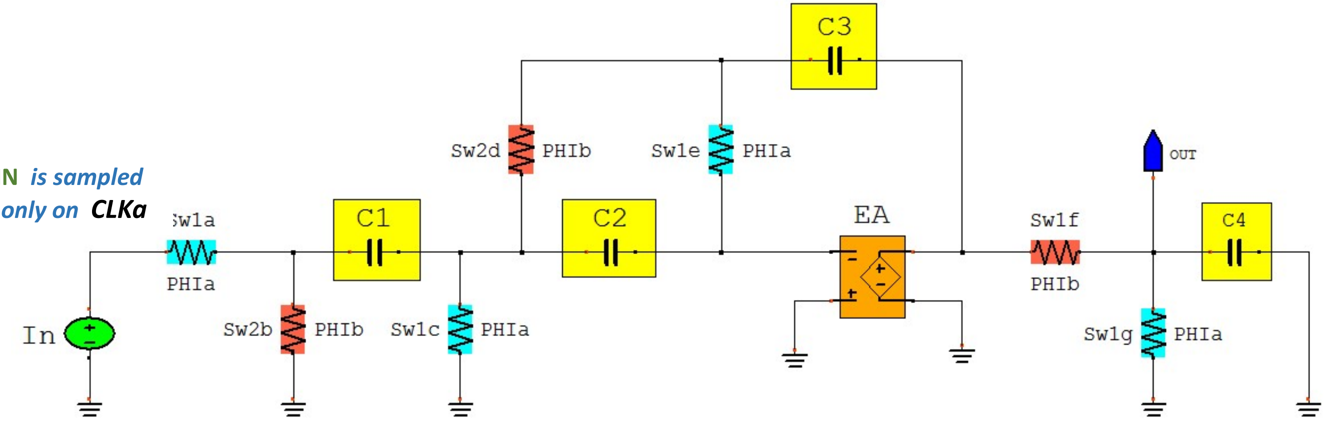
« ***cycle mode*** » means « ***single sampling of the inputs per cycle*** »

Therefore,

it only makes sens to request « ***cycle model generation*** » for ***physical designs*** structurally ***insensitive to intracycle stimuli***

*(**otherwise** this can also be achieved by **externally « latch »** the **inputs within cycle**)*

IN *is sampled only on* ***CLKa***



... time response (when CLKb)

$$OUT_0 = \frac{nbIn_2 \boxed{In_{-1}} + nbIn_0 \boxed{In_{-3}} - db_1 \, OUT_{-2}}{db_3}$$

only odd indexes

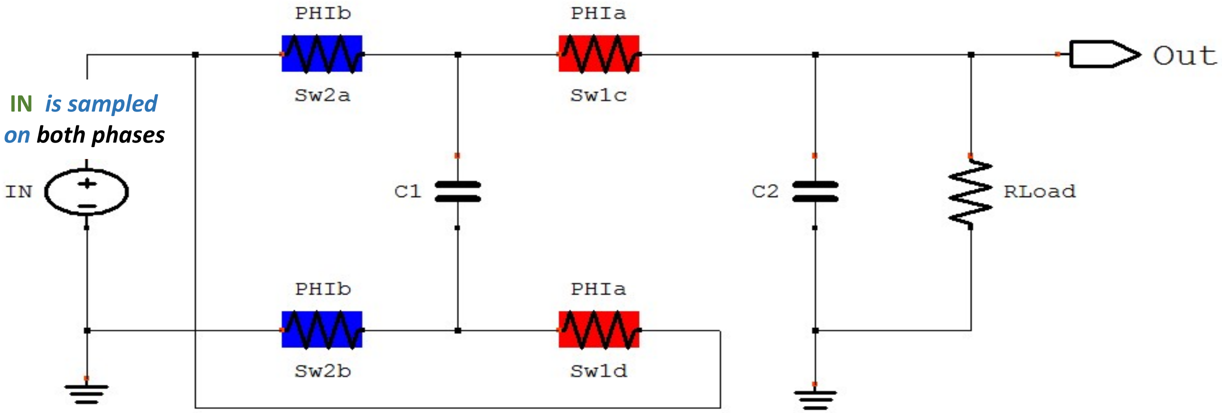
➔ CLKa

... time response (cycle)

$$OUT_0 = \frac{nIn_1 \, In_0 + nIn_0 \, In_{-1} - d_0 \, OUT_{-1}}{d_1}$$

=== NOTE: CYCLE mode will remain exact even in case of intracycle INPUT changes

IN *is sampled on* ***both phases***



... time response (when CLKb)

$$Out_0 = \frac{nbIN_1 \, IN_{-1} + nbIN_0 \boxed{IN_{-2}} - db_0 \, Out_{-2}}{db_2}$$

intracycle

even index

➔ CLKb

... time response (cycle)

$$\xcancel{Out_0 = \frac{nIN_1 \, IN_0 + nIN_0 \, IN_{-1} - d_0 \, Out_{-1}}{d_1}}$$

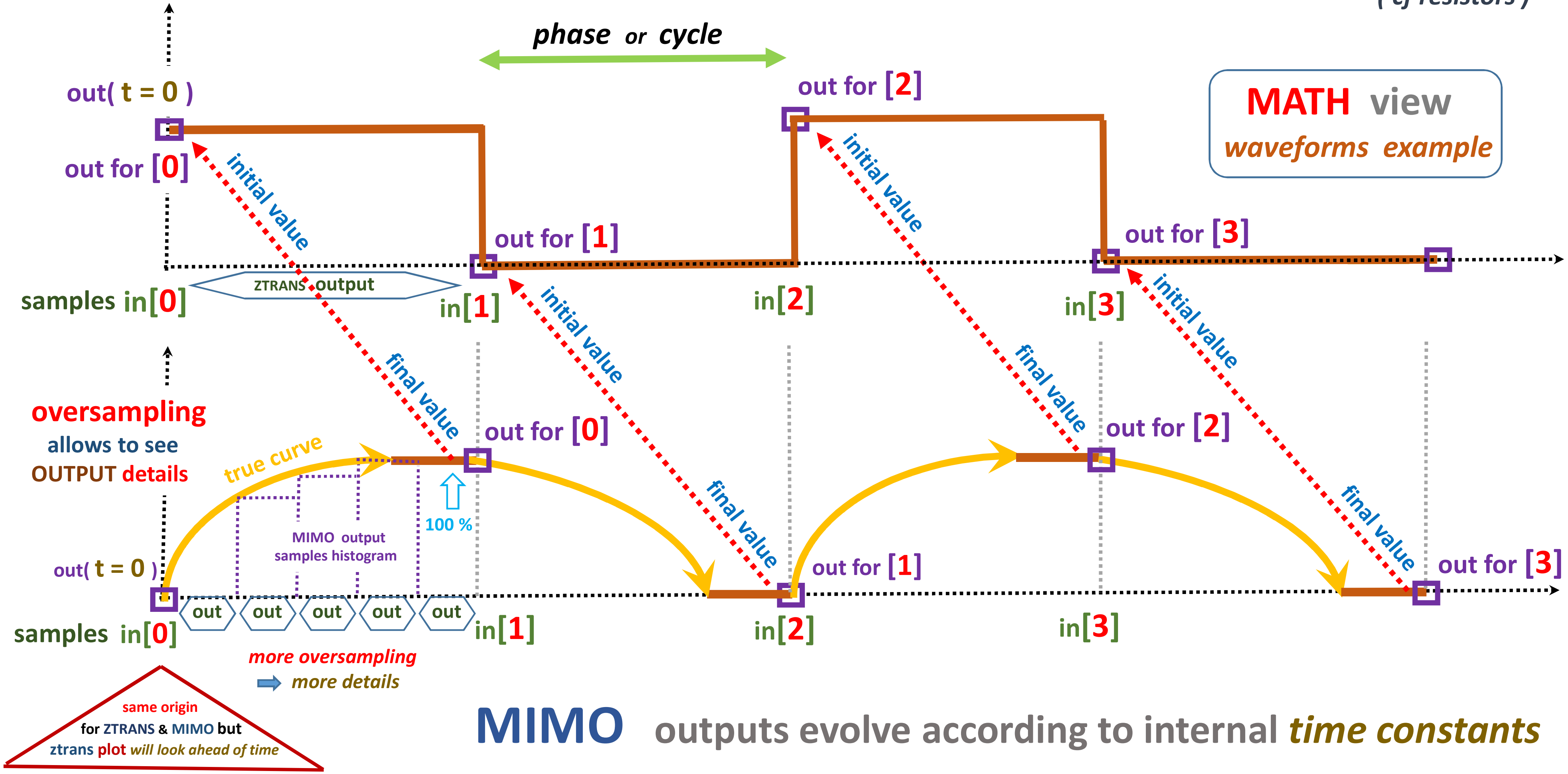
=== NOTE: CYCLE mode will not remain exact in case of intracycle INPUT changes due to CLKb coefficients $= \left[IN_{-2} \right]$

ZTRANS outputs evolve like MIMO would do if the time constants were removed

(cf resistors)

MATH view

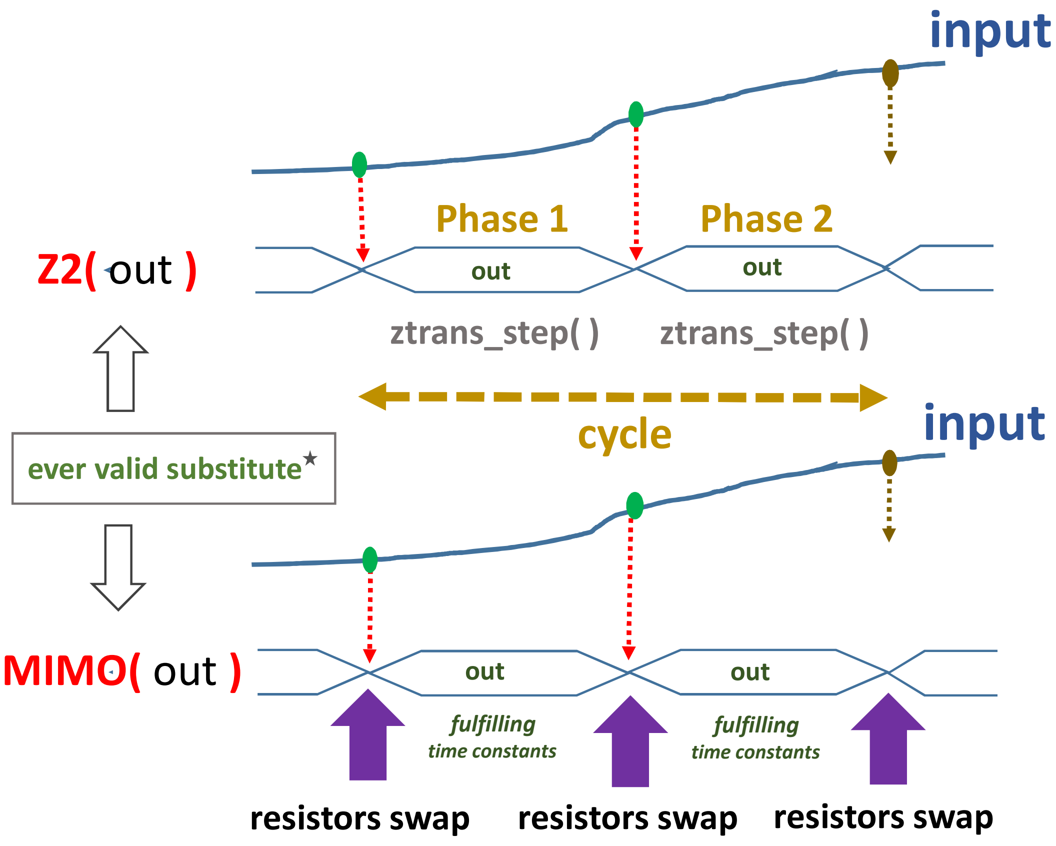
waveforms example



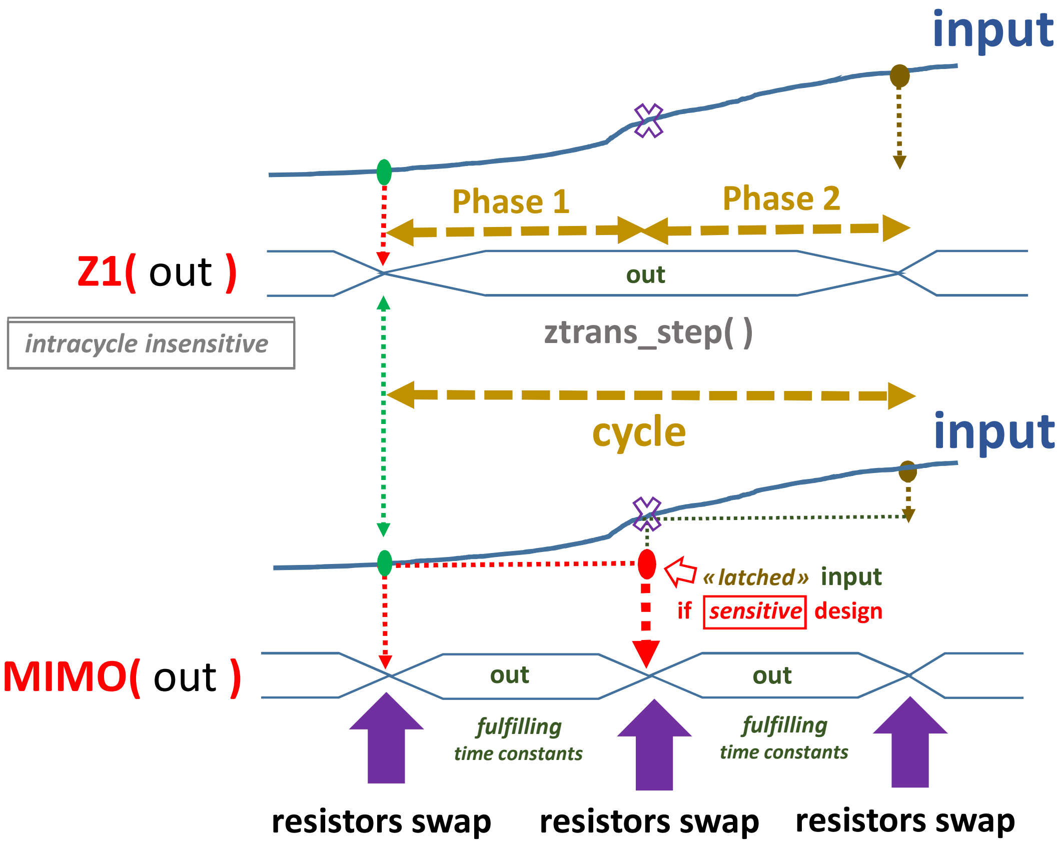
MIMO outputs evolve according to internal *time constants*

ZTRANS / MIMO COMPARISON (*basic case :* cycle = 2 steps)

with MIMO PHASES



with MIMO CYCLE



★ *but possibly not identical due the MIMO analog time constants (and/or slewrate)*

Z2 / MIMO COMPARISON (MIMO *oversampling*)

with MIMO PHASES

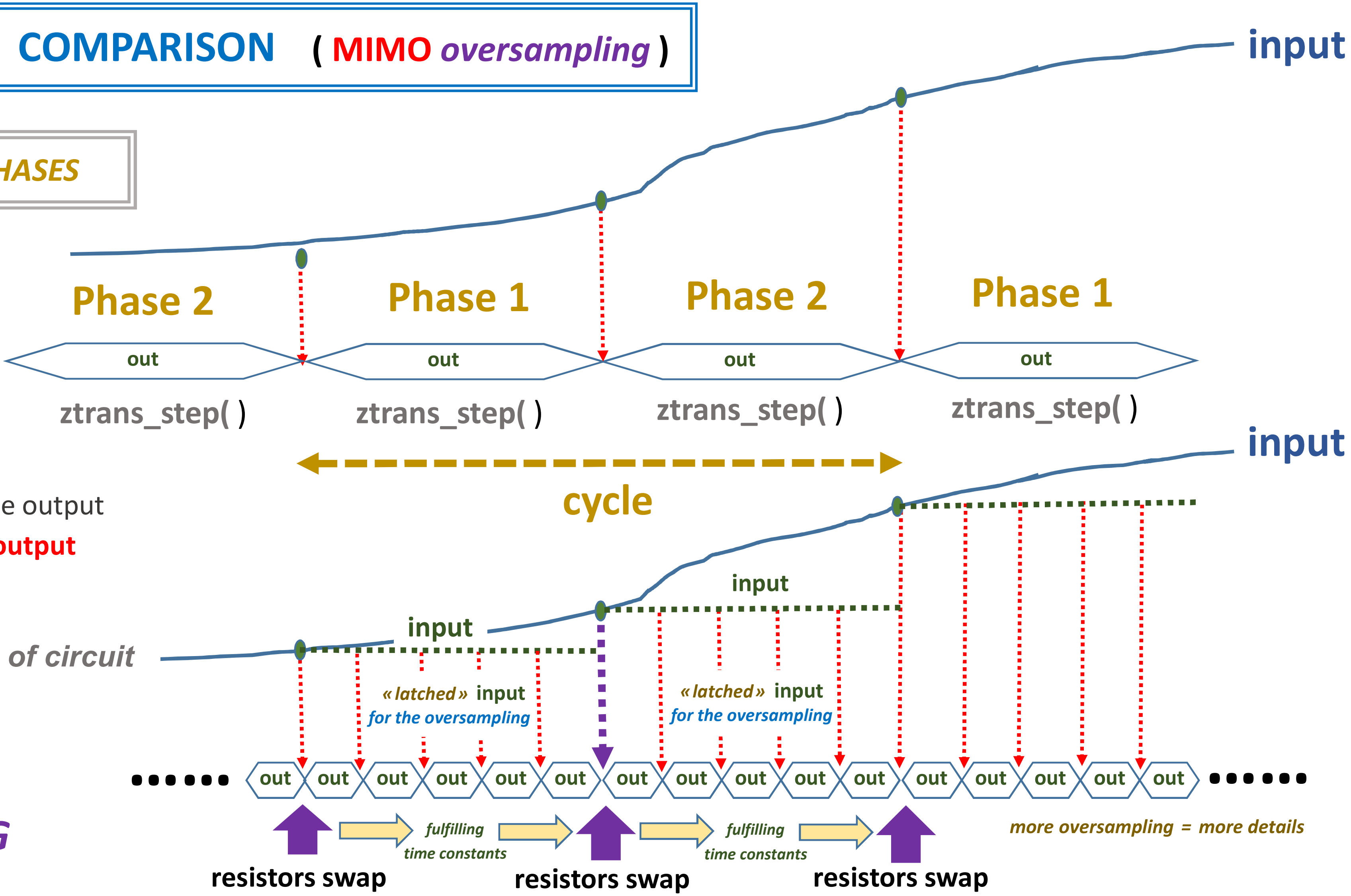
Z2(out)

MIMO detailed curved line output
modelized by **Z squared output**

***« INTEGRATOR »** type of circuit*

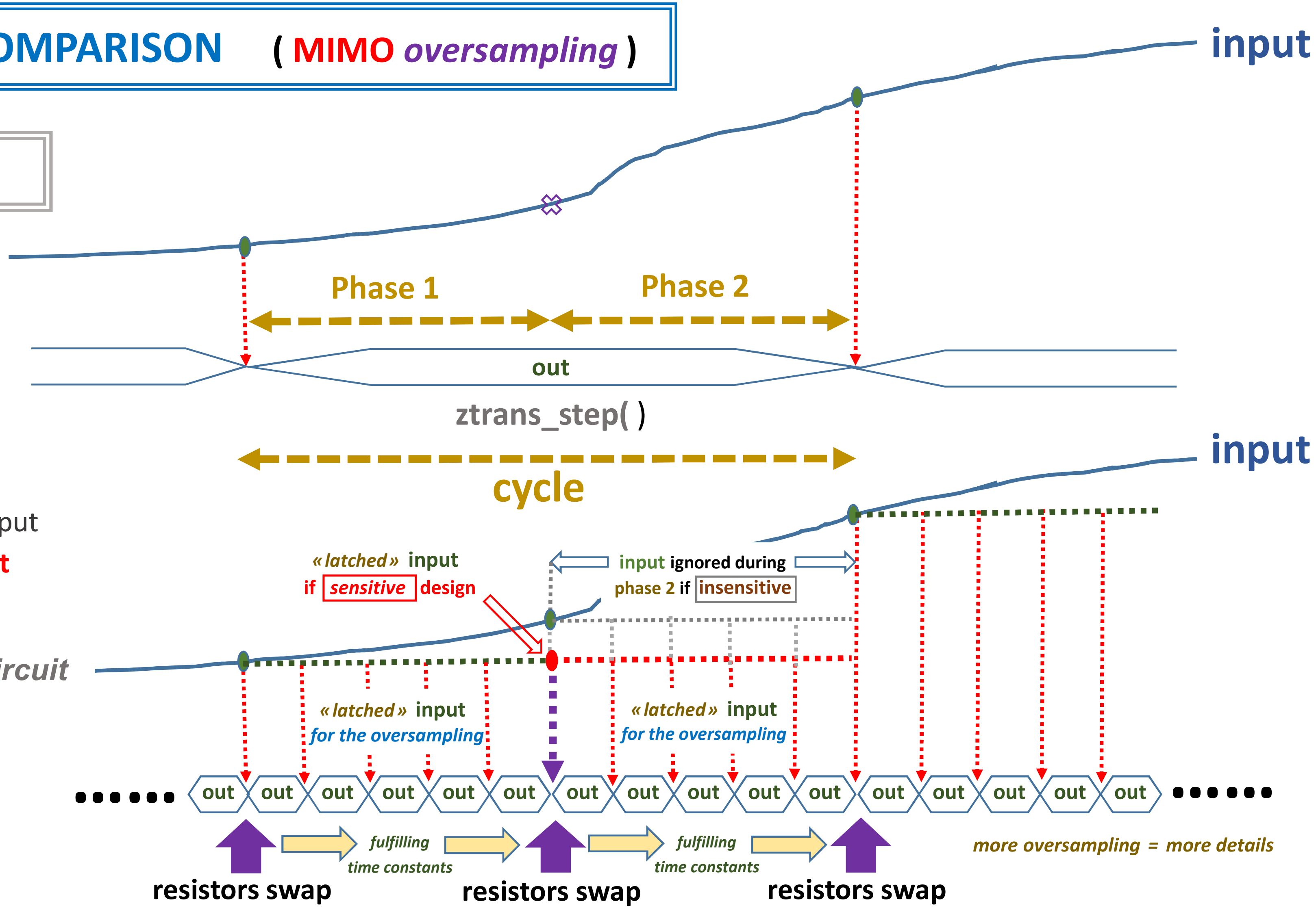
MIMO(out)

OVERSAMPLING



Z1 / MIMO COMPARISON (MIMO *oversampling*)

with MIMO CYCLE



Z1(out)

intracycle insensitive

MIMO detailed curved line output modelized by **Z squared output**

« INTEGRATOR » ***type of circuit***

MIMO(out)

OVERSAMPLING

Comparison Z1(Out) versus Z2(Out)

Z1(Out)

... time response (cycle) *one sampling and one output value per cycle*

$$Out_0 = \frac{nOffs_1\ Offs_0 + nVC2_1\ VC2_0 + nOffs_0\ Offs_{-1} + nVC2_0\ VC2_{-1} + nVin_0\ Vin_{-1} - d_0\ Out_{-1}}{d_1}$$

Z2(Out)

... time response (when CLKa) *irrelevant : cycle unique value masks any clka transient output*

$$Out_0 = \frac{naOffs_2\ Offs_0 + naVC2_2\ VC2_0 + naVin_1\ Vin_{-1} + naOffs_0\ Offs_{-2} + naVC2_0\ VC2_{-2} - da_0\ Out_{-2}}{da_2}$$

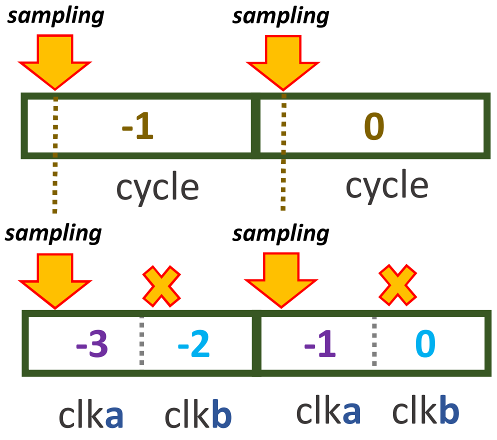
... time response (when CLKb) *equation of clkb output calculates by construct the final [clka clkb] output value*

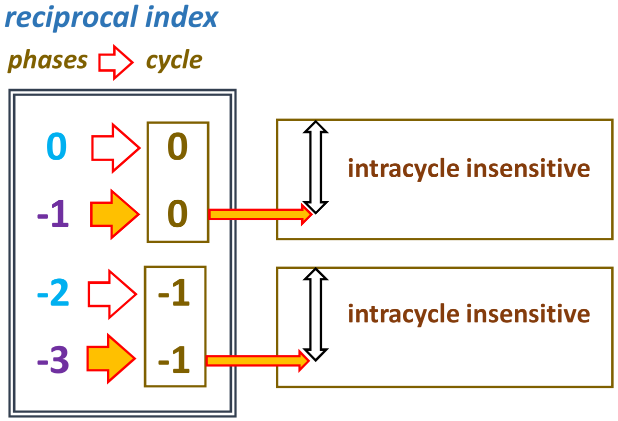
$$Out_0 = \frac{nbOffs_2\ Offs_0 + nbVC2_2\ VC2_0 + nbOffs_1\ Offs_{-1} + nbVC2_1\ VC2_{-1} + nbOffs_0\ Offs_{-2} + nbVC2_0\ VC2_{-2} + nbVin_0\ Vin_{-2} - db_0\ Out_{-2}}{db_2}$$

this expression plotted all along the cycle = Z1(Out)

mode *cycle* index (= t)

mode *phases* index (= t)





Comparison Z1(Out) versus Z2(Out)

Z1(Out)

... time response (cycle) *one sampling and one output value per cycle*

$$Out_0 = \frac{nOffs_1\, Offs_0 + nVC2_1\, VC2_0 + nOffs_0\, Offs_{-1} + nVC2_0\, VC2_{-1} + nVin_0\, Vin_{-1} - d_0\, Out_{-1}}{d_1}$$

Z2(Out)

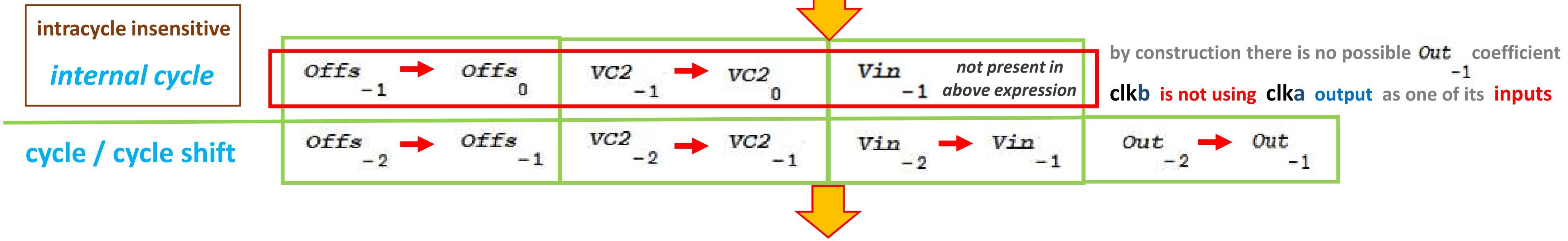
... time response (when CLKa) *irrelevant : **cycle** unique value masks any **clka** transient output*

$$Out_0 = \frac{naOffs_2\, Offs_0 + naVC2_2\, VC2_0 + naVin_1\, Vin_{-1} + naOffs_0\, Offs_{-2} + naVC2_0\, VC2_{-2} - da_0\, Out_{-2}}{da_2}$$

... time response (when CLKb) *equation of **clkb** output calculates by construct the final [**clka clkb**] output value*

$$Out_0 = \frac{nbOffs_2\, Offs_0 + nbVC2_2\, VC2_0 + nbOffs_1\, Offs_{-1} + nbVC2_1\, VC2_{-1} + nbOffs_0\, Offs_{-2} + nbVC2_0\, VC2_{-2} + nbVin_0\, Vin_{-2} - db_0\, Out_{-2}}{db_2}$$

this expression plotted all along the cycle = Z1(Out)



intracycle insensitive

internal cycle

$Offs_{-1}$ → $Offs_0$ | $VC2_{-1}$ → $VC2_0$ | Vin_{-1} *not present in above expression*

by construction there is no possible Out_{-1} coefficient

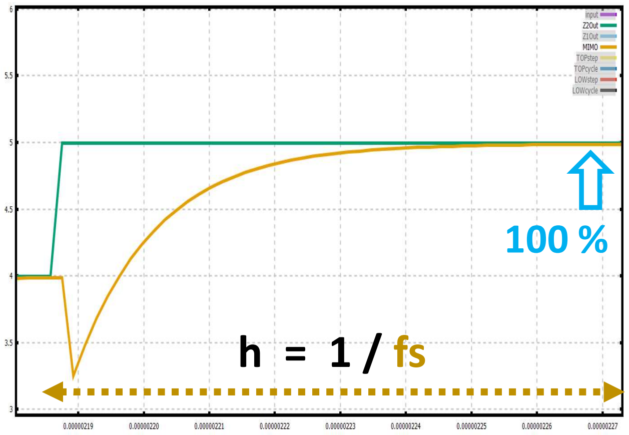
clkb is not using clka output as one of its inputs

cycle / cycle shift

$Offs_{-2}$ → $Offs_{-1}$ | $VC2_{-2}$ → $VC2_{-1}$ | Vin_{-2} → Vin_{-1} | Out_{-2} → Out_{-1}

$$Out_0 = \frac{(nbOffs_2 + nbOffs_1)\, Offs_0 + (nbVC2_2 + nbVC2_1)\, VC2_0 + nbOffs_0\, Offs_{-1} + nbVC2_0\, VC2_{-1} + nbVin_0\, Vin_{-1} - db_0\, Out_{-1}}{db_2}$$

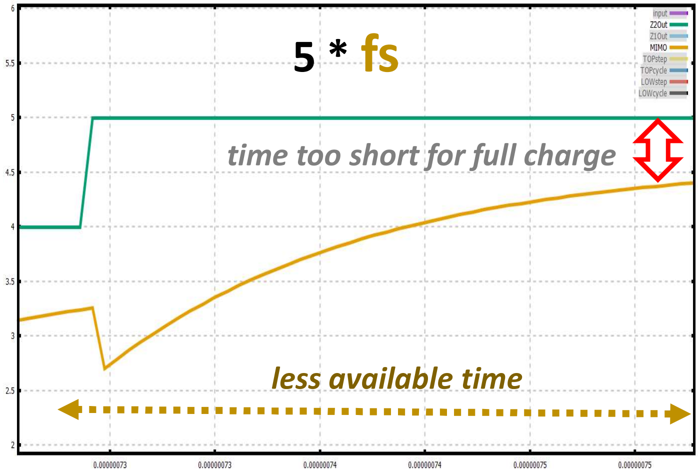
fs ≈ function(*rate of decay*)



charge MIMO < ZTRANS = *100 %*

when fs ↗

charge < 100 % ↙

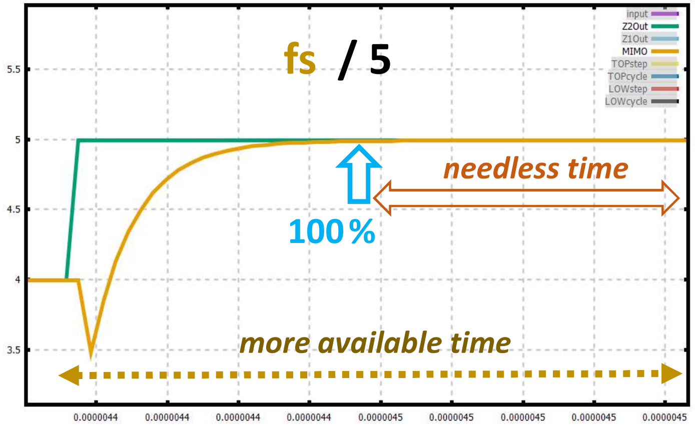


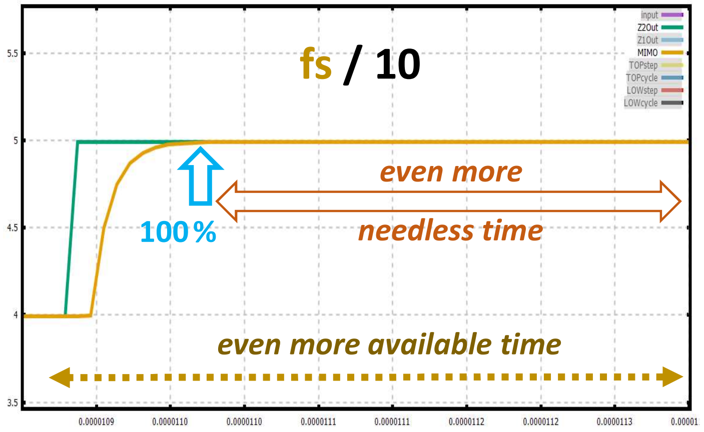
when fs ↙

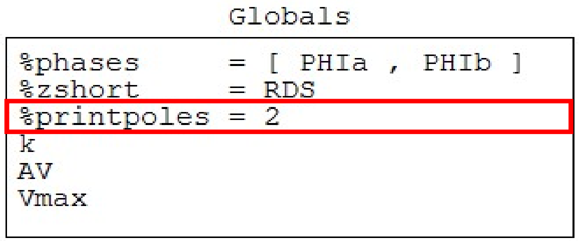
ZTRANS

visual aspect

MIMO

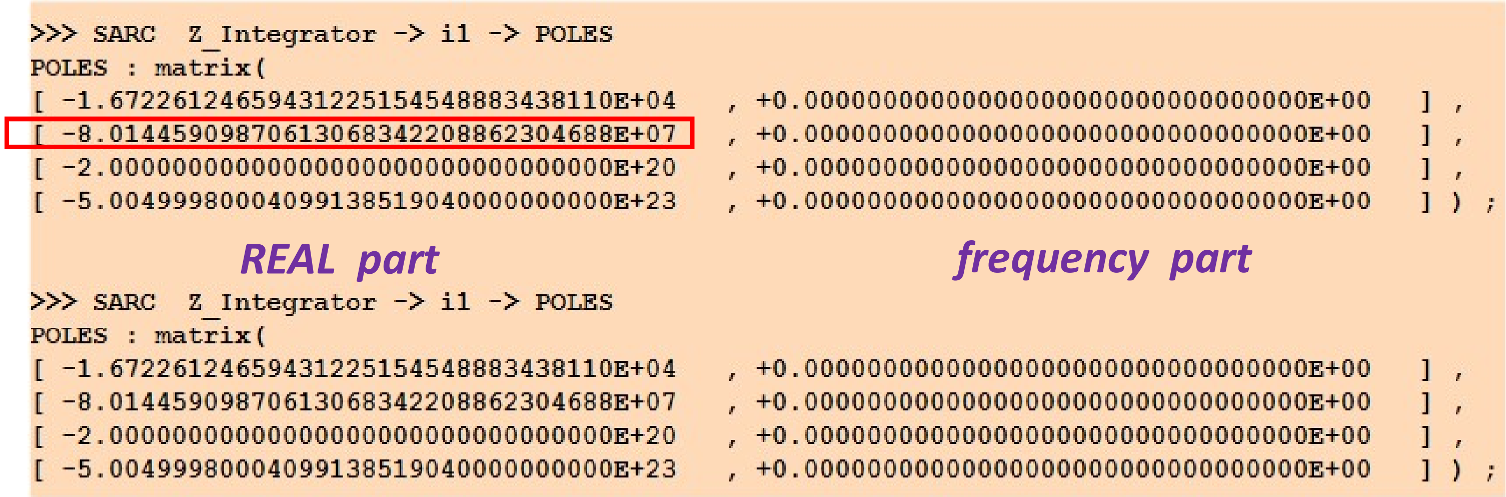






```
Globals

%phases       = [ PHIa , PHIb ]
%zshort       = RDS
%printpoles = 2
k
AV
Vmax
```

rate of decay **a**



```
>>> SARC  Z_Integrator -> i1 -> POLES
POLES : matrix(
[ -1.67226124659431225154548883438110E+04   , +0.00000000000000000000000000000000E+00   ] ,
[ -8.01445909870613068342208862304688E+07   , +0.00000000000000000000000000000000E+00   ] ,
[ -2.00000000000000000000000000000000E+20   , +0.00000000000000000000000000000000E+00   ] ,
[ -5.00499980004099138519040000000000E+23   , +0.00000000000000000000000000000000E+00   ] ) ;
```

REAL part ***frequency part***

```
>>> SARC  Z_Integrator -> i1 -> POLES
POLES : matrix(
[ -1.67226124659431225154548883438110E+04   , +0.00000000000000000000000000000000E+00   ] ,
[ -8.01445909870613068342208862304688E+07   , +0.00000000000000000000000000000000E+00   ] ,
[ -2.00000000000000000000000000000000E+20   , +0.00000000000000000000000000000000E+00   ] ,
[ -5.00499980004099138519040000000000E+23   , +0.00000000000000000000000000000000E+00   ] ) ;
```

$$e^{-at} = e^{-t/\tau}$$

$$\tau = \frac{1}{a} \approx$$ **35 %** ***full charge time***

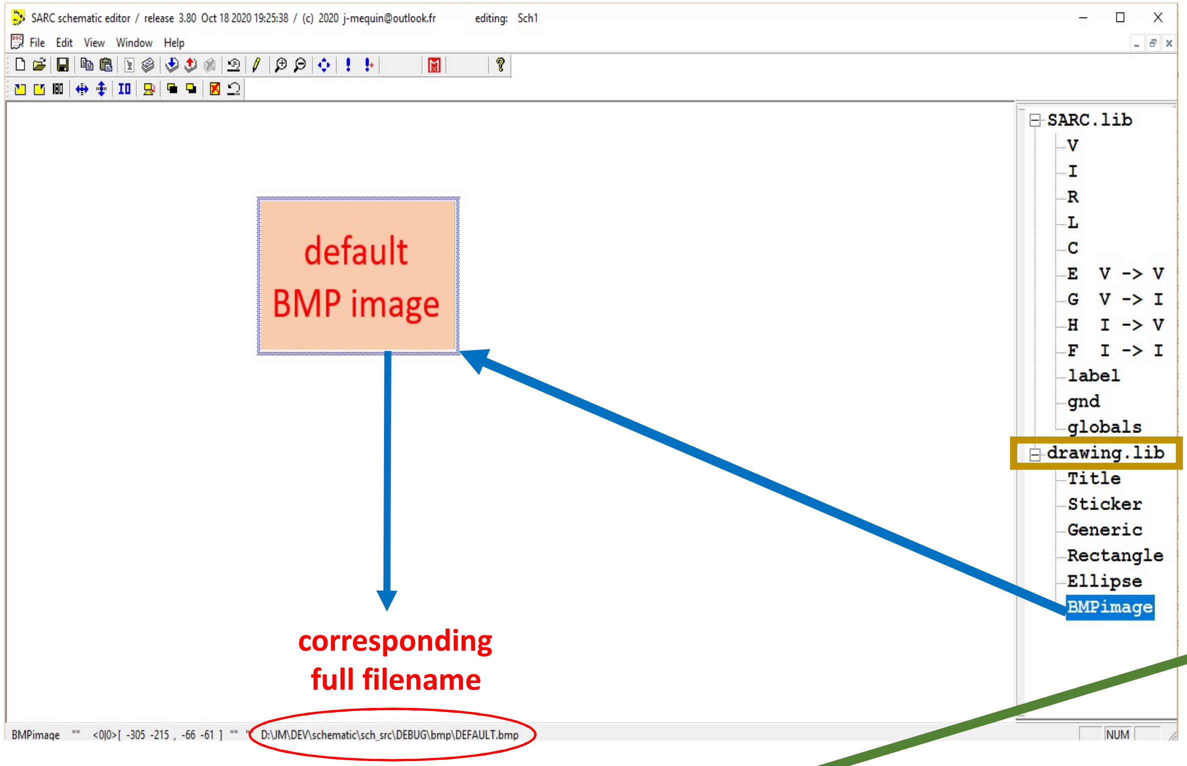
10 τ ➡ $e^{-1/10} \approx$ **90 %**

fs $\approx \frac{a}{10}$ for **90 %**

n = - 1 / ln(%)

5 *is a recommended rule of thumb*

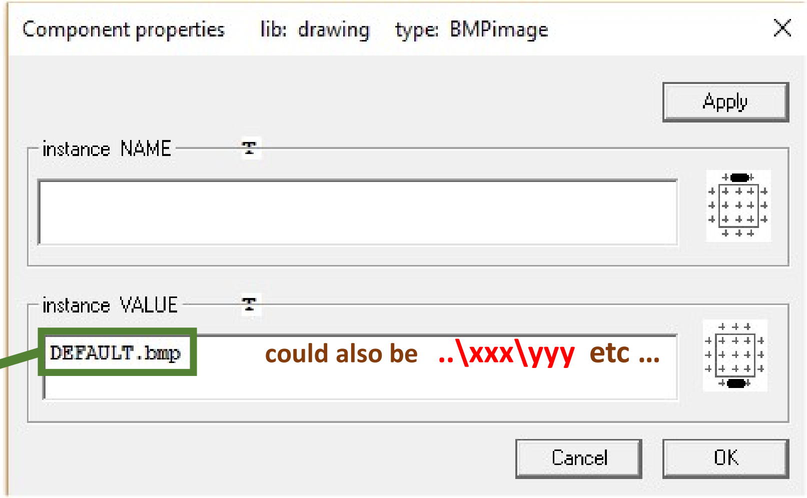
BMP image insertion



SARC schematic editor / release 3.80 Oct 18 2020 19:25:38 / (c) 2020 j-mequin@outlook.fr editing: Sch1

BMPimage "" <0|0>[-305 -215 , -66 -61] "" " D:\JM\DEV\schematic\sch_src\DEBUG\bmp\DEFAULT.bmp

corresponding
full filename



Component properties lib: drawing type: BMPimage

instance VALUE: DEFAULT.bmp

could also be ..\xxx\yyy etc ...

can be replaced with

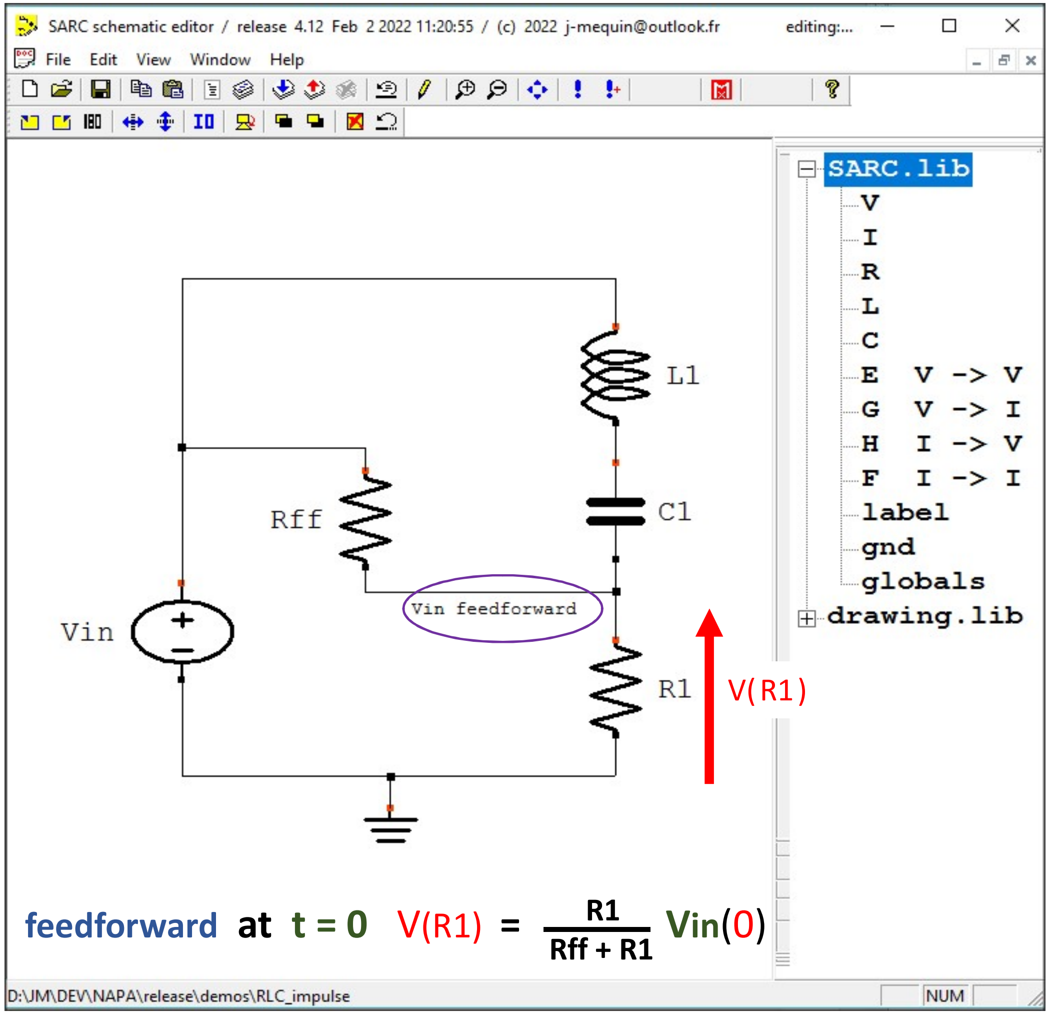
a **filename** that can be found in the **document folder** (*document must have been already saved at least once*)

a **filename** that can be found in the **schematic editor installation bmp/ folder** (= *visible by all documents*)

a **full filename**

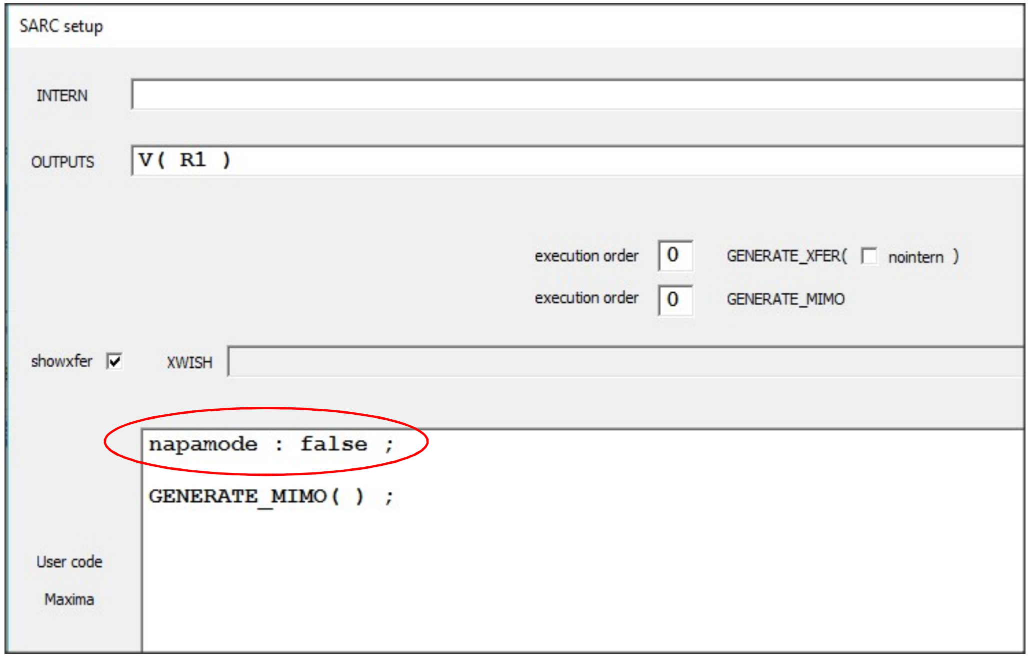
impulse response

standalone «C» validation model



D:\JM\DEV\NAPA\release\demos\RLC_impulse

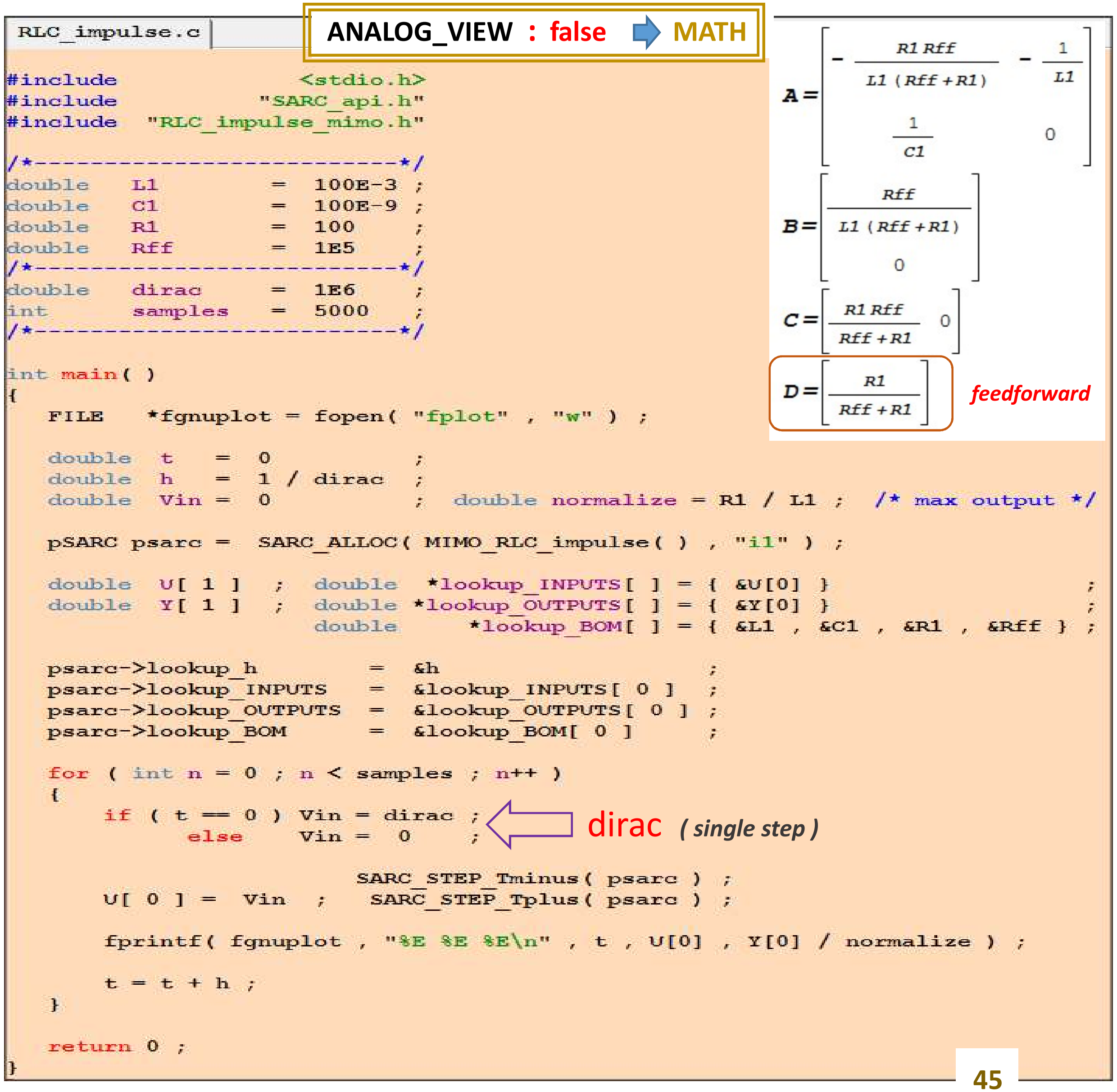
feedforward at $t = 0$ $V(R1) = \dfrac{R1}{Rff + R1} Vin(0)$



INTERN	
OUTPUTS	V(R1)
execution order	0 GENERATE_XFER(□ nointern)
execution order	0 GENERATE_MIMO
showxfer	☑
XWISH	

User code Maxima

```
napamode : false ;

GENERATE_MIMO( ) ;
```



ANALOG_VIEW : false ➡ MATH

RLC_impulse.c

```c
#include                  <stdio.h>
#include               "SARC_api.h"
#include    "RLC_impulse_mimo.h"

/*----------------------------*/
double    L1         =  100E-3  ;
double    C1         =  100E-9  ;
double    R1         =  100     ;
double    Rff        =  1E5     ;
/*----------------------------*/
double    dirac      =  1E6     ;
int       samples    =  5000    ;
/*----------------------------*/

int main( )
{
   FILE    *fgnuplot = fopen( "fplot" , "w" ) ;

   double  t    =  0            ;
   double  h    =  1 / dirac    ;
   double  Vin  =  0            ;  double normalize = R1 / L1 ;  /* max output */

   pSARC psarc =  SARC_ALLOC( MIMO_RLC_impulse( ) , "i1" ) ;

   double  U[ 1 ]  ;  double   *lookup_INPUTS[ ] = { &U[0] }                      ;
   double  Y[ 1 ]  ;  double *lookup_OUTPUTS[ ] = { &Y[0] }                       ;
                      double      *lookup_BOM[ ] = { &L1 , &C1 , &R1 , &Rff } ;

   psarc->lookup_h          =  &h                     ;
   psarc->lookup_INPUTS     =  &lookup_INPUTS[ 0 ]   ;
   psarc->lookup_OUTPUTS    =  &lookup_OUTPUTS[ 0 ]  ;
   psarc->lookup_BOM        =  &lookup_BOM[ 0 ]      ;

   for  ( int n = 0 ; n < samples ; n++ )
   {
       if  ( t == 0 ) Vin = dirac ;
             else     Vin =   0   ;

                           SARC_STEP_Tminus( psarc )  ;
       U[ 0 ] =  Vin   ;    SARC_STEP_Tplus( psarc )  ;

       fprintf( fgnuplot , "%E %E %E\n" , t , U[0] , Y[0] / normalize ) ;

       t = t + h ;
   }

   return 0 ;
}
```

dirac (single step)

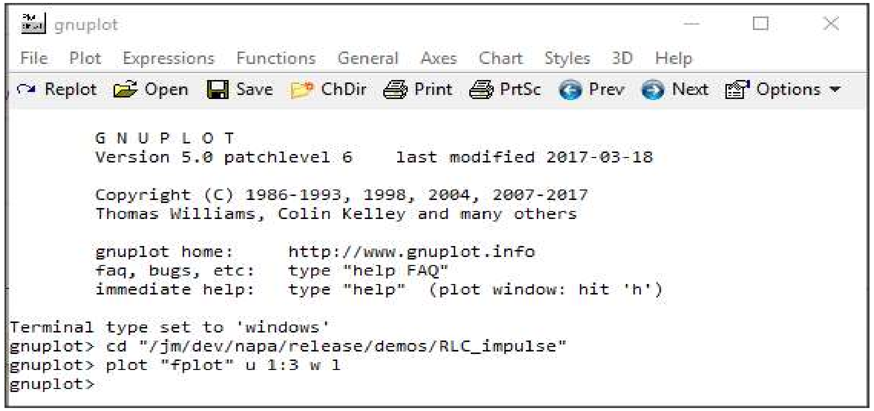
$$A=\begin{bmatrix} -\dfrac{R1\,Rff}{L1\,(Rff+R1)} & -\dfrac{1}{L1} \\ \dfrac{1}{C1} & 0 \end{bmatrix}$$

$$B=\begin{bmatrix} \dfrac{Rff}{L1\,(Rff+R1)} \\ 0 \end{bmatrix}$$

$$C=\begin{bmatrix} \dfrac{R1\,Rff}{Rff+R1} & 0 \end{bmatrix}$$

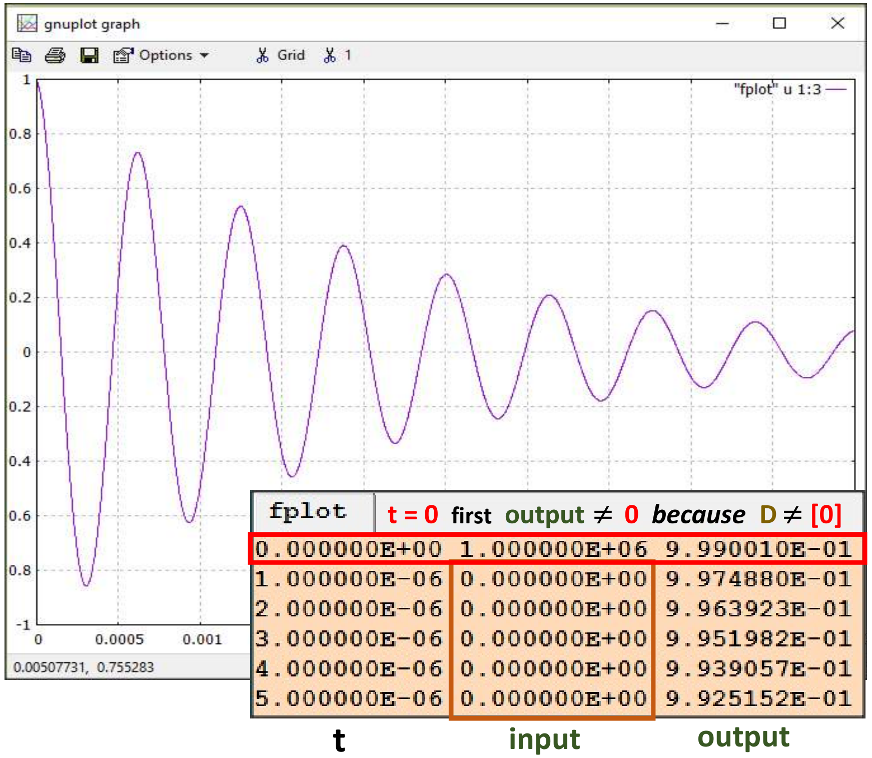
$$\boxed{D=\begin{bmatrix} \dfrac{R1}{Rff+R1} \end{bmatrix}}$$

feedforward



gnuplot
File Plot Expressions Functions General Axes Chart Styles 3D Help
Replot Open Save ChDir Print PrtSc Prev Next Options

```
        G N U P L O T
        Version 5.0 patchlevel 6    last modified 2017-03-18

        Copyright (C) 1986-1993, 1998, 2004, 2007-2017
        Thomas Williams, Colin Kelley and many others

        gnuplot home:     http://www.gnuplot.info
        faq, bugs, etc:   type "help FAQ"
        immediate help:   type "help"  (plot window: hit 'h')

Terminal type set to 'windows'
gnuplot> cd "/jm/dev/napa/release/demos/RLC_impulse"
gnuplot> plot "fplot" u 1:3 w 1
gnuplot>
```



fplot   t = 0 first output ≠ 0 because D ≠ [0]

```
0.000000E+00 1.000000E+06 9.990010E-01
1.000000E-06 0.000000E+00 9.974880E-01
2.000000E-06 0.000000E+00 9.963923E-01
3.000000E-06 0.000000E+00 9.951982E-01
4.000000E-06 0.000000E+00 9.939057E-01
5.000000E-06 0.000000E+00 9.925152E-01
```

t   input   output

APPENDIX

The **wxPAUSE( )** function can be used to force a **pause** during the execution of a **wxMaxima** script

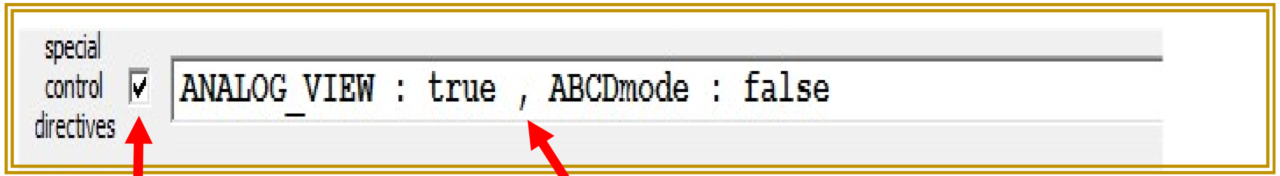
default SARC *environment variables*

		default		
napamode	**:**	**false**	**;**	**true** / **false**
ANALOG_VIEW	**:**	**false**	**;**	**true** / **false**
convention	**:**	**"PSC"**	**;**	**"PSC"** / **"GLC“**
ABCDmode	**:**	**true**	**;**	**true** / **false**

or GCC *compile option*
-DGCC_ANALOG_VIEW=0/1

could be *redefined* in the **Maxima** *initialisation environment file*

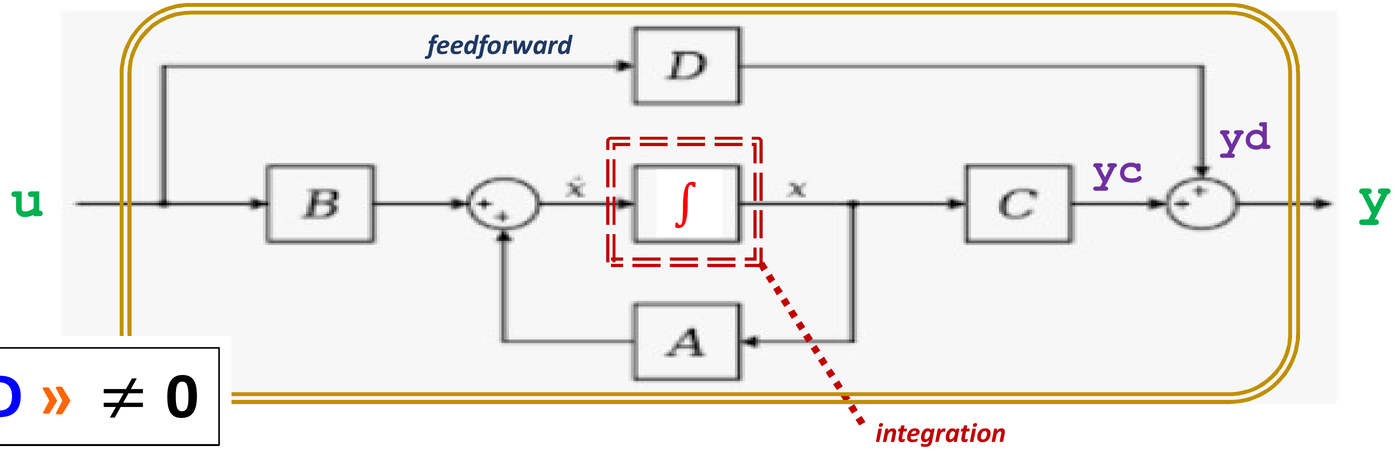
or **in a *SARC setup pop up* menu**



special control directives ☑ ANALOG_VIEW : true , ABCDmode : false

***enable* or *disable* these directives**

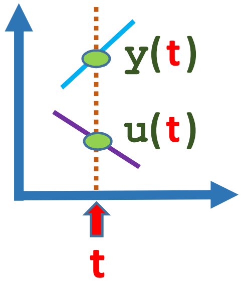
comma



when « D » ≠ 0

mathematical response $y(t) = \boxed{yc(t) + yd(t)} = \boxed{fc(\,u(t-1)\,) + fd(\,u(t)\,)}$

zero delay dependency



This is not an issue for the SARC validation environment that simulates standalone « C » model *individually*

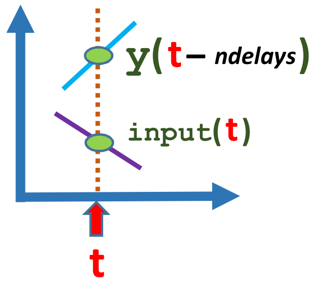
but, *independently of* ***SARC*** *, zero delay blocks are almost always* ***delayed**** *by simulators* (*including* **NAPA**)

simulator response $y(t) = \boxed{fc(\,u(t-2)\,)} + \boxed{fd(\,u(t-1)\,)}$

this is ok

but implies ***plots***

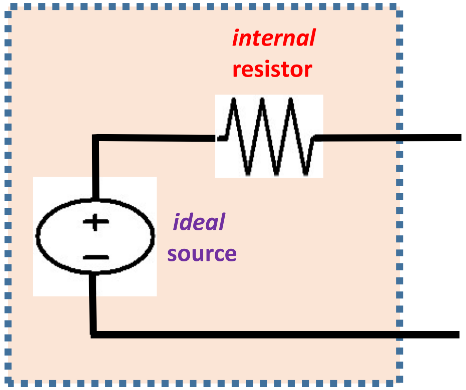
(imperceptible with small steps)

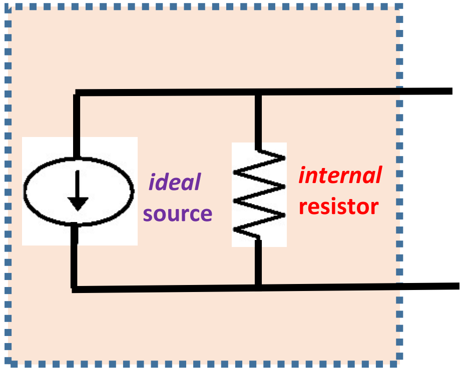


* *to* *prevent* *the* *eventuality* *of a* ***zero delay*** *assembly* *building* *a snake biting its own tail*

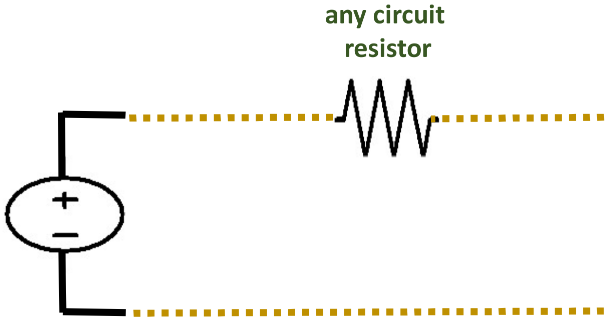
(regarding **SARC** *the fact that* **D** *could be* ≠ **0** *should not qualify the block as "zero delay" since the feedforward is handled internally only once per step)*

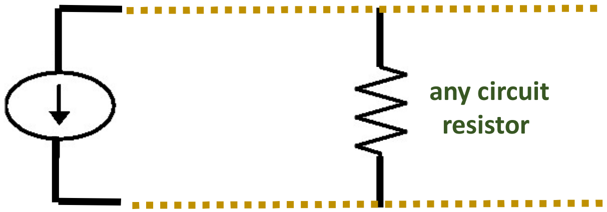
academic
physical source





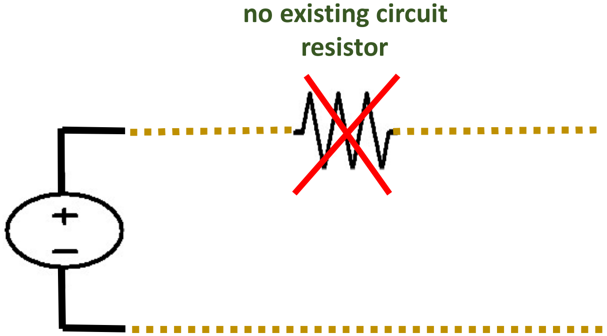
SARC
proper source

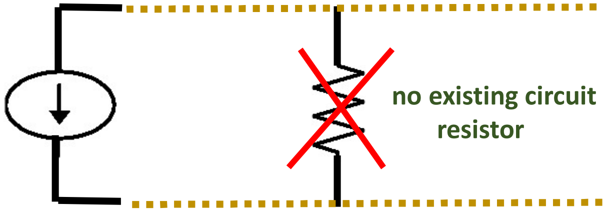




SARC
boundless source

NO RESISTOR ➡ *NO TIME CONSTANT*

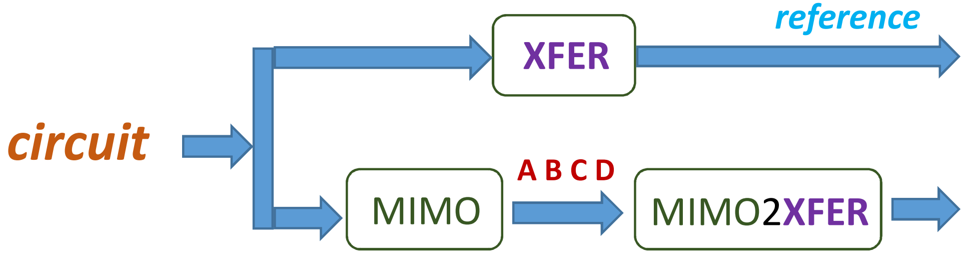




QC methods

for « *healthy* » outputs (without *Warnings*)

1)

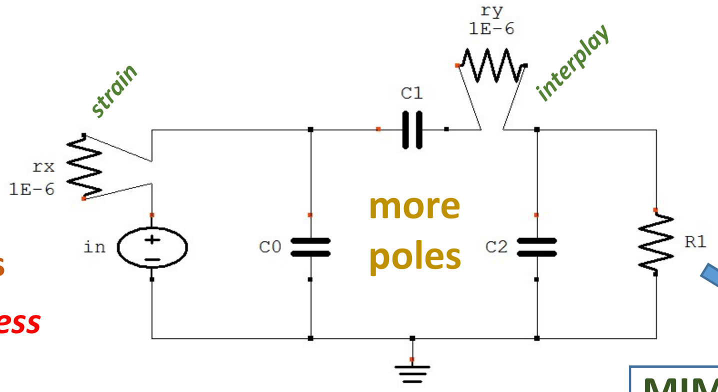


??? *transfer functions* *should be* *the same*

this verifies that A B C D are the correct ones

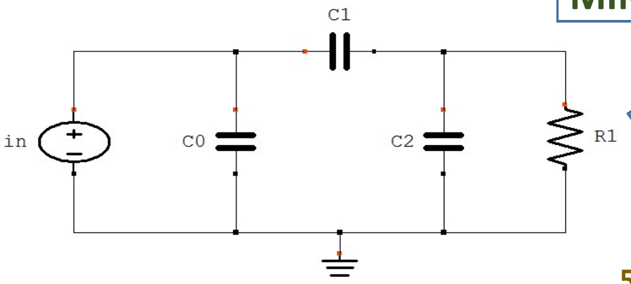
2)

updated circuit
with small resistors
to eliminate ALL *boundless*
(*strain* *and* *interplay*)



more
poles

boundless circuit
under QC

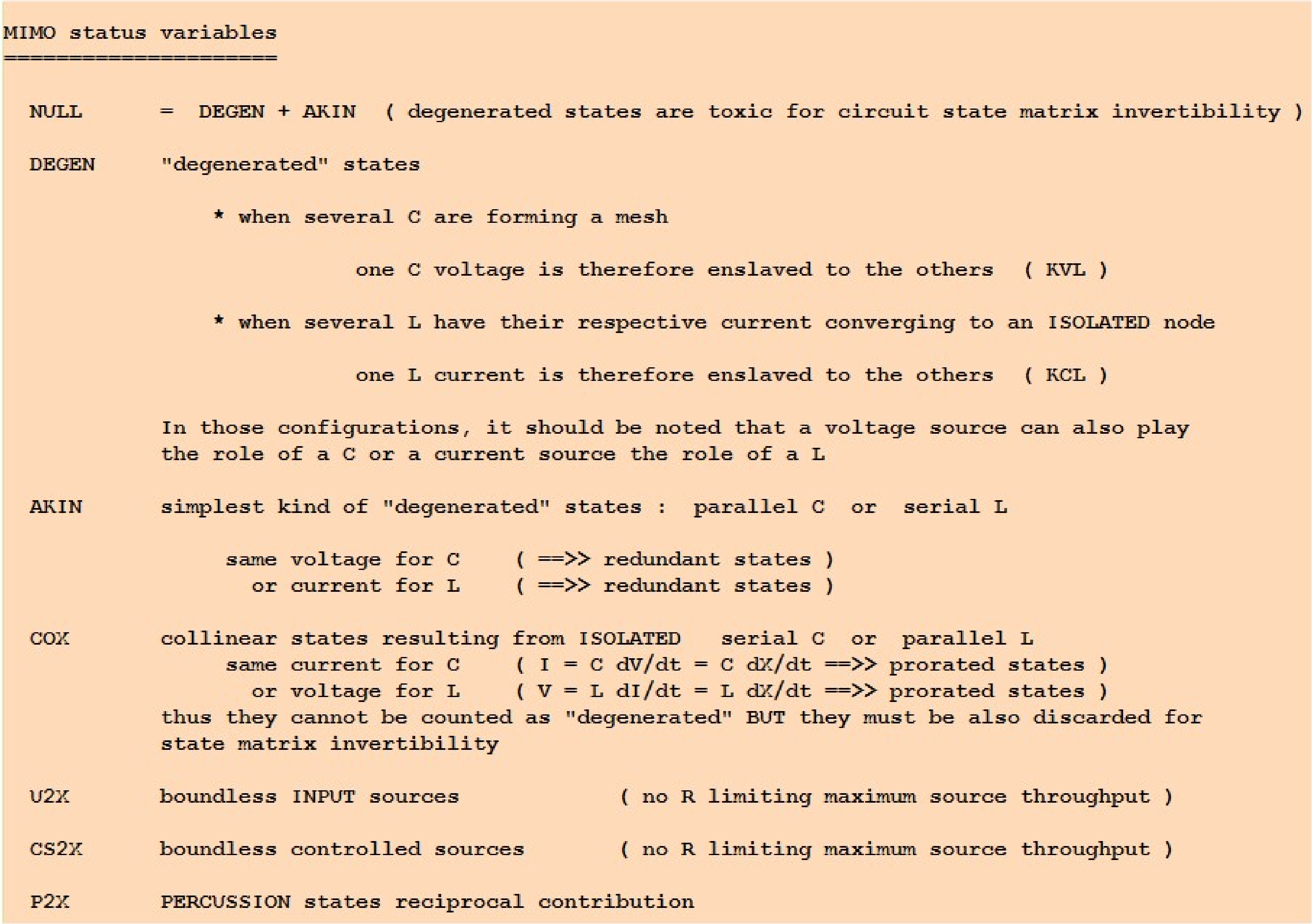


reference

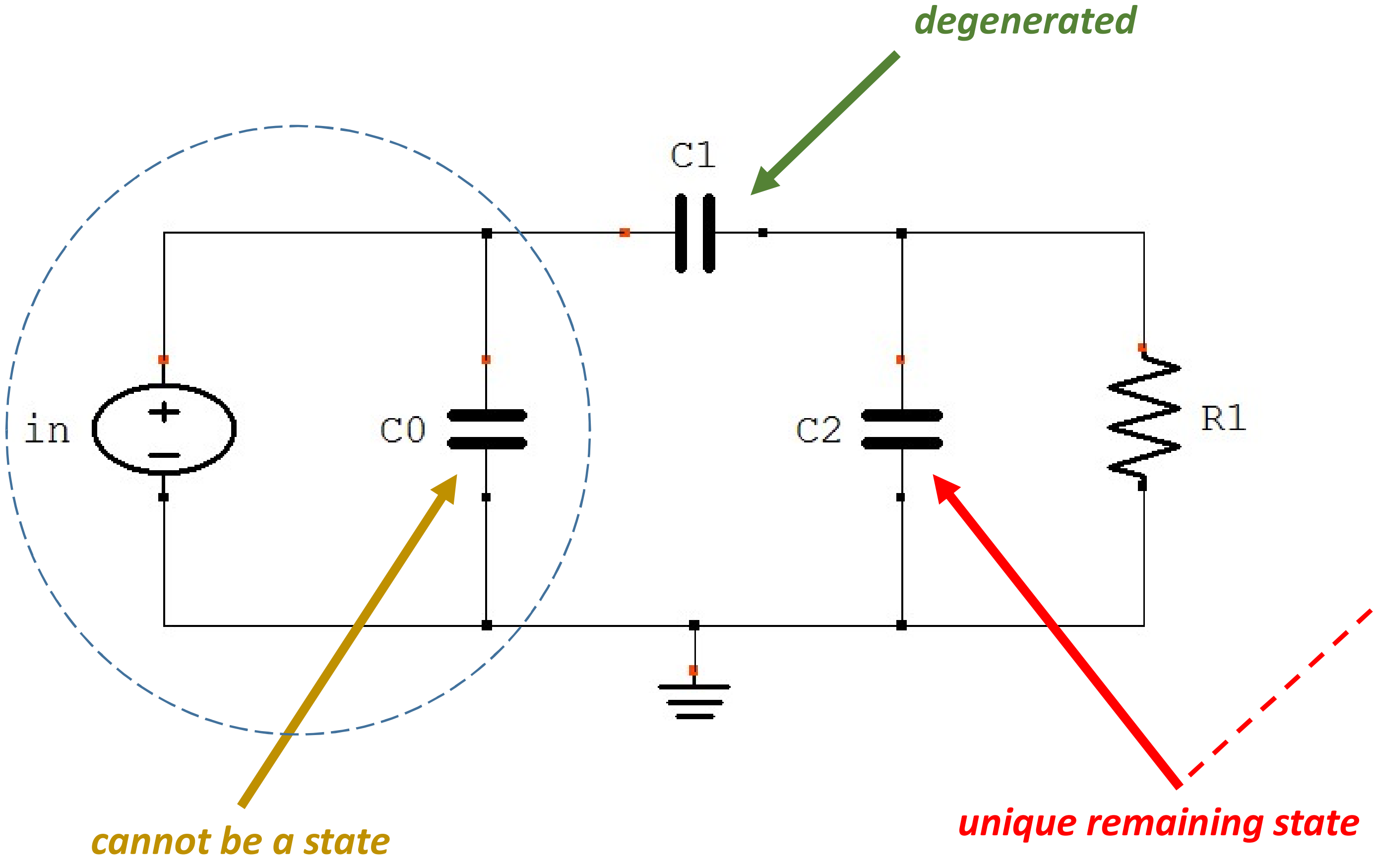
MIMO *simulations*

??? *plots* *should be* *the same*

this verifies that the “*boundless*” are well handled



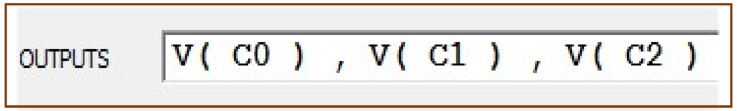
```
MIMO status variables
=====================

  NULL      =  DEGEN + AKIN  ( degenerated states are toxic for circuit state matrix invertibility )

  DEGEN     "degenerated" states

                * when several C are forming a mesh

                           one C voltage is therefore enslaved to the others   ( KVL )

                * when several L have their respective current converging to an ISOLATED node

                           one L current is therefore enslaved to the others   ( KCL )

            In those configurations, it should be noted that a voltage source can also play
            the role of a C or a current source the role of a L

  AKIN      simplest kind of "degenerated" states :  parallel C  or  serial L

                 same voltage for C    ( ==>> redundant states )
                   or current for L    ( ==>> redundant states )

  COX       collinear states resulting from ISOLATED   serial C  or  parallel L
                 same current for C    ( I = C dV/dt = C dX/dt ==>> prorated states )
                   or voltage for L    ( V = L dI/dt = L dX/dt ==>> prorated states )
            thus they cannot be counted as "degenerated" BUT they must be also discarded for
            state matrix invertibility

  U2X       boundless INPUT sources            ( no R limiting maximum source throughput )

  CS2X      boundless controlled sources       ( no R limiting maximum source throughput )

  P2X       PERCUSSION states reciprocal contribution
```



degenerated

cannot be a state

unique remaining state



V(C0) , V(C1) , V(C2)

```
... NLC          : 3
... NESE         : 3
... NSTATES      : 1
... XWISH        : [C0,C2]
 SERIAL    []
PARALLEL  [[C2,R1]]
```

$$STATES = \begin{bmatrix} V(C2) \end{bmatrix}$$

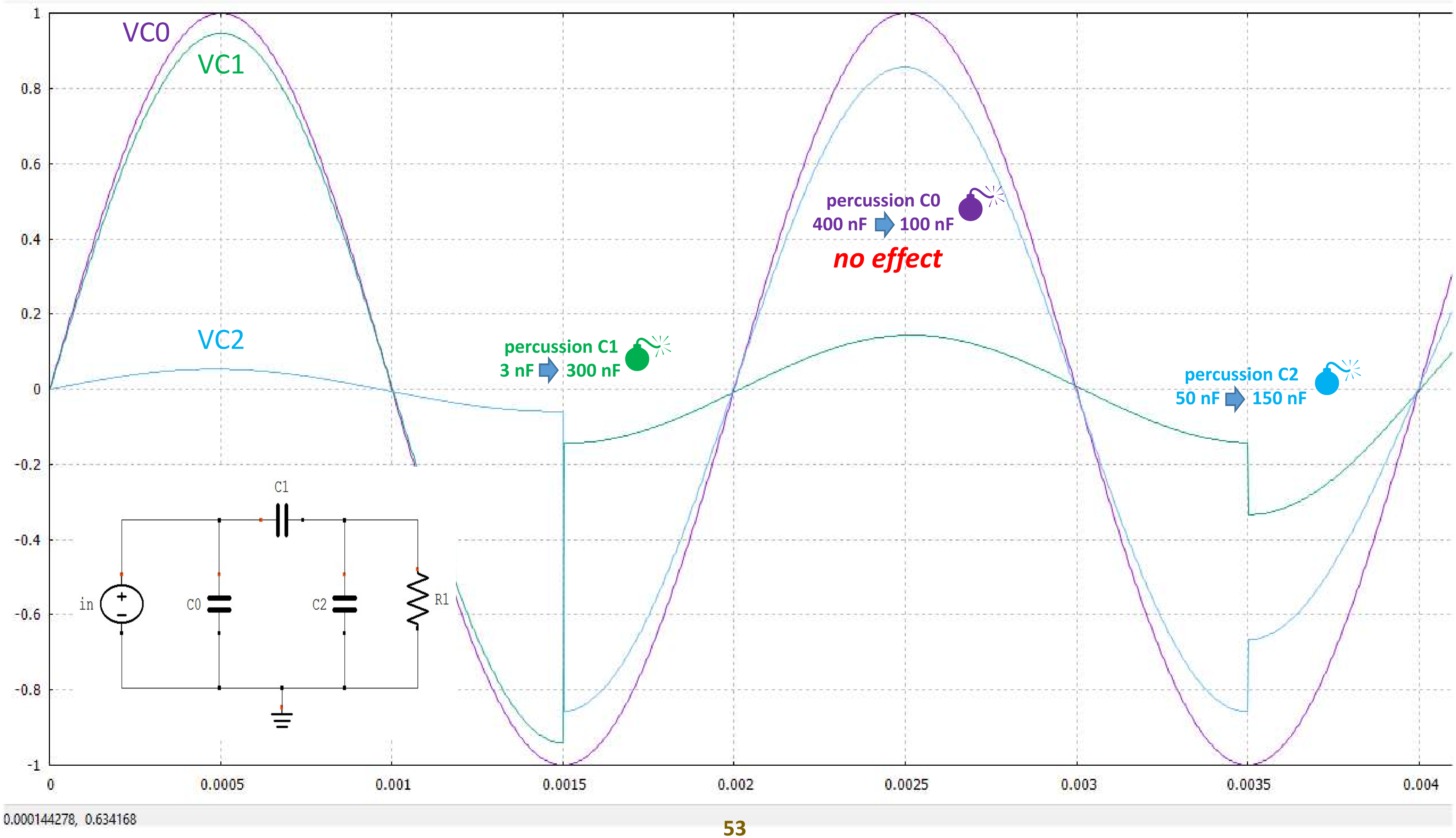
$$ESE = \begin{bmatrix} [C0] \\ [C1] \\ [C2] \end{bmatrix}$$

$$A = \begin{bmatrix} -\frac{1}{(C2 + C1)\ R1} \end{bmatrix}$$

$$B = \begin{bmatrix} -\frac{C1}{(C2 + C1)^2\ R1} \end{bmatrix}$$

$$C = \begin{bmatrix} 0 \\ -1 \\ 1 \end{bmatrix}$$

$$D = \begin{bmatrix} 1 \\ \frac{C2}{C2 + C1} \\ \frac{C1}{C2 + C1} \end{bmatrix}$$



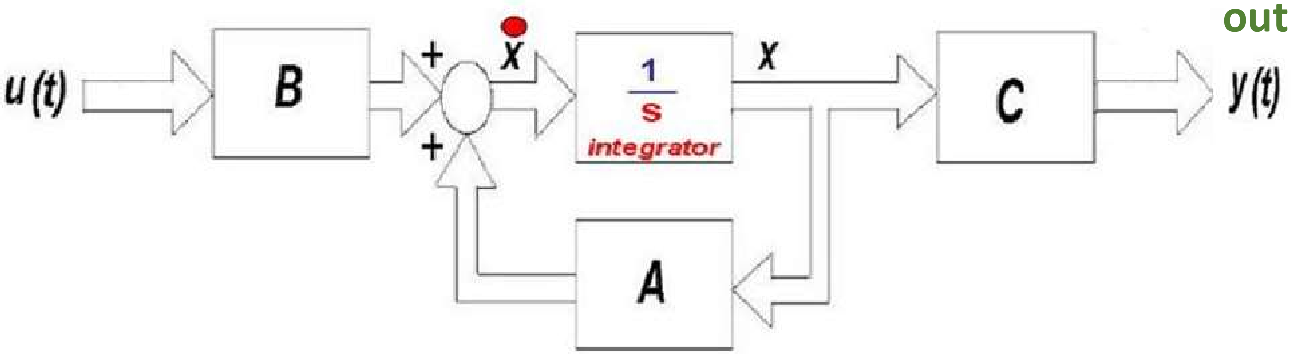
percussion C0
400 nF ➡ 100 nF

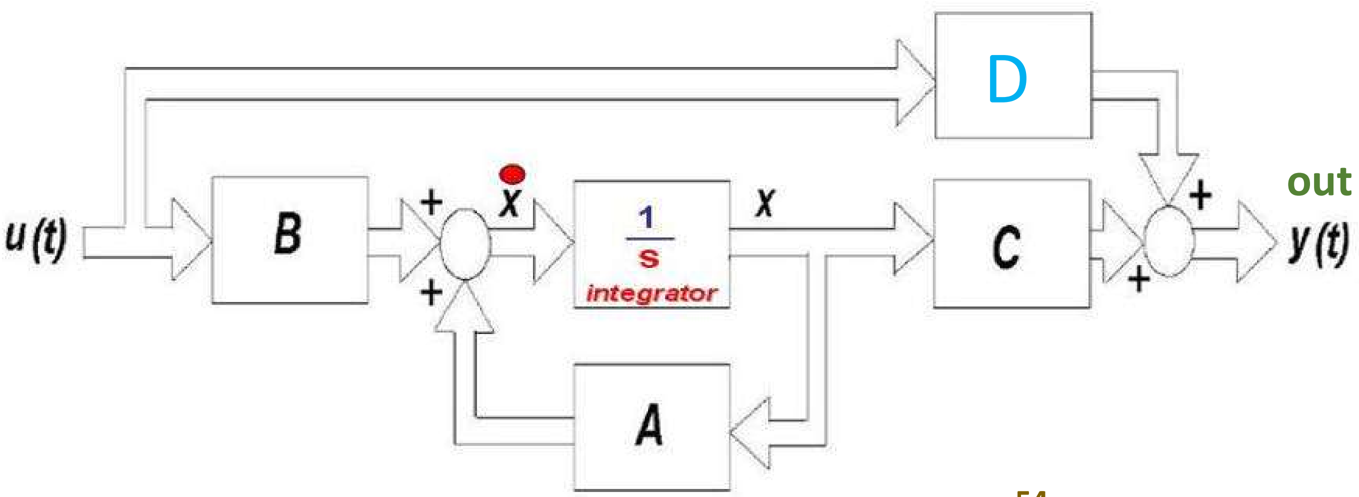
no effect

percussion C1
3 nF ➡ 300 nF

percussion C2
50 nF ➡ 150 nF

0.000144278, 0.634168





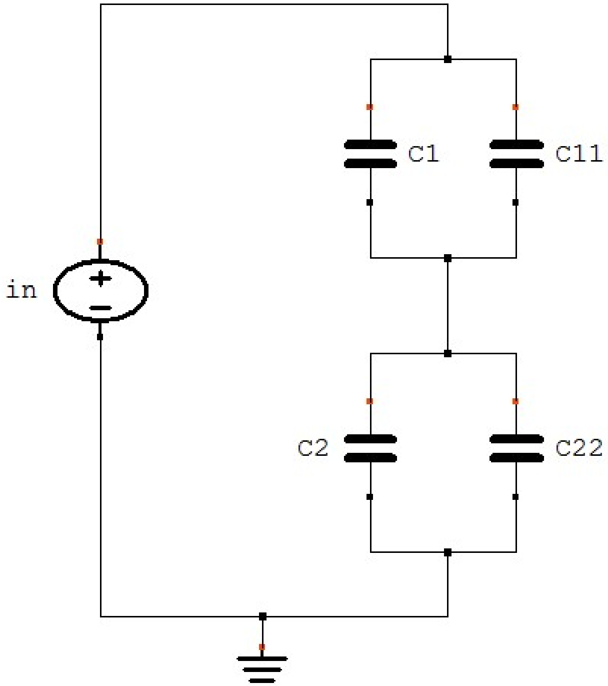
MIMO2XFER

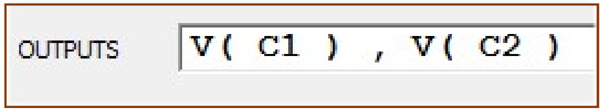
PROPER ***transfer function***

$$D[\text{out}] = 0$$

BIPROPER ***transfer function***

$$D[\text{out}] \neq 0$$




OUTPUTS V(C1) , V(C2)

GENERATE_XFER

SERIAL []

PARALLEL [[C1,C11],[C2,C22]]

1 $V(C1)/in = \dfrac{C22+C2}{C22+C2+C11+C1}$

2 $V(C2)/in = \dfrac{C11+C1}{C22+C2+C11+C1}$

no |s| term ⟺ no pole

GENERATE_MIMO

SERIAL []

PARALLEL [[C1,C11],[C2,C22]]

STATES = none

$$ESE = \begin{bmatrix} [C1, C11] \\ [C2, C22] \end{bmatrix}$$

$$D = \begin{bmatrix} \dfrac{C22+C2}{C22+C2+C11+C1} \\ \dfrac{C11+C1}{C22+C2+C11+C1} \end{bmatrix}$$

no state ⟺ no eigenvalue

only D *feedforward* matrix

MIMO2XFER

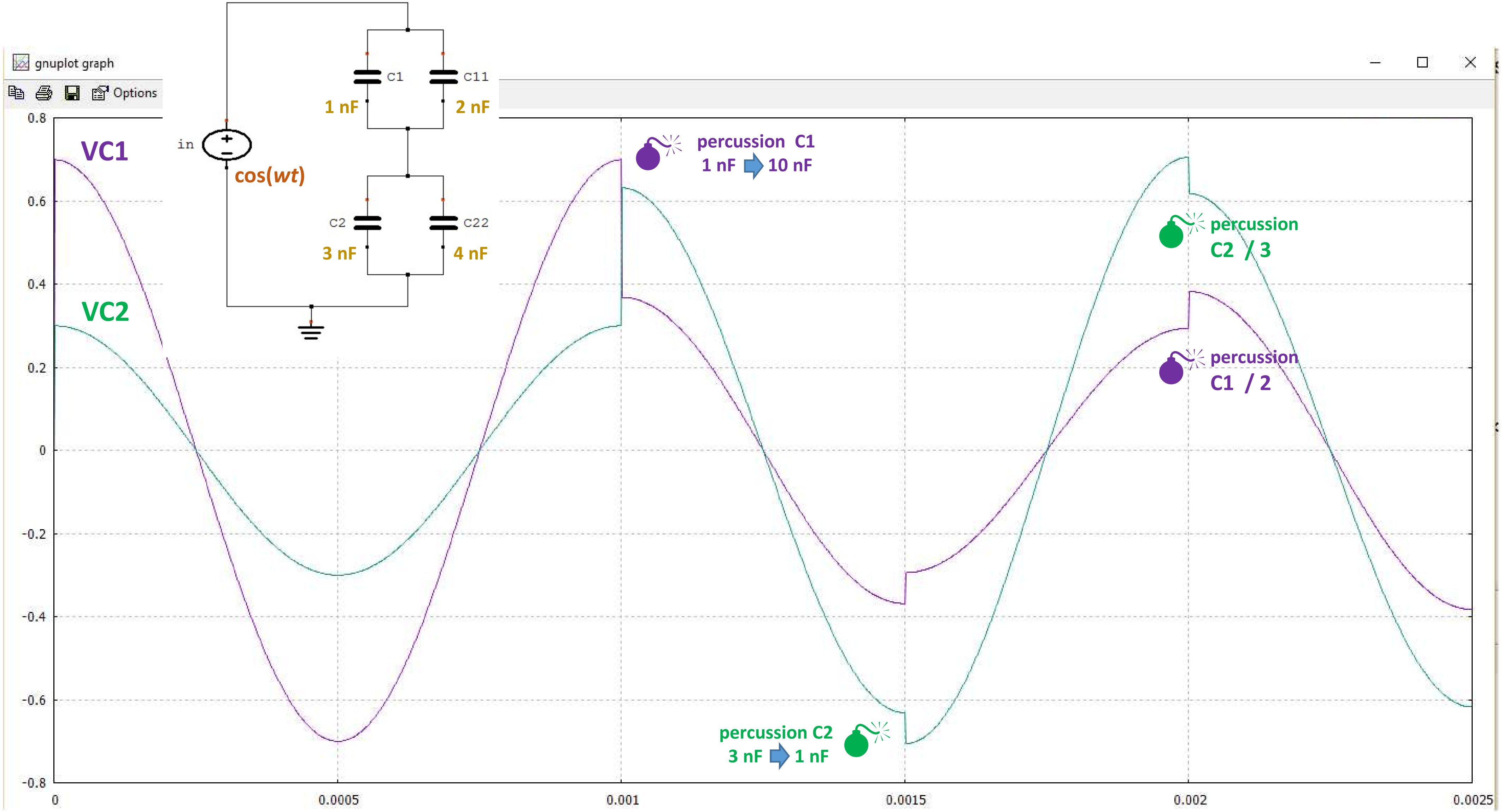
$$XFER = C\,\boxed{(\,|s|\,I - A\,)}^{-1}\,B + D$$

... INPUTS : [in] OMITTED : n/a

... OUTPUTS : [V(C1),V(C2)]

1 $V(C1)/in = \dfrac{C22+C2}{C22+C2+C11+C1}$

2 $V(C2)/in = \dfrac{C11+C1}{C22+C2+C11+C1}$

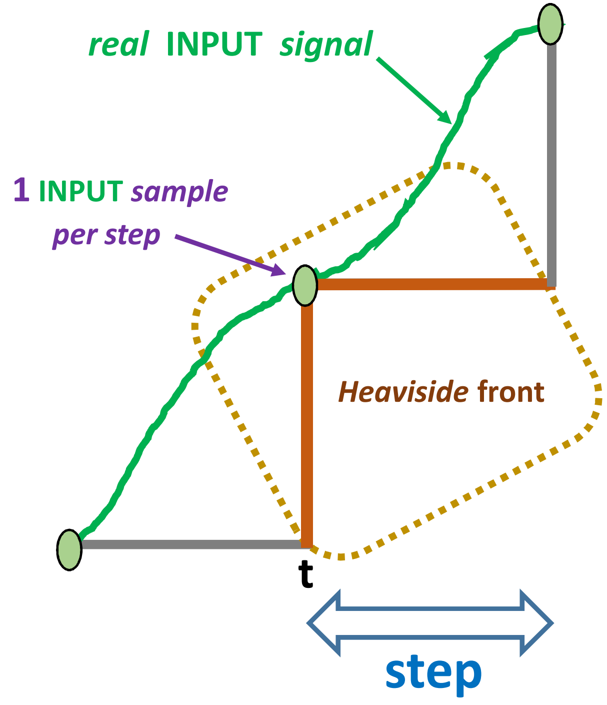

gnuplot graph
Options
VC1
VC2
in
cos(*wt*)
C1
1 nF
C11
2 nF
C2
3 nF
C22
4 nF
percussion C1
1 nF ➡ 10 nF
percussion C2
3 nF ➡ 1 nF
percussion
C2 / 3
percussion
C1 / 2
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
0
0.0005
0.001
0.0015
0.002
0.0025

MATH / ANALOG VIEW

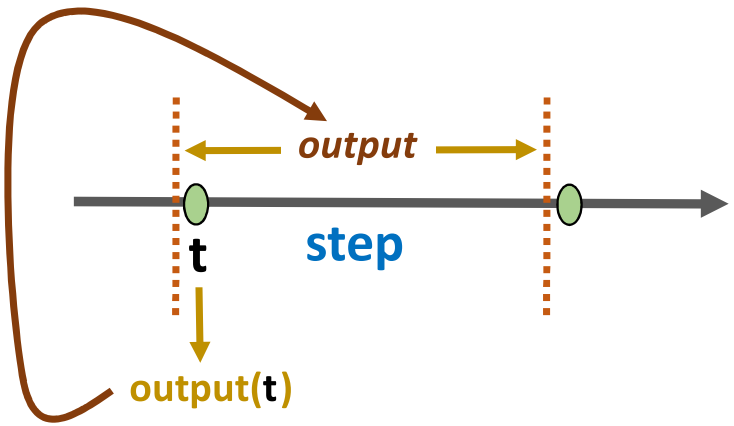
CLASSICAL SIMULATION

(almost always not accepting "on fly" component value change)

1 INPUT sample ➡ 1 OUTPUT data

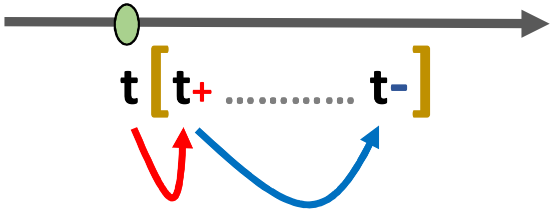


1 plot per step



SARC SIMULATION

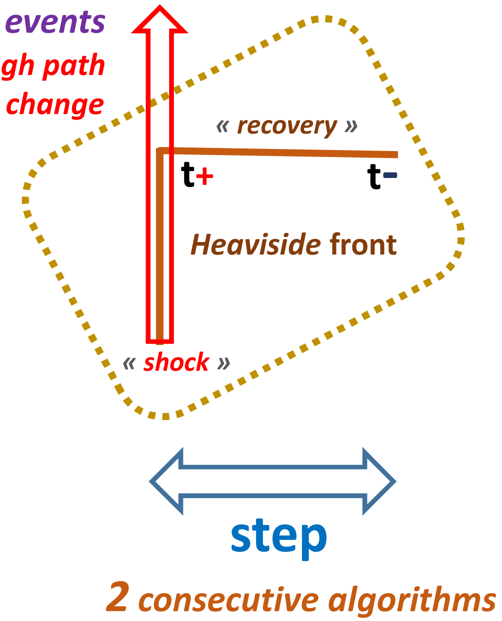
1 INPUT sample ➡ 2 OUTPUT data



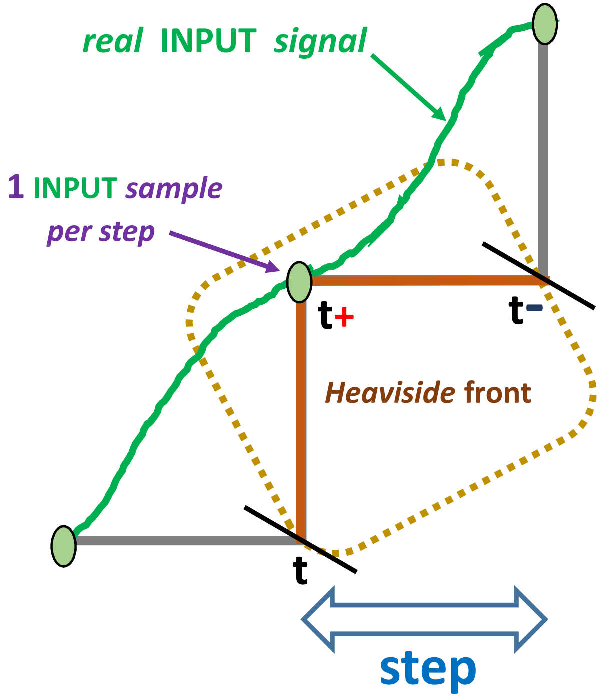
singular events

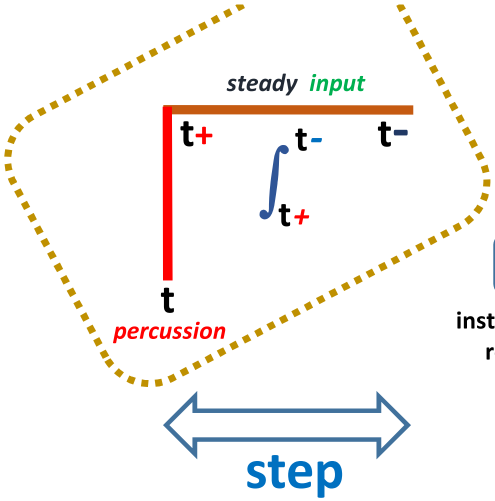
direct feedthrough path

or / and component(s) value change



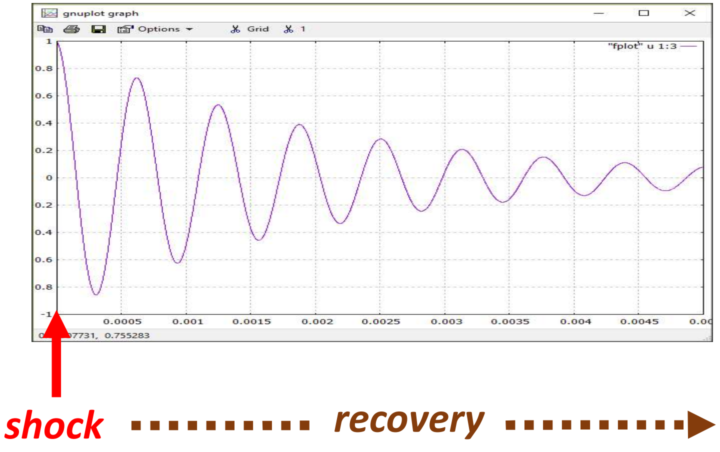
2 *consecutive algorithms*





2 OUTPUT data
per step

t+	t-
instantaneous response	time constants response



SARC SIMULATION

1 plot per step

The **VIEW** *is a* ***user choice*** *that* ***does not affect*** *the overall*

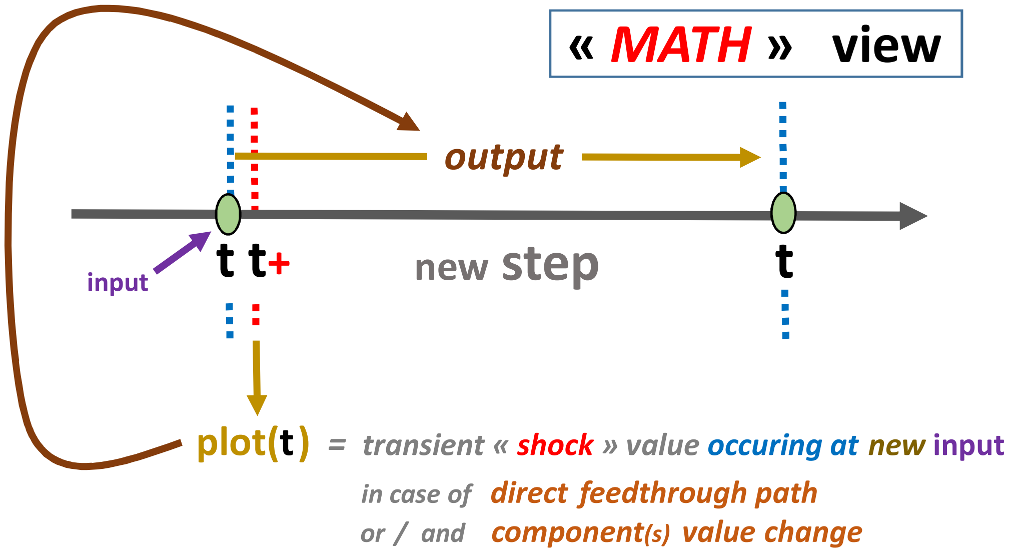
accuracy of the simulation

which remains based on the ***consecutive processing***

of ***both*** **t+** *and* **t-**

However, this is only true when the ***entire simulated circuit*** has been **edited** as a ***single SARC schematic***

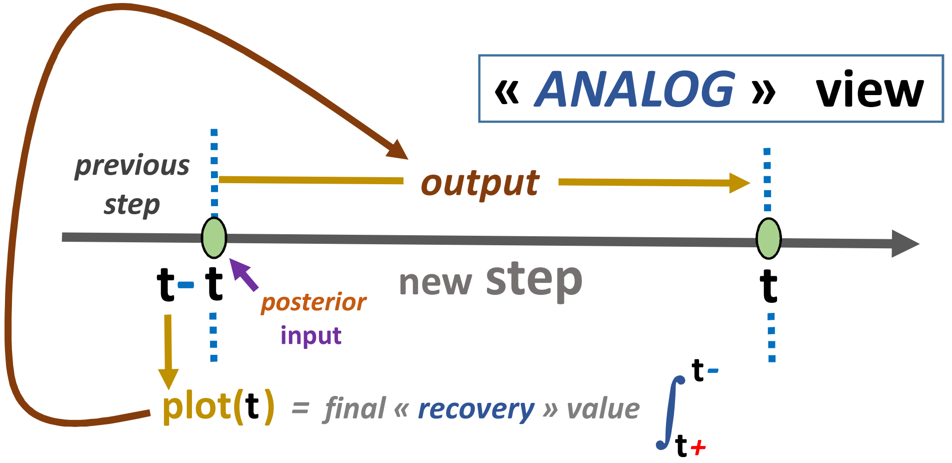
ANALOG **view** is required to simulate ***assembly of blocks***



plot(t) = *transient « **shock** » value* ***occuring at new*** input

in case of ***direct feedthrough path***

or / and ***component(s) value change***

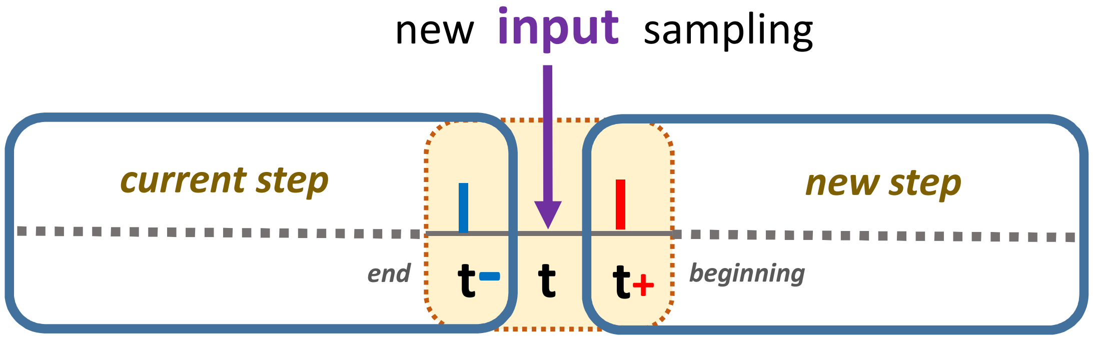


plot(t) = *final « **recovery** » value* $\int_{t+}^{t-}$

$$\boxed{Y = C\,.\,X + D\,.\,\boxed{U}} \;\Leftarrow \underset{\Delta t}{\int} \sqsupset\; \boxed{\dot{X} = A\,.\,X + B\,.\,\boxed{U}}$$

integration delay

direct feedthrough path ⇔ $D \neq 0$



if feedthrough and input changing ⇒ $t^{+} \neq t^{-}$

and / or if component percussions ⇒ $t^{+} \neq t^{-}$

otherwise input changing or unchanged ⇒ $t^{+} = t^{-}$

because **SARC** *has its intrinsic* ***integration delay***

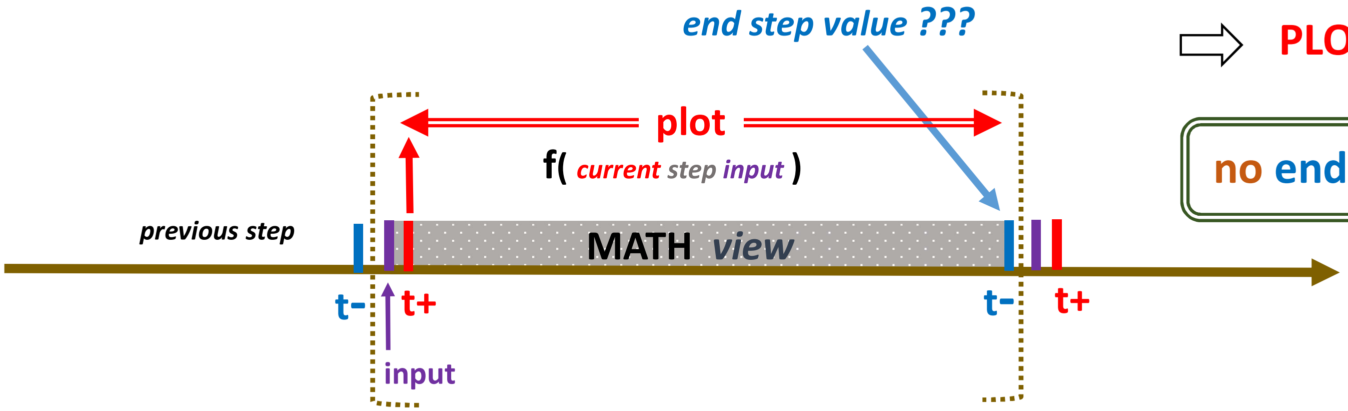
MATH = ANALOG

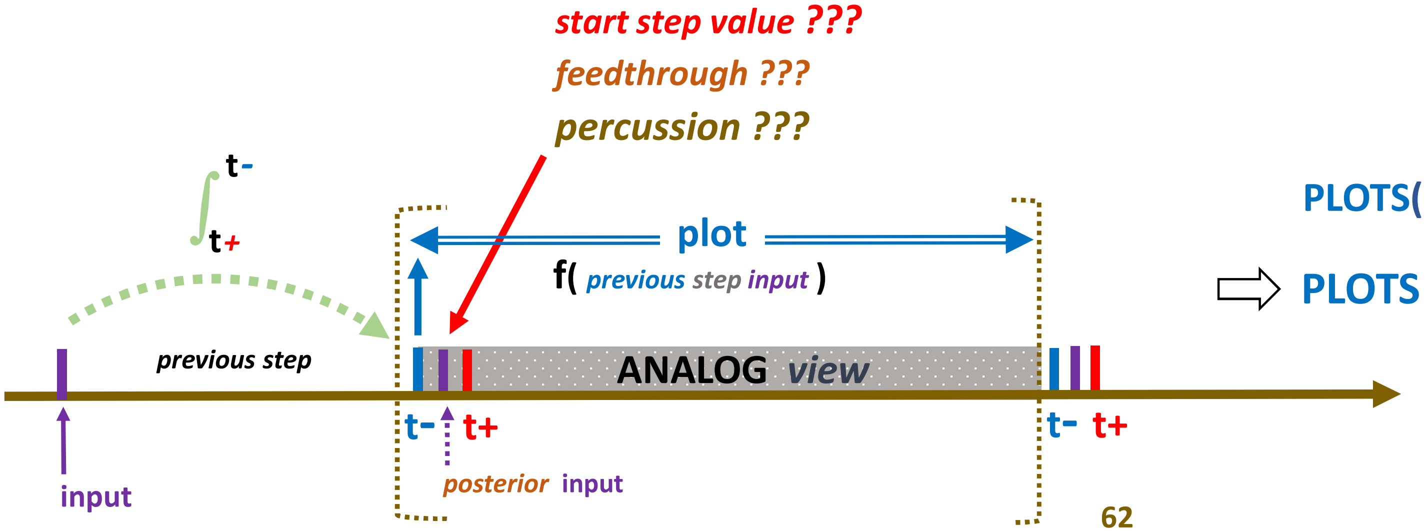
SARC uncertainty

$$\text{PLOTS}(\,t\,) = f(\ \text{INPUTS}[\,t + \varepsilon\,]\ )$$

⇨ **PLOTS** *are* *concomitant* *with* **INPUTS**

no end value *to enable assembly of blocks*





$$\text{PLOTS}(\,t\,) = f(\ \text{INPUTS}[\,t - \varepsilon\,]\ )$$

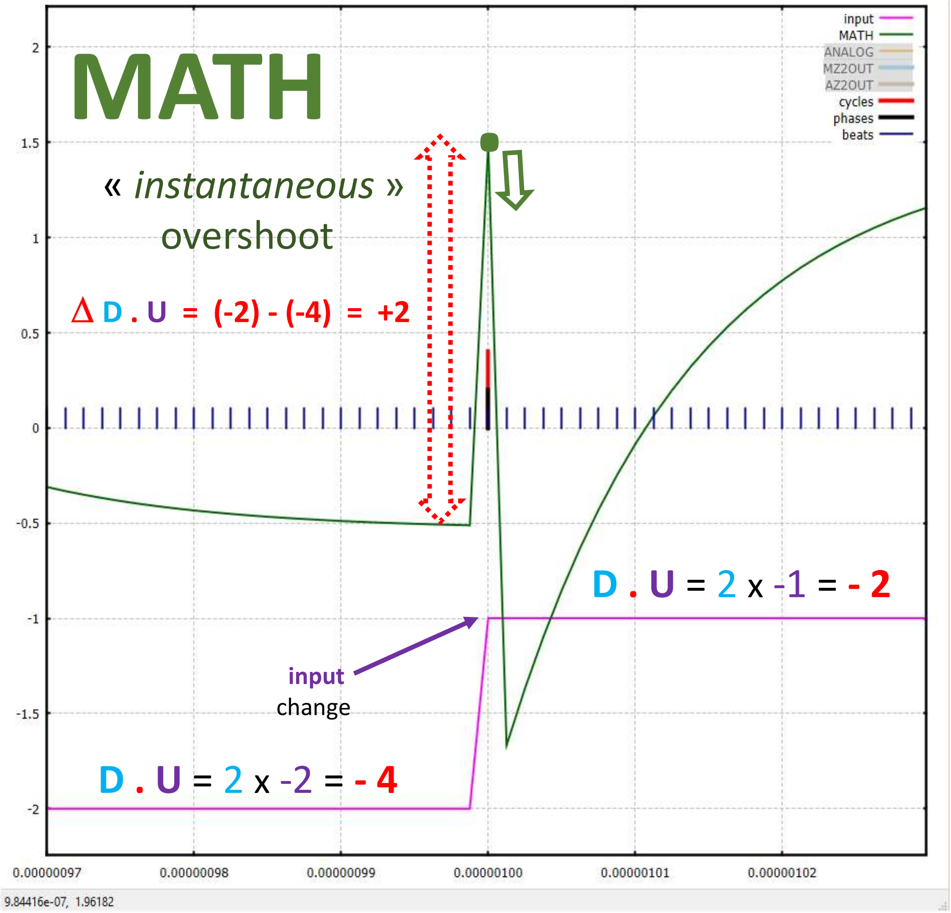
⇨ **PLOTS** *are* *postdated* *versus* **INPUTS**

SARC uncertainty

$$\boxed{Y = C \, . \, X + \boxed{D \, . \, U}}$$

direct feedthrough path matrix $D = \begin{bmatrix} 2 , 0 \end{bmatrix} \neq 0$

inputs vector $U = \begin{bmatrix} -2 \\ 0 \end{bmatrix}$ *followed by* $\begin{bmatrix} -1 \\ 0 \end{bmatrix}$



MATH

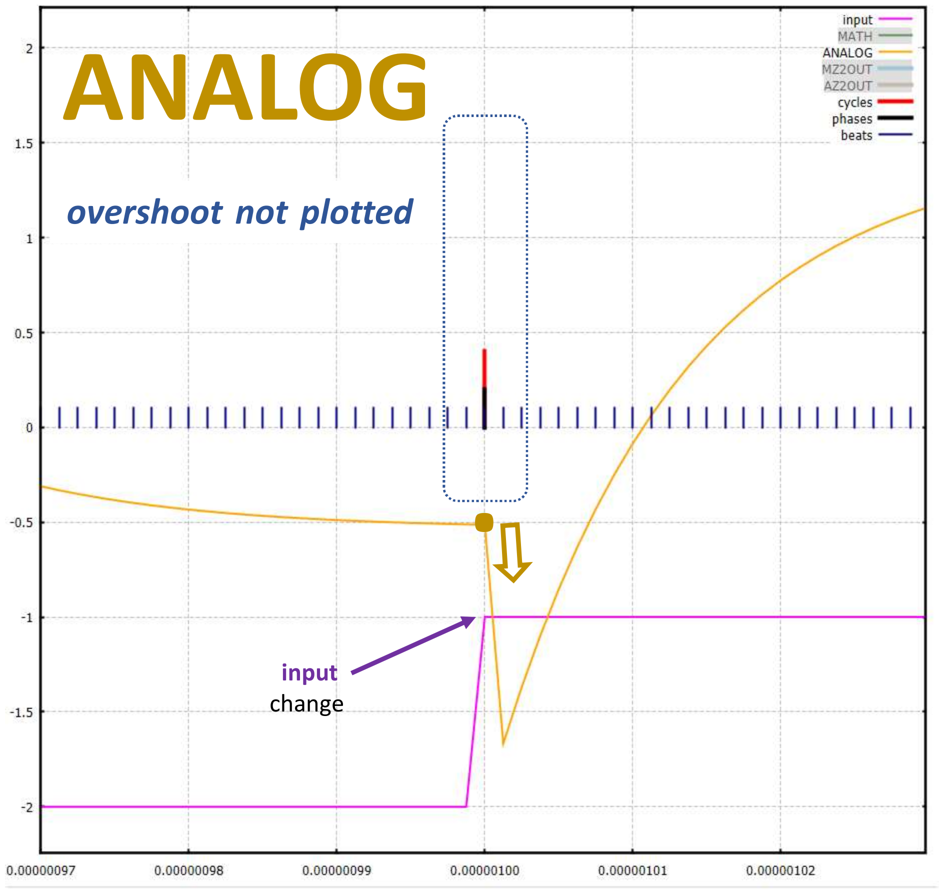
« *instantaneous* »
overshoot

$\Delta D . U = (-2) - (-4) = +2$

$D . U = 2 \times -1 = -2$

input
change

$D . U = 2 \times -2 = -4$



ANALOG

overshoot not plotted

input
change

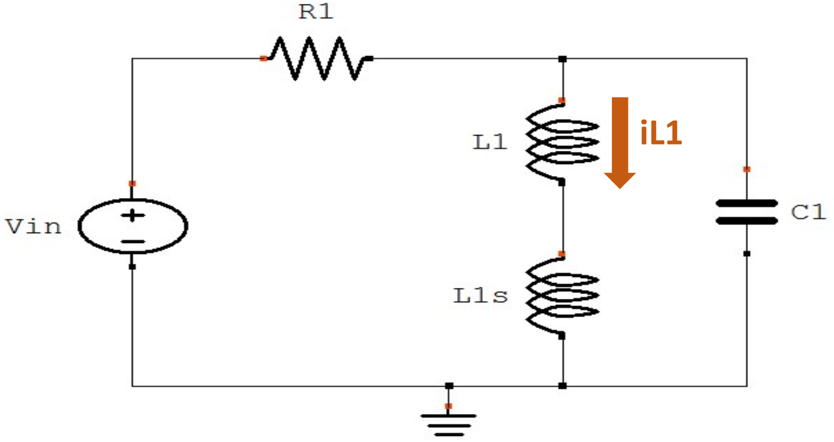
SARC uncertainty

$$Y = C \cdot \boxed{X} + D \cdot U$$

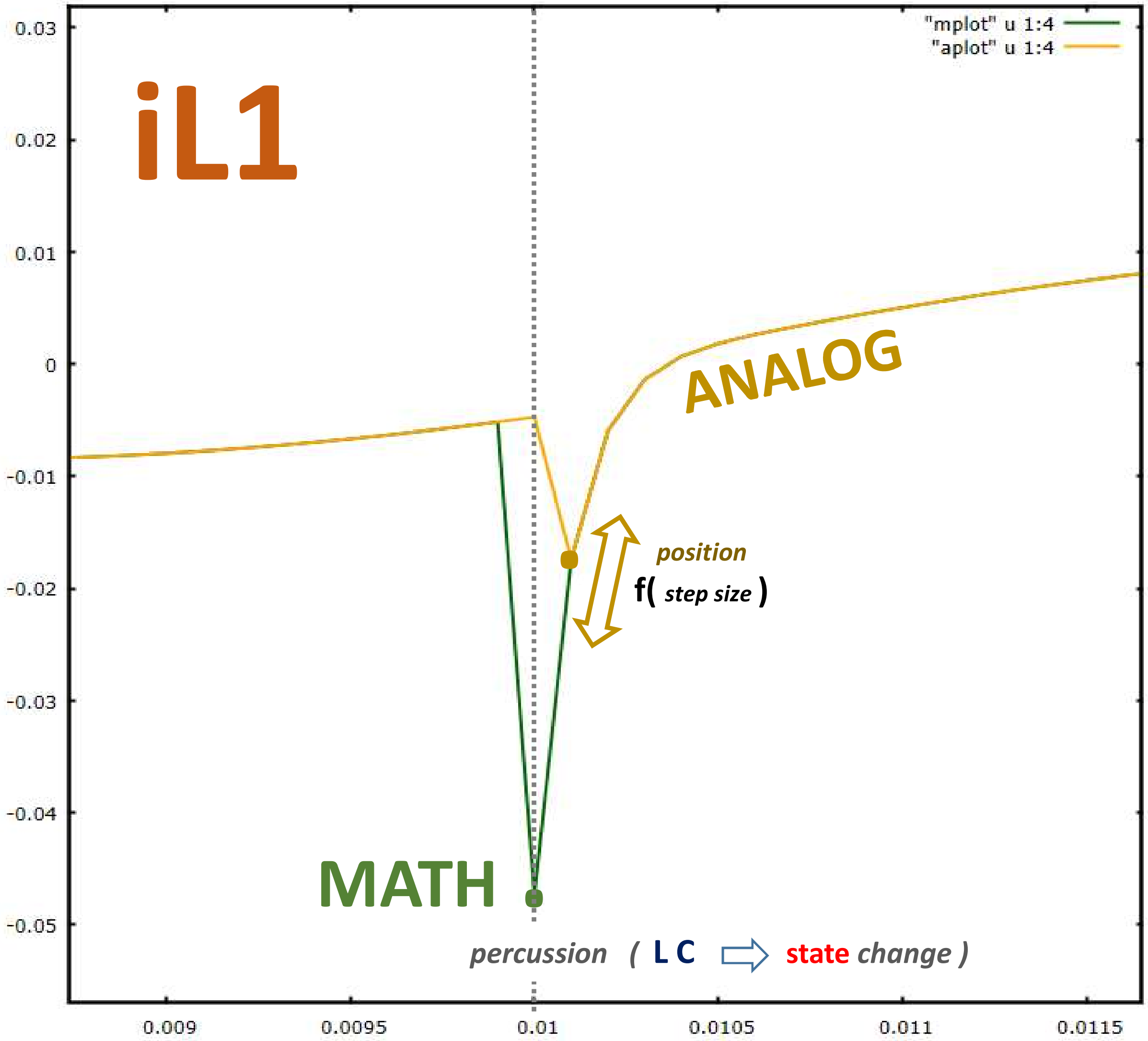
state

component value change

could affect C *or/and* D (and X if L C)



L1 = 90 mH
L1s = 10 mH
	at t = **0.01**	L1 = 6 mH
		L1s = 4 mH

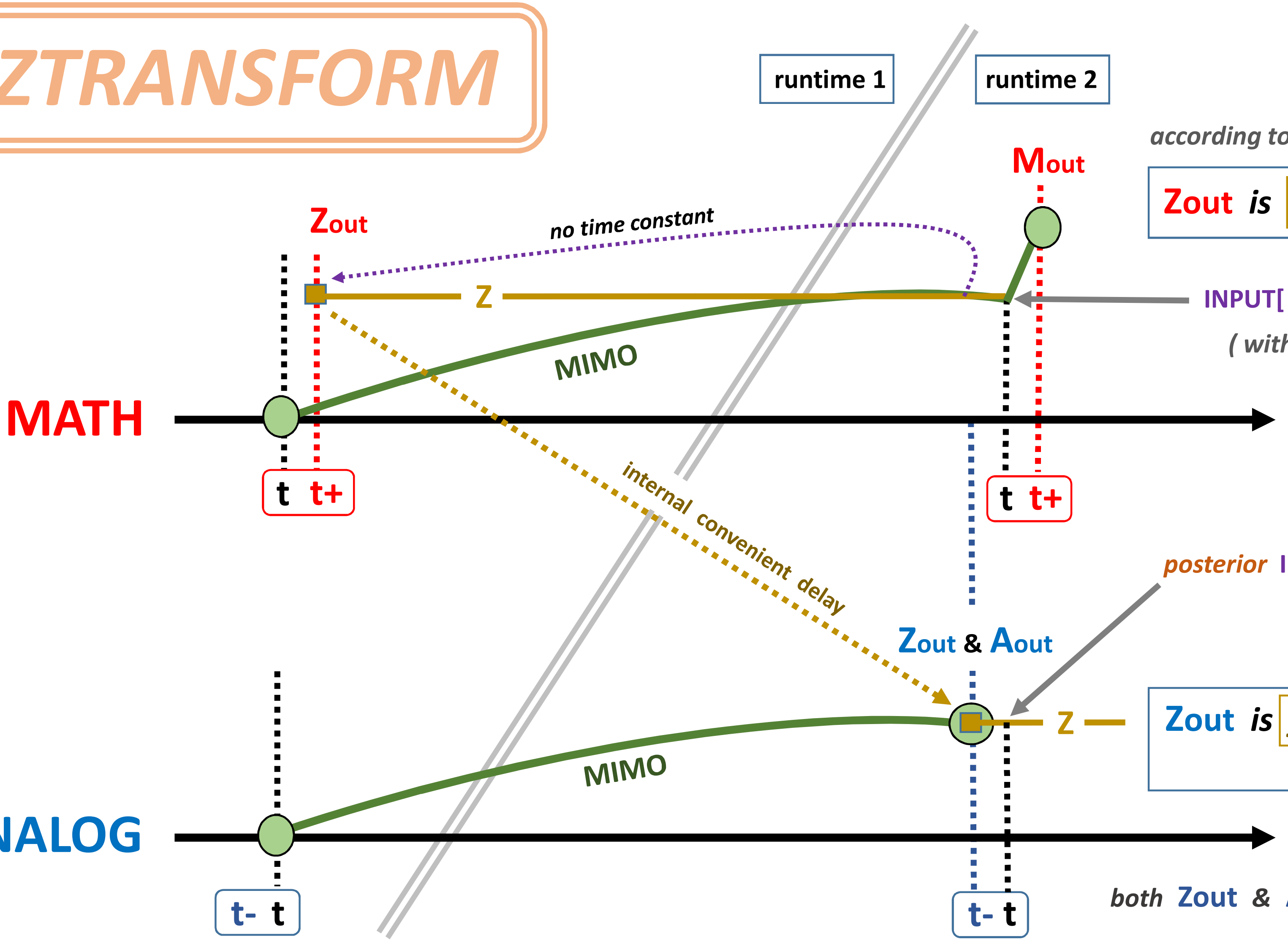


position

f(*step size*)

percussion (L C ⇨ state change)

ZTRANSFORM



according to no time constant equations

Zout is antidated versus Mout

INPUT[t] is taken into account

(with possible percussions)

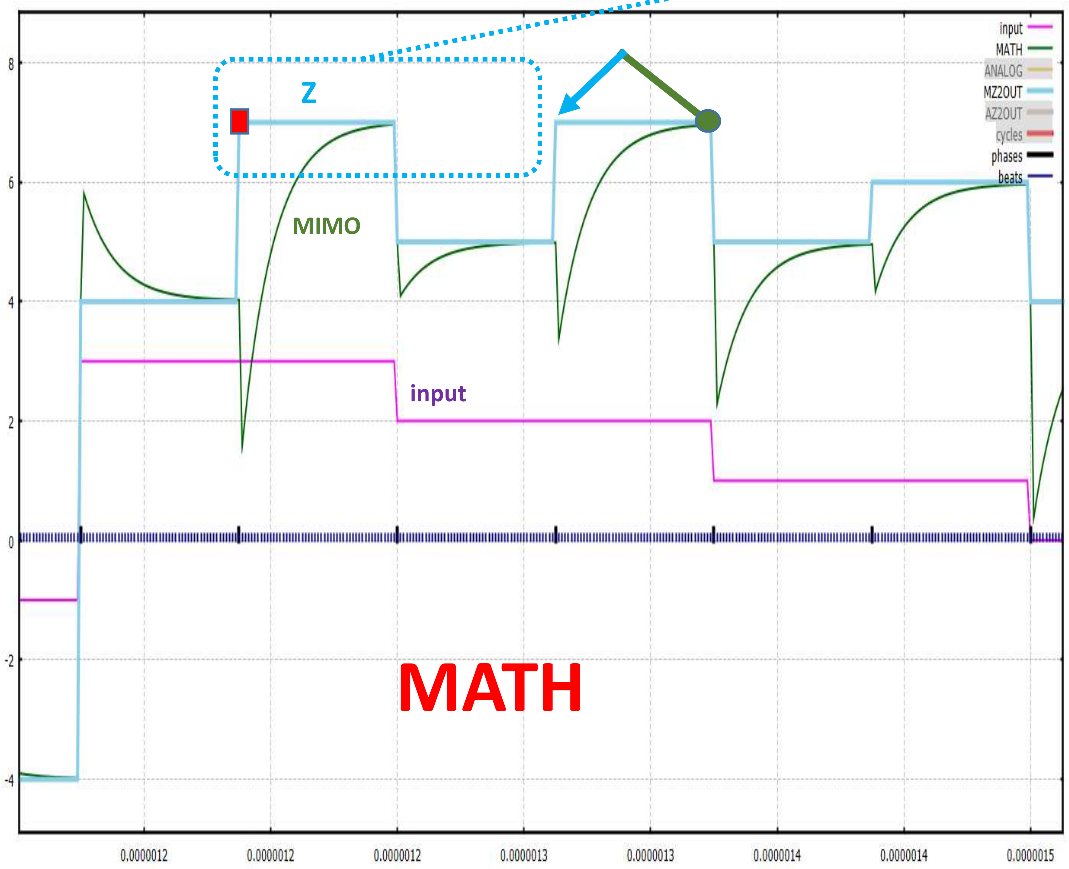
posterior INPUT[t]

Zout is forced concomitant with Aout

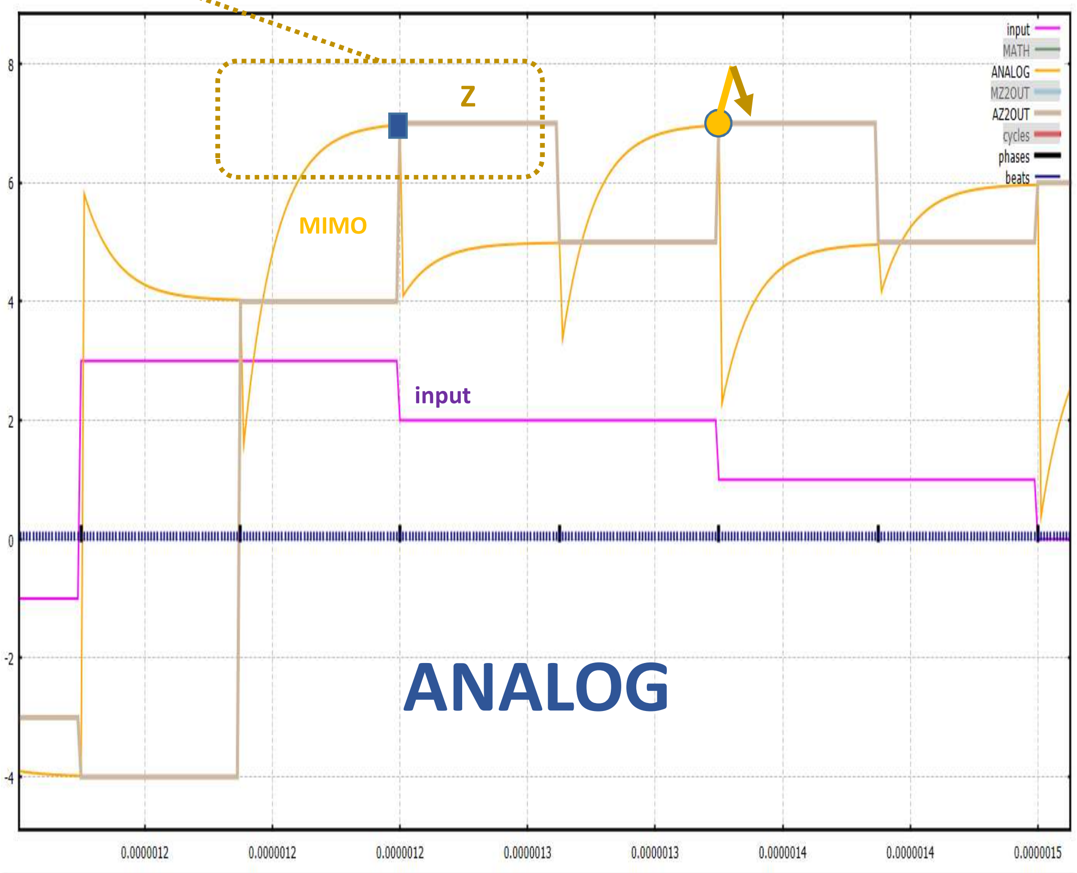
(to make waveforms analysis easier)

both Zout & Aout are postdated versus INPUTS

Z *earlier than* **MIMO**



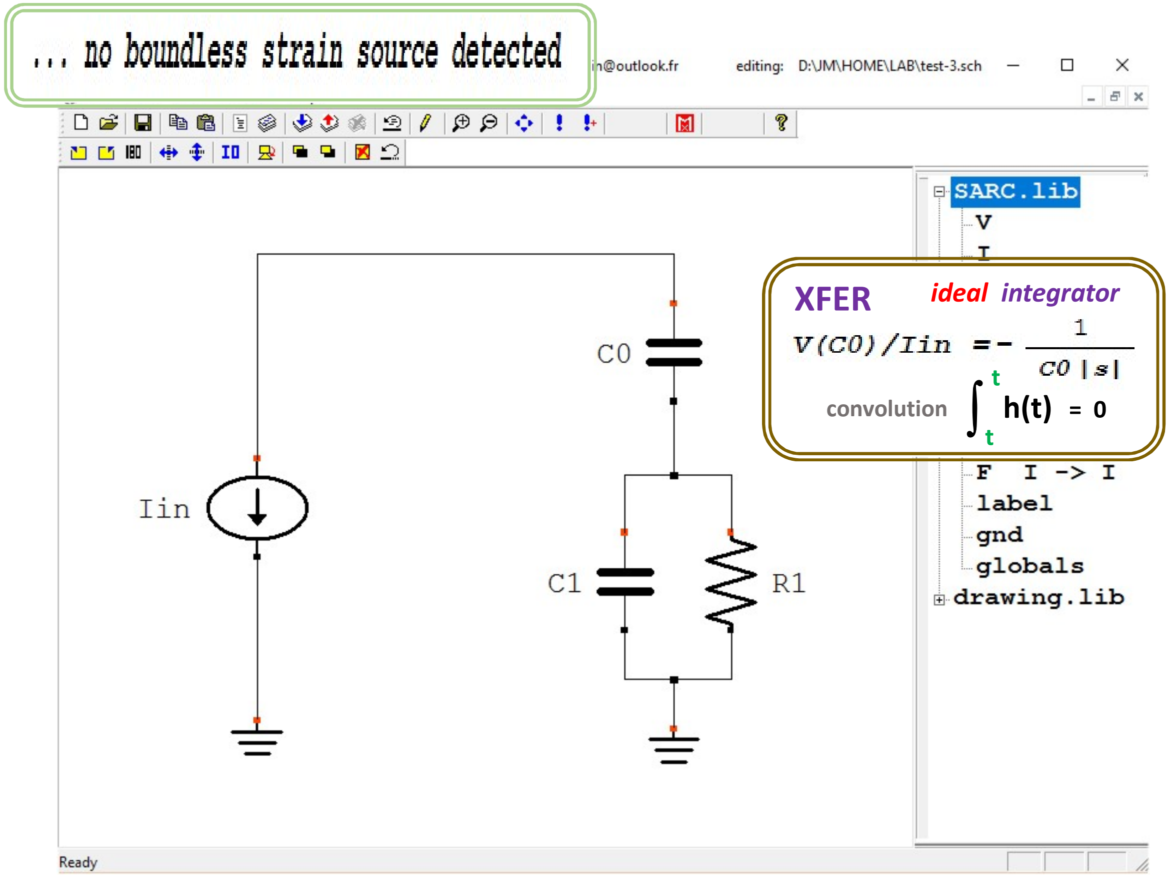
Z & **MIMO** *concomitant*



runtime WARNINGS

WARNING

Y = 0



... no boundless strain source detected

...n@outlook.fr editing: D:\JM\HOME\LAB\test-3.sch

XFER *ideal integrator*

$$V(C0)/Iin = -\frac{1}{C0\,|s|}$$

convolution $\int_{t}^{t} h(t) = 0$

conversion **MIMO** ⇨ **XFER**

$$1 \ \text{warning}\ V(C0)/Iin = 0$$

$$2 \ \text{proper}\ V(C1)/Iin = -\frac{R1}{C1\,R1\,|s| + 1}$$

MIMO simulation

```
### Warning : OUTPUT( V(C0) ) = 0
```

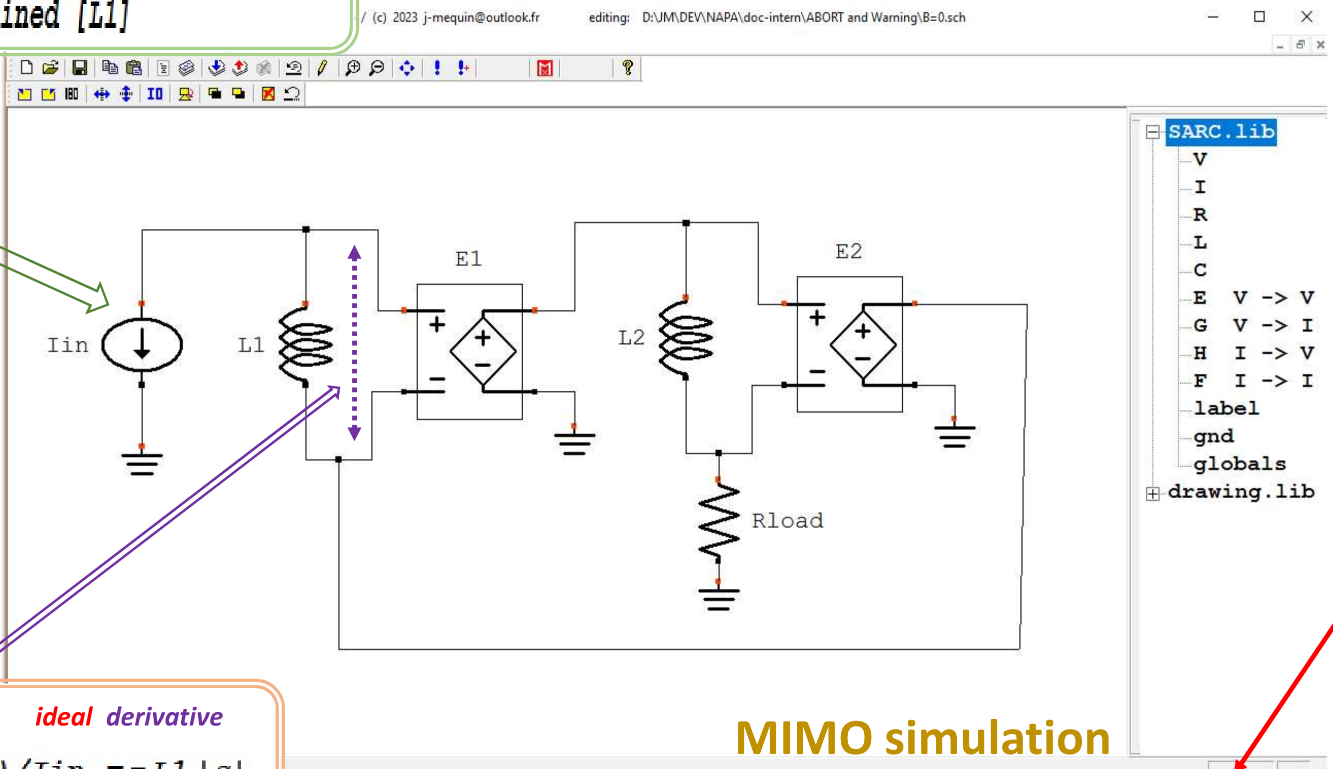
please note : the fact that **V(C0)** *is flagged does not prevent* **V(C1)** *to be correct*

WARNING

B = 0

```
### Warning : boundless strain sources [Iin]
==> LC boundless strained [L1]
```

boundless source


/ (c) 2023 j-mequin@outlook.fr editing: D:\JM\DEV\NAPA\doc-intern\ABORT and Warning\B=0.sch
Iin
L1
E1
L2
E2
Rload
SARC.lib
V
I
R
L
C
E V -> V
G V -> I
H I -> V
F I -> I
label
gnd
globals
drawing.lib

$$STATES = \left[I(L2) \right]$$

A B C D

$$A = \left[-\frac{Rload}{L2} \right]$$

$$B = \left[0 \right]$$

$$C = \begin{bmatrix} 1 \\ -E1\ Rload \end{bmatrix}$$

$$D = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

XFER *ideal derivative*

improper $cV(E1)/Iin = -L1\ |s|$

MIMO simulation

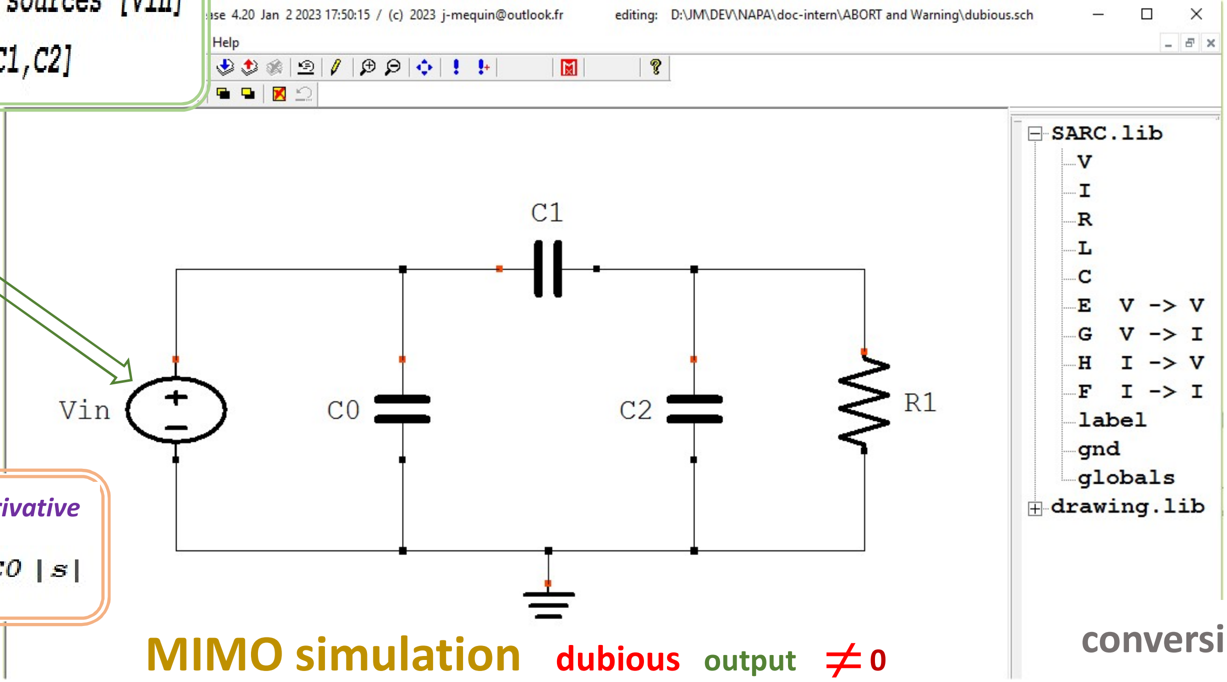
```
### Warning : OUTPUT( cV(E1) ) bad since B = 0
### Warning : OUTPUT( I(Rload) ) bad since B = 0
```

WARNING

$Y \neq 0$

```
### Warning : boundless strain sources [Vin]
==> LC boundless strained [C0,C1,C2]
```

boundless source



ase 4.20 Jan 2 2023 17:50:15 / (c) 2023 j-mequin@outlook.fr editing: D:\JM\DEV\NAPA\doc-intern\ABORT and Warning\dubious.sch
Help
C1
Vin
C0
C2
R1
SARC.lib
V
I
R
L
C
E V -> V
G V -> I
H I -> V
F I -> I
label
gnd
globals
drawing.lib

XFER *ideal derivative*

$\boxed{improper}\ I(C0)/Vin = C0\,|s|$

MIMO simulation **dubious** output $\neq 0$

```
### Warning : OUTPUT( I(C1) ) D[dubious states]
```

conversion MIMO ➡ XFER

$\boxed{biproper}\ \dfrac{C1^{2}\,|s|}{(C2+C1)^{\boxed{2}}\,R1\,|s|+C2+C1}$

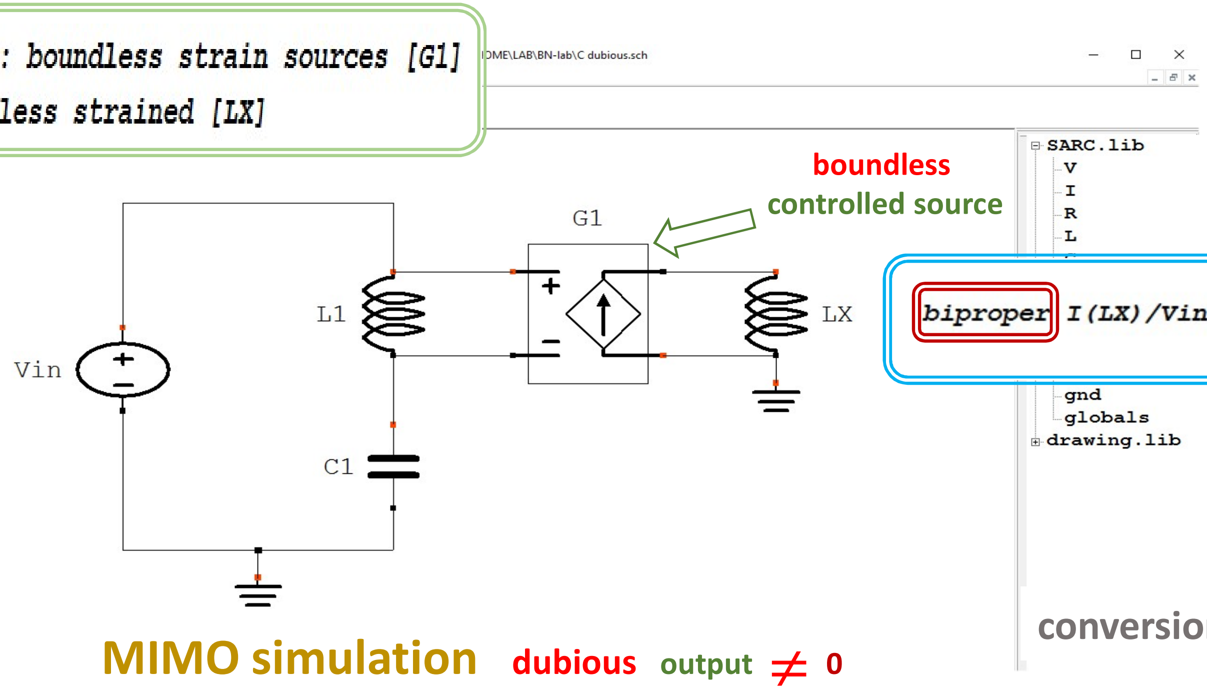
$\neq$

BUT corresponding XFER $\boxed{improper}\ I(C1)/Vin = \dfrac{C1\,C2\,R1\,|s|^{2}+C1\,|s|}{(C2+C1)\,R1\,|s|+1}$

WARNING

$Y \neq 0$

```
### Warning : boundless strain sources [G1]
==> LC boundless strained [LX]
```



boundless **controlled source**

biproper $I(LX)/Vin = \dfrac{C1\ G1\ L1\ |s|^{2}}{C1\ L1\ |s|^{2}+1}$

conversion **MIMO** ⇨ **XFER**

MIMO simulation **dubious** **output** $\neq$ **0**

```
### Warning : OUTPUT( V(LX) ) C[dubious states]
```

proper $-\dfrac{G1\ LX\ |s|}{C1\ L1\ |s|^{2}+1}$

$\neq$

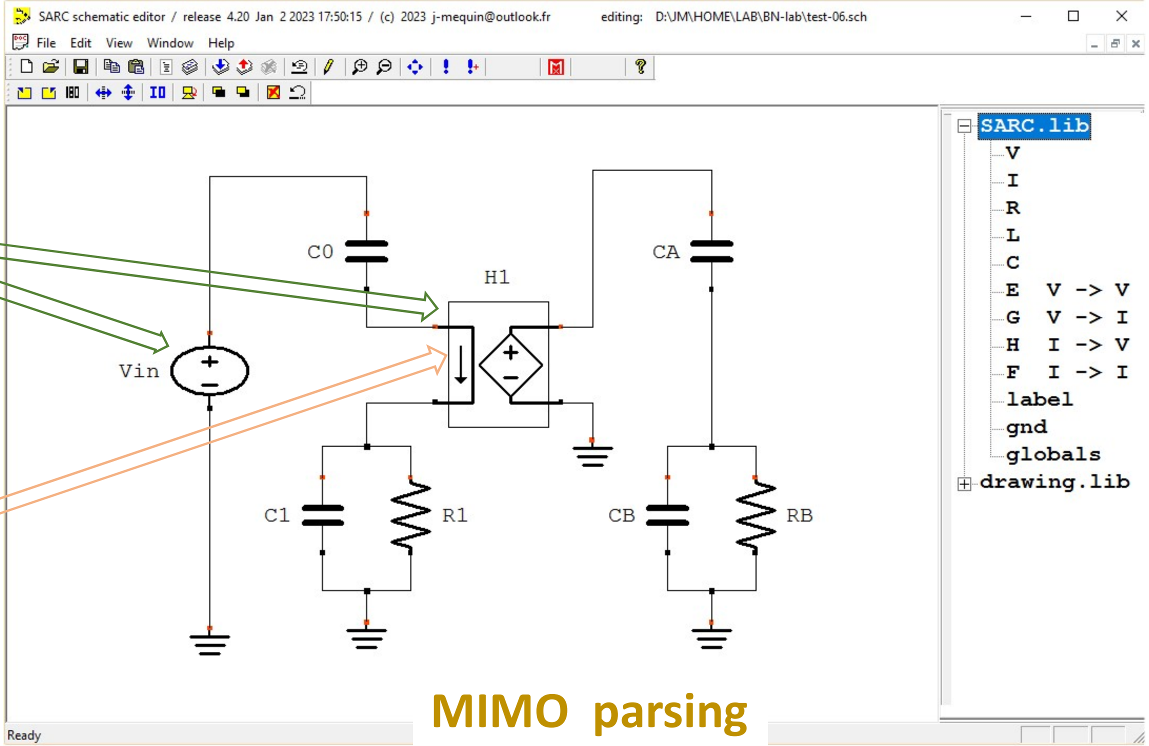
BUT corresponding **XFER** improper $V(LX)/Vin = \dfrac{C1\ G1\ L1\ LX\ |s|^{3}}{C1\ L1\ |s|^{2}+1}$

BAD SCHEMATIC

ABORT

```
### Warning : boundless strain sources [Vin,H1]
==> LC boundless strained [C0,C1,CA,CB]
```

boundless sources



SARC schematic editor / release 4.20 Jan 2 2023 17:50:15 / (c) 2023 j-mequin@outlook.fr editing: D:\JM\HOME\LAB\BN-lab\test-06.sch

XFER

$$\boxed{\mathit{improper}}\ cI(H1)/Vin = \frac{C0\,C1\,R1\,|s|^{2} + C0\,|s|}{(C1+C0)\,R1\,|s|+1}$$

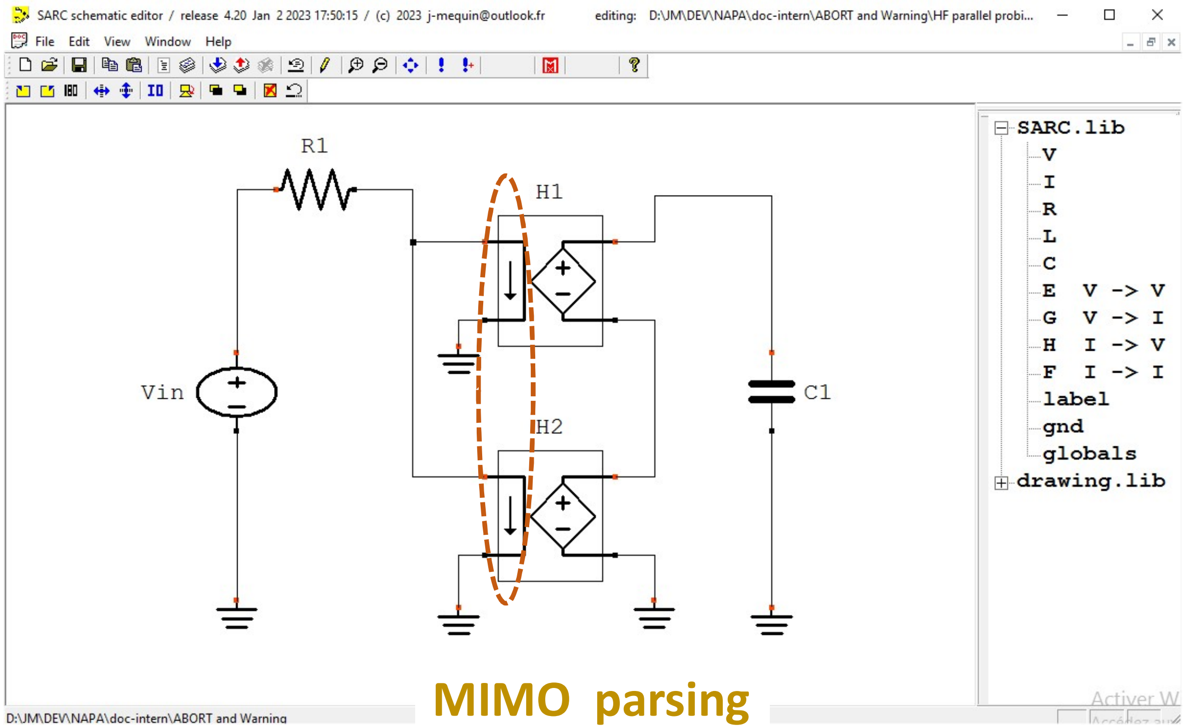
MIMO parsing

$$\left[\left[\overset{U}{Vin}, \overset{X}{CA}, \frac{C0\,C1\,CB\,H1\,\boxed{|s|}}{(C1+C0)\,(CB+CA)}\right], \left[\overset{U}{Vin}, \overset{X}{CB}, \frac{C0\,C1\,CA\,H1\,\boxed{|s|}}{(C1+C0)\,(CB+CA)}\right]\right]$$

```
*** ERROR: improper STATES ["CA","CB"]
```

BAD SCHEMATIC ABORT

schematic may look good but may not be simulable


SARC schematic editor / release 4.20 Jan 2 2023 17:50:15 / (c) 2023 j-mequin@outlook.fr
editing: D:\JM\DEV\NAPA\doc-intern\ABORT and Warning\HF parallel probi...
File
Edit
View
Window
Help
R1
H1
H2
Vin
C1
SARC.lib
V
I
R
L
C
E V -> V
G V -> I
H I -> V
F I -> I
label
gnd
globals
drawing.lib
D:\JM\DEV\NAPA\doc-intern\ABORT and Warning

MIMO parsing

```
*** ERROR: parallel CURRENT probing for CONTROL side [["H1","H2"]]
```