SYNTAX of a Maxima file describing a Linear Circuit *to produce its transfer functions*

this module has been written by Jacques Mequin to compute transfer functions
and to create the matrices used in time domain simulations using SARC techniques.

Use 'Alt m' or tool 'Maxima Run' from the Crimson Editor to process the Maxima file.


Gustav Robert Kirchhoff
1824-1887
Prof. Dr., Physiker.
Arbeiten auf dem Gebiet der Spektralanalyse
(mit Bunsen), der Elektrizitätslehre und der
Strahlungstheorie (Kirchhoffsche Regeln 1845,
Kirchhoffsches Strahlungsgesetz 1859).

A "netlist" is a set of COMPONENTS described in a table :

```
NETLIST( component1, component2, ... ) ;
```

each "component" is defined as a *[type, instance_name, node_id, ...]*

It is important to note that the INSTANCE NAME will also represent the VALUE of the component.

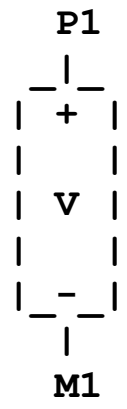
type	V, I, E, G, H, F, R, L or C
instance_name	
node_id	"gnd", GND, Gnd ... meaning "ground"

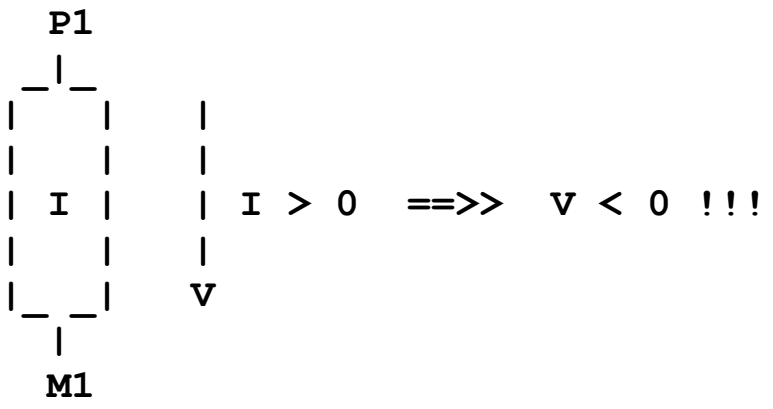
The syntax of the components of this netlist is described here below.

Place a minus sign '-' in front of an instance name to force a swap of its node connections. For controlled sources, only the source side will be permuted.

In netlist,

Independent Sources can be referenced in output list as V(<id>)
or I(<id>)



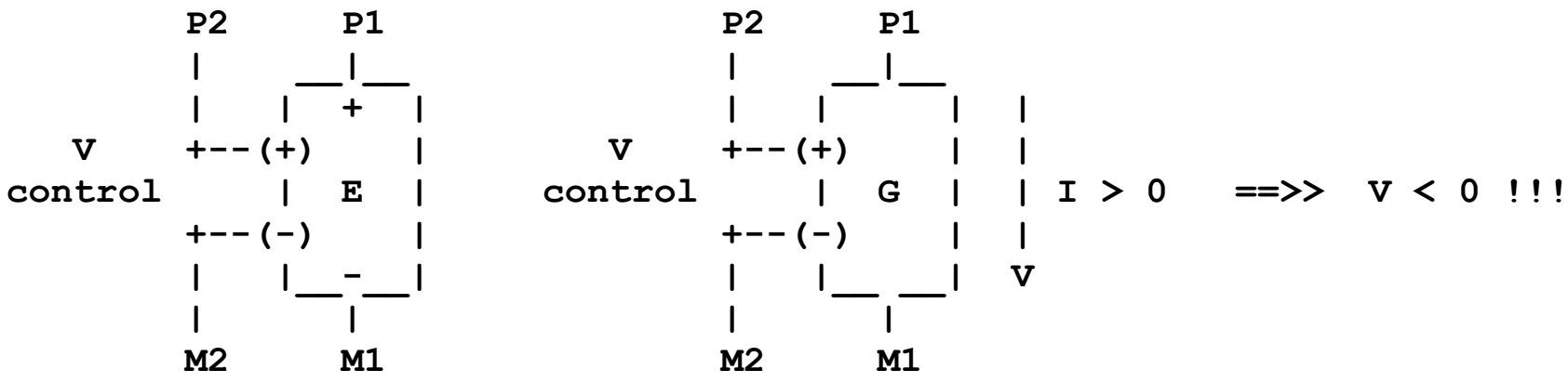



```
[ V,  id, P1, M1 ] ;      voltage source
[ I,  id, P1, M1 ] ;      current source
```

and also `[ V, -id, P1, M1 ]` *equivalent to* `[ V, id, M1, P1 ]`
`[ I, -id, P1, M1 ]` *equivalent to* `[ I, id, M1, P1 ]`

In netlist,

VOLTAGE Controlled Sources can be referenced in output list as V(<id>)
or I(<id>)
and the V control as cV(<id>)

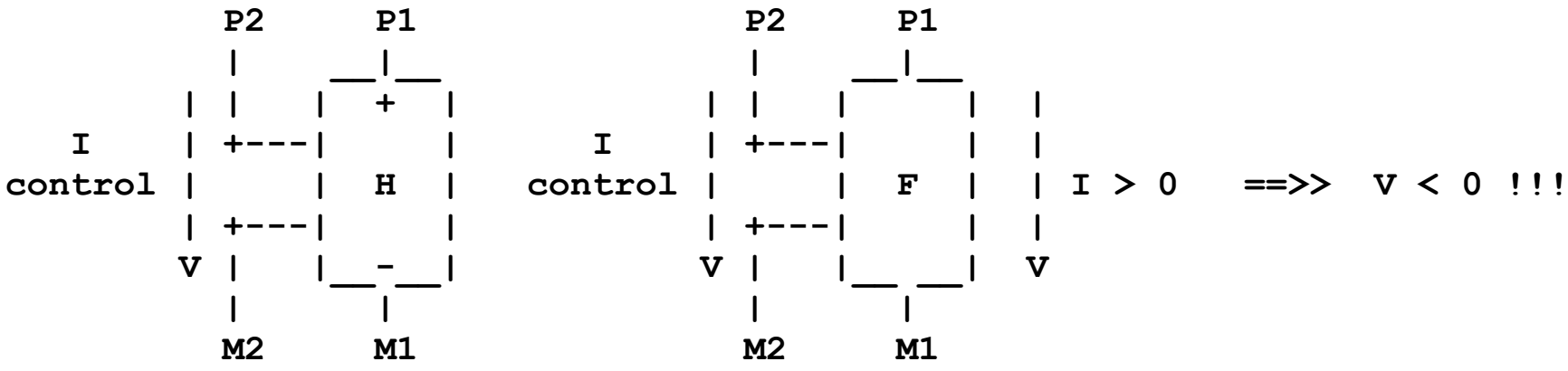


[E, id, P1, M1, P2, M2] ; voltage controlled voltage source
[G, id, P1, M1, P2, M2] ; voltage controlled current source

and also *[E, -id, P1, M1, P2, M2] equivalent to [E, id, M1, P1, P2, M2]*
[G, -id, P1, M1, P2, M2] equivalent to [G, id, M1, P1, P2, M2]

In netlist,

CURRENT Controlled Sources can be referenced in output list as V(<id>)
or I(<id>)
and the I control as cI(<id>)

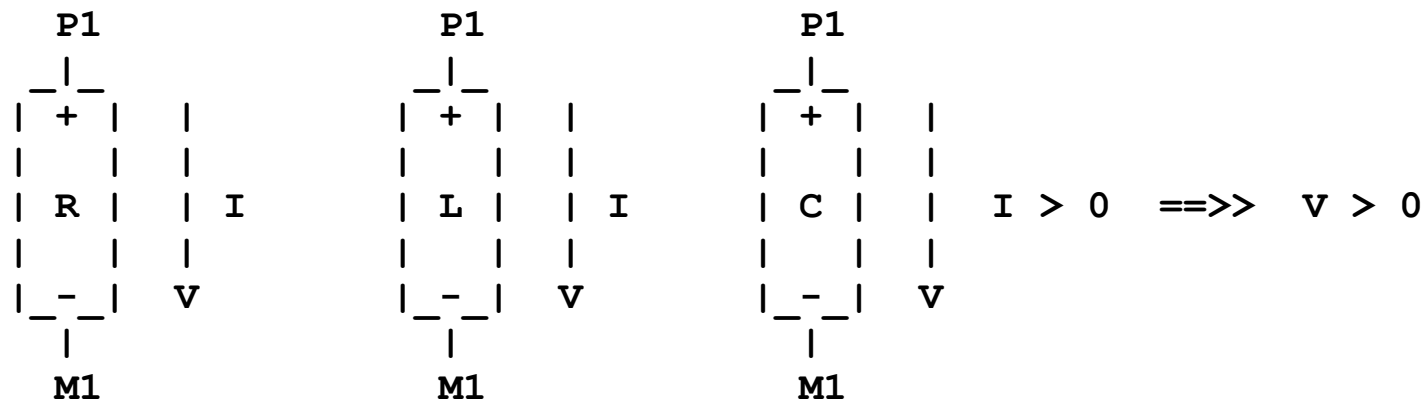


[H, id, P1, M1, P2, M2] ; current controlled voltage source
[F, id, P1, M1, P2, M2] ; current controlled current source

and also *[H, -id, P1, M1, P2, M2] equivalent to [H, id, M1, P1, P2, M2]*
[F, -id, P1, M1, P2, M2] equivalent to [F, id, M1, P1, P2, M2]

In netlist,

Passives Elements can be referenced in output list as V(<id>)
as I(<id>)




```
[ R,  id, P1, M1 ] ;      resistance
[ L,  id, P1, M1 ] ;      inductance
[ C,  id, P1, M1 ] ;      capacitance
```

and also
```
[ R, -id, P1, M1 ]
[ L, -id, P1, M1 ]
[ C, -id, P1, M1 ]
```

As the INSTANCE NAME represents also the VALUE of the component and because the instance name is by definition unique, it is necessary to use the function AKA() to attribute the same value to several instances.

Alias values of the netlist with the function AKA():

```
AKA( replacement_variable , xxx , ... ) ;
```

where xxx is a variable to be eliminated from the set of parameters and replaced by the replacement variable.

```
AKA( replacement_variable , xxx , ... , value ) ;
```

where xxx is a variable to be eliminated from the set of parameters and replaced by a numerical value.

```
Examples:       AKA( RA , R1 , R2 , R3 ) ;
                AKA( RX , RB, RC, RD, "1000.0" ) ;
```

>>> To define a CONDUCTANCE, write AKA(1/GA , R1 , R2 , R3) ;

Some voltage or current sources may be considered as parameters and not as inputs and therefore do not need to appear in the list of transfer functions:

```
INTERN( xxx , ... ) ;
```

where xxx is the ID of q voltage or current source.

There is a special instruction to suggest to order a list to determine state variables. It may help to improve a little the precision of the computations.

This instruction should be placed before 'GENERATE_MIMO()'. This instruction is optional.

```
XWISH( Cxxx, Lyyy, ... ) ;
```

There is also a print facility :

```
PRINTPOLES(num) ;
```

This function triggers at most 'num' times the print of the poles in 'stderr' (or logfile if activated) when recomputing matrices at change of R, L or V values.

To document the file, a schematic could be added.

A schematic is a 'WYSIWYG' string of alphanumeric characters :

```
SCHEMATIC( "......
.....................
....................." ) ;
```

Schematic is optional and is used to document the header file generated by GENERATE_MIMO( ) function. This schematic will be transferred automatically for processing by NAPA as a WYSIWIG comment.

Outputs are listed as a sequence :

```
OUTPUTS( xxx, ... ) ;
```

where 'xxx' could be :

```
V(<n>)    to get the voltage of the node <n>

V(<id>)   a voltage output of a component <id>
I(<id>)   a current output of a component <id>
W(<id>)   a power   output of a component <id>

cV(<id>)  the control of a voltage controlled source <id>
cI(<id>)  the control of a current controlled source <id>
```

Placing a minus sign "-" in front of an OUTPUT component name will invert its value : example V(-R2)

Other outputs

'xxx' could be also

eV(<n>)	interpolation error on node <n>
eI(<id>)	interpolation error on a component <id> output
ecV(<id>)	interpolation error on a component <id> output
ecI(<id>)	interpolation error on a component <id> output
mV(<n>)	maximum interpolation error on node <n>
mI(<id>)	maximum interpolation error on a component <id> output
mcV(<id>)	maximum interpolation error on a component <id> output
mcI(<id>)	maximum interpolation error on a component <id> output
J() ..	the energy stored in L and C elements of the module

Currently 2 output functions are available :

for ILT : **GENERATE_XFER( ) ;** **(*use with duser function 'ilt'*)**

This function produces the transfer function(s) on screen.

After the execution of the function GENERATE_XFER(), you can access in the opened Maxima session to a computed transfer function as

XFER[n]

where 'n' is the number printed in front of the response.

for SARC : **GENERATE_MIMO( ) ;** **(*use with duser function 'sarc'*)**

This function generates a header file which wiil have to be referenced in the NAPA file to run the simulation.

A MAXIMA FILE EXAMPLE : '*SK_filter.mac*' to produce header file '*SK_filter.hdr*'

```
SCHEMATIC( "
              o-------o--------o-------o-------o-----o
              |       |        |       |       |
             [c]      o----->  |       |       |
              |                [gm]    |     [co]
i----i--[r]--s--[r]--b----->   |       |       |
     |                |        |       |       |
   [vin]             [c]       |     [rds]     |
     |                |        |       |       |
   _gnd_            _gnd_    _gnd_   _gnd_   _gnd_    " ) ;

NETLIST(  [ V ,  vin ,  i ,  gnd          ],
          [ R ,   r1 ,  i ,    s          ],
          [ C ,   c2 ,  s ,    o          ],
          [ R ,   r3 ,  s ,    b          ],
          [ C ,   c4 ,  b ,  gnd          ],
          [ G ,   gm ,  o ,  gnd,  o,  b ],
          [ R ,  rds ,  o ,  gnd          ],
          [ C,    co ,  o ,  gnd          ] ) ;

OUTPUTS( V(o), V(b), I(r1), I(vin), J() ) ;
AKA( r, r1, r3 ) ;  AKA( c, c2, c4 ) ;  AKA( 1/gds, rds ) ;

GENERATE_MIMO( ) ;    /* output header file for NAPA 'sarc' function */
```

APPENDIX

The file **'maxima-init.mac'** used by Maxima at launch has been edited to get access to the functions implemented in directory **'/Simulate/MaximaDos/MACfiles'**

```
declare( "|", alphabetic ) ;

setup_autoload ( "SARCENV.mac", SCHEMATIC, NETLIST           ) ;
setup_autoload ( "SARCENV.mac", OUTPUTS, AKA, AKA2NUM, XWISH ) ;
setup_autoload ( "SARCENV.mac", PRINTPOLES    ) ;
setup_autoload ( "SARCENV.mac", GENERATE_XFER, GENERATE_MIMO ) ;

/* etc … */
```