1 Author

Symbolic Analysis of Linear Electric Circuits with Maxima
SALECx version 1.0 (2019-08-26) for Maxima 5.38+, wxMaxima 16+

Dr. Dejan Tošić, Full Professor, tosic@etf.rs
University of Belgrade -- School of Electrical Engineering
11000 Belgrade, Serbia

2 License

3 Acknowledgement

I thank Prof. Dr. Predrag Pejović for permanent encouragement and valuable discussions related to this project.

4 Presented and Published

Application of Free Software and Open Hardware,
PSSOH 2019, International Conference,
University of Belgrade -- School of Electrical Engineering,
Belgrade, Serbia, October 26, 2019. http://pssoh.etf.bg.ac.rs/

5 SALECx in a Nutshell

SALECx is a Maxima program for solving linear time-invariant finite electric circuits in the complex domain of the Unilateral Laplace Transform or Phasor Transform.

5.1 Algorithm

SALECx uses Modified Nodal Analysis (MNA) to formulate equations and solve circuits.

One node, referred to as the reference node is labeled by zero, 0. Other nodes are labeled by consecutive integers starting from one, 1.

For all nodes except the reference node, Node 0, SALECx formulates the Kirchhoff's current law (KCL) equations. The reference direction for current is OUT OF the node (leaving the node).

The currents are expressed in terms of node voltages.
The node voltage of the reference node is set to zero, 0.

If a current cannot be expressed in terms of node voltages then the current becomes a MNA variable and the corresponding element equation is added to the system of the MNA equations.

MNA variables are node voltages, V[1], V[2], V[3], ... and currents of the ports which are not voltage controlled, i.e. the currents which cannot be expressed in terms of node voltages. These currents are labeled by I["id"] or I["id",pin] where "id" uniquely specifies a circuit element and pin stands for an integer assigned to a circuit node, 1, 2, 3, ...

5.2 Reserved symbols

s -- complex frequency, the Laplace variable [radian/second]

I -- MNA current variables
I[label] or I[label,node]

V -- MNA voltage variables, node voltages
V[1], V[2], V[3] ...
V[0] is set to 0

5.3 Units

All quantities are assumed to be in SI units, the International System of Units (SI), adopted by the General Conference on Weights and Measures in 1960.

5.4 Electric Circuit Specification

The circuit to be analyzed is specified as a list
[circuitElement_1, circuitElement_2 ..., circuitElement_N].

A circuit element is specified as a list of the form
```
[type, label, a, b, p]
[type, label, a, b, p, IC]
[type, label, [a1,a2], b]
[type, label, [a1,a2], [b1,b2], p]
[type, label, [a1,a2], [b1,b2], p, IC]
```

type -- string that specifies the element type:
"R", "L", "C", "I", "V", "Z", "Y", "OpAmp",
"VCVS", "VCCS", "CCCS", "CCVS", "IT", "K", "T".

label -- string that uniquely identifies circuit element, e.g.
"Vgen", "Isource", "Rin", "Cfb", "Lprimary", "Zload", etc.

```
one-port element
 a -- positive terminal
 b -- negative terminal
```

```
two-port element
 a1 -- positive terminal of the 1st port
 a2 -- negative terminal of the 1st port
 b1 -- positive terminal of the 2nd port
 b2 -- negative terminal of the 2nd port
```

```
p -- parameter or parameters if p is list
```

```
IC -- initial conditions at 0-minus
 Vo for capacitors
 Io for inductors
 [Io1,Io2] for linear inductive transformers
```

5.5 Element Catalog

5.6 One-port elements

```
Resistor
["R", "id", plusTerminal, minusTerminal, resistance]
```

```
Inductor
["L", "id", plusTerminal, minusTerminal, inductance]
["L", "id", plusTerminal, minusTerminal, inductance, Io]
Io -- initial condition, initial current at 0-minus
from plusTerminal, across the element, to minusTerminal
```

```
Capacitor
["C", "id", plusTerminal, minusTerminal, capacitance]
["C", "id", plusTerminal, minusTerminal, capacitance, Vo]
Vo -- initial condition, initial voltage at 0-minus
Vo = V[plusTerminal] - V[minusTerminal]
```

```
Current source (ideal independent current generator)
["I", "id", plusTerminal, minusTerminal, excitation]
excitation is the source (generator) current
from plusTerminal, across the element, to minusTerminal
```

```
Voltage source (ideal independent voltage generator)
["V", "id", plusTerminal, minusTerminal, excitation]
excitation is the source (generator) voltage
voltage = V[plusTerminal] - V[minusTerminal]
```

```
Impedance
["Z", "id", plusTerminal, minusTerminal, impedance]
```

```
Admitance
["Y", "id", plusTerminal, minusTerminal, admittance]
```

5.7 Operational Amplifier

```
Operational Amplifier (Ideal OpAmp)
["OpAmp", "id", [nonInvertingTerminal, invertingTerminal], outputTerminal]
I["id"] is current into outputTerminal, MNA current variable
```

5.8 Controlled Sources

```
["VCVS", "id", [plusControllingTerminal, minusControllingTerminal],
 [plusControlledTerminal, minusControlledTerminal], voltageGain]
I["id"] is current into plusControlledTerminal, MNA current variable
```

```
["VCCS", "id", [plusControllingTerminal, minusControllingTerminal],
 [plusControlledTerminal, minusControlledTerminal], transconductance]
```

```
["CCCS", "id", [plusControllingTerminal, minusControllingTerminal],
 [plusControlledTerminal, minusControlledTerminal], currentGain]
I["id"] is current into plusControllingTerminal, MNA current variable
```

```
["CCVS", "id", [plusControllingTerminal, minusControllingTerminal],
 [plusControlledTerminal, minusControlledTerminal], transresistance]
I["id"] is current into plusControlledTerminal, MNA current variable
```

5.9 Transformers

```
Ideal Transformer
["IT", "id", [plusPrimaryTerminal, minusPrimaryTerminal],
 [plusSecondaryTerminal, minusSecondaryTerminal], turnsRatio]
I["id"] is current into plusPrimaryTerminal, MNA current variable
```

```
Linear Inductive Transformer
["K", "id", [plusPrimaryTerminal, minusPrimaryTerminal],
 [plusSecondaryTerminal, minusSecondaryTerminal], [L1,L2,L12]]
["K", "id", [plusPrimaryTerminal, minusPrimaryTerminal],
 [plusSecondaryTerminal, minusSecondaryTerminal], [L1,L2,L12], [Io1,Io2]]
I["id",plusPrimaryTerminal] is
 current into plusPrimaryTerminal, MNA current variable
I["id",plusSecondaryTerminal] is
 current into plusSecondaryTerminal, MNA current variable
```

5.10 ABCD two-port

```
["ABCD", "id", [plusPrimaryTerminal, minusPrimaryTerminal],
 [plusSecondaryTerminal, minusSecondaryTerminal], [[A,B],[C,D]]]
I["id",plusPrimaryTerminal] current into plusPrimaryTerminal
I["id",plusSecondaryTerminal] current OUT OF plusSecondaryTerminal
```

5.11 Transmission lines

```
Transmission Line, Phasor Transform
["T", "id", [plusSendingTerminal, minusSendingTerminal],
 [plusReceivingTerminal, minusReceivingTerminal], [Zc,theta]]
theta [radian] -- electrical length
I["id",plusSendingTerminal] current into plusSendingTerminal
I["id",plusReceivingTerminal] current OUT OF plusReceivingTerminal
```

```
Transmission Line, Laplace Transform
["T", "id", [plusSendingTerminal, minusSendingTerminal],
 [plusReceivingTerminal, minusReceivingTerminal], [Zc,tau]]
tau [second] -- delay (one-way time delay)
I["id",plusSendingTerminal] current into plusSendingTerminal
I["id",plusReceivingTerminal] current into plusReceivingTerminal
```

5.12 Calling SALECx

```
Laplace Transform s-domain
 SALECx[circuitSpecification]
```

```
Phasor Transform j*omega-domain, sinusoidal steady state
 SALECx[circuitSpecification, omegaPhasorTransform]
omegaPhasorTransform [radian] -- angular frequency
```

5.13 Options

```
Return only the response
 SALECxPrint: false
```

```
Return some analysis details and the response
 SALECxPrint: true
```

5.14 Declaration and Initialization

```
Declare complex domain
 domain: complex$
```

```
Remove values of symbols, e.g.
 remvalue(Ig, s, Vg, Z, Yeq)$
```

```
Declare complex variables, e.g.
 declare([Ig, s, Vg, Z, Yeq], complex)$
```

```
Declare real variables, e.g.
 declare([Cload, L12, R, Vgeff, omega1], real)$
```

Declare integer variables, e.g.
 declare(nHarmonic, integer)$

Make assumptions, e.g.
 assume(C > 0, L2 > 0, Vgeff > 0, notequal(m, 0), n > -1)$

Introduce aliases, e.g.
 alias(j, %i)$

5.15 Circuit Graph Assumption

The electric circuit graph is assumed to be connected.

If the graph is not connected then
(1) identify the disconnected components,
(2) choose one node in each component, and
(3) connect the chosen nodes to make the graph connected.

The refence node (ground) is numbered by zero, 0.
The other nodes are numbered by consecutive integers starting from one, 1.

6 References

6.1 Classic

Charles A. Desoer, Ernest S. Kuh,
Basic Circuit Theory, New York, NY, McGraw-Hill, 1969.

Leon O. Chua, Charles A. Desoer, and Ernest S. Kuh,
Linear and nonlinear circuits, New York, NY, McGraw-Hill, 1987.

6.2 General

Charles K. Alexander, Matthew N. O. Sadiku,
Fundamentals of Electric Circuits, 6/e, New York, NY, McGraw-Hill, 2017.

James W. Nilsson, Susan A. Riedel,
Electric Circuits, 10/e, Upper Saddle River, NJ, Prentice Hall, 2015.

J. David Irwin, R. Mark Nelms,
Basic Engineering Circuit Analysis, 11/e, Hoboken, NJ, Wiley, 2015.

James A. Svoboda, Richard C. Dorf,
Introduction to Electric Circuits, 9/e, Hoboken, NJ, Wiley, 2014.

William H. Hayt, Jr., Jack E. Kemmerly, Steven M. Durbin,
Engineering circuit analysis, 8/e, New York, NY, McGraw-Hill, 2012.

Farid N. Najm, Circuit Simulation,
Hoboken, New Jersey, John Wiley & Sons, 2010.

Omar Wing, Classical Circuit Theory,
Springer Science+Business Media, LLC, New York, NY, 2008.

Wai-Kai Chen (Editor),
Circuit Analysis and Feedback Amplifier Theory,
CRC Press, Taylor & Francis Group, Boca Raton, FL, 2006.

6.3 Power Engineering

Arieh L. Shenkman, Transient Analysis of Electric Power Circuits Handbook,
Springer, Dordrecht, The Netherlands, 2005.

Arieh L. Shenkman, Circuit Analysis for Power Engineering Handbook,
Springer, Dordrecht, The Netherlands, 1998.

6.4 Transmission Lines

Paul R. Clayton, Analysis of Multiconductor Transmission Lines, 2/e,
Hoboken, NJ, Wiley IEEE Press, 2008.

7 ElementStamp (subprogram)

8 SALECx (main program)

```
(%i2)   SALECx(circuit_, [w_]) := block([i_, n_],
         if w_=[] then PhasorTransform_: false
                  else PhasorTransform_: true,

         if w_#[] then
          print("Phasor Transform at angular frequency ", first(w_)),
         if w_=[] then remvalue(s)
                  else s: %i*first(w_),

         n_: lmax(flatten(
          map(lambda([x], part(x,[3,4])), circuit_)
         )),

         elementValues_: map(lambda([x],
          if length(x)>4 then part(x,5) else false), circuit_
         ),

         initialConditions_: map(lambda([x],
          if length(x)=6 then part(x,6) else false), circuit_
         ),

         remvalue(I, J, JJ, V, VV),
         for i_: 0 thru n_ do J[i_]: 0,
         JJ: [],
         V[0]: 0,
         potentials_: makelist(V[i_], i_, n_),
         VV: [],

         m_: map(ElementStamp, circuit_),

         equationsVn_: makelist(J[i]=0, i, n_),
         equationsMNA_: append(equationsVn_, JJ),

         variablesMNA_: append(potentials_, VV),

         responseMNA_: linsolve(equationsMNA_, variablesMNA_),

         if SALECxPrint then (
         print("Symbolic Analysis of Linear Electric Circuits with Maxima"),
         print("SALECx version 1.0, Prof. Dr. Dejan Tošić, tosic@etf.rs"),
         print("Number of nodes excluding 0 node: ", n_),
         print("Electric circuit specification:", circuit_),
         print("Supported element: ", m_),
         print("Element values: ", elementValues_),
         print("Initial conditions: ", initialConditions_),
         print("MNA equations: ", equationsMNA_),
         print("MNA variables: ", variablesMNA_)
         ),

        responseMNA_) $
```

8.1 SALECxPrint (reserved symbol, verbose option)

```
(%i3)   SALECxPrint: false $
```

9 domain, declare, assume

```
(%i4)   domain: complex$

(%i5)   remvalue(I, J, JJ, s, t, V, VV, omega)$

(%i6)   remvalue(C, C1, C2, Ea, Eb, Ec, g,
         I1, I2, Ig, Io, Io1, Io2,
         L, L1, L2, L12, m,
         R, R1, R2, R3, R4, R5,
         V1, V2, Vg, Vgeff, Vo, Vstep,
         Y1, Y2, Z, Z0, Z1, Z2, Zc,
         theta, thetag, tau)$

(%i7)   declare([Ea, Eb, Ec,
         I, I1, I2, Ig, s,
         V, V1, V2, Vg,
         Y1, Y2, Z, Z0, Z1, Z2],
        complex)$

(%i8)   declare([C, C1, C2, g,
         Io, Io1, Io2,
         L, L1, L2, L12, m,
         R, R1, R2, R3, R4, R5,
         t, Vgeff, Vo, Vstep, Zc,
         omega, theta, thetag, tau],
        real)$

(%i9)   declare(n, integer)$

(%i10)  assume(C > 0, C1 > 0, C2 > 0,
         L > 0, L1 > 0, L2 > 0, L12 > 0,
         notequal(m, 0), n > -1,
         R > 0, R1 > 0, R2 > 0, R3 > 0, R4 > 0, R5 > 0,
         Vgeff > 0, Vstep > 0, Zc > 0, omega > 0, tau > 0)$

(%i11)  alias(j, %i)$
```

10 The Simplest Circuit

```
(%i12)  Vg_Shema: [ ["V", "Vgen", 1, 0, Vg] ]$

(%i13)  Vg_Response: SALECx(Vg_Shema);
```

(Vg_Response) $[V_1 = Vg, I_{Vgen} = 0]$

11 Ig Simple Circuit

```
(%i14)  IgR_Shema: [ ["I", "Igen", 0, 1, Ig],
                     ["R", "R", 1, 0, R] ]$
```

```
(%i15) IgR_Response: SALECx(IgR_Shema);
```

(IgR_Response) $[V_1 = Ig\ R]$

12 Vg Simple Circuit

```
(%i16) VgR_Shema: [ ["V", "Vgen", 1, 0, Vg],
                    ["R", "R", 1, 0, R] ]$
```

```
(%i17) VgR_Response: SALECx(VgR_Shema);
```

(VgR_Response) $[V_1 = Vg, I_{Vgen} = -\frac{Vg}{R}]$

13 Capacitor Simple Circuit

```
(%i18) VgRCVo_Shema: [
        ["V", "Vgen", 1, 0, Vg],
        ["R", "R", 1, 2, R],
        ["C", "C", 2, 0, C, Vo]];
```

(VgRCVo_Shema) $[[V, Vgen, 1, 0, Vg], [R, R, 1, 2, R], [C, C, 2, 0, C, Vo]]$

```
(%i19) VgRCVo_Response_PT: SALECx(VgRCVo_Shema, omega);
```

Phasor Transform at angular frequency ω

(VgRCVo_Response_PT) $[V_1 = Vg, V_2 = \frac{Vg}{\%i\ C\ R\ \omega + 1}, I_{Vgen} = -\frac{\%i\ C\ Vg\ \omega}{\%i\ C\ R\ \omega + 1}]$

```
(%i20) V2PT: V[2], VgRCVo_Response_PT;
```

(V2PT) $\frac{Vg}{\%i\ C\ R\ \omega + 1}$

```
(%i21) VgRCVo_Response: SALECx(VgRCVo_Shema),
        SALECxPrint: true;
```

Symbolic Analysis of Linear Electric Circuits with Maxima
SALECx version 1.0, Prof. Dr. Dejan Tošić, tosic@etf.rs
Number of nodes excluding 0 node: 2
Electric circuit specification: $[[V, Vgen, 1, 0, Vg], [R, R, 1, 2, R], [C, C, 2, 0, C, Vo]]$
Supported element: [true, true, true]
Element values: $[Vg, R, C]$
Initial conditions: [false, false, Vo]
MNA equations: $[\frac{V_1 - V_2}{R} + I_{Vgen} = 0, V_2\ C\ s - C\ Vo + \frac{V_2 - V_1}{R} = 0, V_1 = Vg]$
MNA variables: $[V_1, V_2, I_{Vgen}]$

(VgRCVo_Response) $[V_1 = Vg, V_2 = \frac{C\ R\ Vo + Vg}{C\ R\ s + 1}, I_{Vgen} = -\frac{C\ Vg\ s - C\ Vo}{C\ R\ s + 1}]$

```
(%i22) V2s: V[2], VgRCVo_Response, Vg=Vstep/s;
```

(V2s) $$\frac{\frac{Vstep}{s}+C\,R\,Vo}{C\,R\,s+1}$$

```
(%i23) v2ilt: ilt(V2s,s,t), expand;
```

(v2ilt) $$-Vstep\,\%e^{-\frac{t}{C\,R}}+Vo\,\%e^{-\frac{t}{C\,R}}+Vstep$$

```
(%i24) v2t: factorout(v2ilt,Vstep,Vo);
```

(v2t) $$(Vo-Vstep)\,\%e^{-\frac{t}{C\,R}}+Vstep$$

14 Ideal Transmission Line with Zc and tau, Laplace Transform, s-domain

```
(%i25) TLineZc_Shema: [
        ["V", "Vgen", 3, 0, Vg],
        ["R", "R1", 3, 1, Zc],
        ["T", "TL", [1,0], [2,0], [Zc,tau]],
        ["R", "R2", 2, 0, Zc]
       ]$
```

```
(%i26) TLineZc_Response: SALECx(TLineZc_Shema);
```

(TLineZc_Response) $$\left[V_1=\frac{Vg}{2},V_2=\frac{Vg\,\%e^{-s\,\tau}}{2},V_3=Vg,I_{TL,2}=-\frac{Vg\,\%e^{-s\,\tau}}{2\,Zc},I_{TL,1}=\frac{Vg}{2\,Zc},I_{Vgen}=-\frac{Vg}{2\,Zc}\right]$$

I["TL",1] is current into transmission line pin 1 and
I["TL",2] is current into transmission line pin 2.

```
(%i27) V1s: V[1], TLineZc_Response;
```

(V1s) $$\frac{Vg}{2}$$

```
(%i28) V2s: V[2], TLineZc_Response;
```

(V2s) $$\frac{Vg\,\%e^{-s\,\tau}}{2}$$

```
(%i29) V[1]/I["TL",1], TLineZc_Response;
```

(%o29) Zc

```
(%i30) V[2]/I["TL",2], TLineZc_Response;
```

(%o30) $-Zc$

**Zc [Ohm] is characteristic impedanse of transmission line.
tau [second] is transmission line one-way time delay.
tau = D/v = D*sqrt(Lprim*Cprim) = D/(KVF*c0),
D [meter] is length, KVF is velocity factor.
c0 = 299792458 [meter/second]**

**Maxima does not have rules and patterns, yet,
to compute the Inverse Laplace Transform of V2s:**

```
(%i31) ilt(V2s,s,t);
```

(%o31) $\mathrm{ilt}\left(\frac{Vg\,\%e^{-s\,\tau}}{2}, s, t\right)$

15 Ideal Transmission Line with Zc and theta, at omega, Phasor Transfor, j*omega domain

```
(%i32) TLineZc_PT_Shema: [
         ["V", "Vgen", 3, 0, Vg],
         ["R", "R1", 3, 1, Zc],
         ["T", "TL", [1,0], [2,0], [Zc,theta]],
         ["R", "R2", 2, 0, Zc]
       ]$
```

```
(%i33) TLineZc_PT_Response: SALECx(TLineZc_PT_Shema, omega);
```

Phasor Transform at angular frequency ω

(TLineZc_PT_Response) $[V_1=\frac{Vg}{2}, V_2=\frac{Vg}{2\,\%i\,\sin(\theta)+2\cos(\theta)}, V_3=Vg, I_{TL,2}=\frac{Vg}{Zc\,(2\,\%i\,\sin(\theta)+2\cos(\theta))}, I_{TL,1}=\frac{Vg}{2\,Zc}, I_{Vgen}=-\frac{Vg}{2\,Zc}]$

```
(%i34) TLineZc_PT_Response, exponentialize;
```

(%o34) $[V_1=\frac{Vg}{2}, V_2=\frac{Vg\,\%e^{-\%i\,\theta}}{2}, V_3=Vg, I_{TL,2}=\frac{Vg\,\%e^{-\%i\,\theta}}{2\,Zc}, I_{TL,1}=\frac{Vg}{2\,Zc}, I_{Vgen}=-\frac{Vg}{2\,Zc}]$

**I["TL",1] is current into transmission line pin 1, I(z=0), and
I["TL",2] is current *OUT OF* transmission line pin 2, I(z=D).**

```
(%i35) V1w: V[1], TLineZc_PT_Response;
```

(V1w) $\frac{Vg}{2}$

```
(%i36) V2w: V[2], TLineZc_PT_Response, exponentialize;
```

(V2w) $\frac{Vg\,\%e^{-\%i\,\theta}}{2}$

theta [radian] is electrical length.
theta = beta*D = 2*pi*D/lambda, D [meter] is line length.

```
(%i37) V[1]/I["TL",1], TLineZc_PT_Response;
```
(%o37) Zc

```
(%i38) V[2]/I["TL",2], TLineZc_PT_Response;
```
(%o38) Zc

16 OTA-C Filter

Second-order bandpass and lowpass filters:
the single-ended transconductor-C realization.
Rolf Schaumann, Mac E. Van Valkenburg,
Design of Analog Filters,
Oxford University Press, 2001. Figure 16.22, p. 631

```
(%i39) OTA_C_Shema: [
         ["V", "Vgen", 4, 0, Vg],
         ["R", "R", 4, 1, R],
         ["VCCS", "OTA1", [1,0], [2,0], g],
         ["VCCS", "OTA2", [2,0], [2,0], g],
         ["VCCS", "OTA3", [2,0], [3,0], g],
         ["VCCS", "OTA4", [3,0], [0,2], g],
         ["C", "C1", 2, 0, C],
         ["C", "C2", 3, 0, C]
       ]$
```

```
(%i40) OTA_C_Response: SALECx(OTA_C_Shema),
        SALECxPrint: true;
```

Symbolic Analysis of Linear Electric Circuits with Maxima
SALECx version 1.0, Prof. Dr. Dejan Tošić, tosic@etf.rs
Number of nodes excluding 0 node: 4
Electric circuit specification: [[V,$Vgen$,4,0,Vg],[R,R,4,1,R],[$VCCS$,$OTA1$,[1,0],[2,0],g],[$VCCS$,$OTA2$,[2,0],[2,0],g],[$VCCS$,$OTA3$,[2,0],[3,0],g],[$VCCS$,$OTA4$,[3,0],[0,2],g],[C,$C1$,2,0,C],[C,$C2$,3,0,C]]
Supported element: [true,true,true,true,true,true,true,true]
Element values: [Vg,R,g,g,g,g,C,C]
Initial conditions: [false,false,false,false,false,false,false,false]

MNA equations: $\left[\frac{V_1-V_4}{R}=0, V_2\,C\,s-V_3\,g+V_2\,g+V_1\,g=0, V_3\,C\,s+V_2\,g=0, \frac{V_4-V_1}{R}+I_{Vgen}=0, V_4=Vg\right]$

MNA variables: $[V_1, V_2, V_3, V_4, I_{Vgen}]$

(OTA_C_Response) $\left[V_1=Vg, V_2=-\frac{C\,Vg\,g\,s}{C^2\,s^2+C\,g\,s+g^2}, V_3=\frac{Vg\,g^2}{C^2\,s^2+C\,g\,s+g^2}, V_4=Vg, I_{Vgen}=0\right]$

(%i41) Hs2bandpass: V[2]/Vg, OTA_C_Response;

(Hs2bandpass) $-\frac{C\,g\,s}{C^2\,s^2+C\,g\,s+g^2}$

(%i42) Hs3lowpass: V[3]/Vg, OTA_C_Response;

(Hs3lowpass) $\frac{g^2}{C^2\,s^2+C\,g\,s+g^2}$

(%i43) numHs: num(Hs2bandpass);

(numHs) $-C\,g\,s$

(%i44) zeros: solve(numHs=0,s);

(zeros) $[s=0]$

(%i45) denHs: denom(Hs2bandpass);

(denHs) $C^2\,s^2+C\,g\,s+g^2$

(%i46) poles: solve(denHs=0,s);

(poles) $\left[s=-\frac{\left(\sqrt{3}\,\%i+1\right)g}{2\,C},\ s=\frac{\left(\sqrt{3}\,\%i-1\right)g}{2\,C}\right]$

17 Three-phase Circuit

(%i47)
```
ThreePhase_Shema: [
 ["Z", "Z0", 7, 0, Z0],
 ["V", "Ea", 4, 7, Ea],
 ["V", "Eb", 5, 7, Eb],
 ["V", "Ec", 6, 7, Ec],
 ["Z", "Zv1", 1, 4, Z],
 ["Z", "Zv2", 2, 5, Z],
 ["Z", "Zv3", 3, 6, Z],
 ["L", "L", 1, 2, L],
 ["R", "R", 2, 3, R],
 ["C", "C", 3, 1, C]
]$
```

(%i48)
```
ThreePhase_Response: SALECx(ThreePhase_Shema, omega)$
```

Phasor Transform at angular frequency ω

(%i49)
```
ThreePhase_Response_CL: ThreePhase_Response,
 C = 1/(sqrt(3)*R*omega),
 L = sqrt(3)*R/omega$
```

(%i50)
```
ThreePhase_Response_E: ThreePhase_Response_CL,
 Eb = Ea*exp(-j*2*%pi/3),
 Ec = Ea*exp(-j*4*%pi/3),
ratsimp;
```

(ThreePhase_Response_E)
$$\left[V_1=\frac{Ea\,R}{Z+R},V_2=-\frac{\left(3\,\%i+\sqrt{3}\right)Ea\,R}{2\,\sqrt{3}\,Z+2\,\sqrt{3}\,R},V_3=\frac{\left(3\,\%i-\sqrt{3}\right)Ea\,R}{2\,\sqrt{3}\,Z+2\,\sqrt{3}\,R},V_4=Ea,V_5=-\frac{\left(\sqrt{3}\,\%i+1\right)Ea}{2},V_6=\frac{\left(\sqrt{3}\,\%i-1\right)Ea}{2},V_7=0,I_{Ec}=-\frac{\left(\sqrt{3}\,\%i-1\right)Ea}{2\,Z+2\,R},I_{Eb}=\frac{\left(\sqrt{3}\,\%i+1\right)Ea}{2\,Z+2\,R},I_{Ea}=-\frac{Ea}{Z+R}\right]$$

(%i51)
```
I12: (V[1]-V[2])/(j*L*omega), ThreePhase_Response_E,
 L = sqrt(3)*R/omega,
ratsimp, factor;
```

(I12)
$$-\frac{\left(\sqrt{3}\,\%i-1\right)Ea}{2\left(Z+R\right)}$$

(%i52)
```
I23: (V[2]-V[3])/R, ThreePhase_Response_E, ratsimp;
```

(I23)
$$-\frac{\sqrt{3}\,\%i\,Ea}{Z+R}$$

(%i53)
```
I31: (V[3]-V[1])*(j*C*omega), ThreePhase_Response_E,
 C = 1/(sqrt(3)*R*omega),
ratsimp, factor;
```

(I31)
$$-\frac{\left(\sqrt{3}\,\%i+1\right)Ea}{2\left(Z+R\right)}$$

(%i54)
```
I123: matrix([I12],[I23],[I31]);
```

(I123)
$$\begin{bmatrix}-\frac{\left(\sqrt{3}\,\%i-1\right)Ea}{2\left(Z+R\right)}\\ -\frac{\sqrt{3}\,\%i\,Ea}{Z+R}\\ -\frac{\left(\sqrt{3}\,\%i+1\right)Ea}{2\left(Z+R\right)}\end{bmatrix}$$

(%i55)
```
remvalue(a)$
```

(%i56)
```
A: matrix([1,   1,   1],
          [1, a^2,   a],
          [1, a,   a^2]),
a = exp(j*2*%pi/3);
```

(A)
$$\begin{bmatrix}1 & 1 & 1\\ 1 & \left(\frac{\sqrt{3}\,\%i}{2}-\frac{1}{2}\right)^2 & \frac{\sqrt{3}\,\%i}{2}-\frac{1}{2}\\ 1 & \frac{\sqrt{3}\,\%i}{2}-\frac{1}{2} & \left(\frac{\sqrt{3}\,\%i}{2}-\frac{1}{2}\right)^2\end{bmatrix}$$

(%i57)
```
IsymmetricalComponents: invert(A).I123,
ratsimp, factor;
```

(IsymmetricalComponents)
$$\begin{bmatrix} -\frac{2\,\%i\,Ea}{\sqrt{3}\,(Z+R)} \\ -\frac{\%i\,(\sqrt{3}\,\%i-1)\,Ea}{2\,\sqrt{3}\,(Z+R)} \\ 0 \end{bmatrix}$$

(%i58)
```
Iabc: [-I["Ea"],-I["Eb"],-I["Ec"]], ThreePhase_Response_E;
```

(Iabc)
$$\left[\frac{Ea}{Z+R}, -\frac{(\sqrt{3}\,\%i+1)\,Ea}{2\,Z+2\,R}, \frac{(\sqrt{3}\,\%i-1)\,Ea}{2\,Z+2\,R}\right]$$

(%i59)
```
IabcSymmetricalComponents: invert(A).Iabc, ratsimp;
```

(IabcSymmetricalComponents)
$$\begin{bmatrix} 0 \\ \frac{Ea}{Z+R} \\ 0 \end{bmatrix}$$

(%i60)
```
substitute([j*sqrt(3)+1 = polarform(j*sqrt(3)+1),
            j*sqrt(3)-1 = polarform(j*sqrt(3)-1)],
 Iabc);
```

(%o60)
$$\left[\frac{Ea}{Z+R}, -\frac{2\,\%e^{\frac{\%i\,\pi}{3}}\,Ea}{2\,Z+2\,R}, \frac{2\,\%e^{\frac{2\,\%i\,\pi}{3}}\,Ea}{2\,Z+2\,R}\right]$$

18 Lumped Wilkinson Power Divider

(%i61)
```
lumpedWilkinson_Shema: [
 ["V", "Vgen", 4, 0, Vg],
 ["R", "R1", 1, 4, R],
 ["R", "R2", 2, 0, R],
 ["R", "R3", 3, 0, R],
 ["R", "R4", 2, 3, 2*R],
 ["L", "L2", 1, 2, sqrt(2)*R/omega],
 ["L", "L1", 1, 3, sqrt(2)*R/omega],
 ["C", "C1", 1, 0, 1/(sqrt(2)*R*omega)],
 ["C", "C2", 3, 0, 1/(sqrt(2)*R*omega)],
 ["C", "C3", 1, 0, 1/(sqrt(2)*R*omega)],
 ["C", "C4", 2, 0, 1/(sqrt(2)*R*omega)]
]$
```

(%i62)
```
lumpedWilkinson_Response:
 SALECx(lumpedWilkinson_Shema, omega),
ratsimp;
```

Phasor Transform at angular frequency ω

(lumpedWilkinson_Response)
$$\left[V_1=\frac{Vg}{2}, V_2=-\frac{\%i\,Vg}{2^{3/2}}, V_3=-\frac{\%i\,Vg}{2^{3/2}}, V_4=Vg, I_{Vgen}=-\frac{Vg}{2\,R}\right]$$

(%i63) `V[2]-V[3], lumpedWilkinson_Response;`

(%o63) 0

(%i64) `P2: abs(V[2])^2/R, lumpedWilkinson_Response;`

(P2) $$\frac{|Vg|^2}{8\,R}$$

(%i65) `P3: abs(V[3])^2/R, lumpedWilkinson_Response;`

(P3) $$\frac{|Vg|^2}{8\,R}$$

(%i66) `Pg: Vg*conjugate(I["Vgen"]), lumpedWilkinson_Response;`

(Pg) $$-\frac{Vg\,\overline{Vg}}{2\,R}$$

(%i67) `P1: abs(Vg-V[1])^2/R, lumpedWilkinson_Response;`

(P1) $$\frac{|Vg|^2}{4\,R}$$

(%i68) `P1 + P2 + P3 + Pg = 0, ratsimp;`

(%o68) $$\frac{|Vg|^2-Vg\,\overline{Vg}}{2\,R}=0$$

(%i69)
```
P1 + P2 + P3 + Pg = 0,
 Vg = Vgeff*exp(%i*thetag), ratsimp;
```

(%o69) $0=0$

19 Wilkinson Power Divider

(%i70)
```
Wilkinson_Shema: [
  ["V", "Vgen", 4, 0, Vg],
  ["R", "R1", 1, 4, R],
  ["R", "R2", 2, 0, R],
  ["R", "R3", 3, 0, R],
  ["R", "R4", 2, 3, 2*R],
  ["T", "T1", [1,0], [2,0], [sqrt(2)*R,%pi/2]],
  ["T", "T2", [1,0], [3,0], [sqrt(2)*R,%pi/2]]
]$
```

(%i71)
```
Wilkinson_Response: SALECx(Wilkinson_Shema, omega),
ratsimp;
```

Phasor Transform at angular frequency ω

(Wilkinson_Response) $$\left[V_1=\frac{Vg}{2},V_2=-\frac{\%i\,Vg}{2^{3/2}},V_3=-\frac{\%i\,Vg}{2^{3/2}},V_4=Vg,I_{T2,3}=-\frac{\%i\,Vg}{2^{3/2}\,R},I_{T2,1}=\frac{Vg}{4\,R},I_{T1,2}=-\frac{\%i\,Vg}{2^{3/2}\,R},I_{T1,1}=\frac{Vg}{4\,R},I_{Vgen}=-\frac{Vg}{2\,R}\right]$$

19.1 IR1, 2, 3

```
(%i72) V[2] - V[3], Wilkinson_Response;
(%o72) 0
```

```
(%i73) IR1: I["T1",1] + I["T2",1], Wilkinson_Response;
```
(IR1) $\frac{Vg}{2\,R}$

```
(%i74) IR2: V[2]/R, Wilkinson_Response;
```
(IR2) $-\frac{\%i\,Vg}{2^{3/2}\,R}$

```
(%i75) IR3: V[3]/R, Wilkinson_Response;
```
(IR3) $-\frac{\%i\,Vg}{2^{3/2}\,R}$

19.2 PR1, 2, 3

```
(%i76) PR1: R*abs(IR1)^2;
```
(PR1) $\frac{|Vg|^2}{4\,R}$

```
(%i77) PR2: R*abs(IR2)^2;
```
(PR2) $\frac{|Vg|^2}{8\,R}$

```
(%i78) PR3: R*abs(IR3)^2;
```
(PR3) $\frac{|Vg|^2}{8\,R}$

```
(%i79) Pg: Vg*conjugate(I["Vgen"]);
```
(Pg) $\overline{I_{Vgen}}\,Vg$

```
(%i80) PR1 + PR2 + PR3 + Pg = 0, Wilkinson_Response;
```
(%o80) $\frac{|Vg|^2}{2\,R}-\frac{Vg\,\overline{Vg}}{2\,R}=0$

```
(%i81) PR1 + PR2 + PR3 + Pg = 0, Wilkinson_Response,
        Vg = Vgeff*exp(j*thetag), ratsimp;
```
(%o81) $0=0$

19.3 I["Vgen"]

(%i82) I["Vgen"] + IR1 = 0;

(%o82) $\frac{Vg}{2\,R}+I_{Vgen}=0$

(%i83) I["Vgen"] + IR1 = 0, Wilkinson_Response;

(%o83) $0=0$

19.4 props

(%i84) props;

(%o84) [$nset$, $\{,\}$, $trylevel$, $maxmin$, $nummod$, $conjugate$, $erf\_generalized$, β, $desolve$, $eliminate$, $adjoint$, $invert\_by\_adjoint$, $wxmaxima$, Ea, Eb, Ec, I, $I1$, $I2$, Ig, s, V, $V1$, $V2$, Vg, $Y1$, $Y2$, Z, $Z0$, $Z1$, $Z2$, C, $C1$, $C2$, g, Io, $Io1$, $Io2$, L, $L1$, $L2$, $L12$, m, R, $R1$, $R2$, $R3$, $R4$, $R5$, t, $Vgeff$, Vo, $Vstep$, Zc, ω, θ, $thetag$, τ, n]

(%i85) properties(I);

(%o85) [*database info*, kind(I, *complex*)]

(%i86) properties(J);

(%o86) [*hashed array*]

(%i87) properties(V);

(%o87) [*hashed array*, *database info*, kind(V, *complex*)]

20 Noninverting OpAmp Amplifier

```
(%i88) nonInvOpAmp_Shema: [
        ["V", "Vgen", 1, 0, Vg],
        ["R", "R1", 2, 0, R1],
        ["R", "R2", 2, 3, R2],
        ["OpAmp", "OpAmp", [1,2], 3]
       ]$
```

(%i89) nonInvOpAmp_Response: SALECx(nonInvOpAmp_Shema);

(nonInvOpAmp_Response) $[V_1=Vg, V_2=Vg, V_3=\frac{R2\,Vg+R1\,Vg}{R1}, I_{OpAmp}=-\frac{Vg}{R1}, I_{Vgen}=0]$

(%i90) voltageGain: V[3]/Vg, nonInvOpAmp_Response, expand;

(voltageGain) $\frac{R2}{R1}+1$

21 Voltage Follower

```
(%i91) VoltageFollower_Shema: [
        ["V", "Vgen", 1, 0, Vg],
        ["R", "Rfb", 2, 3, R],
        ["OpAmp", "OpAmp", [1,2], 3]
       ]$
```

```
(%i92) VoltageFollower_Response: SALECx(VoltageFollower_Shema);
```

(VoltageFollower_Response) $[V_1=Vg, V_2=Vg, V_3=Vg, I_{OpAmp}=0, I_{Vgen}=0]$

```
(%i93) voltageGain: V[3]/Vg, VoltageFollower_Response;
```

(voltageGain) 1

22 Riordan Gyrator Synthetic Inductor

```
(%i94) Riordan_Shema: [
        ["V", "Vgen", 1, 0, Vg],
        ["OpAmp", "OpAmp1", [1,4], 5],
        ["R", "R1", 4, 0, R1],
        ["C", "C2", 4, 5, C2],
        ["R", "R3", 5, 2, R3],
        ["OpAmp", "OpAmp2", [1,2], 3],
        ["R", "R4", 2, 3, R4],
        ["R", "R5", 1, 3, R5]
       ]$
```

```
(%i95) Riordan_Response: SALECx(Riordan_Shema),
        SALECxPrint: true;
```

Symbolic Analysis of Linear Electric Circuits with Maxima
SALECx version 1.0, Prof. Dr. Dejan Tošić, tosic@etf.rs
Number of nodes excluding 0 node: 5
Electric circuit specification: [[V, Vgen, 1, 0, Vg], [OpAmp, OpAmp1, [1, 4], 5], [R, R1, 4, 0, R1], [C, C2, 4, 5, C2], [R, R3, 5, 2, R3], [OpAmp, OpAmp2, [1, 2], 3], [R, R4, 2, 3, R4], [R, R5, 1, 3, R5]]
Supported element: [true, true, true, true, true, true, true, true]
Element values: [Vg, false, R1, C2, R3, false, R4, R5]
Initial conditions: [false, false, false, false, false, false, false, false]

MNA equations: $\left[\frac{V_1-V_3}{R5}+I_{Vgen}=0, \frac{V_2-V_3}{R4}+\frac{V_2-V_5}{R3}=0, \frac{V_3-V_1}{R5}+\frac{V_3-V_2}{R4}+I_{OpAmp2}=0, \left(V_4-V_5\right) C2\, s+\frac{V_4}{R1}=0, \left(V_5-V_4\right) C2\, s+\frac{V_5-V_2}{R3}+I_{OpAmp1}=0, V_1-V_2=0, V_1-V_4=0, V_1=Vg\right]$

MNA variables: $[V_1, V_2, V_3, V_4, V_5, I_{OpAmp2}, I_{OpAmp1}, I_{Vgen}]$

(Riordan_Response) $\left[V_1=Vg, V_2=Vg, V_3=-\frac{R4\, Vg-C2\, R1\, R3\, Vg\, s}{C2\, R1\, R3\, s}, V_4=Vg, V_5=\frac{C2\, R1\, Vg\, s+Vg}{C2\, R1\, s}, I_{OpAmp2}=\frac{R5\, Vg+R4\, Vg}{C2\, R1\, R3\, R5\, s}, I_{OpAmp1}=-\frac{C2\, R3\, Vg\, s+Vg}{C2\, R1\, R3\, s}, I_{Vgen}=-\frac{R4\, Vg}{C2\, R1\, R3\, R5\, s}\right]$

22.1 Input Impedance

```
(%i96) Zin: Vg/(-I["Vgen"]), Riordan_Response;
```

(Zin) $$\frac{C2\ R1\ R3\ R5\ s}{R4}$$

```
(%i97) Lsynthetic: Zin/s;
```

(Lsynthetic) $$\frac{C2\ R1\ R3\ R5}{R4}$$

```
(%i98) V54: V[5] - V[4], Riordan_Response, ratsimp;
```

(V54) $$\frac{Vg}{C2\ R1\ s}$$

23 Wien Bridge Oscillator

```
(%i99) Wien_Shema: [
   ["R", "R1", 1, 0, R],
   ["C", "C1", 1, 0, C],
   ["VCVS", "Amp", [1,0], [2,0], 3],
   ["R", "R2", 1, 3, R],
   ["C", "C2", 3, 2, C, Vo]
  ]$
```

```
(%i100) Wien_Reponse: SALECx(Wien_Shema);
```

(Wien_Reponse) $$[V_1 = \frac{C\ R\ Vo}{C^2\ R^2\ s^2+1}, V_2 = \frac{3\ C\ R\ Vo}{C^2\ R^2\ s^2+1}, V_3 = \frac{C^2\ R^2\ Vo\ s+2\ C\ R\ Vo}{C^2\ R^2\ s^2+1}, I_{Amp} = -\frac{C^2\ R\ Vo\ s+C\ Vo}{C^2\ R^2\ s^2+1}]$$

```
(%i101) v2t: ilt(ev(V[2],Wien_Reponse), s, t);
```

(v2t) $$3\ Vo\ \sin\left(\frac{t}{C\ R}\right)$$

24 ABCD Circuit

```
(%i102) ABCD_Shema: [ ["I", "Igen1", 1, 0, Ig1],
                      ["Y",  "Y1", 1, 0,  Y1],
                      ["ABCD", "ABCD", [1,0], [2,0], [[a11,a12],[a21,a22]]],
                      ["I", "Igen2", 2, 0, Ig2],
                      ["Y",  "Y2", 2, 0,  Y2]
                    ]$
```

```
(%i103) ABCD_Response: SALECx(ABCD_Shema);
```

(ABCD_Response) $$[V_1 = -\frac{Ig2\ (a11\ a22-a12\ a21)+Ig1\ (Y2\ a12+a11)}{Y2\ a22+a21+Y1\ (Y2\ a12+a11)}, V_2 = -\frac{Ig2\ a22+Ig2\ Y1\ a12+Ig1}{Y2\ a22+a21+Y1\ (Y2\ a12+a11)}, I_{ABCD,2} = \frac{Ig2\ a21+Ig2\ Y1\ a11-Ig1\ Y2}{Y2\ a22+a21+Y1\ (Y2\ a12+a11)}, I_{ABCD,1} = \frac{Ig2\ Y1\ (a11\ a22-a12\ a21)+Ig1\ (-Y2\ a22-a21)}{Y2\ a22+a21+Y1\ (Y2\ a12+a11)}]$$

```
(%i104) ABCD_Response_noIg: eliminate(ABCD_Response, [Ig1,Ig2]);
```

(ABCD_Response_noIg) $[-(Y2\,a12+a11)(Y2\,a22+a21+Y1\,Y2\,a12+Y1\,a11)^2((V_2\,a11-V_1)\,a22-V_2\,a12\,a21+I_{ABCD,1}\,a12),-(Y2\,a12+a11)(Y2\,a22+a21+Y1\,Y2\,a12+Y1\,a11)^2(I_{ABCD,2}\,a11\,a22+(V_1-I_{ABCD,2}\,a12)\,a21-I_{ABCD,1}\,a11)]$

```
(%i105) ABCD_Response_VI: solve(ABCD_Response_noIg, [V[1],I["ABCD",1]]);
```

(ABCD_Response_VI) $[[V_1=I_{ABCD,2}\,a12+V_2\,a11,I_{ABCD,1}=I_{ABCD,2}\,a22+V_2\,a21]]$

I["ABCD",1] is current into pin 1, and
I["ABCD",2] is current *OUT OF* pin 2.

25 Time and Date

```
(%i106) timedate();
```

(%o106) *2019-08-26 17:00:24+02:00*

25.1 aliases, arrays

```
(%i107) aliases;
```

(%o107) [%i]

```
(%i108) arrays;
```

(%o108) $[J,V]$

```
(%i109) arrayinfo(J);
```

(%o109) [*hashed*, 1, [0], [1], [2], [3], [4], [5], [6], [7]]

```
(%i110) arrayinfo(V);
```

(%o110) [*hashed*, 1, [0]]

25.2 facts, functions

```
(%i111) facts();
(%o111) [kind(Ea,complex),kind(Eb,complex),kind(Ec,complex),
kind(I,complex),kind(I1,complex),kind(I2,complex),kind(Ig,complex),
kind(s,complex),kind(V,complex),kind(V1,complex),kind(V2,complex),
kind(Vg,complex),kind(Y1,complex),kind(Y2,complex),kind(Z,complex),
kind(Z0,complex),kind(Z1,complex),kind(Z2,complex),kind(C,real),
kind(C1,real),kind(C2,real),kind(g,real),kind(Io,real),
kind(Io1,real),kind(Io2,real),kind(L,real),kind(L1,real),
kind(L2,real),kind(L12,real),kind(m,real),kind(R,real),kind(R1,real)
,kind(R2,real),kind(R3,real),kind(R4,real),kind(R5,real),
kind(t,real),kind(Vgeff,real),kind(Vo,real),kind(Vstep,real),
kind(Zc,real),kind(ω,real),kind(θ,real),kind(thetag,real),
kind(τ,real),kind(n,integer),C>0,C1>0,C2>0,L>0,L1>0,L2>0,L12>0,
notequal(m,0),n>-1,R>0,R1>0,R2>0,R3>0,R4>0,R5>0,Vgeff>0,Vstep>
0,Zc>0,ω>0,τ>0]
```

```
(%i112) functions;
(%o112) [ElementStamp(e),SALECx(circuit_,[w])]
```

25.3 infolists, props

```
(%i113) infolists;
(%o113) [labels,values,functions,macros,arrays,myoptions,props,
aliases,rules,gradefs,dependencies,let_rule_packages,structures]
```

```
(%i114) props;
(%o114) [nset,{,},trylevel,maxmin,nummod,conjugate,erf_generalized,
β,desolve,eliminate,adjoint,invert_by_adjoint,wxmaxima,Ea,Eb,Ec,I,
I1,I2,Ig,s,V,V1,V2,Vg,Y1,Y2,Z,Z0,Z1,Z2,C,C1,C2,g,Io,Io1,Io2,L
,L1,L2,L12,m,R,R1,R2,R3,R4,R5,t,Vgeff,Vo,Vstep,Zc,ω,θ,thetag,τ,
n]
```

```
(%i115) propvars(complex);
(%o115) [Ea,Eb,Ec,I,I1,I2,Ig,s,V,V1,V2,Vg,Y1,Y2,Z,Z0,Z1,Z2]
```

```
(%i116) propvars(real);
(%o116) [C,C1,C2,g,Io,Io1,Io2,L,L1,L2,L12,m,R,R1,R2,R3,R4,R5,t
,Vgeff,Vo,Vstep,Zc,ω,θ,thetag,τ]
```

25.4 values

```
(%i117) values;
(%o117) [SALECxPrint,Vg_Shema,PhasorTransform_,elementValues_,
initialConditions_,potentials_,m,equationsVn_,equationsMNA_,
variablesMNA_,responseMNA_,Vg_Response,IgR_Shema,IgR_Response,
VgR_Shema,VgR_Response,VgRCVo_Shema,VgRCVo_Response_PT,V2PT,
VgRCVo_Response,V2s,v2ilt,v2t,TLineZc_Shema,tau_,a1_,a2_,b1_,b2_,
TLineZc_Response,V1s,TLineZc_PT_Shema,TLineZc_PT_Response,V1w,V2w,
OTA_C_Shema,OTA_C_Response,Hs2bandpass,Hs3lowpass,numHs,zeros,denHs
,poles,ThreePhase_Shema,ThreePhase_Response,ThreePhase_Response_CL,
ThreePhase_Response_E,I12,I23,I31,I123,A,IsymmetricalComponents,Iabc
,IabcSymmetricalComponents,lumpedWilkinson_Shema,
lumpedWilkinson_Response,P2,P3,Pg,P1,Wilkinson_Shema,
Wilkinson_Response,IR1,IR2,IR3,PR1,PR2,PR3,nonInvOpAmp_Shema,
nonInvOpAmp_Response,voltageGain,VoltageFollower_Shema,
VoltageFollower_Response,Riordan_Shema,Riordan_Response,Zin,
Lsynthetic,V54,Wien_Shema,Wien_Reponse,ABCD_Shema,JJ,VV,a11_,a12_,
a21_,a22_,ABCD_Response,ABCD_Response_noIg,ABCD_Response_VI]
```

26 End-of-File

```
(%i118) timedate();
(%o118) 2019-08-26 17:00:25+02:00
```

END-OF-FILE