

Virtual Build (technological)

<http://www.partsim.com/>

Online Circuit Simulator x

www.partsim.com/simulator/#78462

Fre Antwerp Login Fin Utils Articles Jobs Esquilha PhdVenues PhdVenues Altitude atomp

PartSim

DEMO Differential Amplifier (work will not be saved)

Subcircuit Report x Transient Analysis x BOM

Project Save Save As New Open

Spice Run Models Export Netlist Share Print Edit Cut Copy Paste Delete Undo Redo Insert Textbox

Components Probes

Search Parts

- Fairchild Semiconductor... 50
- IXYS 17
- Infineon Technologies 107
- N-Channel MOSFETs 107
- Maxim IC 1,082
- Microchip
- Nichicon
- On Semiconductor
- Vishay

AUTOMOTIVE INDUSTRIAL

COMPUTER PRODUCT

ELECTRIC ELECTRO

ELECTRO

107 Results

IP MOS

Category: Infineon Technologies
Subcategory: N-Channel MOSFETs
Section: VENDOR PARTS
Description: MOSFET N-CH 600V TO220-3
Name: IPP6IR160P6XKA1
Infineon

15V

V1

2N2222

TR1

R2 1K

R3 1K

R4 1K

Re

z 0.05

Vout

1K R1

2N2222

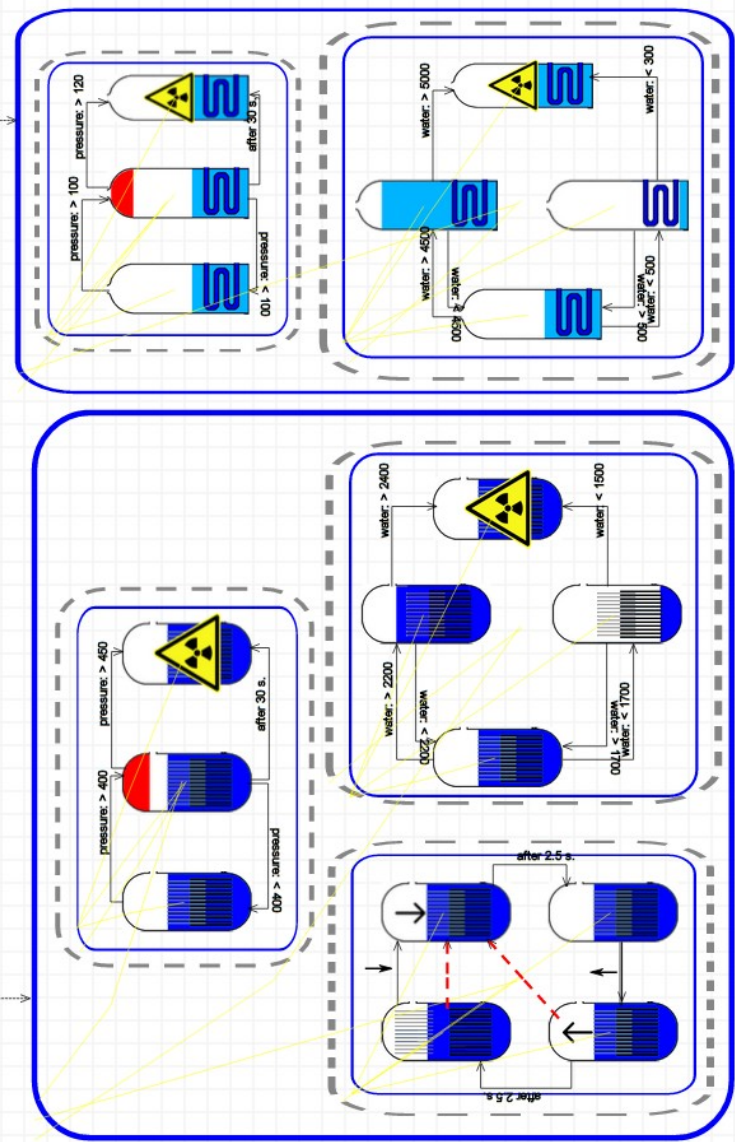
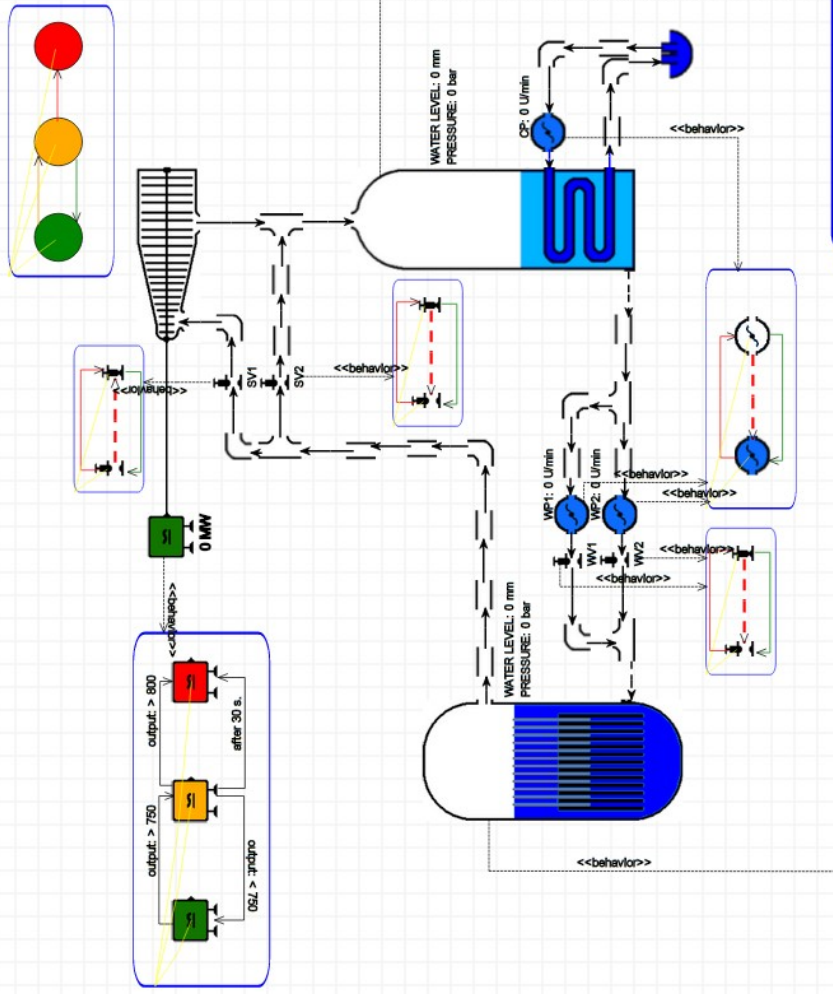
TR2

V2

SINE (0 -100m 40Hz)

SINE (0 100m 500Hz)

Design (Space Exploration)
as a service





Boric Acid Transportation Pump

Product parameters

Design standards : RCC-M

Flow : 16.6m³/h

Head : 85m

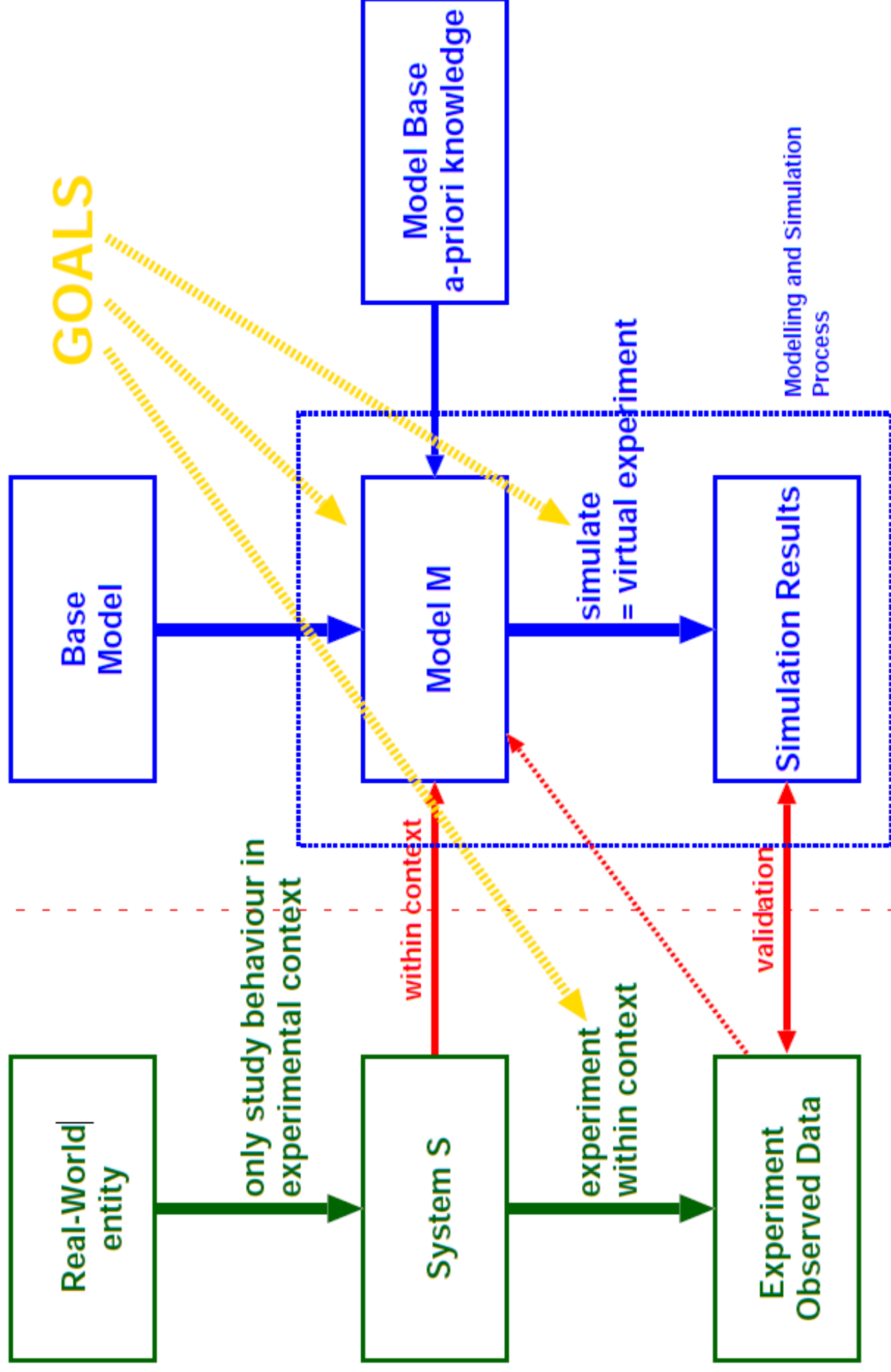
Temperature : ~80°C

Pressure : 1.6MPa

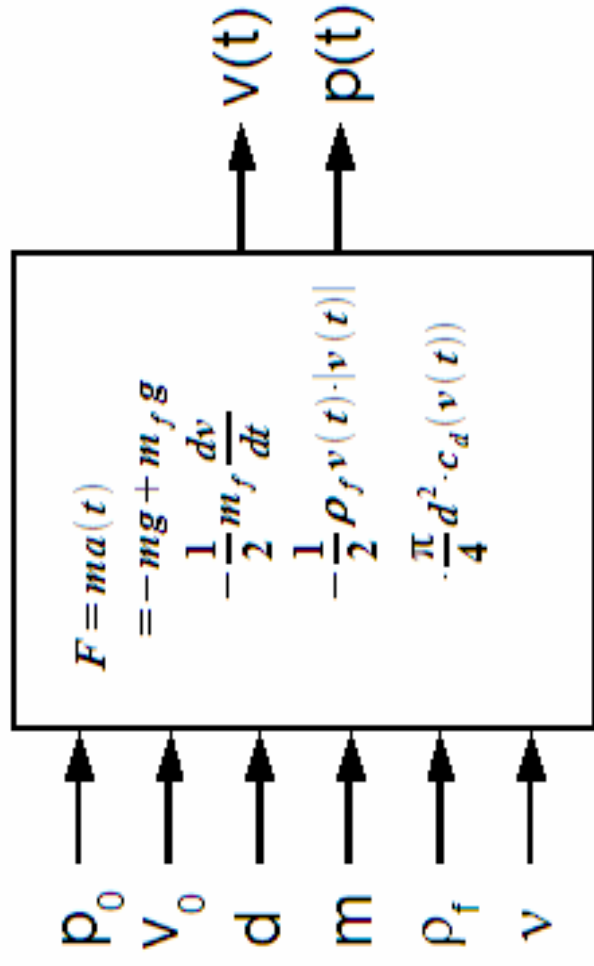
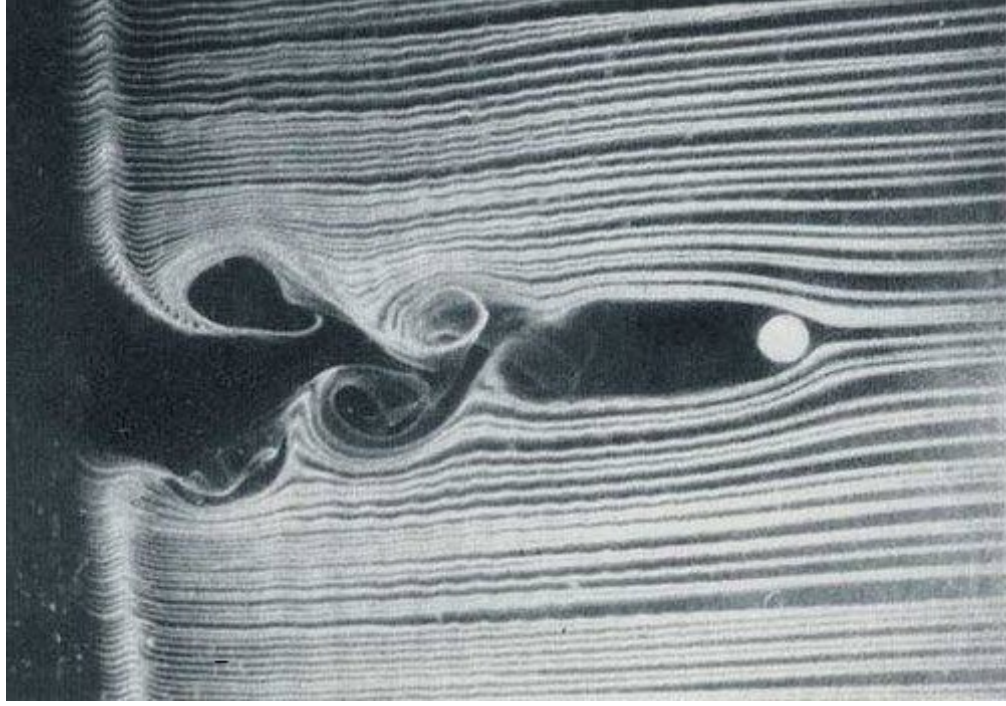
Used in 600MWe 、 900MWe 、 1000MWe PWR nuclear power plant boric acid transportation system.

REALITY

MODEL



Model Validity ... Context?



Spiegel, M., Reynolds, P. F., & Brogan, D. C.
A Case Study of Model Context for Simulation Composability and Reusability.
In *Proceedings of the Winter Simulation Conference*, 2005. (Vol. 2005, pp. 437–444). IEEE.
<http://doi.org/10.1109/WSC.2005.1574279>

1. Invariant Constraints

1.a Sphere Attributes

1. Sphere Property - The body is a sphere and it remains spherical.
2. Smooth Property - The body is smooth and it remains smooth.
3. Impermeable Property - The body is completely impermeable.
4. Initial Velocity - The body has an initial velocity of v_0 that has no horizontal component of motion.
5. Angular Velocity - The body has no initial angular velocity.
6. Constant Mass - The mass of the body remains constant over time. The body does not experience ablation or accretion.
7. Constant Diameter - The diameter of the body remains constant over time.
8. Distribution of Mass - The body has a centrally symmetric mass distribution that remains constant over time.
9. Uncertainty Principle - The diameter of the body is much greater than the Planck length.
10. Brownian Motion - The mass and diameter of the body are large enough such that Brownian motion of the fluid has negligible impact on the body.
11. General Relativity - The mass of the body is low enough to ignore the gravitational curvature of space-time.

1.c Earth Attributes

18. Flat Terrain - The ground does not have terrain and remains flat for all $t > 0$.
19. Coriolis Effect - The Earth is not rotating. We ignore the Coriolis effect.

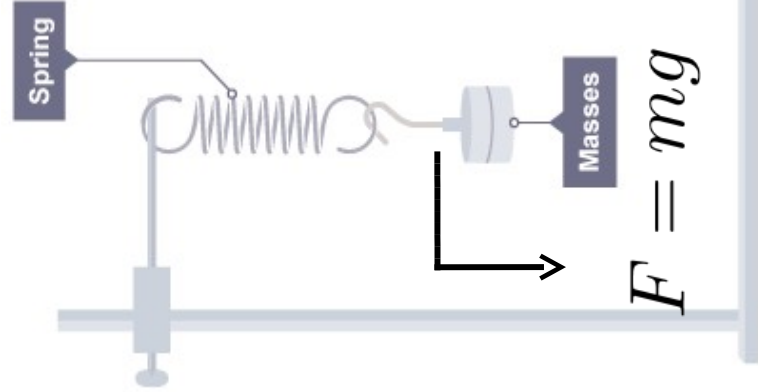
2. Dynamic Constraints

20. Mach Speed - The velocity of the body is sufficiently less than the speed of sound for that medium.
21. Special Relativity - The velocity of the body is sufficiently less than the speed of light for that medium.
22. Reynolds Number - The Reynolds number remains between 10^{-2} and 10^7 for all $t > 0$. The Reynolds number is a function of velocity.

3. Inter-Object Constraints

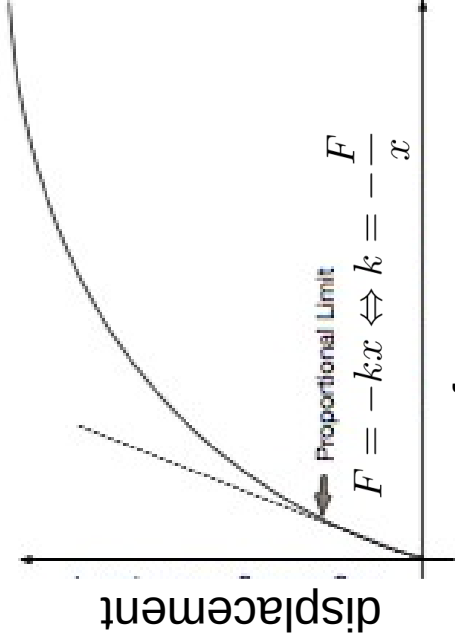
23. Sphere/Fluid Interaction - The body and the fluid interact only through buoyancy and drag. For example, the body cannot dissolve in the fluid, nor can the body transfer heat to the fluid.
24. Sphere/Earth Interaction - The body and the earth interact only through the gravitational force.
25. Fluid/Earth Interaction - The fluid and the earth do not interact.
26. Closed System - The Earth, sphere, and fluid do not interact with any other objects.
27. Simple Gravity - Gravity is a constant downward force of 9.8 m/s^2 .
28. One-Sided Gravity - The mass of the body is much less than the mass of the Earth. The Earth is not affected by the gravitational pull of the body.
29. Inelastic Collision - The collision between the sphere and the ground is perfectly inelastic.

Implicit Assumptions!



Experimental spring results, with mass m in kg and displacement x (± 0.0001) in cm

m	x	m	x	m	x	m	x
1	2.100	3	6.3749	5	10.4915	7	14.6081
2	4.3166	4	8.4332	6	12.5489	8	16.7774
						9	19.0012

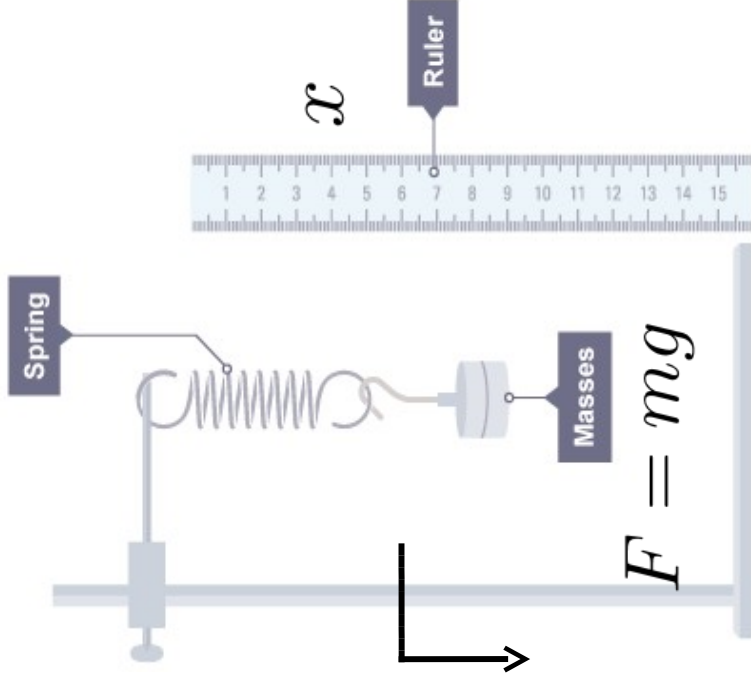
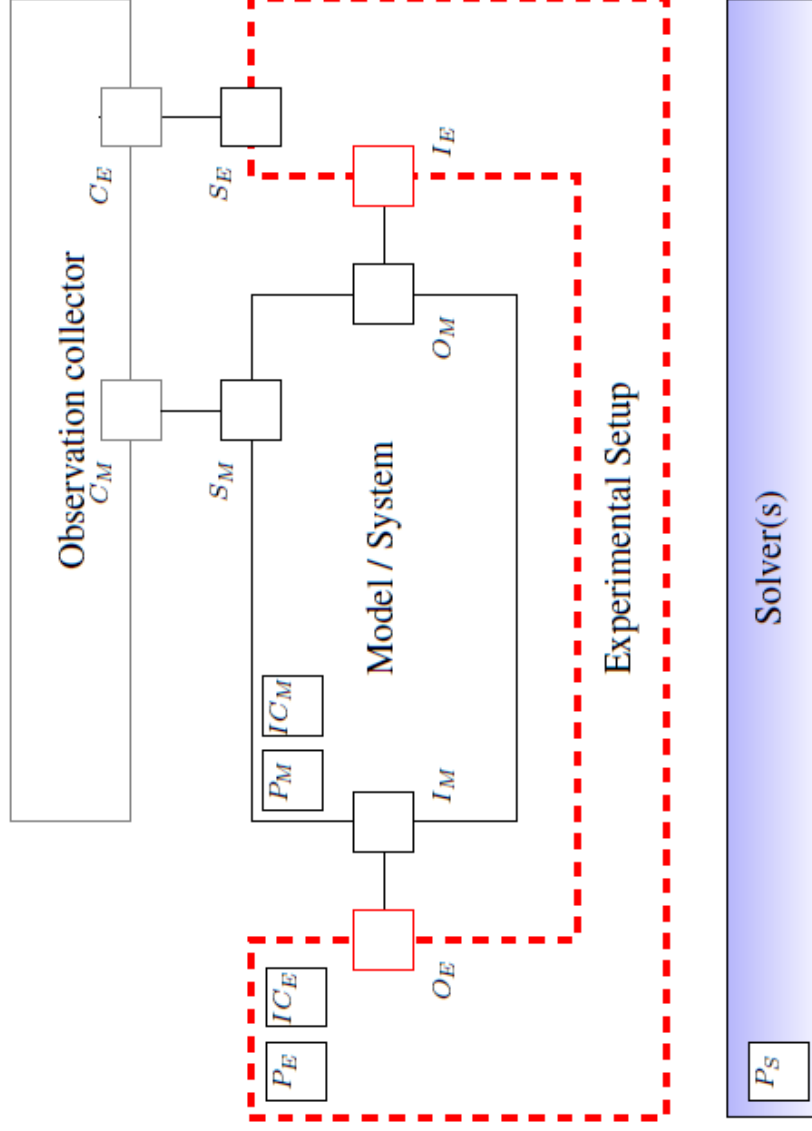


force

O.D.	FREE LENGTH	I.D.	RATE	SUGG. MAX. DEFL.	SUGG. MAX. LOAD	SOLID LENGTH	WIRE DIA.	TOTAL COILS	MAT'L	WIRE SIZE
Inches	Inches	Inches	Lbs./in.	Inches	Lbs.	Inches	Inches			
0.036	.59	.022	.6	.15	.39	.35	.007	49.0	SST	C
0.036	.63	.024	.6	.16	.25	.25	.006	40.0	SST	C
0.040	.66	.020	.5	.13	1.4	.50	.010	48.5	MW	C
0.040	1.02	.028	.7	.35	1.6	.30	.006	49.5	MW	C
0.054	1.37	.036	.9	.09	2.2	.16	.009	16.5	SST	C
0.054	1.37	.038	1.0	.37	.60	.32	.008	39.0	MW	C
0.057	.13	.045	1.1	.07	.25	.04	.006	5.75	MW	C
0.057	.13	.045	1.1	.05	.17	.04	.006	5.75	SST	C
0.057	.13	.043	1.1	.06	1.8	.05	.007	6.00	MW	C
0.057	.13	.043	1.1	.04	1.2	.05	.007	6.00	SST	C
0.057	.13	.041	1.0	.05	.57	.06	.008	6.13	MW	C
0.057	.13	.041	1.0	.03	.37	.06	.008	6.13	SST	C
0.057	.19	.045	1.1	.11	.25	.06	.006	8.13	MW	C
0.057	.19	.045	1.1	.08	.17	.06	.006	8.13	SST	C
0.057	.19	.043	1.1	.10	.40	.07	.007	8.88	MW	C
0.057	.19	.043	1.1	.07	.26	.07	.007	8.88	SST	C
0.057	.19	.041	1.0	.08	.57	.08	.008	8.75	MW	C
0.057	.19	.041	1.0	.06	.37	.08	.008	8.75	SST	C
0.057	.25	.045	1.1	.15	.25	.07	.006	10.3	MW	C
0.057	.25	.045	1.1	.11	.17	.07	.006	10.3	SST	C
0.057	.25	.043	1.1	.13	.40	.08	.007	11.0	MW	C
0.057	.25	.043	1.1	.10	.26	.08	.007	11.0	SST	C
0.057	.25	.041	1.0	.11	.57	.10	.008	11.5	MW	C
0.057	.25	.041	1.0	.08	.37	.10	.008	11.5	SST	C
0.057	.31	.045	1.1	.14	.25	.08	.006	12.4	MW	C
0.057	.31	.045	1.1	.14	.17	.08	.006	12.4	SST	C
0.057	.31	.043	1.1	.17	.40	.10	.007	13.5	MW	C
0.057	.31	.043	1.1	.12	.26	.10	.007	13.5	SST	C
0.057	.31	.041	1.0	.21	.57	.12	.008	14.3	MW	C
0.057	.31	.041	1.0	.14	.37	.12	.008	14.3	SST	C



Validity “Frame” ~ reproducibility



Denil, J., Kiklovits, S., Mosterman, P. J., Vallecillo, A., & Vangheluwe, H. (2017). The experiment model and validity frame in M&S. In *Proceedings of the Symposium on Theory of Modeling & Simulation* (Vol. 49).

Vanherpen, K., Denil, J., De Meulenaere, P., & Vangheluwe, H. (2016). Ontological Reasoning as an Enabler of Contract-Based Co-design.

In C. Berger, M. R. Mousavi, & R. Wisniewski (Eds.), *Cyber Physical Systems. Design, Modeling, and Evaluation: 6th International Workshop, CyPhy 2016, Pittsburgh, PA, USA, October 6, 2016, Revised Selected Papers* (pp. 101–115). Cham: Springer International Publishing. http://doi.org/10.1007/978-3-319-51738-4_8

A-Causal Modelling in Context

- Problem-Specific (technological)
- Domain-Specific (e.g., translational mechanical)
- (general) Laws of Physics
- Power Flow/Bond Graphs (physical: energy/power)
- Computationally a-causal
(Mathematical and Object-Oriented) ← **Modelica**
- Causal Block Diagrams (data flow)
- Numerical (Discrete) Approximations
- Computer Algorithmic + Numerical
(Floating Point vs. Fixed Point)
- As-Fast-As-Possible vs. Real-time (XiL)
- Hybrid (discrete-continuous) modelling/simulation
- Hiding IP: Composition of Functional Mockup Units (FMI)
- Dynamic Structure

Dokumentgivare
Lund Institute of Technology
Hedlaggare
Karl Johan Aström
Förläggare
Hilding Elmqvist

Dokumentnr
REPORT LUTED/ (TRF-1015)/1-226/ (1978)

Dokumenttitel
REPORT LUTED/ (TRF-1015)/1-226/ (1978)

Utgivningsdatum
May 1978

Ärendebeteckning
6016

Hilding Elmqvist



Dokumenttitel och undertitel

A Structured Model Language for Large Continuous Systems

Referat (sammandrag)

A model language, called DYMOLA, for continuous dynamical systems is proposed. Large models are conveniently described hierarchically using a submodel concept. The ordinary differential equations and algebraic equations need not be converted to assignment statements. There is a concept, cut, which corresponds to connection mechanisms of complex types, and there are facilities to describe the connection structure of a system. A model can be manipulated for different purposes such as simulation and static calculations. The model equations are sorted and they are converted to assignment statements using formula manipulation. A translator for the model language is also included.

Referat skrivet av

Author

Förslag till ytterligare nyckelord

nonlinear systems, compiler, permutations, graph theory

Klassifikationssystem och -klass(er)

Indexord (ange källa)

Mathematical models, Simulation languages, Computerized simulation, Nonlinear systems, Ordinary differential equations, Compilers.

(Thesaurus of Engineering and Scientific Terms, Eng. Joint Council, USA)

Omfång

226 pages

Övriga bibliografiska uppgifter

Språk

English

Sekretessuppgifter

ISSN

ISBN

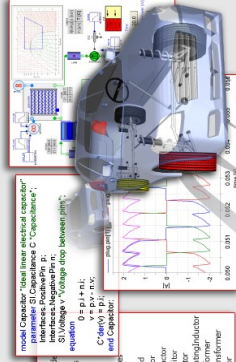
Dokumentet kan erhållas från

Department of Automatic Control
Lund Institute of Technology
P O Box 725, S-220 07 Lund 7, Sweden

Pris

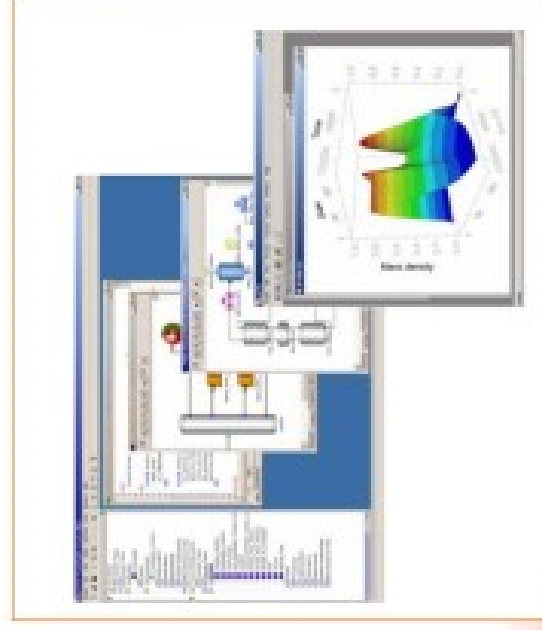
5850

Blankett LU 11:25 1976-07

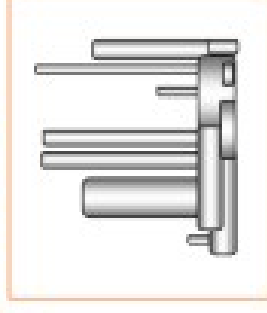


gPROMS ModelBuilder

Model development validation
& maintenance

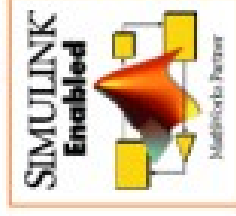


gO:RUN



Packaged models for execution-only
("runtime") applications

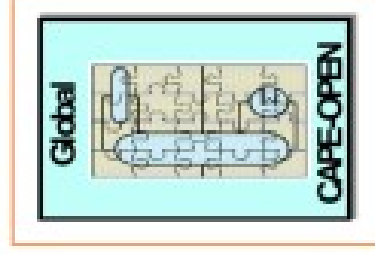
gO:Simulink
gO:MATLAB



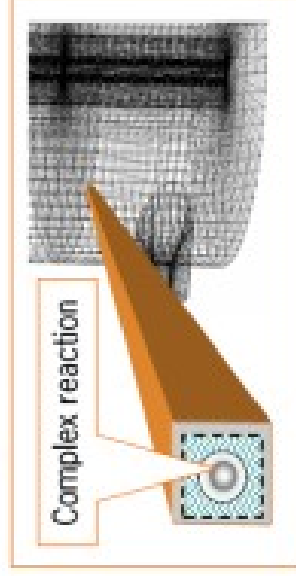
Detailed dynamic process models in
MATLAB and Simulink®

Detailed unit operation models in
CAPE-OPEN flow-sheeting packages

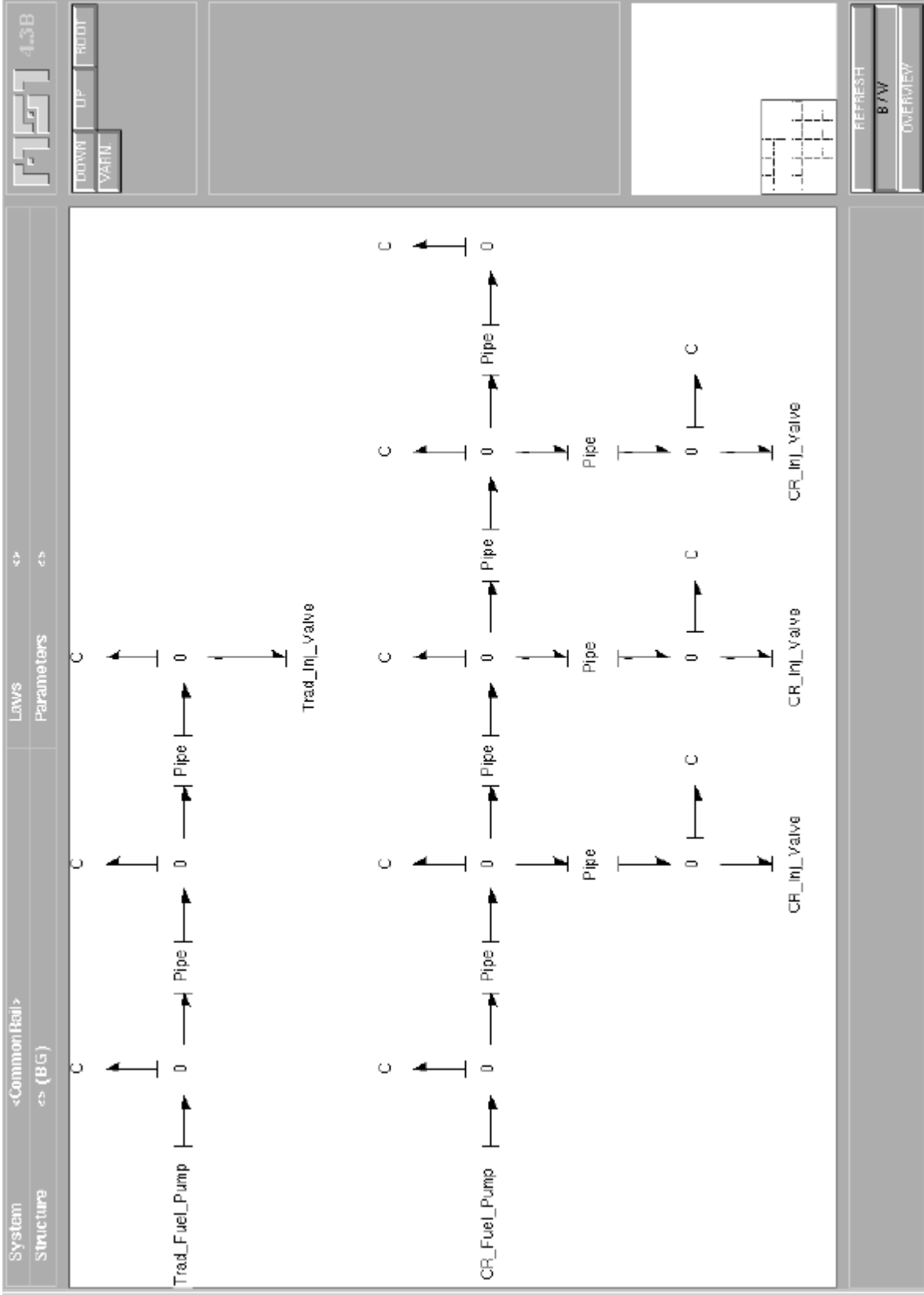
gO:CAPE-OPEN



gO:CFD



Advanced reaction modelling for CFD tools





20-SIM

20-sim Editor on: Baxandall.emx

File Edit View Insert Model Drawing Settings Tools Help

Model Library

model

- Ai
- Ai_dB
- Ao
- Ao_dB
- C1
- C2
- C3
- C4
- Ground1
- Ground2
- Ground4
- Ground5
- R3
- R4
- R5
- R6
- R7
- R8
- R_high

Interface Icon Globals

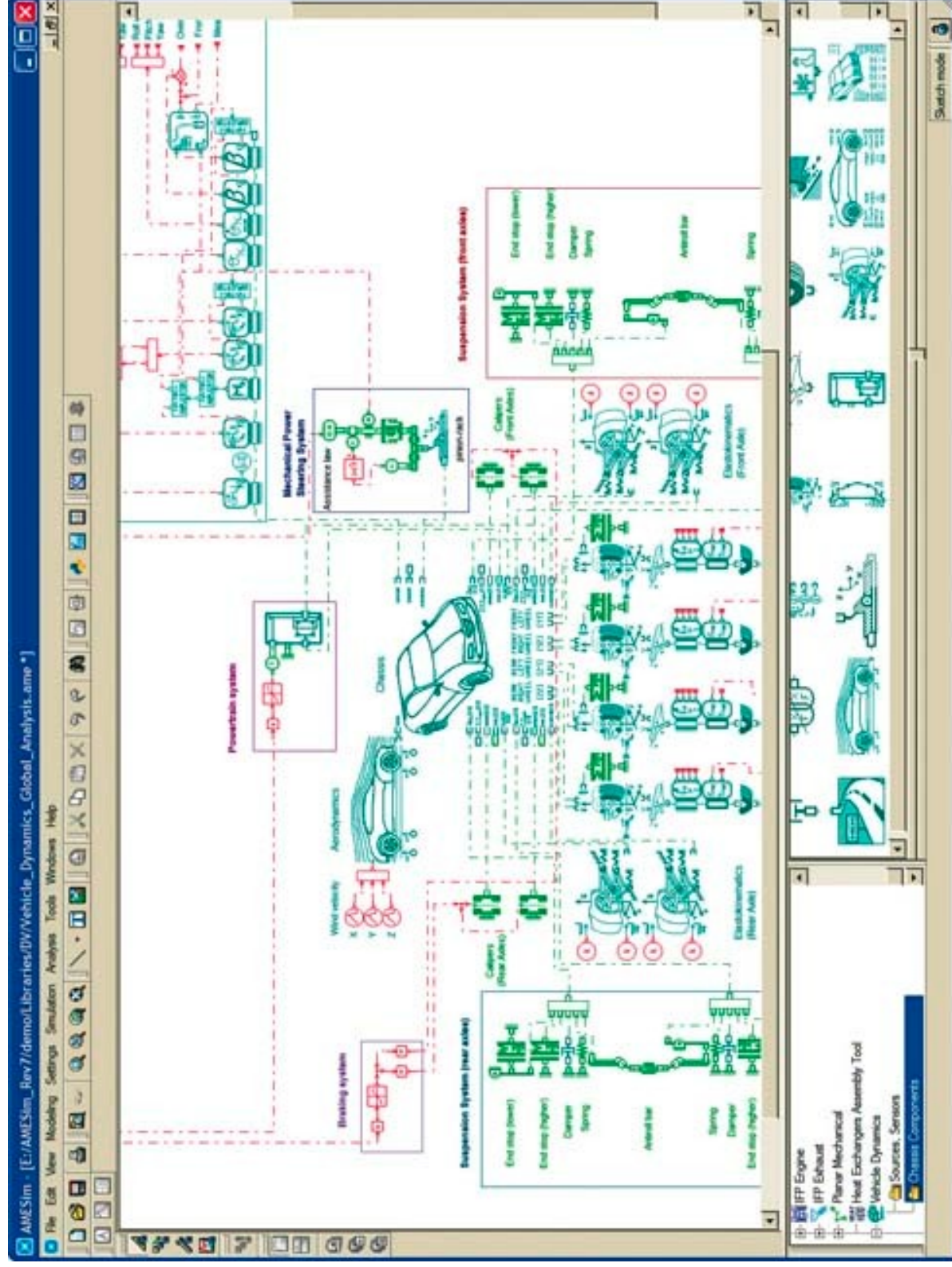
Output Process Find

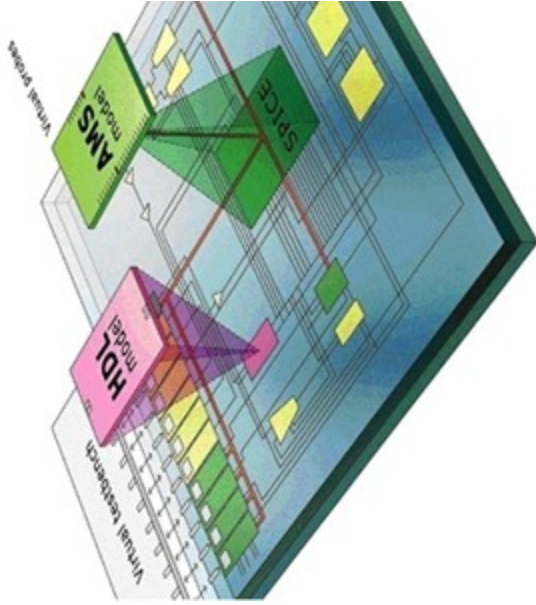
81 equations
67 variables
4 independent states
2 algebraic loops

The model has 0 errors and 0 warnings.

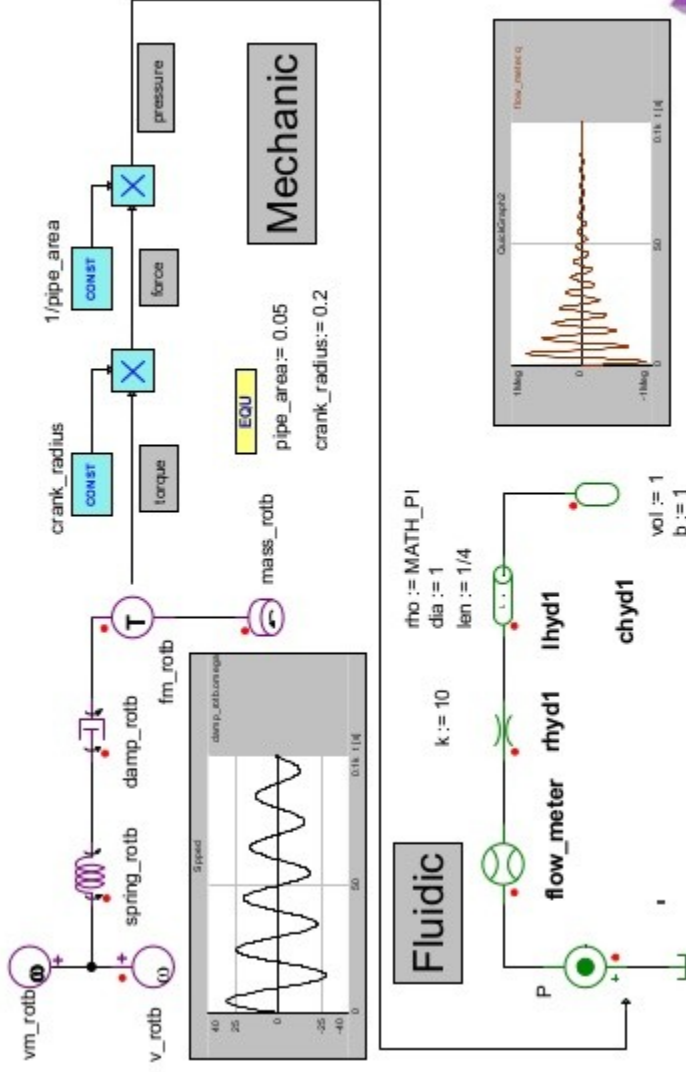
The model has 0 errors and 0 warnings.

Imagine.Lab AMESim



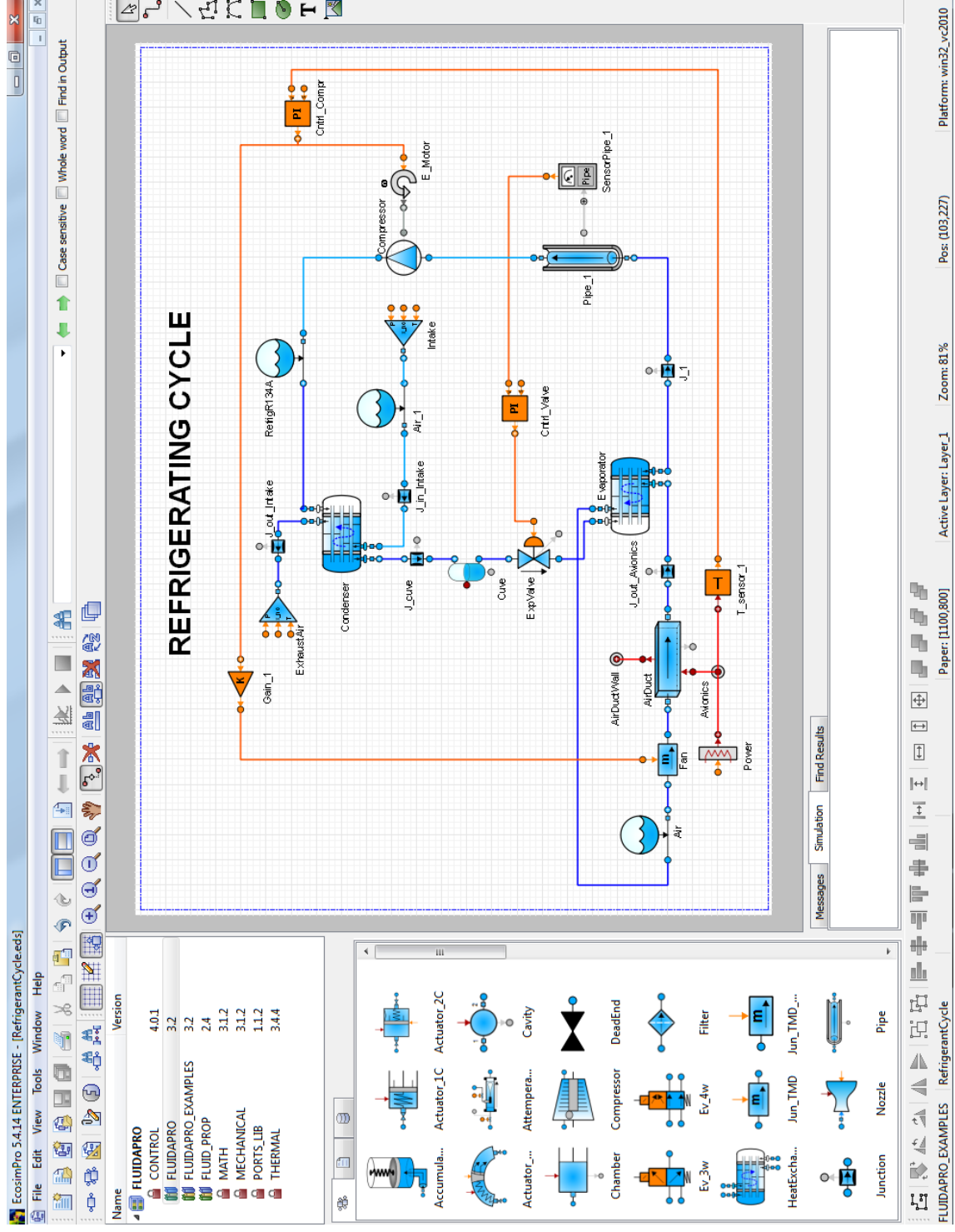


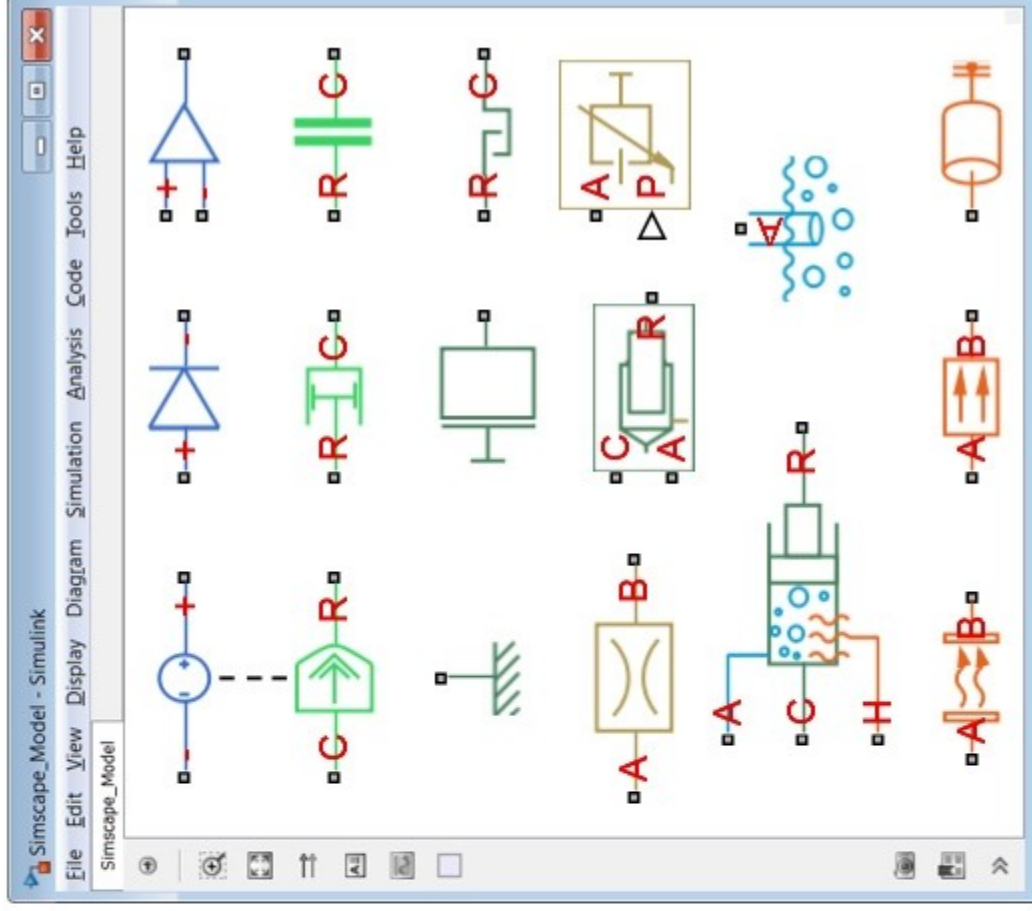
VHDL-AMS Multi Domain Design



EcosimPro

Modelling and Simulation Software





News

[EOOLT 2017](#)

The EOOLT workshop took successfully place in Munich, Germany on December 1.

Proceedings are now available on ACM Digital Library

Modelica Scalable

Test Suite

A new suite of scalable test models [can be found here](#).

Welcome to the EOOLT community!

This site is intended to be a meeting point for researchers and practitioners working in the area of equation-based object-oriented modeling languages and tools. The site's main purpose is to host the workshop pages for the EOOLT workshop series. Below you can find links to the current and past events, together with links to the open access workshop proceedings.

This site is maintained by [David Broman](#). If you have any questions or comments, please send an [email](#).



EOOLT 2017, December 1, Munich, Germany
8th International Workshop on Equation-Based
Object-Oriented Modeling Languages and Tools

[EOOLT 2017 Proceedings \(ACM Digital Library\)](#)

[Workshop site](#)



EOOLT 2016, April 18, Milano, Italy
7th International Workshop on Equation-Based
Object-Oriented Modeling Languages and Tools

[EOOLT 2016 Proceedings \(ACM Digital Library\)](#)

[Workshop site \(archived\)](#)



EOOLT 2014, Berlin, Germany
6th International Workshop on Equation-Based
Object-Oriented Modeling Languages and Tools

[EOOLT 2014 Proceedings \(ACM Digital Library\)](#)

[Workshop site \(archived\)](#)

- Modelica
- User's Guide
- Blocks
- Mechanics
- Fluid
- Electrical
- Analog
- Examples
- Basic
- Ground
- Resistor
- Conductor
- Capacitor
- Inductor
- SaturatingInductor
- Transformer
- M_Transformer
- Gyrator

model Capacitor "Ideal linear electrical capacitor";

parameter SI.Capacitance C "Capacitance";

Interfaces.PositivePin p;

Interfaces.NegativePin n;

SI.Voltage v "Voltage drop between pins";

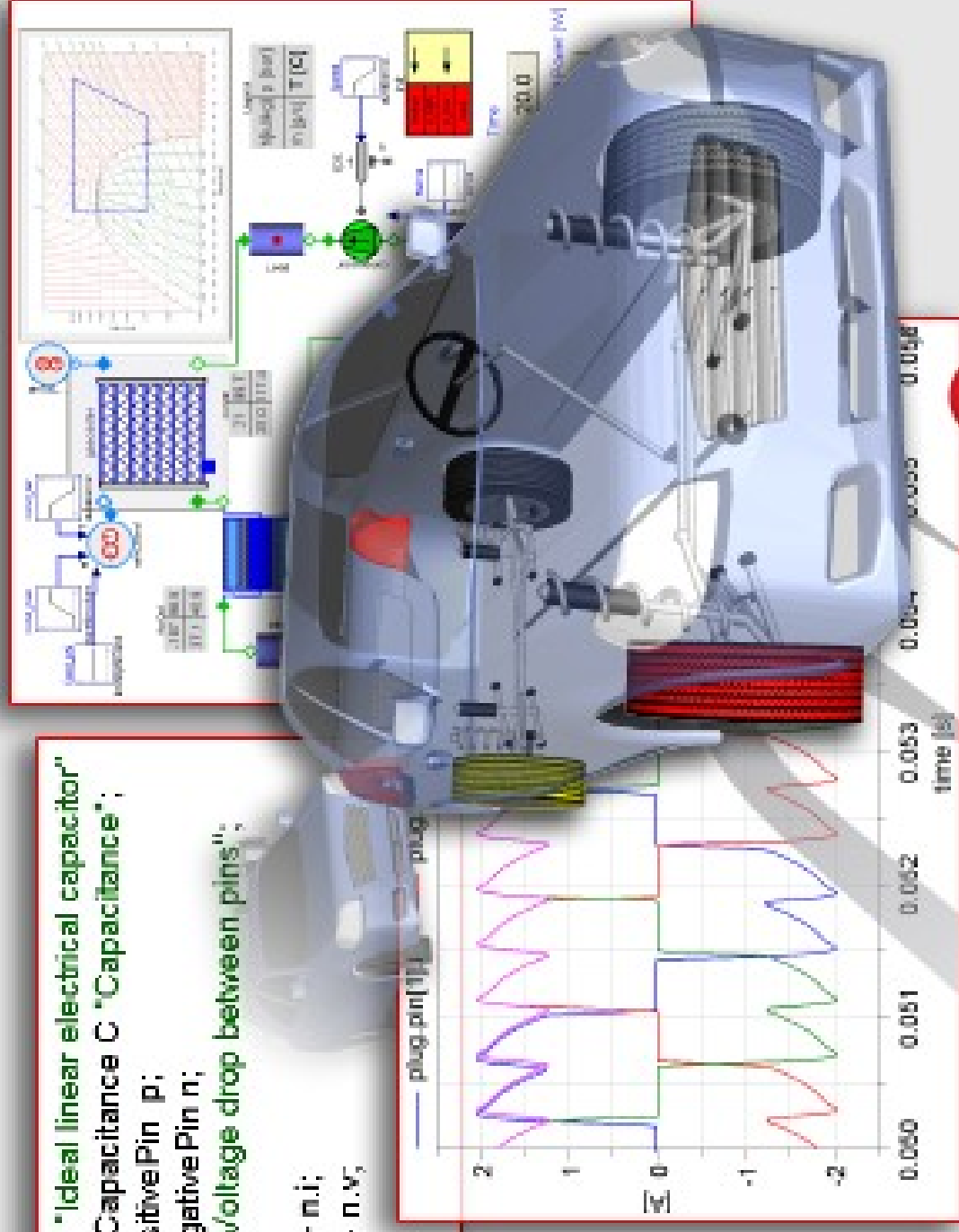
equation

$$0 = p.i + n.i;$$

$$v = p.v - n.v;$$

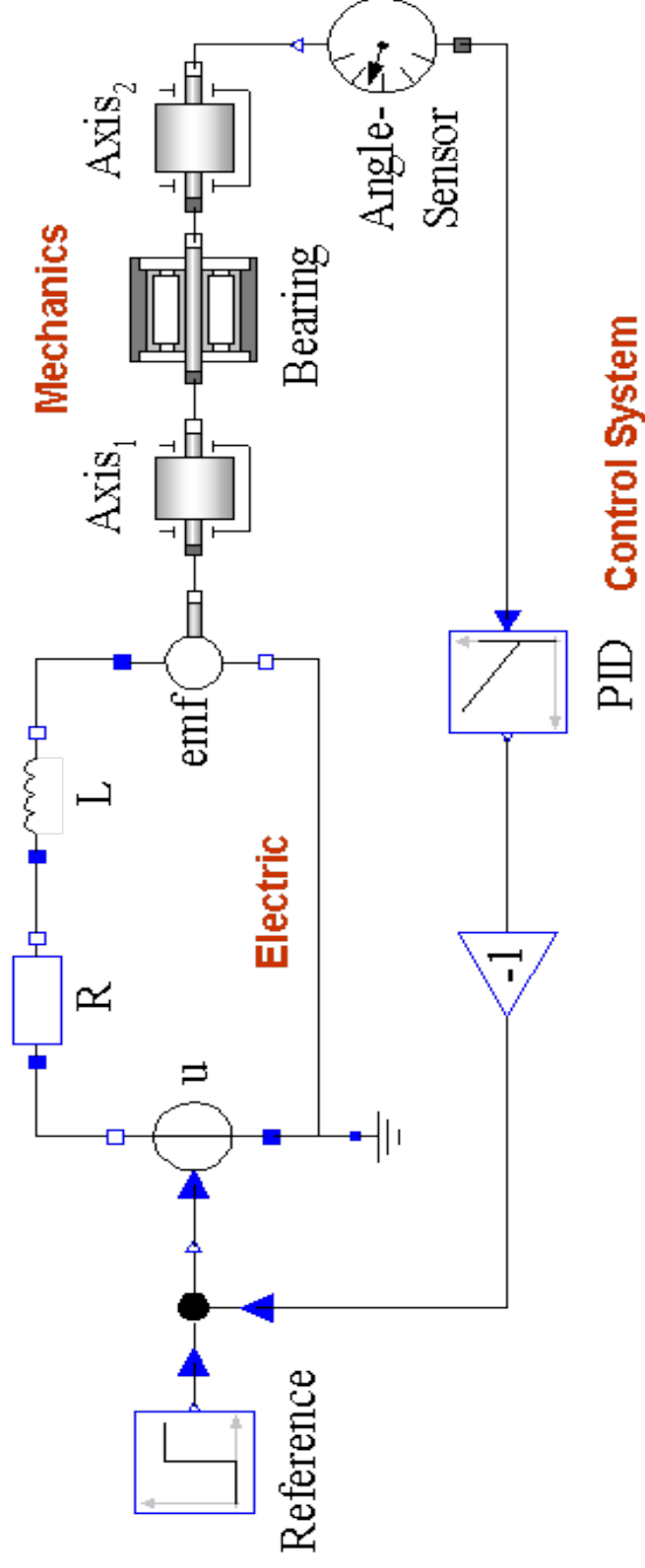
$$C \cdot \text{der}(v) = p.i;$$

end Capacitor;

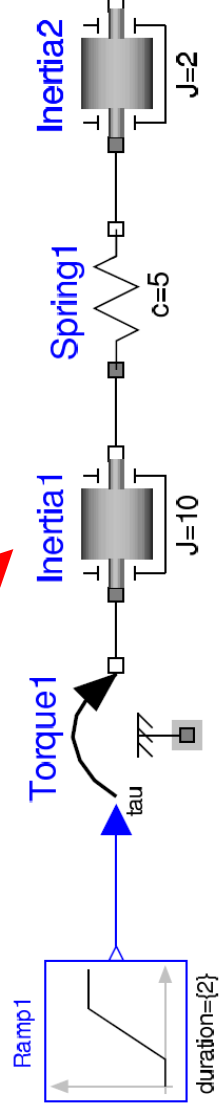


MODELICA

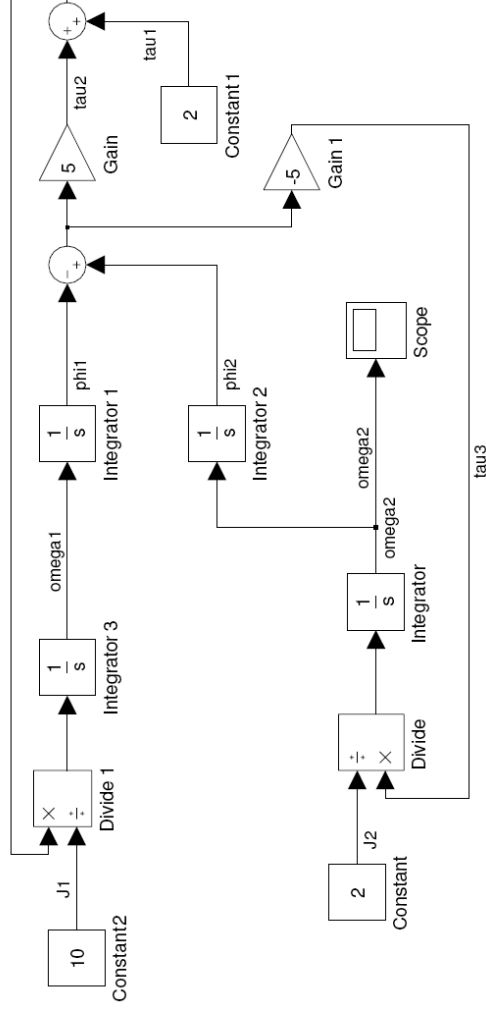
<http://www.modelica.org>



Keeps the physical structure



Acausal model (Modelica)



Causal block-based model (Simulink)

- Model exchange/re-use standard (Modelica Association)
- Modelica Standard Library (MSL)
- Object-oriented, hierarchical; semantics based on flattening
- Computationally a-causal modelling; semantics based on DAEs
- Originated in Hilding Elmquist's 1978 PhD thesis @ Lund
- Early 1990's: Modelica Design Team (started in SiE)

- hybrid (discrete-time/discrete-event) constructs (e.g., used to model network protocols based on TrueTime <http://www.control.lth.se/truetime/>)
- Limited support for Dynamic Structure models (i.e., no “agents”)
- Separate model from its (numerical) solution ...
- Generate Functional Mockup Interface (FMI) compliant simulation units
- Currently: many commercial and open (e.g., OpenModelica) tools
- Related: Mathworks Simscape, EcosimPro, NMF, gProms, ...

Electrical Types

```
type Time = Real (final quantity="Time", final unit="s");
type ElectricPotential = Real (final quantity="ElectricPotential",
                               final unit="V");
type Voltage = ElectricPotential;
type ElectricCurrent = Real (final quantity="ElectricCurrent",
                             final unit="A");
type Current = ElectricCurrent;
```

Beware: variables are **signals** (functions of **time**)!

Libraries

VolumeDensityOfCharge

SurfaceDensityOfCharge

ElectricFieldStrength

ElectricPotential

Modelica Text View

C:/OpenModelica1.9.1Beta2/lib/omlibrary/Modelica 3.2.1/Stunits.mo

Line: 1, Col: 0

1 type ElectricPotential = Real(final quantity = "ElectricPotential", final unit = "V");

Libraries

VolumeDensityOfCharge

SurfaceDensityOfCharge

ElectricFieldStrength

ElectricPotential

Voltage

Modelica Text View

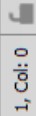
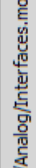
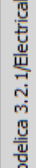
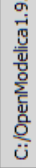
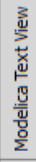
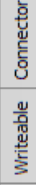
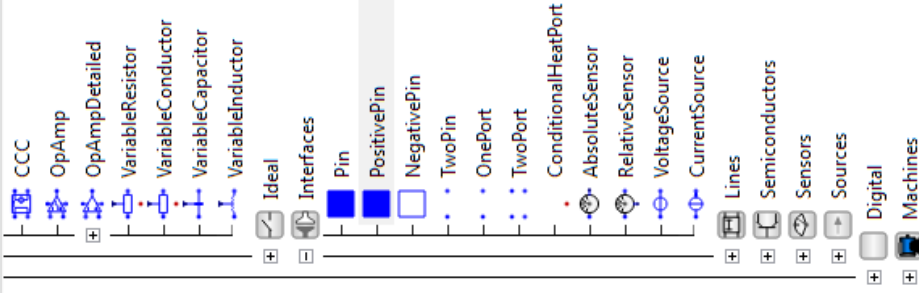
C:/Open...nts.mo

Line: 1, Col: 0

1 type Voltage = ElectricPotential;

Electrical Pin Interface

```
connector PositivePin "Positive pin of an electric component"  
    Voltage v "Potential at the pin";  
    flow Current i "Current flowing into the pin";  
end PositivePin;
```



```
1 connector PositivePin "Positive pin of an electric component"  
2 Modelica.SUnits.Voltage v "Potential at the pin" annotation(unassignedMessage = "An electrical  
3 potential cannot be uniquely calculated."  
4 The reason could be that  
5 - a ground object is missing (Modelica.Electrical.Analog.Basic.Ground)  
6 - to define the zero potential of the electrical circuit, or  
7 - a connector of an electrical component is not connected.");  
8 flow Modelica.SUnits.Current i "Current flowing into the pin" annotation(unassignedMessage = "An  
9 electrical current cannot be uniquely calculated."  
10 The reason could be that  
11 - a ground object is missing (Modelica.Electrical.Analog.Basic.Ground)  
12 - to define the zero potential of the electrical circuit, or  
13 - a connector of an electrical component is not connected.");  
14 annotation(defaultComponentName = "pin_p", Documentation(info = "<html>  
15 <p>Connectors PositivePin and NegativePin are nearly identical. The only difference is that the  
16 icons are different in order to identify more easily the pins of a component. Usually, connector  
17 PositivePin is used for the positive and connector NegativePin for the negative pin of an electrical  
18 component.</p>  
19 <ul>  
20 <li><i>1998 </i></li>  
21 <li><i>by Christoph Clauss<br>initially implemented<br></i></li>  
22 </ul>  
23 </html>"), Icon(coordinatesSystem(preserveAspectRatio = true, extent = {{-100,-100},{100,100}}),  
24 graphics = {Rectangle(extent = {{-100,100},{100,-100}}, lineColor = {0,0,255}, fillColor =  
25 {0,0,255}, fillPattern = FillPattern.Solid)), Diagram(coordinatesSystem(preserveAspectRatio = true,  
26 extent = {{-100,-100},{100,100}}, graphics = {Rectangle(extent = {{-40,40},{40,-40}}, lineColor =  
27 {0,0,255}, fillColor = {0,0,255}, fillPattern = FillPattern.Solid), Text(extent = {{-160,110},  
28 {40,50}}, lineColor = {0,0,255}, textString = "%name"))));  
29 end PositivePin;
```

Electrical Port

```
partial model OnePort
  "Component with two electrical pins p and n
   and current i from p to n"
  Voltage v "Voltage drop between the two pins (= p.v - n.v)";
  Current i "Current flowing from pin p to pin n";
  PositivePin p;
  NegativePin n;
equation
  v = p.v - n.v;
  0 = p.i + n.i;
  i = p.i;
end OnePort;
```

Libraries

CCC

OpAmp

OpAmpDetailed

VariableResistor

VariableConductor

VariableCapacitor

VariableInductor

Ideal

Interfaces

Pin

PositivePin

NegativePin

TwoPin

OnePort

TwoPort

ConditionalHeatPort

AbsoluteSensor

RelativeSensor

VoltageSource

CurrentSource

Lines

Semiconductors

Sensors

Sources

Digital

Machines

MultiPhase

Modelica Text View

Model

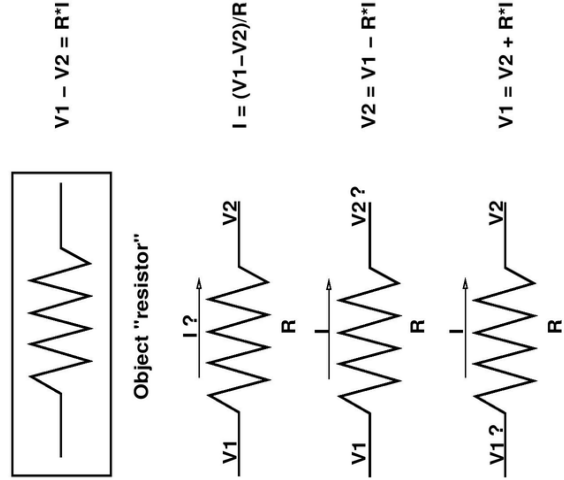
Writable

C:/OpenModelica1.9.1Beta2/lib/omlibrary/Modelica 3.2.1/Electrical/Analog/Interfaces.mo

Line: 1, Col: 0

```
1 partial model OnePort "Component with two electrical pins p and n and current i from p to n"
2 SI.Voltage v "Voltage drop between the two pins (= p.v - n.v)";
3 SI.Current i "Current flowing from pin p to pin n";
4 PositivePin p "Positive pin (potential p.v > n.v for positive voltage drop v)"
5 annotation(Placement(transformation(extent = {{-110,-10},{-90,10}}, rotation = 0)));
6 NegativePin n "Negative pin" annotation(Placement(transformation(extent = {{110,-10},{90,10}},
7 rotation = 0)));
8 equation
9 v = p.v - n.v;
10 0 = p.i + n.i;
11 i = p.i;
12 annotation(Documentation(info = "<html>
13 <p>Superclass of elements which have <b>two</b> electrical pins: the positive pin connector
14 <i>p</i>, and the negative pin connector <i>n</i>. It is assumed that the current flowing into pin p
15 is identical to the current flowing out of pin n. This current is provided explicitly as current
16 i.</p>
17 </html>", revisions = "<html>
18 <ul>
19 <li><i>1998</i>
20 by Christoph Clauss<br>initially implemented<br>
21 </li>
22 </ul>
23 </html>"), Diagram(coordinatesSystem(preserveAspectRatio = true, extent = {{-100,-100},{100,100}}),
24 graphics = {Line(points = {{-110,20},{-85,20}}, color = {160,160,164}),Polygon(points = {{-95,23},
25 {-85,20},{-95,17},{-95,23}}, lineColor = {160,160,164}, fillColor = {160,160,164}, fillPattern =
26 FillPattern.Solid),Line(points = {{90,20},{115,20}}, color = {160,160,164}),Line(points = {{-125,0},
27 {-115,0}}, color = {160,160,164}),Line(points = {{-120,-5},{-120,5}}, color =
28 {160,160,164}),Text(extent = {{-110,25},{-90,45}}, lineColor = {160,160,164}, textString =
29 "i"),Polygon(points = {{105,23},{115,20},{105,17},{105,23}}, lineColor = {160,160,164}, fillColor =
30 {160,160,164}, fillPattern = FillPattern.Solid),Line(points = {{115,0},{125,0}}, color =
31 {160,160,164}),Text(extent = {{90,45},{110,25}}, lineColor = {160,160,164}, textString = "i")));
32 end OnePort;
```

Object-oriented re-use and causality



Object "resistor"

Electrical Resistor

```

model Resistor "Ideal linear electrical resistor"
  extends OnePort;
  parameter Resistance R=1 "Resistance";
equation
  R*i = v;
end Resistor;
  
```



Modelica.Electrical.Analog.Basic.Resistor

myRLCNetwork* Model Modelica Text View C:/OpenModelica1.9.1Beta2/lib/omlibrary/Modelica 3.2.1/Electrical/Analog/Basic.mo

Line: 1, Col: 0

```

1 model Resistor "Ideal linear electrical resistor"
2   parameter Modelica.SIunits.Resistance R(start = 1) "Resistance at temperature T_ref";
3   parameter Modelica.SIunits.Temperature T_ref = 300.15 "Reference temperature";
4   parameter Modelica.SIunits.LinearTemperatureCoefficient alpha = 0 "Temperature coefficient of resistance
   (R_actual = R*(1 + alpha*(T_heatPort - T_ref)))";
5   extends Modelica.Electrical.Analog.Interfaces.OnePort;
6   extends Modelica.Electrical.Analog.Interfaces.ConditionalHeatPort(T = T_ref);
7   Modelica.SIunits.Resistance R_actual "Actual resistance = R*(1 + alpha*(T_heatPort - T_ref))";
8 equation
9   assert(1 + alpha * (T_heatPort - T_ref) >= Modelica.Constants.eps, "Temperature outside scope of model!");
10  R_actual = R * (1 + alpha * (T_heatPort - T_ref));
11  v = R_actual * i;
12  LossPower = v * i;
13  annotation(Documentation(info = "<html>
14    <p>The linear resistor connects the branch voltage <i>v</i> with the branch current <i>i</i> by <i>i*R = v</i>."
15    </p></html>", revisions = "<html>
16      <ul>
17        <li><i>August 07, 2009 </i>
18          by Anton Haumer<br> temperature dependency of resistance added<br>
19        </li>
20        <li><i>March 11, 2009 </i>
21          by Christoph Clauss<br> conditional heat port added<br>
22        </li>
23        <li><i>1998 </i>
24          by Christoph Clauss<br> initially implemented<br>
25        </li>
26      </ul>
27    </html>"), Icon(coordinatesSystem(preserveAspectRatio = true, extent = {{-100,-100},{100,100}}), graphics =
   {Rectangle(extent = {{-70,30},{70,-30}}), lineColor = {0,0,255}, fillColor = {255,255,255}, fillPattern =
   FillPattern.Solid),Line(points = {{-90,0},{-70,0}}, color = {0,0,255}),Line(points = {{70,0},{90,0}}, color =
   {0,0,255}),Text(extent = {{-144,-40},{142,-72}}, lineColor = {0,0,0}, textString = "R=$R"),Line(visible =
   useHeatPort, points = {{0,-100},{0,-30}}, color = {127,0,0}, smooth = Smooth.None, pattern =
   LinePattern.Dot),Text(extent = {{-152,87},{148,47}}, textString = "%name", lineColor = {0,0,255})),
   Diagram(coordinatesSystem(preserveAspectRatio = true, extent = {{-100,-100},{100,100}}), graphics =
   {Rectangle(extent = {{-70,30},{70,-30}}, lineColor = {0,0,255}),Line(points = {{-96,0},{-70,0}}, color =
   {0,0,255}),Line(points = {{70,0},{96,0}}, color = {0,0,255})}));
28 end Resistor;

```

X: -15.03 Y: 154.06

Welcome

Modeling

Plotting

Libraries

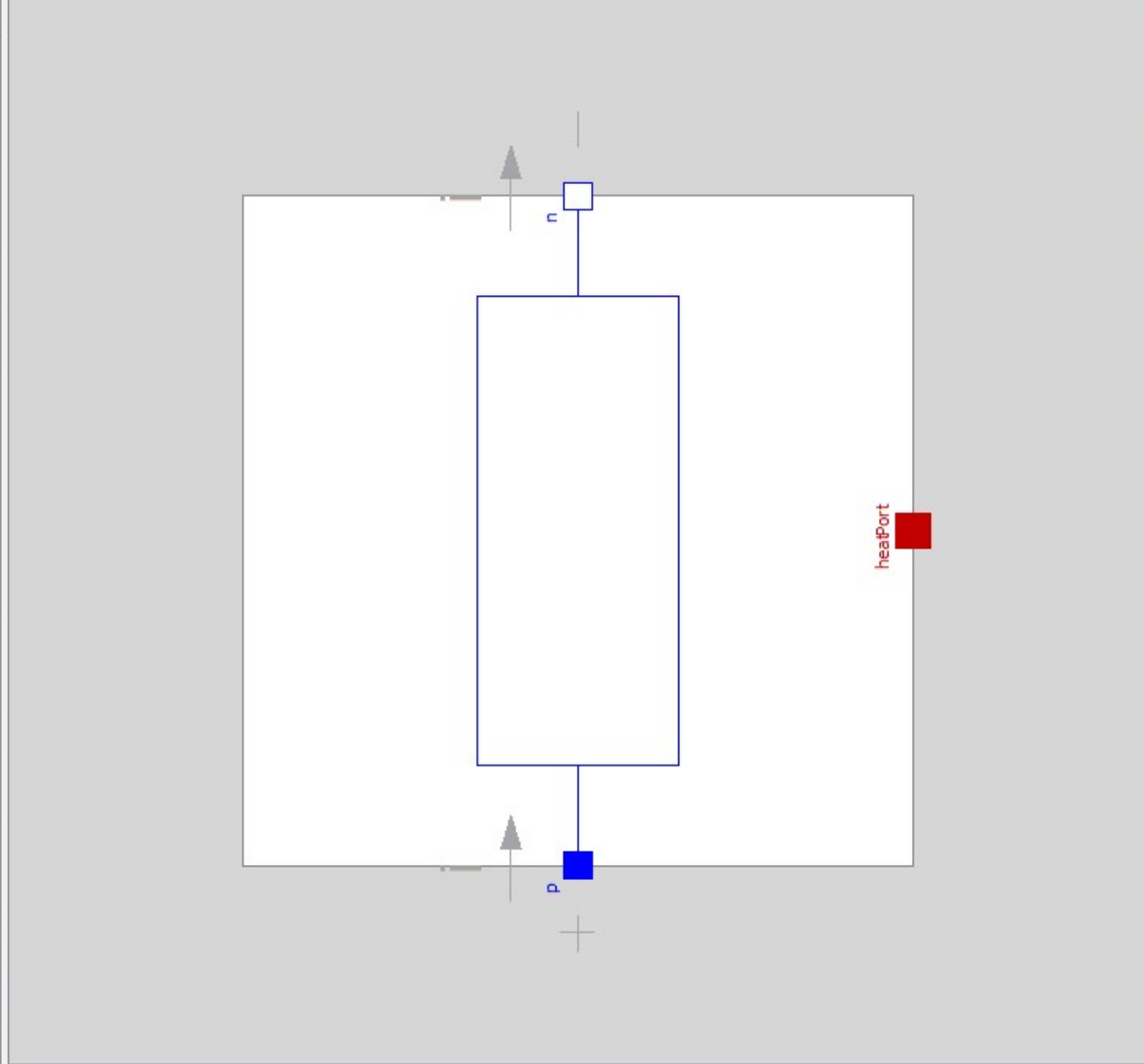
- Blocks
- ComplexBlocks
- StateGraph
- Electrical
 - Analogue
 - Examples
 - Basic
 - Ground
 - Resistor
 - HeatingResistor
 - Conductor
 - Capacitor
 - Inductor
 - SaturatingInductor
 - Transformer
 - M_Transformer
 - Gyrator
 - EMF
 - TranslationalEMF
 - VCV
 - VCC
 - CCV
 - CCC
 - OpAmp
 - OpAmpDetailed
 - VariableResistor
 - VariableConductor
 - VariableCapacitor
 - VariableInductor
 - Ideal
 - Interfaces
 - Lines
 - Semiconductors

myRLCnetwork*

Modelica.Electrical.Analog.Basic.Resistor

Diagram View

Line: 1, Col: 0

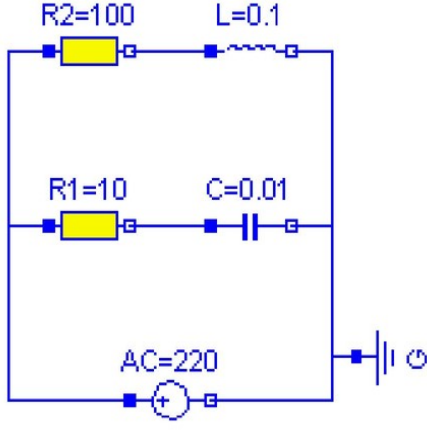


The circuit

```

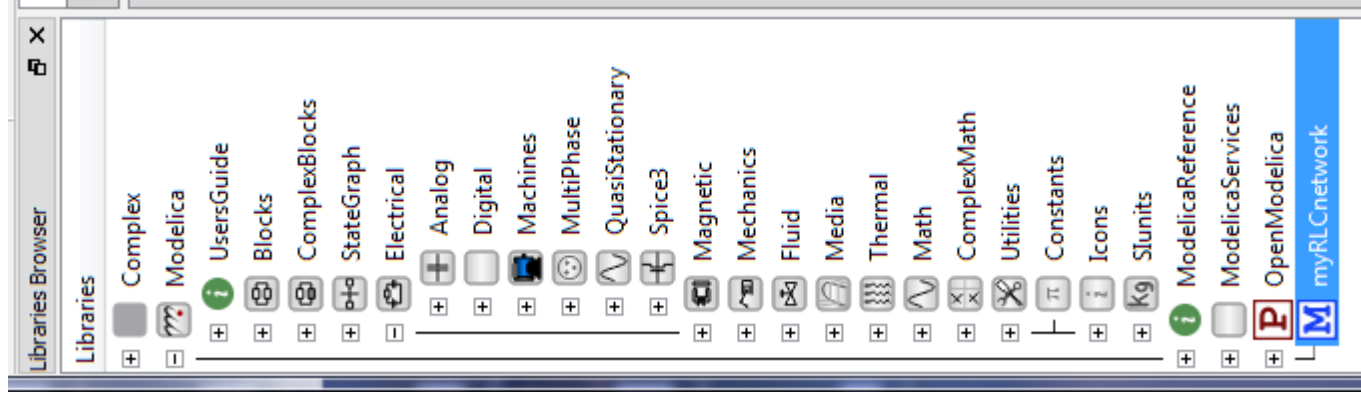
model circuit
  Resistor R1(R=10);
  Capacitor C(C=0.01);
  Resistor R2(R=100);
  Inductor L(L=0.1);
  VsourceAC AC;
  Ground G;
equation
  connect(AC.p, R1.p);
  connect(R1.n, C.p);
  connect(C.n, AC.n);
  connect(R1.p, R2.p);
  connect(R2.n, L.p);
  connect(L.n, C.n);
  connect(AC.n, G.p);
end circuit;

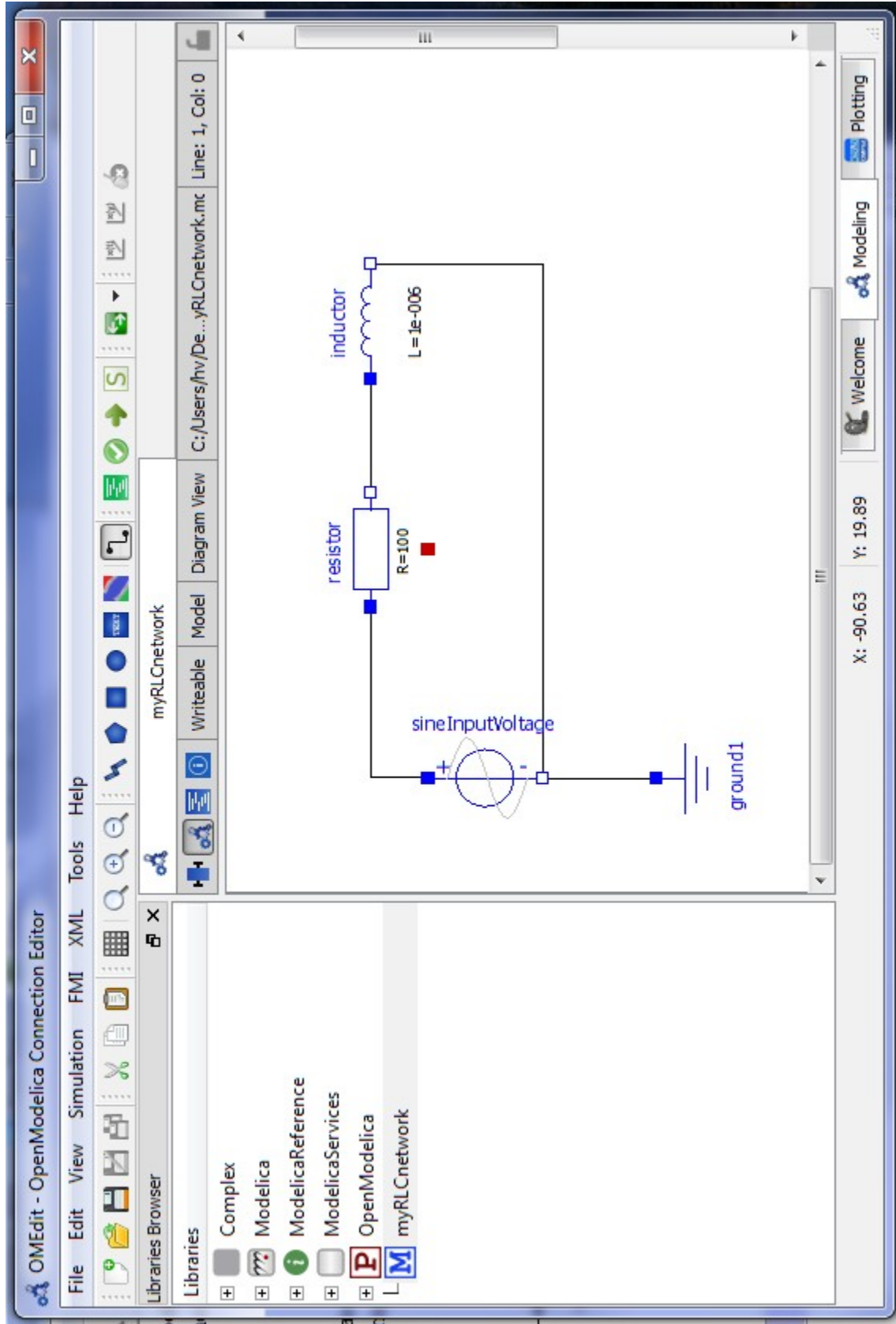
```



Meaning: set of Differential Algebraic Equations (DAEs) obtained by

1. expanding inheritance/instantiation
2. flattening hierarchy, unique names
3. expanding connect() into equations (across vs. flow)



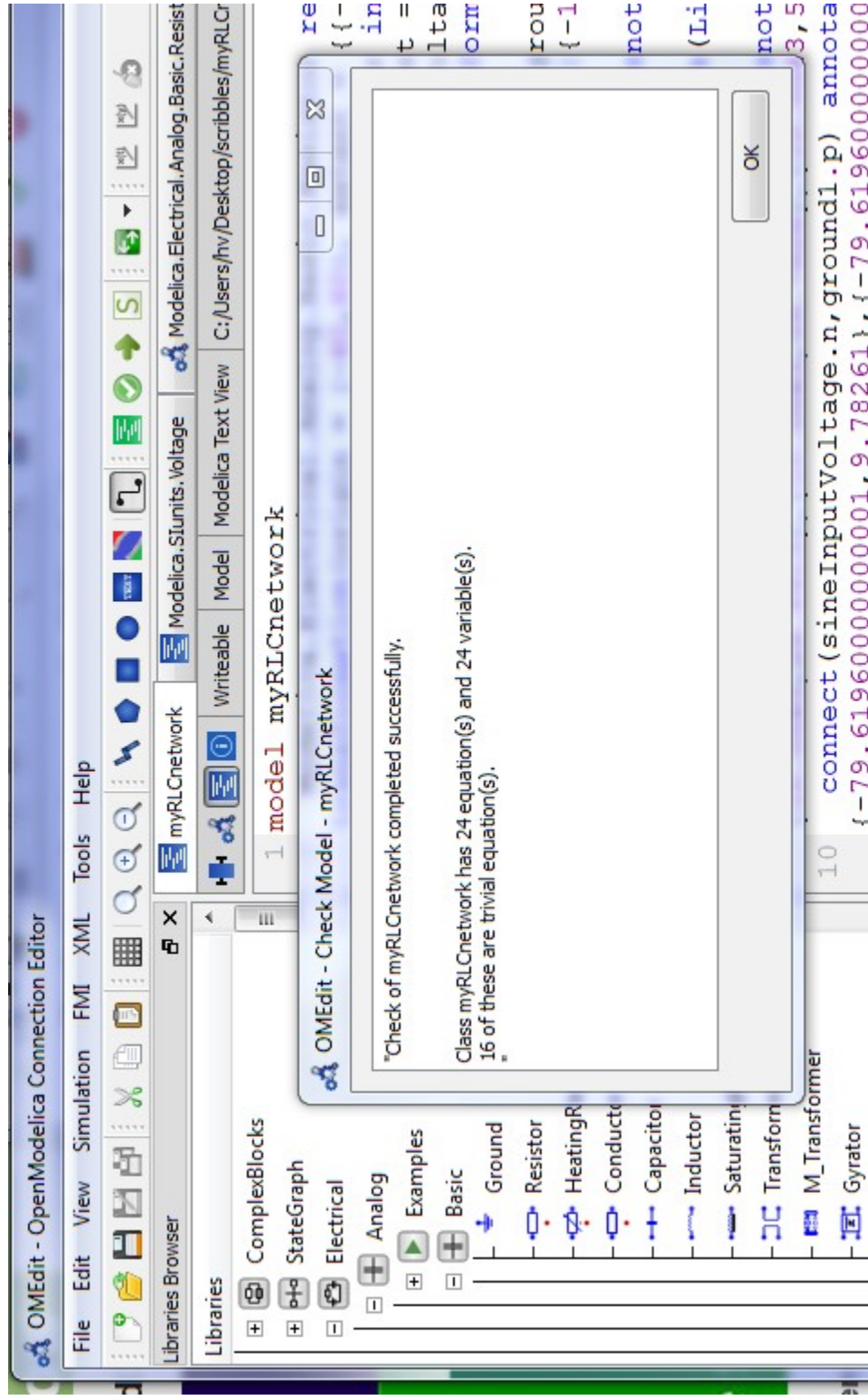


- Complex
- Modelica
- ModelicaReference
- ModelicaServices
- OpenModelica
- myRLCnetwork

```

1 model myRLCnetwork
2   Modelica.Electrical.Analog.Basic.Resistor(R = 100) annotation(Placement(visible
   = true, transformation(origin = {-40,60}, extent = {{-10,-10},{10,10}}, rotation = 0));
3   Modelica.Electrical.Analog.Basic.Inductor(inductor(L = 1e-006)
   annotation(Placement(visible = true, transformation(origin = {0,60}, extent = {{-10,-10},
   {10,10}}, rotation = 0));
4   Modelica.Electrical.Analog.Sources.SineVoltage sineInputVoltage(V = 10, freqHz = 50)
   annotation(Placement(visible = true, transformation(origin = {-80,40}, extent =
   {{-10,-10},{10,10}}, rotation = -90));
5   Modelica.Electrical.Analog.Basic.Ground ground1 annotation(Placement(visible = true,
   transformation(origin = {-80,0}, extent = {{-10,-10},{10,10}}, rotation = 0));
6 equation
7   connect(resistor.p,sineInputVoltage.p) annotation(Line(points = {{-50,60},{-79.8913,60},
   {-79.8913,49.4565},{-79.8913,49.4565}}));
8   connect(resistor.n,inductor.p) annotation(Line(points = {{-30,60},{-9.78261,60},
   {-9.78261,59.2391},{-9.78261,59.2391}}));
9   connect(sineInputVoltage.n,inductor.n) annotation(Line(points = {{-80,30},{-80,29.8913},
   {10.0543,29.8913},{10.0543,59.7826},{10.0543,59.7826}}));
10  connect(sineInputVoltage.n,ground1.p) annotation(Line(points = {{-80,30},{-80,9.78261},
   {-79.61960000000001,9.78261},{-79.61960000000001,9.78261}}));
11  annotation(Icon(coordinatesSystem(extent = {{-100,-100},{100,100}}, preserveAspectRatio =
   true, initialScale = 0.1, grid = {2,2}), Diagram(coordinatesSystem(extent = {{-100,-100},
   {100,100}}, preserveAspectRatio = true, initialScale = 0.1, grid = {2,2}));
12 end myRLCnetwork;

```



```

class myRLCnetwork
    Real resistor.v(quantity = "ElectricPotential", unit = "V") "Voltage drop between the two pins (= p.v - n.v)";
    Real resistor.i(quantity = "ElectricCurrent", unit = "A") "Current flowing from pin p to pin n";
    Real resistor.p.v(quantity = "ElectricPotential", unit = "V") "Potential at the pin";
    Real resistor.p.i(quantity = "ElectricCurrent", unit = "A") "Current flowing into the pin";
    Real resistor.n.v(quantity = "ElectricPotential", unit = "V") "Potential at the pin";
    Real resistor.n.i(quantity = "ElectricCurrent", unit = "A") "Current flowing into the pin";
    parameter Boolean resistor.useHeatPort = false "true, if HeatPort is enabled";
    Real resistor.LossPower(quantity = "Power", unit = "W") "Loss power leaving component via HeatPort";
    Real resistor.T_heatPort(quantity = "ThermodynamicTemperature", unit = "K", displayUnit = "degC", min = 0.0, start = 288.15, nominal = 300.0) "Temperature of HeatPort";
    parameter Real resistor.R(quantity = "Resistance", unit = "Ohm", start = 1.0) = 100.0 "Resistance at temperature T_ref";
    parameter Real resistor.T_ref(quantity = "ThermodynamicTemperature", unit = "K", displayUnit = "degC", min = 0.0, start = 288.15, nominal = 300.0) = 300.15 "Reference temperature";
    parameter Real resistor.alpha(quantity = "LinearTemperatureCoefficient", unit = "1/K") = 0.0 "Temperature coefficient of resistance (R_actual = R*(1 + alpha*(T_heatPort - T_ref)))";
    Real resistor.R_actual(quantity = "Resistance", unit = "Ohm") "Actual resistance = R*(1 + alpha*(T_heatPort - T_ref))";
    parameter Real resistor.T(quantity = "ThermodynamicTemperature", unit = "K", displayUnit = "degC", min = 0.0, start = 288.15, nominal = 300.0) = resistor.T_ref "Fixed device temperature if useHeatPort = false";
    Real inductor.v(quantity = "ElectricPotential", unit = "V") "Voltage drop between the two pins (= p.v - n.v)";
    Real inductor.i(quantity = "ElectricCurrent", unit = "A", start = 0.0) "Current flowing from pin p to pin n";
    Real inductor.p.v(quantity = "ElectricPotential", unit = "V") "Potential at the pin";
    Real inductor.p.i(quantity = "ElectricCurrent", unit = "A") "Current flowing into the pin";
    Real inductor.n.v(quantity = "ElectricPotential", unit = "V") "Potential at the pin";
    Real inductor.n.i(quantity = "ElectricCurrent", unit = "A") "Current flowing into the pin";
    parameter Real inductor.L(quantity = "Inductance", unit = "H", start = 1.0) = 1e-006 "Inductance";
    Real sineInputVoltage.v(quantity = "ElectricPotential", unit = "V") "Voltage drop between the two pins (= p.v - n.v)";
    Real sineInputVoltage.i(quantity = "ElectricCurrent", unit = "A") "Current flowing from pin p to pin n";
    Real sineInputVoltage.p.v(quantity = "ElectricPotential", unit = "V") "Potential at the pin";
    Real sineInputVoltage.p.i(quantity = "ElectricCurrent", unit = "A") "Current flowing into the pin";
    Real sineInputVoltage.n.v(quantity = "ElectricPotential", unit = "V") "Potential at the pin";
    Real sineInputVoltage.n.i(quantity = "ElectricCurrent", unit = "A") "Current flowing into the pin";
    parameter Real sineInputVoltage.offset(quantity = "ElectricPotential", unit = "V") = 0.0 "Voltage offset";
    parameter Real sineInputVoltage.startTime(quantity = "Time", unit = "s") = 0.0 "Time offset";
    parameter Real sineInputVoltage.phase(quantity = "ElectricPotential", unit = "V", start = 1.0) = 10.0 "Amplitude of sine wave";
    parameter Real sineInputVoltage.phase(quantity = "Angle", unit = "rad", displayUnit = "deg") = 0.0 "Phase of sine wave";
    parameter Real sineInputVoltage.freqHz(quantity = "Frequency", unit = "Hz", start = 1.0) = 50.0 "Frequency of sine wave";
    output Real sineInputVoltage.signalSource.y "Connector of Real output signal";
    parameter Real sineInputVoltage.signalSource.amplitude = sineInputVoltage.V "Amplitude of sine wave";
    parameter Real sineInputVoltage.signalSource.freqHz(quantity = "Frequency", unit = "Hz", start = 1.0) = sineInputVoltage.freqHz
    "Frequency of sine wave";
    parameter Real sineInputVoltage.signalSource.phase(quantity = "Angle", unit = "rad", displayUnit = "deg") = sineInputVoltage.phase
    "Phase of sine wave";
    parameter Real sineInputVoltage.signalSource.offset = sineInputVoltage.offset "Offset of output signal";
    parameter Real sineInputVoltage.signalSource.startTime(quantity = "Time", unit = "s") = sineInputVoltage.startTime "Output = offset for
    time < startTime";
    protected constant Real sineInputVoltage.signalSource.pi = 3.141592653589793;
    Real ground1.p.v(quantity = "ElectricPotential", unit = "V") "Potential at the pin";
    Real ground1.p.i(quantity = "ElectricCurrent", unit = "A") "Current flowing into the pin";

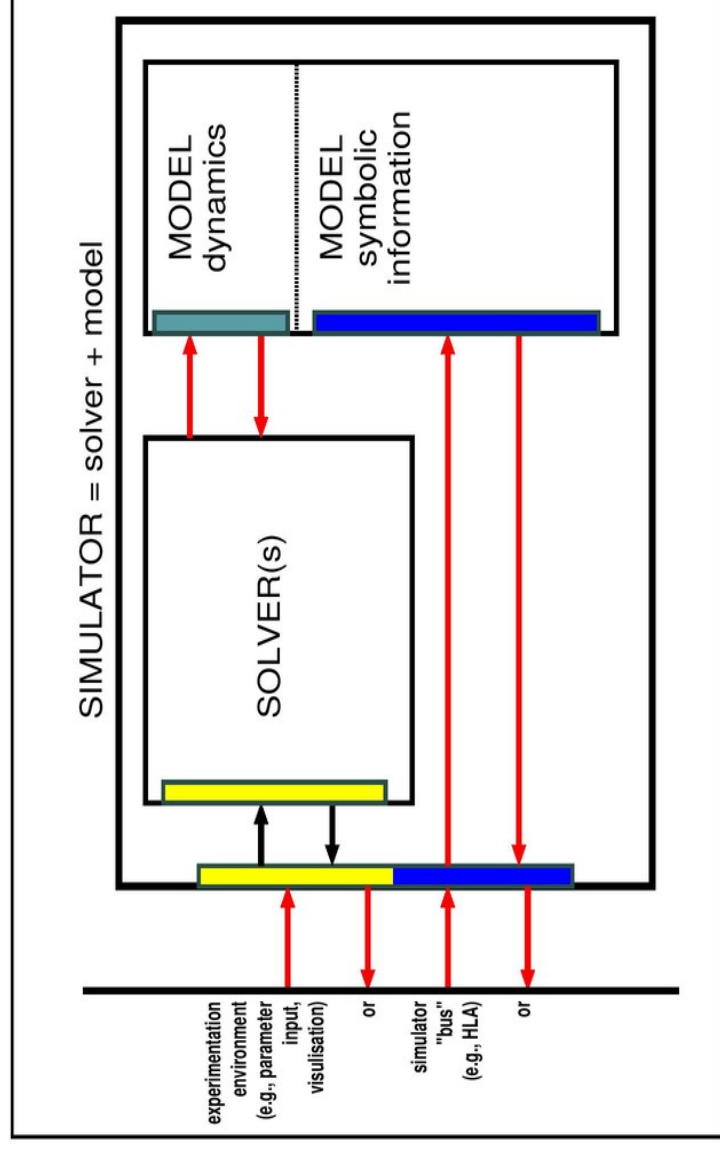
```

equation

```
assert(1.0 + resistor.alpha * (resistor.T_heatPort - resistor.T_ref) >= 1e-015, "Temperature outside scope of model!");
resistor.R_actual = resistor.R * (1.0 + resistor.alpha * (resistor.T_heatPort - resistor.T_ref));
resistor.v = resistor.R_actual * resistor.i;
resistor.LossPower = resistor.v * resistor.i;
resistor.v = resistor.p.v - resistor.n.v;
0.0 = resistor.p.i + resistor.n.i;
resistor.i = resistor.p.i;
resistor.T_heatPort = resistor.T;
inductor.L * der(inductor.i) = inductor.v;
inductor.v = inductor.p.v - inductor.n.v;
0.0 = inductor.p.i + inductor.n.i;
inductor.i = inductor.p.i;
sineInputVoltage.signalSource.y = sineInputVoltage.signalSource.offset + (if time <
sineInputVoltage.signalSource.startTime then 0.0 else sineInputVoltage.signalSource.amplitude * sin(6.283185307179586 *
sineInputVoltage.signalSource.freqHz * (time - sineInputVoltage.signalSource.startTime) +
sineInputVoltage.signalSource.phase));
sineInputVoltage.v = sineInputVoltage.signalSource.y;
sineInputVoltage.v = sineInputVoltage.p.v - sineInputVoltage.n.v;
0.0 = sineInputVoltage.p.i + sineInputVoltage.n.i;
sineInputVoltage.i = sineInputVoltage.p.i;
ground1.p.v = 0.0;
resistor.p.i + sineInputVoltage.p.i = 0.0;
resistor.n.i + inductor.p.i = 0.0;
inductor.n.i + sineInputVoltage.n.i + ground1.p.i = 0.0;
resistor.p.v = sineInputVoltage.p.v;
inductor.p.v = resistor.n.v;
ground1.p.v = inductor.n.v;
ground1.p.v = sineInputVoltage.n.v;
end myRLCnetwork;
```

Model-Solver Interface

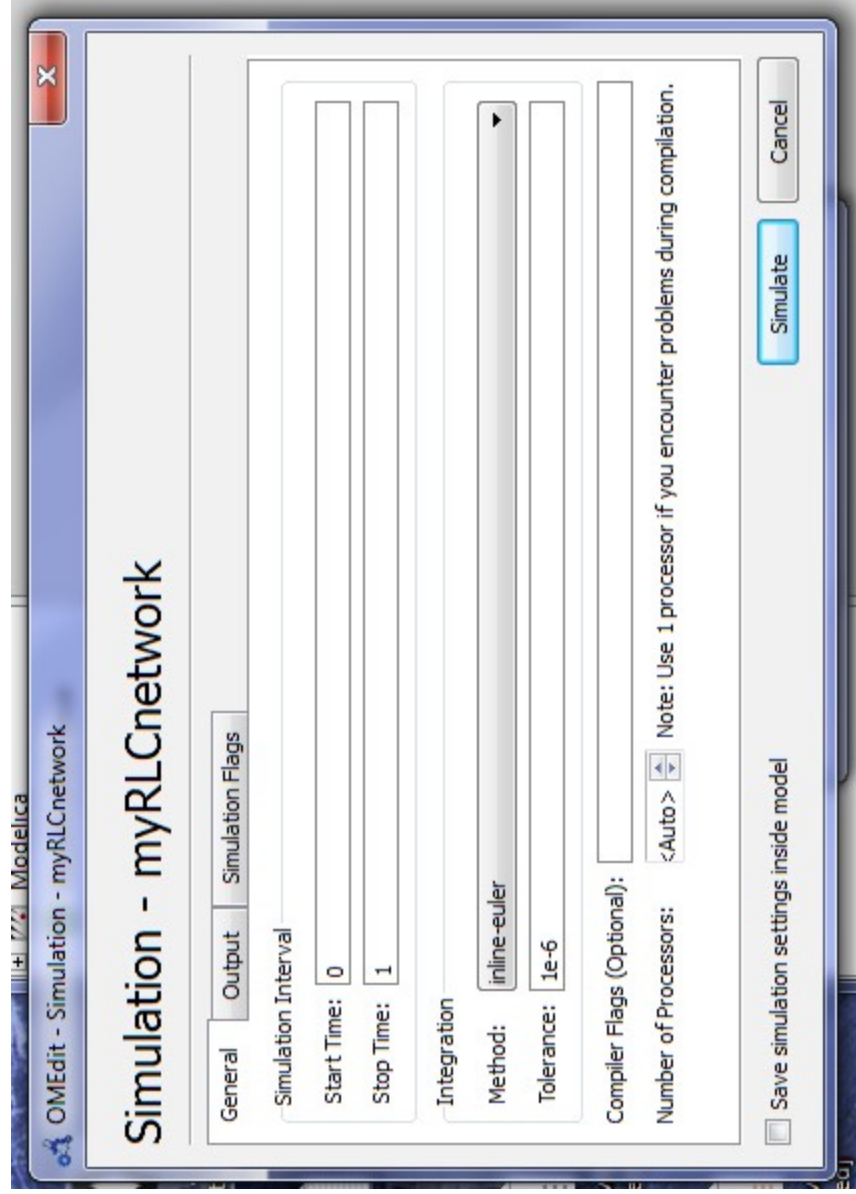
Simulator-Environment Interface



Output Compilation

```
"C:\OpenModelica1.9.1Beta2\MinGW\bin\mingw32-make.exe" -j4 -f myRLCnetwork.makefile
gcc -falign-functions -mssse2 -mfpmath=sse -I"C:/OpenModelica1.9.1Beta2//include/omc/c" -I. -DOPENMODELICA_XML_FROM_FILE_AT_RUNTIME -c -o
myRLCnetwork.o myRLCnetwork.c
gcc -falign-functions -mssse2 -mfpmath=sse -I"C:/OpenModelica1.9.1Beta2//include/omc/c" -I. -DOPENMODELICA_XML_FROM_FILE_AT_RUNTIME -c -o
myRLCnetwork_functions.o myRLCnetwork_functions.c
gcc -falign-functions -mssse2 -mfpmath=sse -I"C:/OpenModelica1.9.1Beta2//include/omc/c" -I. -DOPENMODELICA_XML_FROM_FILE_AT_RUNTIME -c -o
myRLCnetwork_records.o myRLCnetwork_records.c
gcc -falign-functions -mssse2 -mfpmath=sse -I"C:/OpenModelica1.9.1Beta2//include/omc/c" -I. -DOPENMODELICA_XML_FROM_FILE_AT_RUNTIME -c -o
myRLCnetwork_0lexo.o myRLCnetwork_0lexo.c
gcc -falign-functions -mssse2 -mfpmath=sse -I"C:/OpenModelica1.9.1Beta2//include/omc/c" -I. -DOPENMODELICA_XML_FROM_FILE_AT_RUNTIME -c -o
myRLCnetwork_02nls.o myRLCnetwork_02nls.c
gcc -falign-functions -mssse2 -mfpmath=sse -I"C:/OpenModelica1.9.1Beta2//include/omc/c" -I. -DOPENMODELICA_XML_FROM_FILE_AT_RUNTIME -c -o
myRLCnetwork_03lsy.o myRLCnetwork_03lsy.c
gcc -falign-functions -mssse2 -mfpmath=sse -I"C:/OpenModelica1.9.1Beta2//include/omc/c" -I. -DOPENMODELICA_XML_FROM_FILE_AT_RUNTIME -c -o
myRLCnetwork_04set.o myRLCnetwork_04set.c
gcc -falign-functions -mssse2 -mfpmath=sse -I"C:/OpenModelica1.9.1Beta2//include/omc/c" -I. -DOPENMODELICA_XML_FROM_FILE_AT_RUNTIME -c -o
myRLCnetwork_05evt.o myRLCnetwork_05evt.c
gcc -falign-functions -mssse2 -mfpmath=sse -I"C:/OpenModelica1.9.1Beta2//include/omc/c" -I. -DOPENMODELICA_XML_FROM_FILE_AT_RUNTIME -c -o
myRLCnetwork_06inz.o myRLCnetwork_06inz.c
gcc -falign-functions -mssse2 -mfpmath=sse -I"C:/OpenModelica1.9.1Beta2//include/omc/c" -I. -DOPENMODELICA_XML_FROM_FILE_AT_RUNTIME -c -o
myRLCnetwork_07dly.o myRLCnetwork_07dly.c
gcc -falign-functions -mssse2 -mfpmath=sse -I"C:/OpenModelica1.9.1Beta2//include/omc/c" -I. -DOPENMODELICA_XML_FROM_FILE_AT_RUNTIME -c -o
myRLCnetwork_08bnd.o myRLCnetwork_08bnd.c
myRLCnetwork_05evt.c: In function 'myRLCnetwork_zeroCrossingDescription':
myRLCnetwork_05evt.c:51: warning: assignment discards qualifiers from pointer target type
gcc -falign-functions -mssse2 -mfpmath=sse -I"C:/OpenModelica1.9.1Beta2//include/omc/c" -I. -DOPENMODELICA_XML_FROM_FILE_AT_RUNTIME -c -o
myRLCnetwork_09alg.o myRLCnetwork_09alg.c
gcc -falign-functions -mssse2 -mfpmath=sse -I"C:/OpenModelica1.9.1Beta2//include/omc/c" -I. -DOPENMODELICA_XML_FROM_FILE_AT_RUNTIME -c -o
myRLCnetwork_10asr.o myRLCnetwork_10asr.c
gcc -falign-functions -mssse2 -mfpmath=sse -I"C:/OpenModelica1.9.1Beta2//include/omc/c" -I. -DOPENMODELICA_XML_FROM_FILE_AT_RUNTIME -c -o
myRLCnetwork_11mix.o myRLCnetwork_11mix.c
gcc -falign-functions -mssse2 -mfpmath=sse -I"C:/OpenModelica1.9.1Beta2//include/omc/c" -I. -DOPENMODELICA_XML_FROM_FILE_AT_RUNTIME -c -o
myRLCnetwork_12jac.o myRLCnetwork_12jac.c
gcc -falign-functions -mssse2 -mfpmath=sse -I"C:/OpenModelica1.9.1Beta2//include/omc/c" -I. -DOPENMODELICA_XML_FROM_FILE_AT_RUNTIME -c -o
myRLCnetwork_13opt.o myRLCnetwork_13opt.c
gcc -falign-functions -mssse2 -mfpmath=sse -I"C:/OpenModelica1.9.1Beta2//include/omc/c" -I. -DOPENMODELICA_XML_FROM_FILE_AT_RUNTIME -c -o
myRLCnetwork_14lnz.o myRLCnetwork_14lnz.c
gcc -I. -o myRLCnetwork.exe myRLCnetwork_functions.o myRLCnetwork_records.o myRLCnetwork_06inz.o myRLCnetwork_07dly.o myRLCnetwork_08bnd.o myRLCnetwork_09alg.o
myRLCnetwork_10asr.o myRLCnetwork_11mix.o myRLCnetwork_12jac.o myRLCnetwork_13opt.o myRLCnetwork_14lnz.o -falign-functions -mssse2 -mfpmath=sse
-I"C:/OpenModelica1.9.1Beta2//include/omc/c" -I. -DOPENMODELICA_XML_FROM_FILE_AT_RUNTIME -Wl,--stack,0x2000000,-rpath,"C:/OpenModelica1.9.1Beta2//lib/omc" -Wl,-
rpath,"C:/OpenModelica1.9.1Beta2//lib/omc" -I"C:/OpenModelica1.9.1Beta2//lib" -lregex -lexpat -lgc -lpthread -lposix -lstdc++ -lstdc++ -lstdc++ -lstdc++
libgcc -luuid -loleaut32 -lole32 -lws2_32 -lsundials_kinsol -lsundials_nvecserial -lipopt -lcoinmumps -lcoinmetis -lphthread -lm -lgfortranbegin -
lgfortran -lmingw32 -lgcc_eh -lmoeldname -lmingwex -lmsvcrt -luser32 -lkernel32 -ladvapi32 -lshell32 -llapack-mingw -ltmglib-mingw -lblas-mingw -
lf2c -linteractive -lwsock32 -llis -lstdc++
```

	29/09/20...	C File	myRLCnetwork	14 KB
	29/09/20...	Application	myRLCnetwork	9,960 KB
	29/09/20...	LIBS File	myRLCnetwork.libs	0 KB
	29/09/20...	Text Document	myRLCnetwork	0 KB
	29/09/20...	MAKEFILE File	myRLCnetwork.makefile	2 KB
	29/09/20...	O File	myRLCnetwork.o	17 KB
	29/09/20...	C File	myRLCnetwork_01exo	2 KB
	29/09/20...	O File	myRLCnetwork_01exo.o	2 KB
	29/09/20...	C File	myRLCnetwork_02nls	2 KB
	29/09/20...	O File	myRLCnetwork_02nls.o	1 KB
	29/09/20...	C File	myRLCnetwork_03lsy	2 KB
	29/09/20...	O File	myRLCnetwork_03lsy.o	1 KB
	29/09/20...	C File	myRLCnetwork_04set	2 KB
	29/09/20...	O File	myRLCnetwork_04set.o	1 KB
	29/09/20...	C File	myRLCnetwork_05evt	3 KB
	29/09/20...	O File	myRLCnetwork_05evt.o	2 KB
	29/09/20...	C File	myRLCnetwork_06inz	7 KB
	29/09/20...	O File	myRLCnetwork_06inz.o	5 KB
	29/09/20...	C File	myRLCnetwork_07dly	2 KB
	29/09/20...	O File	myRLCnetwork_07dly.o	1 KB
	29/09/20...	C File	myRLCnetwork_08bnd	7 KB
	29/09/20...	O File	myRLCnetwork_08bnd.o	5 KB
	29/09/20...	C File	myRLCnetwork_09alg	2 KB
	29/09/20...	O File	myRLCnetwork_09alg.o	1 KB
	29/09/20...	C File	myRLCnetwork_10asr	2 KB
	29/09/20...	O File	myRLCnetwork_10asr.o	1 KB
	29/09/20...	C File	myRLCnetwork_11mix	2 KB
	29/09/20...	H File	myRLCnetwork_11mix.h	0 KB
	29/09/20...	O File	myRLCnetwork_11mix.o	1 KB
	29/09/20...	C File	myRLCnetwork_12jac	4 KB
	29/09/20...	H File	myRLCnetwork_12jac.h	2 KB

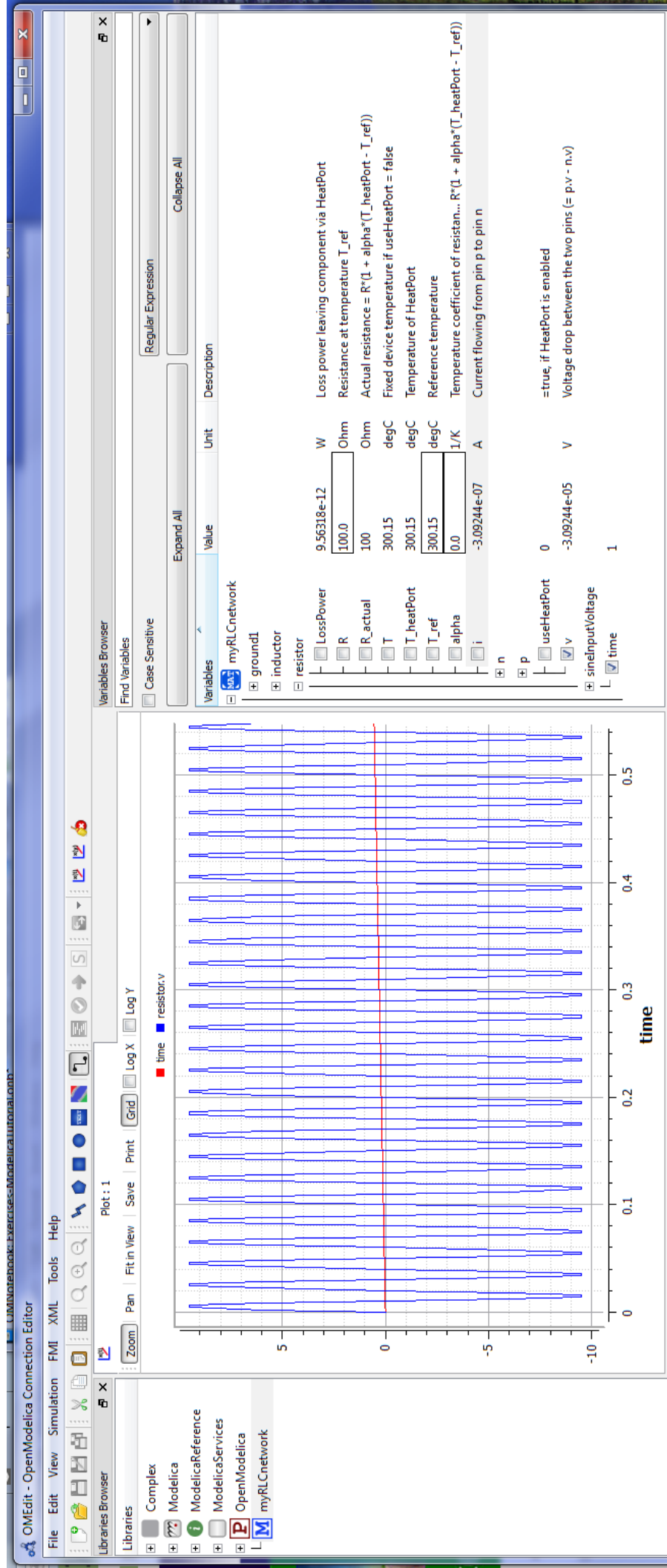


OMEdit - myRLCnetwork Simulation Output

Output Compilation

```

C:/Users/hv/AppData/Local/Temp/OpenModelica/OMEdit/myRLCnetwork.exe -port=49502 -logFormat=xml -w -lv=LOG_STATS
LOG_STATS | info | ## STATISTICS ##
LOG_STATS | info | timer |
| | | 0.0150538s [ 46.9%] pre-initialization
| | | 4.18139e-005s [ 0.1%] initialization
| | | 2.0907e-005s [ 0.1%] steps
| | | 0.0157118s [ 49.0%] creating output-file
| | | 0.000115588s [ 0.4%] event-handling
| | | 0.000295738s [ 0.9%] overhead
| | | 0.000824114s [ 2.6%] simulation
| | | 0.0320637s [100.0%] total
LOG_STATS | info | events |
| | | 0 state events
| | | 0 time events
LOG_STATS | info | solver: DASSL |
| | | 2431 steps taken
| | | 3266 calls of functionODE
| | | 165 evaluations of jacobian
| | | 73 error test failures
| | | 0 convergence test failures
LOG_STATS | info | ## END STATISTICS ##
    
```



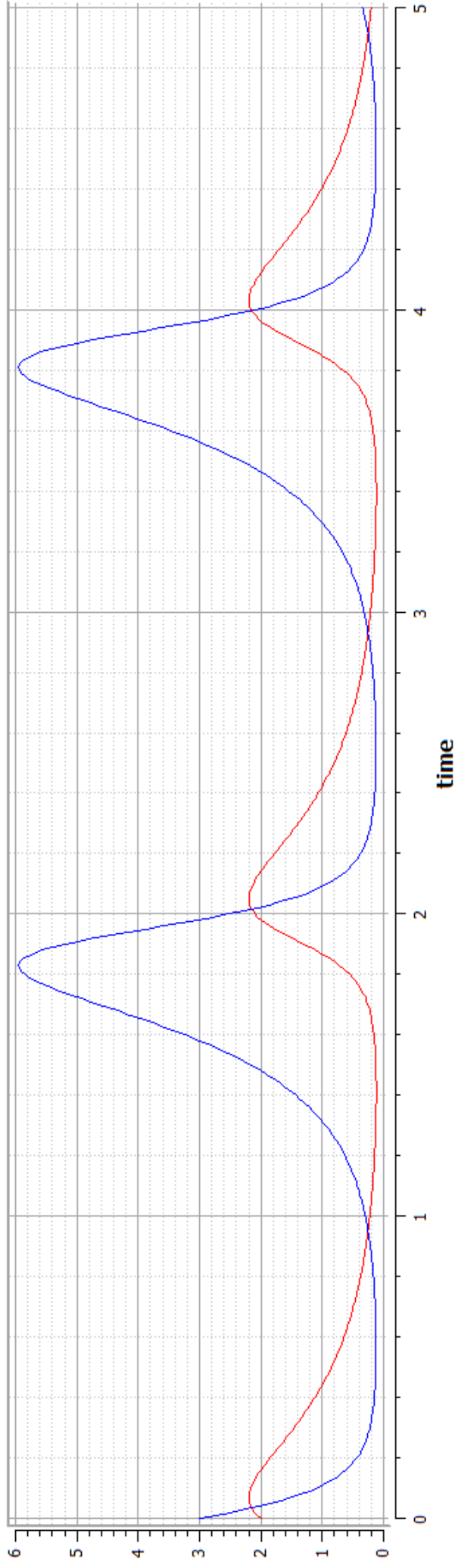
```
model mySimpleEqnSet "simple equation set"
  Real x(start=2, fixed=true);
  Real y(start=3, fixed=true);
equation
  der(x) = 2*x*y-3*x;
  der(y) = 5*y-7*x*y;
end mySimpleEqnSet;
```

plot({x,y})

[done]

Zoom Pan Fit in View Save Print Grid Log X Log Y

x y



plotParametric(x,y)

[done]

Zoom Pan Fit in View Save Print Grid Log X Log Y

y(x)

