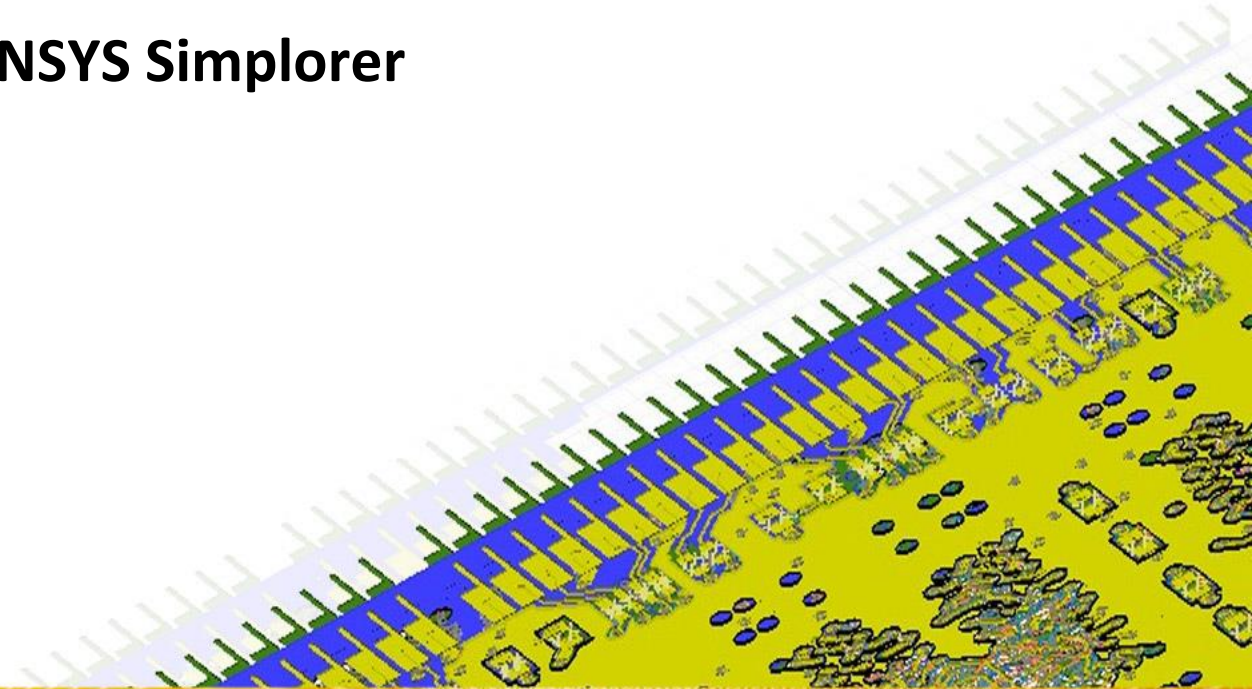




Module 02: Specific Usage

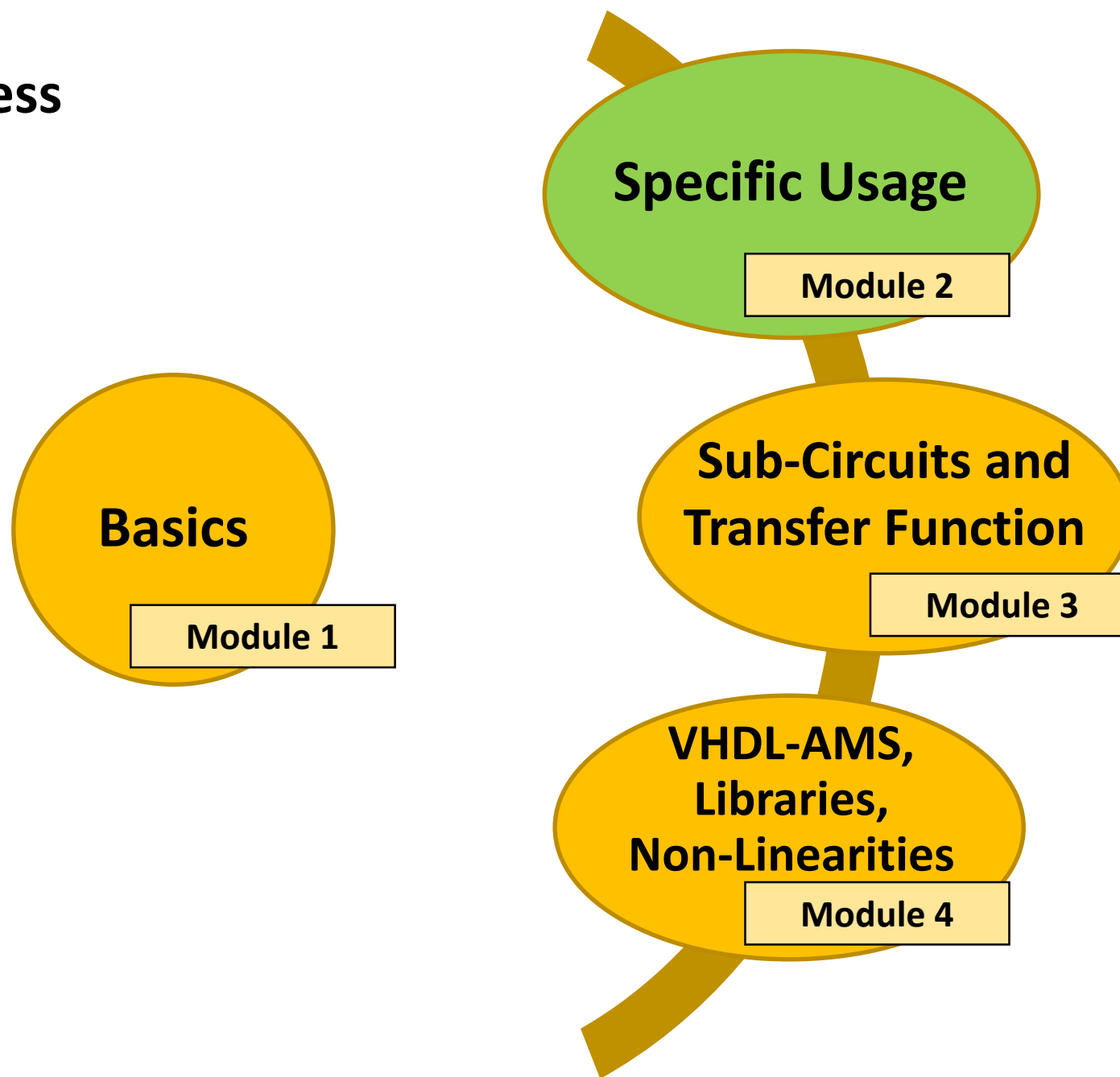
Introduction to ANSYS Simplorer



Overview

- **Syntax and data exchange**
- **Dedicated Components**
 - **Blocks**
 - **Source blocks + DATAPAIRS**
 - **FML_INIT, FML**
 - **GAIN, SUM, INTG**
 - **State Machines**
 - **STATE_11, TRANS**
- **Workshop 2.1: Control of a DCMP Motor**
- **Workshop 2.2: Single Phase Inverter with hysteresis (Bang-Bang) Control**

Overall Process



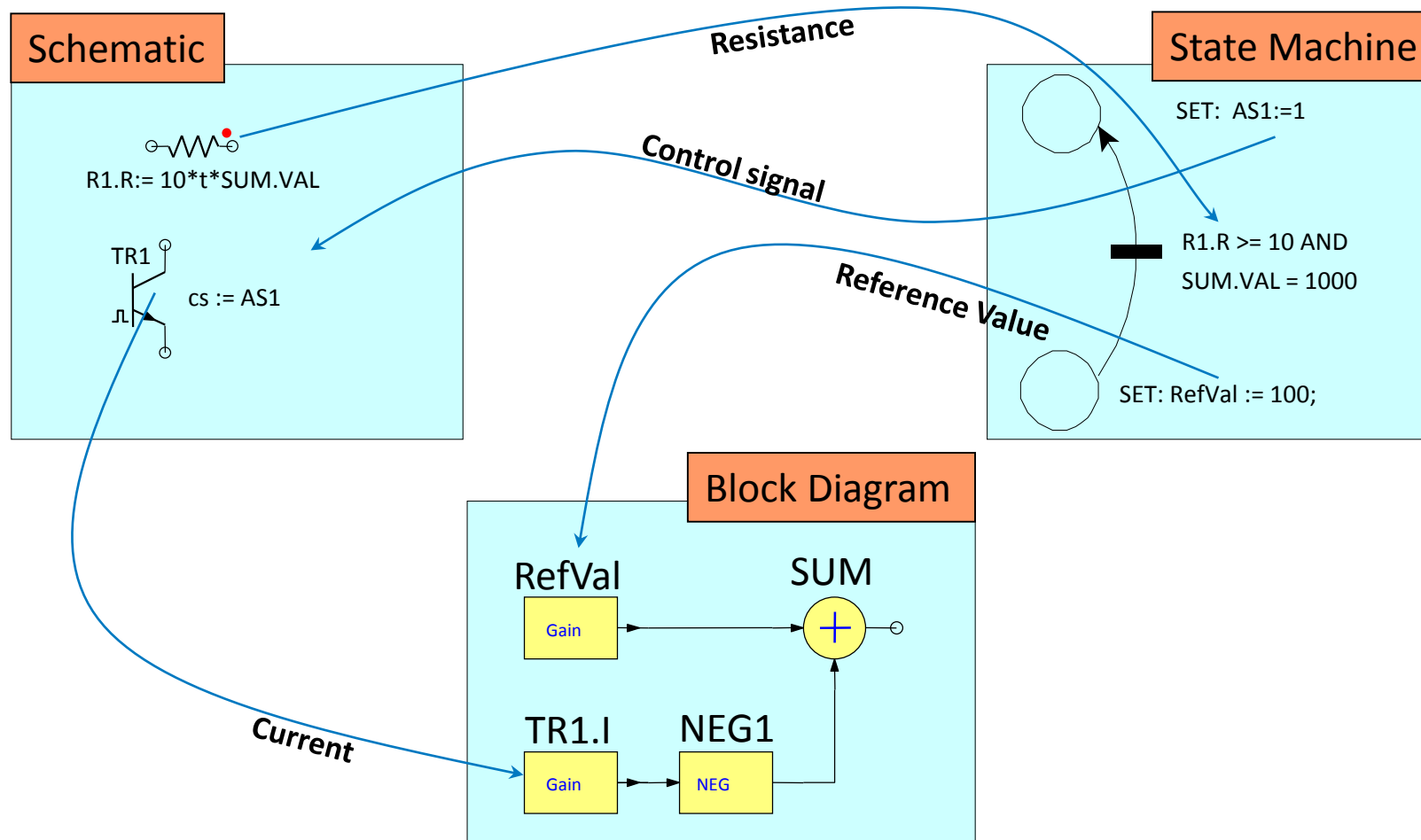
Components Syntax and Naming Conventions

- Syntax

- Default names of Simplorer components are R, L, C, E, etc.
- Once a component is instantiated an index is added after the name (e.g. R1, R2, R3, ...)
- By default the **Across** and **Through** quantities for conservative components and **Outputs** for not-conservative ones are accessible and calculated for further postprocessing and/or analysis
- The **Across** quantity is denoted with the letter “V” as extension of component name for electric components. For example, the expression **L1.V** denotes the *voltage drop across* inductance L1
- The **Through** quantity is denoted with the letter “I” as extension of component name for electric components. For example, the expression **L1.I** denotes the *current flowing through* inductance L1
- Components of different Nature have different letters to indicate **Through** and **Across** quantities
- The **Output** quantity is always denoted with “VAL” as extension of component name. For example, the expression **STEP1.VAL** denotes the *output signal* of source block STEP1
- All the above mentioned expressions together with numerous others (please see next three slides) can be used to define complex connections between components without drawing a wired connection in the schematic, allowing a great flexibility in modelling complex structures

Open Data Exchange

- In Simplorer it is possible to exchange data without the need of physical connections, as explained in the following figure (for Naming Conventions please see next slide)



Naming Conventions

– Examples for electrical components R, L, C



R_SHUNT.R

resistance value

R_SHUNT.I

current of the component

R_SHUNT.dI

1st derivative of the current

R_SHUNT.V

voltage of the component

R_SHUNT.dV

1st derivative of the voltage



C_LINK.C

Capacitance Value

C_LINK.VO

Initial voltage value

C_LINK.Q

Capacitance Charge

C_LINK.dQ

1st derivative of Capacitance Charge



L_LOAD.L

Inductance value

L_LOAD.IO

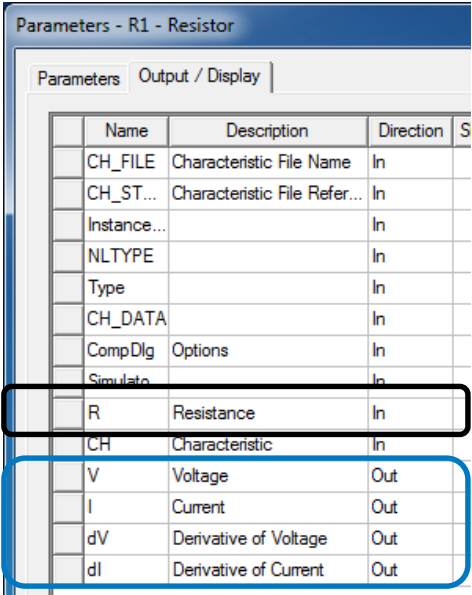
Initial current value

L_LOAD.PSI

inductor flux linkage

L_LOAD.dPSI

1st Derivative of flux linkage



The screenshot shows the 'Parameters - R1 - Resistor' dialog box. It has two tabs: 'Parameters' and 'Output / Display'. The 'Output / Display' tab is selected, showing a table of parameters. The table has columns: Name, Description, Direction, and S. The parameters are listed in the table, with some highlighted by a black box and others by a blue box.

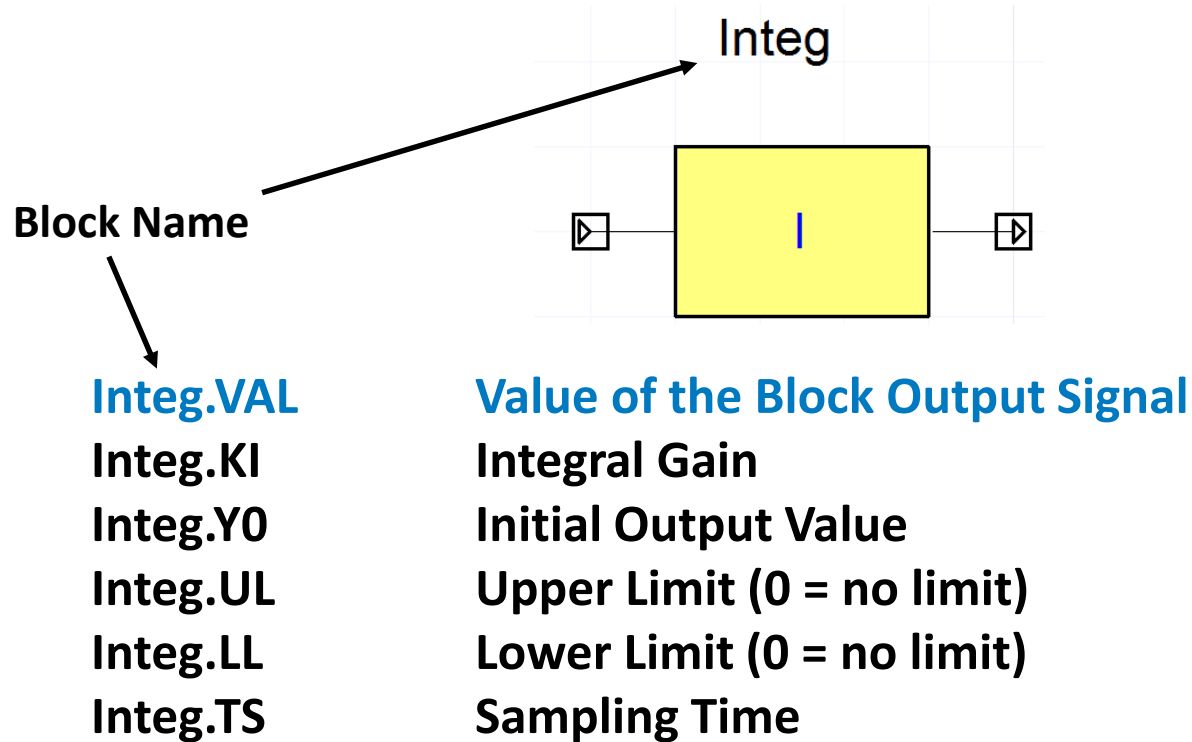
Name	Description	Direction	S
CH_FILE	Characteristic File Name	In	
CH_ST...	Characteristic File Refer...	In	
Instance...		In	
NLTYPE		In	
Type		In	
CH_DATA		In	
CompDlg	Options	In	
Simulate		In	
R	Resistance	In	
CH	Characteristic	In	
V	Voltage	Out	
I	Current	Out	
dV	Derivative of Voltage	Out	
dI	Derivative of Current	Out	

Inputs (Black)

Outputs (Blue)

Naming Conventions (Control Blocks)

– Example for Integrator Block



Parameters - Integ - Integrator

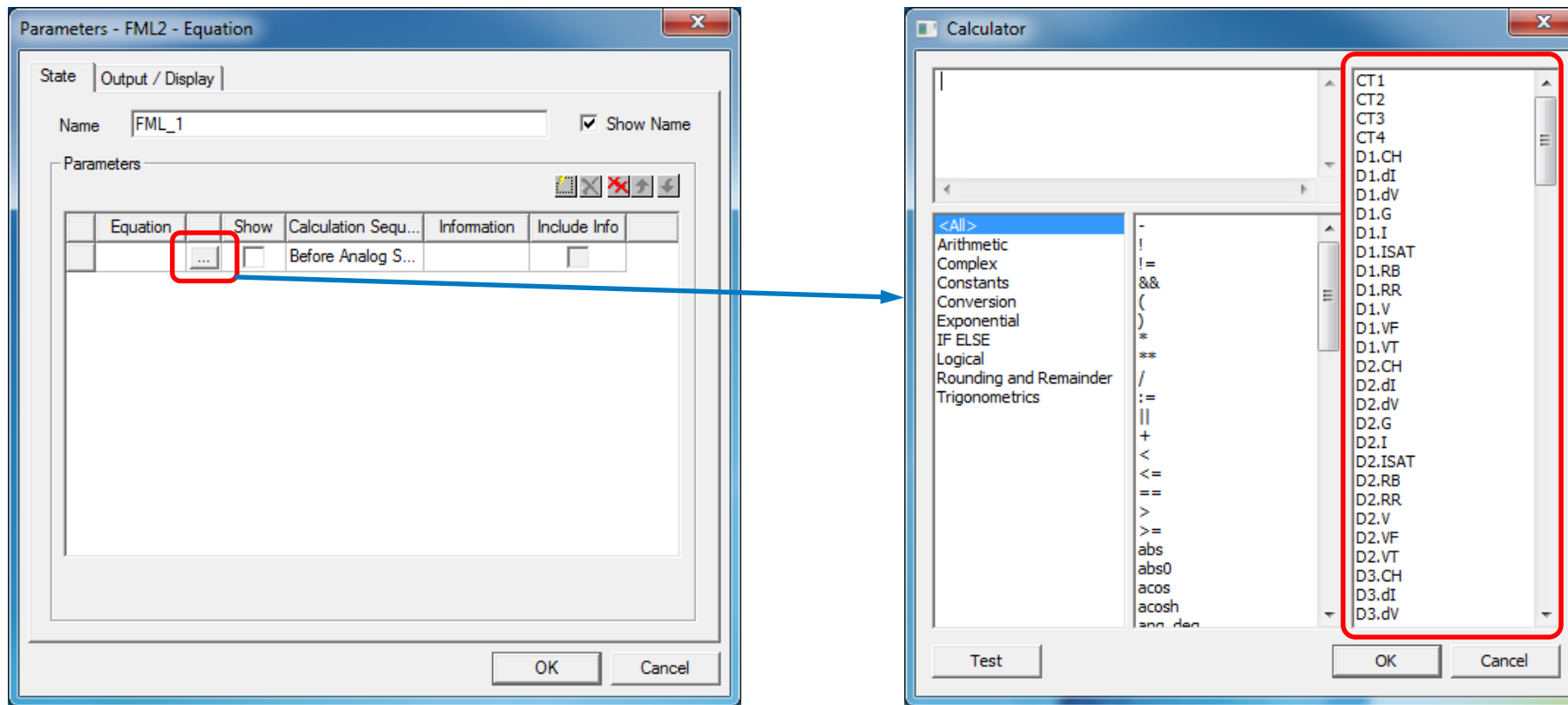
Parameters Output / Display

Name	Description	Direction
INPUT	Input Signal	In
KI	Integral Gain	In
Y0	Initial Value for t=0	In
UL	Upper Limit of Output Si...	In
LL	Lower Limit of Output Si...	In
TRIGSIG	Variable holding the Trigg...	In
TRIGVAL	Output Signal after trigg...	In
CTRL	Trigger setting	In
TS	Sample Time	In
VAL	Value	Out
INTEG_FORMULA	Integration Method	In

Inputs (Black)
Outputs (Blue)

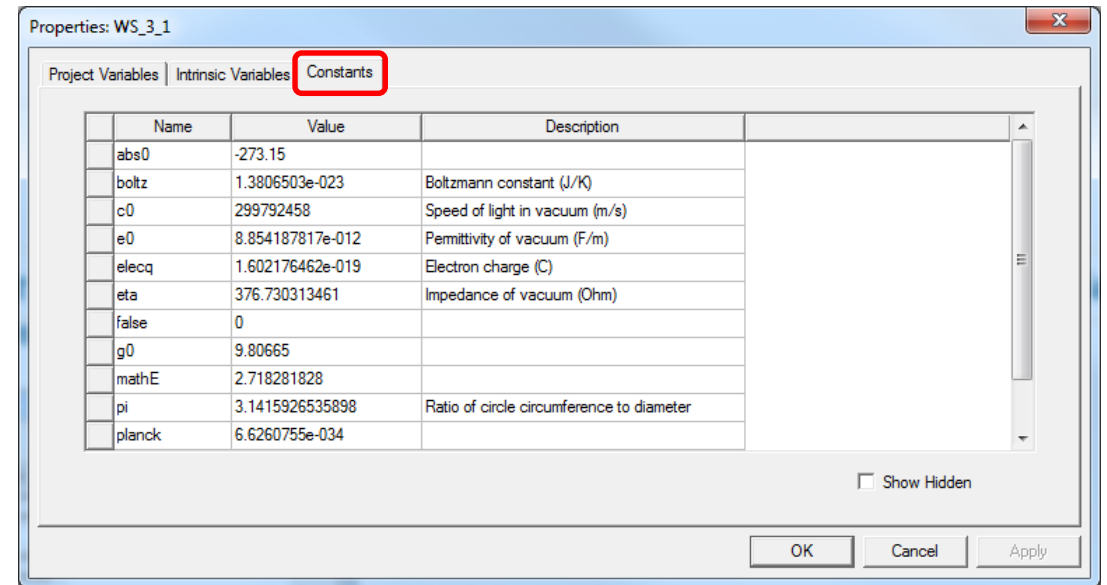
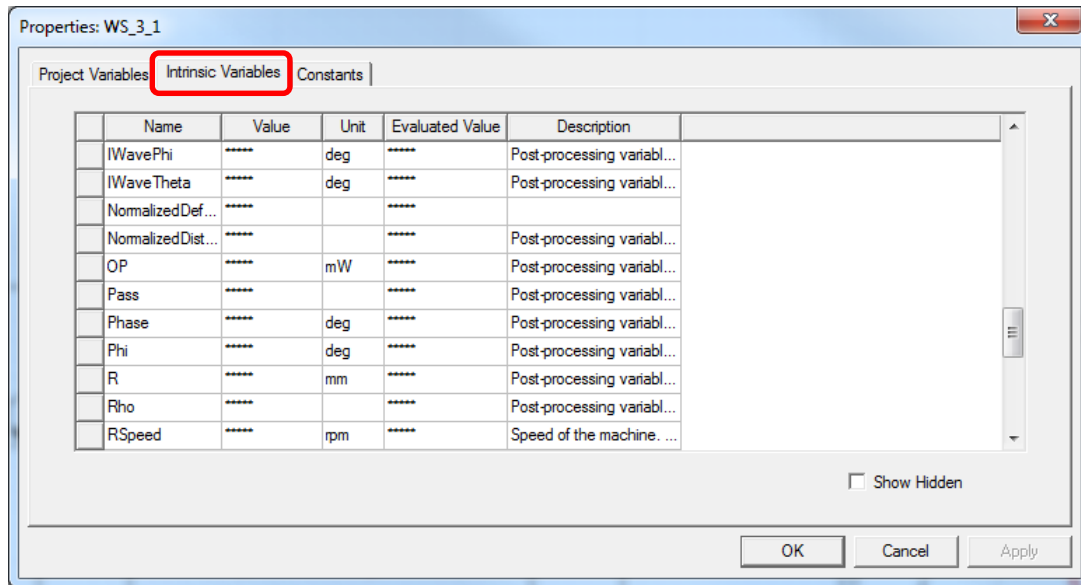
List of Naming via FML (EQU) Block

- If in the Schematic is inserted an **FML (EQU)** Block, it is possible to inspect a complete list of Naming present in the Schematic by clicking the button 



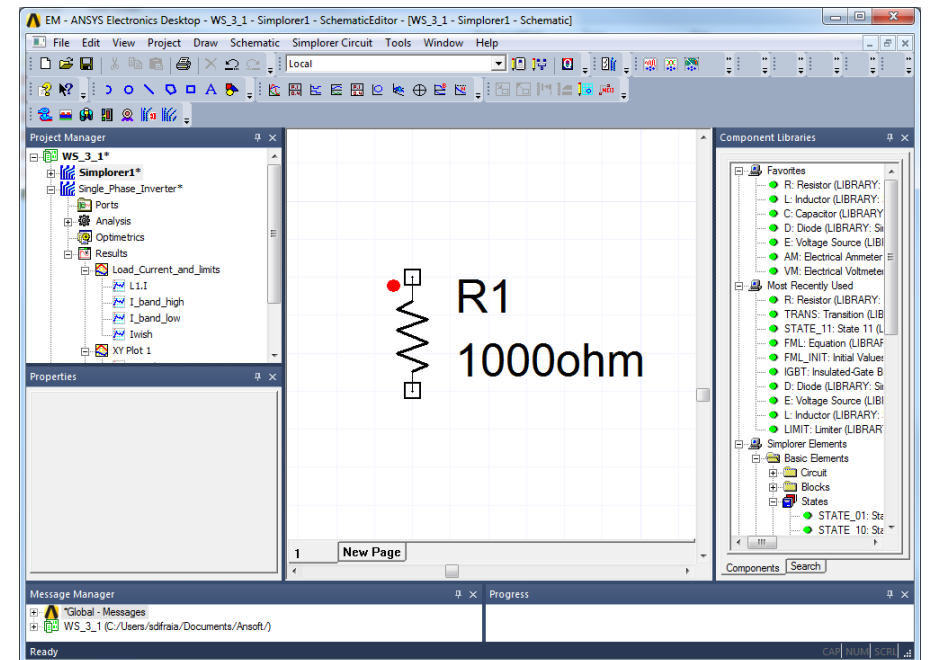
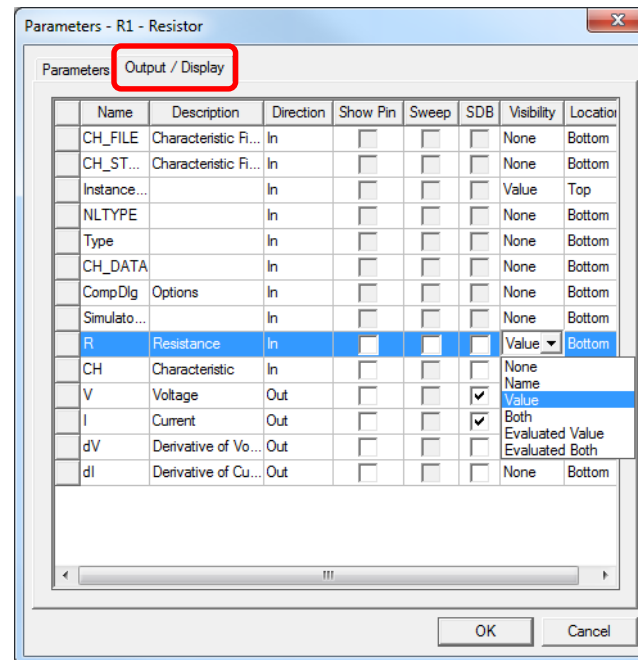
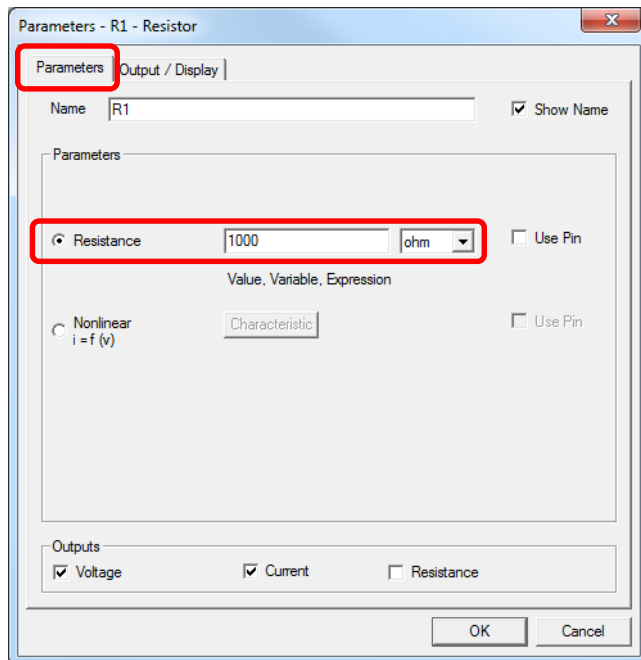
Reserved Variables

- From menu item **Project** → **Project Variables** it is possible to access both the **Intrinsic Variables** and the already defined default **Constants**



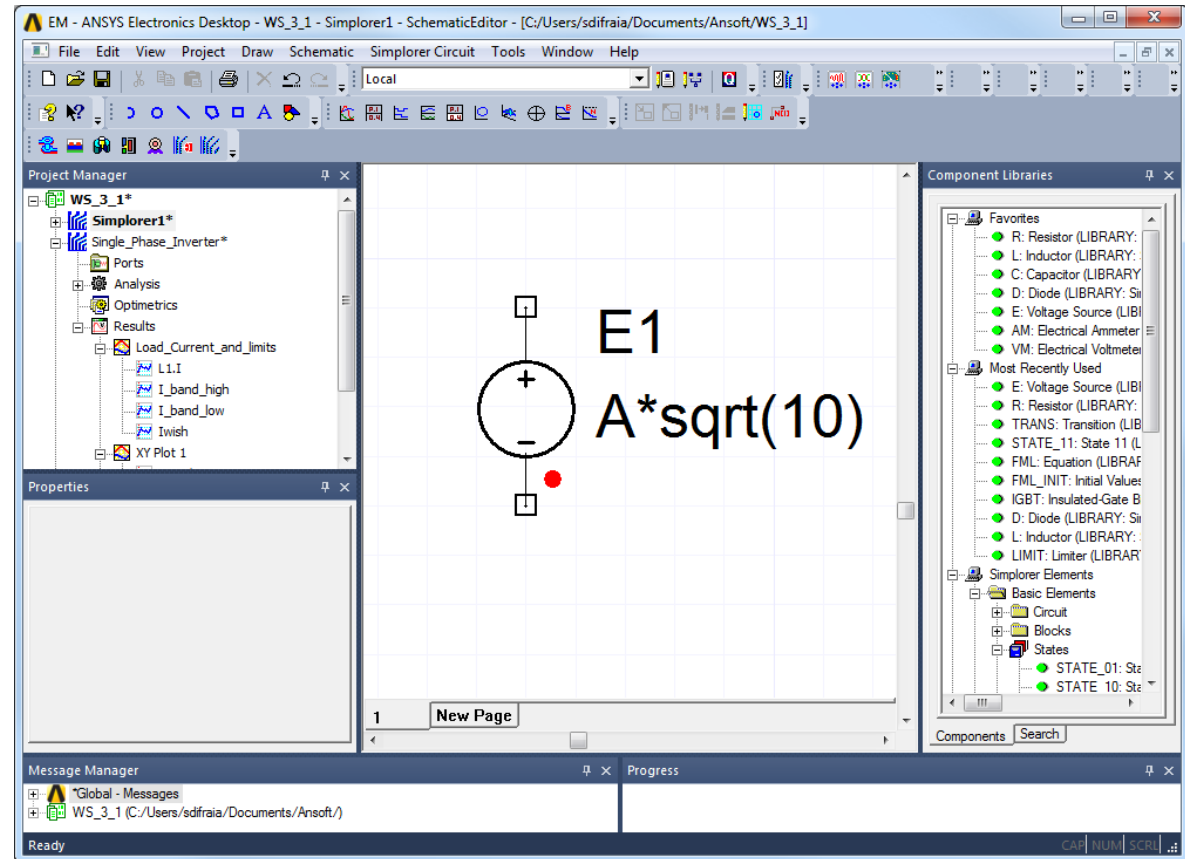
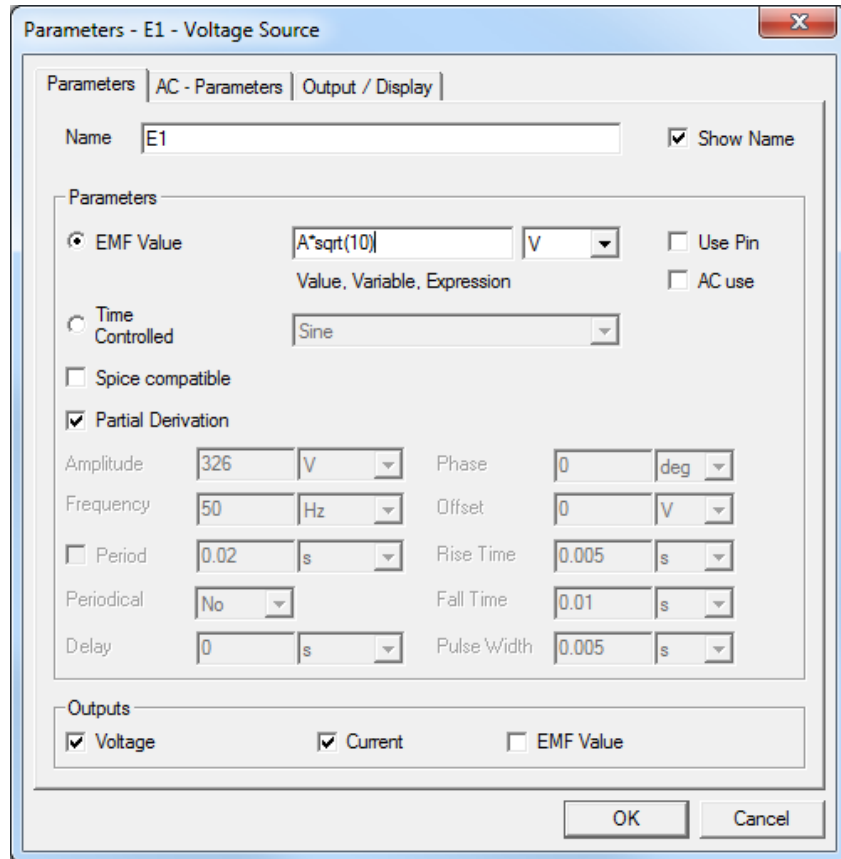
Component Parameters

- Once a component is placed in the Schematic window it is possible to edit its properties (Parameters) by double click on the component itself
- In the **Parameters Tab** it is possible to change the component properties, while in the **Output/ Display Tab** one can set properties visibility



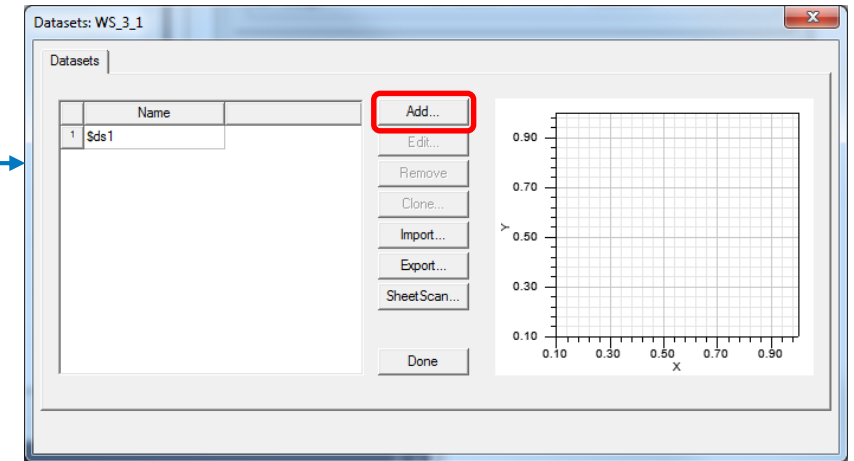
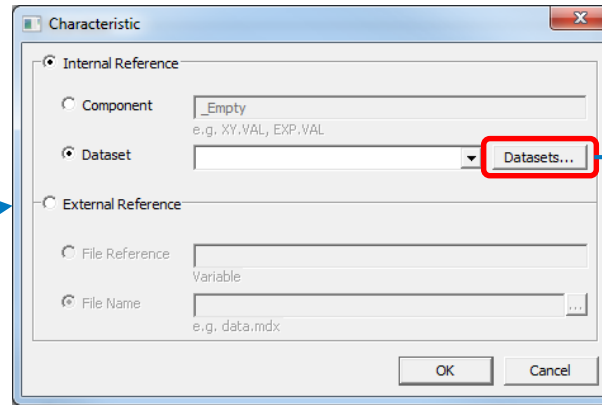
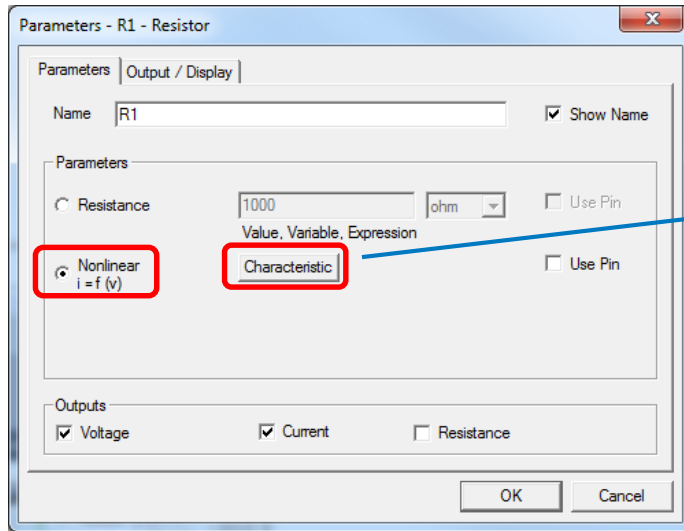
Component Parameters

- Note that component parameters can be equations containing variables and simulator parameters (e.g. time)

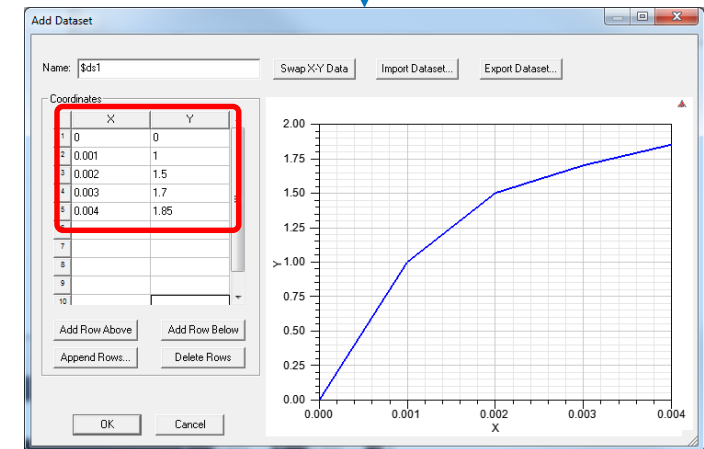
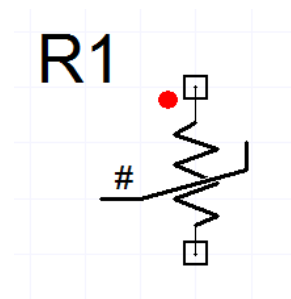


Component Parameters and Dataset

- This example shows a resistor non-linear characteristic where $i = f(v)$ using a purposely dedicated Dataset



Select **Nonlinear $i=f(v)$** and click the **Characteristic** button. Note the symbol will also change

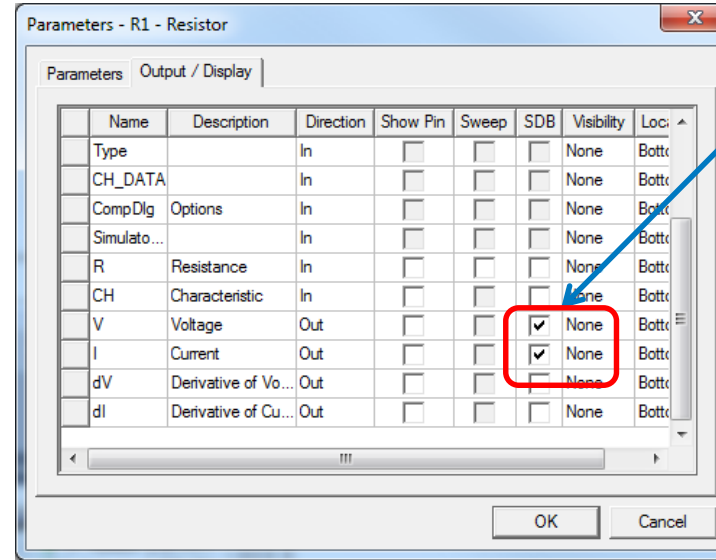
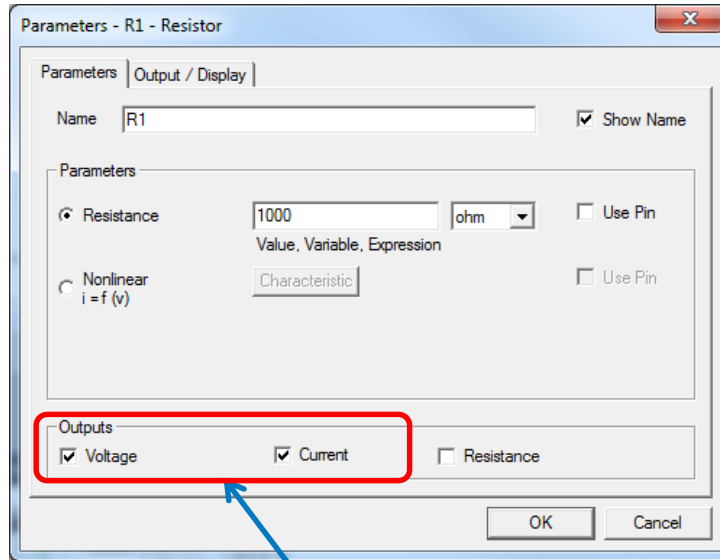


Component Parameters

Other Nonlinear Component's X-Y Relations

Non-linear component	x-y-Relation	X-Value	Y-Value
Resistance	$i = f(V)$	Voltage	Current
Capacitance	$v = f(q)$	Charge	Voltage
Inductance	$I = f(\varphi)$	Flux	Current
Dual Capacitance	$C = f(v)$	Voltage	Capacitance
Dual Inductance	$L = f(i)$	Current	Inductance

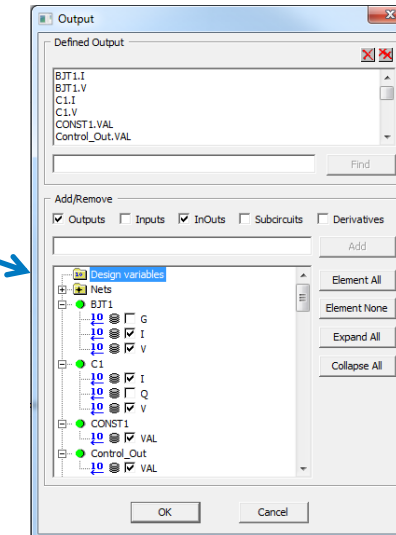
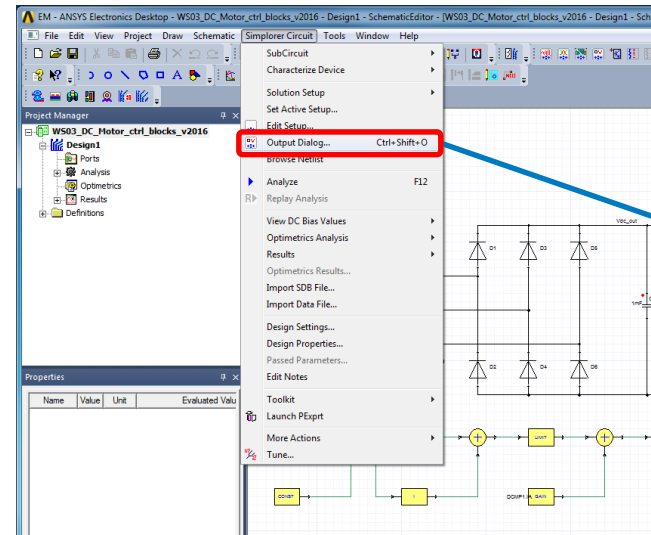
Output Definitions



Outputs can also be defined in the “Output/Display” tab of the Parameters window

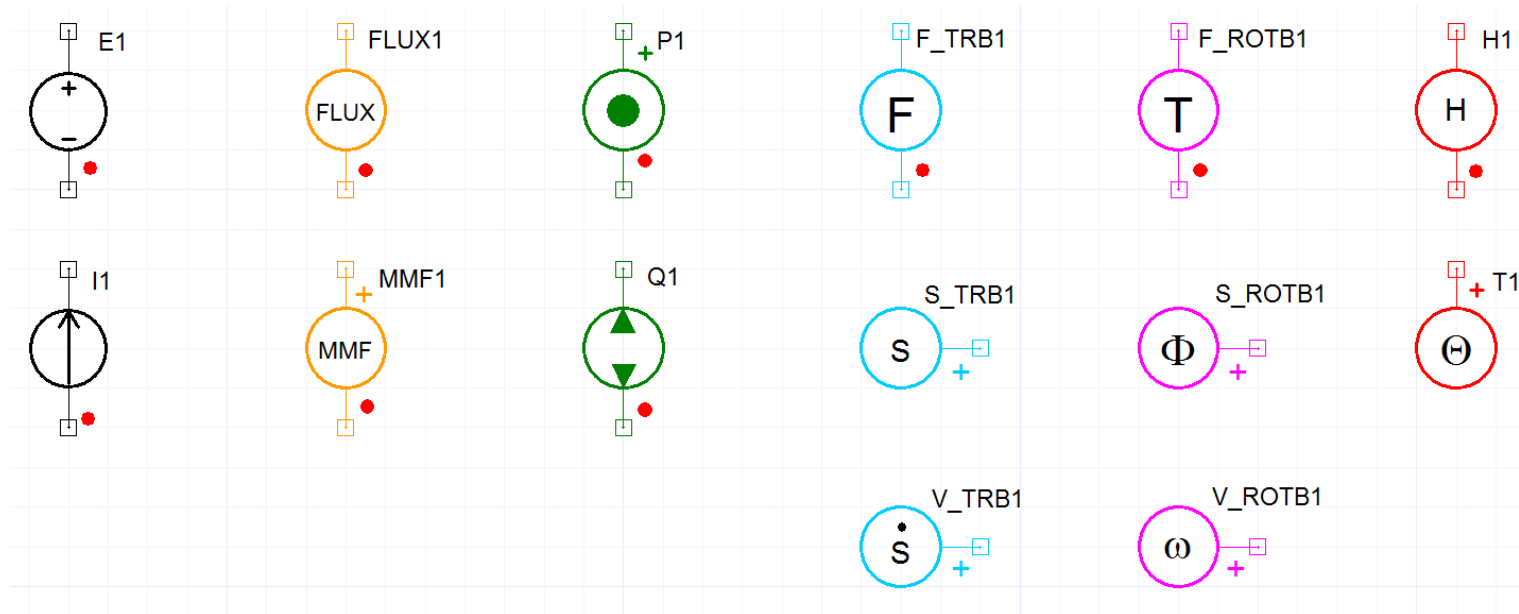
Outputs can also be defined globally via the menu item *Simplorer Circuit* → *Output Dialog*

Outputs are defined and saved in the data base file (SDB) so they can be viewed after simulation. By default, several outputs are defined and can be seen in the parameters window



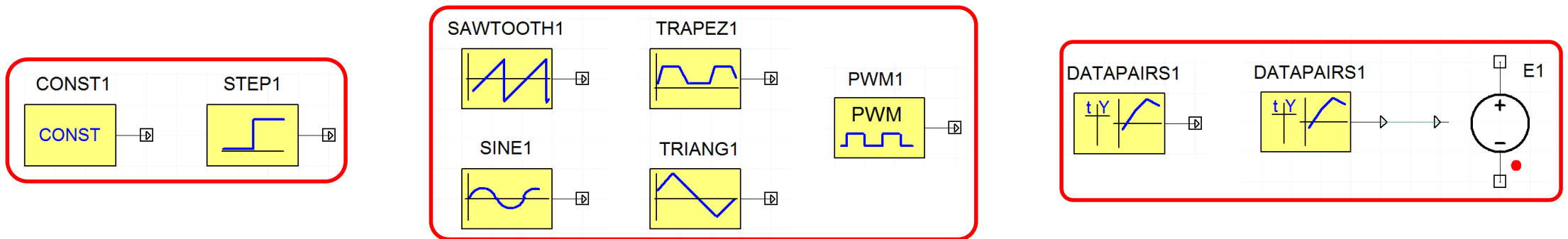
Sources Components

- There are several types of sources in Simplorer; the most used conservative components sources are:
 - electrical **voltage** and **current** sources – color code: black
 - magnetic **flux** and **magneto-motive force** sources – color code: orange
 - fluidic **pressure** and **flow** source – color code: green
 - mechanical **force**, **torque**, **position** and **velocity** sources – color code: light blue for translational, violet for rotational
 - thermal **heat flow** and **temperature** sources – color code: red



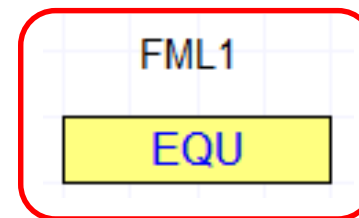
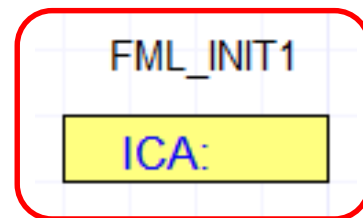
Sources Blocks

- The Blocks (non-conservative) sources provide a mathematical signal as output, with no physical meaning
- Several Blocks can be used as Sources; between them the most often used are:
 - the **CONST** and **STEP** blocks under *Basic Elements* → *Blocks* → *Sources Blocks*
 - The **SAWTOOTH**, **SINE**, **TRAPEZ**, **TRIANG** and **PWM** blocks under *Basic Elements* → *Tools* → *Time Functions*
 - The **DATAPAIRS** block: it transforms a predefined dataset in a time function, using the dataset first column values as times and the second column values as output signal. DATAPAIRS is also under *Basic Elements* → *Tools* → *Time Functions*
- All the blocks sources output signals can be used to drive conservative components sources



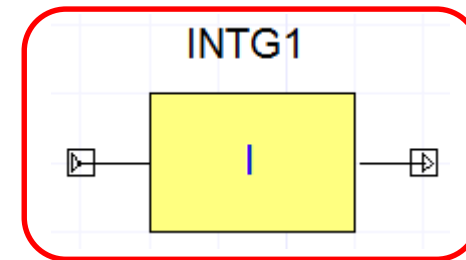
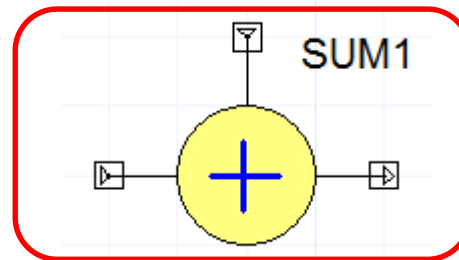
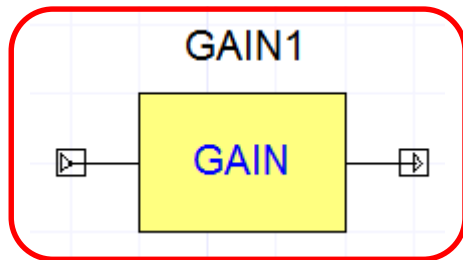
FML_INIT (ICA) and FML (EQU) Blocks

- The two blocks **FML_INIT (ICA)** and **FML (EQU)** are under *Basic Elements → Tools → Equations*
- The **FML_INIT (ICA)** Block is used to specify the initial value of any parameters used in the simulation. Variables defined in an Initial Value Assignment are computed once at the simulation start ($T = 0$). They can be used in each component on the schematic
- Parameters, parametric variables, and parameter ports are allowed on the right-hand side as they are evaluated once during simulation. Quantities and signals are not allowed
- The **FML (EQU)** Block provides mathematical expressions to modules. The formula interpreter manages mathematical and logical expressions, which consist of operands and operators. In addition, mathematical standard functions are supported. Other already defined variables are allowed as operands. The mathematical expressions are evaluated at every time step
- The name of a variable is defined by the user. It can be up to 50 characters long and must start with a letter. The spelling of the variable names is case sensitive

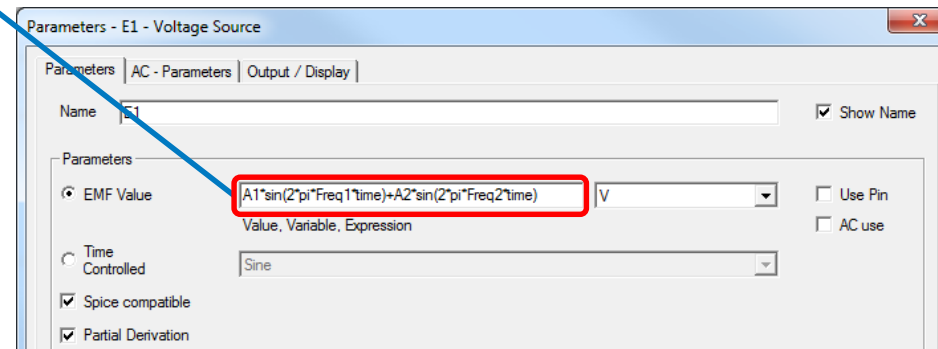
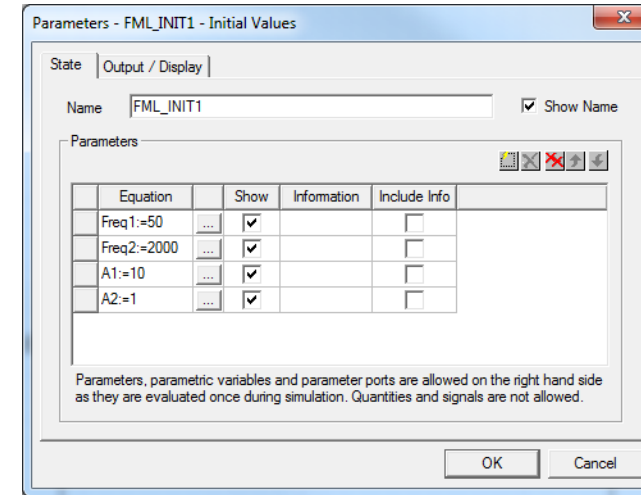
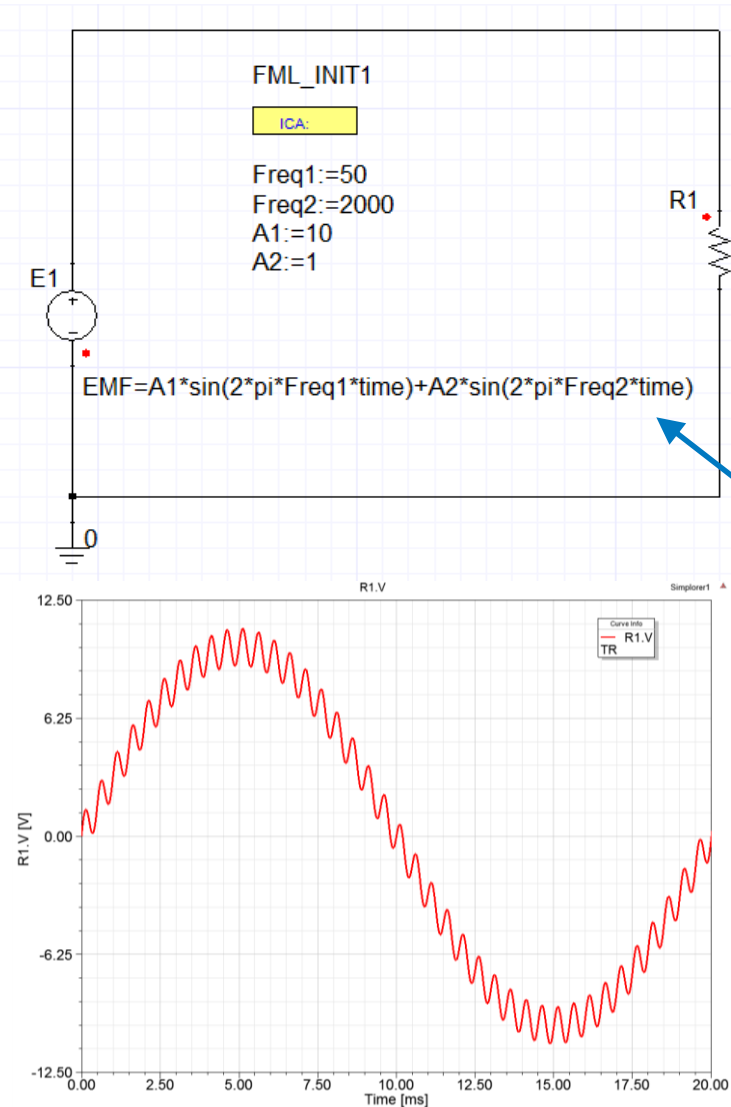


GAIN, SUM, INTG Blocks

- The **GAIN** Block is under *Basic Elements → Blocks → Continuous Blocks*
- It is used to multiply an input signal by the coefficient **KP (Proportional Gain)** and provide the multiplication result as output
- The **SUM** Block is under *Basic Elements → Blocks → Signal Processing Blocks*
- It is used to sum (with sign) two or more signal and provide the operation result as output
- The **INTG** Block is under *Basic Elements → Blocks → Continuous Blocks*
- It is used to perform a scaling and integration of the input signal and provides the operation result as output. The scaling coefficient **KI (Integral Gain)** needs to be defined
- It can be used both for TR and AC simulations



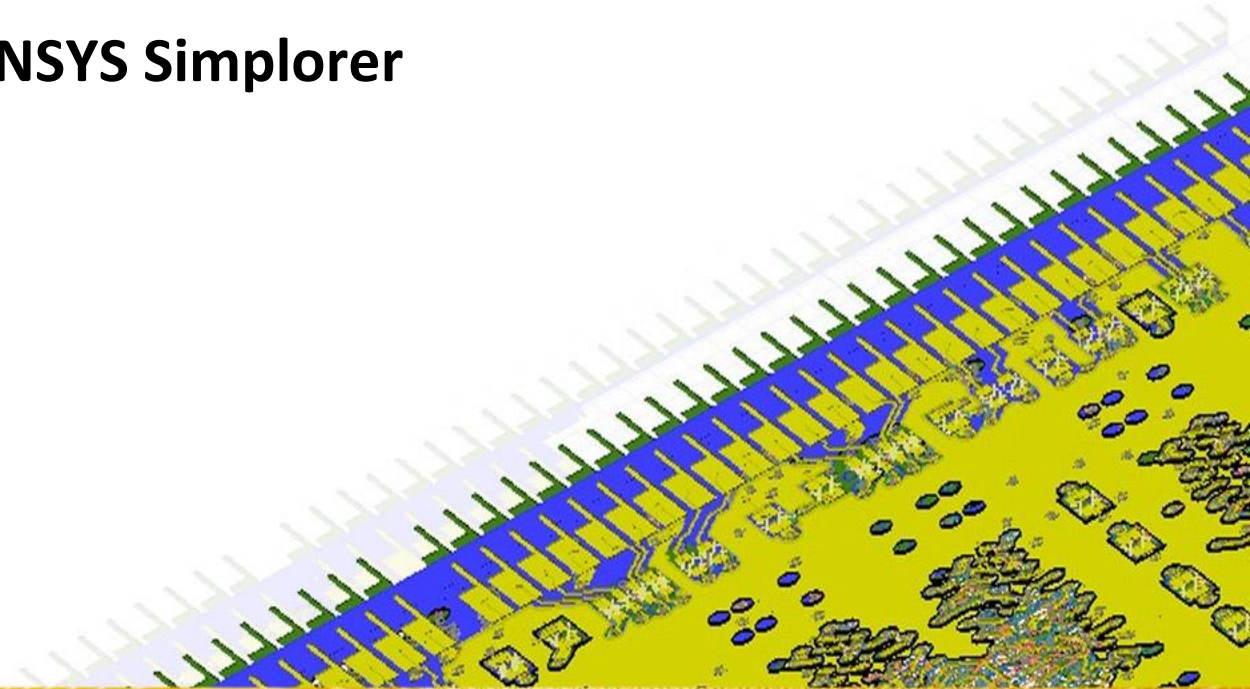
Sources Components Using FML_INIT Block





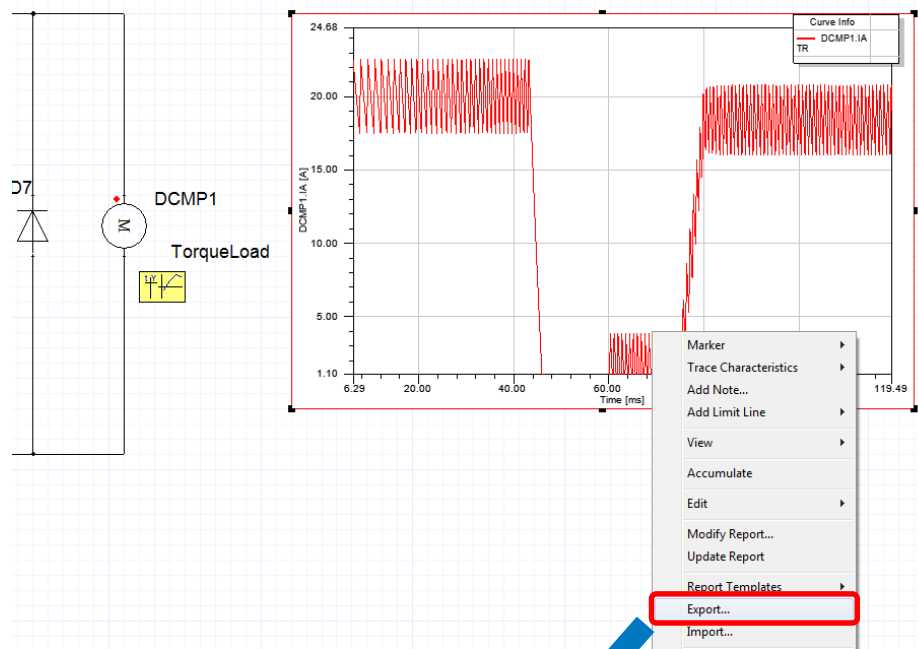
Data Import/Export

Introduction to ANSYS Simplorer



Data Export

The user can export complex waveform data to a variety of formats



Export Report: WS2_1 - DC_Motor_control - Results

File: *.csv Browse...

☐ Export Uniform Points

Start: 0 ms Stop: 120 ms Step: 12 ms

OK Cancel

Export Report: WS2_1 - DC_Motor_control - Results

Save in: Examples

Name	Date modified	Type	Size
WS_2_2.aedtresults	14.04.2016 08:33	File folder	
WS_3_1.aedtresults	14.04.2016 08:33	File folder	
WS1_1.aedtresults	16.03.2016 16:31	File folder	
WS2_1.aedtresults	14.04.2016 15:52	File folder	

File name: *.csv

Save as type: Comma delimited data files (*.csv)

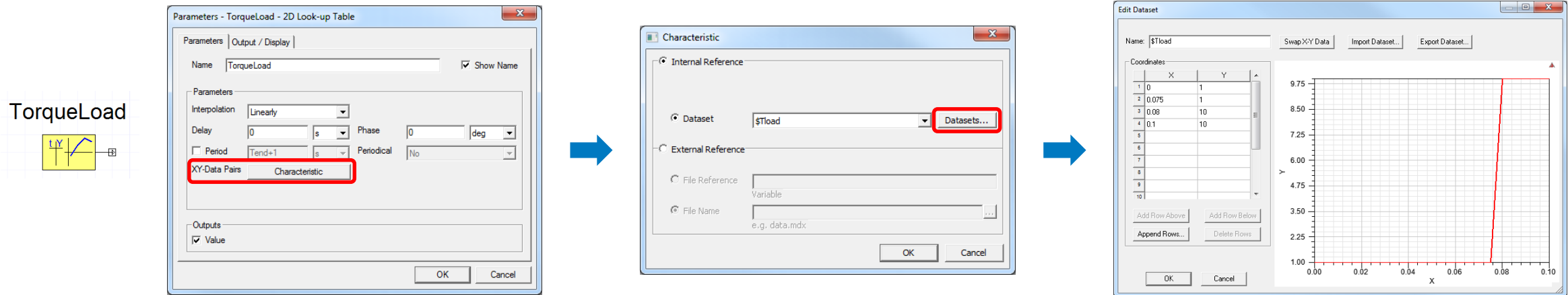
Save Cancel

Comma delimited data files (*.csv)
Tab delimited data files (*.tab)
Ansoft PlotData files (*.dat)
Post processor format files (*.txt)
Ansoft ReportData files (*.rdat)

	A	B	C	D	E
1	0	0			
2	0.001	2.48E-14			
3	0.002	7.86E-14			
4	0.003	1.66E-13			
5	0.004	2.95E-13			
6	0.005	4.77E-13			
7	0.006	7.31E-13			
8	0.007	1.08E-12			
9	0.008	1.58E-12			
10	0.009	2.27E-12			
11	0.01	3.27E-12			
12	0.011	4.72E-12			
13	0.012	6.88E-12			

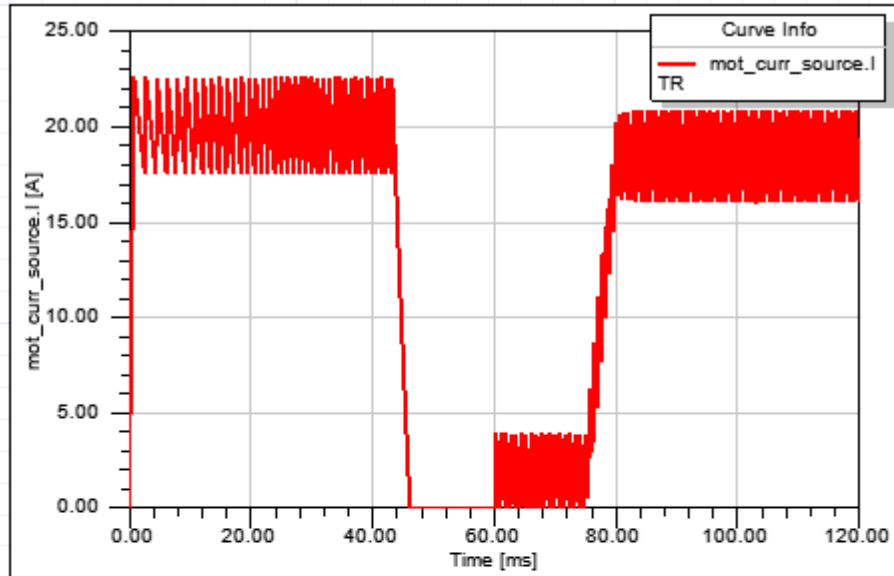
Data Import

- As already said, the DATAPAIRS block is used to import the data from a previously created file or dataset
- In the present example is shown a **DATAPAIRS** block named **TorqueLoad** and its linked Dataset named **\$Tload** (Project Variable), representing a torque step behavior vs time

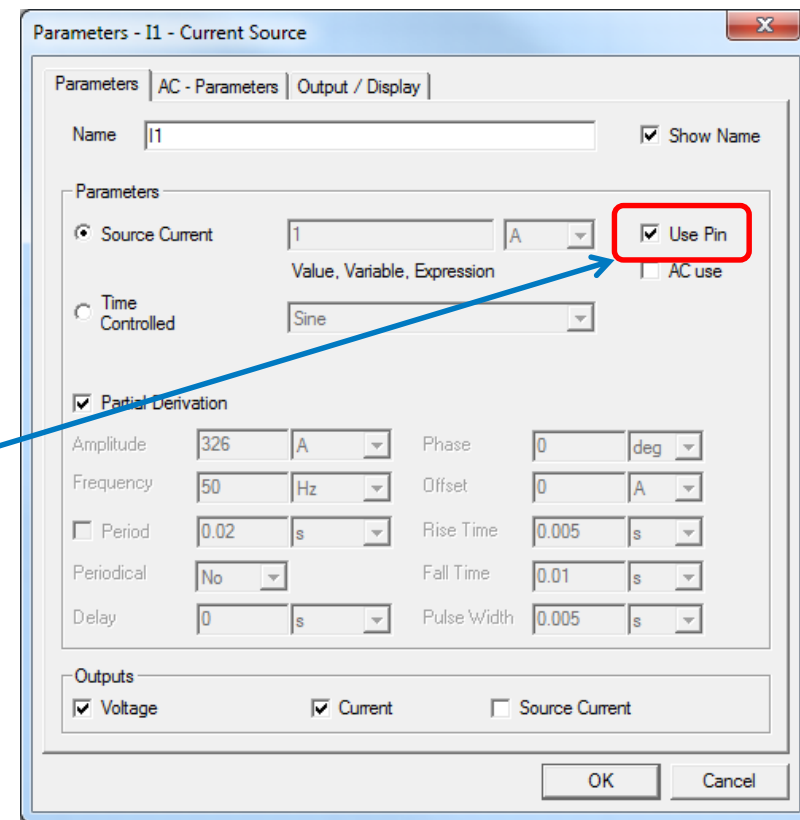
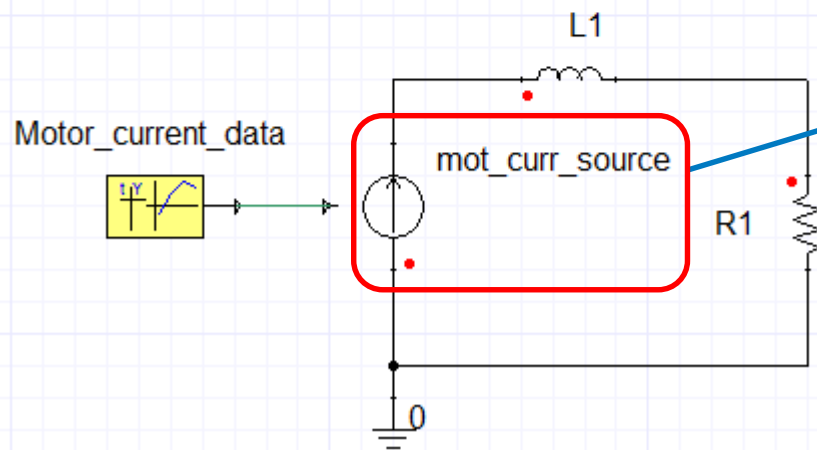


- Previously exported data or actual test bench data can be imported and used in other simulations as an input
- Note this can be a parameter to many different types of models, not just sources

Data Import



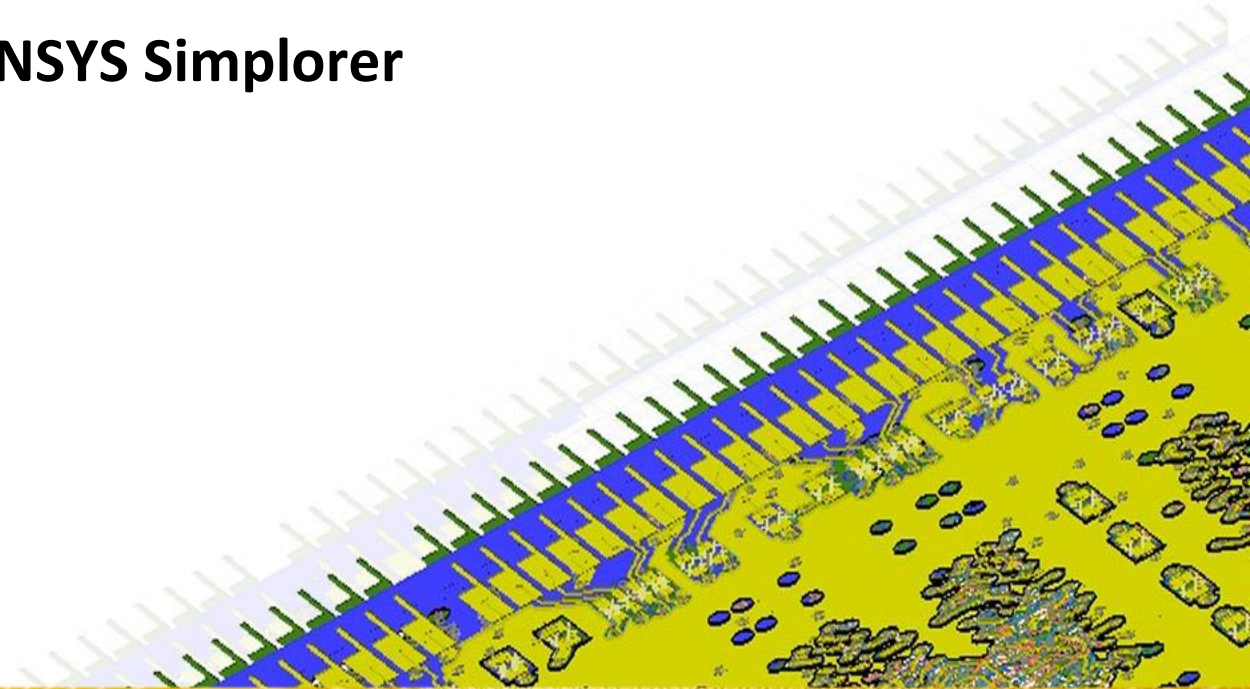
In this example, the input to the current source is set to take an external signal via a pin





State Machines

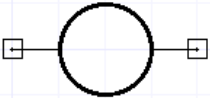
Introduction to ANSYS Simplorer



State Machines

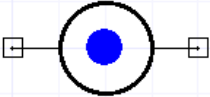
- State machines (or State Graphs) are a very powerful and flexible way to create a set of even complex commands following a desired logic
- State machines are often used for creating control logic depending on system “states” conditions
- State machines component are under *Simplorer Elements* → *Basic Elements* → *States*
- The most used (at least at basic level) State components are **STATE_11** and **TRANS**

STATE_11_1



Inactive state

STATE_11_1



Active state

Parameters - STATE_11_1 - State 11

State | Output / Display |

Name: STATE_11_1 ☒ Show Name

State Parameters

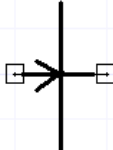
Valid for: ☒ TR ☐ DC ☒ Activate State

Actions

Action type	Name	Value	Information	Show
SET	Name	Value	...	Calculation once at the moment of activation

OK Cancel

TRANS1



Parameters - TRANS1 - Transition

Transition | Output / Display |

Name: TRANS1 ☒ Show Name

Parameters

Priority: 5 1 = highest

Condition for transition ☒ Show Condition

0 Edit...

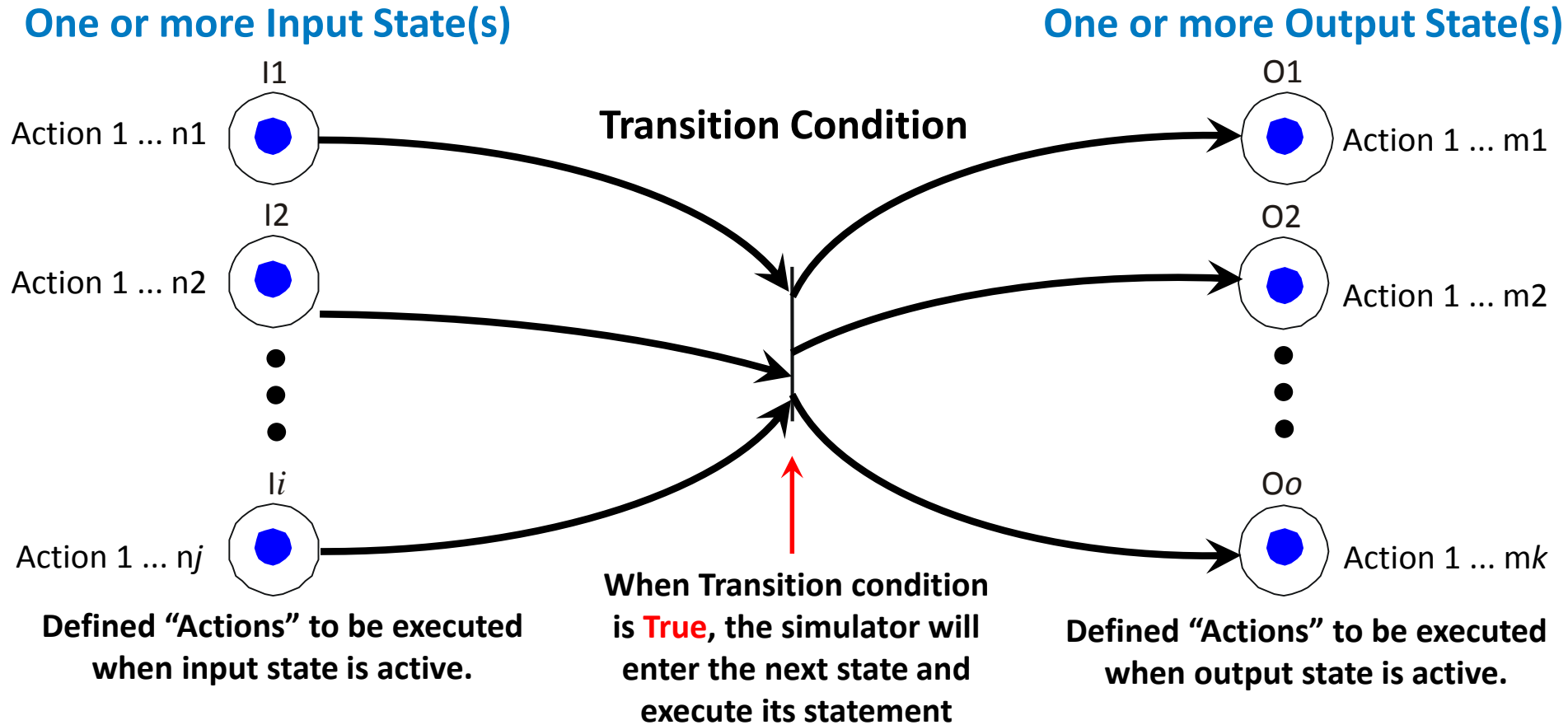
Logical Expression, Value

The use of equality in logical expressions (<= , == , >=) instead of simple relational operators (< , >), forces the simulator to slow down and approach the transition point with maximum accuracy.

OK Cancel

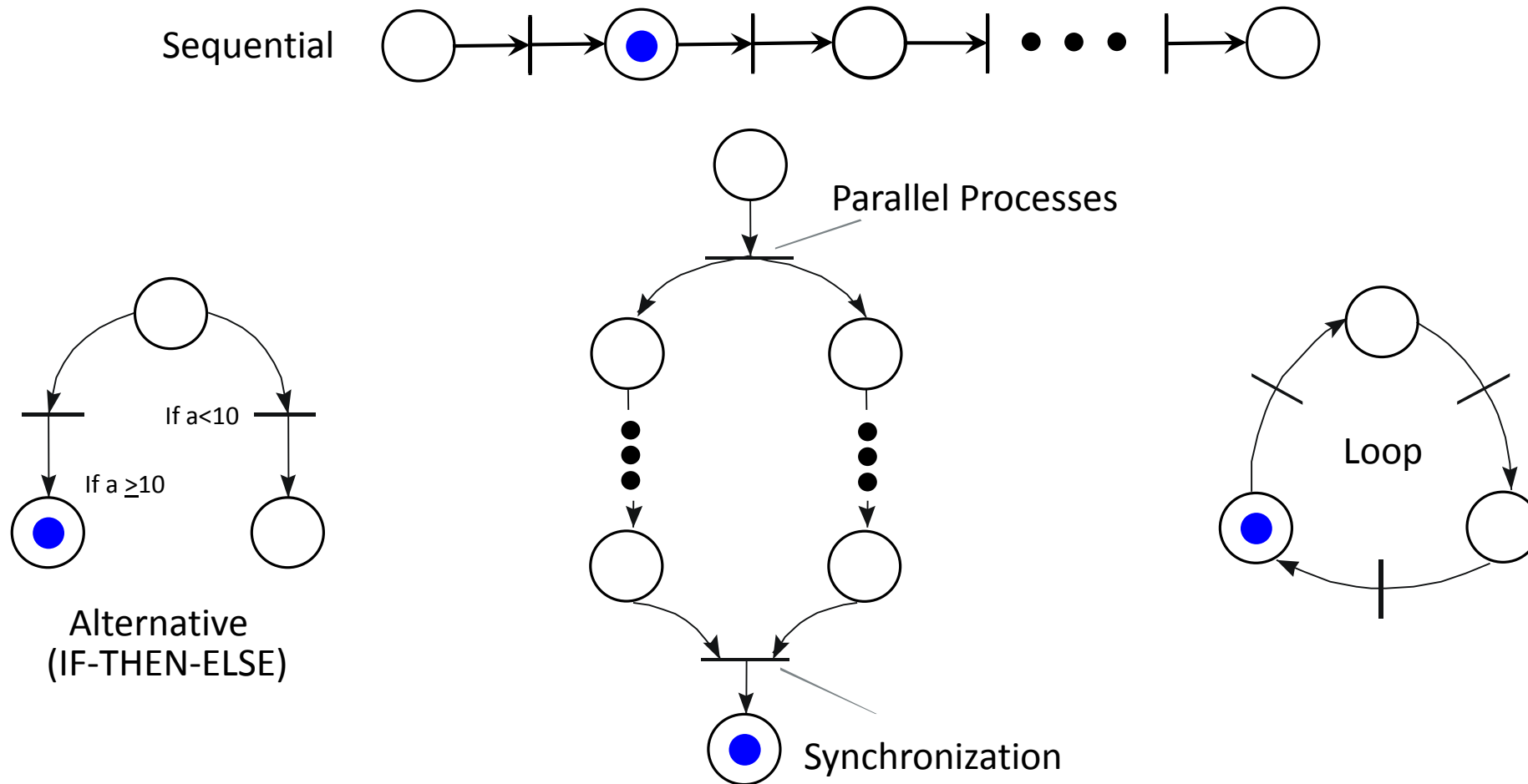
State Machines – Basic Concept

- The **State** components allow to define variables/signals values in case the related component is active
- The **Trans** component allow to define suitable conditions (using all available variables and quantities), permitting the transition between different states in case the defined condition(s) is/are **TRUE**



State Machines – Like Programming

- The **State Machines** are a sort of visual programming language; the classical **loops**, **if-then-else**, **parallel** and **sequential** processes can be easily realized as shown below

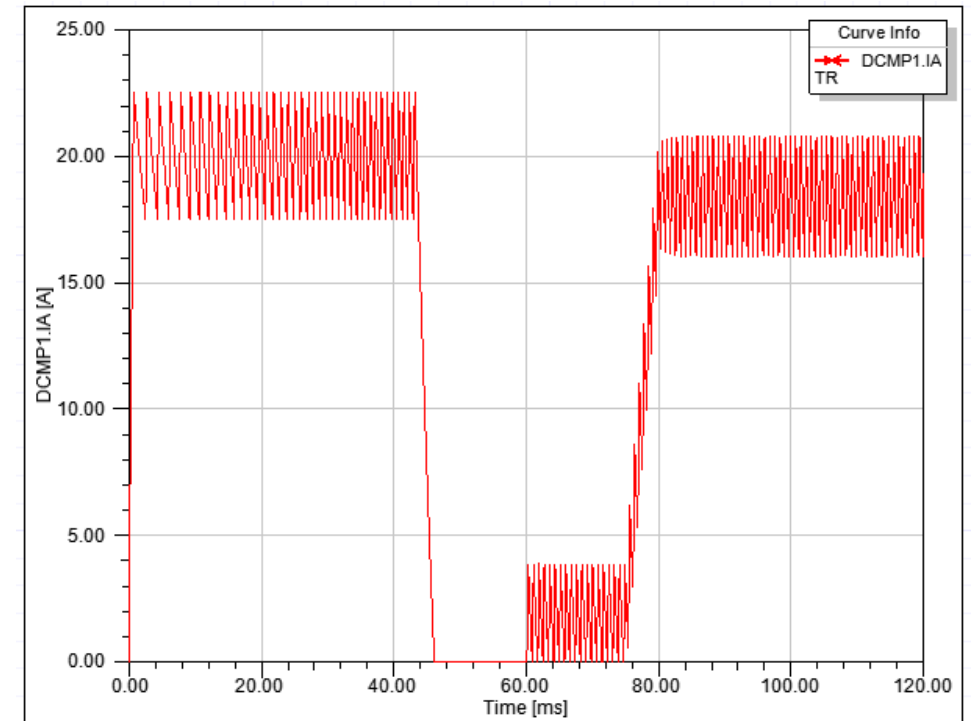
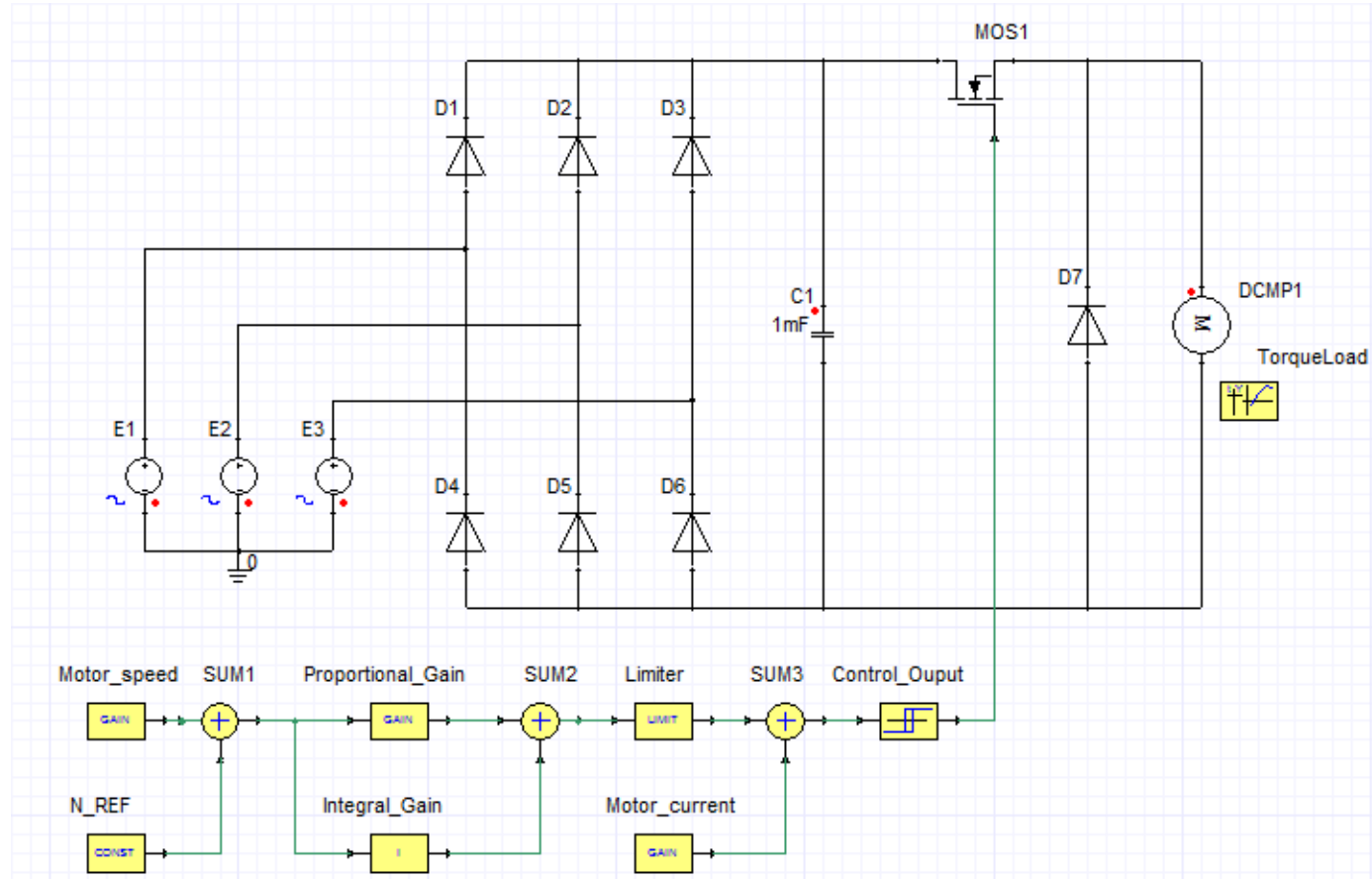


Summary

What have we learned in this session?

- **Syntax and data exchange**
- **Dedicated Components**
 - **Blocks**
 - **Source blocks + DATAPAIRS**
 - **FML_INIT, FML**
 - **GAIN, SUM, INTG**
 - **State Machines**
 - **STATE_11, TRANS**
- **Data Import/Export**

Workshop 2.1 – DC Motor Control



Workshop 2.2 – Single Phase Inverter – Hysteresis (Bang-Bang) Control

