MSS

Mechatronic System Simulation

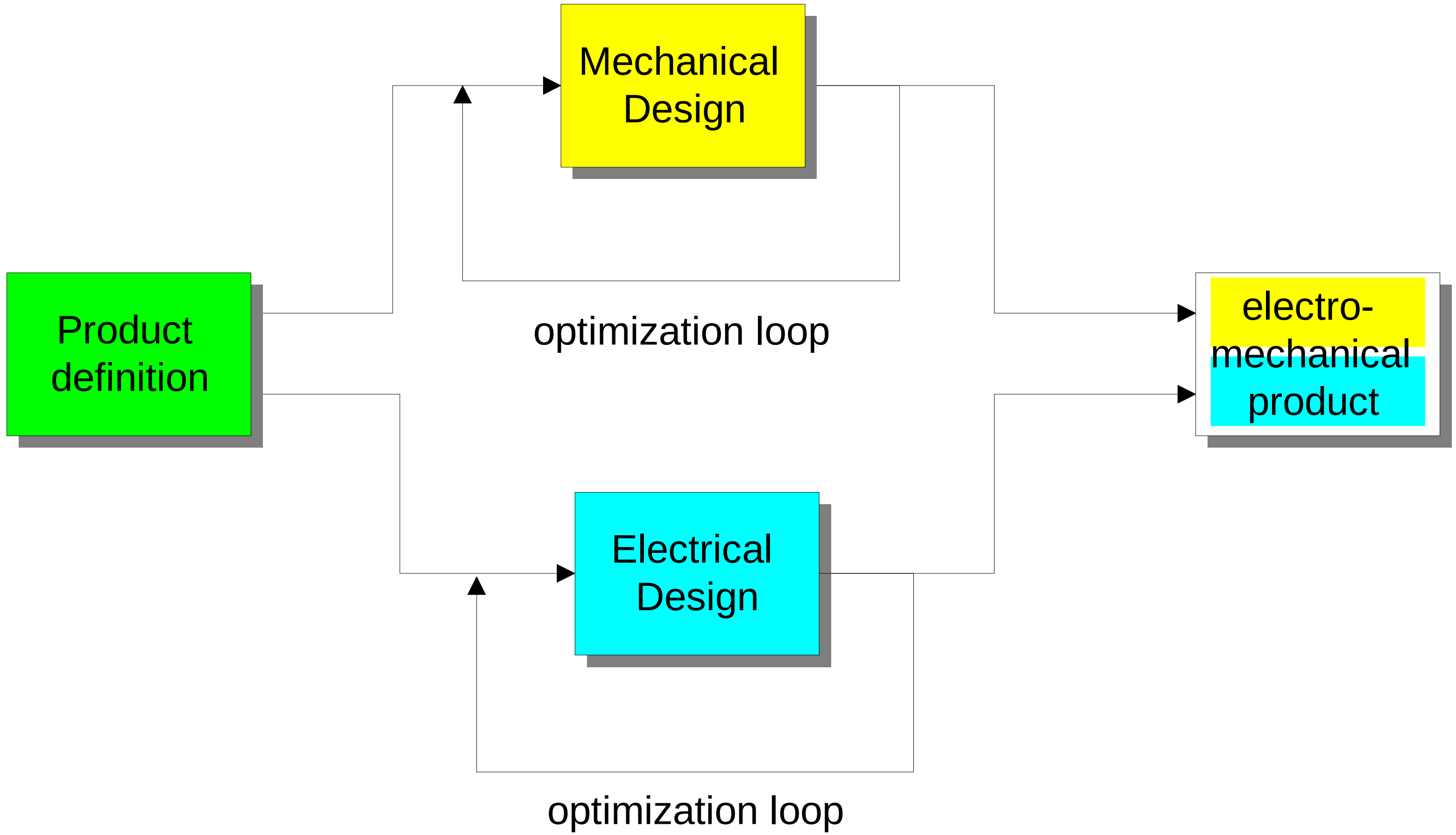
Today

- Introduction
- Content of the Module
- Examples
- Definitions

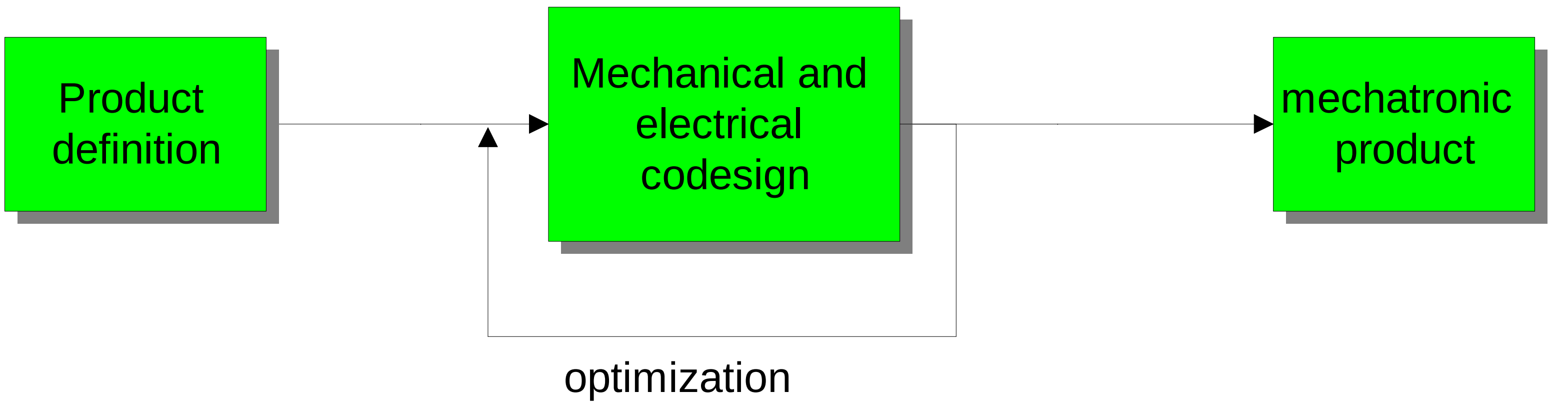
Background

- Simulation is one of the major skills for nowadays development process (important topic in conferences)
- Multi- Domain Simulation is very important for the development of Mechatronic Systems

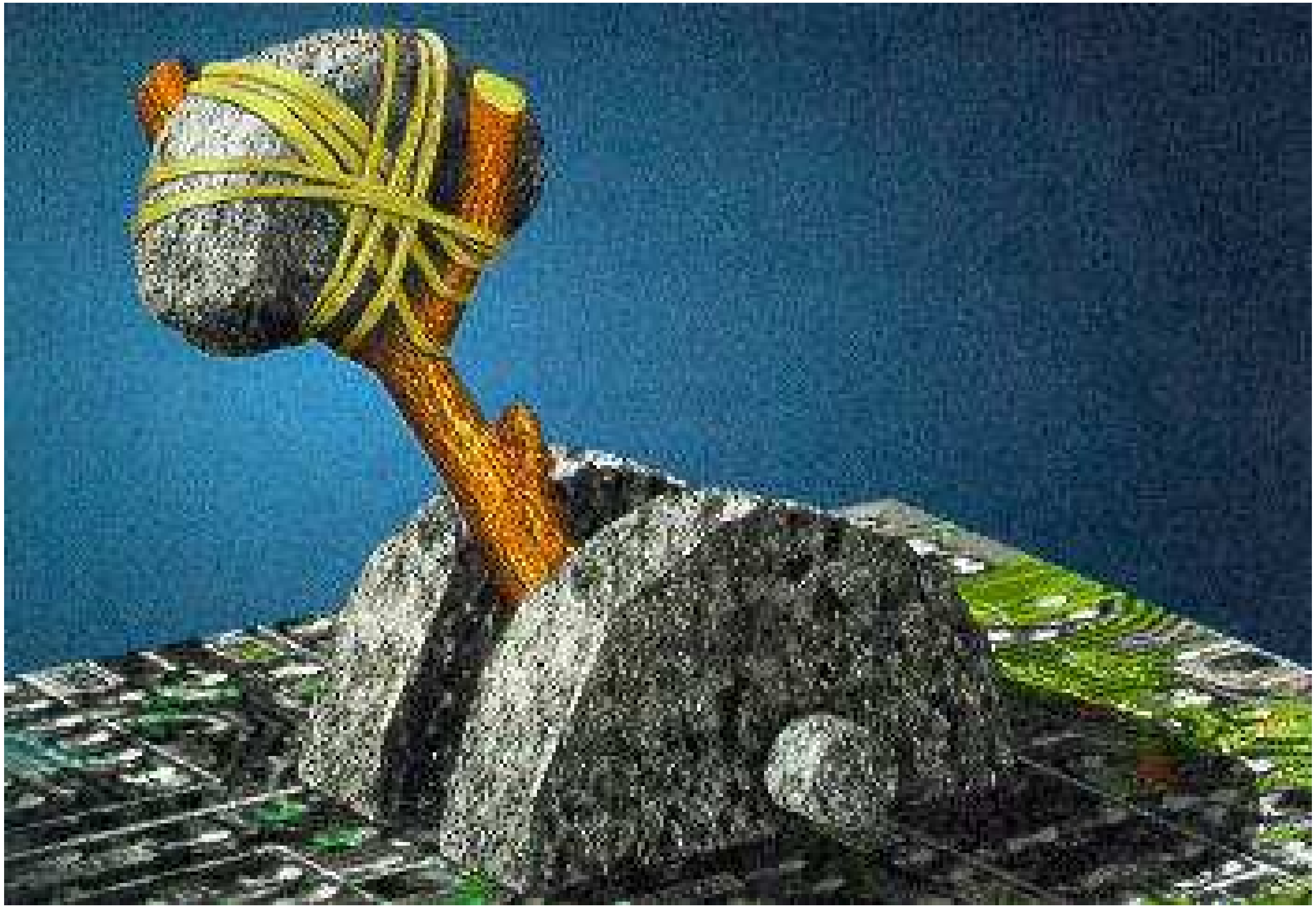
Conventional Developement Process



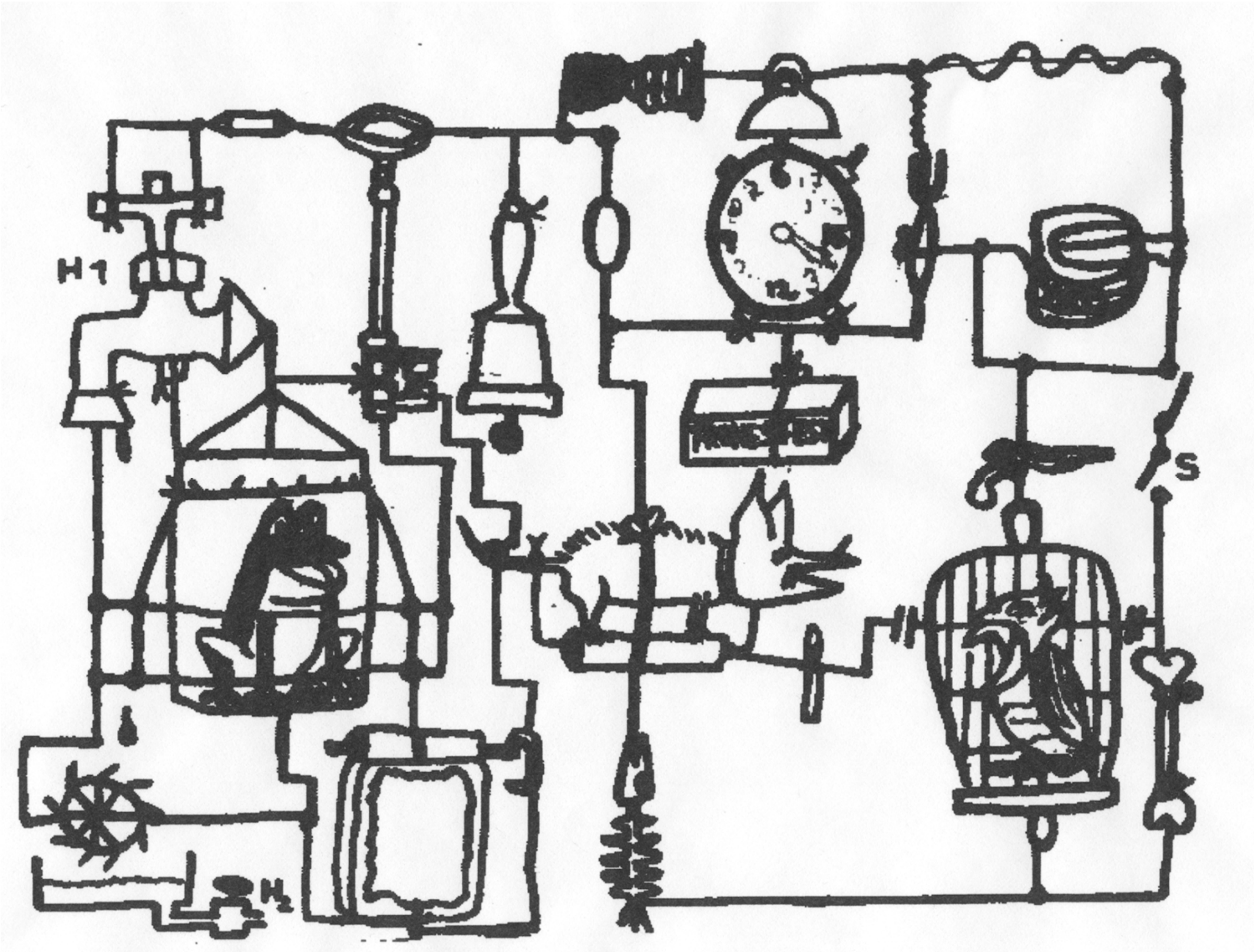
„Mechatronic“ Multi- Domain Development Process



Example for a “Mechatronic Product”



Mechatronic Circuitry



Objective

- Objective of the module MSS is to get familiar with the most important systems for simulation:
  - ***Lumped Element Simulation*** *(component based models)*
  - ***Integrated Systems Simulation*** *(behavior based models)*

Exam

- The main objective of the module is to get familiar with the simulation methods and tools
- At least a part of your practical exercises is element of the exam
- So take your chance to practice and to gather experience and points for your exam
- Participation of at least 80% of the classes is mandatory for admission to the exam

The Module

Mechatronic Systems Simulation

consists of

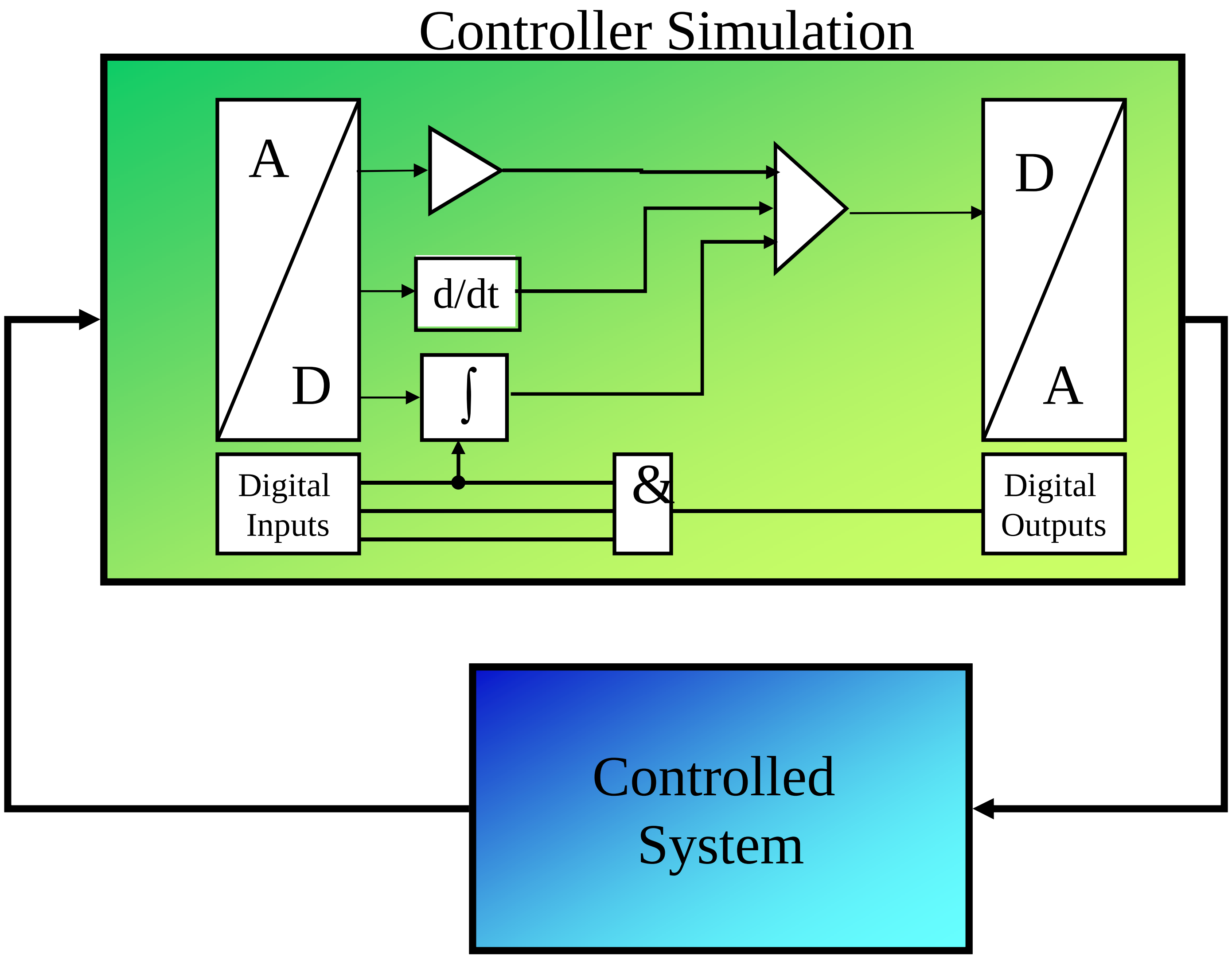
- Part A: Integrated Systems Simulation (Jochen Theis, M.Sc.)
  Tools: **Matlab/Simulink/Stateflow** *(Mathworks)*
  Thursday Group A: 10:00 – 12:00,
  Group B: 12:00 – 14:00
  Start: next week
- Part B: Lumped Element Simulation (Prof. G. Schmitz)
  Tool: **Saber** *(Synopsis)*
  Friday Group A: 10:00 – 12:00,
  Group B: 12:00 – 14:00,
  Start: Tomorrow

Part A: Integrated Systems Simulation

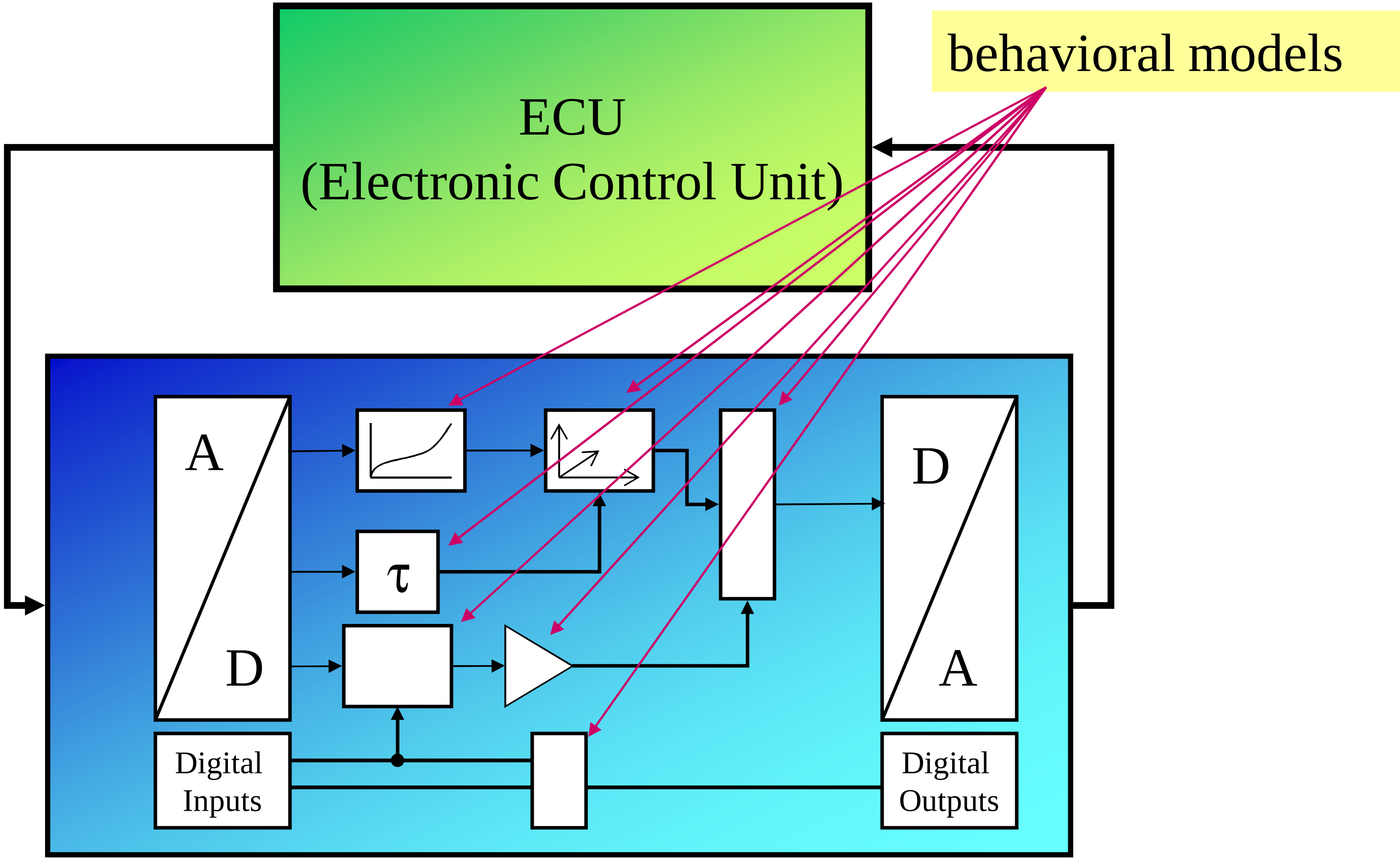
Examples

Matlab/Simulink/Stateflow

Example: Rapid Controller Prototyping

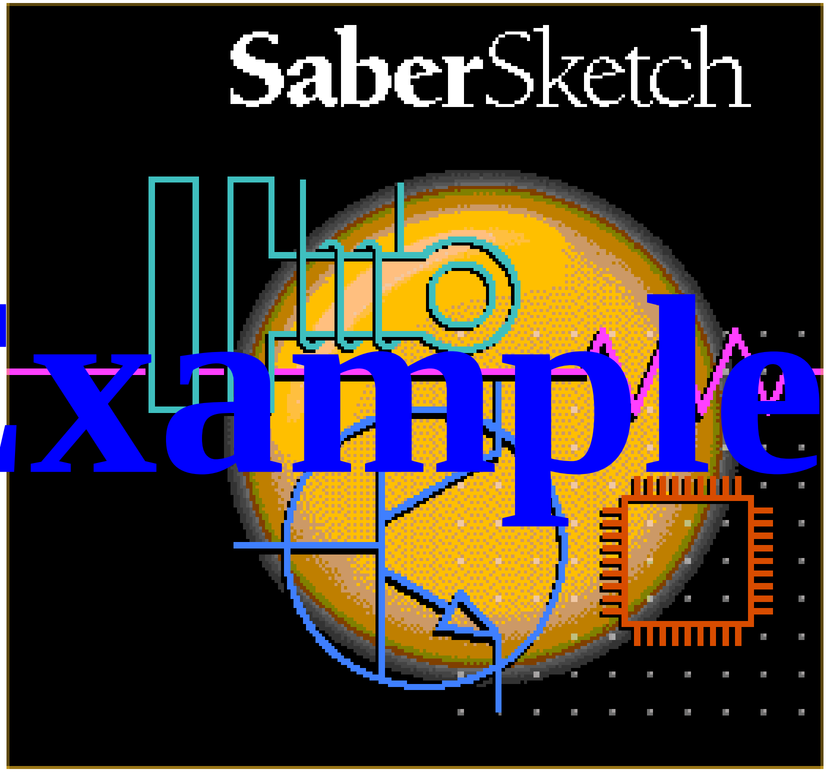


Example HIL (Hardware In The Loop)



Simulated Hardware (= controlled system)

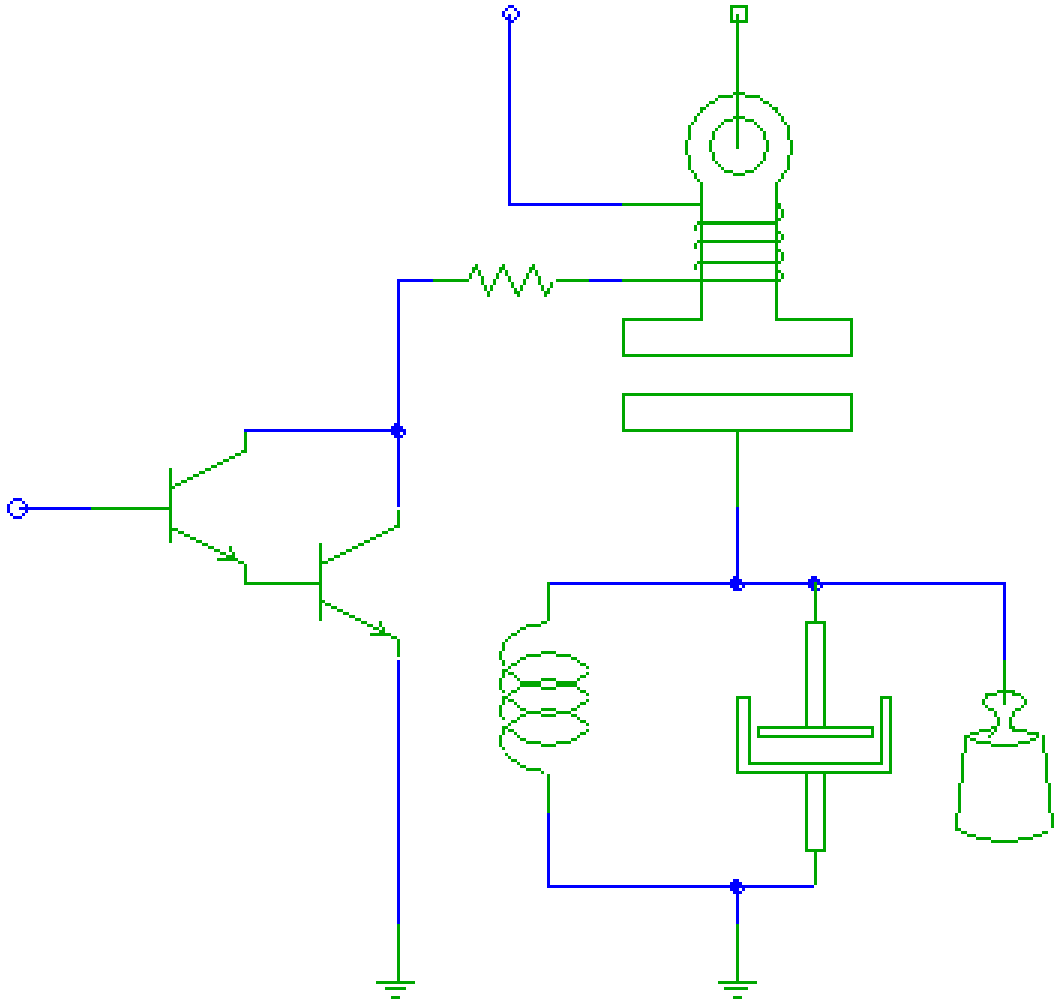
Part A: Lumped Element Simulation



Examples

Saber

Lumped Element Simulation



Simulation Differences

Lumped Element	Integrated System Simulation
o discrete components with component parameters (e.g. capacitor with capacity value)	o functional blocks with behavioral description / parameters (e.g. capacitor as integrator of current)
+ physical units possible	- no physical units (normalized variables)
- Real-Time simulation not (yet) possible	+ Real- Time simulation possible with coupling to the real world (real hardware)

will be continued…

Tomorrow

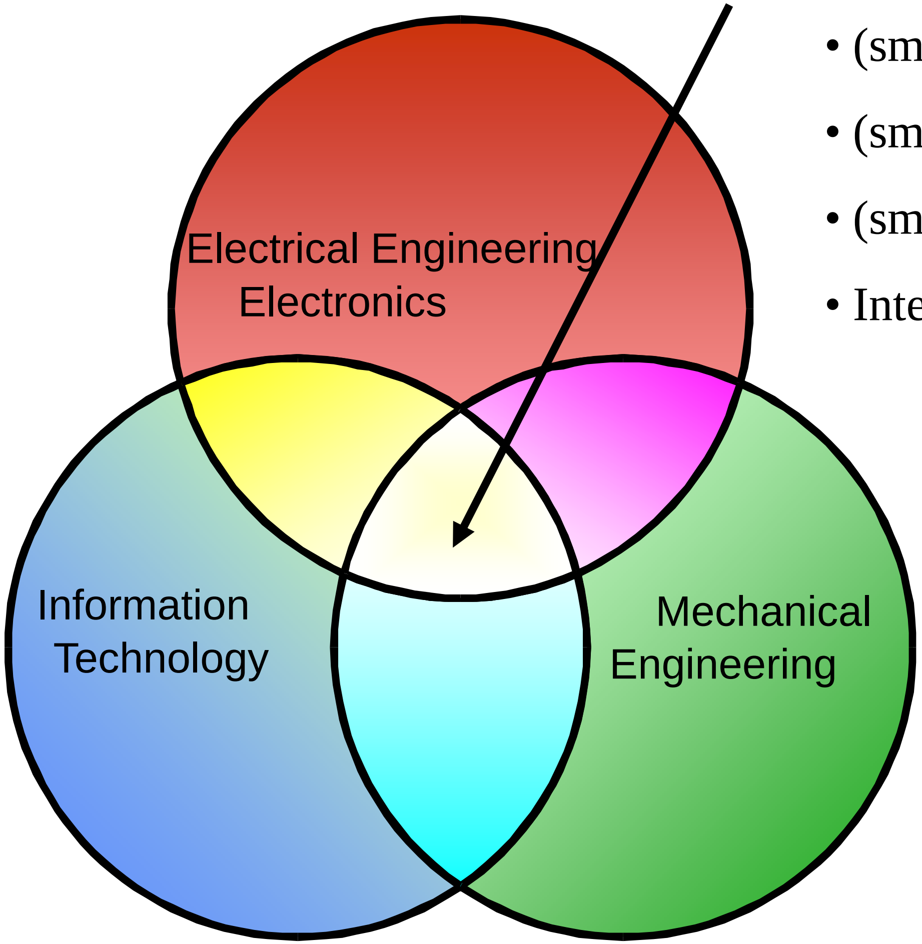
splitted in groups

find more information:

http://www.mechatronics.fh-aachen.de/

Definition

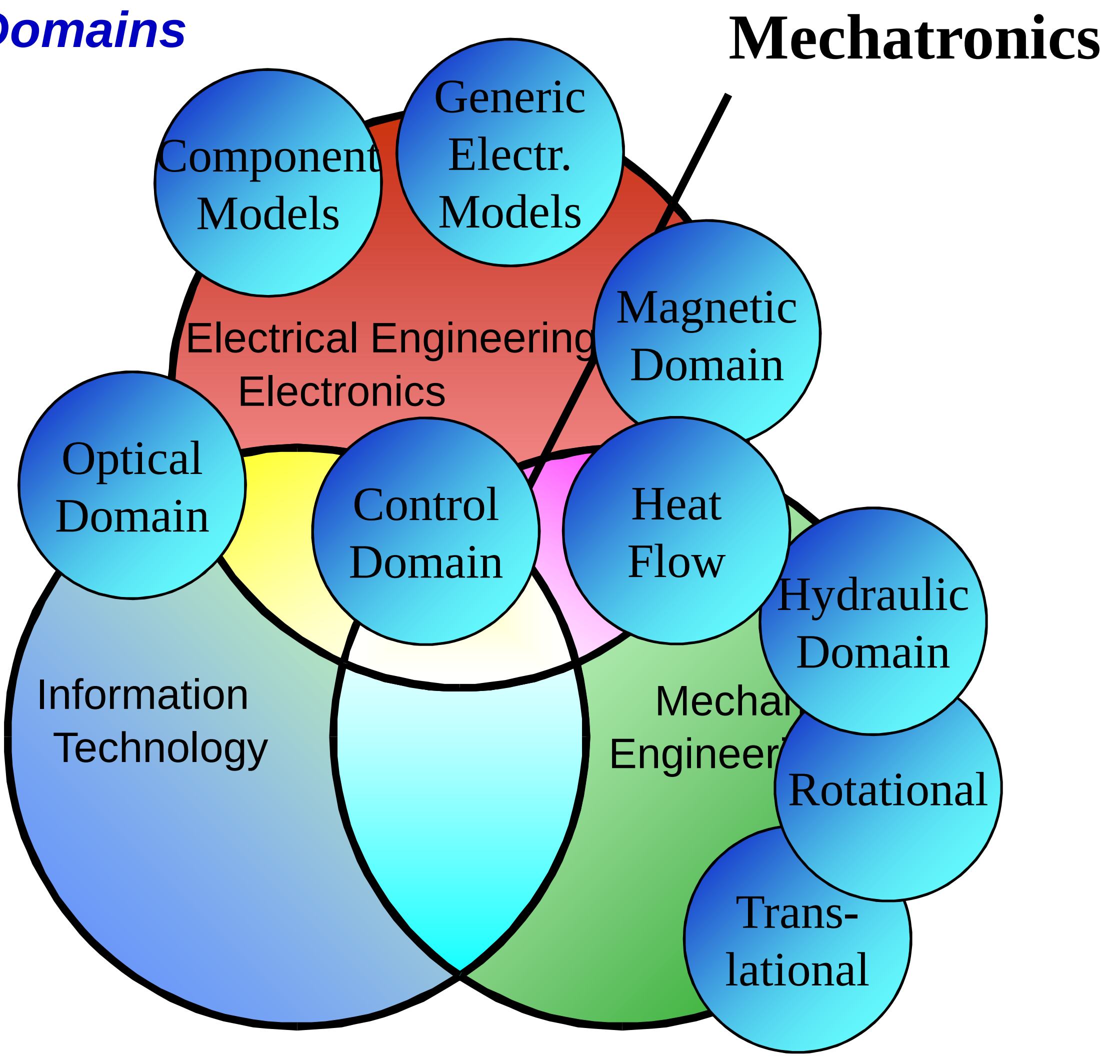
Mechatronics



- (smart) Actuators
- (smart) Sensors
- (smart) Sensor/Actuator- Systems
- Interconnection Systems

- Mikrosystems
- Robotics
- Multi Domain Simulation
- HIL
- Rapid Controller Prototyping

Domains


Mechatronics
Component Models
Generic Electr. Models
Magnetic Domain
Electrical Engineering Electronics
Optical Domain
Control Domain
Heat Flow
Hydraulic Domain
Information Technology
Mechan Engineer
Rotational
Trans- lational

Lumped Element Simulation

Kirchhoff's First Law

Mechanical Equivalent:
The sum of forces in one node is zero

The algebraic sum of the currents flowing through a junction is zero. Currents approaching the junction are + while currents going away from the junction are -.

Kirchh Law

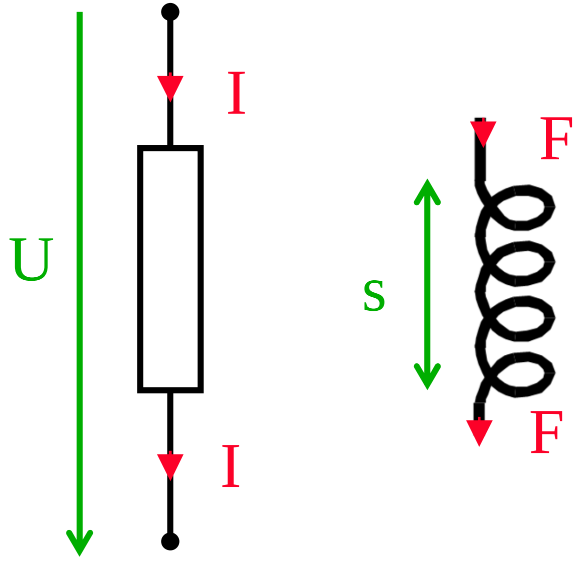
Important:
Distance as a **signed** variable

Mechanical Equivalent:
The sum of distances in a loop is zero

The algebraic sum of the potential differences in a circuit loop must be zero. Potential rises are + while potential drops are -.

Two different types of variables

- Across variables:
  - Voltage, voltage drop, distance, velocity, pressure, temperature difference

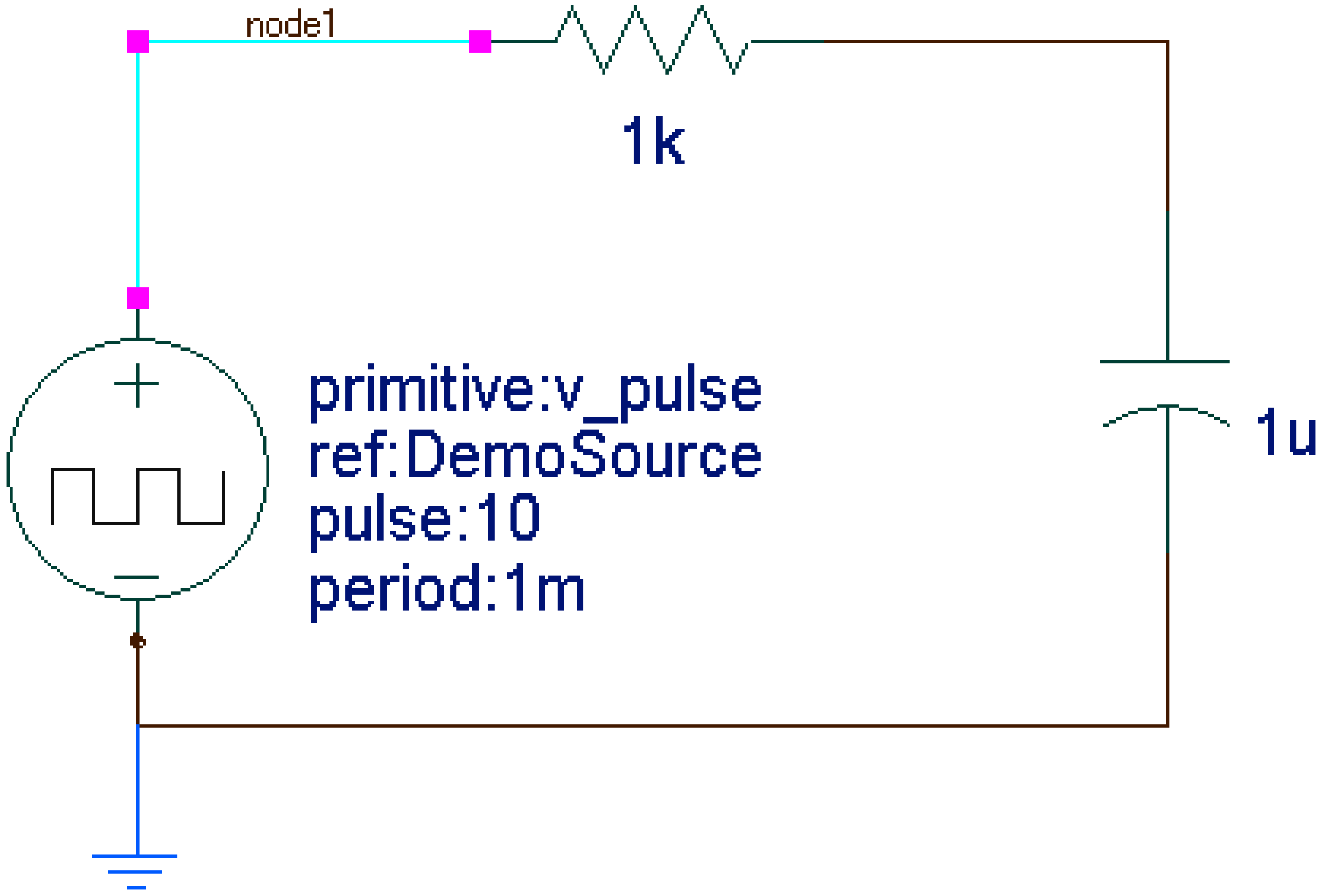


- Through variables:
  - (electric) current, force, liquid flow, heat flow

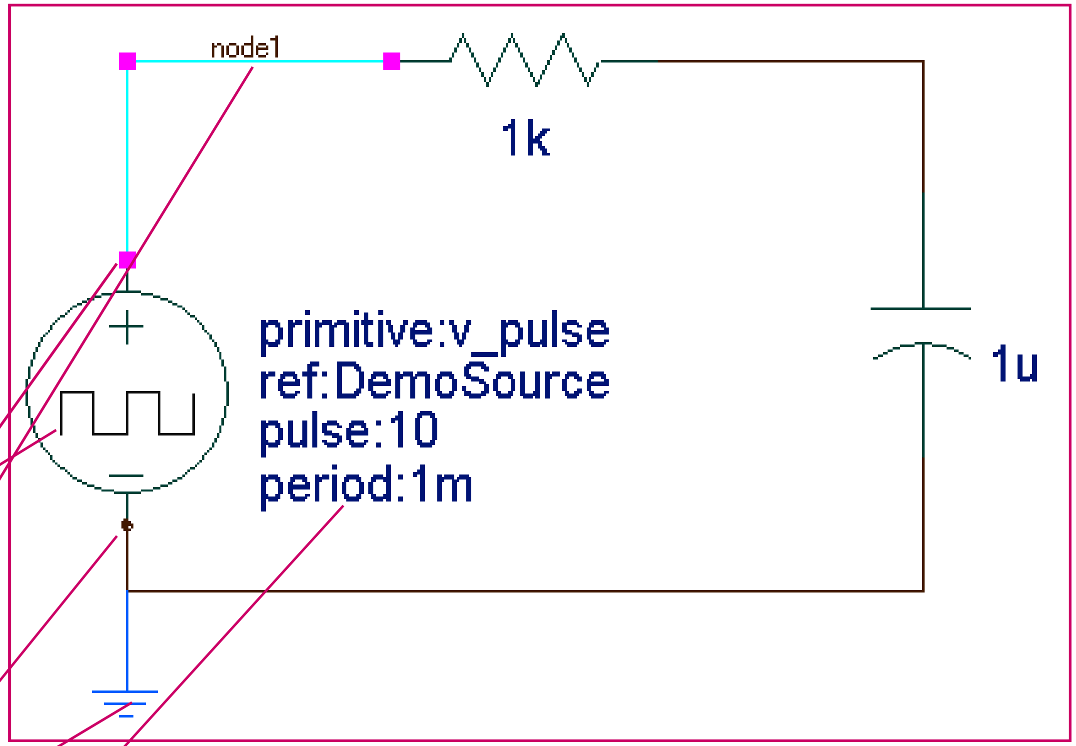
Through and Across variables in different domains

Domain	**Through variable**	**Across variable**
Electronics	Current	Voltage
Translational mechanics	Force	Position, velocity
Rotational mechanics	Torque	Angle, angular velocity
Thermal systems	Power	Temperature
Hydraulic systems	Flow rate	Pressure
Magnetic systems	Flux	Magnetomotive force

Saber Sketch Circuit Design Example: RC- element

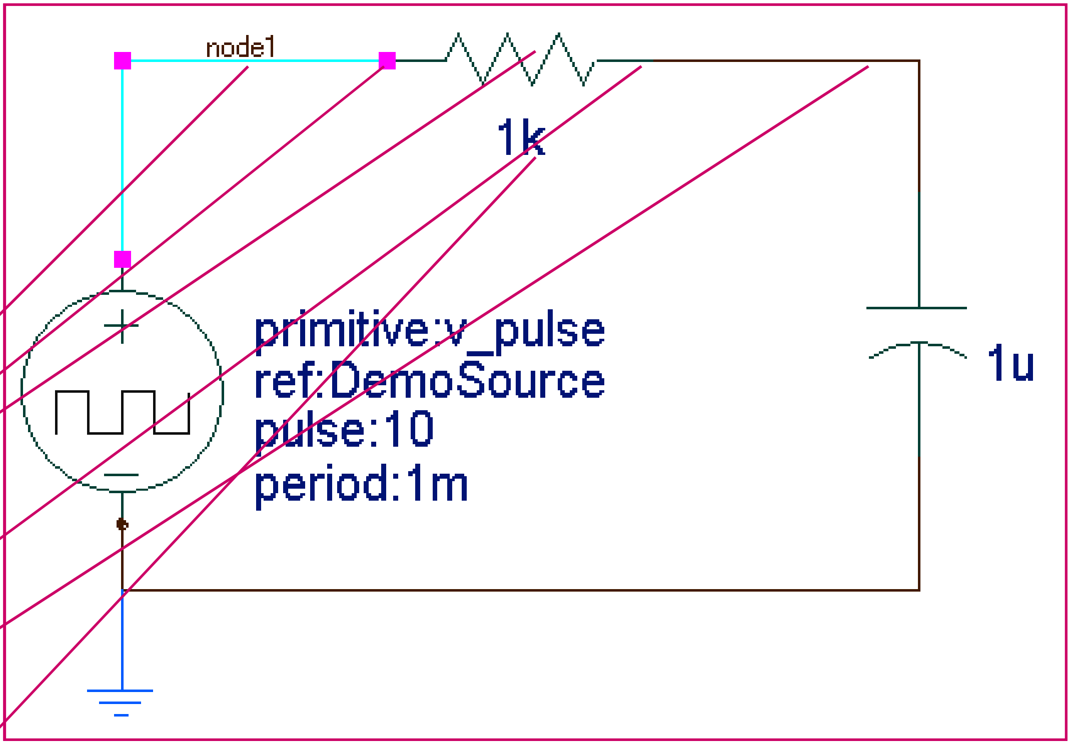


*Modell RC- Demo Netlisting (Filename *.sin)*



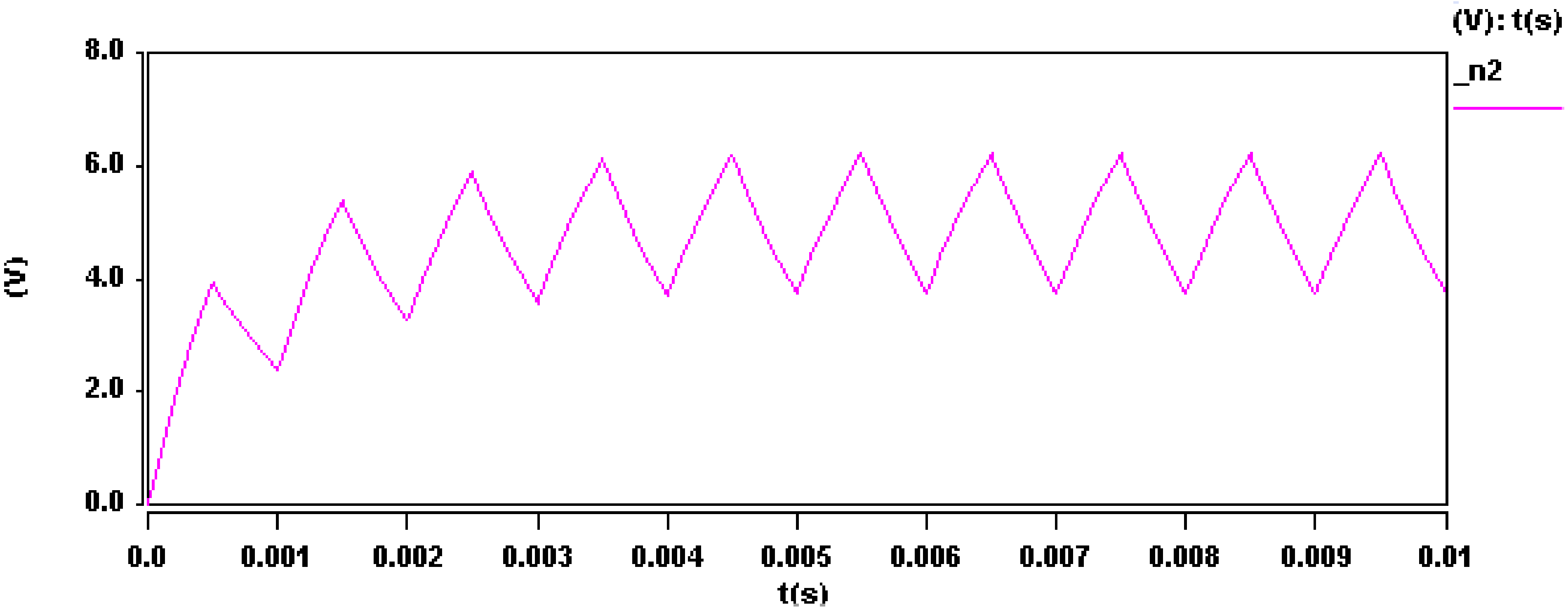
v_pulse.DemoSource p:node1 m:0 = period=1m, initial=0, width=0.5m, pulse=10
r.r1 p:node1 m:_n2 = rnom=1k
c.c1 p:_n2 m:0 = c=1u

*Modell RC- Demo Netlisting (Filename *.sin)*



v_pulse.DemoSource p:node1 m:0 = period=1m, initial=0, width=0.5m, pulse=10
r.r1 p:node1 m:_n2 = rnom=1k
c.c1 p:_n2 m:0 = c=1u

RC-Demo Simulation Result



Caution! Traps are everywhere in Saber

- If nothing else is selected all units are entered in the mks - system
- Entries are not case- sensitive (m = M = milli and not Mega!)
- Units are integral part of variables.
- While entering a value of a parameter only the decimal power (eg. k = 10^3) is provided, not the physical unit (a value for the mass entered as 50g does not mean 50 grams but means 50 Giga kilograms)
- Don’t use special characters like German “Umlaute” (ä,ö,ü,ß)
- The decimal point is a dot, not a comma
- Don’t forget the Gnd- node (Saber Reference Level Null)

Table of Prefixes for Decimal Power in Saber

Decimal Power	Prefix	Saber letter
10^{12}	Tera	T
10^{9}	Giga	G
10^{6}	Mega	Meg (caution: not M)
10^{3}	kilo	k
10^{-3}	milli	m
10^{-6}	mikro	u
10^{-9}	nano	n
10^{-12}	pico	p
10^{-15}	femto	f
10^{-18}	atto	a