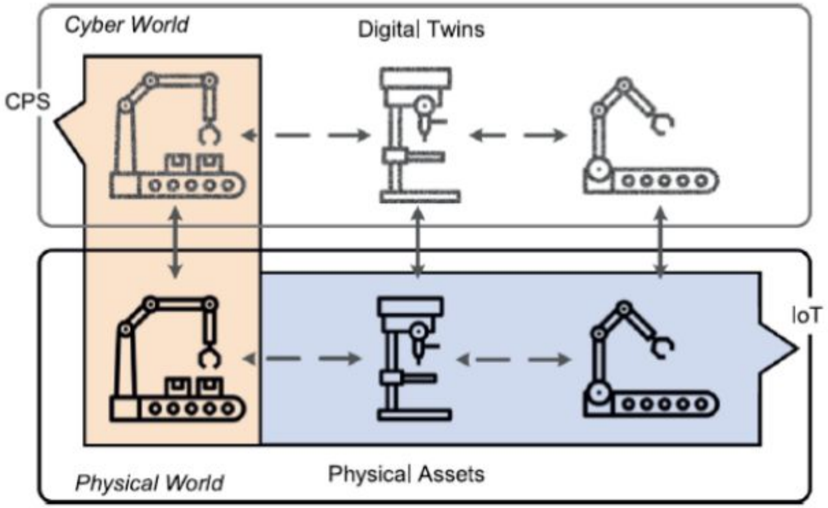
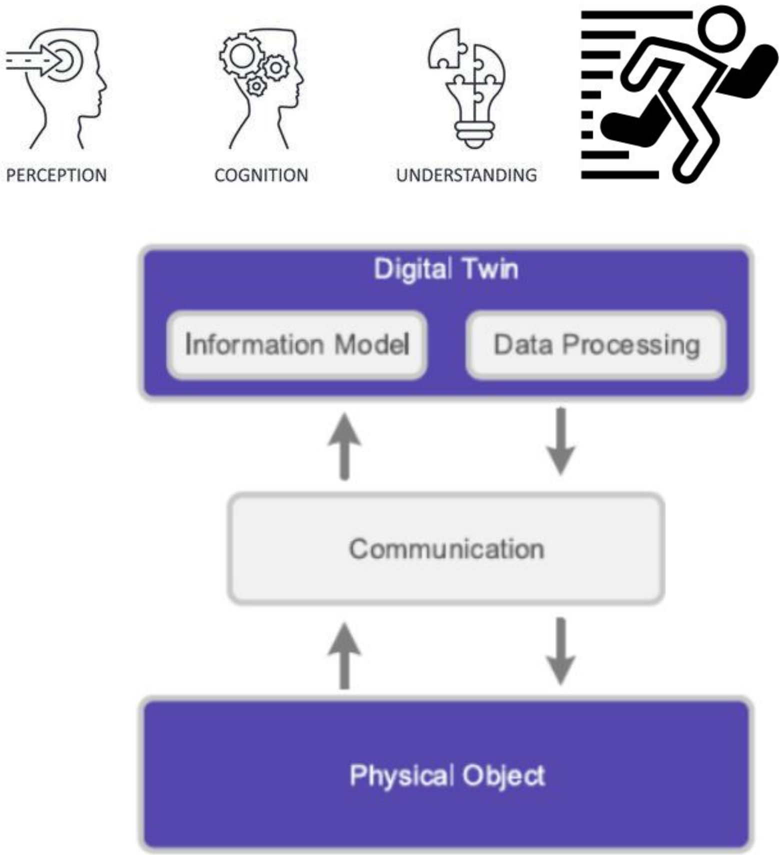
Simulation of digital twin and introduction to hardware communication

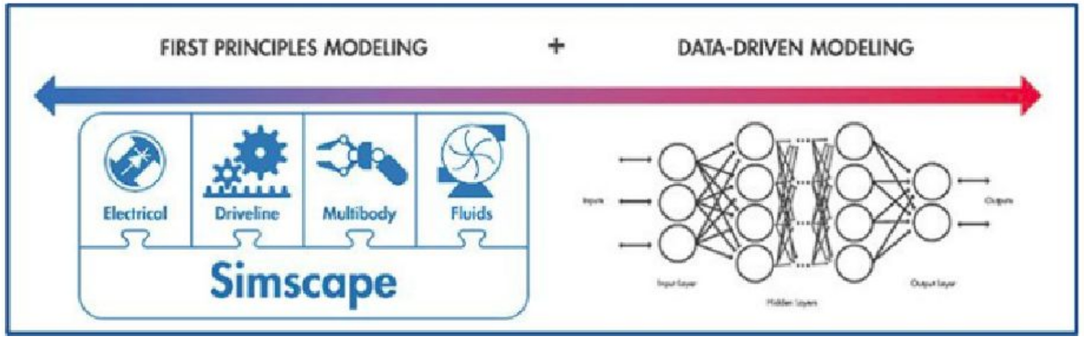
Speaker: Eng. Giulio Mattera

mail:giulio.mattera@unina.it

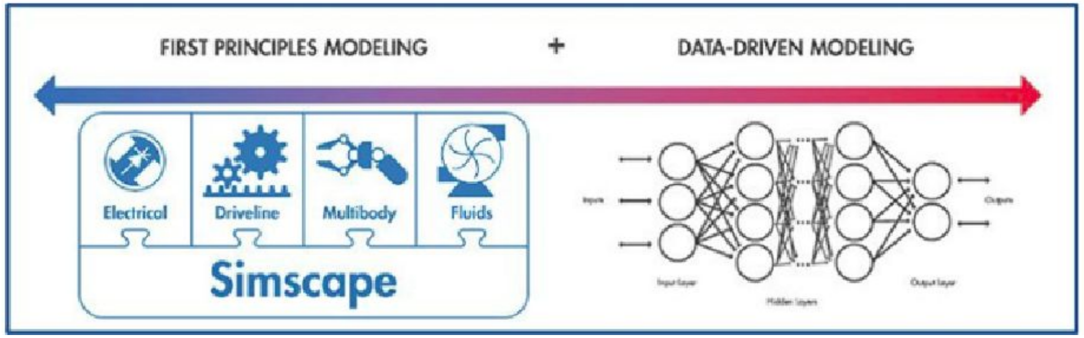
Recap


Cyber World
Digital Twins
CPS
IoT
Physical World
Physical Assets


PERCEPTION
COGNITION
UNDERSTANDING
Digital Twin
Information Model
Data Processing
Communication
Physical Object


FIRST PRINCIPLES MODELING
+
DATA-DRIVEN MODELING
Electrical
Driveline
Multibody
Fluids
Simscape

Modeling : what it means?



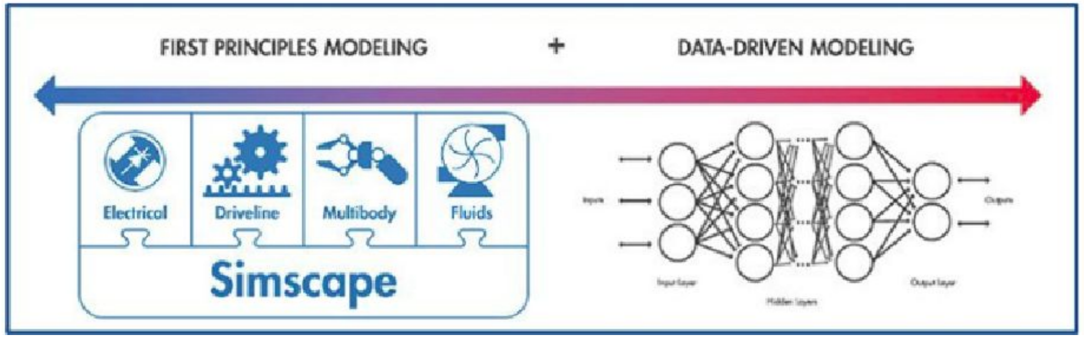
A model is a description of a system using mathematical concepts. In general, two classes of models can be identified:

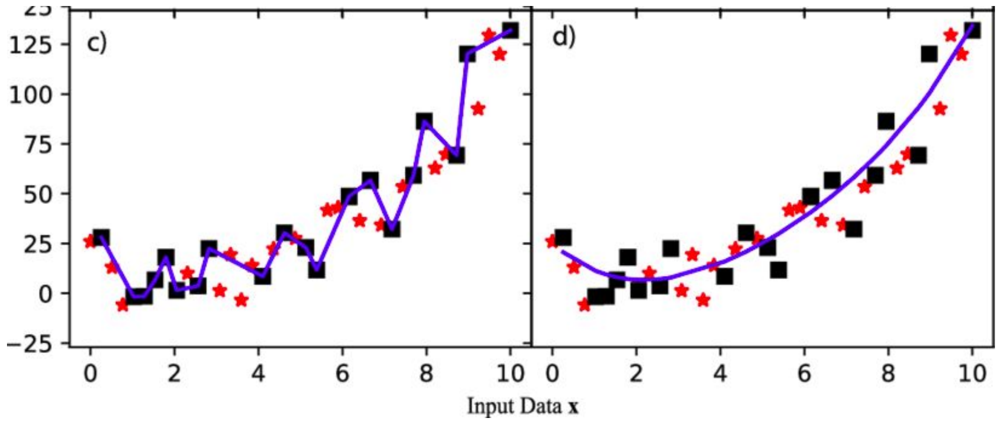
- models based purely on (many) experiments;
- models based on physical first principles and (a few) parameter identification experiments.

Modeling : what it means?

A model is a description of a system using mathematical concepts. In general, two classes of models can be identified:

- models based purely on (many) experiments;
- models based on physical first principles and (a few) parameter identification experiments.





https://www.w3schools.com/python/python_ml_polynomial_regression.asp

https://www.geeksforgeeks.org/python-implementation-of-polynomial-regression/

Modeling : what it means?

A model is a description of a system using mathematical concepts. In general, two classes of models can be identified:

- models based purely on (many) experiments;
- models based on physical first principles and (a few) parameter identification experiments.

```python
import numpy as np
from scipy.integrate import odeint
import matplotlib.pyplot as plt

# Initialization
T_START, T_STOP, TS  = 0, 40, 0.1
# Initial condition
X0 = [0,0]
t = np.arange(T_START, T_STOP+1, TS)
# Function that returns dx/dt
def msd_dynamics(x, t):

    c = 4 # Damping constant
    k = 2 # Stiffness of the spring
    m = 20 # Mass
    F = 5

    dx1dt = x[1]
    dx2dt = (F - c*x[1] - k*x[0])/m

    dxdt = [dx1dt, dx2dt]

    return dxdt

# Solve ODE
x = odeint(msd_dynamics, X0, t)


x1 = x[:,0]
x2 = x[:,1]

# Plot the Results
plt.plot(t,x1)
plt.plot(t,x2)
plt.title('Simulation of Mass-Spring-Damper System')
plt.xlabel('t')
plt.ylabel('x(t)')
plt.legend(["x1", "x2"])
plt.grid()
plt.show()
```

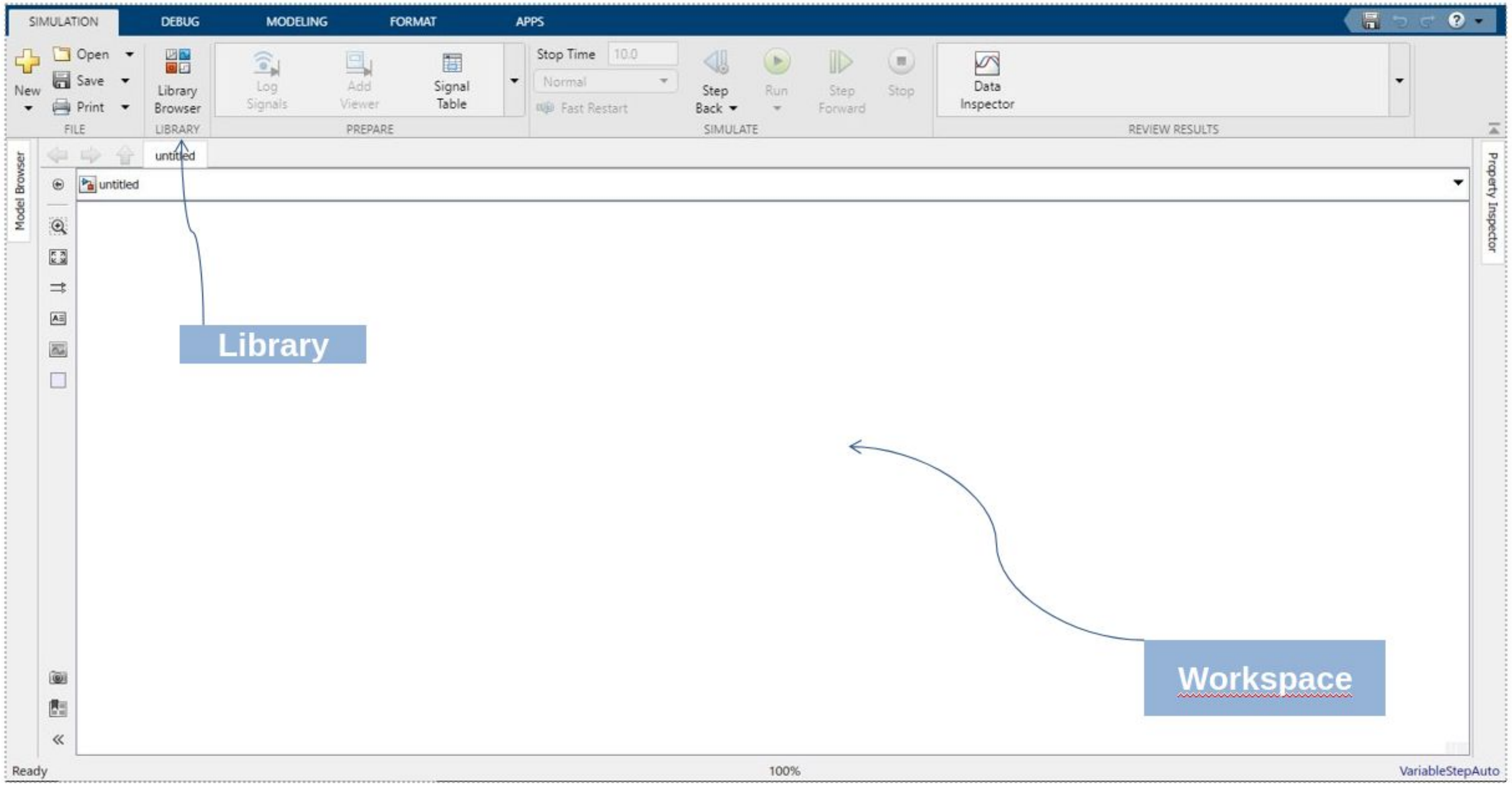
Introduction to Simulink

- Block-diagram environment
- Model, simulate, and analyze multidomain systems
- Design, implement, and test:
    - Control systems
    - Signal processing systems
    - Communications systems
    - Other dynamic systems
- Platform for Model-Based Design
- Mathematical modeling approach

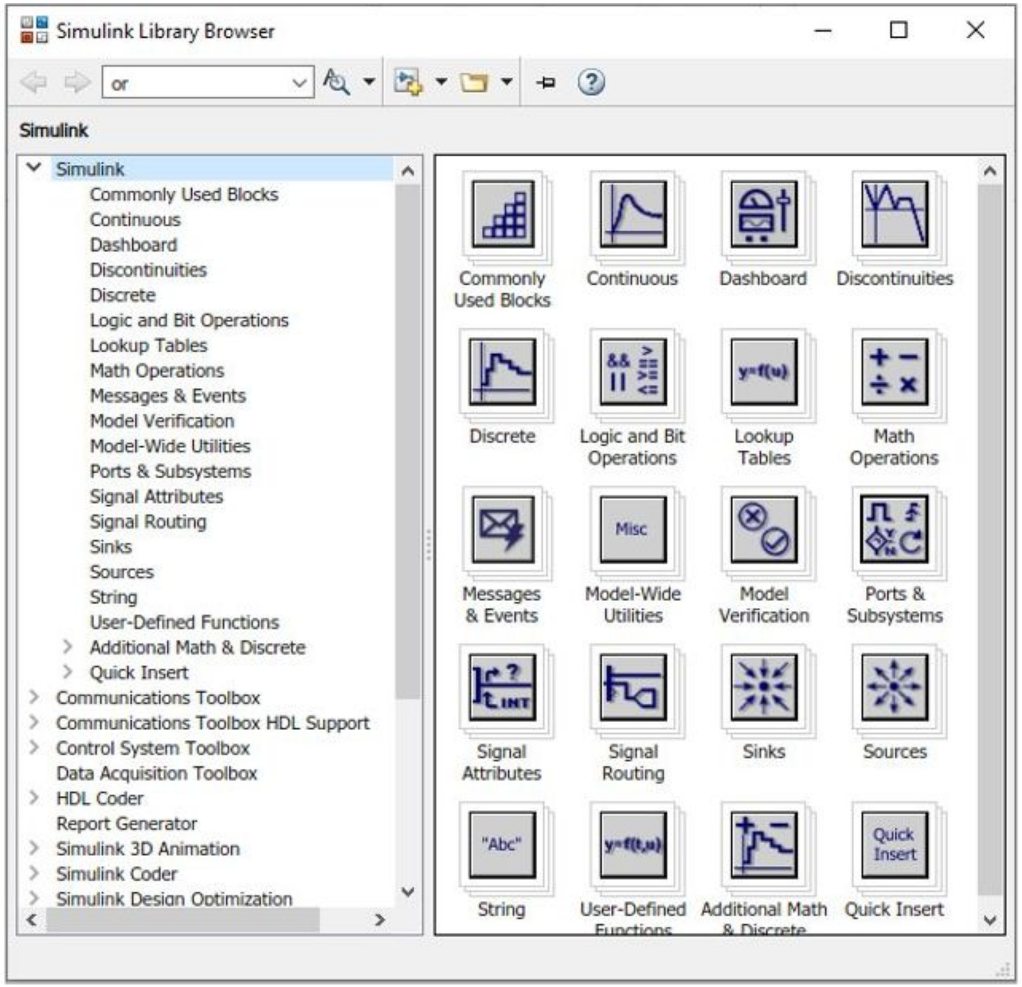
Introduction to Simulink

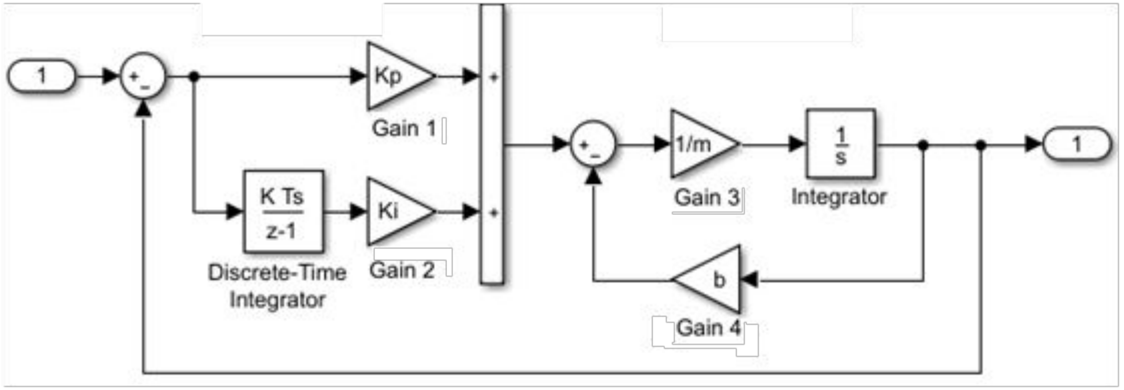
- Graphical modeling and simulation environment
- Libraries of continuous-time and discrete-time blocks
- Simulation engine with fixed-step and variable-step ODE solvers
- Scopes and data displays for viewing simulation results
- Project and data management tools
- MATLAB Function block for importing MATLAB algorithms
- Legacy Code Tool for importing C and C++ code into models

Introduction to Simulink



Introduction to Simulink



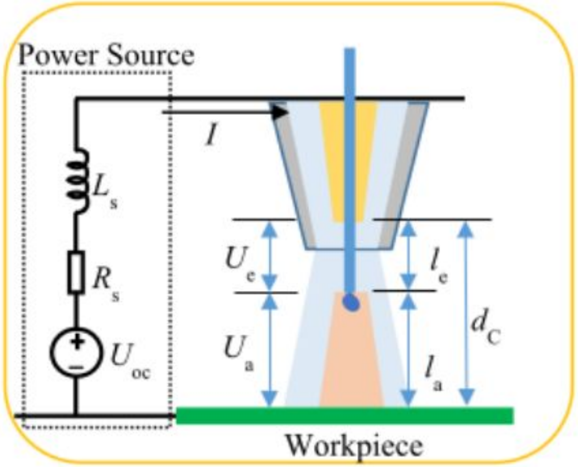


DEMO : Simulink key concepts

DEMO : MSD

DEMO : DC Motor

DEMO : Gas Metal Arc Welding model



$$V = L\frac{di}{dt} + i(R + R_a + R_w) + l_a E_a + ilk_e + U_0$$

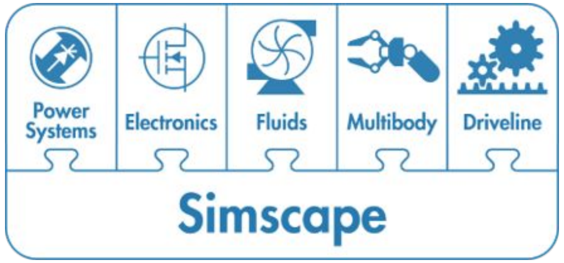
$$\dot{m} = (c_1 i + c_2 l_s i^2)\rho_w$$

$$\dot{l}_s = WFS - \frac{1}{\pi r_w^2}\dot{m}$$

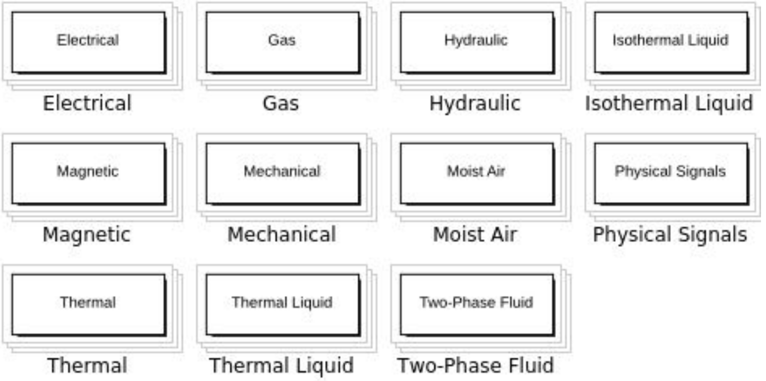
$$l_a = CTWD - l_s$$

Multiphysical modeling : Simscape

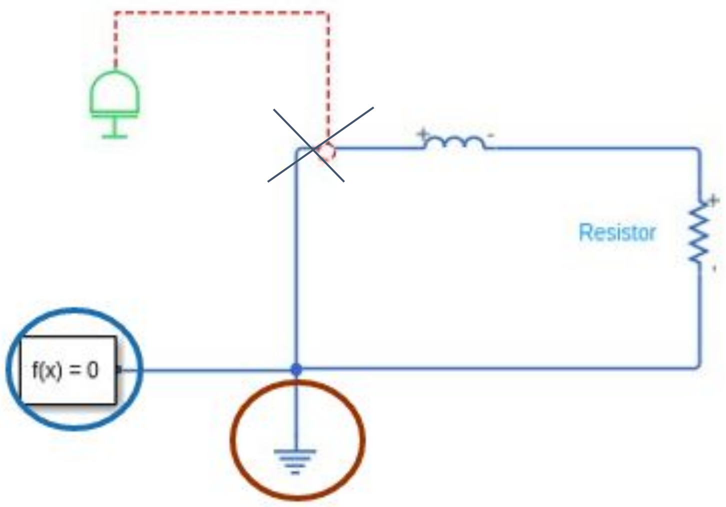
- Is a graphical tool in simulink ecosystem that allow us to model very complex multyphysical systems using physical block instead of equations



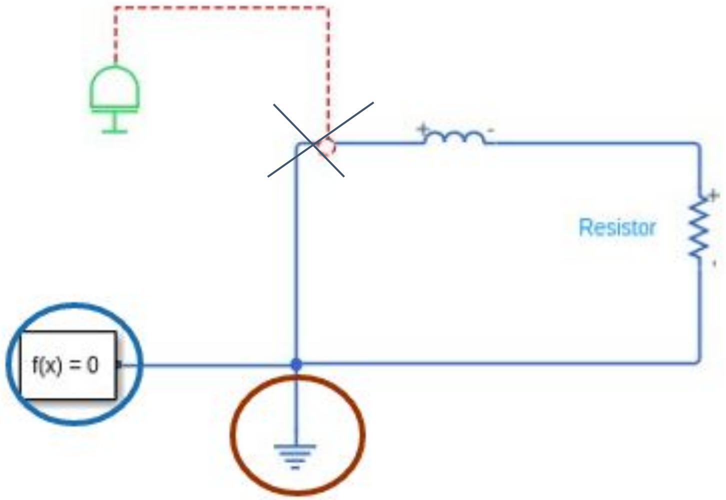
- There are several libraries with basics physical components that you can drop in your working envirnoment, like simulink blocks

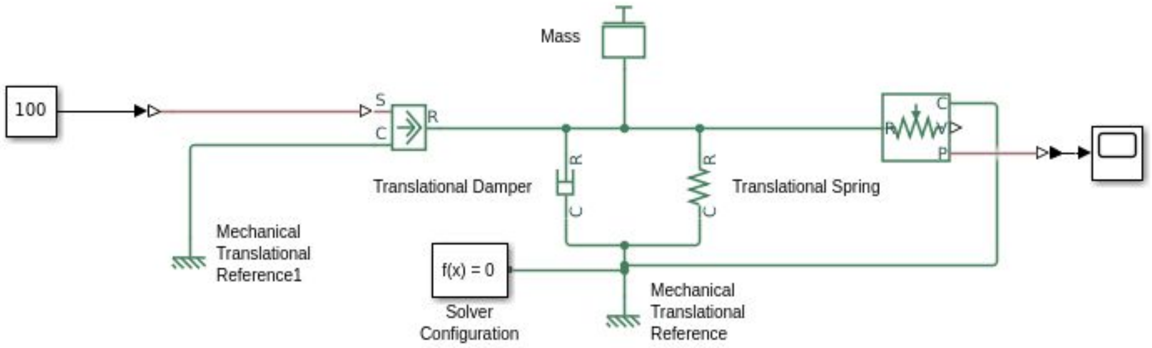


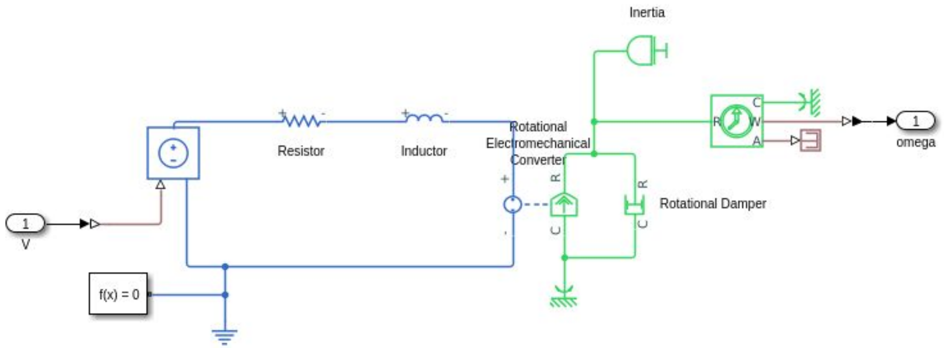
Multiphysical modeling : Simscape



Multiphysical modeling : Simscape







Why physical modeling?

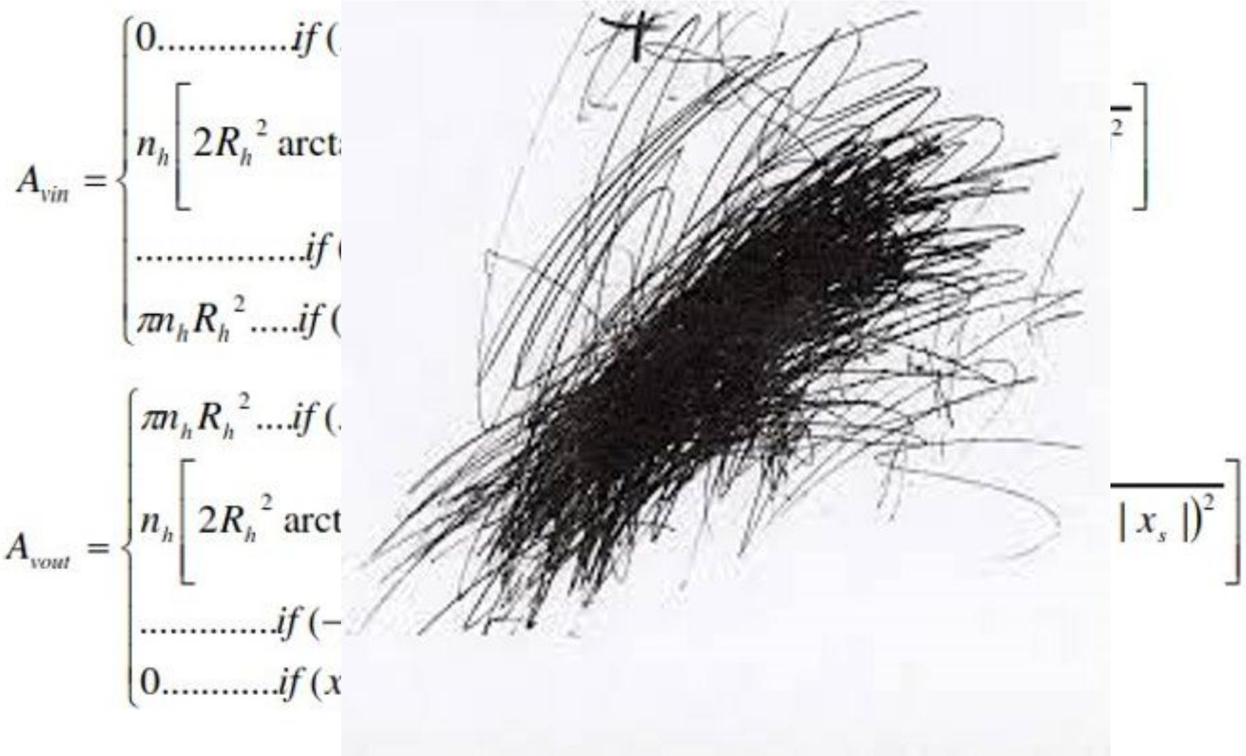
$$A_{vin} = \begin{cases} 0 \ldots\ldots\ldots\ldots if\,(x_s) \le (p_w - R_h) \\ n_h \left[2R_h^{\,2} \arctan\left(\sqrt{\dfrac{R_h - p_w + x_s}{R_h + p_w - x_s}} \right) - (p_w - x_s)\sqrt{R_h^{\,2} - (p_w - x_s)^2} \right] \\ \ldots\ldots\ldots\ldots\ldots\ldots if\,(p_w - R_h) < x_s < (p_w + R_h) \\ \pi n_h R_h^{\,2} \ldots\ldots if\,(x_s) \ge (p_w + R_h) \end{cases}$$

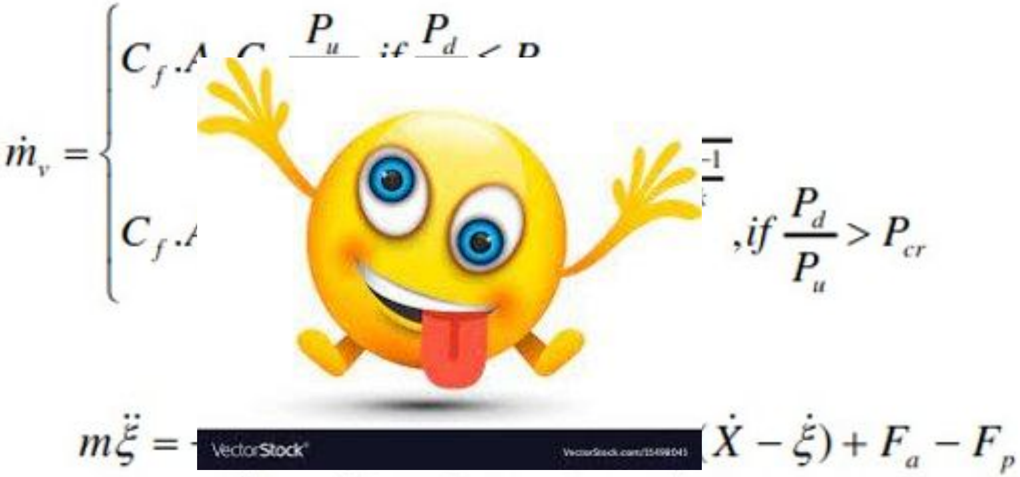
$$\dot{m}_v = \begin{cases} C_f . A_v . C_1 . \dfrac{P_u}{\sqrt{T}} , if\ \dfrac{P_d}{P_u} \le P_{cr} \\ C_f . A_v . C_2 . \dfrac{P_u}{\sqrt{T}} . \left(\dfrac{P_d}{P_u} \right)^{\frac{1}{k}} \sqrt{1 - \left(\dfrac{P_d}{P_u} \right)^{\frac{k-1}{k}}} , if\ \dfrac{P_d}{P_u} > P_{cr} \end{cases}$$

$$A_{vout} = \begin{cases} \pi n_h R_h^{\,2} \ldots if\,(x_s) \le (-p_w - R_h) \\ n_h \left[2R_h^{\,2} \arctan\left(\sqrt{\dfrac{R_h - p_w + |\,x_s\,|}{R_h + p_w - |\,x_s\,|}} \right) - (p_w - |\,x_s\,|)\sqrt{R_h^{\,2} - (p_w - |\,x_s\,|)^2} \right] \\ \ldots\ldots\ldots\ldots\ldots if\,(-p_w - R_h) < x_s < (R_h - p_w) \\ 0 \ldots\ldots\ldots\ldots if\,(x_s) \ge (R_h - p_w) \end{cases}$$

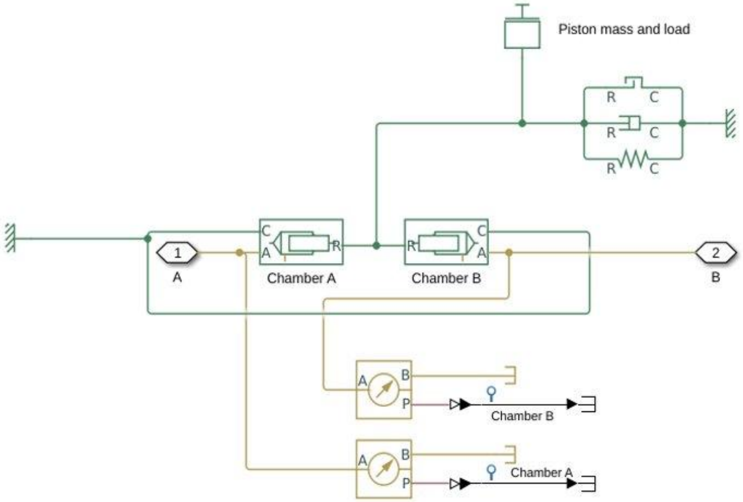
$$m\ddot{\xi} = -k_1 \xi - c_1 \dot{\xi} + k_s (X - \xi) + c_s (\dot{X} - \dot{\xi}) + F_a - F_p$$

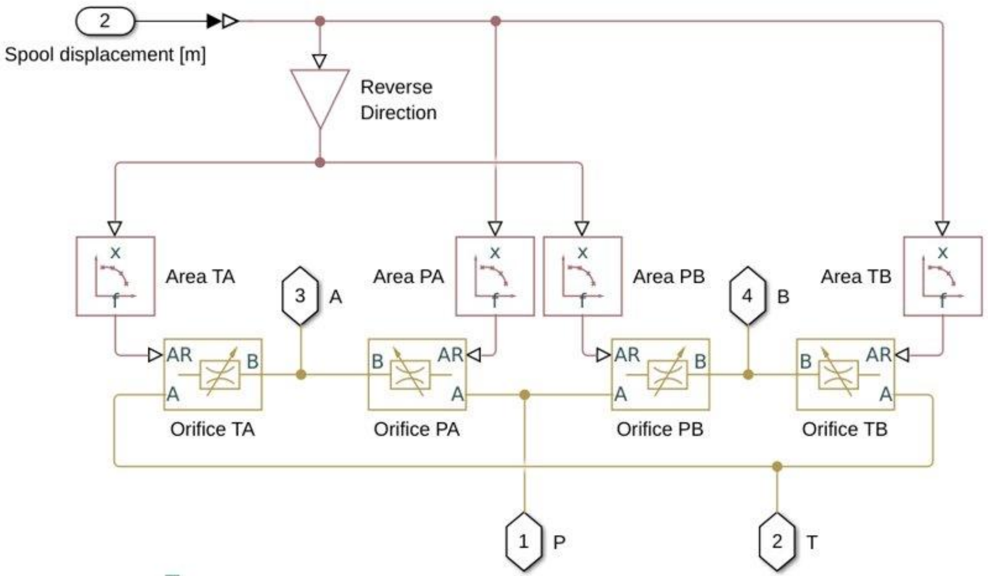
Why physical modeling?

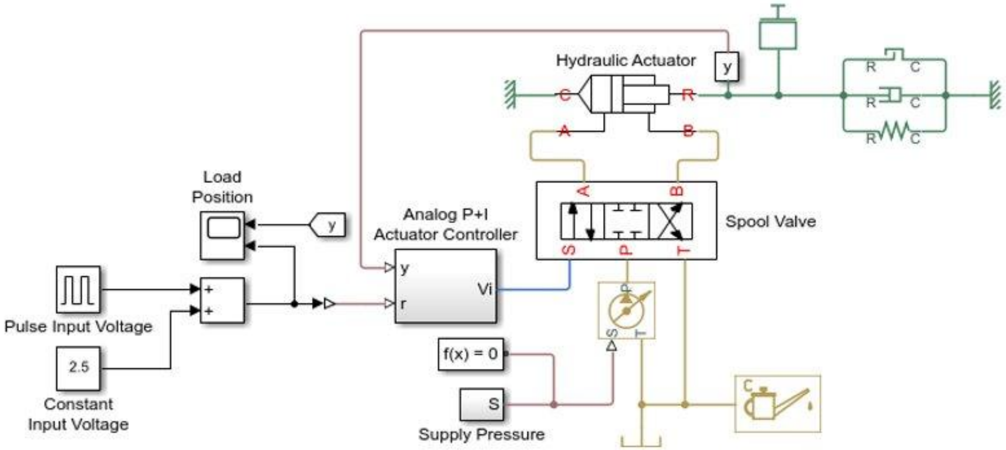




Why physical modeling?





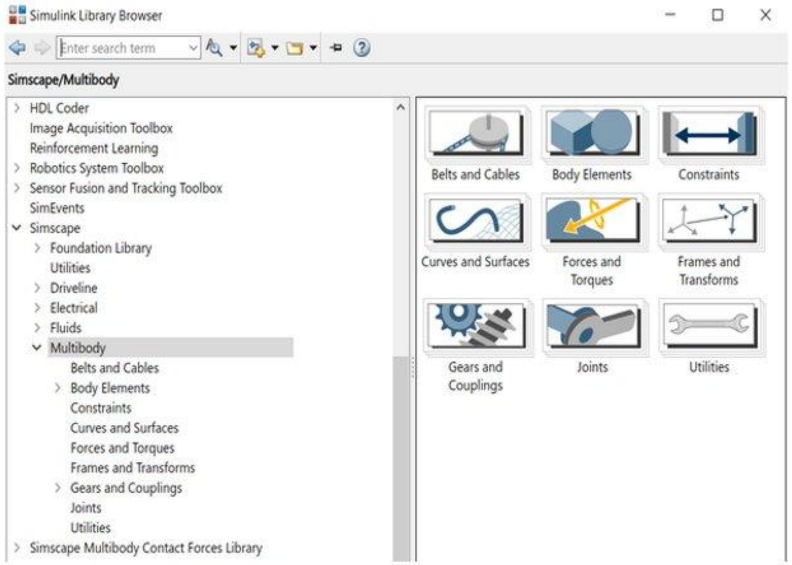


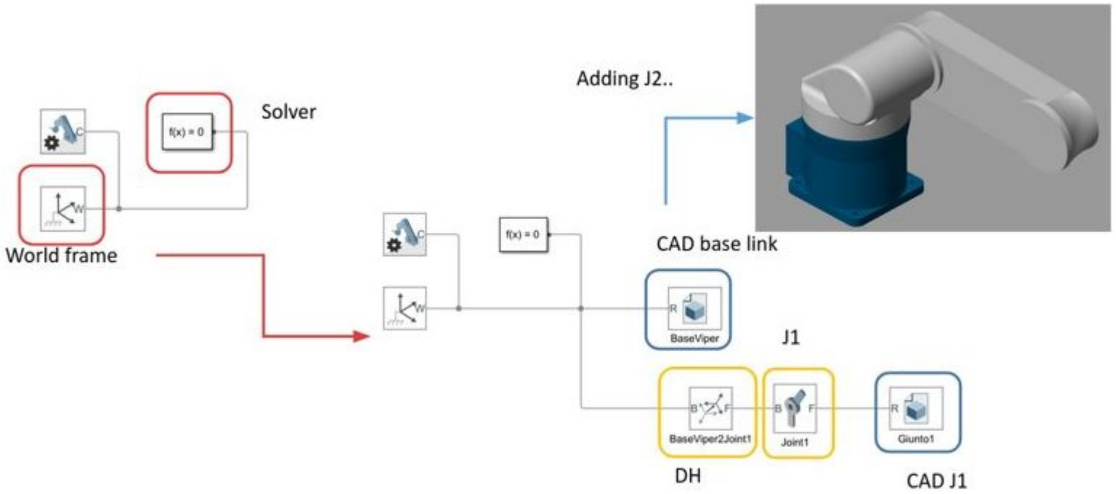
DEMO : MSD

DEMO : DC Motor

Multibody library

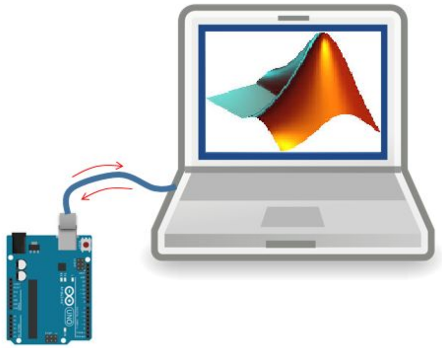
- Multibody library allow you to visualize your models
- You can create your robotic structures thought
  - Frames and transformations
  - Joints
  - Body elements (link)
  - Other kinematics constraints and couplings

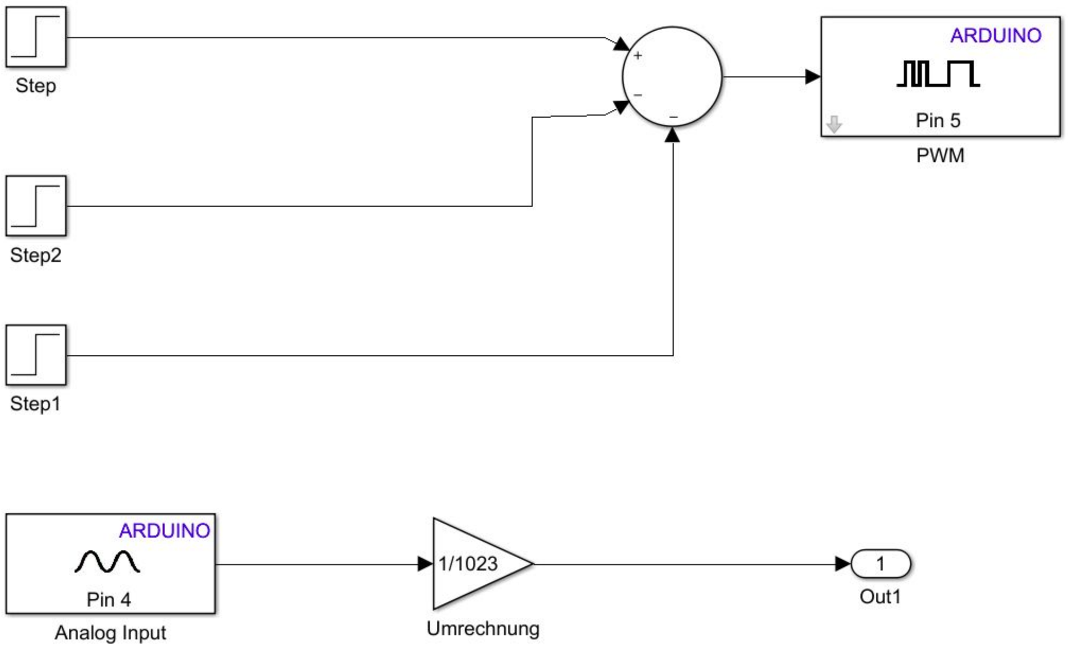




DEMO : Developing and control a mechatronic system model with simscape and multibody library

Connect models with hardware





https://support.industry.siemens.com/cs/document/109749187/digitalization-with-tia-portal-virtual-commissioning-with-simatic-and-simulink?dti=0&lc=en-DZ

https://www.youtube.com/watch?v=vNpyb0YgTDY

https://it.mathworks.com/videos/run-models-interactively-on-arduino-and-raspberry-pi-1549462466264.html