How to harness the power of Python programming within the Simba simulation environment

By Peyman Razmi

March 28th, 2024

POWERSYS

Vaasan yliopisto
UNIVERSITY OF VAASA

How to harness the power of Python programming within the Simba simulation environment

AGENDA

- Introduction to Python Programming
- Python environment and Libraries
- Introduction to Simba environment, its capabilities
- Power electronic simulation in Simba
- Management of Simba Simulation with Python
- Introduction to Julia programming in Simba simulation

POWERSYS

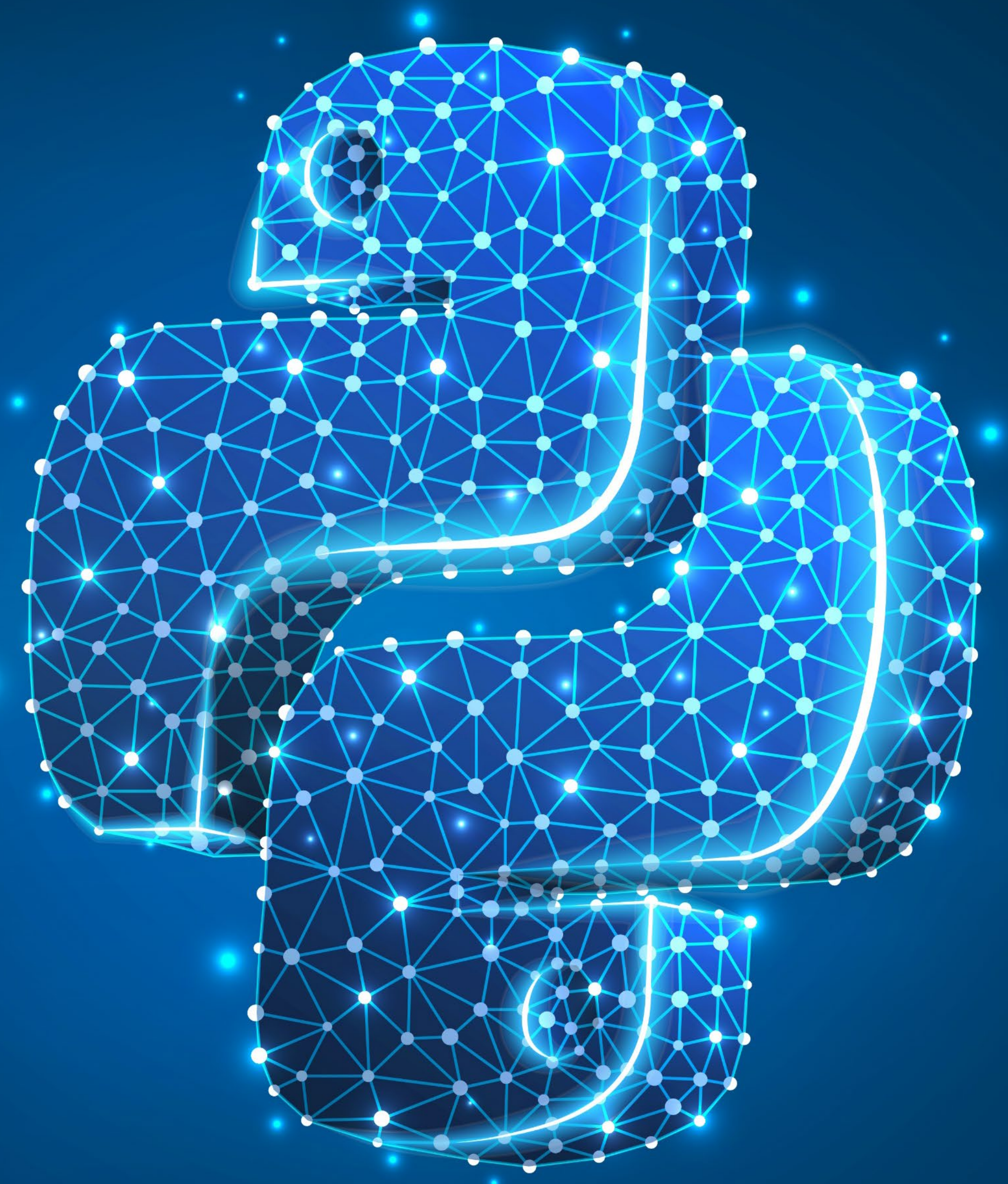
University of Vaasa
FINLAND

What is Python?

Python is a popular high-level programming language used in various applications

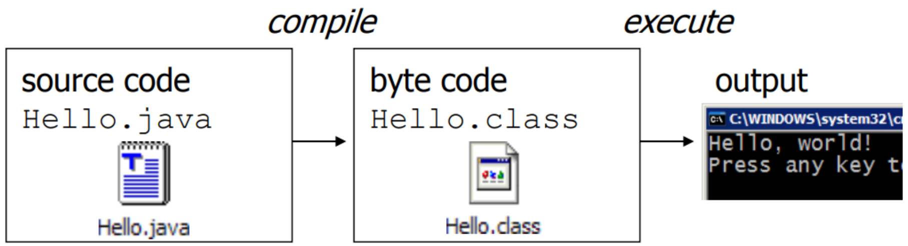
- Python is an easy language to learn because of its simple syntax

- Python can be used for simple tasks such as plotting or for more complex tasks like Algebra programming, optimization and machine learning algorithms.



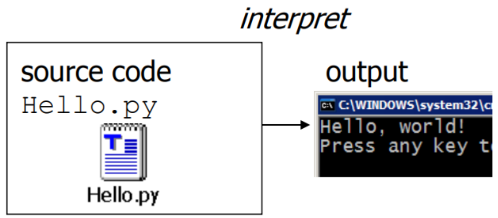
Introduction to Python

Many languages require you to compile (translate) your program into a form that the machine understands.



Python

Python is instead directly interpreted into machine instructions.

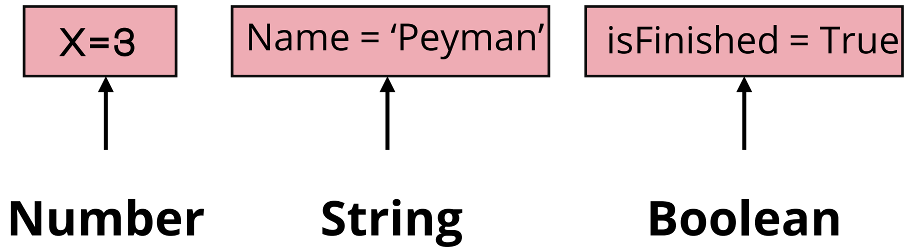


The interpreter provides an interactive environment to play with the language

Variables, Objects, and Classes

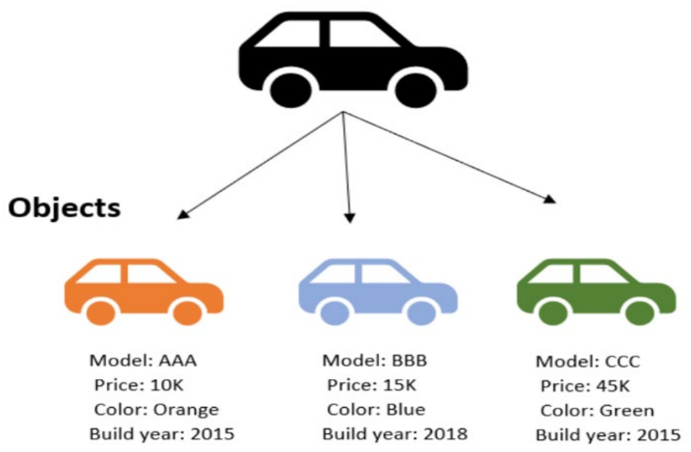
- A variable is a reference to a value stored in a computer’s memory.

- Variables can be sorted into a variety of categories (or data types) such as numbers (int/float etc), Boolean values (true/false), and sequences (strings, lists etc).



- An object is a collection of data from a computer’s memory that can be manipulated.

Object Example

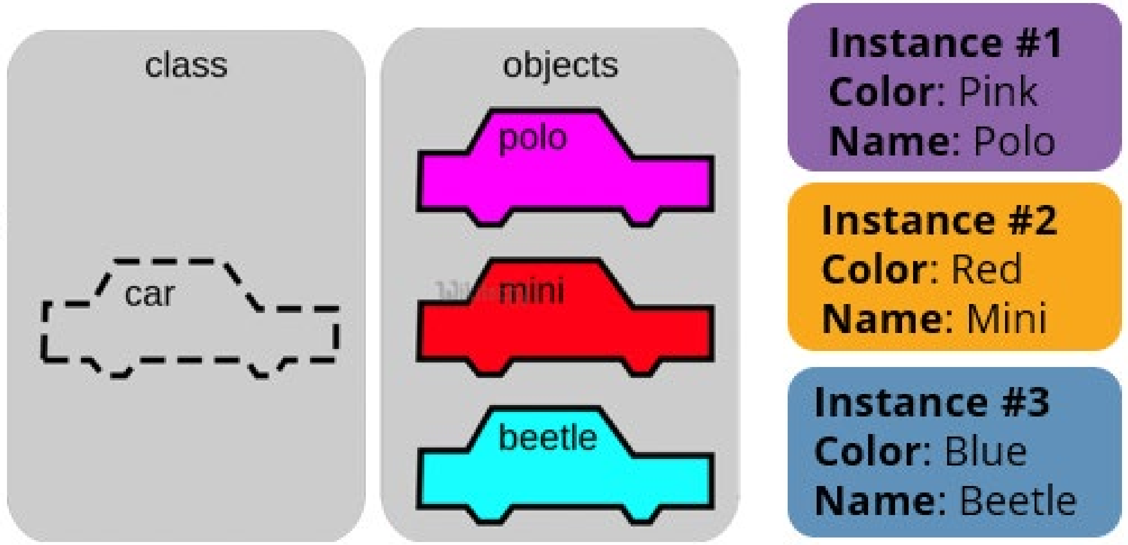


- **Color:** The color of the car (e.g., red, blue, black).
- **Make:** The manufacturer of the car (e.g., Ford).
- **Model:** The model name or number (e.g., Camry, F-150, X5).
- **Year:** The manufacturing year of the car.
- **Engine Size:** The size of the engine (e.g., 2.0L, 3.5L).
- **Fuel Type:** The type of fuel the car uses (e.g., gasoline, diesel, electric).
- **Mileage:** The number of miles or kilometers the car has traveled.
- **IsNew:** A boolean variable indicating whether the car is new or used.

Classes

A **class** is a collection of objects who share the same set of variables.

- The definition of the class provides a blueprint for all the objects within it **(instances)**.
- Instances may share the same variables (color, size, shape, etc.), but they do **NOT** share the same values for each variable (blue/red/pink, small/large, square/circular etc.)



Methods

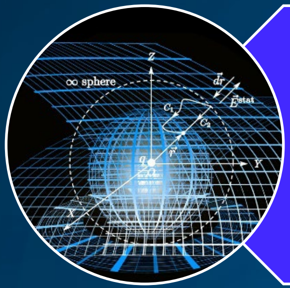
Methods are the functions used to act on/alter an object's data.
They describe what your object can "do."

Methods define actions or behaviors that the car can perform

Functions

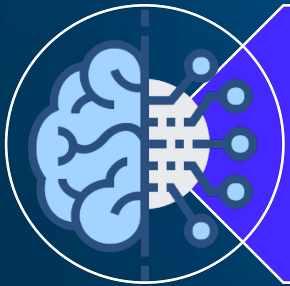
- Start() — Initiates the car's engine
- Stop() — Stops the car's engine
- Accelerate() — Increases the car's speed
- Brake() — Decreases the car's speed
- Dashboard() — Displays info about Current Status

Python Libraries



Scientific Computing

- Numpy
- Pandas
- Scipy
- Matplotlib



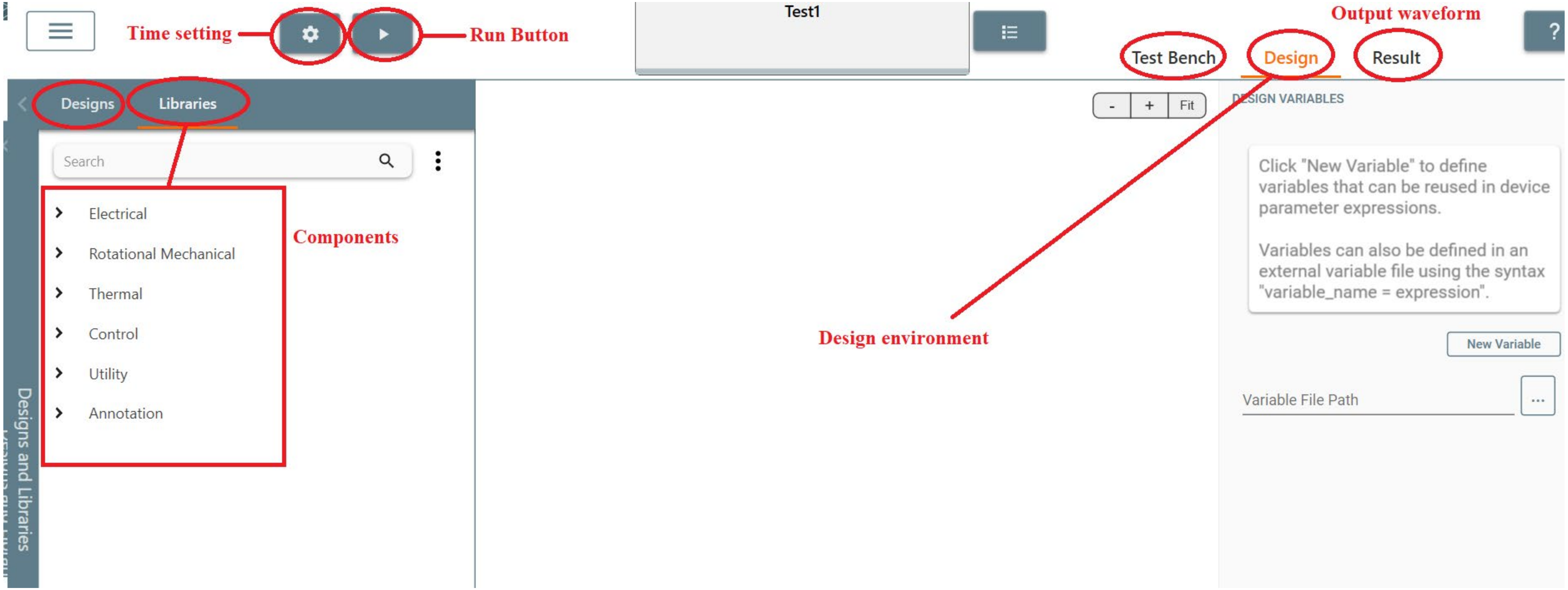
Machine Learning AI

- SckitLear
- TensorFlow
- Keras
- Pythorch

SIMBA Platform

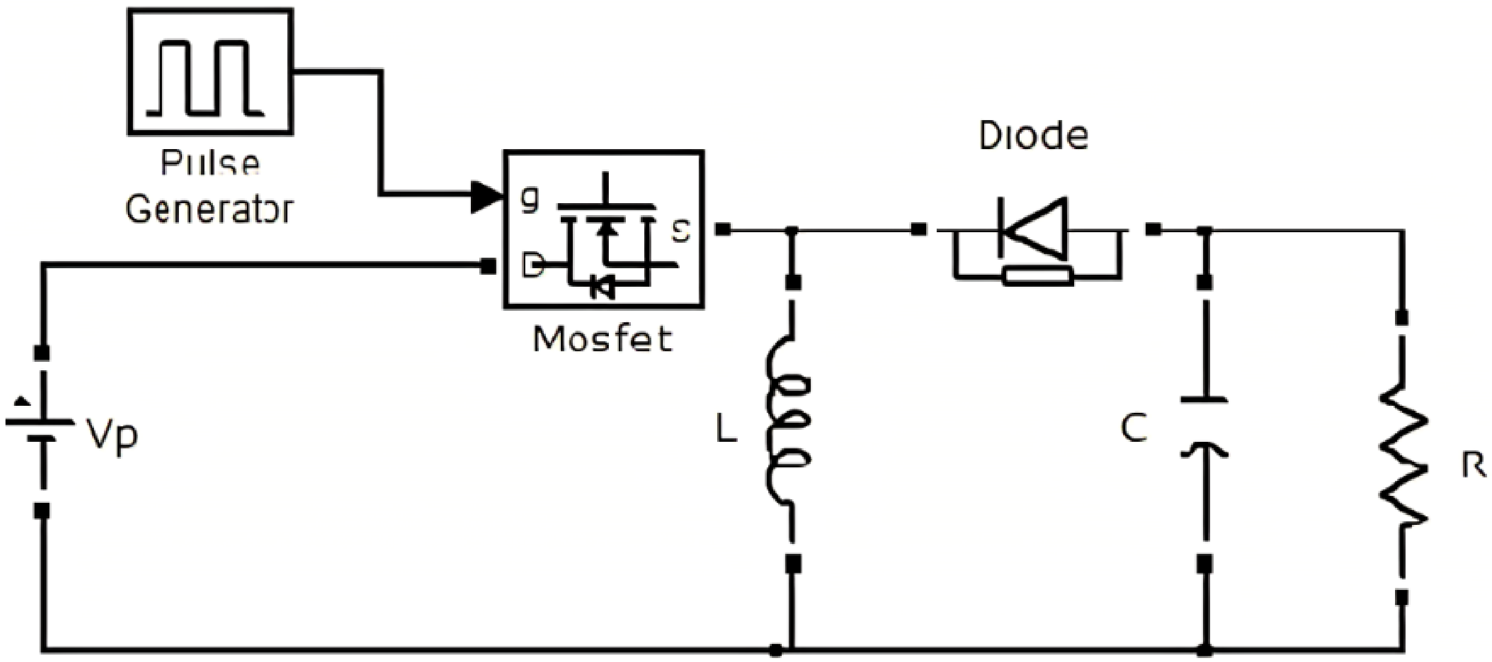
Simba is a software platform used for simulating power electronics and motor drives. It provides tools and features for modeling and analyzing electrical systems and components, allowing engineers and researchers to simulate the behavior of power electronic circuits and motor drive systems.

Simba Simulation Environment



Simba Simulation Environment

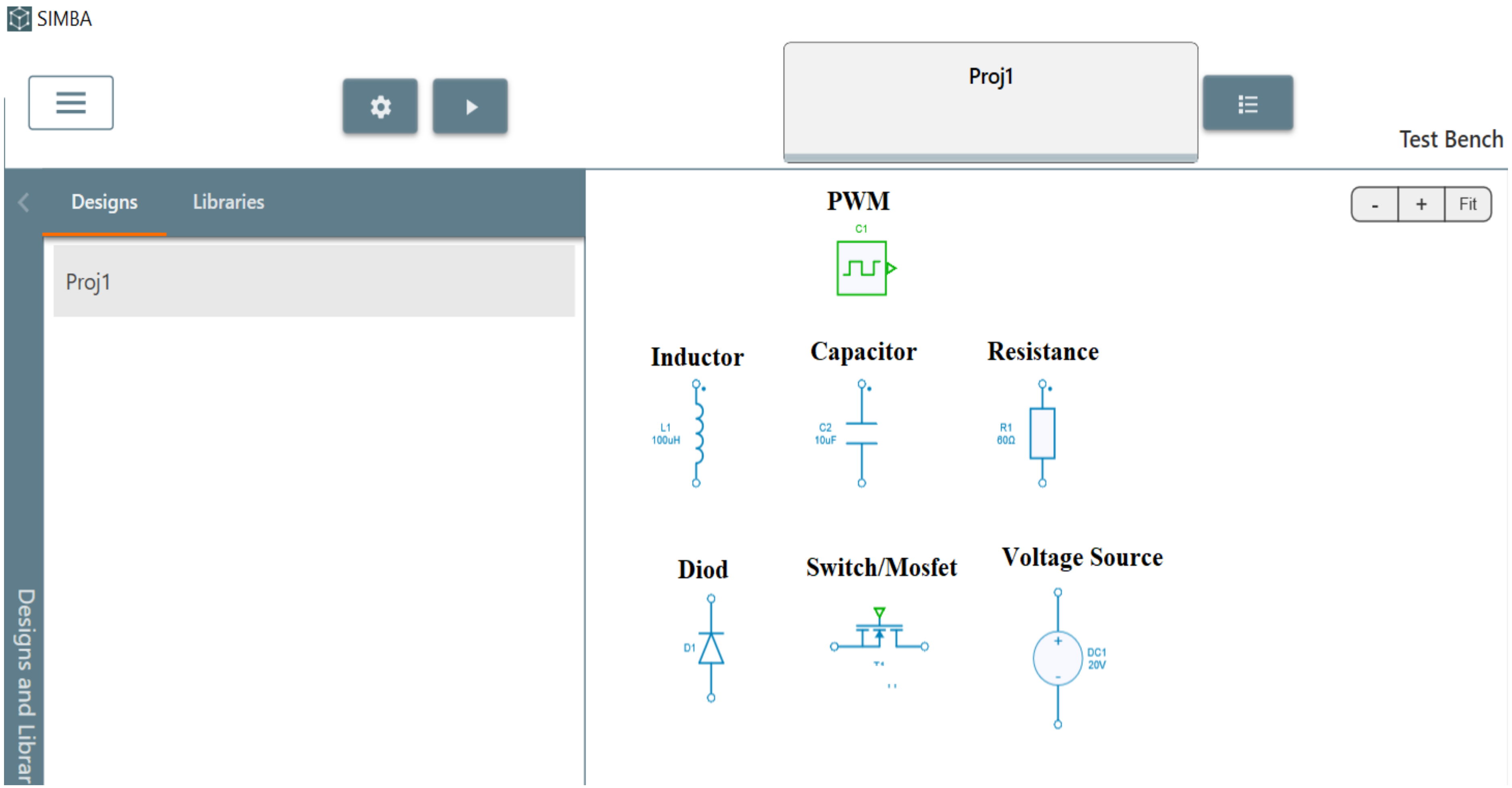
Buck-Boost Converter



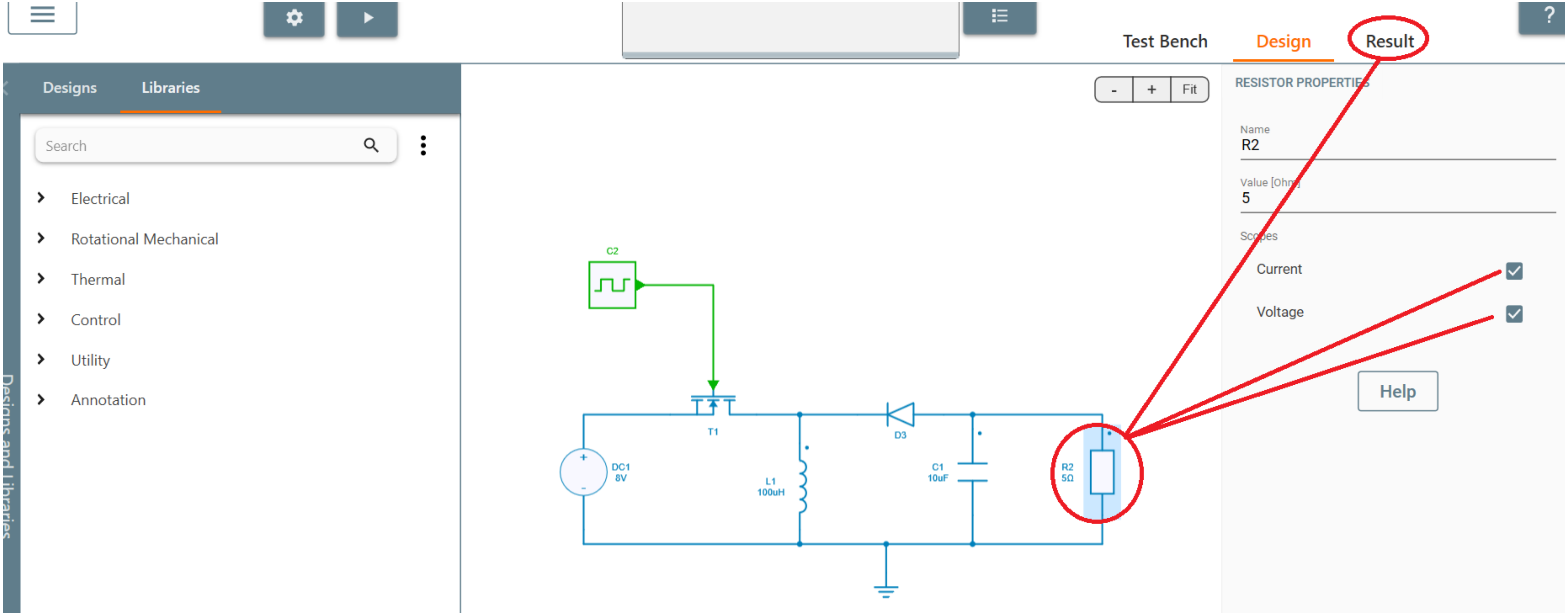
Analysis

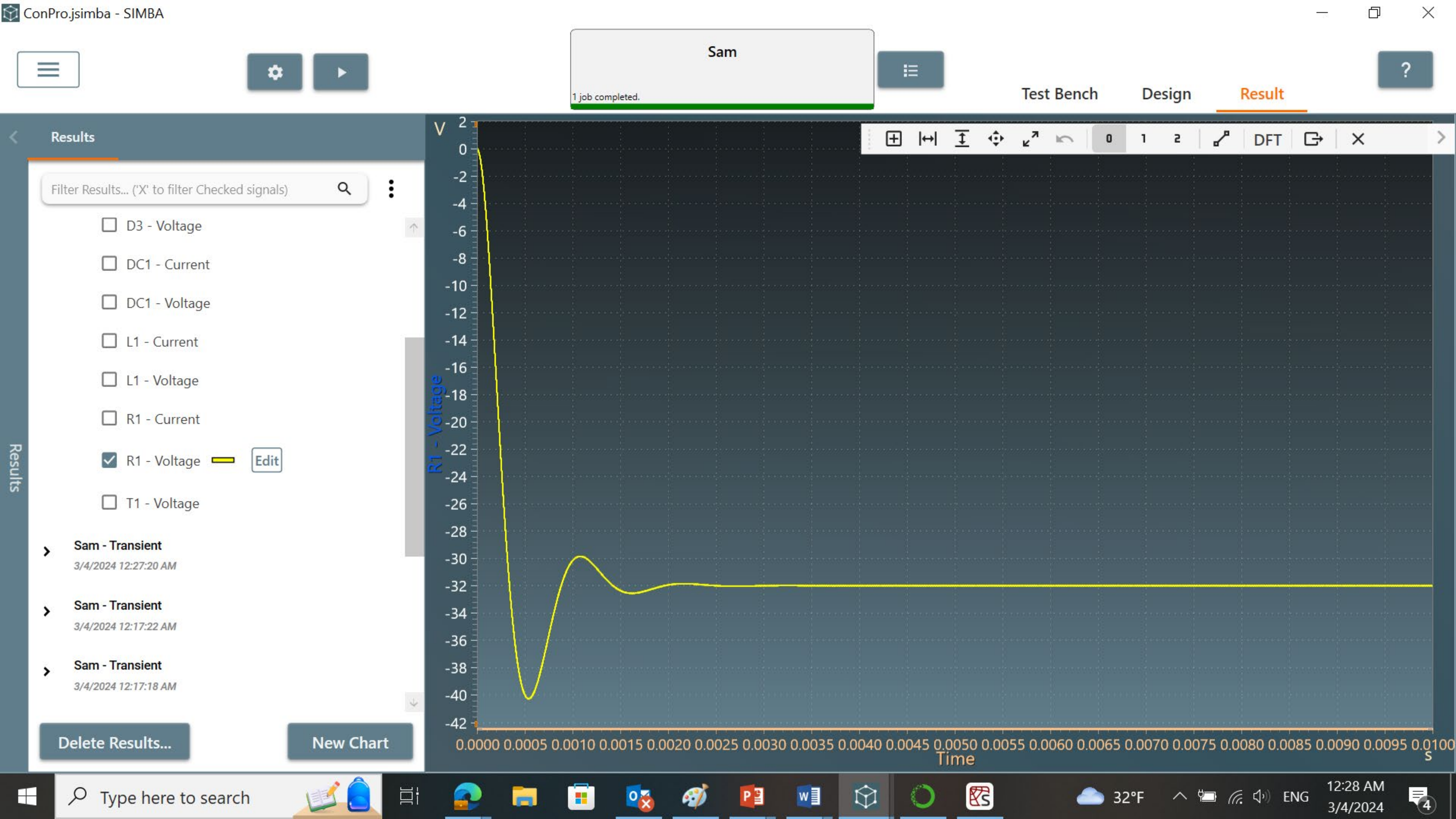
- Simba Environment
- Python programming

Required components/Elements



Design Buck-Boost Converter




ConPro.jsimba - SIMBA
Sam
1 job completed.
Test Bench
Design
Result
Results
Filter Results... ('X' to filter Checked signals)
D3 - Voltage
DC1 - Current
DC1 - Voltage
L1 - Current
L1 - Voltage
R1 - Current
R1 - Voltage
Edit
T1 - Voltage
Sam - Transient
3/4/2024 12:27:20 AM
Sam - Transient
3/4/2024 12:17:22 AM
Sam - Transient
3/4/2024 12:17:18 AM
Delete Results...
New Chart
V
R1 - Voltage
2
0
-2
-4
-6
-8
-10
-12
-14
-16
-18
-20
-22
-24
-26
-28
-30
-32
-34
-36
-38
-40
-42
0
1
2
DFT
0.0000 0.0005 0.0010 0.0015 0.0020 0.0025 0.0030 0.0035 0.0040 0.0045 0.0050 0.0055 0.0060 0.0065 0.0070 0.0075 0.0080 0.0085 0.0090 0.0095 0.0100
Time
s
Type here to search
32°F
ENG
12:28 AM
3/4/2024

Python Programming and Import Designed simulation

Required Libraries

```python
#%% Load modules
import matplotlib.pyplot as plt
from aesim.simba import JsonProjectRepository
import os
#%% Load project

#%% Get project
script_folder = os.path.realpath(os.path.dirname(__file__))
file_path = os.path.join(script_folder, "Proj3.jsimba")
project = JsonProjectRepository(file_path)
BuckBoost = project.GetDesignByName('pro')
```

Import created design

- ✓ script_folder = os.path.realpath(os.path.dirname(__file__)):
  This line finds the real path of the directory where our current Python script is located.

- ✓ file_path = os.path.join(script_folder, "ConPro.jsimba"):
  Here, we're creating the full path to our Simba project file, 'Proj3.jsimba', by joining the script's folder path with the project file's name.

- ✓ project = JsonProjectRepository(file_path):
  Lastly, this line loads our Simba project file into the Python script using a class called JsonProjectRepository. By doing this, our script can directly interact with the project, allowing for manipulation, analysis, or automation within the Simba environment."

Python Programming Explanations

Command Transient Analyze

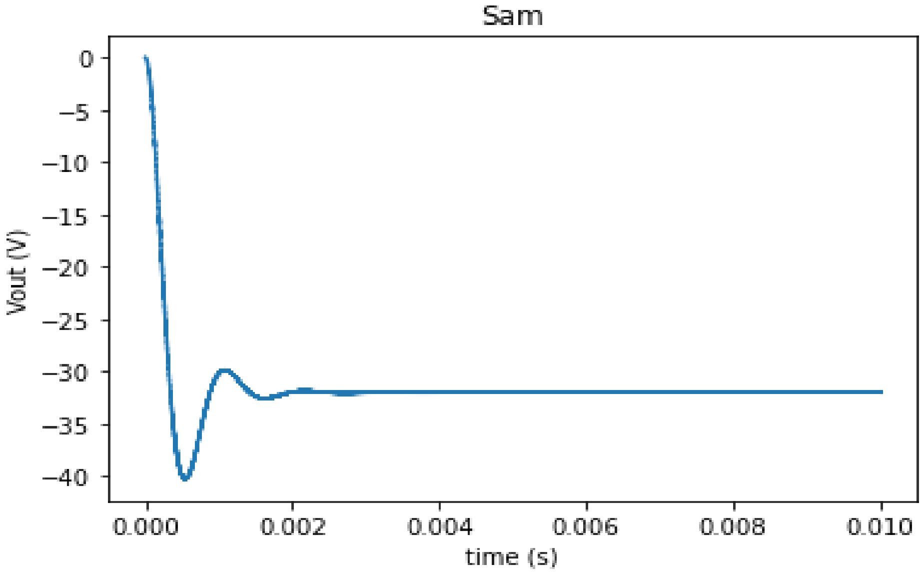
```
job = BuckBoost.TransientAnalysis.NewJob()
status = job.Run()

#%% Get results
t = job.TimePoints
Vout = job.GetSignalByName('R1 - Instantaneous Voltage').DataPoints

#%% Plot Curve
fig, ax = plt.subplots()
ax.set_title(BuckBoost.Name)
ax.set_ylabel('Vout (V)')
ax.set_xlabel('time (s)')
ax.plot(t,Vout)
plt.show()
```

Define Output

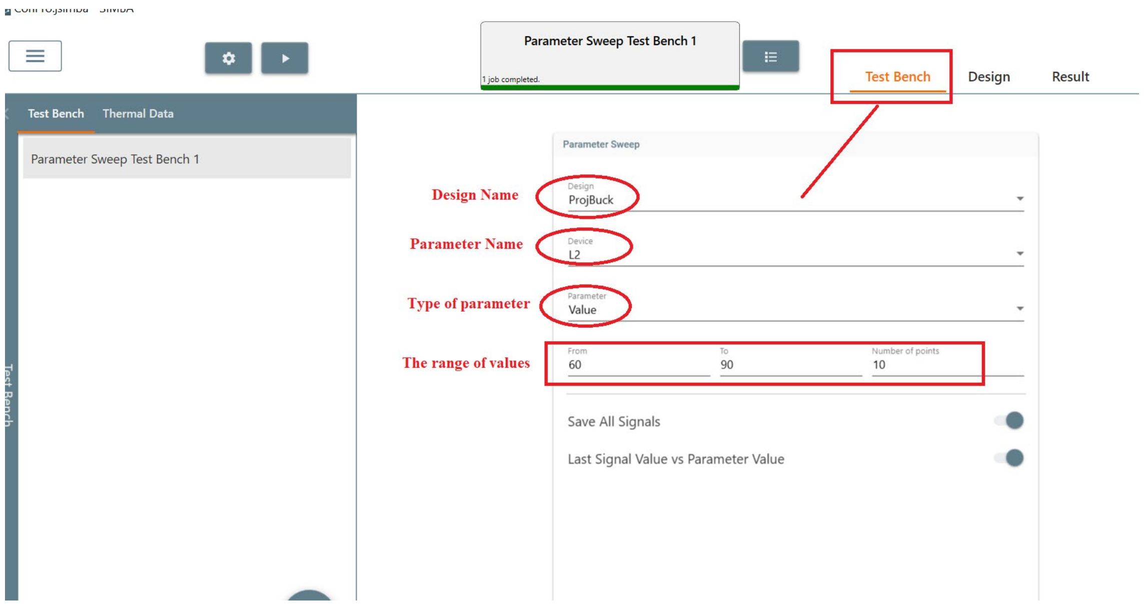
Plotting code with Matplot



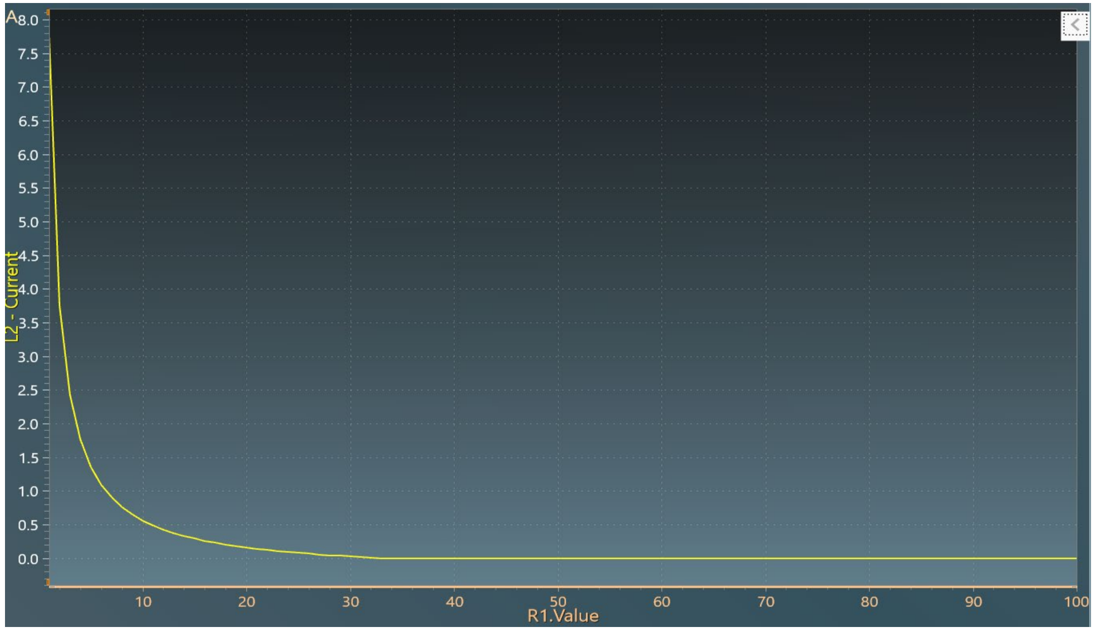
Parameter Sweep in Simba

A "Parameter Sweep" is a process in which a series of simulations are run while systematically varying the parameters of a model to analyze the effects on its behavior or performance.

Parameter Sweep in Simba environment

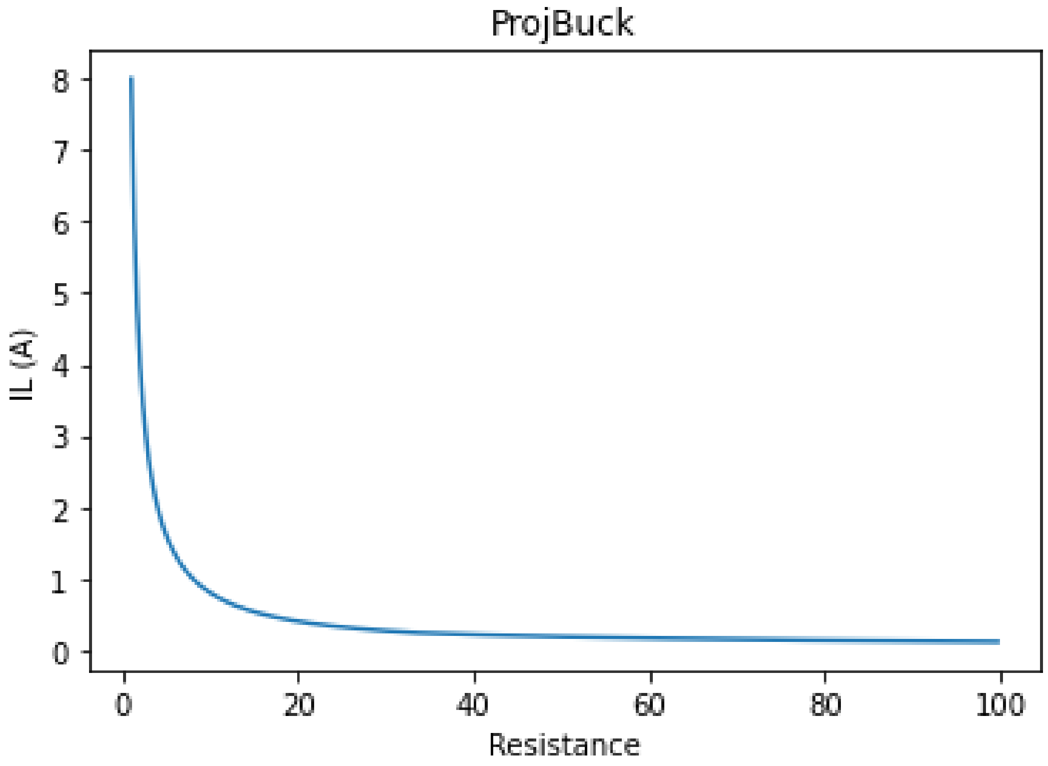


$R = 1\Omega \longrightarrow 100\Omega$



Parameter Sweep with python code

```
R = np.arange(1, 100, 10/30)
ILs = []
for RR in R:
R2 = BuckBoostConverter.Circuit.GetDeviceByName('R1')
    R2.Value=RR
    # Run calculation
    job = BuckBoostConverter.TransientAnalysis.NewJob()
    status = job.Run()
    # Retrieve results
    t = np.array(job.TimePoints)
    IL = np.array(job.GetSignalByName('L2 - Current').DataPoints)
    # Average output voltage for t > 5ms
    indices = np.where(t >= 0.005)
    IL = np.take(IL, indices)
    IL = np.average(IL)
    # Save results
    ILs.append(IL)
```



Julia libraries in Simba

```
using PyCall
using PyPlot


# Import Python libraries in Julia
aesim = pyimport("aesim.simba")
os = pyimport("os")
np = pyimport("numpy")
```