HOW2POWER TODAY
Your Power Design Newsletter

power=product^3
Power Products in 3 Images or Less

ISSUE: October 2020

Power Electronics Simulation Software Is Fast, Powerful And Easy To Use

AESIM.Tech, a Montréal-based startup developing the next generation of power electronics simulation software, is offering the first beta version of its SIMBA power electronics simulation software. SIMBA is described as a lightweight but powerful power electronics simulation environment.

In developing this product, the company's ambition is to create a platform simple enough for students and hobbyists but sufficiently fast and powerful for most complicated use cases. As an example, a five-level NPC power converter with more than 40 switches takes only about 150 ms to simulate 100 ms of real-time (Fig. 1).

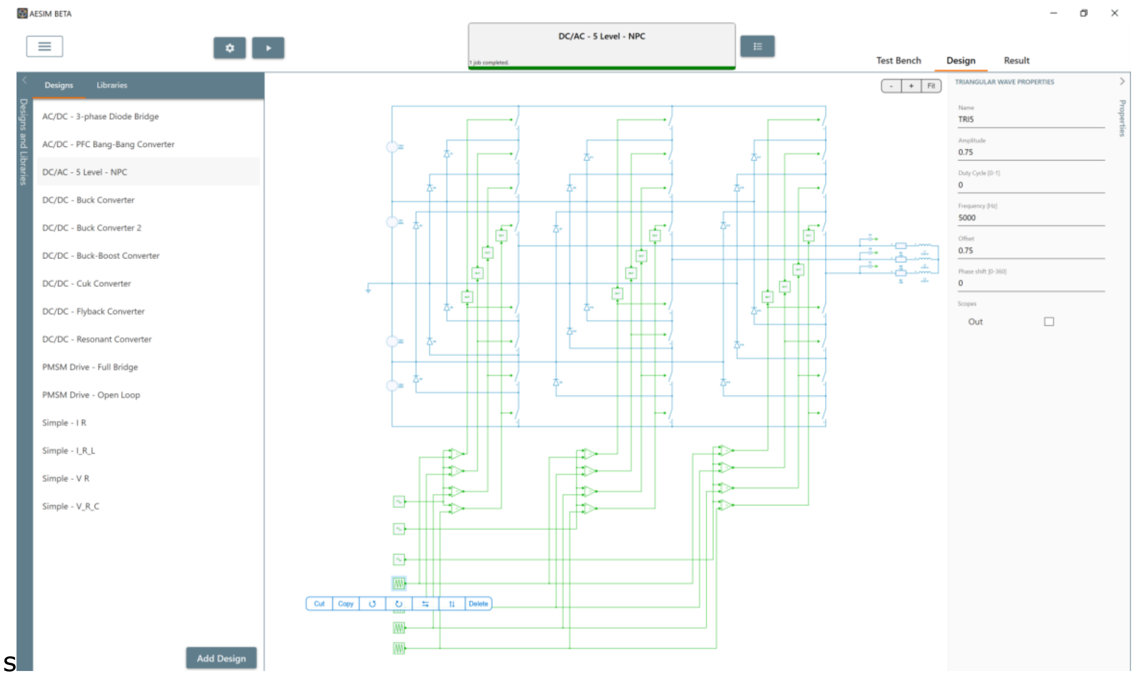
SIMBA includes a new generation of simulation engine called Predictive Time-Step solver. With some other simulators, designers can spend a long time tuning their simulation solver parameters (time step, tolerance...) to find the best possible compromise between speed and accuracy. The "Predictive Time Step" automatically finds and uses the optimal time step to simulate all time constants and events of a system without compromising the accuracy (see Fig. 2).

SIMBA can also be used as an independent Python module for advanced tasks and post-processing. The SIMBA Python Library (pysimba) contains hundreds of functions providing direct access to SIMBA such as creating a circuit, modifying parameters, running a simulation, and retrieving results (see Fig. 3).

"There are a lot of great power electronic simulation tools out there but we think this market is aging and needs fresh air. Most of the tools were developed decades ago and are partly based on outdated technologies. Building a new simulation platform from the ground up with modern technologies will provide the users a superfast solver and a better overall experience. Check it for yourself!" said Emmanuel Rutovic, founder of AESIM.Tech.

SIMBA is still a young tool and many features will be added during the beta period. As a matter of fact, a new Beta version will be released every 2 to 3 weeks and you can check the roadmap on this Github repository. After the beta period, SIMBA will be released.

The beta version is 100% free and includes the user interface and the Python module (Windows only). To join the free public beta, go to simba.io or see the website.



AC/DC - 3-phase Diode Bridge
AC/DC - PFC Bang-Bang Converter
DC/AC - 5 Level - NPC
DC/DC - Buck Converter
DC/DC - Buck Converter 2
DC/DC - Buck-Boost Converter
DC/DC - Cuk Converter
DC/DC - Flyback Converter
DC/DC - Resonant Converter
PMSM Drive - Full Bridge
PMSM Drive - Open Loop
Simple - I R
Simple - I_R_L
Simple - V R
Simple - V_R_C

TRIANGULAR WAVE PROPERTIES

Name: TRI5
Amplitude: 0.75
Duty Cycle [0-1]: 0
Frequency [Hz]: 5000
Offset: 0.75
Phase shift [0-360]: 0
Scopes
Out ☐

S

Fig. 1. Even large converters such as this five-level NPC power converter are simulated quasi-instantaneously.

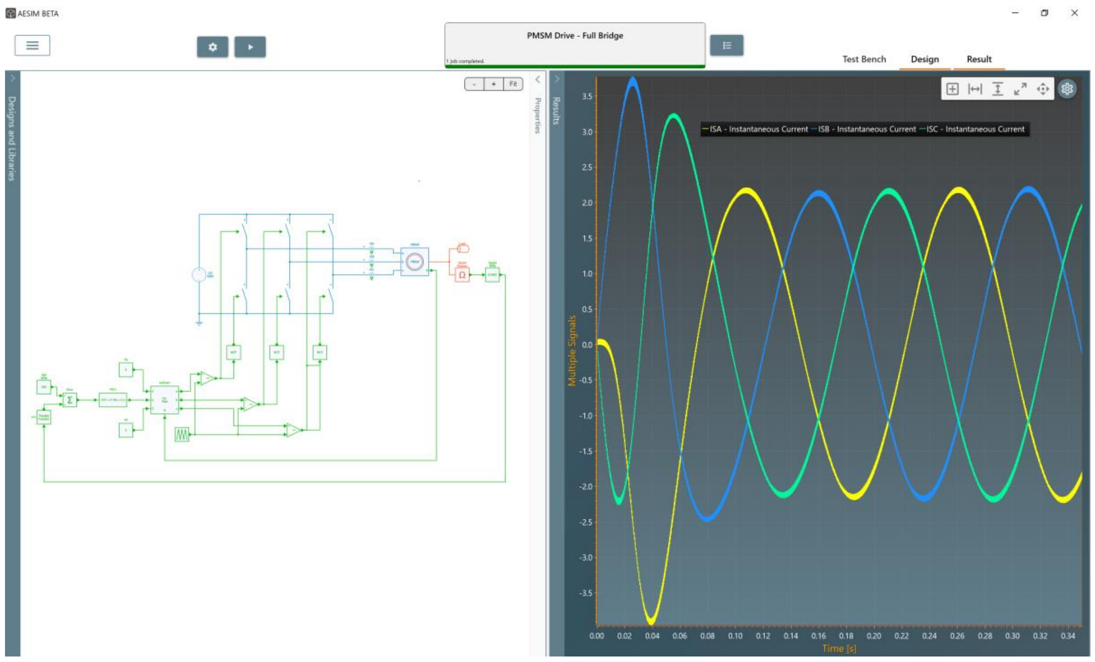
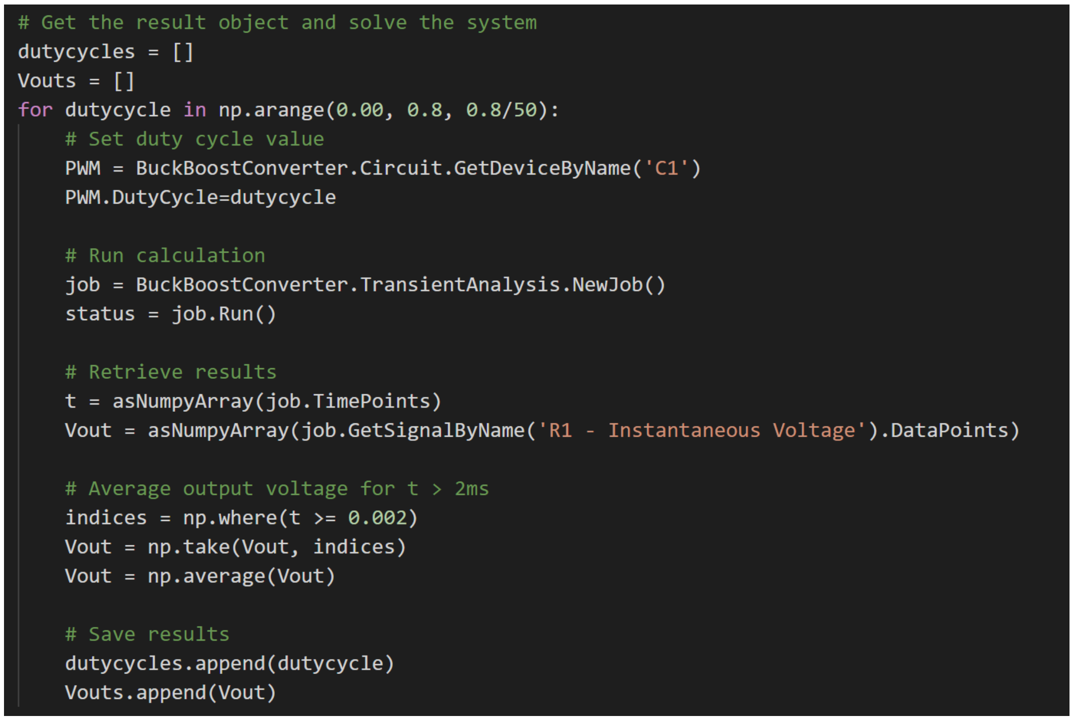


Fig. 2. The SIMBA Predictive Time-Step solver simulates complex systems without compromising the accuracy.



```python
# Get the result object and solve the system
dutycycles = []
Vouts = []
for dutycycle in np.arange(0.00, 0.8, 0.8/50):
    # Set duty cycle value
    PWM = BuckBoostConverter.Circuit.GetDeviceByName('C1')
    PWM.DutyCycle=dutycycle

    # Run calculation
    job = BuckBoostConverter.TransientAnalysis.NewJob()
    status = job.Run()

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

    # Average output voltage for t > 2ms
    indices = np.where(t >= 0.002)
    Vout = np.take(Vout, indices)
    Vout = np.average(Vout)

    # Save results
    dutycycles.append(dutycycle)
    Vouts.append(Vout)
```

Fig. 3. SIMBA Python Library is available for advanced tasks and post-processing.