Developing DC-DC Converter Control with Simulink

Vasco Lenzi
Senior Application Engineer
Key Takeaways

- Graphical programming across our solutions is **intuitive** and **powerful**

- State-of-the-art technologies facilitate the **design** and **verification** of complex systems developed by **multidisciplinary teams**

- Find design errors **early** and cut down development **cost** while increasing delivered **quality**.
Our Project Today

DC/DC LED Developer's Kit

LED Head Lamp

Fig 1: TMDSDCDCLEDKIT

Fig 4: DC/DC LED Lighting Board Block diagram with F28035
ZKW Lichtsysteme GmbH

Rapid Control Prototyping with Simulink Real-Time and Speedgoat:

- Design control algorithms for an innovative LED headlamp projection technology changing its illumination dynamically
- Seamless integration into MathWorks Tools
- Faster time to market

"Model-based design itself has proven to be very flexible, powerful and efficient for our purposes. Using the Mobile real-time target machine from Speedgoat, we were able to completely redesign a functional prototype based on a model and verify it during an afternoon session."

- Matthaeus Artmann, Manager Electronics Engineering Pre- and Module Development, ZKW Lichtsysteme GmbH

Official Speedgoat User Story

https://www.youtube.com/watch?v=wAk9e5w0dSq
Developing DC-DC Converter Control with Simulink

- Model the converter and calculate the most efficient operating region
- Determine power losses and the thermal behaviour of the converter
- Design control algorithm based on time/frequency domain specification
- Design supervisory logic and implement unit testing
- Implement power electronic controls on an embedded platform
Developing DC-DC Converter Control with Simulink

- Model the converter and calculate the most efficient operating region
- Determine power losses and the thermal behaviour of the converter
- Design control algorithm based on time/frequency domain specification
- Design supervisory logic and implement unit testing
- Implement power electronic controls on an embedded platform
Modeling the converter

Power Supply

Load (LEDs)
Modeling Approaches

First Principles

- Physical Networks
- Programming
- Block Diagram
- Modeling Language

Statistical Methods
- System Identification
- Parameter Tuning
- Measured Model

Data-Driven

- Neural Networks
- Physical Networks
- Model

Modeling Approaches

- Symbolic Methods
- Statistical Methods
- Model

- Measured Model
- Data Driven Model

- Measured
- Model
Simscape Products

- MATLAB and Simulink provide foundation for technical computing and algorithm development

- Simscape platform
  - Simulation engine and custom diagnostics
  - Foundation libraries in many domains
  - Language for defining custom blocks

- Simscape add-on libraries
What’s new in Simscape Electrical

- Parametrized Stepper motor block
- Parametrized Battery block
- Faults
  - Dynamic Load from DC or AC supply
  - Constant Power Load
  - Delta-Connected Load, Wye-Connected Load
Developing DC-DC Converter Control with Simulink

- Model the converter and calculate the most efficient operating region
- **Determine power losses and the thermal behaviour of the converter**
- Design control algorithm based on time/frequency domain specification
- Design supervisory logic and implement unit testing
- Implement power electronic controls on an embedded platform
Recap: What have we seen?

- Create heat maps
- Reuse it in extremely fast thermal-focused model for cooling sizing and control

```python
# ee_getpowerlossSummary
```
Convert SPICE models into Simscape components

Netlist

```
.FUNC Idiode (Usd,Tj,Iss) { exp(min(1,log(Iss/I0))) / (1 + exp(min(1,log(Iss/I0)))) }
.FUNC Idiod (Usd,Tj) { a*Idiode(Tj) }
.FUNC Pr (Vss0,Vssp) { Vss0*Vss0/Rm+Vss0/(2*Rm) }
.FUNC J1 (x,da,d,T,da,s,x) { a*(s*(exp(min(1,log(x/I0)))) }
.FUNC QCds (x) { Cds3*min(x,x1)+Cds0*x }
.FUNC QCdp (x) { Cdp3*min(x,x1)+Cdp0*x }
```

subcircuit2ssc

```
components (ExternalAccess=observe)
X1 = test.s5_100_f_var(a=act,rsp=r
    rs=rs,rp=rd,dc=dc,rm=rm);
RG = elec.passive.instrumented_res
LG = foundation.electrical.element
    (i_L.priority=priority.none);
RSA = elec.passive.instrumented_res
LS = foundation.electrical.element
    (i_L.priority=priority.none);
```
Simscape Electrical
SPICE Models

- Manufacturer-specific MOSFETs

- Additional transistor capacitance models in SPICE NMOS and SPICE PMOS blocks
  - Meyer gate or charge conservation

- Conversion Assistant supports table SPICE function

- Validate MOSFET conversions by generating characteristics and comparing with LTspice
Developing DC-DC Converter Control with Simulink

- Model the converter and calculate the most efficient operating region
- Determine power losses and the thermal behaviour of the converter
- **Design control algorithm based on time/frequency domain specification**
- Design supervisory logic and implement unit testing
- Implement power electronic controls on an embedded platform
Voltage Controller Design

- **Requirement**
  - Implement voltage controller and tune it

- **Approach**
  - Create transfer function equivalent model
  - Tune controllers based on requirements
What have we seen?

- Tuning based on System Identification
- Works with any topology
Average switch option for converters and choppers

- Faster simulation by using modulation signal or undersampling as gate signal
  - Bidirectional DC-DC Converter
  - Boost Converter
  - Buck Converter
  - Buck-Boost Converter
  - Converter (Three-Phase)
  - Four-Quadrant Chopper
  - One-Quadrant Chopper
  - Three-Level Converter (Three-Phase)
  - Two-Quadrant Chopper
Developing DC-DC Converter Control with Simulink

- Model the converter and calculate the most efficient operating region
- Determine power losses and the thermal behaviour of the converter
- Design control algorithm based on time/frequency domain specification
- Design supervisory logic and implement unit testing
- Implement power electronic controls on an embedded platform
Recap: What have we seen?

Stateflow
Recap: What have we seen?

Simulink Test

- Create test harnesses and test cases
- Group into suites and test files
- Execute individual or batch
- View result summary
- Analyze results
- Archive, export, report
Developing DC-DC Converter Control with Simulink

- Model the converter and calculate the most efficient operating region
- Determine power losses and the thermal behaviour of the converter
- Design control algorithm based on time/frequency domain specification
- Design supervisory logic and implement unit testing
- Implement power electronic controls on an embedded platform
Automatic Code Generation

- Requirement
  - Generate target-aware, efficient C-code

- Approach
  - Model elaboration for C-code generation
  - Create a first configuration set with Embedded Coder Quick Start
  - Build the code, automatically generate reports
Deploy to Any Processor with Best-in-Class Performance

Models in MATLAB and Simulink can be deployed on embedded devices, edge devices, enterprise systems, the cloud, or the desktop.
MATLAB Connects to Your Hardware Devices

Instrument Control
Oscilloscopes, Signal Generators, Lab Instruments

Data Acquisition
Plug-in data acquisition devices, I/O boards and sound cards

Image and Video Acquisition
Industrial and scientific cameras

Digital Networks
OPC, CAN, J1939, and XCP protocol devices

Hardware support packages
Built-in and downloadable support for a wide range of devices and development boards
Project and File Management

Simulink Project Upgrade

- **Upgrade Project Report**
  - 75% Passed
  - Need attention: 2

- **Check Reports**:
  - Example: $\texttt{myModelFile.m}$
  - Passed checks: 125
  - Need attention checks: 2

- **Warnings**:
  - `\texttt{vivcat} is not recommended. Use `\texttt{vlsize}` instead.`
  - Incompatible code found in the following lines:
    - Line 1: `\texttt{vivcat}`, `\texttt{ config, vary};`
In-the-Loop Verification Methodologies
Software-in-the-Loop

Is the generated code functionally equivalent to the model?
In-the-Loop Verification Methodologies
Processor or FPGA-in-the-Loop

Non-Real-Time functional verification of the algorithm component, C or HDL

Is the generated code functionally equivalent to the model?
In-the-loop verification methodologies

Hardware-in-the-Loop: “HIL”

Does algorithm perform well on actual device with true latencies?

Production embedded target: Structured Text, VHDL, C/C++

Real-Time Machine eg “Speedgoat”
About Speedgoat

- A MathWorks associate company, incorporated in 2006 by former MathWorks employees. Headquarters in Switzerland, with subsidiaries in the USA and Germany

- Provider of real-time target computers, expressly designed for use with Simulink

- Real-time core team of around 200 people within MathWorks and Speedgoat. Closely working with the entire MathWorks organization employing around 5,000 people worldwide
Hardware-in-the-Loop Simulation of SEPIC Converter

Target Computer with FPGA I/O Module

- Model sample time = 500 nanoseconds (2 MHz)

Xilinx Kintex®-7 FPGA
- Floating-point support

DAC - Analog output
- ~ 500 ns settling time
- 16-bit
- ±10V voltage range

PWM Capture - 200 MHz
- 5 nanosecond resolution

Output voltage

PWM signal

100 kHz

Development Computer

Device Under Test

Linearized SEPIC used to deploy to FPGA I/O module via Simcape HDL Workflow

Vout1ADC

Vout1REF

Software

Simulation of SEPIC Converter

10 MHz

Vout1

Device Under Test

Linearized SEPIC used to deploy to FPGA I/O module via Simcape HDL Workflow

Vout1ADC

Vout1REF

Software

Simulation of SEPIC Converter

10 MHz

Vout1

Device Under Test

Linearized SEPIC used to deploy to FPGA I/O module via Simcape HDL Workflow

Vout1ADC

Vout1REF

Software

Simulation of SEPIC Converter

10 MHz

Vout1

Device Under Test

Linearized SEPIC used to deploy to FPGA I/O module via Simcape HDL Workflow

Vout1ADC

Vout1REF

Software

Simulation of SEPIC Converter

10 MHz

Vout1

Device Under Test

Linearized SEPIC used to deploy to FPGA I/O module via Simcape HDL Workflow

Vout1ADC

Vout1REF

Software

Simulation of SEPIC Converter

10 MHz

Vout1

Device Under Test

Linearized SEPIC used to deploy to FPGA I/O module via Simcape HDL Workflow

Vout1ADC

Vout1REF

Software

Simulation of SEPIC Converter

10 MHz

Vout1

Device Under Test

Linearized SEPIC used to deploy to FPGA I/O module via Simcape HDL Workflow

Vout1ADC

Vout1REF

Software

Simulation of SEPIC Converter

10 MHz

Vout1

Device Under Test

Linearized SEPIC used to deploy to FPGA I/O module via Simcape HDL Workflow

Vout1ADC

Vout1REF

Software

Simulation of SEPIC Converter

10 MHz

Vout1

Device Under Test

Linearized SEPIC used to deploy to FPGA I/O module via Simcape HDL Workflow

Vout1ADC

Vout1REF

Software

Simulation of SEPIC Converter

10 MHz

Vout1

Device Under Test

Linearized SEPIC used to deploy to FPGA I/O module via Simcape HDL Workflow

Vout1ADC

Vout1REF

Software

Simulation of SEPIC Converter

10 MHz

Vout1

Device Under Test

Linearized SEPIC used to deploy to FPGA I/O module via Simcape HDL Workflow

Vout1ADC

Vout1REF

Software

Simulation of SEPIC Converter

10 MHz

Vout1

Device Under Test

Linearized SEPIC used to deploy to FPGA I/O module via Simcape HDL Workflow

Vout1ADC

Vout1REF

Software

Simulation of SEPIC Converter

10 MHz

Vout1

Device Under Test

Linearized SEPIC used to deploy to FPGA I/O module via Simcape HDL Workflow

Vout1ADC

Vout1REF

Software

Simulation of SEPIC Converter

10 MHz

Vout1

Device Under Test

Linearized SEPIC used to deploy to FPGA I/O module via Simcape HDL Workflow

Vout1ADC

Vout1REF

Software

Simulation of SEPIC Converter

10 MHz

Vout1

Device Under Test

Linearized SEPIC used to deploy to FPGA I/O module via Simcape HDL Workflow

Vout1ADC

Vout1REF

Software

Simulation of SEPIC Converter

10 MHz

Vout1

Device Under Test

Linearized SEPIC used to deploy to FPGA I/O module via Simcape HDL Workflow

Vout1ADC

Vout1REF

Software

Simulation of SEPIC Converter

10 MHz

Vout1

Device Under Test

Linearized SEPIC used to deploy to FPGA I/O module via Simcape HDL Workflow

Vout1ADC

Vout1REF

Software

Simulation of SEPIC Converter

10 MHz

Vout1
Hardware-in-the-loop simulation of SEPIC converter
“With Model-Based Design, our developer productivity is easily increased tenfold. Simulation and code generation enable us to turn changes around quickly and eliminate human errors in coding. Our algorithms typically work the first time, so we no longer waste a big part of our development cycle debugging code.”

Dr. Robert Turner, ABB  [link]
Key Takeaways

- Graphical programming across our solutions is **intuitive** and **powerful**

- State-of-the-art technologies facilitate the **design** and **verification** of complex systems developed by **multidisciplinary teams**

- Find design errors **early** and cut down development **cost** while increasing delivered **quality**.
Visit the Power Electronics Control Community on MATLAB Central to find Models, Answers, and How-to Videos

https://www.mathworks.com/matlabcentral/topics/power-electronics-control.html
Q&A and Conclusion

- Questions: vlenzi@mathworks.com