

Computer-aided Calibration of IGBT SPICE Model with optiSLang™

WOST 2018

2018-06-21



A. Biswas, M. Cotorogea, P. Türkes, F.J. Niedernostheide
Infineon Technologies AG

Special thanks: Rene Kallmeyer (Dynardo)



Agenda

1

Introduction & Motivation

2

SIMatrix integration

3

Calibration Flow

4

Example 1 : Single transient curve

5

Example 2 : Two static curves (Transfer and Output)

6

Conclusion/Future Outlook

Agenda

1

Introduction & Motivation

2

SIMetrix integration

3

Calibration Flow

4

Example 1 : Single transient curve

5

Example 2 : Two static curves (Transfer and Output)

6

Conclusion/Future Outlook

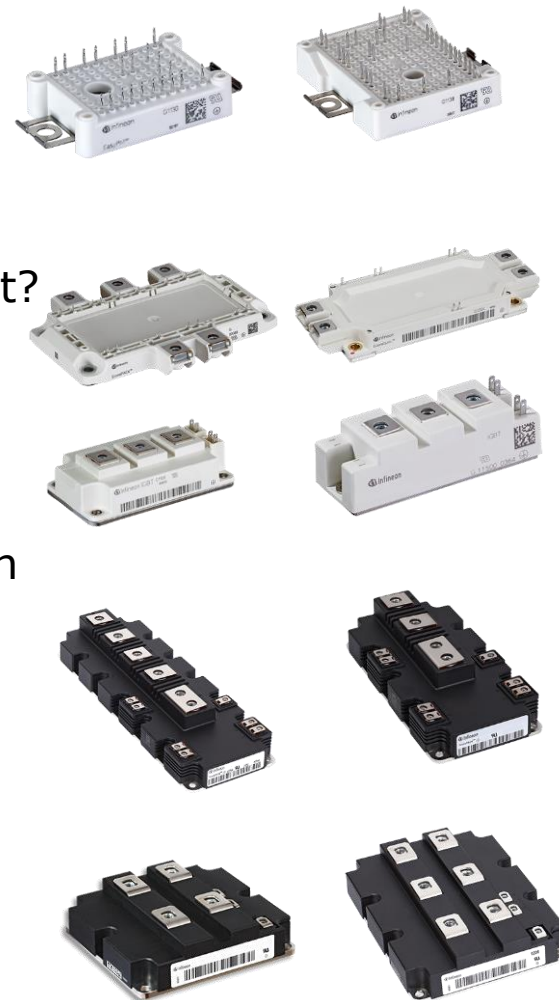
Introduction

- › IGBTs and power diodes are bipolar devices
 - › Wide ranging applications from electric cars, solar/wind energy, traction, transmission line...

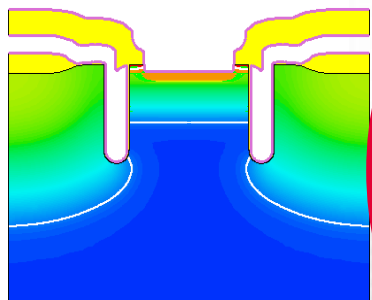
- › Why virtual prototyping in IGBT-module development?
 - › Reduce development costs and time by reducing learning cycles

- › Target: accurately predict switching behavior through circuit simulations

- › Model requirements:
 - › physics based models for IGBTs and diodes
 - › knowledge of parasitic elements and couplings
 - › Fast model implementation and simulation



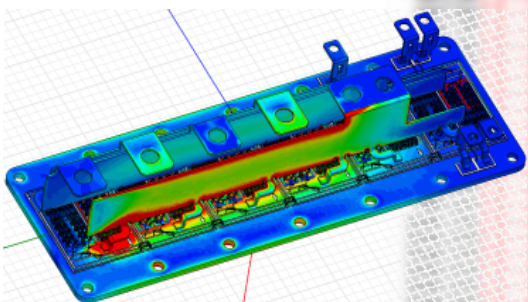
Circuit simulation



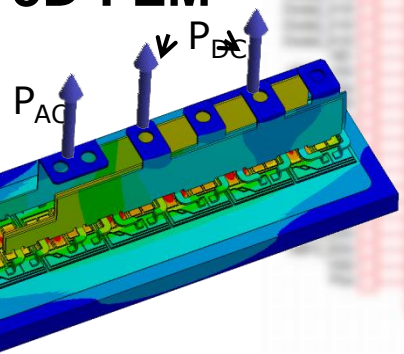
2D-TCAD

Manual model generation and calibration

Physics-based device compact model



3D-FEM



Automated model generation

Minimum loss of precision!

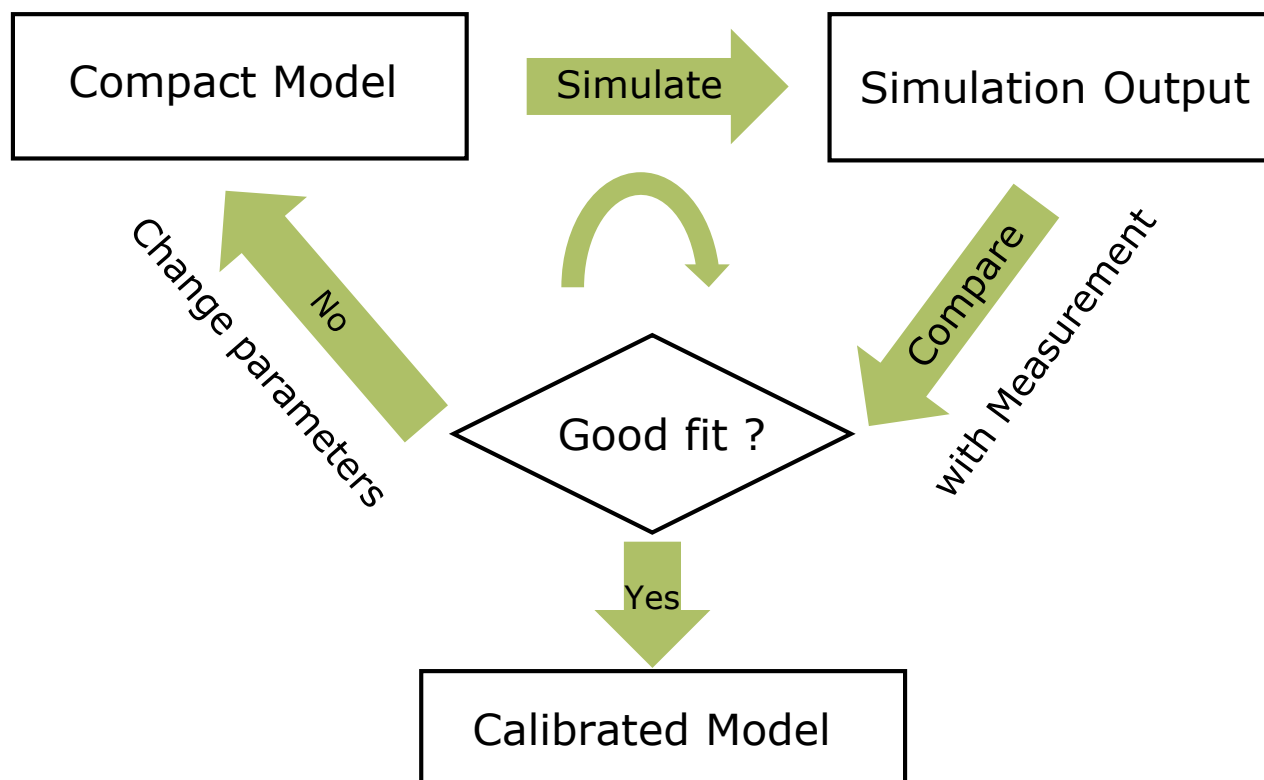
Electrical parasitics compact model

Thermal network as compact model

Circuit simulation

Motivation

- › Currently compact model calibration is a manual and time consuming process



- › To define a standardized calibration flow to be used for all diodes and IGBTs in the future

Agenda

1

Introduction & Motivation

2

SIMetrix integration

3

Calibration Flow

4

Example 1 : Single transient curve

5

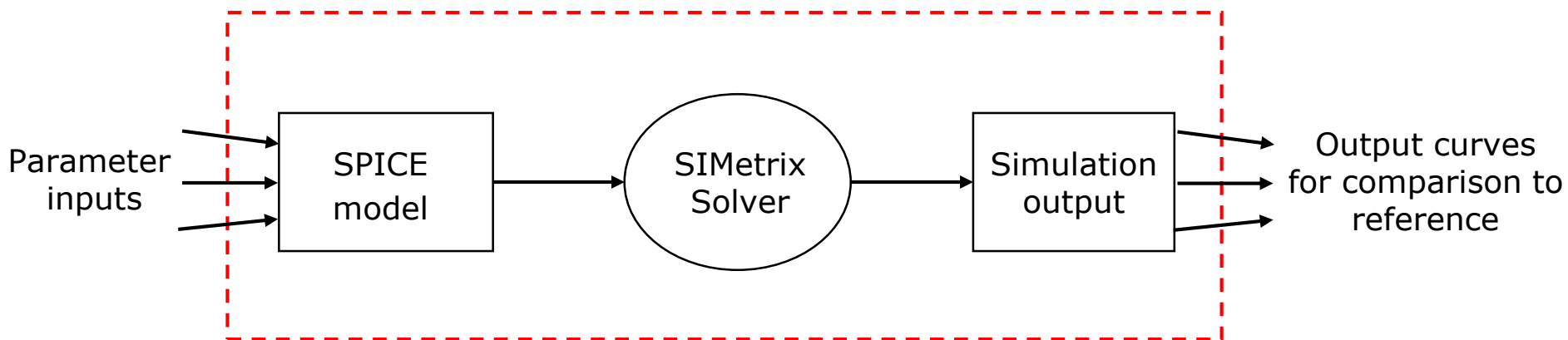
Example 2 : Two static curves (Transfer and Output)

6

Conclusion/Future Outlook

SIMetrix Integration

- › SIMetrix^[1] : Mixed mode circuit simulator supporting SPICE/Verilog-A/VHDL

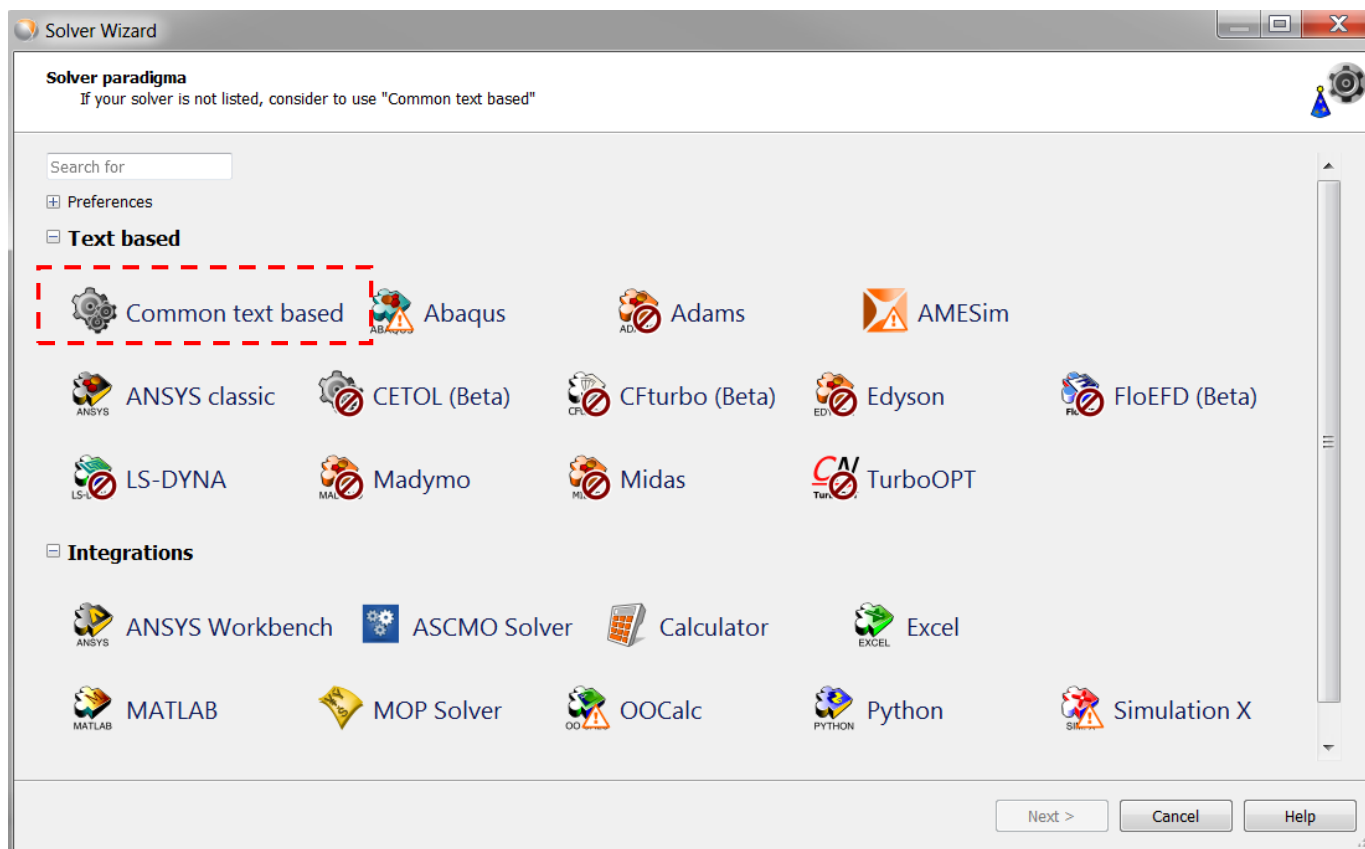


User inputs needed to set up the calibration flow:

- › Definition of parameter variation range
- › Objective and criterion definition for optiSLang
 - › Which areas of the curve are more important than rest?
 - › Weightage factors
 - › For multiple curves : normalization if needed

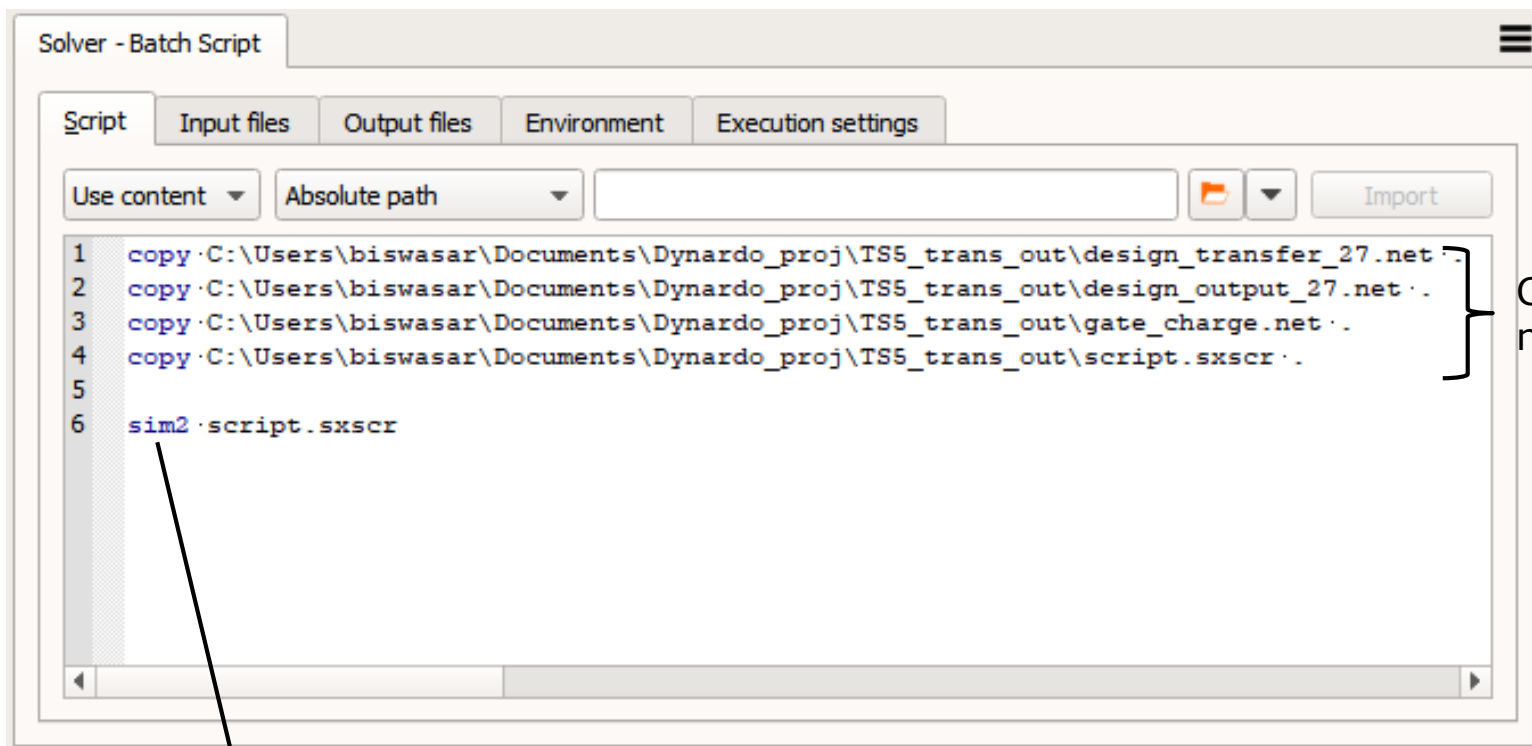
[1] www.simetrix.co.uk

Choice of solver



- > Support for many tool (solver) integration. Example: Matlab, Python, Ansys etc.
- > Any solver with some scripting support can be used with "Common text based"
 - > We use the SIMetrix scripting interface

Solver definition



Copy of required netlist and scripts

Command line call for SIMetrix:

- › Runs the required simulations
- › Saves the results in a format recognized by optiSLang

Definition of reference and simulation outputs

The screenshot shows the ETK (Error Tool Kit) interface. The main window displays a table with two columns: 'Reference' and 'IG_DUT2#'. The 'Reference' column contains numerical values in scientific notation, and the 'IG_DUT2#' column contains corresponding numerical values. The table is highlighted in green. On the right side, the 'Responses' panel shows three defined responses: 'error_norm' with a value of 134.154, 'signal_ref' with a range of [1:640], and 'signal_sim' with a range of [1:664]. Arrows point from these response names to the corresponding columns in the table. Below the table, there are options for 'token wise' and 'column wise' selection, and a 'Show advanced options' button. At the bottom, there are fields for 'Variable Name' (set to 'Reference'), 'Instant visualization', and 'Use as response', along with 'OK', 'Cancel', and 'Apply' buttons.

Reference	IG_DUT2#
1	1.2040731885650105e-005
2	0.033979837657192272
3	0.057495999115626931
4	0.063841160189119089
5	0.059931987884511553
6	0.00746105072582238
7	-0.048332602989947675
8	-0.093053979670433329
9	-0.14042411029115062
10	-0.15510924708264751
11	-0.13573206324144688
12	-0.084039127890244023
13	-0.033619690463526156
14	0.004877342244075997
15	0.023490885184528965
16	0.023552435560036614
17	0.012124792078317279
18	0.0023513220172179257
19	-0.0039615966706550562
20	-0.0039352765732789102
21	-0.00082764058468734057
22	0.00050859619628082725
23	0.00024224368925790737
24	-0.00018300092560054261
25	-0.00019548672014715478
26	
27	

User defined response (error_norm)
 $= \sum (\text{signal_ref} - \text{signal_sim})^2$

Solver output (signal_sim)

Reference data (signal_ref)

› Goal :
 to minimize error_norm

SIMatrix integration

The screenshot displays the **simetrix_int - optiSLang** software interface. The main workspace shows a simulation flow diagram within a green container labeled **SIMatrix Integration**. The flow consists of three sequential steps: **TRENCHSTOPP_H5_650V_upd.lib**, **Solver**, and **Output files**. Red dashed boxes highlight the **Input SPICE Model** (pointing to the first step), **Simulation Outputs** (pointing to the last step), and the overall **SIMatrix Integration** container. To the right, a **Wizards** panel lists several tools: **Solver wizard**, **Sensitivity wizard**, **Optimization wizard**, **Robustness wizard**, and **Reevaluation wizard**. A bracket indicates these wizards act on the primary scenery. Below the main workspace is a **Message log** table with columns for Date, Time, Log level, Actor, and Message.

Date	Time	Log level	Actor	Message
1 201...	16:18:...	INFO		Saving project "simetrix_int"
2 201...	16:17:...	INFO		Opened project "simetrix_int" in "C:\Users\biswasar\Document...
3 201...	16:17:...	INFO		Node locked
4 201...	16:16:...	INFO	...	working directory of "simetrix_int" set to "C:\Users\biswasa...
5 201...	16:16:...	INFO		optiSLang - Version 7.0.0 (47328)

At the bottom of the interface, there are tabs for **Message log** and **Python**. On the right side, there are panels for **Modules** and **Templates**. The **Templates** panel shows a tree structure with **Templates** expanded, containing **ARSM (1)** and **TrenchstopS5_trans_out**.

Wizards to act on the primary scenery

Input model (sub-circuit SPICE model)

List of parameters to be optimized

The screenshot shows a 'Parameter' dialog box with the following content:

- Parameter:** tynardo/ANSYS optiSLang/6.2.0/projects/simatrix_integ/TRENCHSTOPP_H5_650V_upd.lib
- Scalar values:**

```

145 .PARAM wb = {wb0 -- t_trench}
146 * .doping.levels
147 .PARAM Nem = 1e15
148 .PARAM wem = 0.001
149 .PARAM Noffs = 0.8e15
150 .PARAM Loffs = 2.95e-4
151 .PARAM a_norm = 1e15
152 .PARAM b0 = {1e-4/sqrt(0.5)}
153 .PARAM b1 = {1e-4/sqrt(2)}
154 .PARAM b2 = {1e-4/sqrt(5)}
155 .PARAM b3 = {1e-4/sqrt(5)}
156 .PARAM Npk0 = 4.1e15
157 .PARAM wpk0 = {wb-4e-4}
158 .PARAM Npk1 = 1.022e15
159 .PARAM wpk1 = {wb-12.5e-4}
160 .PARAM Npk2 = 0.44e15
161 .PARAM wpk2 = {wb-18.5e-4}
162 .PARAM Npk3 = 0.24e15
163 .PARAM wpk3 = {wb-24e-4}
164 .PARAM Nak = 8.500000000e+16
165 .PARAM Nb = 3.55e13
166 .PARAM Taub = 1e-6
167 .PARAM ETAUB = 0.3
168 * .calculated.parameters
169 .PARAM l_stripe = {sqrt(A)}
170 .PARAM nzell = {l_stripe/pitch}
171 .PARAM wfact = 1
172 .PARAM wchann = {nzell*l_stripe*2*wfact}
173 .PARAM Cgs = {nzell*eps0*eoX*2*lchann*l_stripe/t_ox}
174 .PARAM Cgs_top = {nzell*eps0*eoX*1_mesa*l_stripe/t_ox}
175 .PARAM ACgs = {nzell*2*lchann*l_stripe}
176 .PARAM A_mesa = {nzell*4*w_mesa*l_stripe}

```
- Input slots:** Standard
- Copy Mode:** Force integer as real
- Base:** Parameter_%02d **Format:** default **Marker:** Expandable Stop at line end
- Name:** u_surf **Add** **>>** **Preview**
- Show parametrized locations:** Store file content
- Show additional options:**
- Buttons:** OK, Cancel, Apply

Selection of parameters to be optimized

Agenda

1

Introduction & Motivation

2

SIMatrix integration

3

Calibration Flow

4

Example 1 : Single transient curve

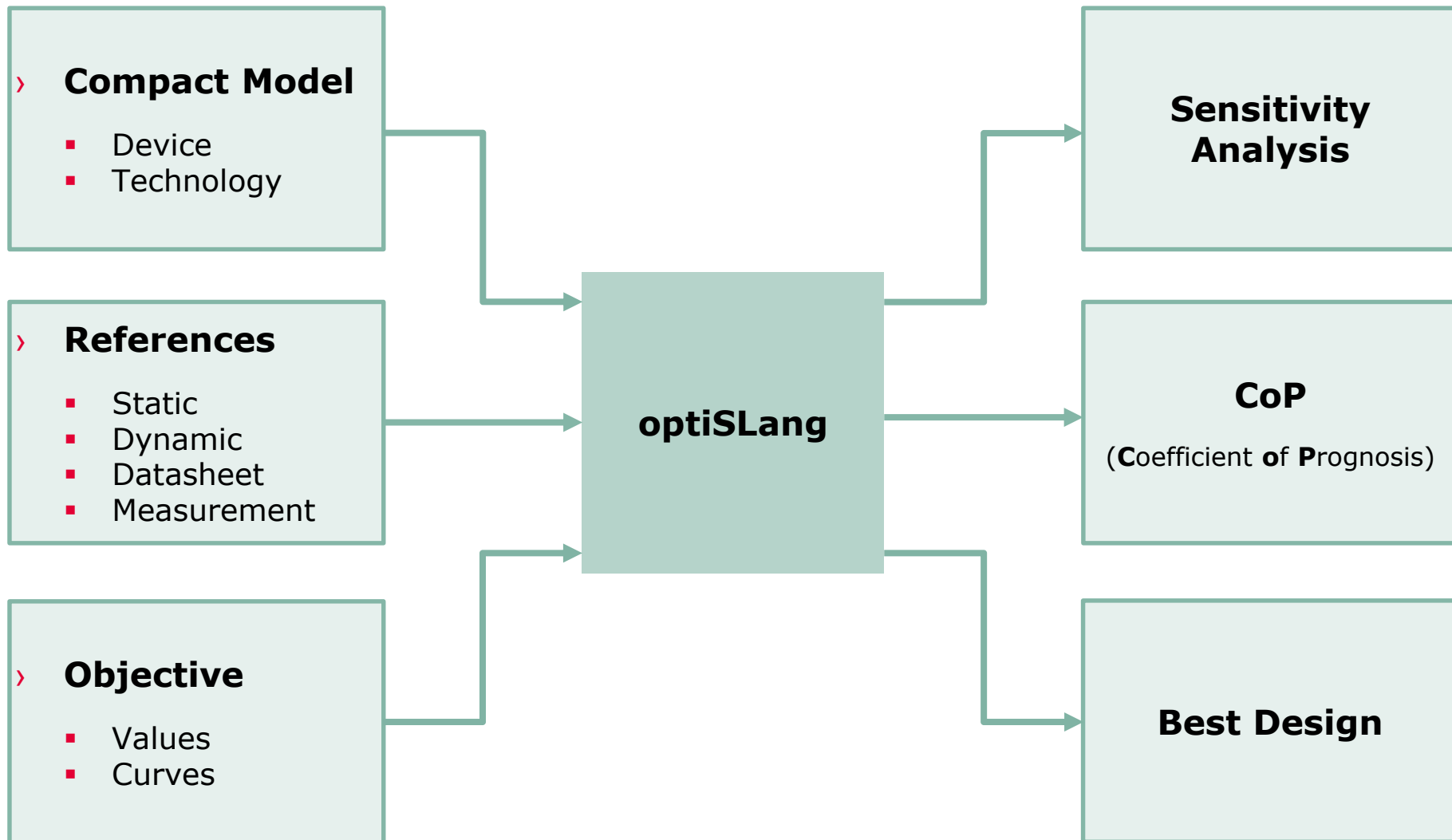
5

Example 2 : Two static curves (Transfer and Output)

6

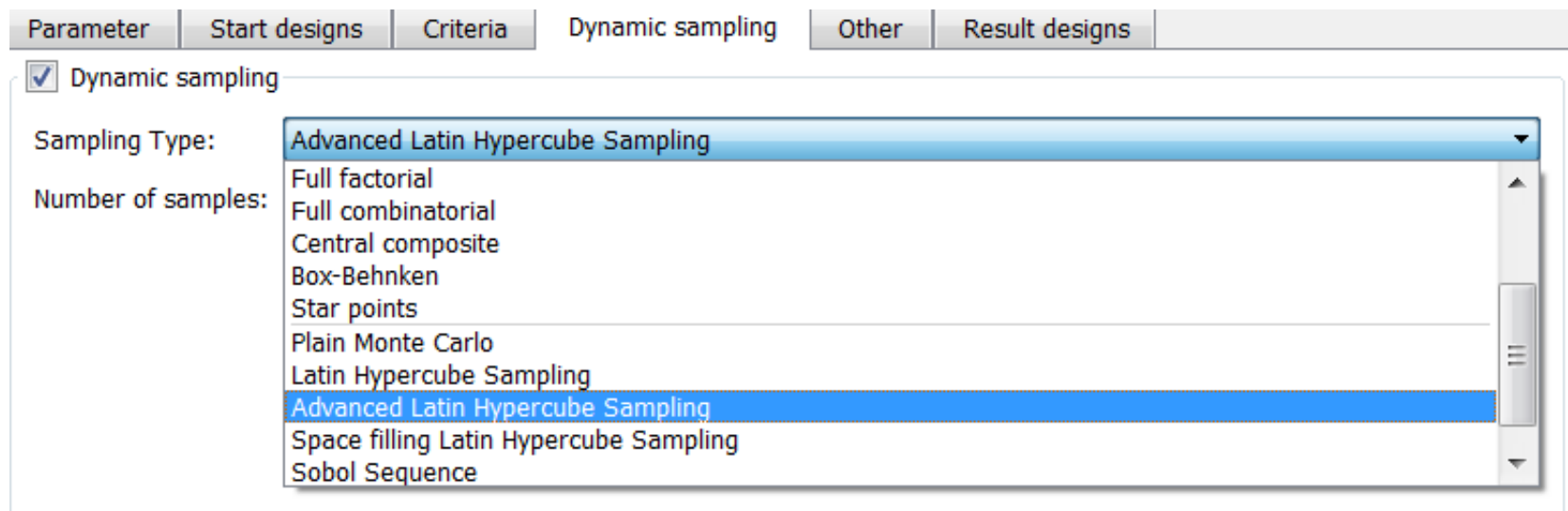
Conclusion/Future Outlook

Calibration Flow



Calibration flow (detailed)

- 1) Identify parameters to be optimized for the calibration
- 2) Define input parameter range, which the optimizer must follow
- 3) Input reference data (measurement) to be calibrated
- 4) Definition of calibration objective using reference and simulation output (quadratic difference)
- 6) Perform a **Sensitivity Analysis** (DoE) using built in wizard



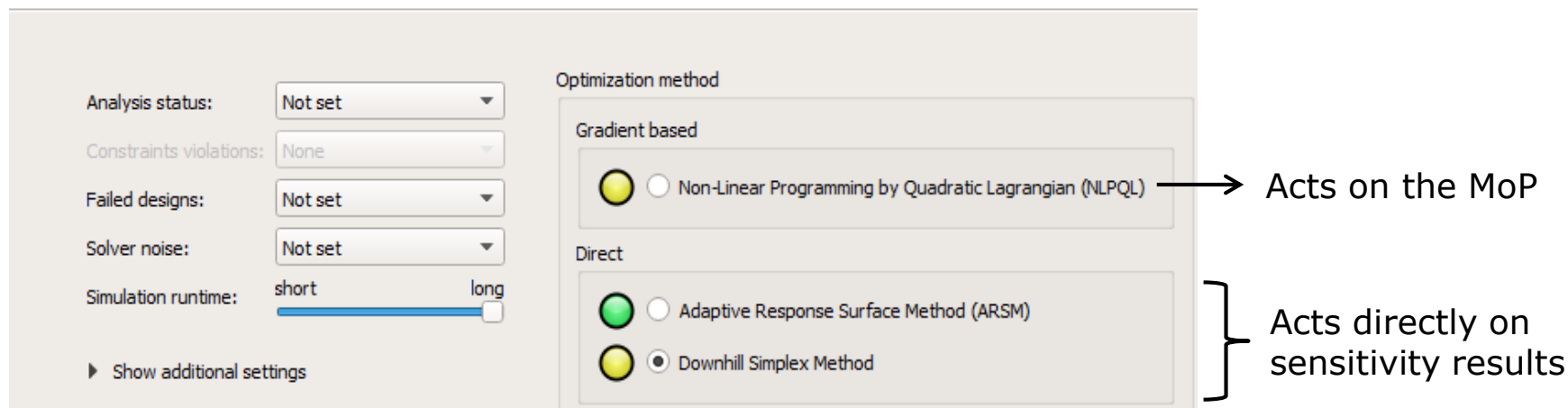
- › ALHS : 100 simulations (default)
- › Full factorial : $2^{\text{Number_of_Parameters}}$ simulations

Calibration flow (contd.)

- 6) The MoP can be edited at this point to remove outliers (if any) to create a reduced design space for the final optimization step
- 7) An **Optimization** step is carried out either on the MOP output or directly
 - A set of best fit parameters are displayed
 - History of parameter variation are shown
 - Comparison of reference and optimized data

Optimization method

Specify the optimization method



Analysis status:

Constraints violations:

Failed designs:

Solver noise:

Simulation runtime:

▶ Show additional settings

Optimization method

Gradient based

Non-Linear Programming by Quadratic Lagrangian (NLPQL)

Direct

Adaptive Response Surface Method (ARSM)

Downhill Simplex Method

Acts on the MoP

Acts directly on sensitivity results

Agenda

1

Introduction & Motivation

2

SIMatrix integration

3

Calibration Flow

4

Example 1 : Single transient curve

5

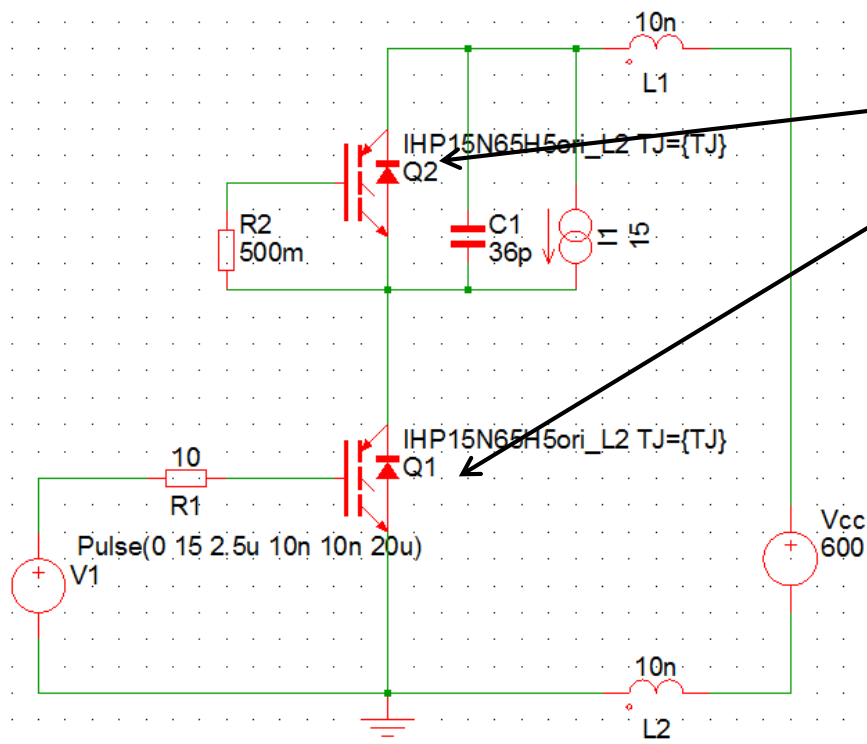
Example 2 : Two static curves (Transfer and Output)

6

Conclusion/Future Outlook

Example 1 : Single transient curve

Calibration on a simulated curve

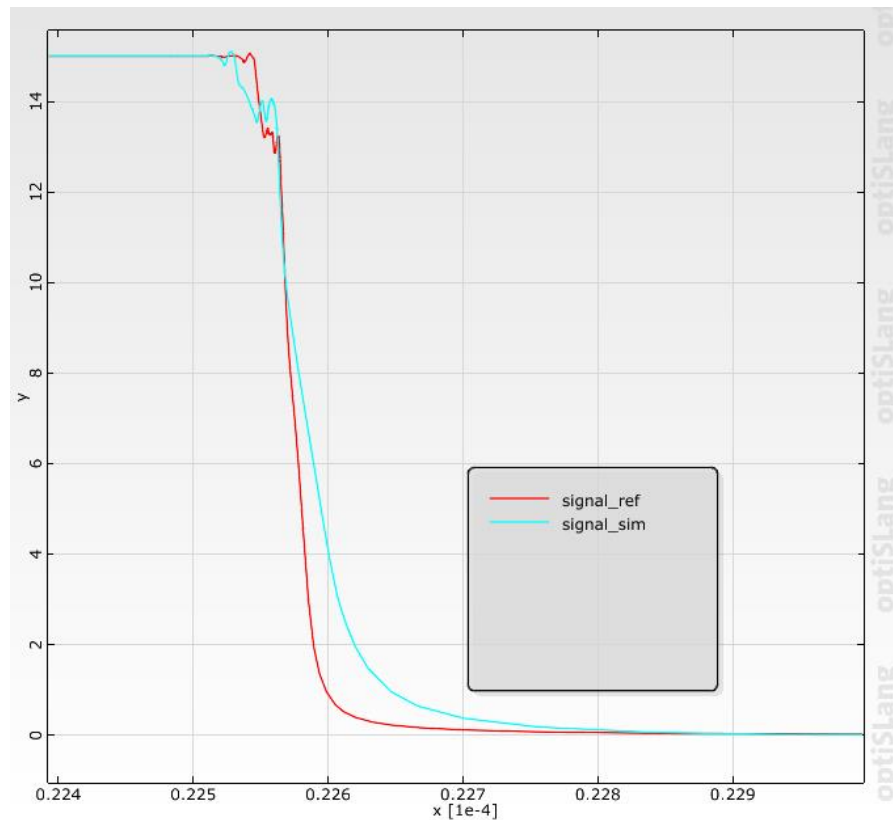
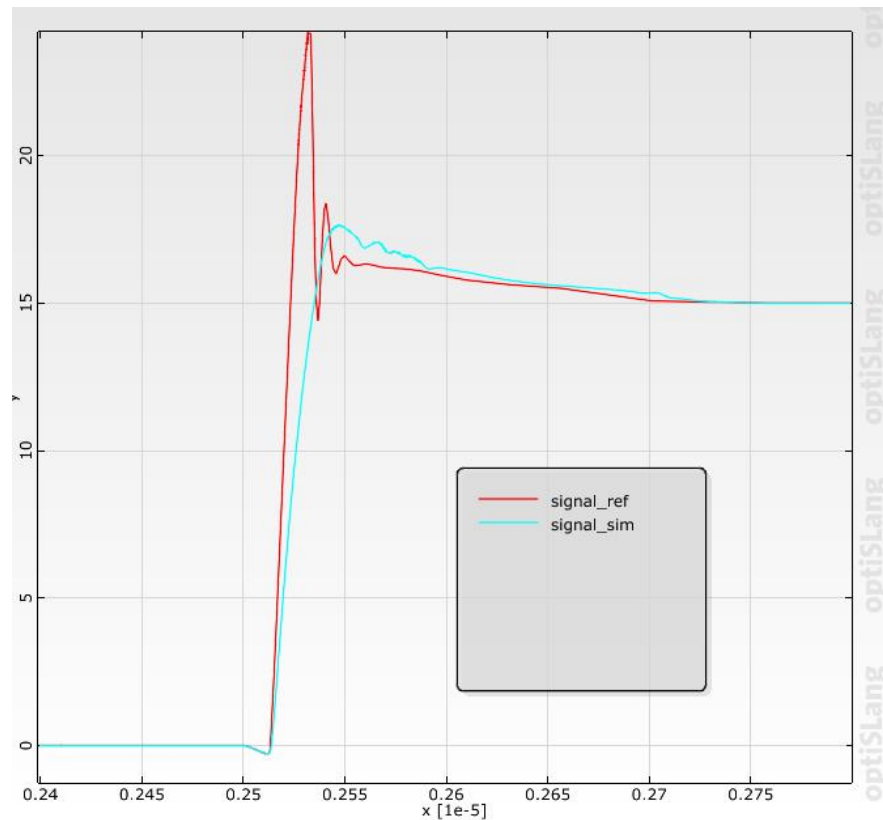


650V IGBT + diode

- > SPICE model manually de-calibrated
- > Test: Can optiSLang give us back the original set of parameters ?










Test schematic used

Starting point of optimization



- > Red: reference
- > blue : uncalibrated initial solver output

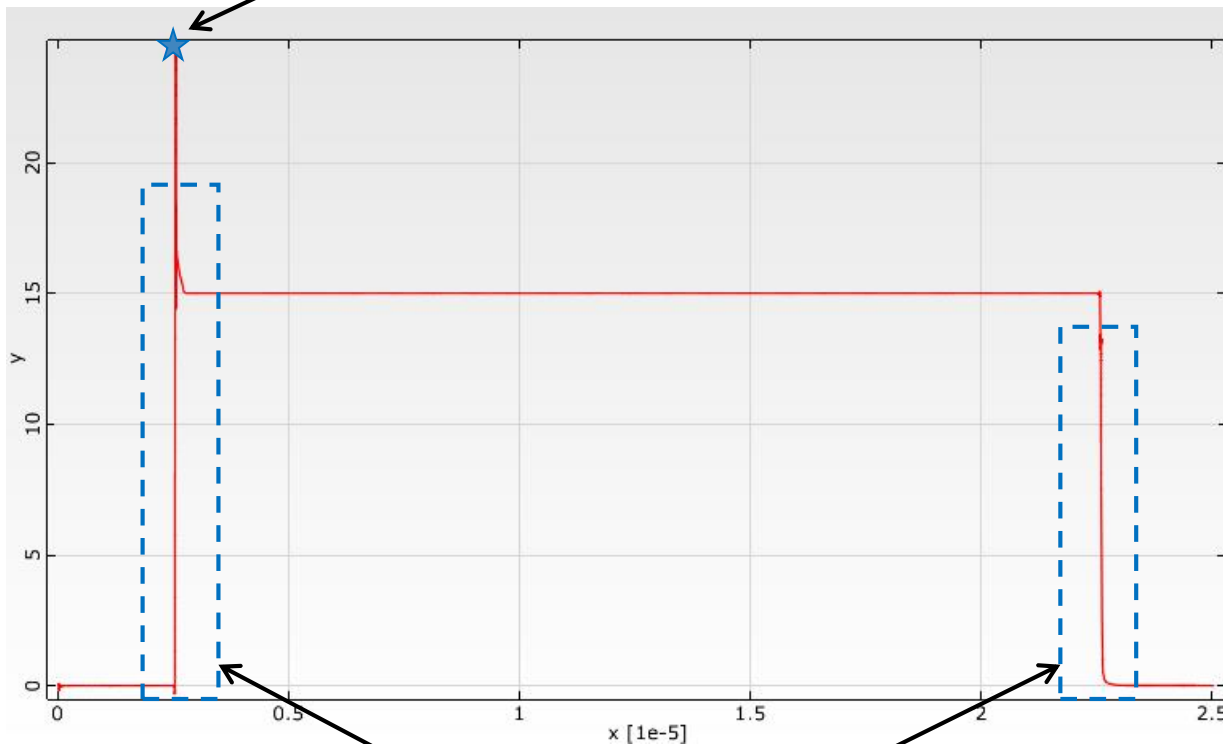
Parameter range definition

	Name	Parameter type	Reference value	Constant	Value type	Resolution	Range		Range plot
1	EMU	Optimization	0.1	<input type="checkbox"/>	REAL	Continuous	-2	1	
2	Ngate	Optimization	6e+09	<input type="checkbox"/>	REAL	Continuous	5.4e+09	6.6e+10	
3	Taub	Optimization	5e-06	<input type="checkbox"/>	REAL	Continuous	1e-06	3e-05	
4	ETAUB	Optimization	0.3	<input type="checkbox"/>	REAL	Continuous	0.25	2	
5	MosN	Optimization	0.1	<input checked="" type="checkbox"/>	REAL	Continuous	0.09	0.11	
6	u_surf	Optimization	75	<input type="checkbox"/>	REAL	Continuous	50	500	
7	Nak	Optimization	8e+16	<input type="checkbox"/>	REAL	Continuous	1e+15	1e+17	
8	Nem	Optimization	1e+16	<input type="checkbox"/>	REAL	Continuous	1e+15	1e+17	
9	Wem	Optimization	0.001	<input type="checkbox"/>	REAL	Continuous	9e-05	0.0011	

- › Parameter range definition
- › 8 Parameters for variation

Calibration objective definition

› Current Maxima



› Two windows defined in turn on and turn off region

Calibration objective definition contd...

Parameter		Responses	
Name	Value	Name	Value
EMU	0.1	current_steps_1_ref	[10]
ETAUB	0.3	current_steps_1_sim	[10]
Nak	8e+16	current_steps_2_ref	[10]

Name	Type	Expression	Criterion
obj_error_norm	Objective	$(\text{max_ord_ref} - \text{max_ord_sim})^2 + \text{euklidnorm}(\text{current_steps_1_ref} - \text{current_steps_1_sim}) + 3 * \text{euklidnorm}(\text{current_steps_2_ref} - \text{current_steps_2_sim})$	MIN

Objective=

$$(\text{max_curr_ref} - \text{max_curr_sim})^2 + \text{euklidnorm}(\text{current_steps_1_ref} - \text{current_steps_1_sim}) + 3 * \text{euklidnorm}(\text{current_steps_2_ref} - \text{current_steps_2_sim})$$

Maxima of curve

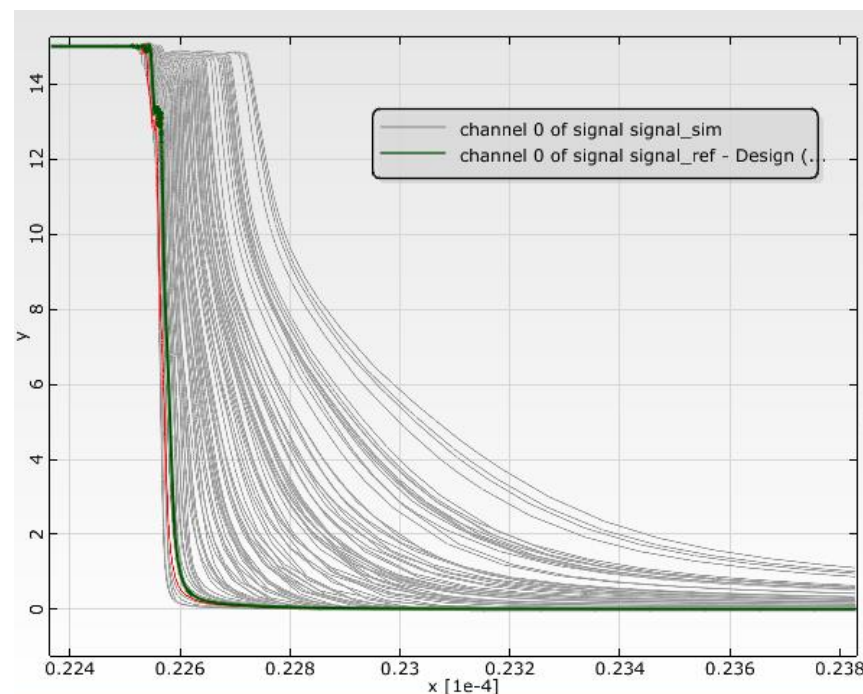
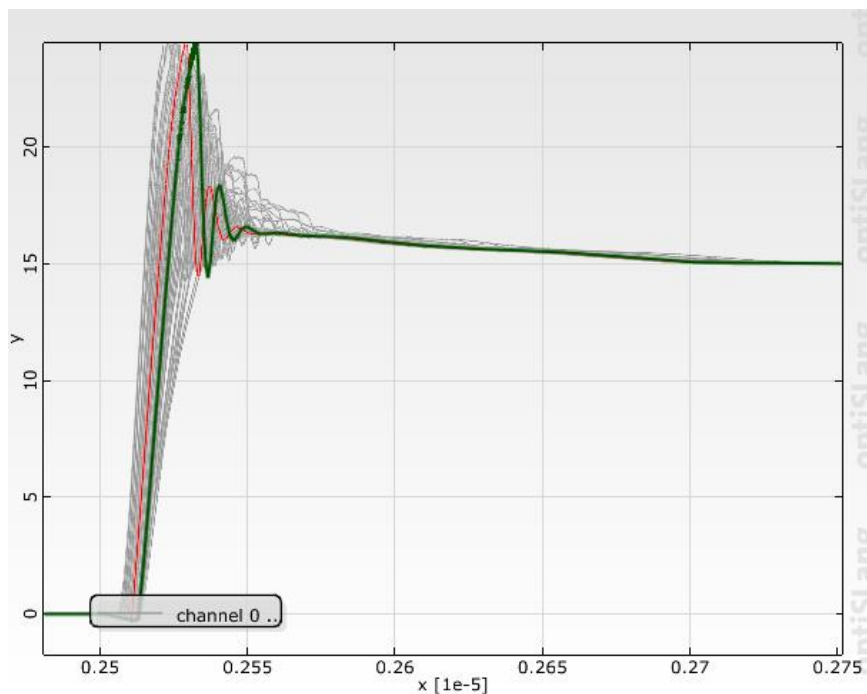
Two windows defined in turn on and turn off region

Weightage can be set depending on importance

- > Goal is to minimize the objective defined.
- > Zero objective would mean that the reference and the simulation are coinciding

Fit summary after sensitivity analysis

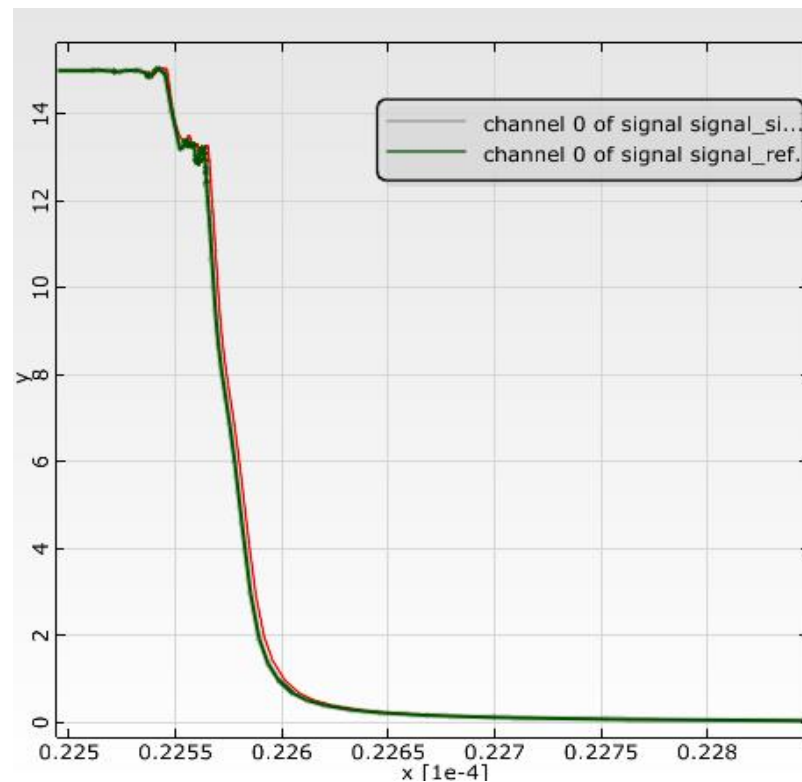
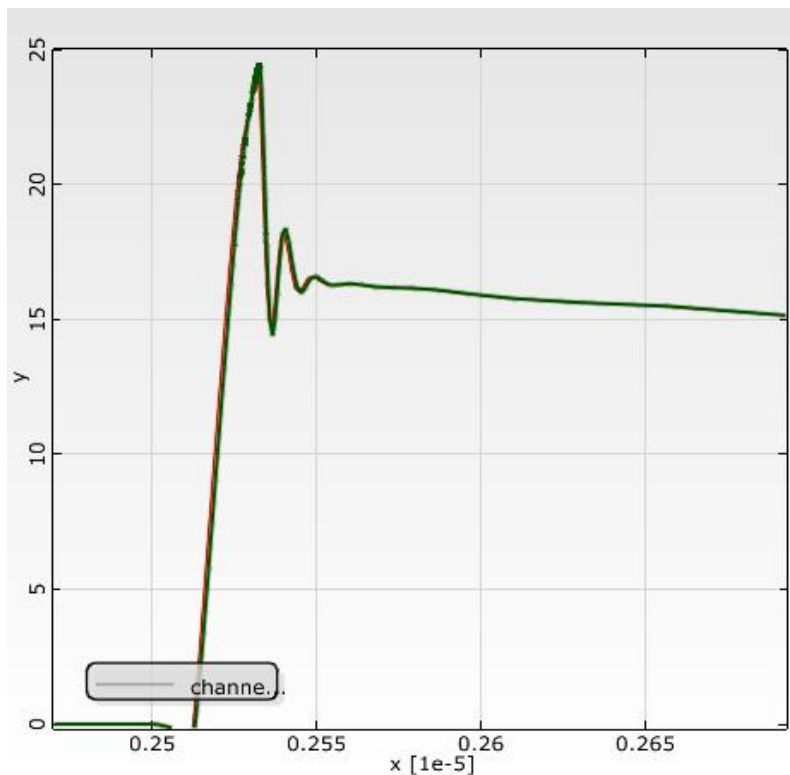
- > Green: reference
- > Red : best design(sensitivity)



- > Grey lines indicate all 100 simulations performed by sensitivity analysis
- > Good enclosure of the target curve means that the design space is well defined by the parameter range
 - > In case of poor enclosure the parameter ranges need to be redefined and Sensitivity analysis has to be redone

Fit summary after final optimization

- > Green : reference
- > Red : best design (optimization)



Parameter	Nk	Nm	ETB	u_s	Tb	Ng
Optimized Value	7.78e16	3.46e15	1.24	500	2.86e-6	1.52e10
Initial Value	8.5e16	2.75e16	1.5	450	2.5-6	2.58e10

Agenda

1

Introduction & Motivation

2

SIMatrix integration

3

Calibration Flow

4

Example 1 : Single transient curve

5

Example 2 : Two static curves (Transfer and Output)








6

Conclusion/Future Outlook

Example 2 : Two static curves (Transfer and Output)

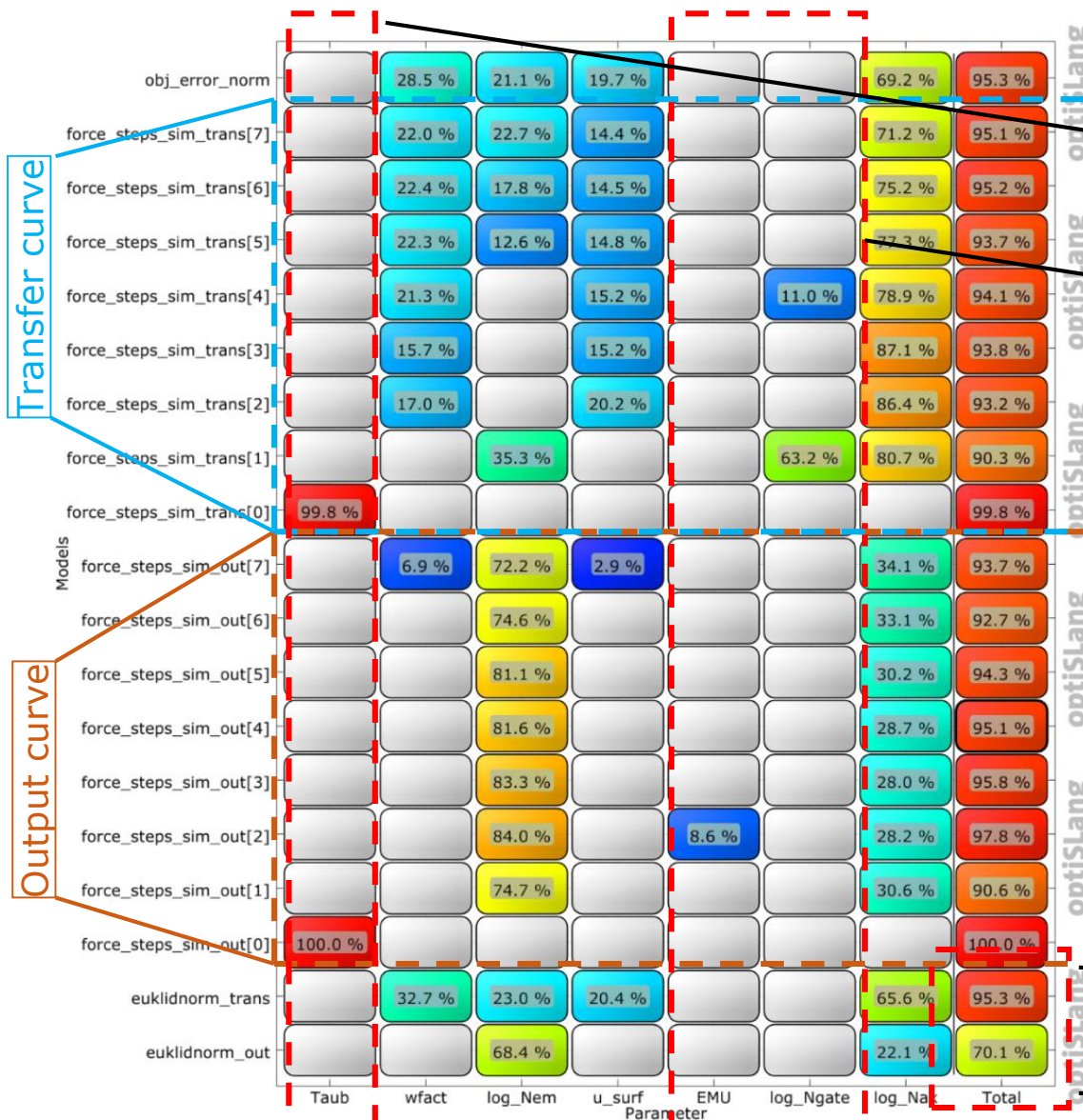
Calibration to measured curves:

Parameter range definition

	Name	Parameter type ▾	Reference value	Constant	Operation	Value type	Resolution	Range		Range plot
1	Taub	Optimization	2.5e-05	<input type="checkbox"/>		REAL	Continuous	1e-06	9e-05	
2	wfact	Optimization	1	<input type="checkbox"/>		REAL	Continuous	0.5	5	
3	log_Nem	Optimization	16	<input type="checkbox"/>		REAL	Continuous	14	17	
4	u_surf	Optimization	400	<input type="checkbox"/>		REAL	Continuous	100	600	
5	EMU	Optimization	-1.75	<input type="checkbox"/>		REAL	Continuous	-6	0	
6	log_Ngate	Optimization	10	<input type="checkbox"/>		REAL	Continuous	8	14	
7	log_Nak	Optimization	16	<input type="checkbox"/>		REAL	Continuous	14.7	17.9	
8	Nem	Dependent	3.4e+16	<input type="checkbox"/>	$3.4 \cdot 10^{\log_Nem}$					
9	Nak	Dependent	7e+16	<input type="checkbox"/>	$7 \cdot 10^{\log_Nak}$					
10	Ngate	Dependent	2.5e+10	<input type="checkbox"/>	$2.5 \cdot 10^{\log_Ngate}$					

- > Logarithmic parameter definition for wide range of variation
- > 7 parameter for variation

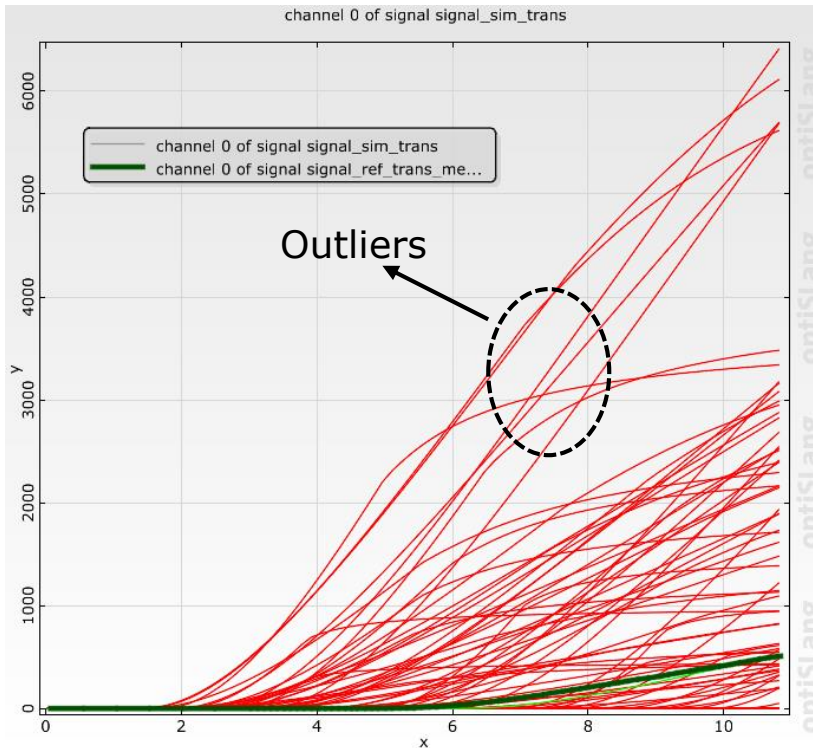
Initial CoP Matrix (Sensitivity analysis)



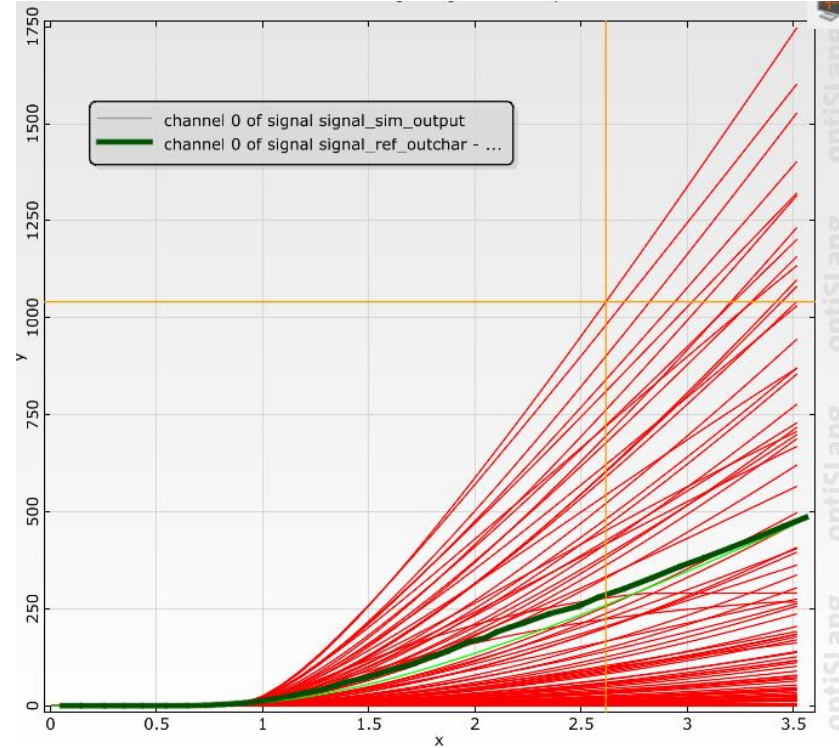
- > Taub only has impact in the initial part of the curves
- > EMU and Ngate has negligible impact on both curves
- > Transfer curve → reduction from 7 to 6 important parameter
- > Output curve → reduction from 7 to 3 important parameter

> different CoP value for Transfer and Output curves → Outliers???

Transfer curve

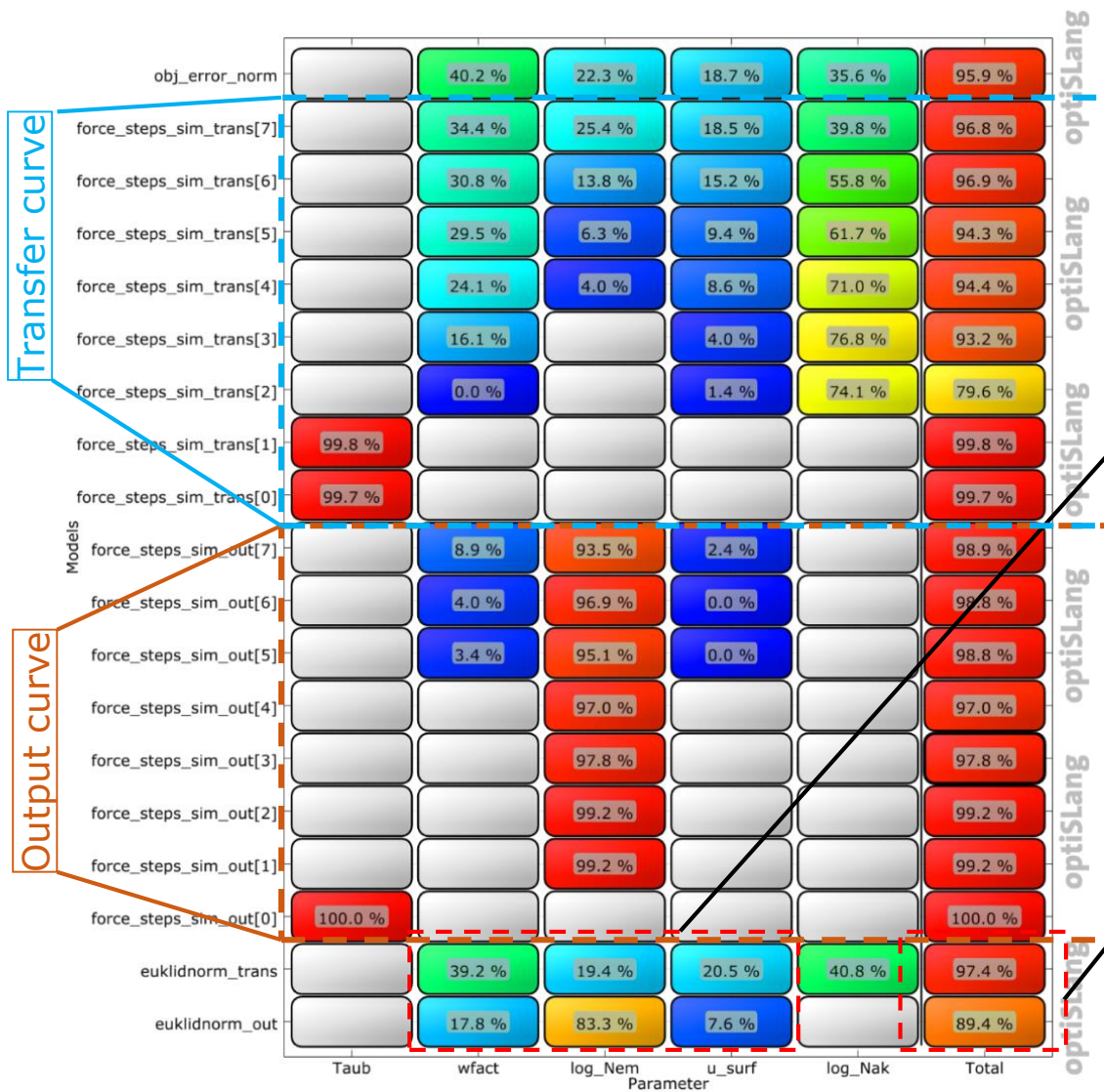


Output curve



- › MOP optimization :
 - Filter out negligible parameters manually
 - Deactivate outlier designs manually from the MOP
 - Goal is to increase the total CoP value -> better calibration possible

Optimized CoP Matrix



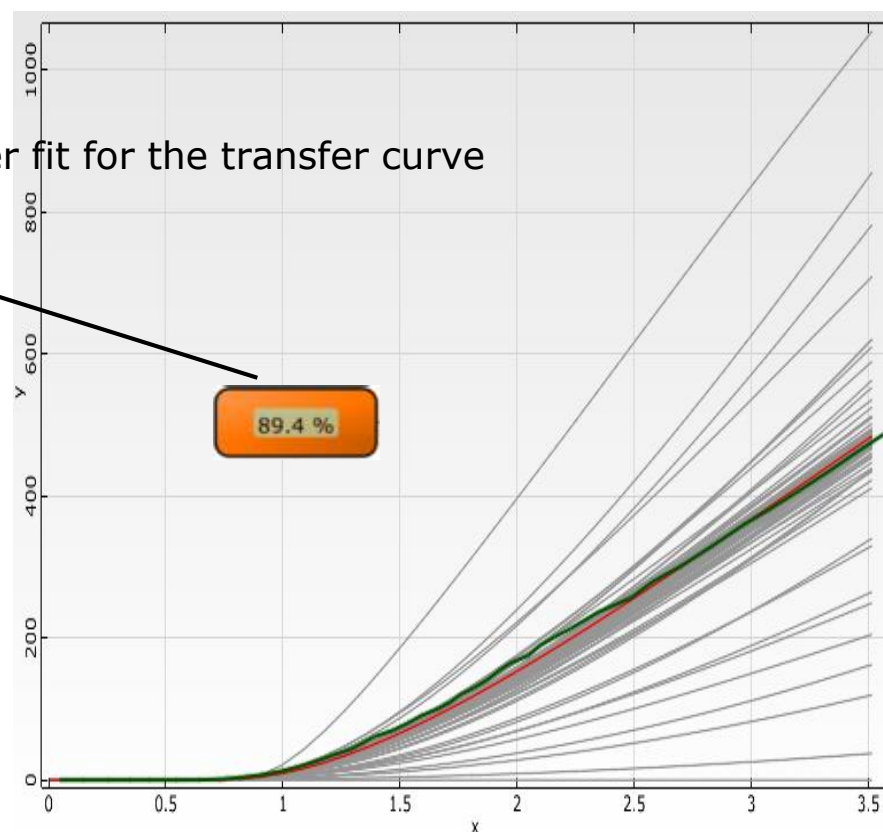
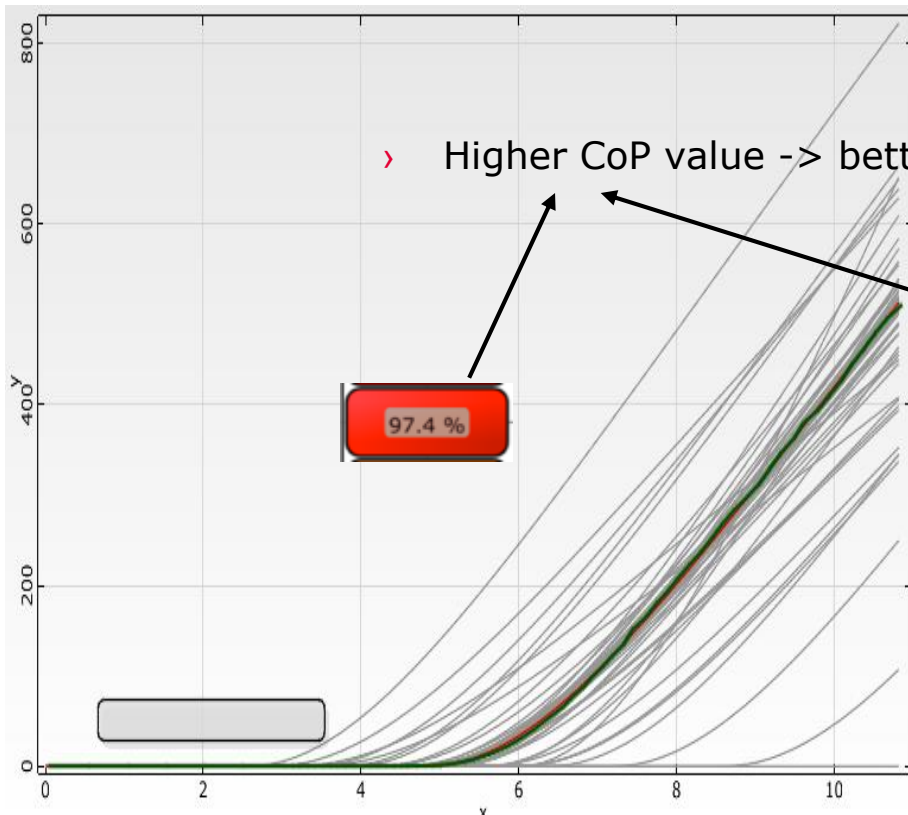
> Parameters having influence on both curves shows the importance of a global calibration

> With MOP optimization we could increase the total CoP value up to 90%

Fit summary after global optimization

Transfer curve

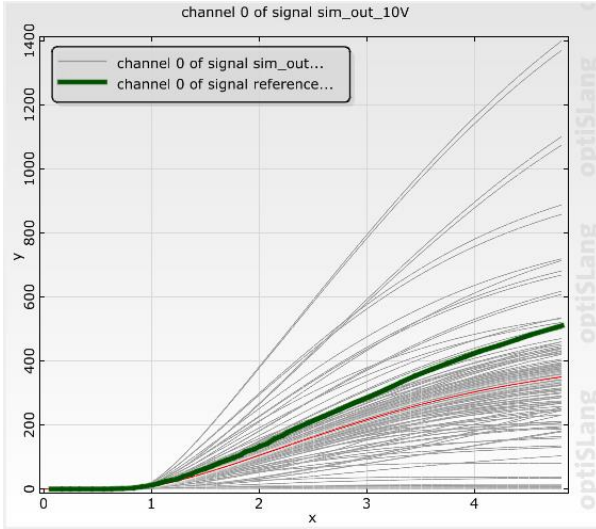
Output curve



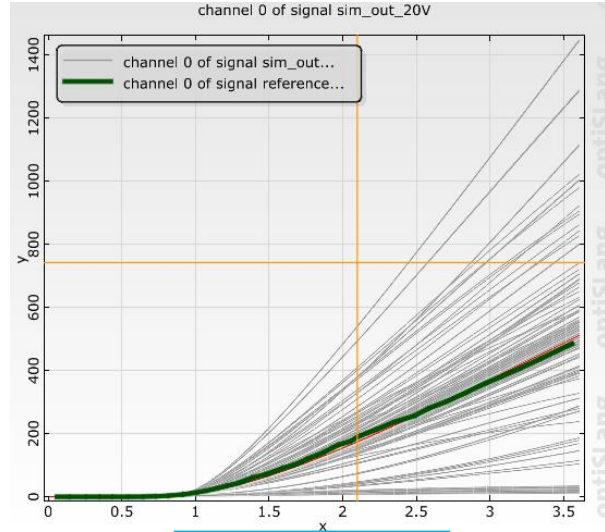
- > Green curve indicates the target curve from measurements
- > Grey curves indicate all simulations performed by direct optimization after Opt. on MOP
- > Red curves indicates the final optimizer fit → excellent agreement

4 curve system fit summary (2 transfer & 2 output curves)

Output curve

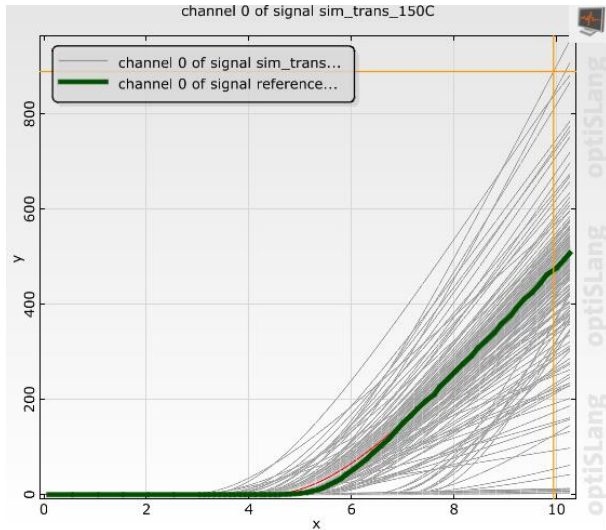


Output curve

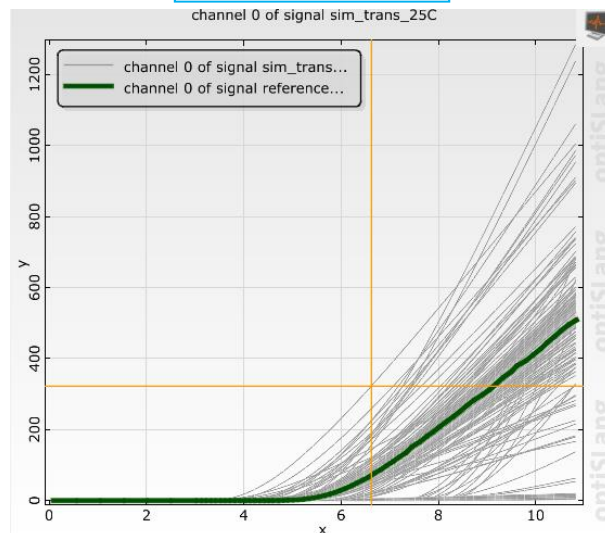


› Excellent agreement for 3 of 4 curves → weighted objective necessary

Transfer curve



Transfer curve



Agenda

1

Introduction & Motivation

2

SIMetrix integration

3

Calibration Flow

4

Example 1 : Single transient curve

5

Example 2 : Two static curves (Transfer and Output)

6

Conclusion/Future Outlook

Conclusion/Future Outlook

- › Current status
 - › optiSLang aided fitting of compact IGBT SPICE model demonstrated
 - › Good fit to measurements achieved in the defined parameter range
 - › Complex transient curve successfully calibrated
 - › Global calibration of upto four static curves (2 transfer & 2 output) demonstrated
 - › Greatly reduces the time and effort needed for compact model calibration

- › Outlook
 - › Evaluation of temperature dependence and transient curves in global calibration
 - › Define a standardized calibration flow with IGBT/diode measurements

THANK YOU FOR YOUR
ATTENTION

QUESTIONS ?



Part of your life. Part of tomorrow.

