OptiY 4.6

Machine Learning Framework for Big Data

World-First Universal and Accurate Model Transformation of 1D-Simulation

February 2021 - OptiY GmbH

Machine Learning: Classification (Meta-Model)

Inputs

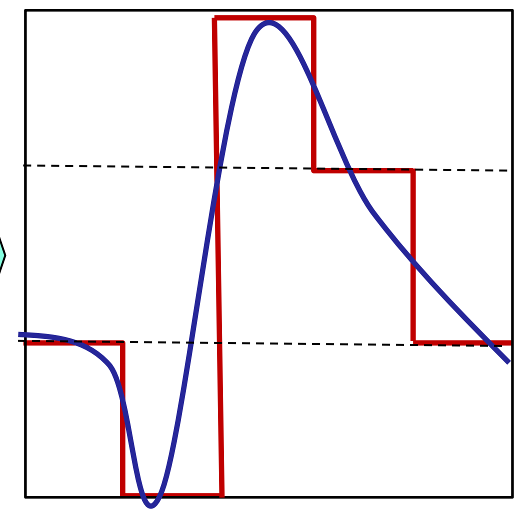
P_1
P_2
P_3
P_4
…
P_n

Algorithms
- Polynomial (Least Square)
- Gaussian Process

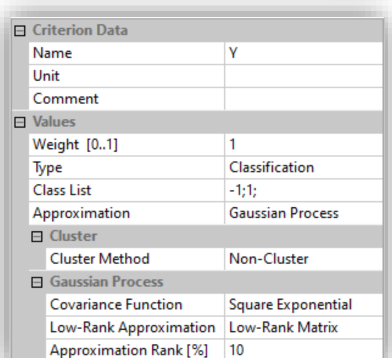
Types
- Binary Classification
- Multi-Classification

Advantages
- Universal as FEM-Simulation
- For Any Data-Structure

Outputs

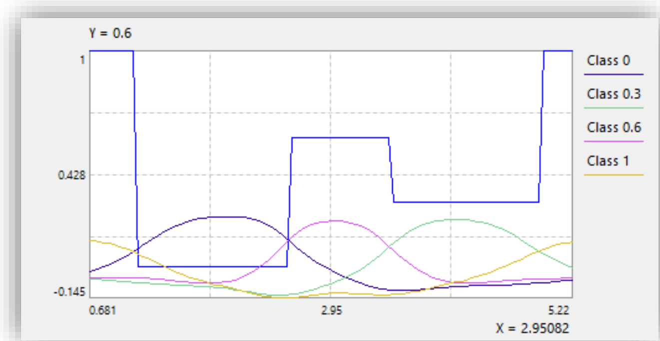


Classification = Digital-Simulation
Regression = Analog-Simulation

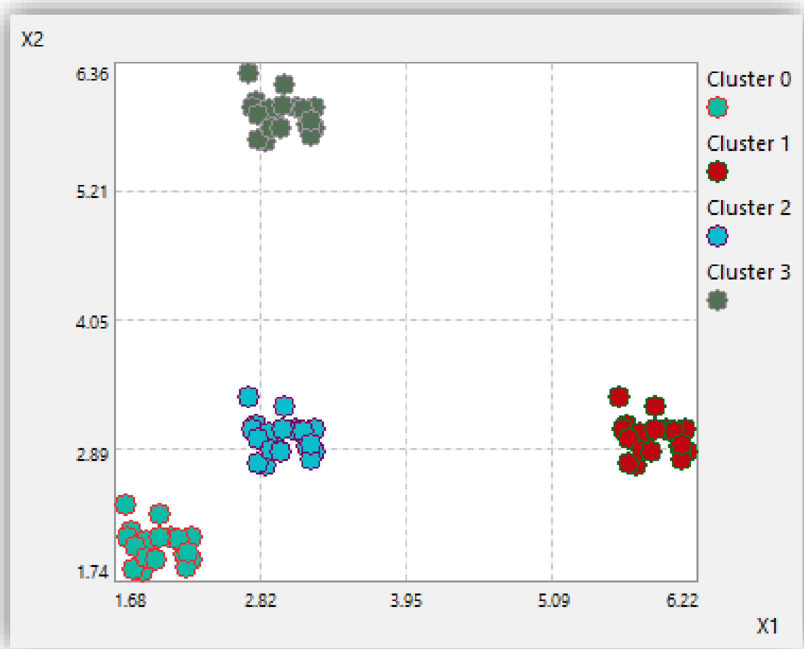


Criterion Data	
Name	Y
Unit	
Comment	
Values	
Weight [0..1]	1
Type	Classification
Class List	-1;1;
Approximation	Gaussian Process
Cluster	
Cluster Method	Non-Cluster
Gaussian Process	
Covariance Function	Square Exponential
Low-Rank Approximation	Low-Rank Matrix
Approximation Rank [%]	10

Applications
- Video tracking
- Object recognition
- Speech recognition
- Credit scoring
- etc…



Machine Learning: Clustering



Values	
Weight [0..1]	1
Type	Classification
Class List	-1;1;
Approximation	Gaussian Process
Cluster	
Cluster Method	K-Means
Include Output	False
Cluster Number	4

Grouping same Objects or Data Points in a Cluster

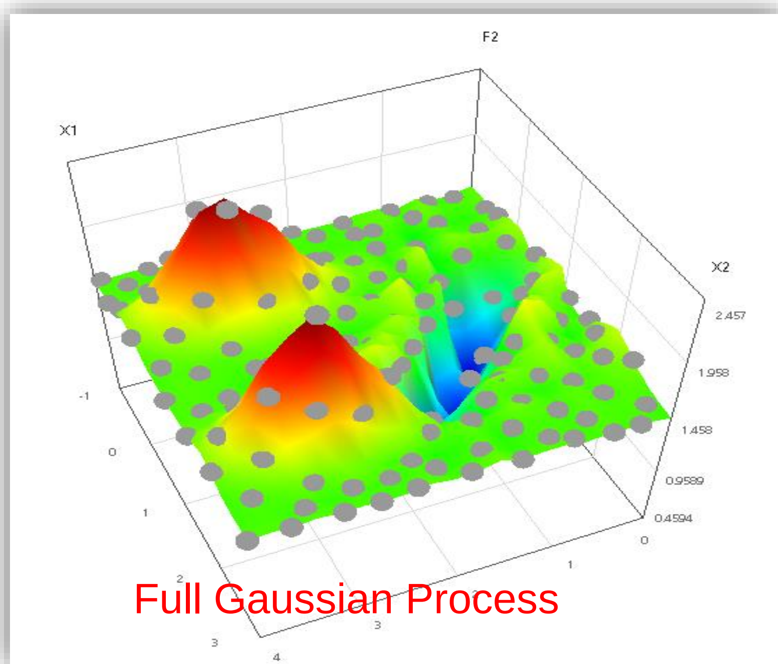
Algorithms

- K-Mean
- Binary Tree

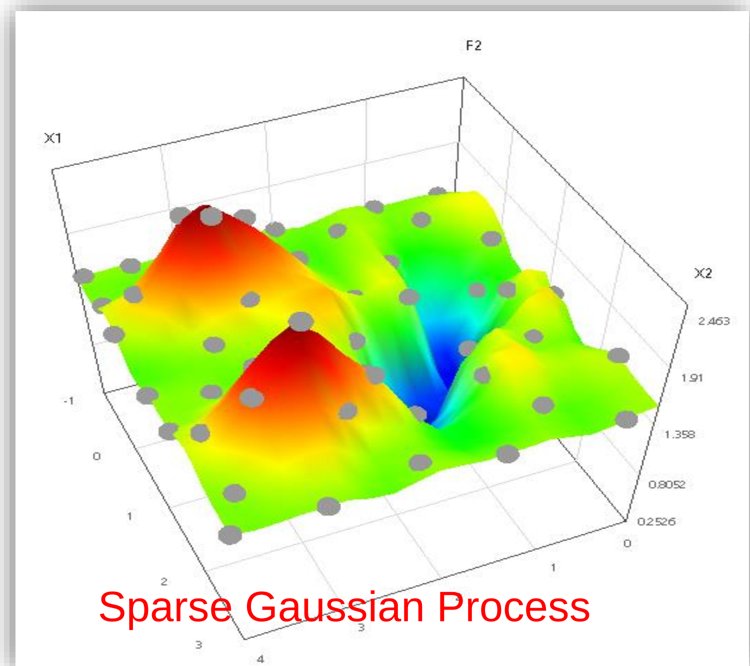
Applications

- Image segmentation
- Anatomy Detection
- Recommender Systems
- Prework for Regression/Classification
- etc …

Sparse Gaussian Process + Regulation of Least Square



Full Gaussian Process



Sparse Gaussian Process

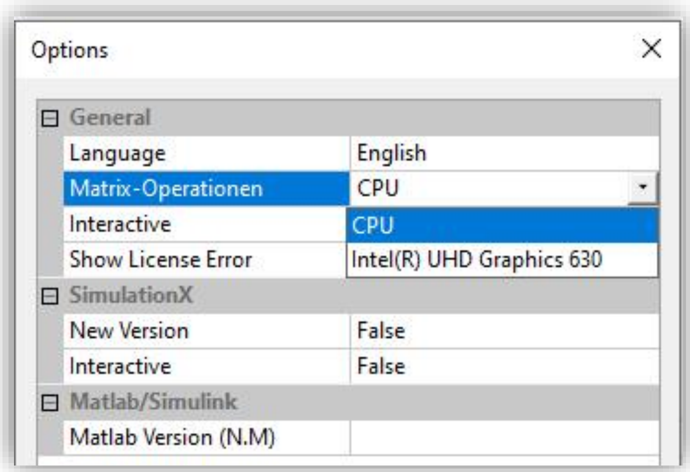
- Same accurate meta-model
- Select best data points / least Square coefficients for meta-model
- Fast inference for classification and regression
- Low-rank matrix approximation for big data

⊟ Values	
Weight [0..1]	1
Type	Regression
Approximation	Gaussian Process
⊟ Cluster	
Cluster Method	Non-Cluster
⊟ Gaussian Process	
Covariance Function	Matérn Class 3/2
Low-Rank Approximation	Low-Rank Matrix
Approximation Rank [%]	40
Gaussian Noise [%]	0.01
⊟ Polynomial	
Polynomial Type	Uniform Order
Polynomial Order	1
Low-Rank Approximation	Full Matrix

Accelerating Machine Learning Using GPU

User can choose CPU or GPU for accelerated training of machine learning

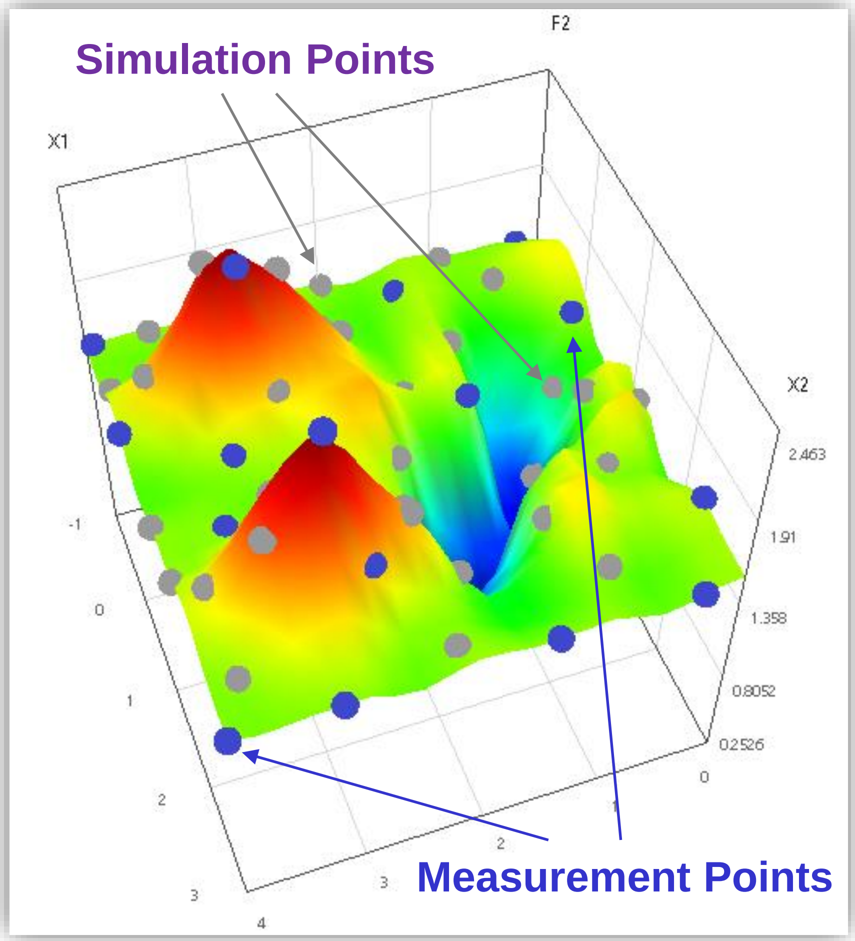
- Normal operations on CPU
- Fast parallel matrix operations on GPU
- Data parallelisms on GPU
- Automatic recognition of GPUs on board



OptiY supports a lot of GPUs

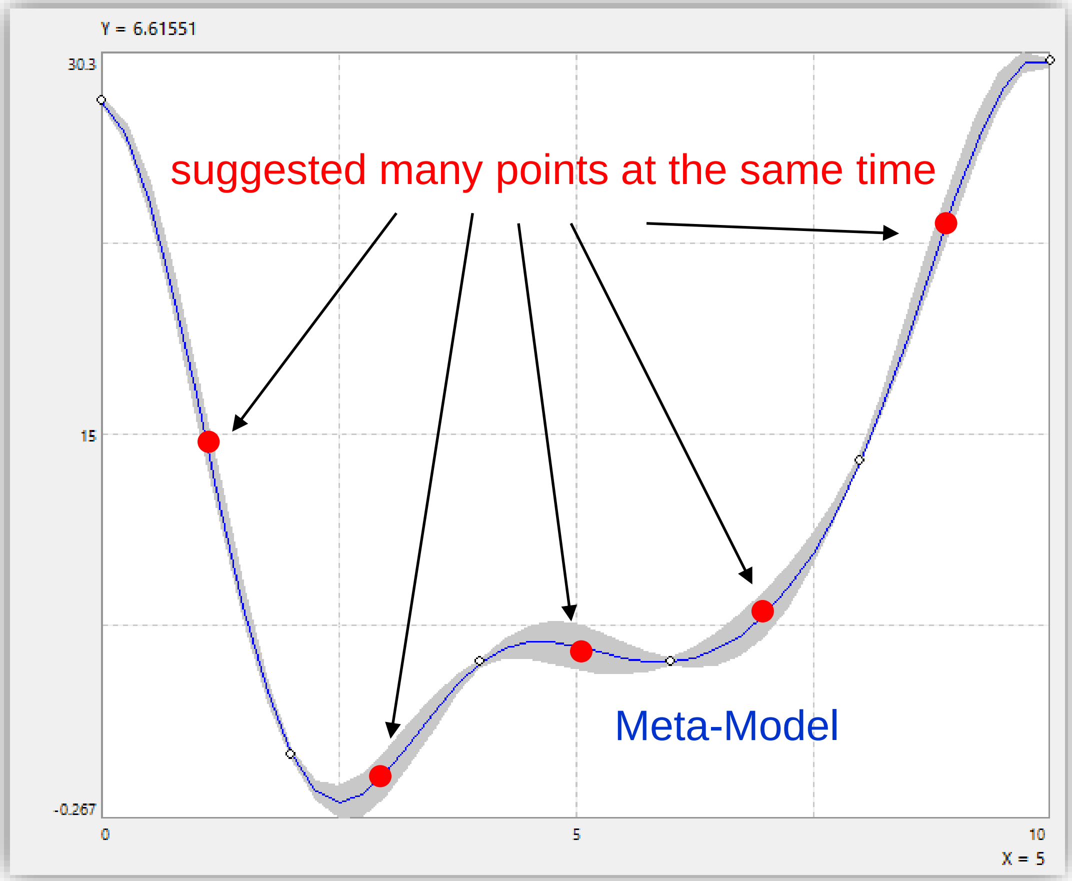
- Double-Precision
- Supported by Windows DirectX 11 or later
- Manufacturer: Nividea, Intel, AMD etc..

Meta-Model of both Measurement and Simulation Data



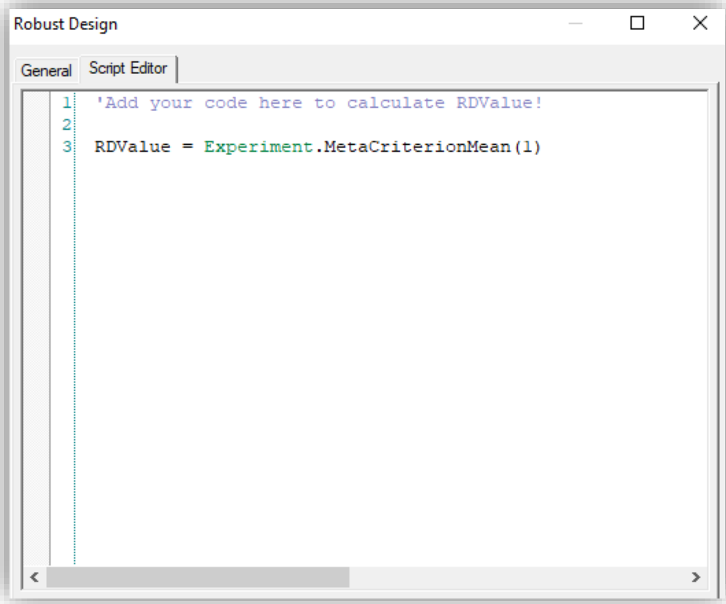
- Import first initial points from measurement or old simulation data for building the first meta-model (**blue points**)
- Based on the first meta-model, suggest new additional points for simulation (**gray points**) and start simulation
- Build better meta-model from both data points
- Fast and efficient design of experiment and meta-modeling

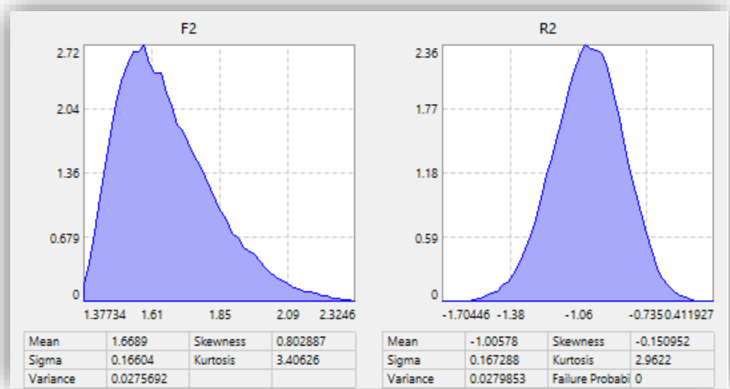
Uniform/Exactly Points Suggestion by adaptive Sampling



- Based on existing metamodel, suggest many points at the same time for next simulation or measurement on prototypes
- All suggested points are uniform scattered and exact located on the maxima of uncertainties of the metamodel.

Customizing Robust Design Optimization

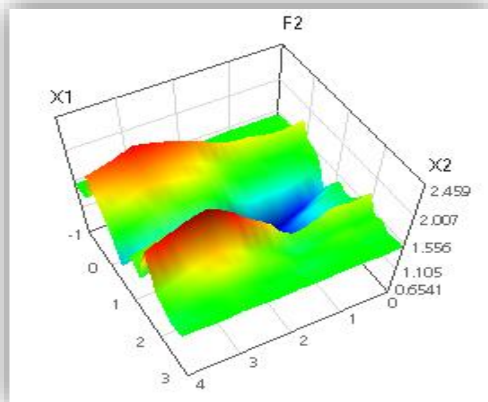


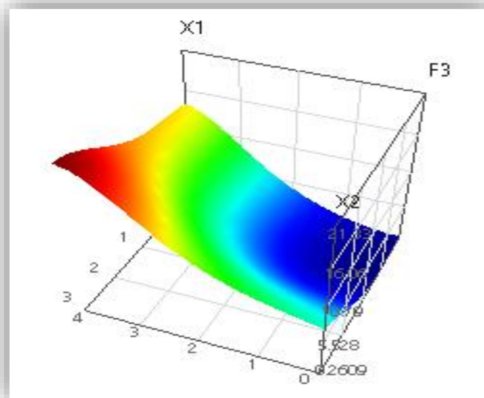


- Easy define customized objective functions or constraints by scripting language VB.NET or C# on Script-Editor
- Access to existing meta-models of experiment based on API with statistical values: tolerance, mean, variance, sigma, cost, 1D-value
- Just-able optimization methods: Hooke-Jeeves or evolution strategies
- Multi-objective optimization possible

MetaDesignParameter(Index)	"Design Parameter" of the stochastic parameter with index (1...count) or name
MetaParameter(Index)	"Virtual Nominal" of the stochastic parameter with index (1...count) or name
MetaTolerance(Index)	"Virtual Tolerance" of the stochastic parameter with index (1...count) or name
MetaCost	The cost function based on virtual tolerances of all stochastic parameters
MetaConstraint(Index)	"Virtual Value" of the constraint with index (1..count) or name based o the mata-model
MetaCriterion(Index)	"Virtual Value" of the criterion with index (1..count) or name based o the mata-model
MetaConstraintMean(Index)	Mean of the constraint with index (1..count) or name based o the mata-model
MetaCriterionMean(Index)	Mean of the criterion with index (1..count) or name based o the mata-model
MetaConstraintVariance(Index)	Variance of the constraint with index (1..count) or name based o the mata-model
MetaCriterionVariance(Index)	Variance of the criterion with index (1..count)or name based o the mata-model
MetaConstraintSigma(Index)	Standard-deviation of the constraint with index (1..count) or name based o the mata-model
MetaCriterionSigma(Index)	Standard-deviation of the criterion with index (1..count) or name based o the mata-model
MetaLifeTime(Index)	LifeTime of the strain withn index (1..count) or name based on the meta-model
Meta1D(Index, Type, Value)	"Virtual Value" of the 1D-variable with index (1..count) or name based o the mata-model Type = { 0 = last value, 1 = max.value, 2 = min. value, 3 = mean value, 4 = sum, 5 = absolute sum, 6 = band, 7 = standard deviation, 8 = integral, 9 = constraint, 10 = data-fitting, 11 = X by Y-point, 12 = Y by X-point, 13 = X by X-leap, 14 = X by Y-leap, 15 = Y by X-leap, 16 = Y by Y-leap, 17 = X tangente, 18 = Y tangente, 19 = first value, 20 = X by max. Y, 21 = X by min Y}

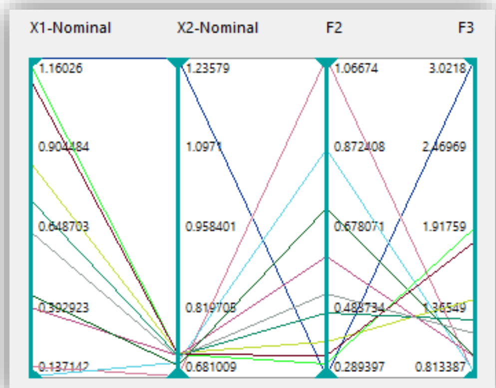
Most Efficient Multi-Objective Optimization





Design-Table

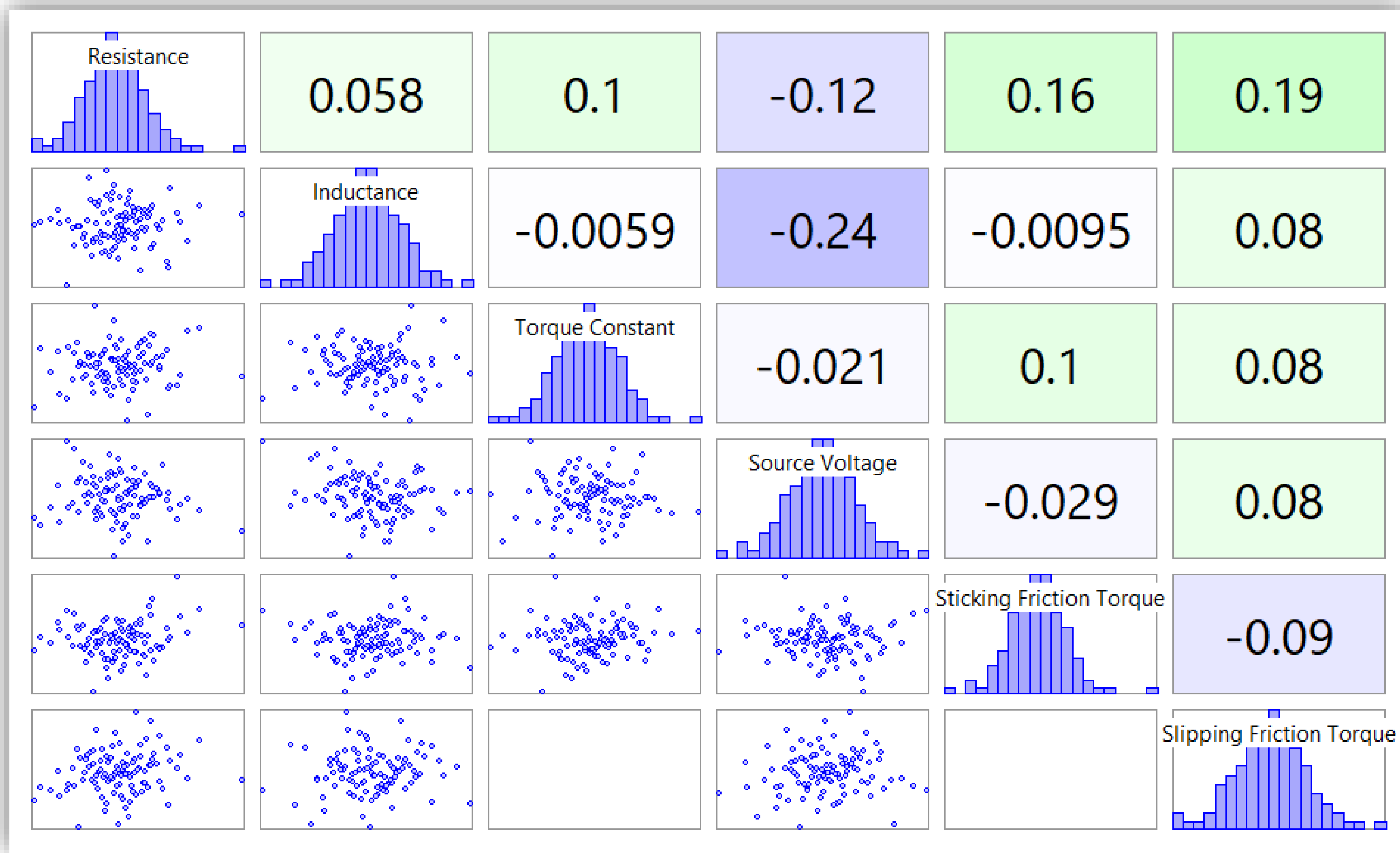
No	X1-Nominal	X2-Nominal	F2	F3
0	1.16026441	1.235792	0.289397299	3.02179623
1	0.168647947	0.681009417	1.0667448	0.813386798
2	0.137141932	0.705510159	0.844872177	0.83822
3	0.82446911	0.722208032	0.373756856	1.35731804
4	0.359347258	0.717857375	0.582903326	0.949300826
5	0.399718185	0.702054508	0.702094734	0.94793731
6	1.1392247	0.720750227	0.32231006	1.84899724
7	1.08969804	0.720640004	0.342505544	1.75742161
8	0.704821868	0.718033411	0.444314629	1.21403909
9	0.605456505	0.716069885	0.492285371	1.11640096



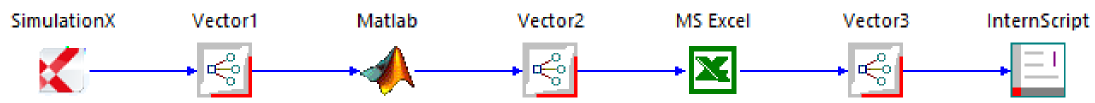
Adaptive Sampling = Loop

- First initial sampled points on design of experiment
- Build first meta-models for single objectives
- Do a multi-objective optimization on meta-models to find Pareto virtual points
- Start simulation of the original model for these new Pareto points
- Rebuild the meta-models for single objectives
- Start the next multi-objective optimization on meta-models

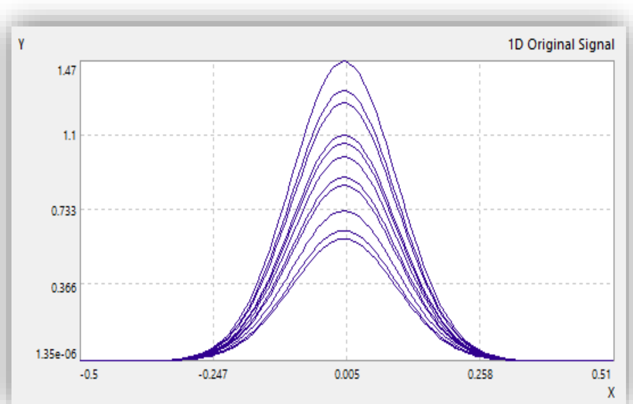
New General Survey: Correlation Matrix

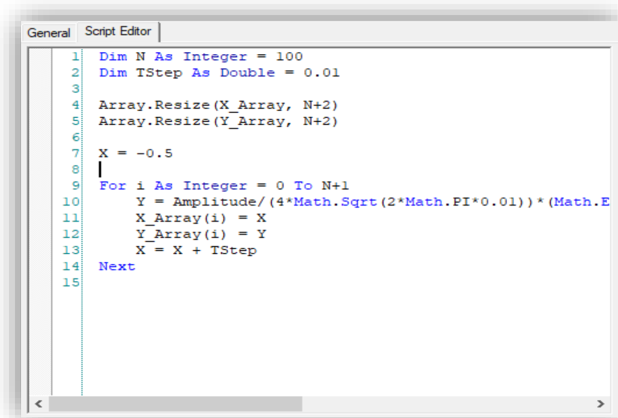


Vector-Transfer: SimulationX/Matlab/MS Excel/Script



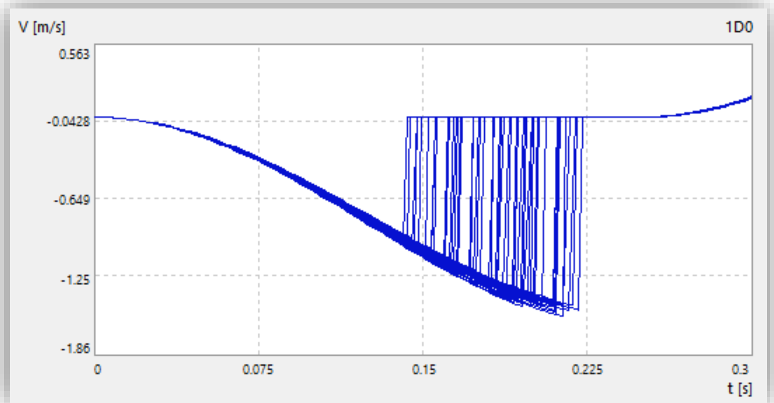
- Vector = Array of double
- Frequently used by dynamic simulation for signal processing
- Easy transfer between different external simulation system SimulationX/Matlab/Excel and Script inside of OptiY

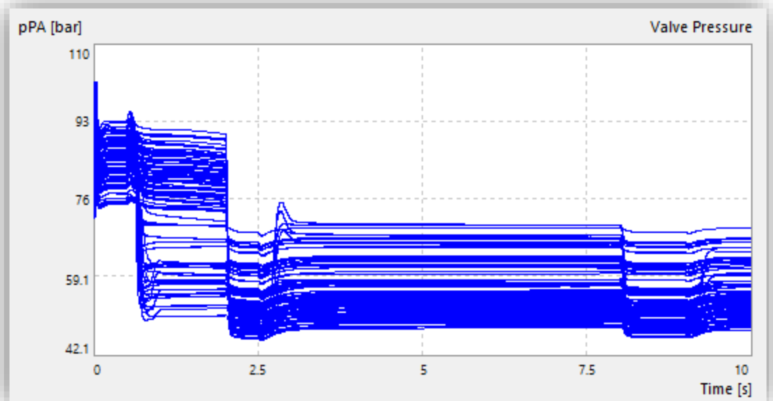




OptiY

Model Transformation of 1D-Simulation





New Algorithms

- Fast randomized algorithms for PCA
- Regulation of least-square
- Accurate meta-model for any complex 1D-curve of physical simulation
- Code-export in different target languages

Advantages

- Universal for any 1D-simulation
- Reliable meta-modeling
- Big data

Code-Export of Meta-Model

- New coding algorithms
- Fast compact parallel codes
- based on matrix operations
- Real-time codes
- Suitable for CPU/GPU/FPGA

C / C++

VB.net

Modelica

Matlab

C/C++

Visual Basic

Modelica Language

MATLAB