Laboratory for Product Development and Lightweight Design
Department of Mechanical Engineering
Technical University of Munich

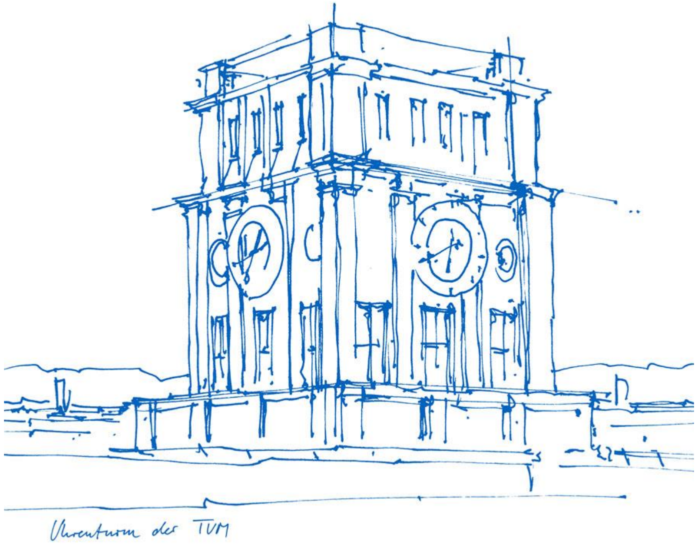
Solution Space Engineering

A Framework for Quantitative Systems Design

Markus Zimmermann

22nd DSM Conference, Cambridge

October 13, 2020



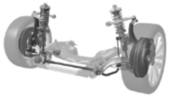
Markus Zimmermann

Academic Training

- TU Berlin, Mechanical Engineering
- University of Michigan, Mechanical Engineering
- Ecole Polytechnique
- MIT, PhD

BMW

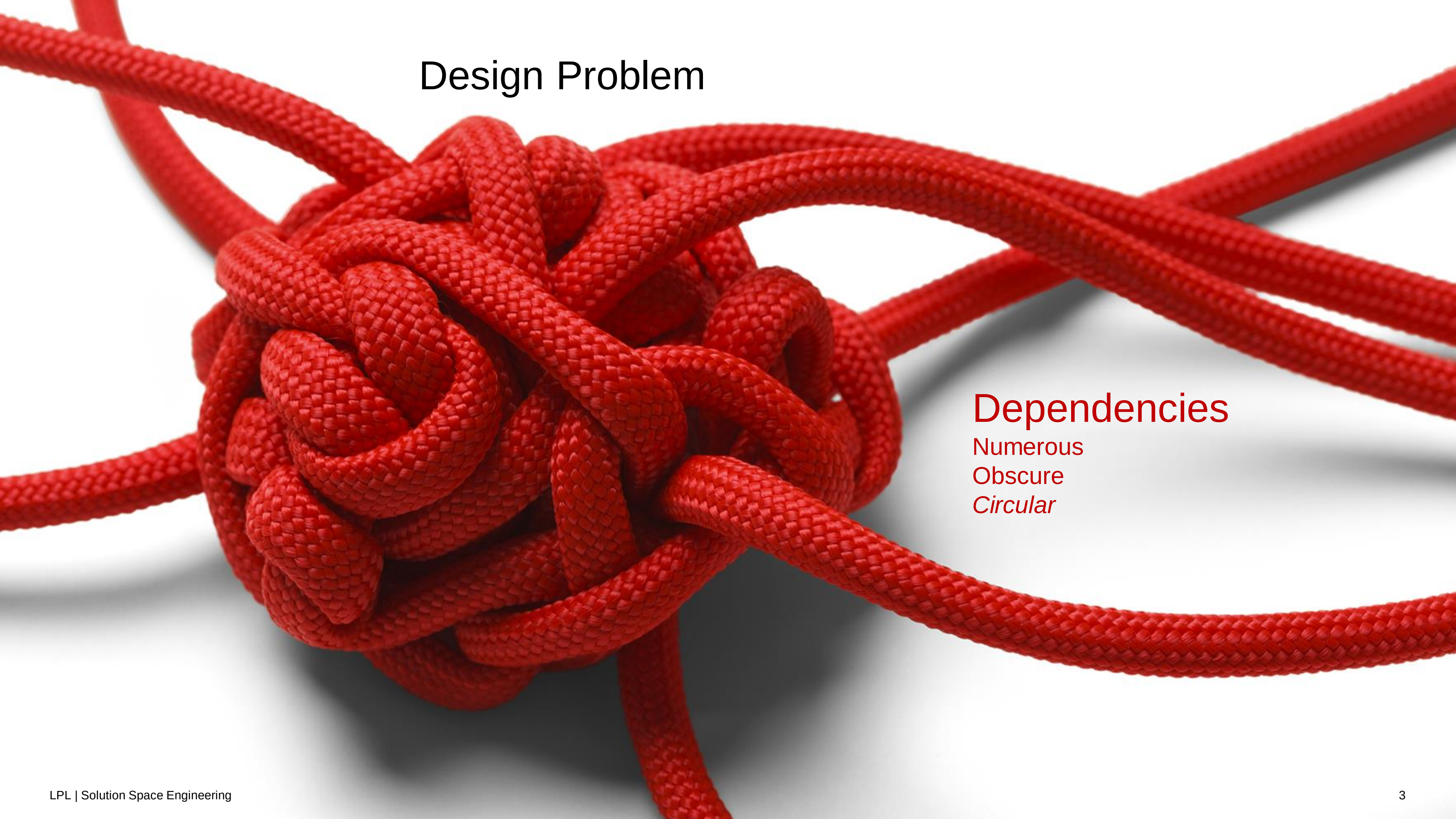
- Body design
- Crash design
- Vehicle dynamics
- Interdisciplinary projects



Technical University of Munich

- Since November 13th 2017

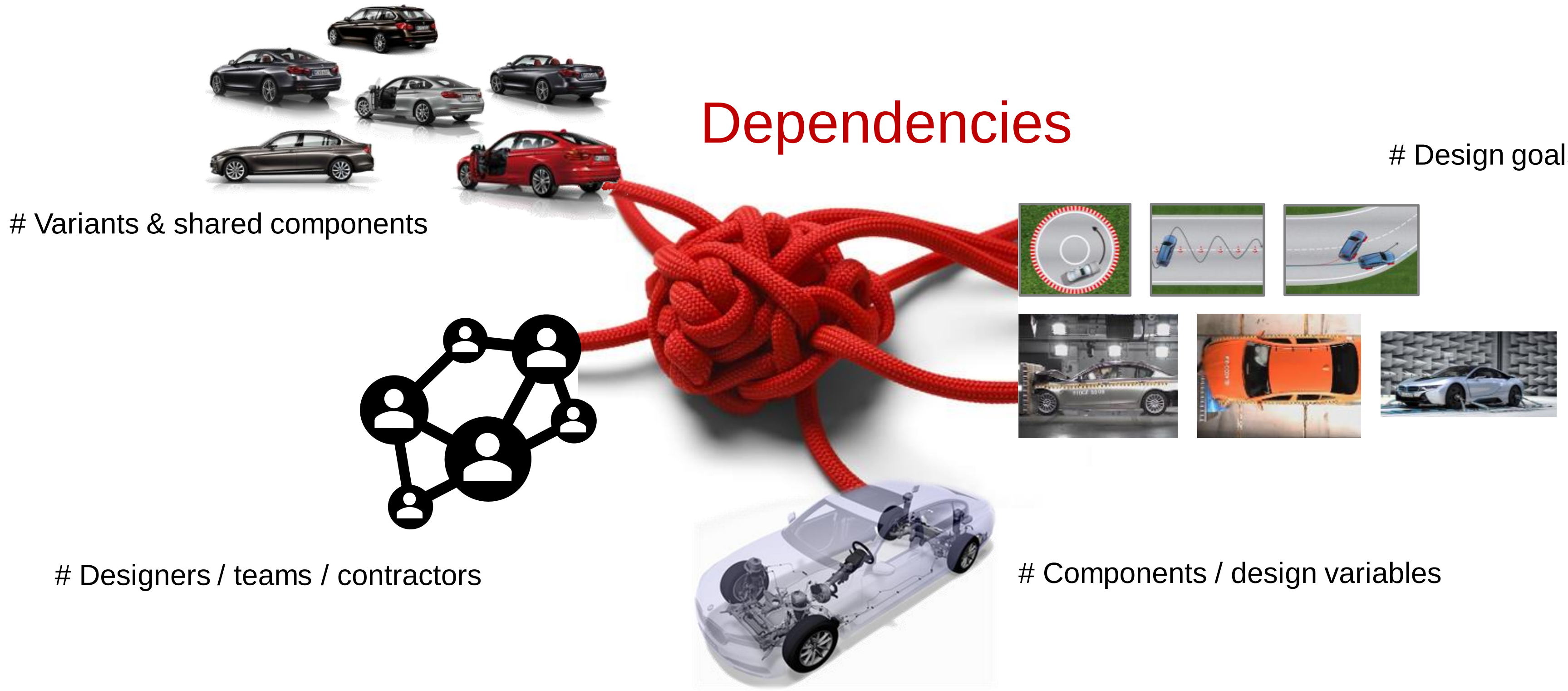
Design Problem



Dependencies

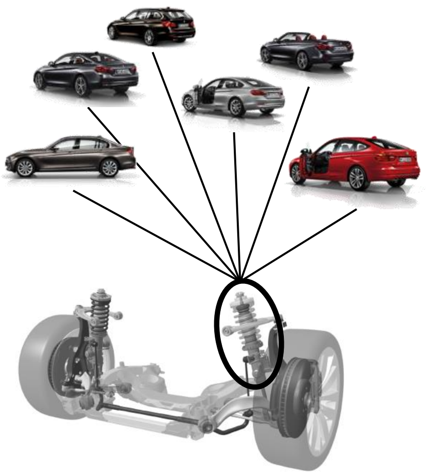
Numerous
Obscure
Circular

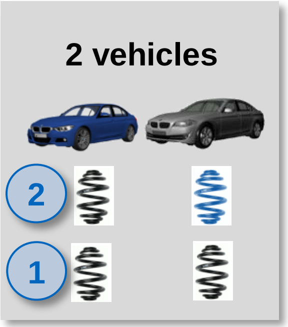
Some Complexity Drivers in Systems Design



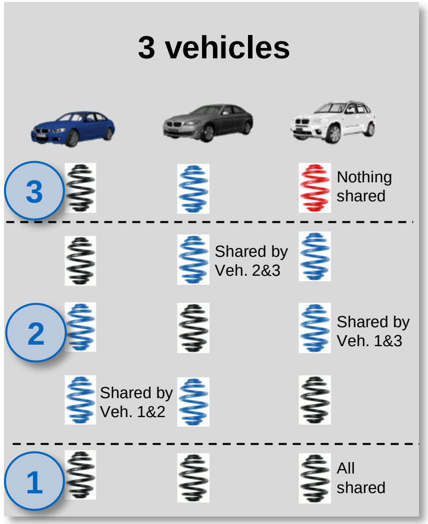
Courtesy of BMW

Example: Variants and Shared Components

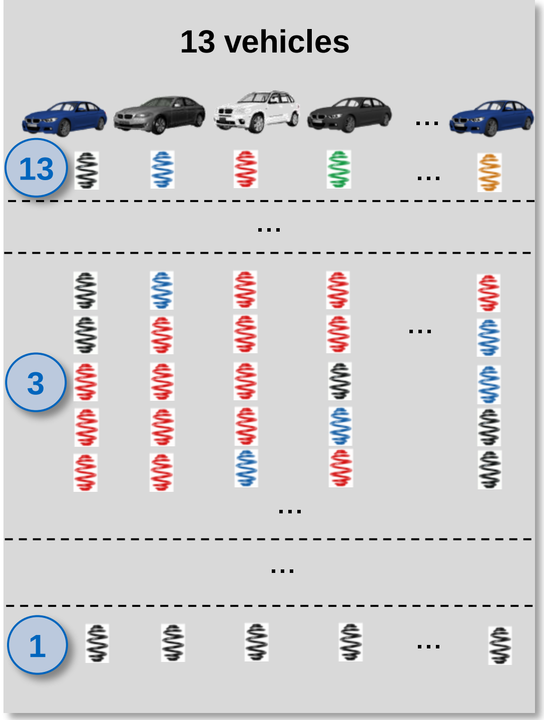




→ **2 configurations**



→ **5 configurations**

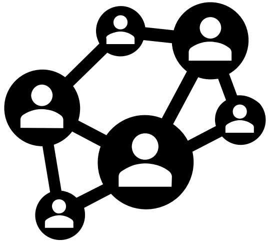


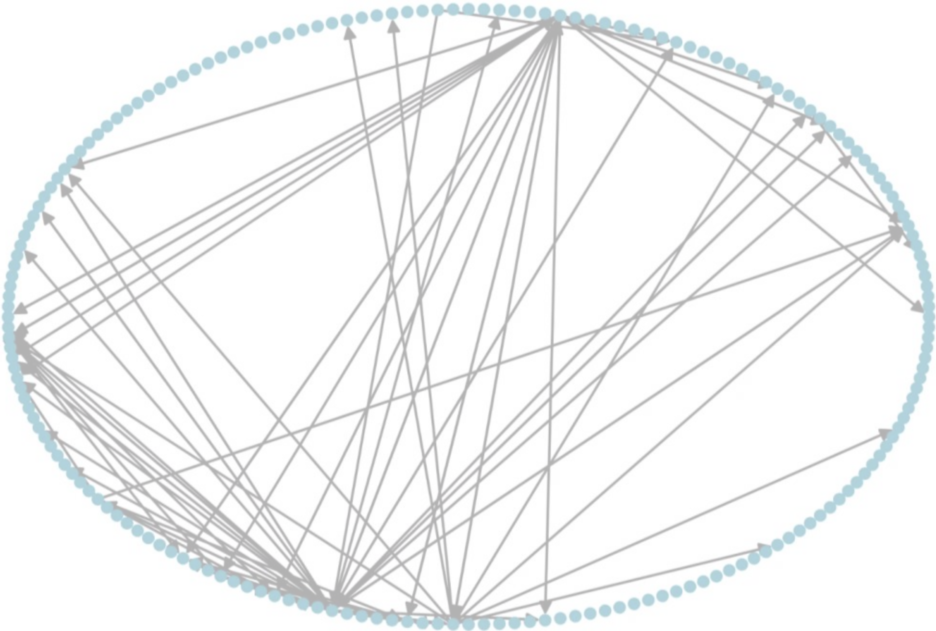
→ **27.6 million configurations!**

Number of different types of parts

- Configurations for 1 part in *n* vehicles: $B_{n+1} = \sum_{i=0}^{n} \binom{n}{i} B_i$

Example: Communication

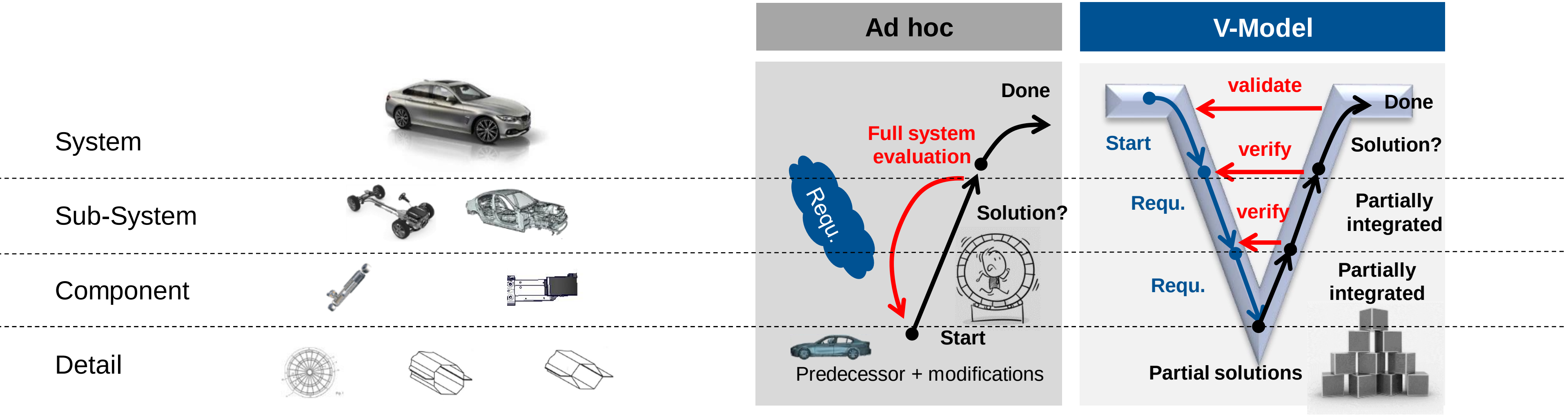




- Development of a Scandinavian biogas powerplant
- 111 stakeholders
- Shown here is monthly email exchange over 5 years

Source: Engineering Systems Division
Technical University of Denmark

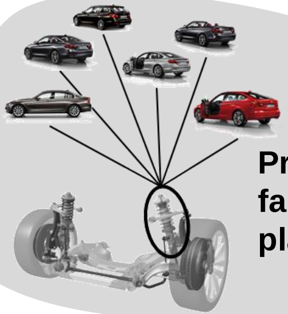
Ingredient 1: V-Model

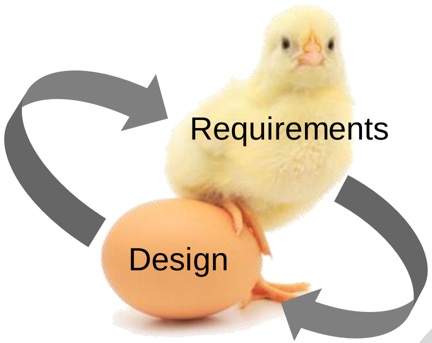


- Ad hoc systems design (typically bottom-up) can be extremely expensive until you get to a satisfactory solution.
- Alternative V-model: Systematic development of requirements → first dependency model by introducing an order.
- **Remaining problem:** How to formulate quantitative requirements that *simultaneously* (1) guarantee that the overall design goal is reached AND (2) provide freedom/can be satisfied? The trouble maker is …

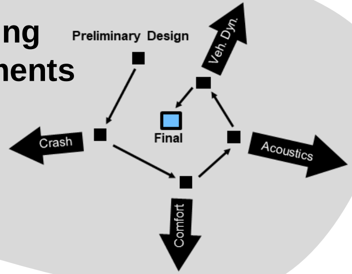
Dilemma of Product Development

Product families & platforms

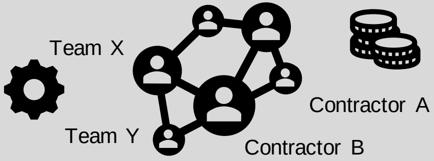




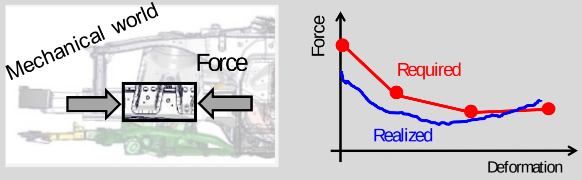
Conflicting requirements



Many parties & interests



Difficulty to realize requirements



- You should know (1) what can be realized (2) other requirements (3) other(4) other products… but you don‘t.

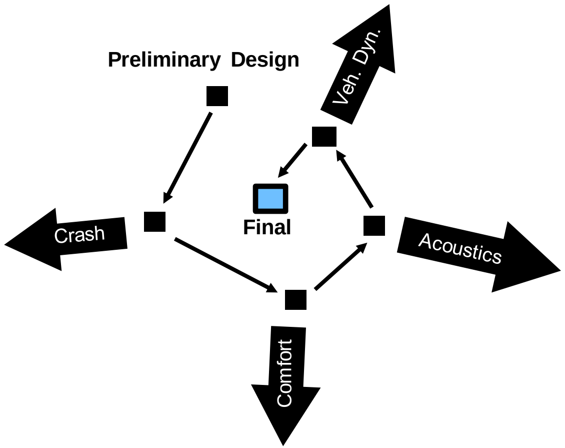
→ uncertainty, complexity, ambiguity … → How to apply the V-model to the mechanical world?

Content

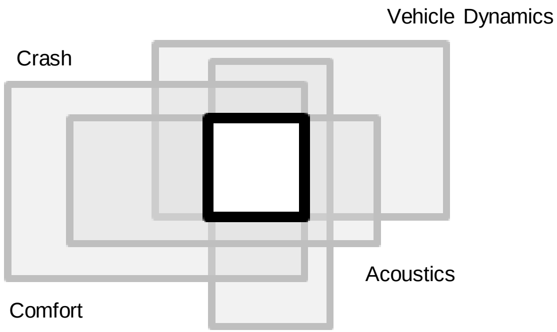
- **Solution Spaces**
- Solution Space Engineering
- Mini Tutorial

Ingredient 2: Solution Spaces

Ad hoc development: **Iteration**

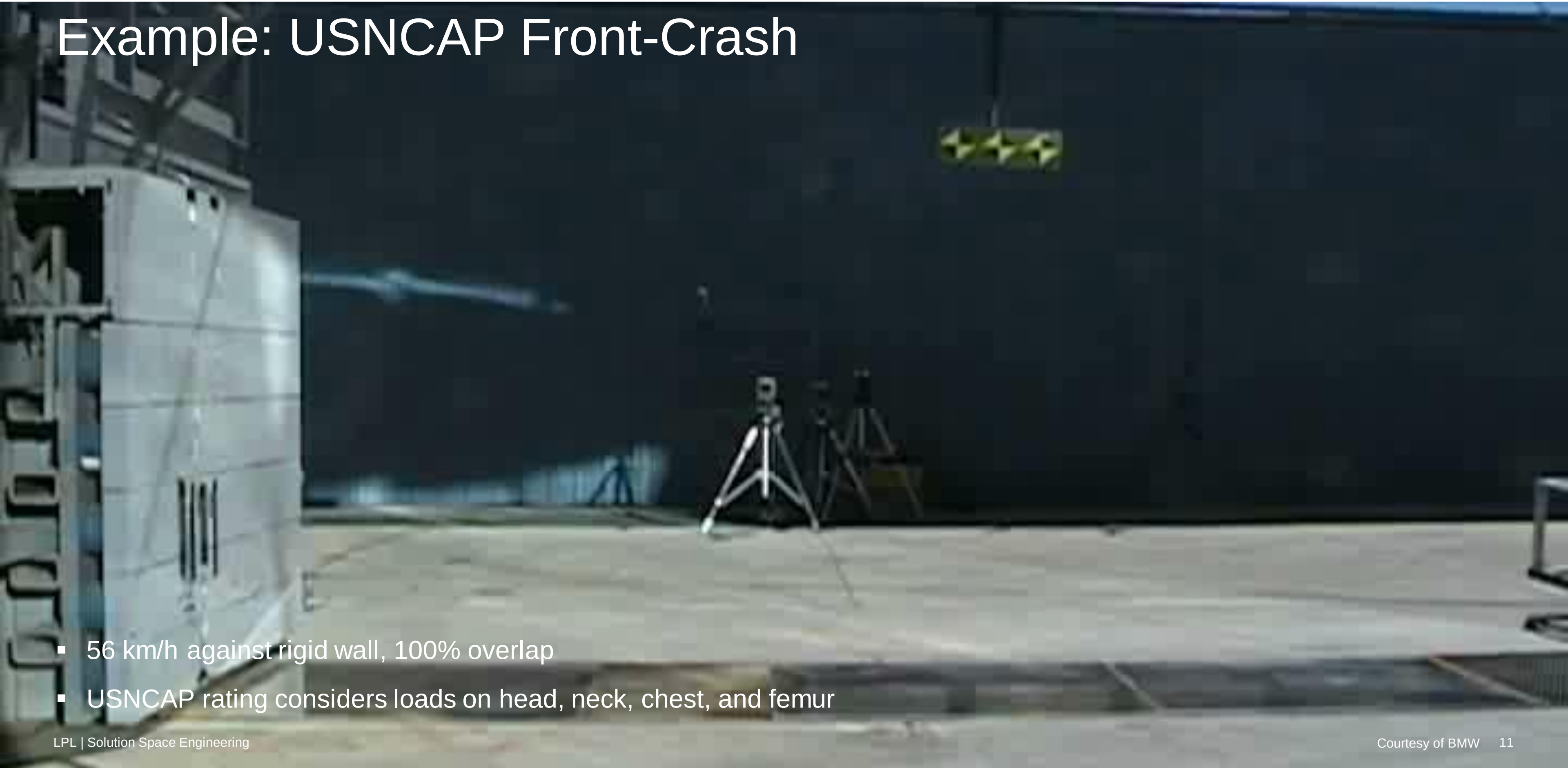


Alternative approach: **Solution Spaces**



- Iterative development with **one design** is prone to conflicts of goals.
- Alternative: **Solution spaces** integrate requirements from different disciplines.

Example: USNCAP Front-Crash

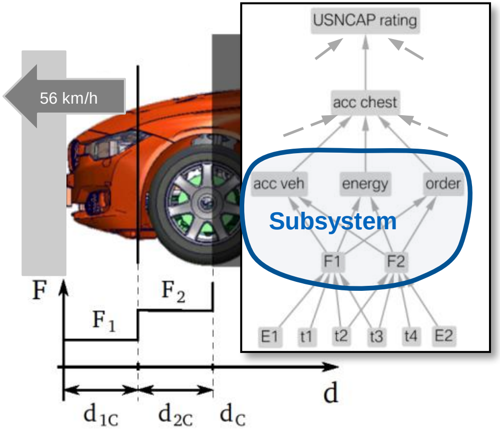


- 56 km/h against rigid wall, 100% overlap
- USNCAP rating considers loads on head, neck, chest, and femur

Courtesy of BMW

Example 1: A Simple Crash Design Problem

Design variables and performance measures



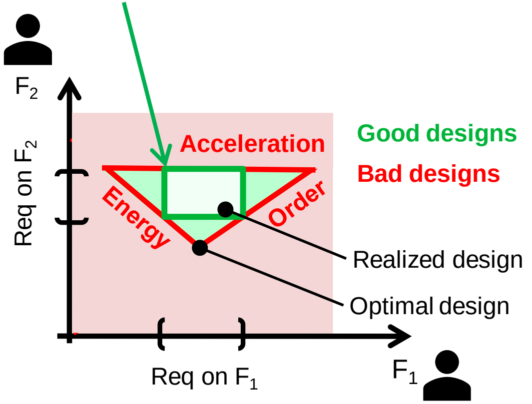
Performance evaluation
$y = f(F_1, F_2)$

Order of deformation
$= F_1 - F_2$ **< 0**

Energy remaining
$= m\,v^2/2 - F_1\,d_1 + F_2\,d_2$ **< 0**

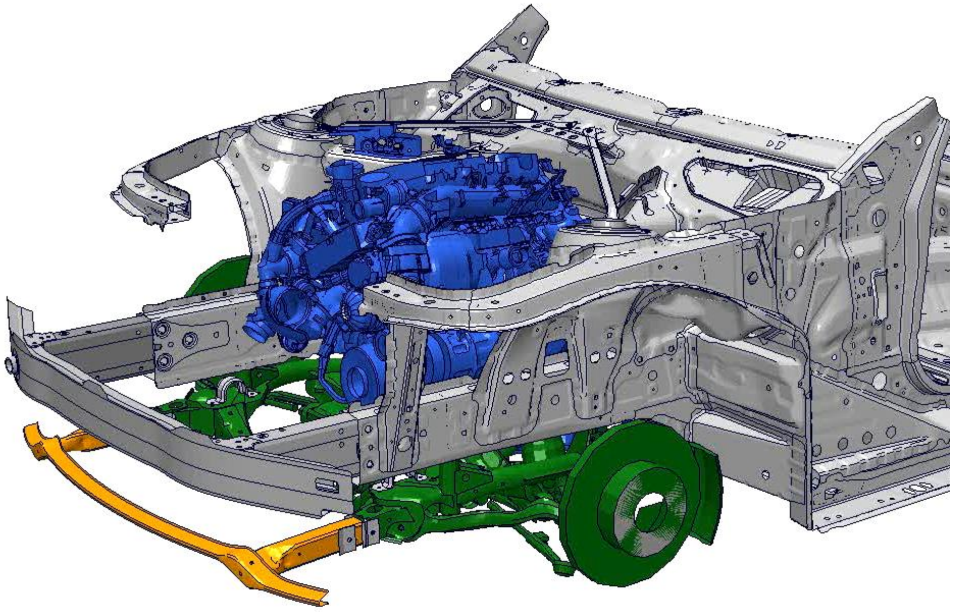
Max. vehicle **Acceleration**
$= F_2 / m$ **$< a_{crit}$**

Solution Space for F_1 & F_2



- Optimal design is **not robust** and **may not be realized**. → Instead: **Maximize the solution space for integrability!**
- Box-shaped solution spaces serve as **requirements** on components and enable parallel & independent design.
- Price to be paid for decoupling: **loss of solution space**.

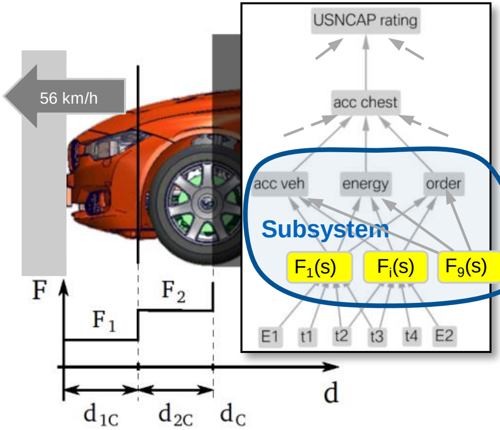
The Real Crash Problem



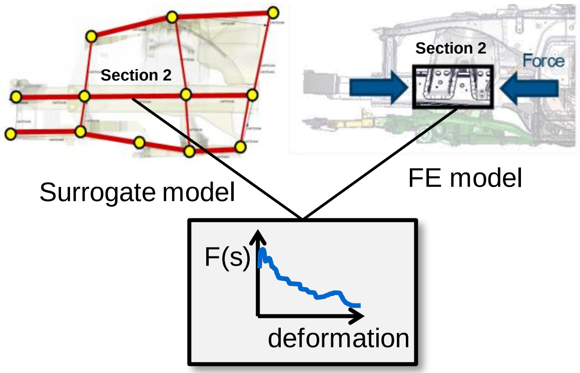
Courtesy of BMW

Example 2: Crash System Design

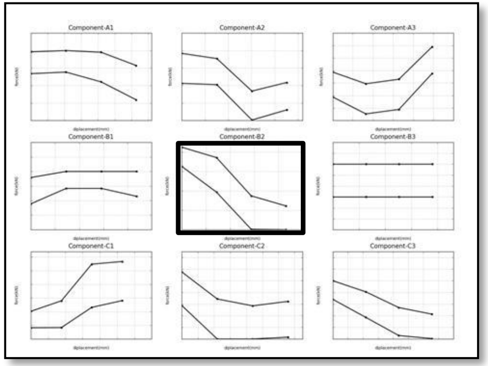
Design variables and performance measures



Performance evaluation
$y = f(F_i(s))$

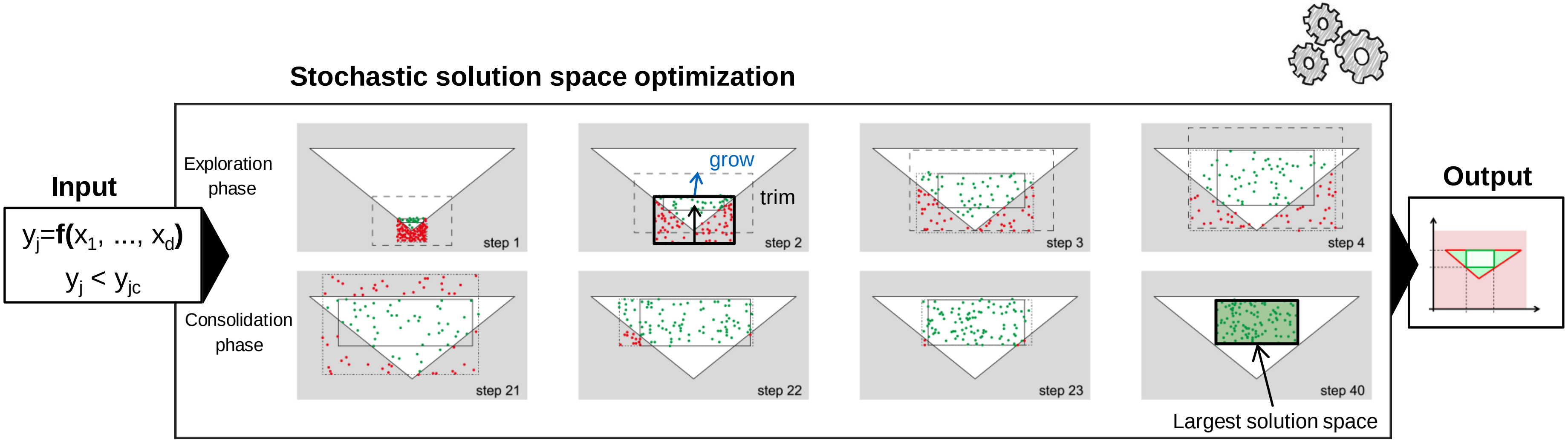


Solution Space for $F_i(s)$



- Performance of real vehicle structure is computed using a physical surrogate model.
- Solution spaces are computed for force-deformation characteristics.
  → How to compute solution spaces in high dimensions?
  → How to use solution spaces for design?

How to Compute Solution Spaces – One Example

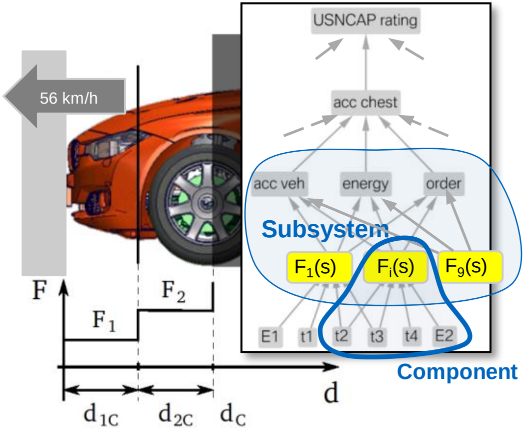


- Algorithm uses iterative stochastic sampling and modification.
- Solves arbitrary high-dimensional and non-linear problems, e.g., 100d crash problem.

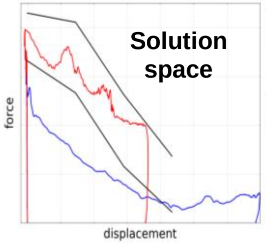
M. Zimmermann, J. Edler von Hoessle: Computing solution spaces for robust design. **International Journal for Numerical Methods in Engineering (2013)**

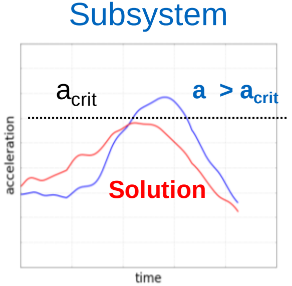
Example 3a: Independent Component Design

Design variables and performance measures

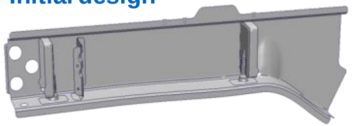


Performance evaluation



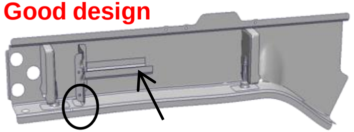


Initial design



Designing a solution

Good design



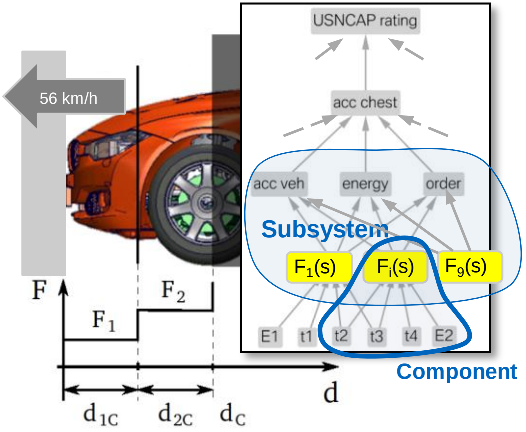
Design measures :
(1) Sheet thickess increased
(2) Extra separation sheet
(3) Notch included

- Independent design towards component requirement → tailored design measures.
- No chicken-or-egg problem.

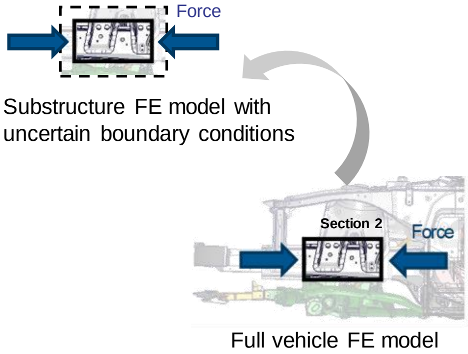
Courtesy of BMW

Example 3b: Independent Component Design – Now Automatic

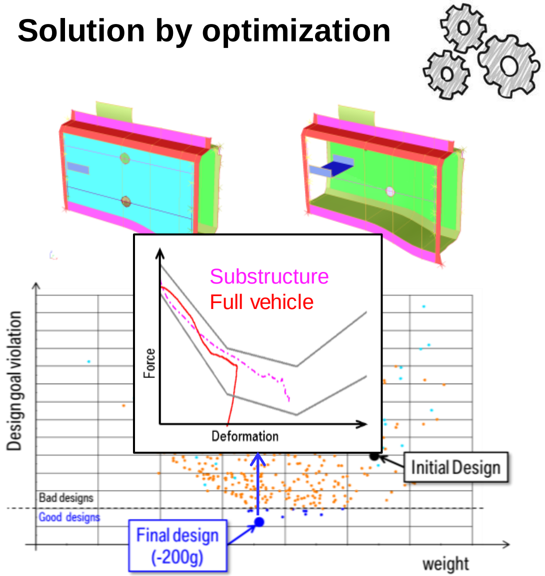
Design variables and performance measures



Performance evaluation



Solution by optimization



- Component design can be be done automatically using parametric optimization.
  → The solution procedures of examples 1, 2 & 3 are similar. How can this be generalized?

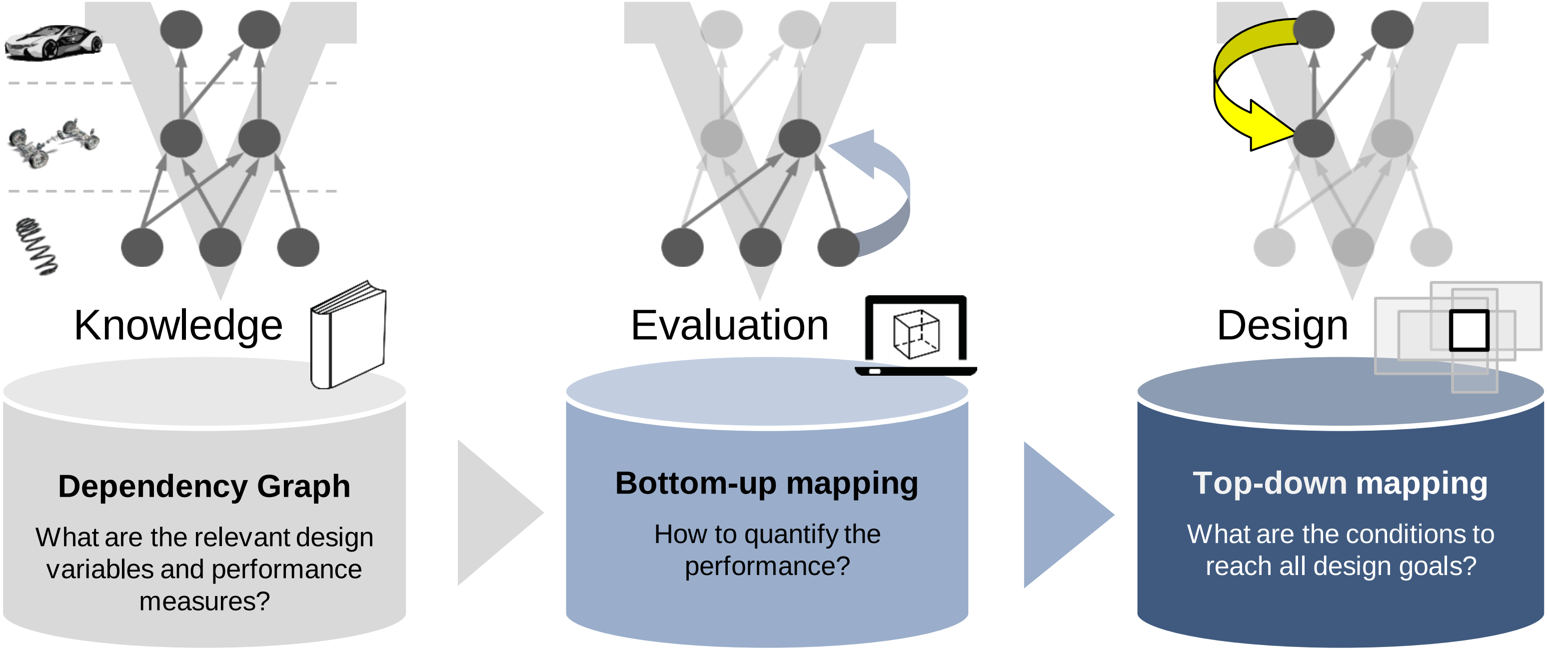
M. Zimmermann, F. Wölfle, H. Zimmer, M. Schäfer, F. Duddeck. Subsystem optimization of the vehicle structure for a frontal crash. **SIMVEC 2012**

Courtesy of BMW

Content

- Solution Spaces
- **Solution Space Engineering**
- Mini Tutorial

Solution Space Engineering



- Solution Space Engineering is a collection of methods and tools for top-down development of complex systems.

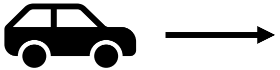
M. Zimmermann, S. Königs, C. Niemeyer, J. Fender, C. Zeherbauer, R. Vitale, M. Wahle. On the design of large systems subject to uncertainty. **Journal of Engineering Design 2017**

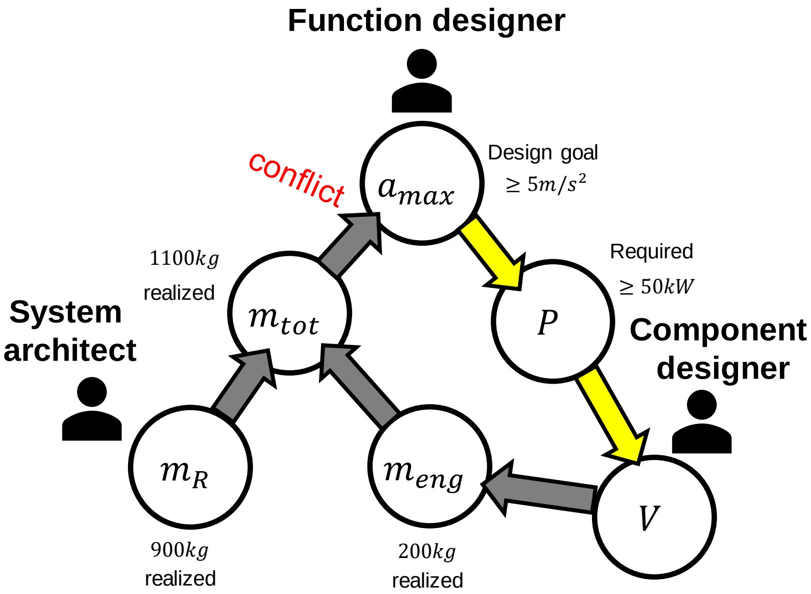
Dependency Graphs

- Complex systems are characterized by a network of dependencies.
- Some dependency models will generate feedback loops.
- Feedback loops make it difficult to find **causes** to problems – where is the root of the cycle?

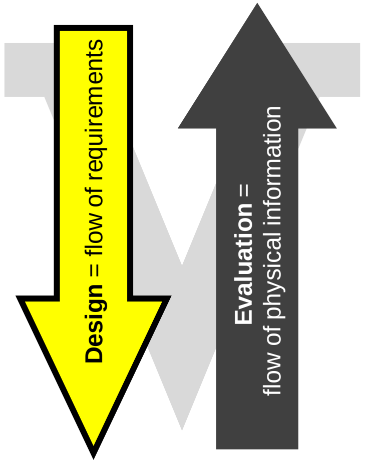
- How to avoid feedback loops?

Example: vehicle accelerating



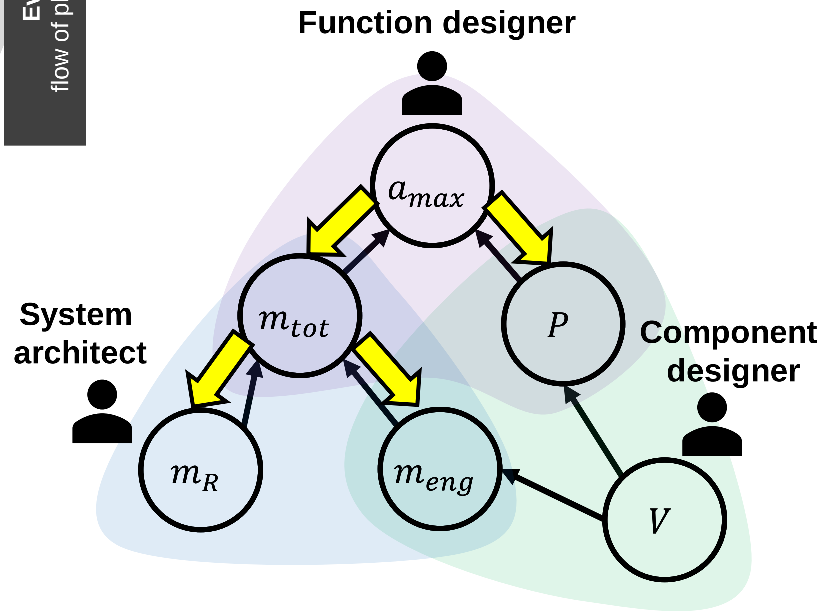


Dependency Graphs



Definition:

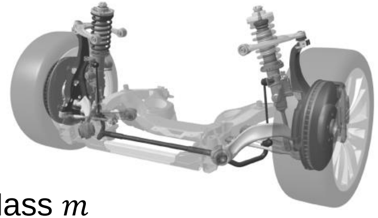
- Dependency graphs model *physical dependencies* between quantifiable properties (design variables, quantities of interest, …) *.*
- They do not know requirements.
  → They sort quantities in the order in which they are measurable.
  → polyhierarchy, **no circular dependencies**



- Requirements are developed going the opposite direction.
- Responsibilites are organized according to dependencies.

Dependency Graphs

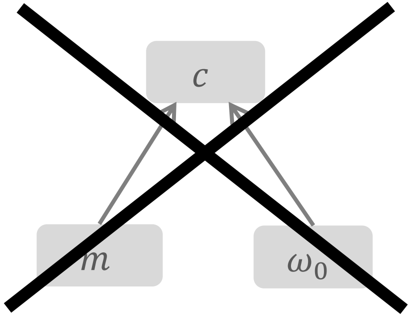
- Simple example: one-mass oscillator.
- What depends on what?
  - Mass and eigenfrequency determine the stiffness?
  - Stiffness and mass determine eigenfrequency?



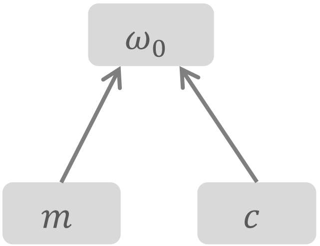
Mass m

Stiffness $c = \omega_0^2\, m$

Eigenfrequency $\omega_0 = \sqrt{c/m}$



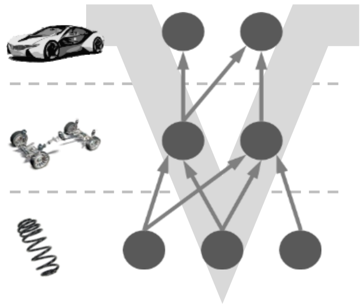
(A)

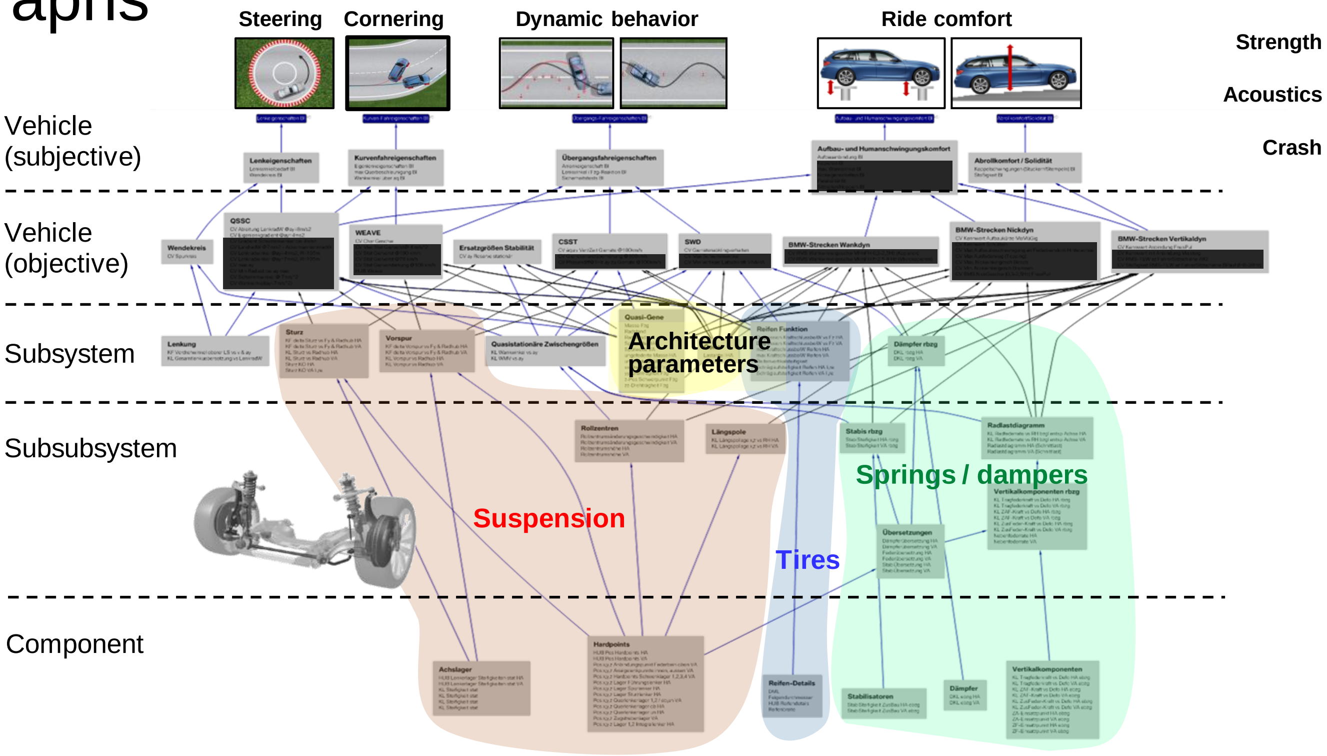


(B)

Courtesy of BMW

Dependency Graphs

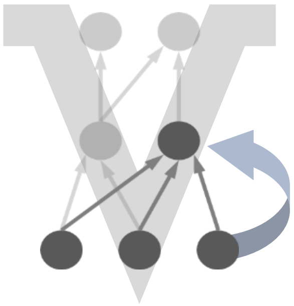


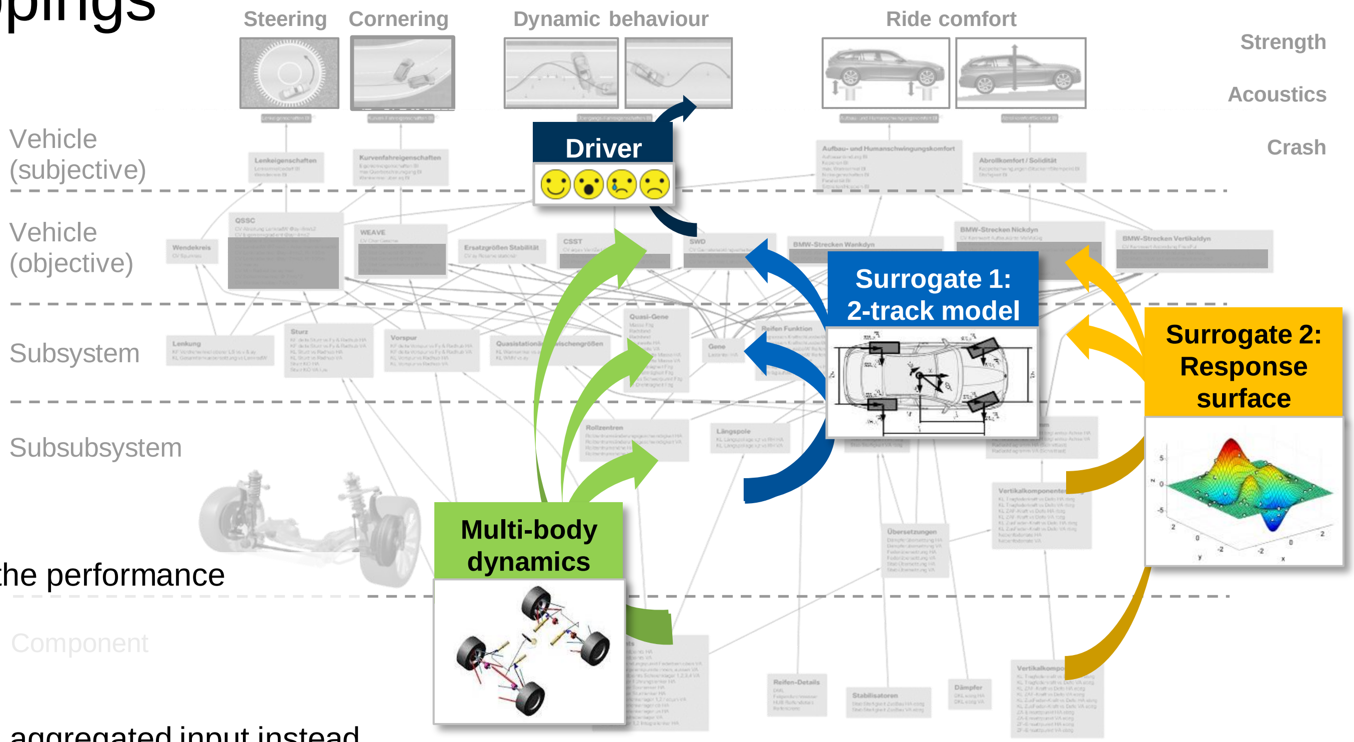


- Connect all Disciplines.
- Organize responsibilities.
- Enable traceable requirement development.

Courtesy of BMW

Bottom-up Mappings

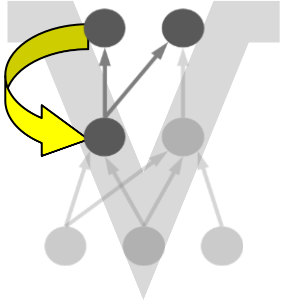


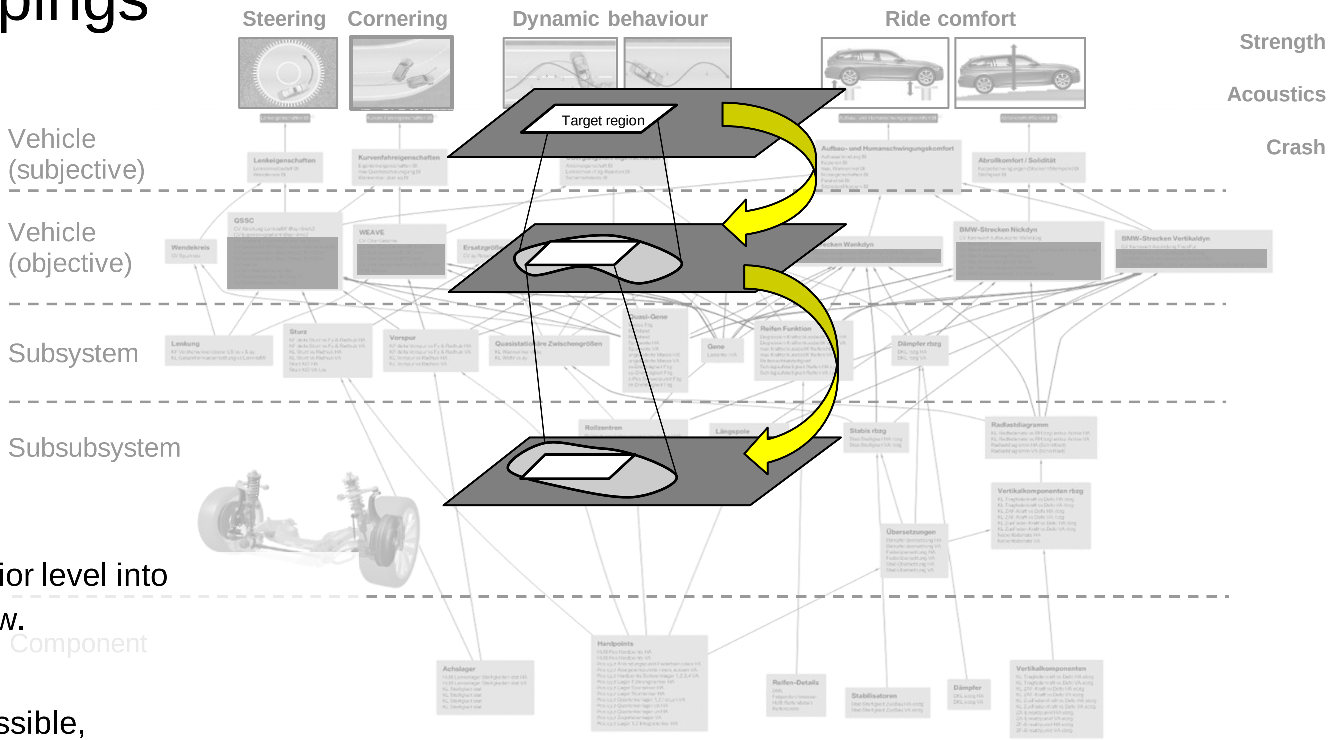


- Function ***y* = *f*(*x*)** to assess the performance of one design.
- Different for each discipline.
- May be surrogates that take aggregated input instead of detailed input. → great for concept development.

Courtesy of BMW

Top-Down Mappings

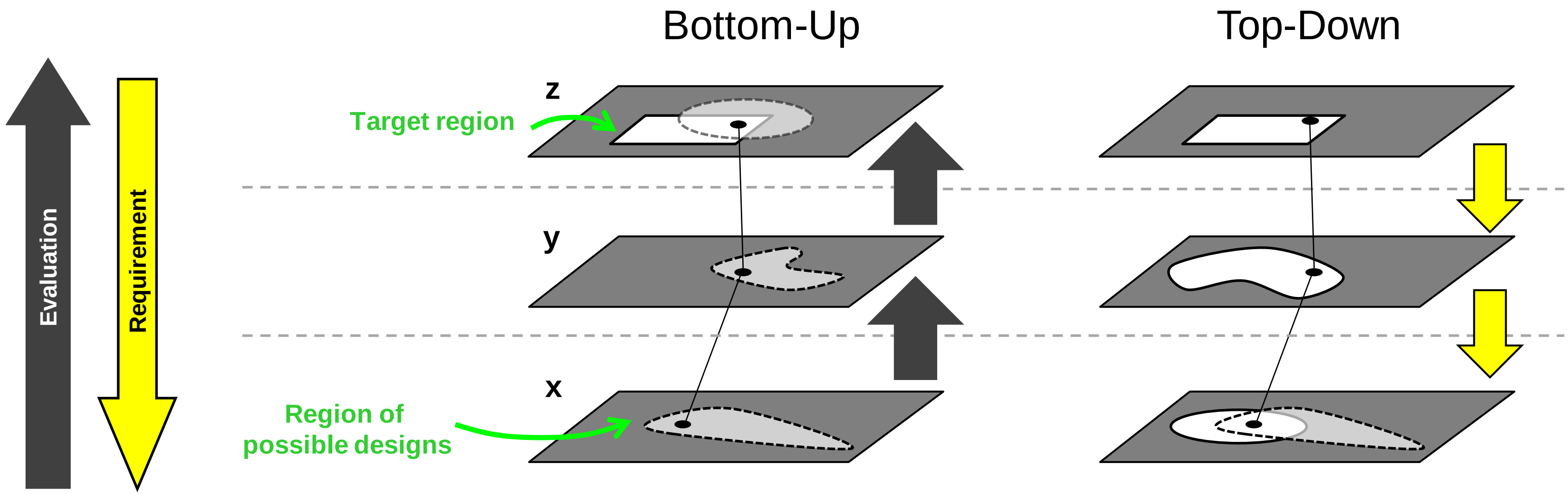




- Turn requirements on superior level into requirements on levels below.
- Should be
  (1) as least restrictive as possible,
  (2) decouple to reduce complexity and
  (3) sufficient for satisfying superior requirements.

Courtesy of BMW

Two Views of Design



- Evaluation-focused: what *is* there?
- Designs are always possible to build (feasible), but are they good?

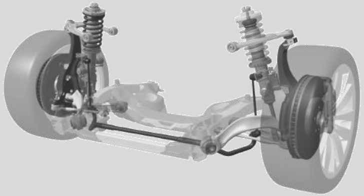
- Requirement-focused: what *should be* there?
- Designs may not be possible to build, but they are always good!

Example 4: Chassis Design for Commonality

13 vehicles

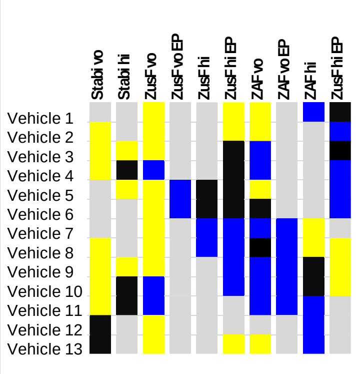
Design problem

2 x Bump stop
Rebound stop
Anti-roll bar



- 10 design variables
- 6 requirements
- **Minimize the number of components!**

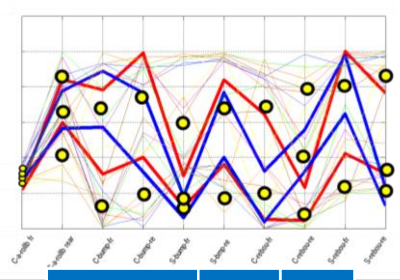
Combinatorics



$(27.6 \times 10^{6})^{10}$ = $\mathbf{2.6 \times 10^{74}}$ possible configurations

→ **New degrees of freedom!**

Result

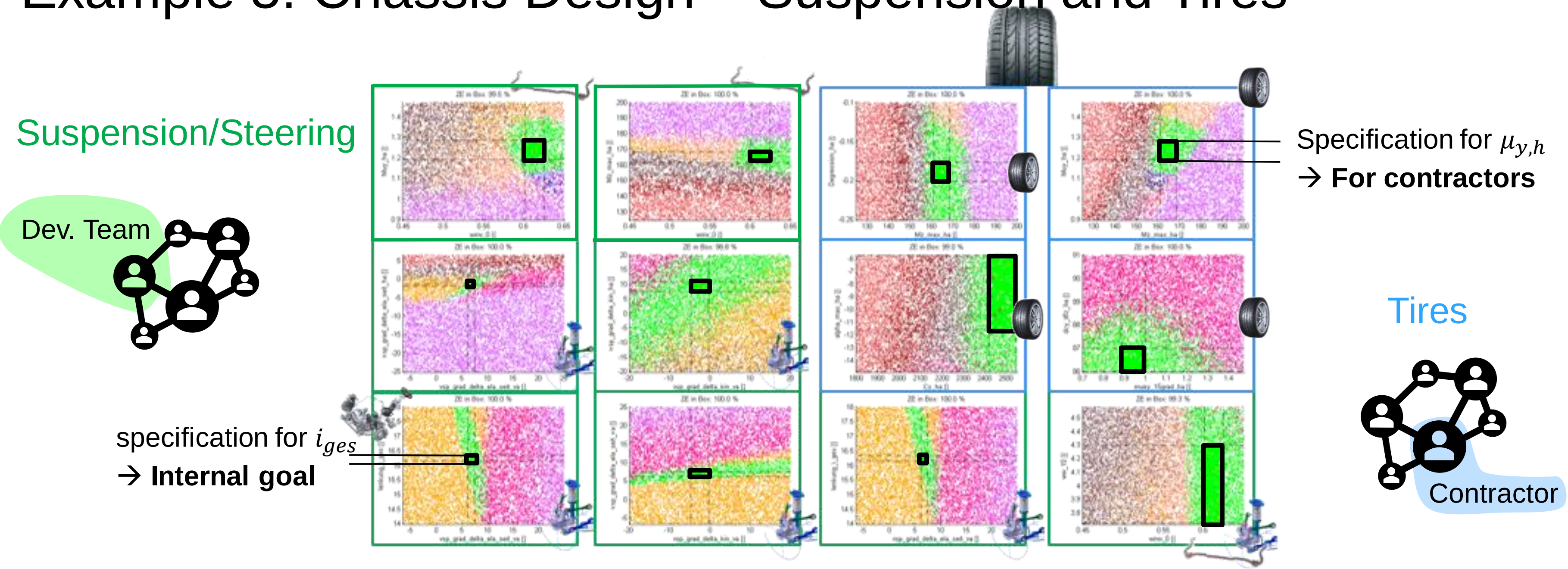


	front	rear
C-anti-roll bar	4	3
C-bump stop	2	2
S-bump stop	3	2
C-rebound	2	3
S-rebound	2	3

- Commonality problem solved using box-shaped solution spaces.
- Result is an important improvement, but probably not globally optimal due to loss of solution space.

M. Eichstetter, S. Müller, M. Zimmermann. Product Family Design Using Solution Spaces. **Journal of Mechanical Design 2015**

Example 5: Chassis Design – Suspension and Tires



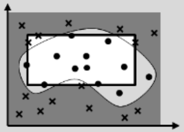
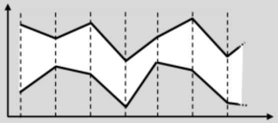
- Requirements on mass, geometrical dimensions, tires, suspension and steering are quantified and passed on to development teams and contractors.
- Solution spaces were constructed using **Selective Design Space Projection → mini tutorial.**

Content

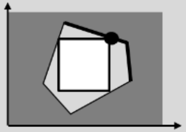
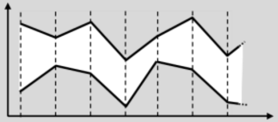
- Solution Spaces
- Solution Space Engineering
- **Mini Tutorial**

Top-down Mappings – Solution Techniques

- **Stochastic iteration**
  - For non-linear, high-dimensional noisy problems
  - Robust (but limited accuracy)

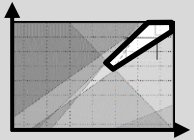
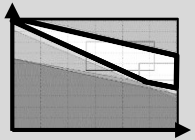



- **Corner tracking**
  - For monotonous performance functions
  - Exact (but limited applicability)

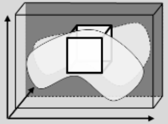
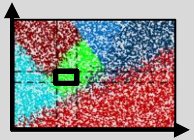
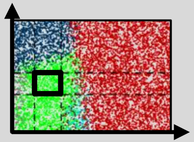



- **Advanced: p-dim. decomposition**
  - For strong interactions between design variables
  - Reduces loss of solution space

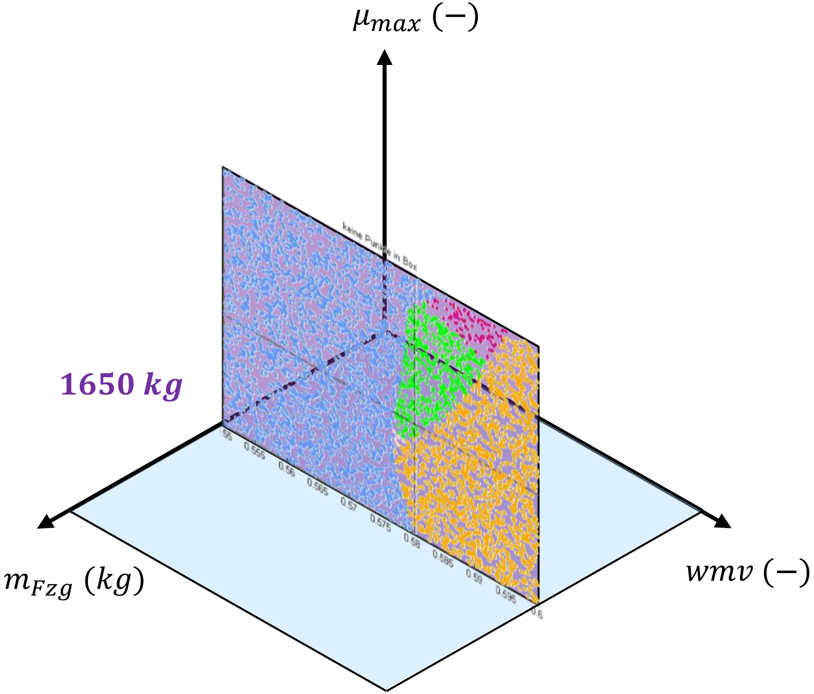
$$f(x,y) = f(x) + f(y) + \varepsilon$$

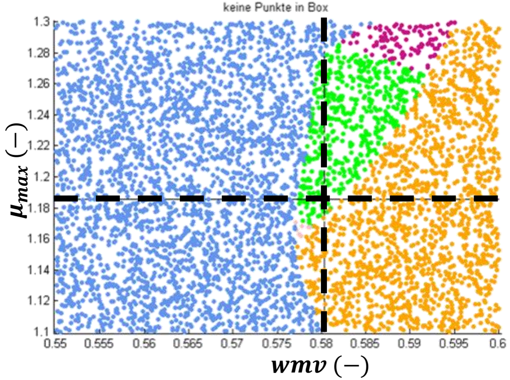



Selective design space projection
- Projects slabs of design space onto 2d-diagrams
- Intuitive (but not automatic)

Section I

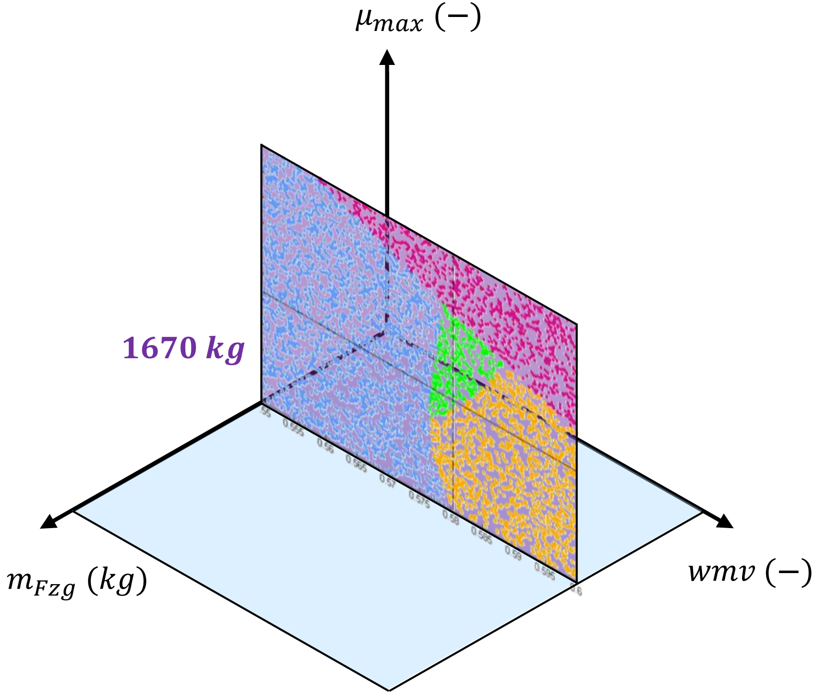


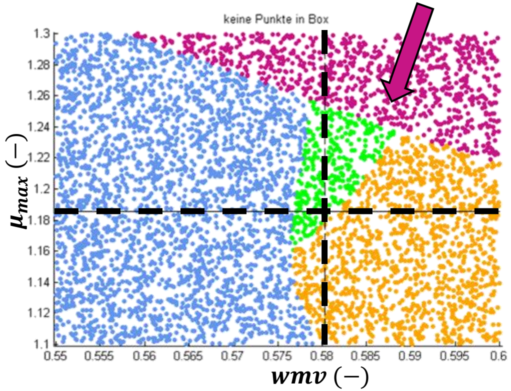


- CV 073 slip angle amplification too large
- CV 171 stability reserve too little
- CV 077 max ay too little
- CV 060 side wind sensitivity too large
- CV 065 steering angle @ 7 m/s² too large

Courtesy of BMW

Section II



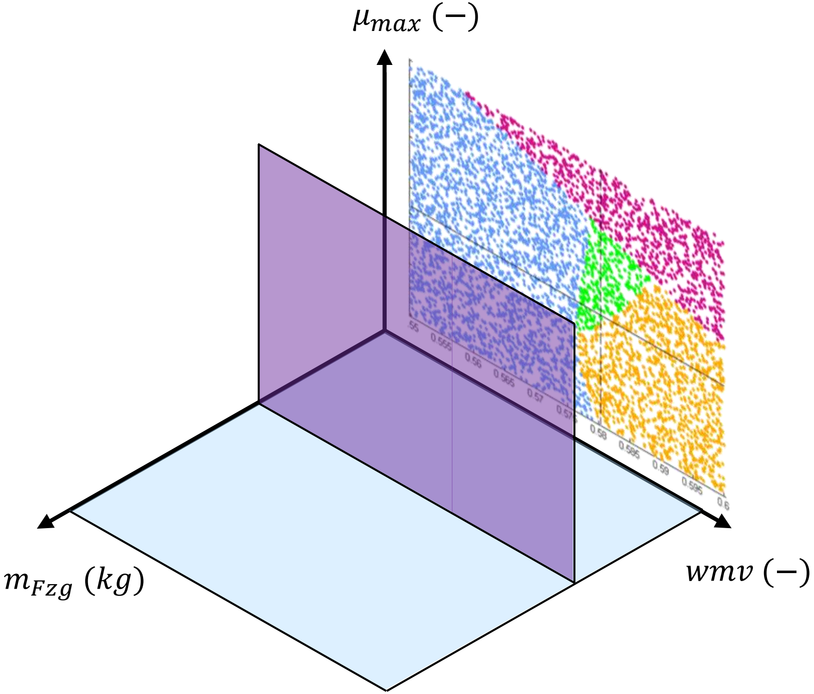


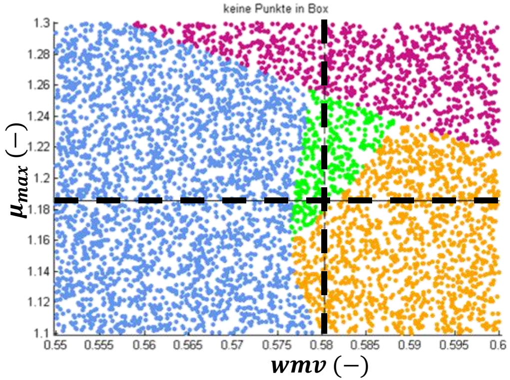
μ_{max}: 1.1 – 1.3
wmv: 0.55 – 0.6
m_{Fzg}: 1550 – 1750

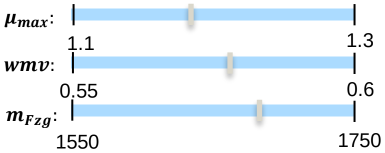
- CV 073 slip angle amplification too large
- CV 171 stability reserve too little
- CV 077 max ay too little
- CV 060 side wind sensitivity too large
- CV 065 steering angle @ 7 m/s² too large

Courtesy of BMW

Section I Projected

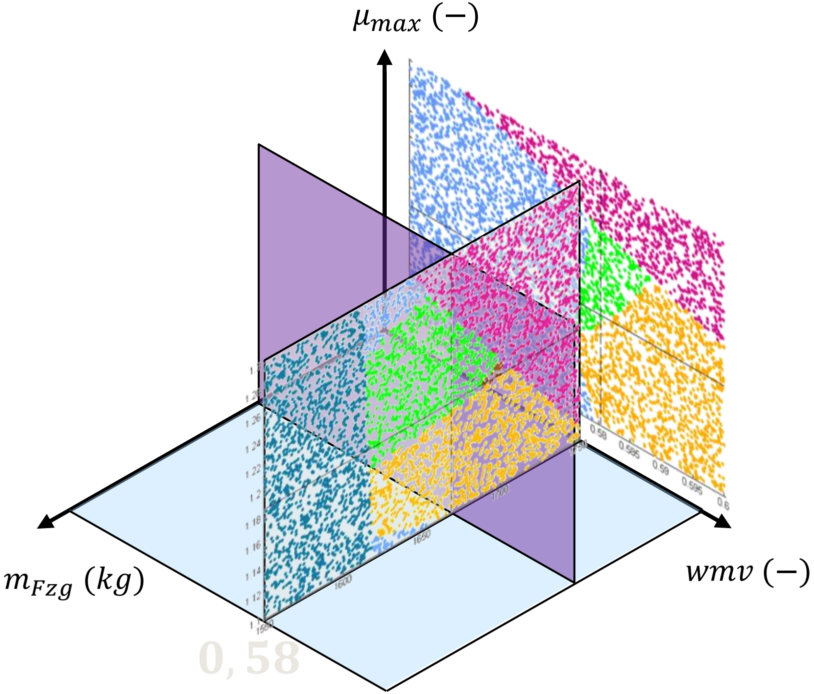


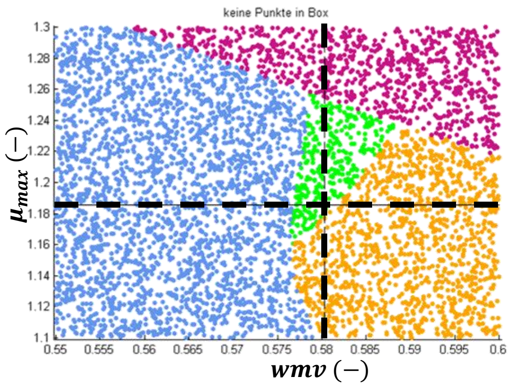


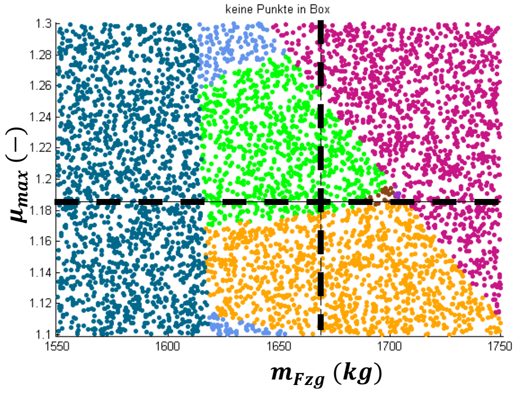


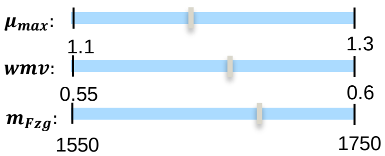
- CV 073 slip angle amplification too large
- CV 171 stability reserve too little
- CV 077 max ay too little
- CV 060 side wind sensitivity too large
- CV 065 steering angle @ 7 m/s² too large

Section III



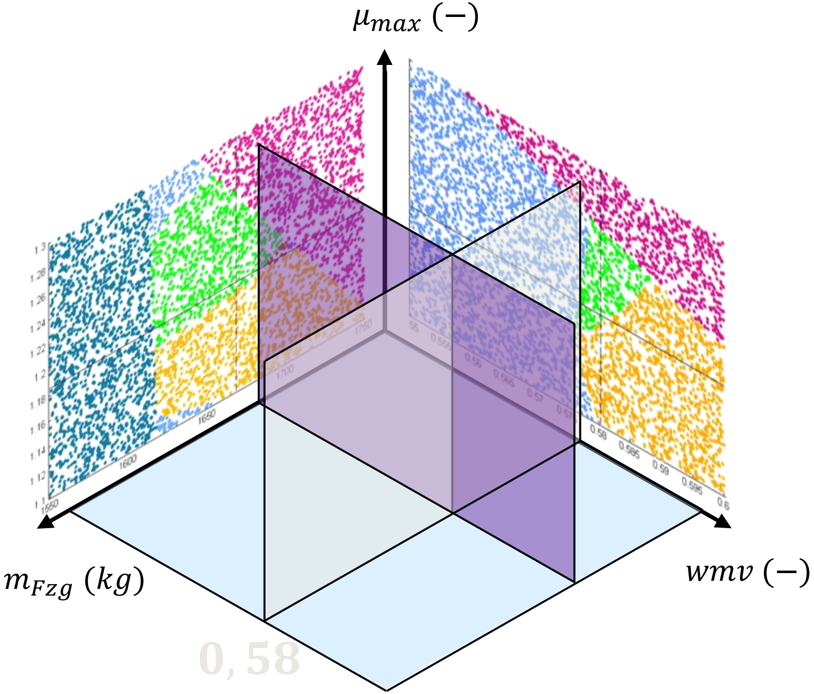


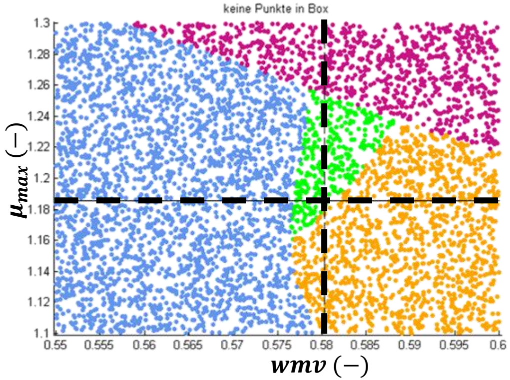


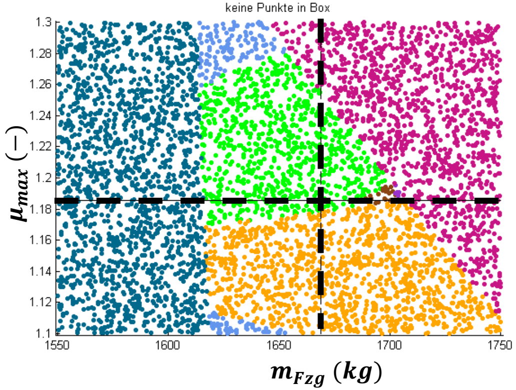


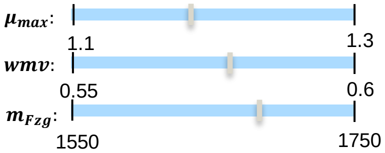
- CV 073 slip angle amplification too large
- CV 171 stability reserve too little
- CV 077 max ay too little
- CV 060 side wind sensitivity too large
- CV 065 steering angle @ 7 m/s² too large

Section III Projected





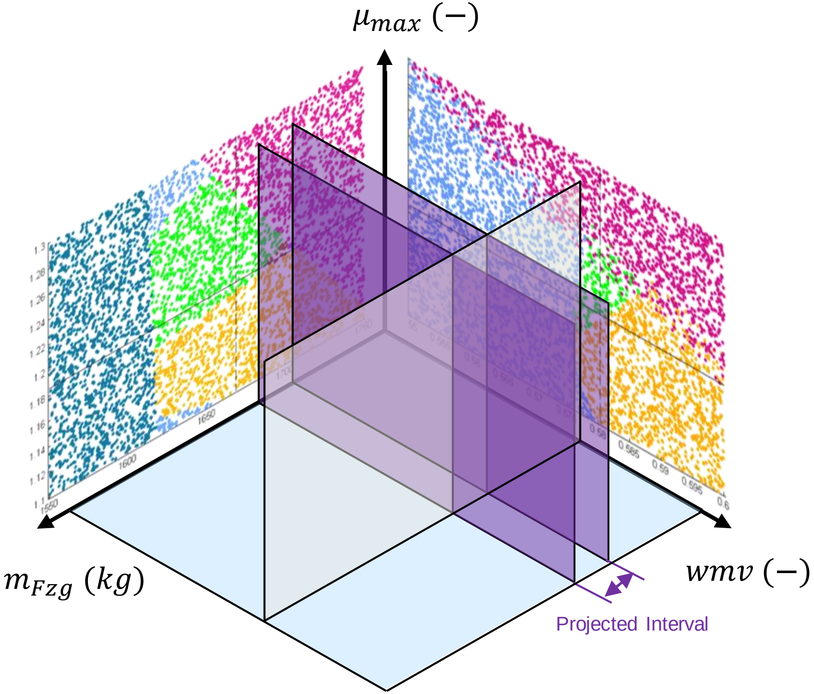


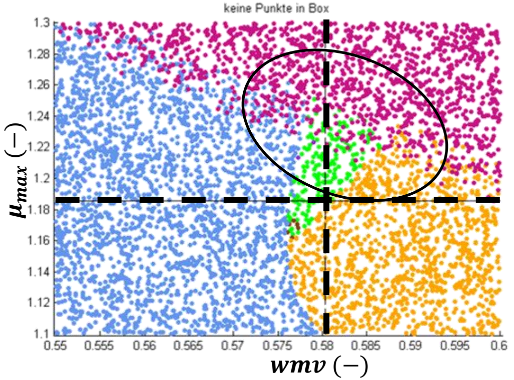


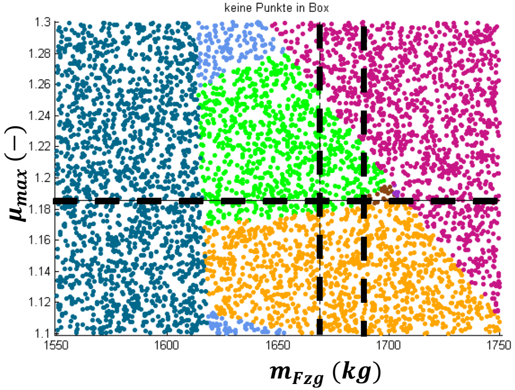
- CV 073 slip angle amplification too large
- CV 171 stability reserve too little
- CV 077 max ay too little
- CV 060 side wind sensitivity too large
- CV 065 steering angle @ 7 m/s² too large

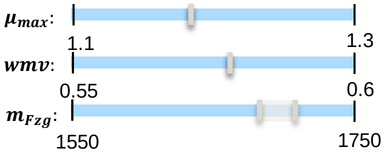
Courtesy of BMW

One Section and One Projected Slab





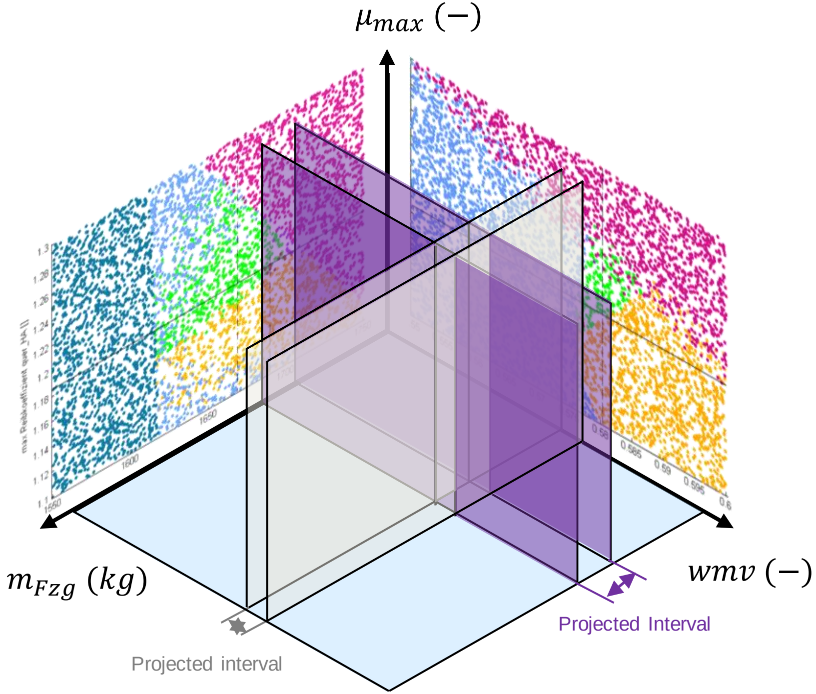


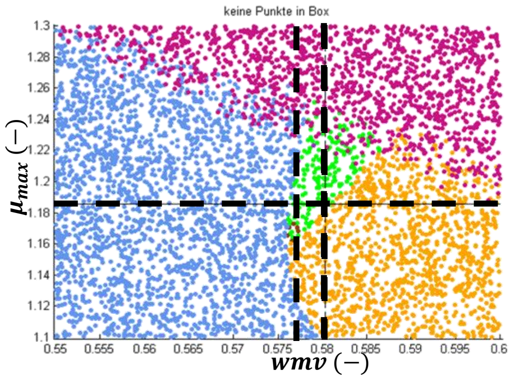


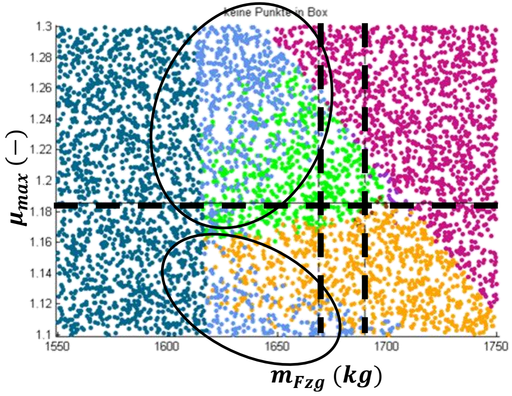
- CV 073 slip angle amplification too large
- CV 171 stability reserve too little
- CV 077 max ay too little
- CV 060 side wind sensitivity too large
- CV 065 steering angle @ 7 m/s² too large

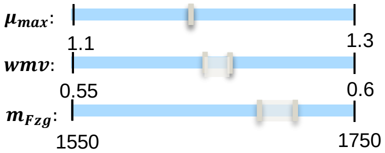
Courtesy of BMW

Two Projected Slabs



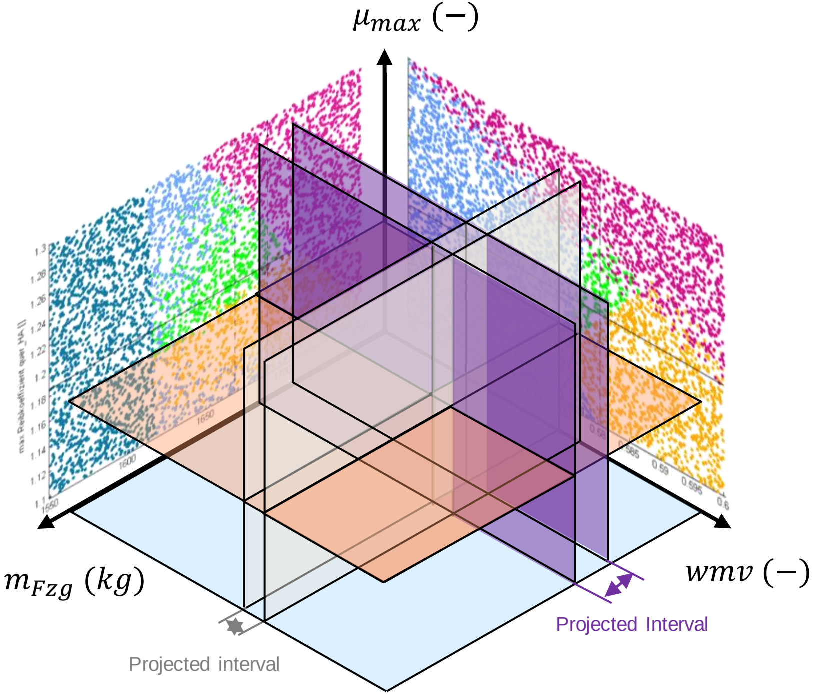


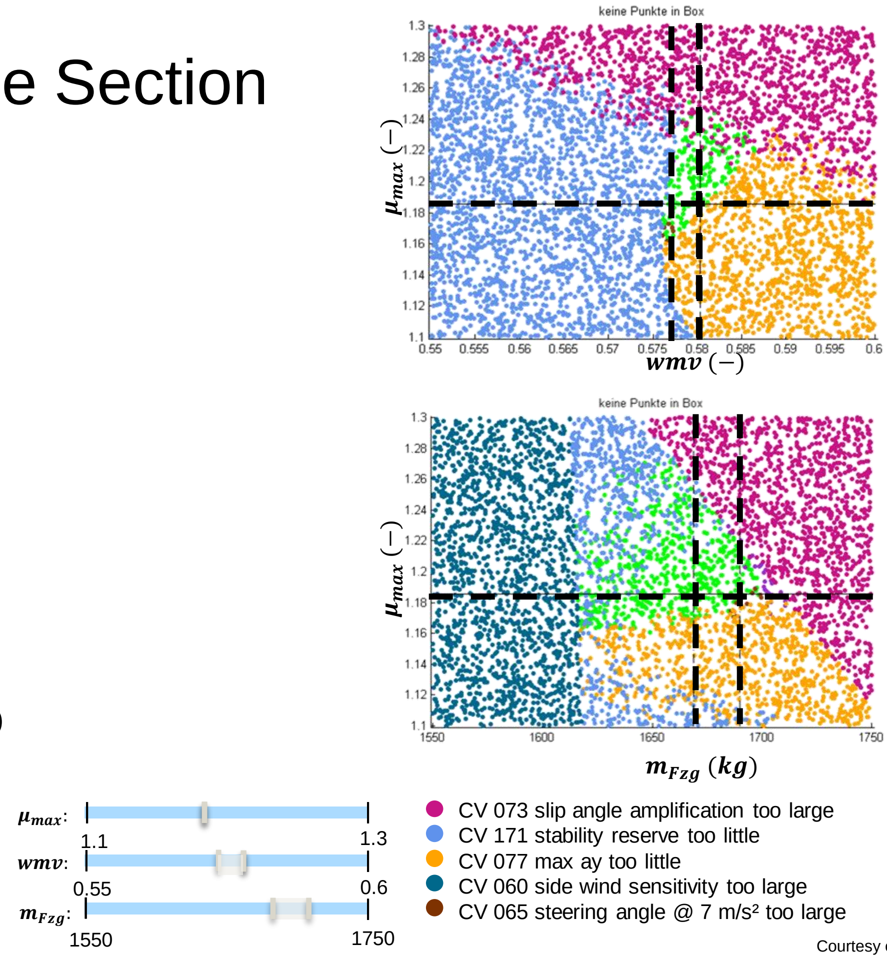




- CV 073 slip angle amplification too large
- CV 171 stability reserve too little
- CV 077 max ay too little
- CV 060 side wind sensitivity too large
- CV 065 steering angle @ 7 m/s² too large

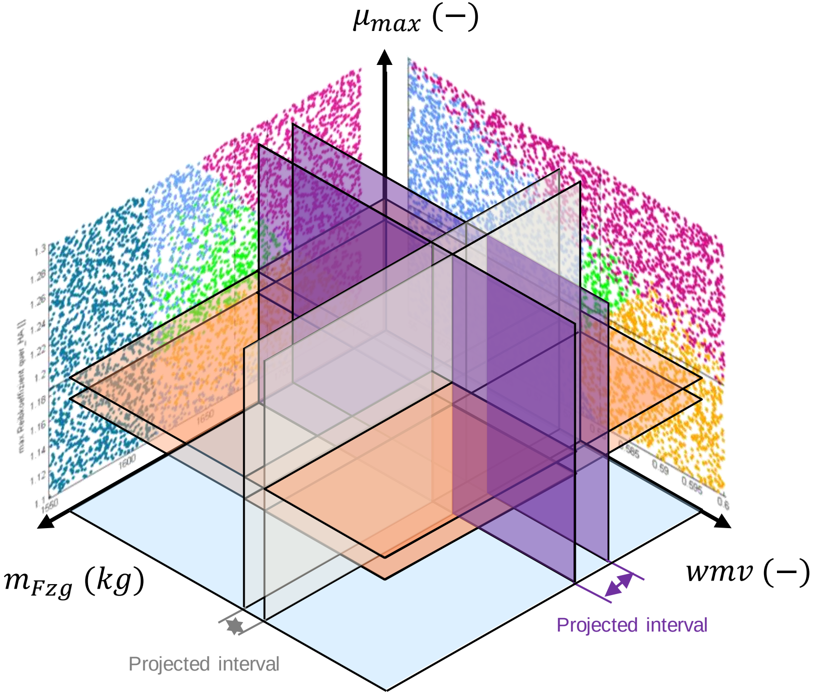
Two Projected Slab and One Section

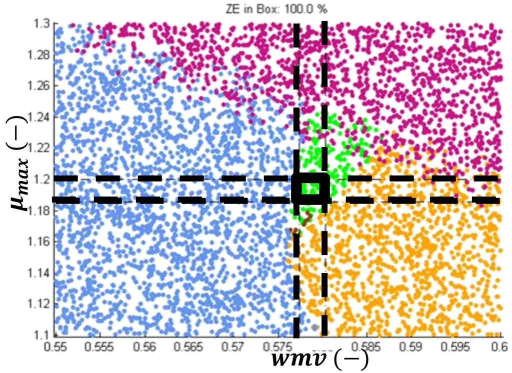


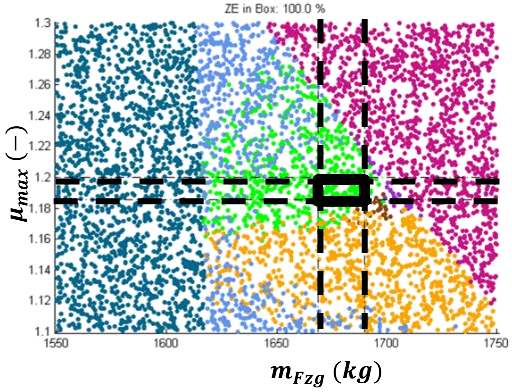


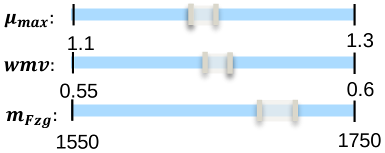
Courtesy of BMW

Three Slabs



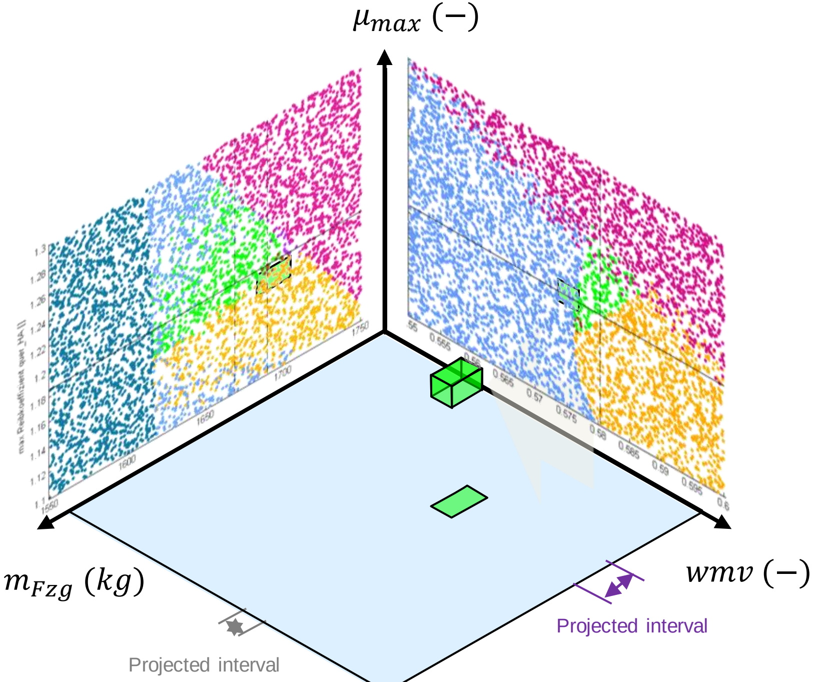


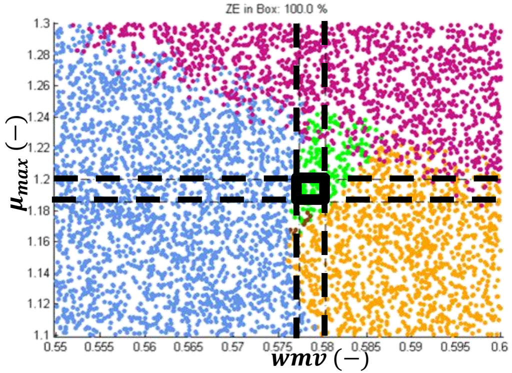


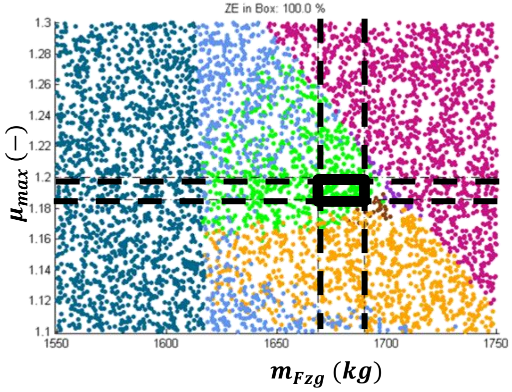


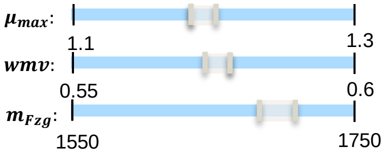
- CV 073 slip angle amplification too large
- CV 171 stability reserve too little
- CV 077 max ay too little
- CV 060 side wind sensitivity too large
- CV 065 steering angle @ 7 m/s² too large

Solution Space









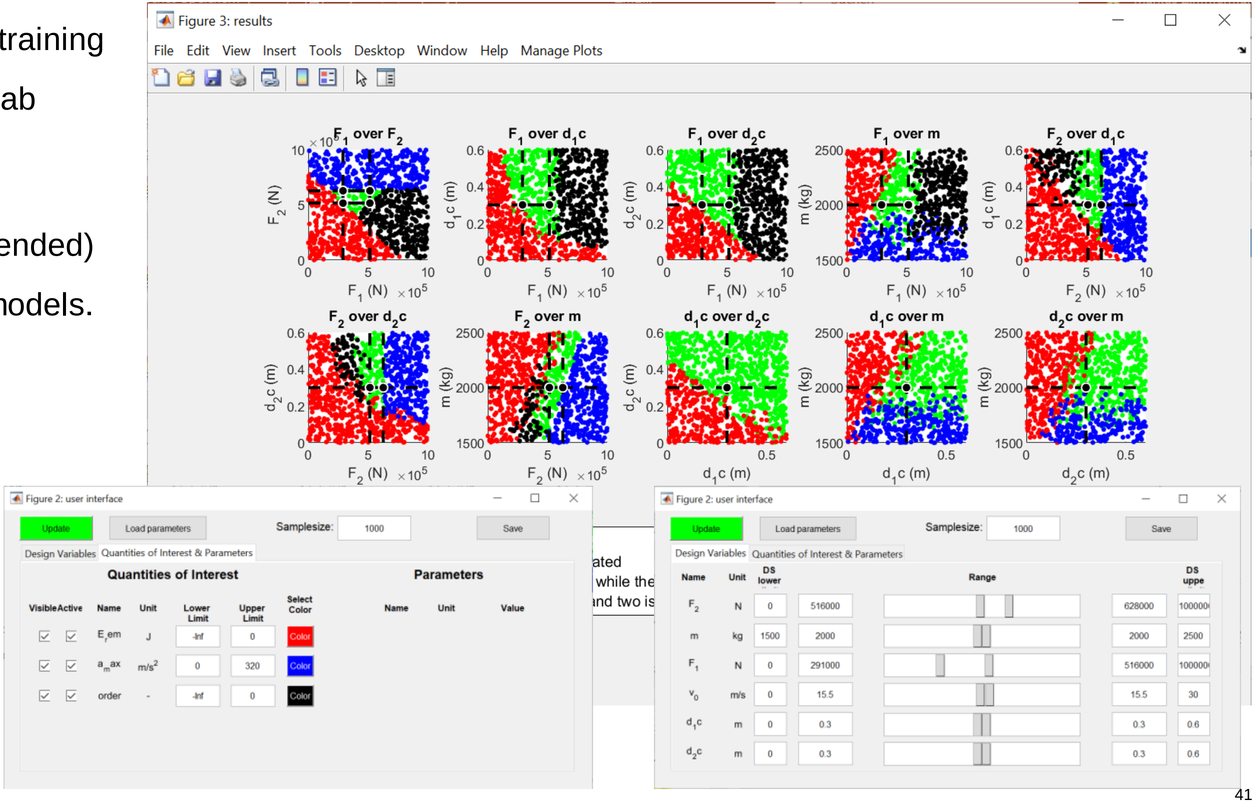
- CV 073 slip angle amplification too large
- CV 171 stability reserve too little
- CV 077 max ay too little
- CV 060 side wind sensitivity too large
- CV 065 steering angle @ 7 m/s² too large

Courtesy of BMW

Basic X-ray Tool v11

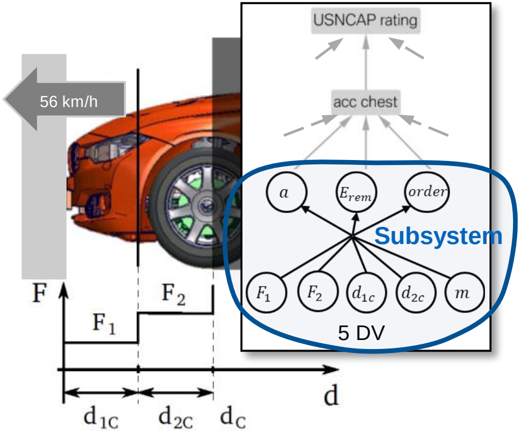
Download at https://www.mw.tum.de/lpl/tools/basic-x-ray-tool/

- Public open-source matlab tool for training
- **Plug and play** – if you have a Matlab license
- Included: bottom-up mappings of Simple Crash Design Problem (extended)
- Easily extendible by other matlab models.



Example 1 *extended*: Simple Crash Design Problem

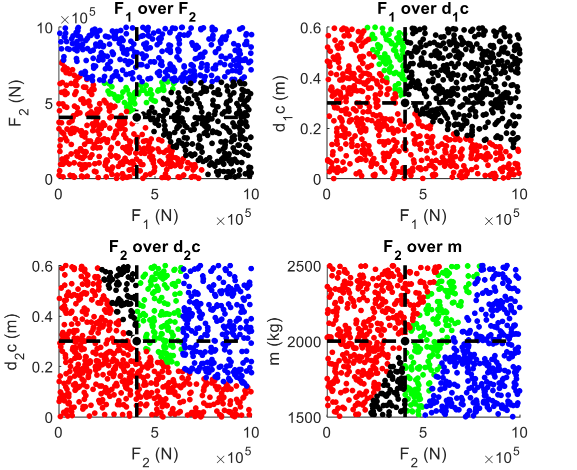
Design variables and performance measures



Performance evaluation
$y = f(F_1, F_2)$

Order of deformation $= \mathbf{F_1 - F_2}$	**< 0**
Energy remaining $= \mathbf{m}\, v^2/2 - \mathbf{F_1\, d_{1c}} + \mathbf{F_2\, d_{2c}}$	**< 0**
Max. vehicle **Acceleration** $= \mathbf{F_2} / \mathbf{m}$	**< a_{crit}**

Solution Space for F_1, F_2, m , d_{1c}, d_{2c}



- Extension by mass and geometry – how to design geometry, mass and body parts simultaneously?
- Design problem available in basic x-ray tool v11.

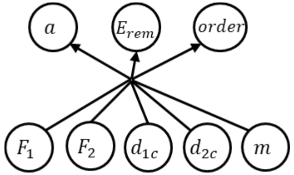
→ **Try to design your own crash structure!**

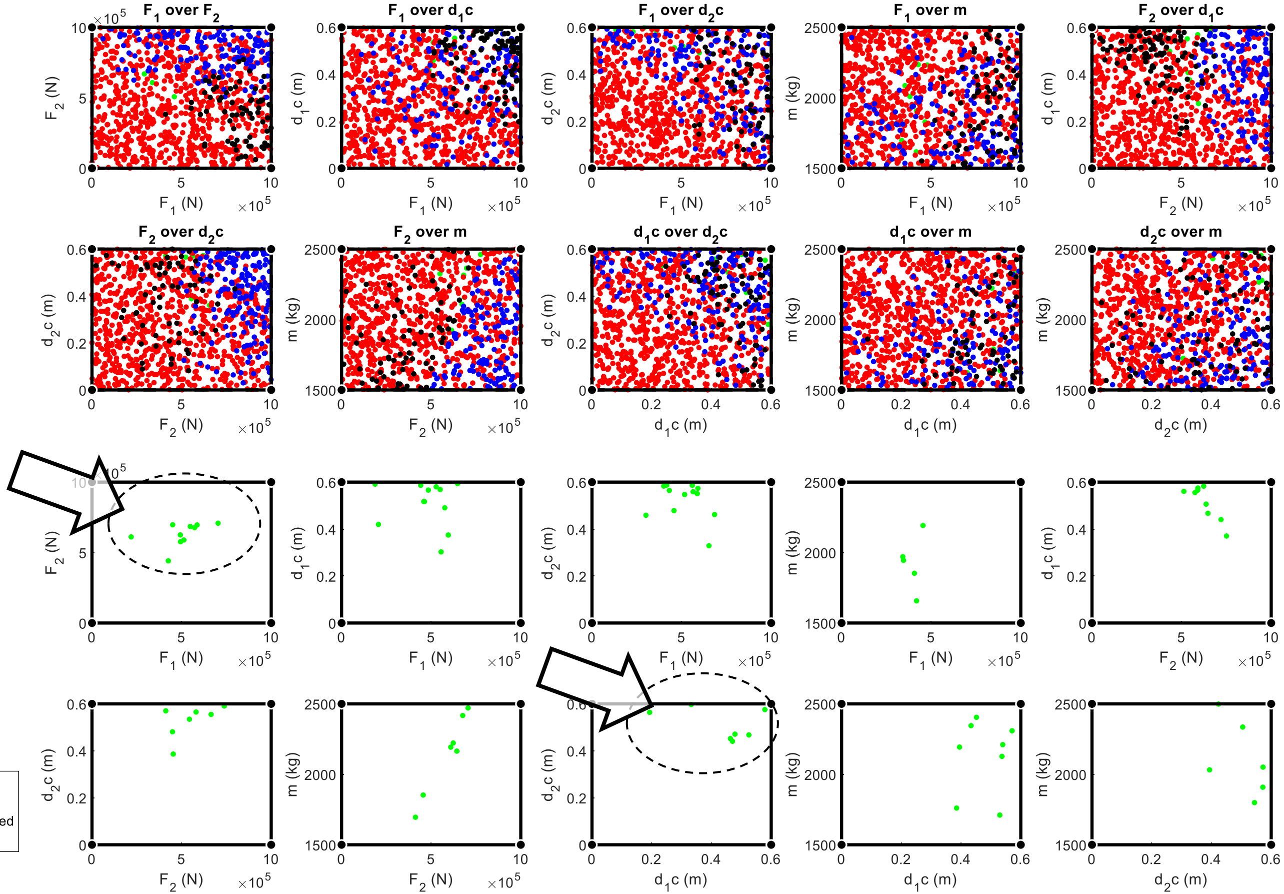
Courtesy of BMW

View 1

Entire design space

- Show only good designs.

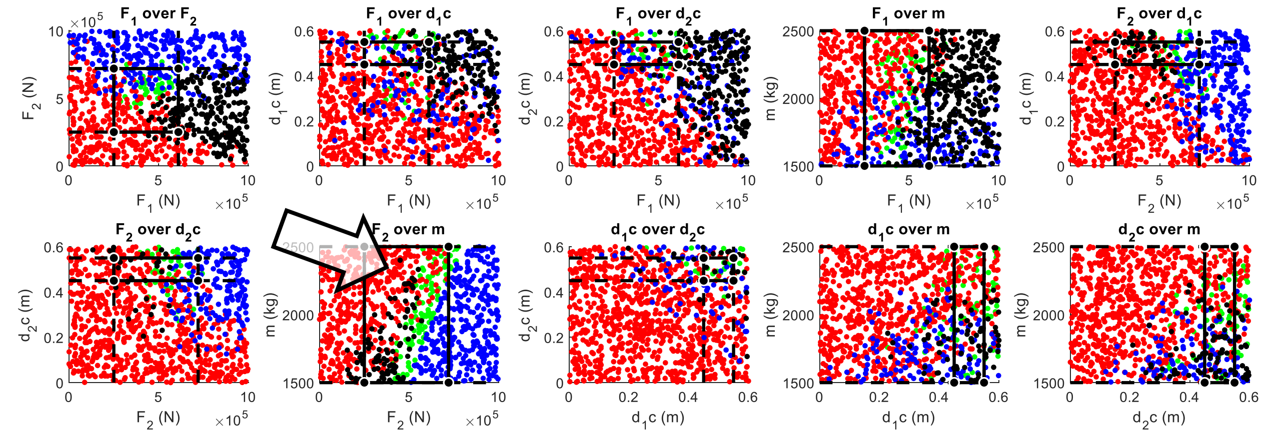




- Good design
- Remaining energy after the crash is violated
- Maximal negative acceleration occuring while the crash is violated
- Order of the deformation of sector one and two is violated

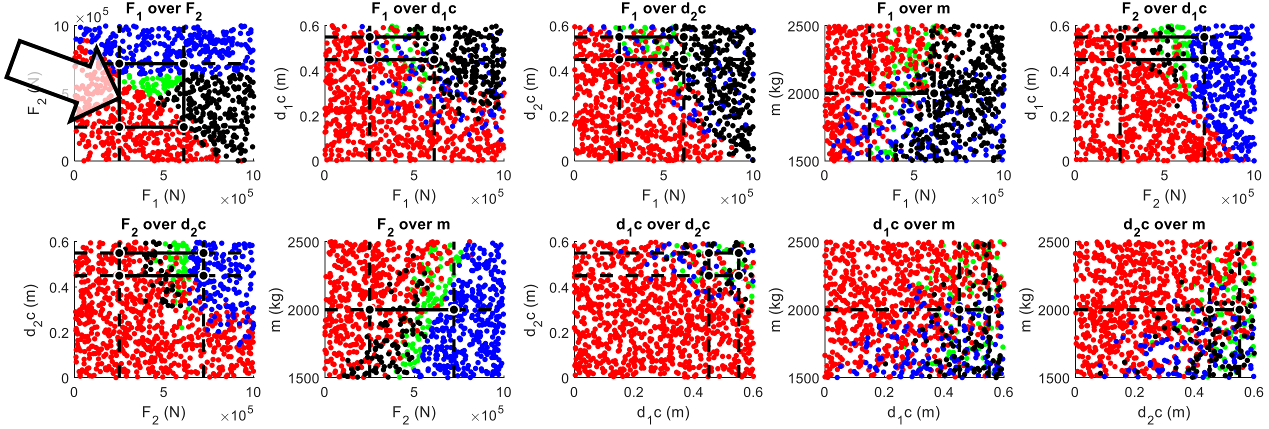
View 1a

- Decrease range of F1, F2, d1c, d2c



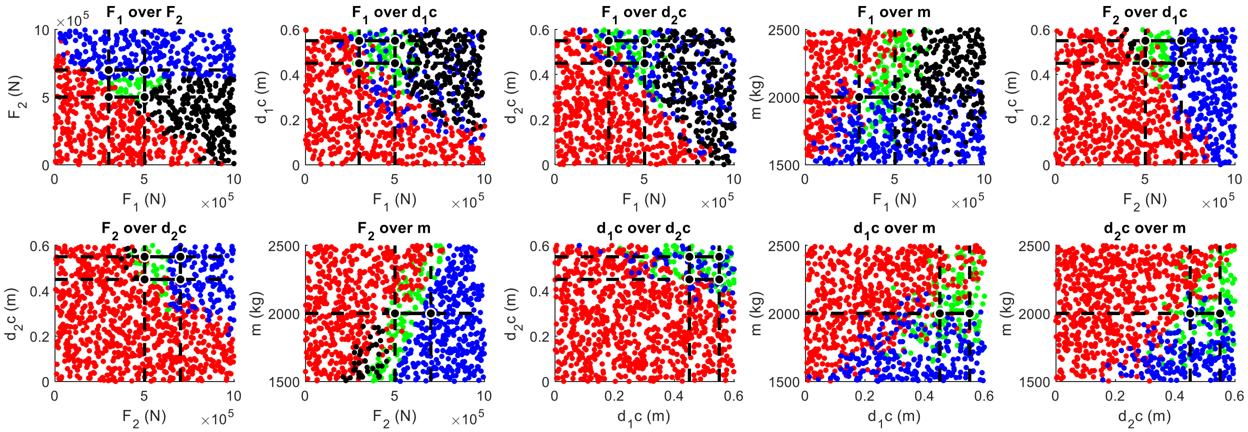
View 1b

- Fix mass



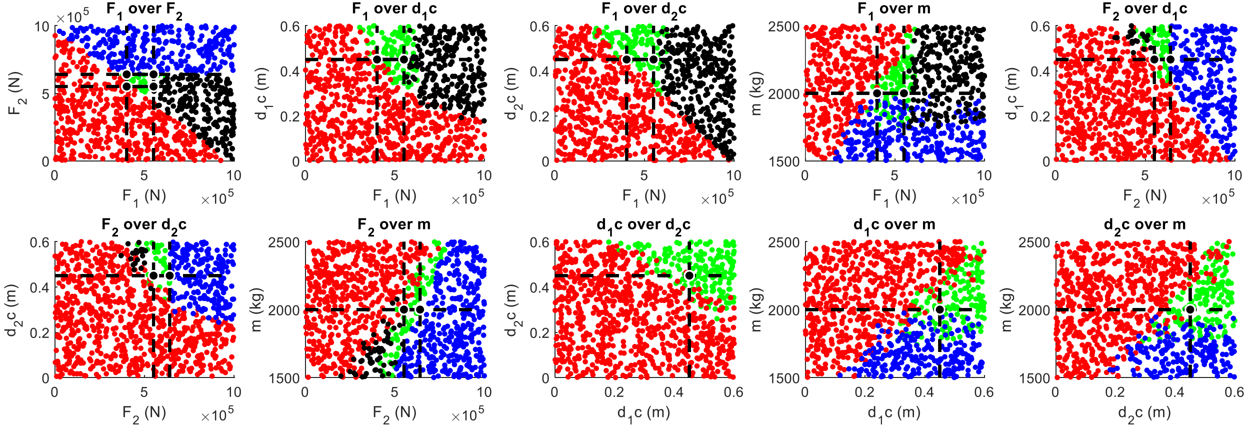
View 1c

- Decrease range of F1, F2, d1c, d2c



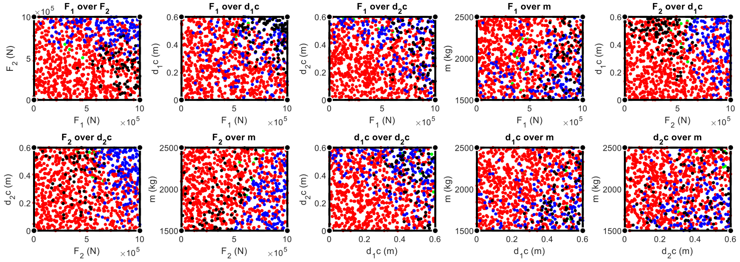
View 1d

- Fix d1c, d2c
- Adjust range of F1, F2

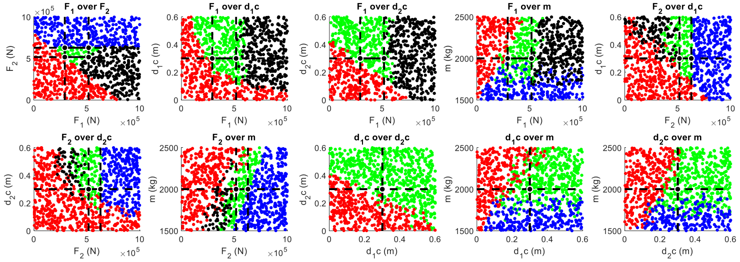


You just constructed a solution to a 5d highly nonlinear Crash problem!

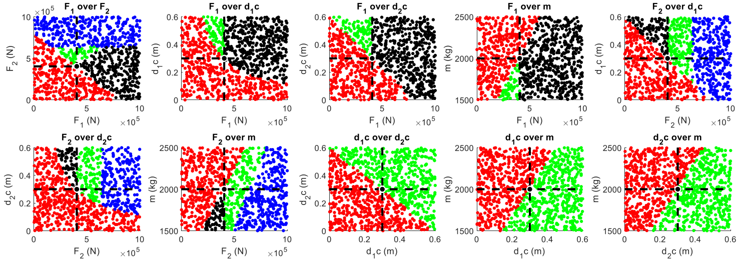
Available Views in Basic X-ray Tool v11



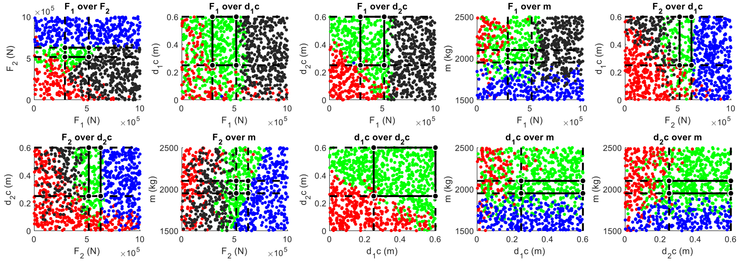
View 1: Complete design space



View 3: maximum box for F1 & F2



View 2: Sections at optimum

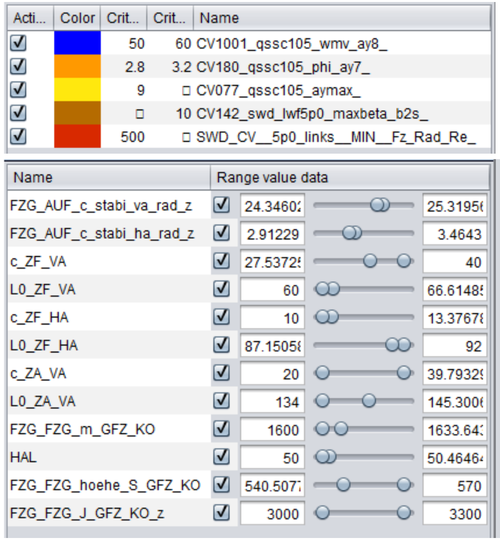


View 4: Large box for all design variables

Professional X-ray Tool – ClearVu Solution Space

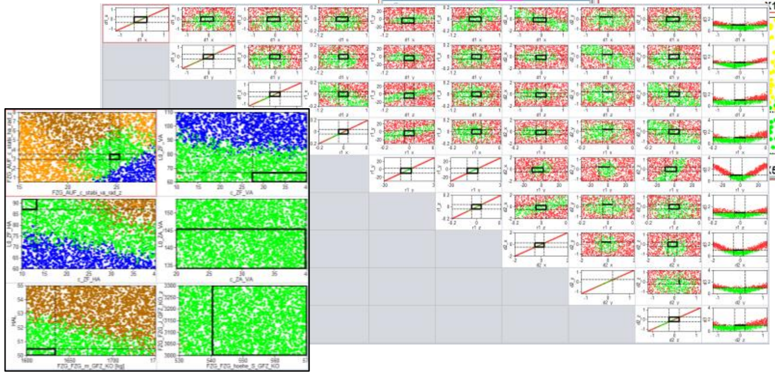
- Developed by BMW and Divis
- Automatic solution space optimization
- Automatic generation of fast models (surrogates)

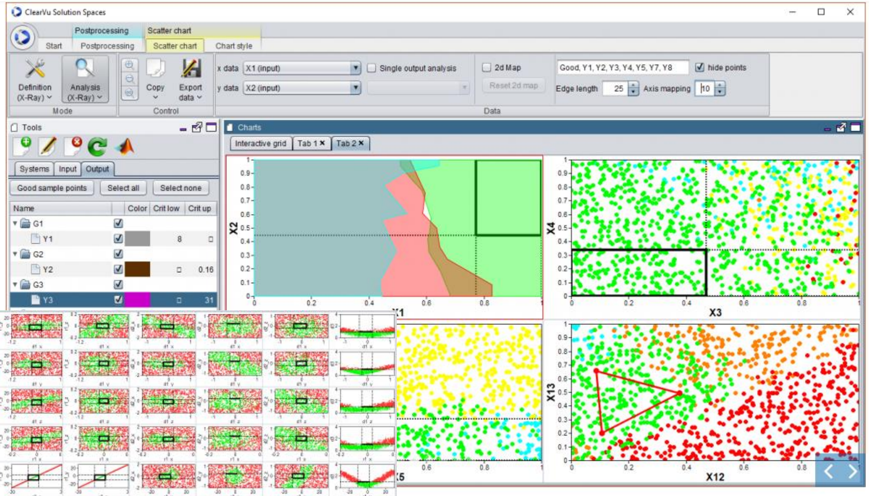
https://divis-gmbh.de/home/software/clearvu-solution-space/



Acti...	Color	Crit...	Crit...	Name
☑		50	60	CV1001_qssc105_wmv_ay8_
☑		2.8	3.2	CV180_qssc105_phi_ay7_
☑		9	☐	CV077_qssc105_aymax_
☑		☐	10	CV142_swd_lwf5p0_maxbeta_b2s_
☑		500	☐	SWD_CV__5p0_links__MIN__Fz_Rad_Re_

Name		Range value data	
FZG_AUF_c_stabi_va_rad_z	☑	24.3460	25.3195
FZG_AUF_c_stabi_ha_rad_z	☑	2.91229	3.4643
c_ZF_VA	☑	27.5372	40
L0_ZF_VA	☑	60	66.6148
c_ZF_HA	☑	10	13.3767
L0_ZF_HA	☑	87.1505	92
c_ZA_VA	☑	20	39.7932
L0_ZA_VA	☑	134	145.300
FZG_FZG_m_GFZ_KO	☑	1600	1633.64
HAL	☑	50	50.4646
FZG_FZG_hoehe_S_GFZ_KO	☑	540.507	570
FZG_FZG_J_GFZ_KO_z	☑	3000	3300





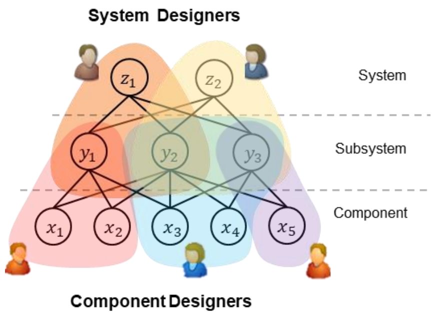
Content

- Solution Spaces
- Solution Space Engineering
- Mini Tutorial

Talks about SSE @ DSM conference 2020

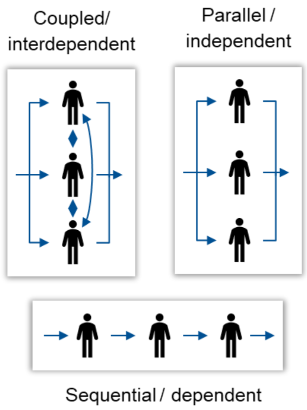
A Role-Activity-Product Model to Simulate Distributed Design Processes

Wöhr, Ferdinand; Königs, Simon; Ring, Philipp; Zimmermann, Markus



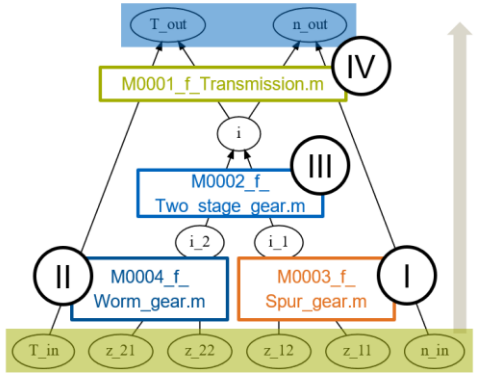
Optimizing Distributed Design Processes for Flexibility and Cost

Daub, Marco; Wöhr, Ferdinand; Zimmermann, Markus



Sequencing of Information in Modular Model-based Systems Design

Rötzer, Sebastian; Rostan, Nicky; Steger, Hans Christian; Vogel-Heuser, Birgit; Zimmermann, Markus



Wed, Oct. 14th: Process Architecture
10:15 USEDT

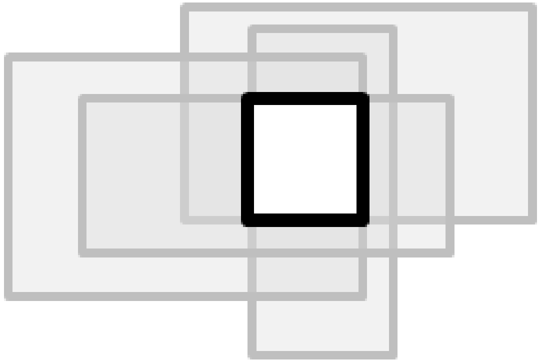
Wed, Oct. 14th: Product Architecture
11:45 USEDT

Summary

- Circular dependencies can be avoided by strict separation of top-down and bottom-up view.
- Requirements formulated as solution spaces enable independent design work.
- Applicable to any system with available bottom-up mappings.

Limitations:

- Requirements often not available
- Models not available or expensive to build
- Loss of solution space



Related Publications

- M. Daub, F. Duddeck, M. Zimmermann, **2020**. Optimizing Component Solution Spaces for Systems Design. Structural and Multidisciplinary Optimization, DOI:10.1007/s00158-019-02456-8.
- H. Harbrecht, D. Tröndle, M. Zimmermann, **2019**. *A sampling-based optimization algorithm for solution spaces with pair-wise-coupled design variables*. Structural and Multidisciplinary Optimization, online: https://doi.org/10.1007/s00158-019-02221-x
- M. Vogt, F. Duddeck, M. Wahle, M. Zimmermann, **2018**. *Optimizing tolerance to uncertainty in systems design with early-and late-decision variables*., IMA Journal of Management Mathematics, DOI:10.1093/imaman/dpy003
- S. Erschen, F. Duddeck, M. Gerdts, M. Zimmermann, **2017**. *On the optimal decomposition of high-dimensional solution spaces of complex systems.* ASCE-ASME Journal of Risk and Uncertainty in Engineering Systems, Part B: Mechanical Engineering, DOI:10.1115/1.4037485
- M. Zimmermann, S. Königs, C. Niemeyer, J. Fender, C. Zeherbauer, R. Vitale, M. Wahle, **2017**. *On the design of large systems subject to uncertainty*. Journal of Engineering Design, vol. 28(4), pp. 233-254
- J. Fender, F. Duddeck, M. Zimmermann, **2017**. *Direct computation of solution spaces for crash design*. Structural and Multidisciplinary Optimization, vol. 55(5), pp. 1787-1796
- L. Graff, H. Harbrecht, M. Zimmermann, **2016**. *On the computation of solution spaces in high dimensions*. Structural and Multidisciplinary Optimization, vol. 54(4), pp. 811-829
- M. Eichstetter, S. Müller, M. Zimmermann, **2015**. *Product Family Design Using Solution Spaces.* Journal of Mechanical Design, vol. 137(12), p. 121401
- J. Fender, L. Graff, H. Harbrecht, M. Zimmermann, **2014**. *Identifying Key Parameters for Design Improvement in High-Dimensional Systems with Uncertainty.* Journal of Mechanical Design, vol. 136(4), p. 041007
- J. Fender, F. Duddeck, M. Zimmermann, **2014**. *On the calibration of simplified vehicle crash models.* Structural and Multidisciplinary Optimization,
vol. 49(3), pp. 455-469
- M. Zimmermann, J. Edler von Hoessle, **2013**. *Computing Solution Spaces for Robust Design*. International Journal for Numerical Methods in Engineering, vol. 94(3), pp. 290-307
- M. Lehar, M. Zimmermann, **2012**. *An inexpensive estimate of failure probability for high-dimensional systems with uncertainty.* Structural Safety, vol. 36-37, pp. 32-38.

Thank you for your attention!

LPL

Plan – Design – Build

Backup

Definitions

A **solution space** is *a set* of **good designs**, i.e., designs that satisfy all requirements. Formally, for a design problem with

(1) design variables $\boldsymbol{x} = (x_i)$,

(2) performance $\boldsymbol{y} = \left(y_j\right)$ and $y_j = f_j(\boldsymbol{x})$ and

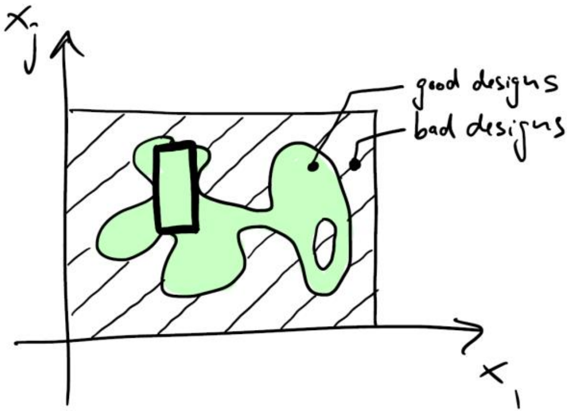
(3) requirements $y_{jl} \leq y_j \leq y_{ju}$,

a solution space Ω satisfies: $y_{jl} \leq f_j(\boldsymbol{x}) \leq y_{ju}, \forall \boldsymbol{x} \in \Omega$.

The **complete solution space** is the set of all good designs.

A **box-shaped solution space** is a solution space expressed as product of permissible intervals $\Omega = [x_{1l}, x_{1u}] \times \cdots \times [x_{dl}, x_{du}]$

A **design space** is the set of all designs considered in the design problem.

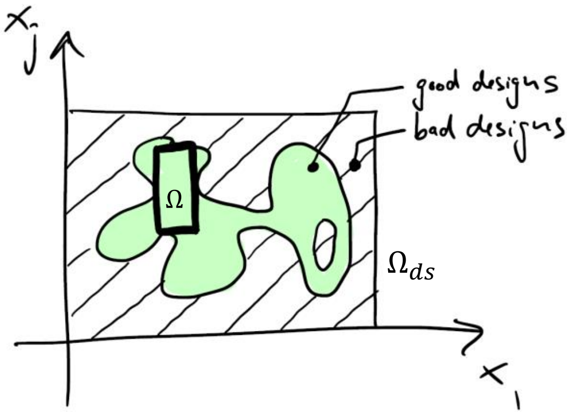


Problem Statement for Box-shaped Solution Spaces

For given output functions $f_j(\boldsymbol{x})$ and
associated requirements=constraints $f_j(\boldsymbol{x}) \leq y_{jc}$:

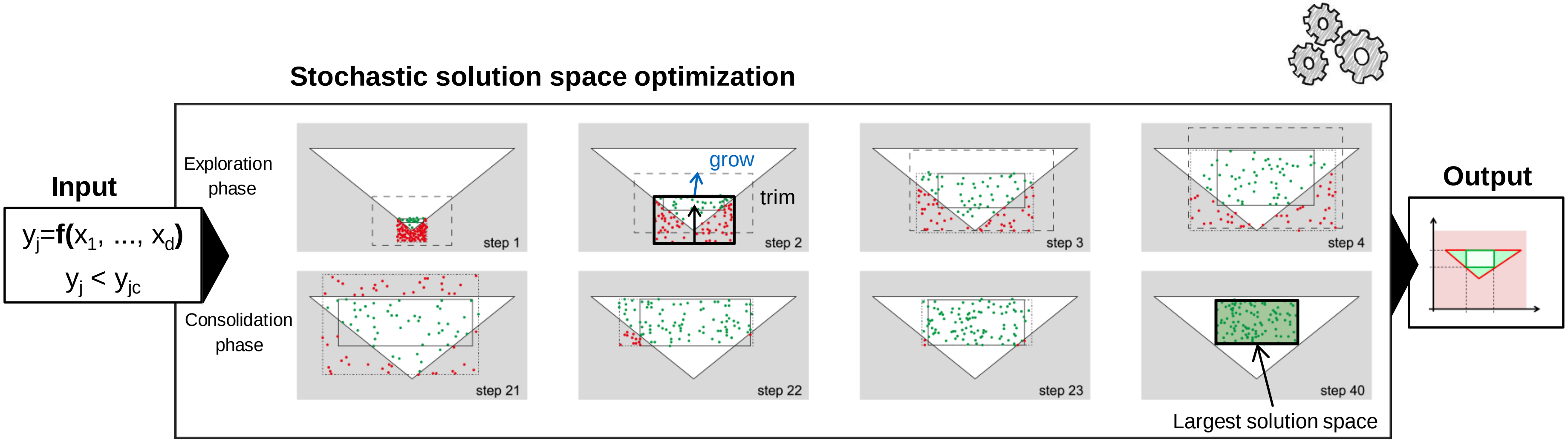
$$\max_{\Omega} \mu(\Omega)$$
$$\text{subject to: } f_j(\boldsymbol{x}) \leq y_{jc} \quad \forall \boldsymbol{x} \in \Omega$$

$\Omega = [x_{1l}, x_{1u}] \times \cdots \times [x_{dl}, x_{du}] \subseteq \Omega_{ds}$
$\mu(\Omega) = (x_{1u} - x_{1l}) \ldots (x_{du} - x_{dl})$



- The general problem may be arbitrarily non-linear, non-convex, not simply connected, …
- How to detect bad designs in your solution space?

Stochastic Iteration – Overview

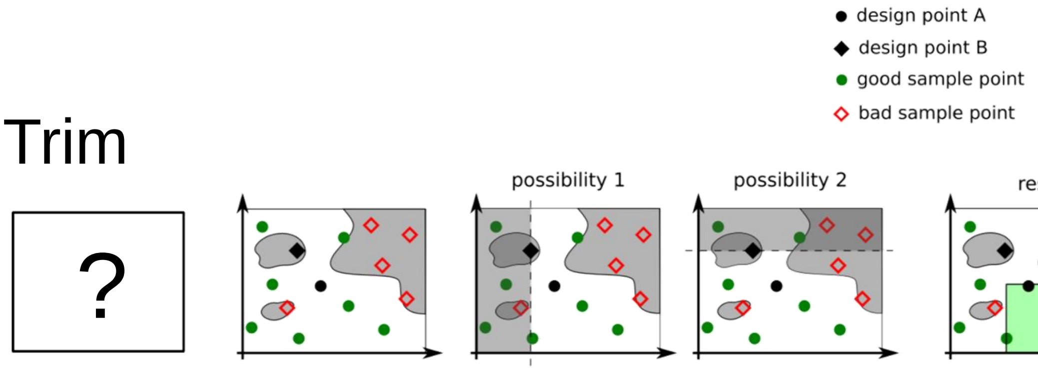


- Algorithm uses iterative stochastic sampling and modification.
- Solves arbitrary high-dimensional and non-linear problems, e.g., 100d crash problem.

M. Zimmermann, J. Edler von Hoessle: Computing solution spaces for robust design. **International Journal for Numerical Methods in Engineering (2013)**

Courtesy of BMW

Stochastic Iteration – Trim

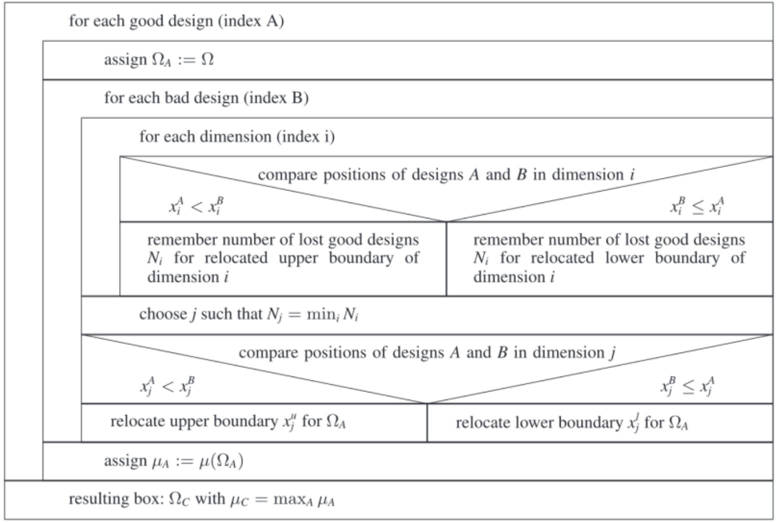


Modification step A / Trim:

- Input: candidate box with sample data $[y_A, \boldsymbol{x}_A]$
- Output: new candidate box with all bad sample points removed

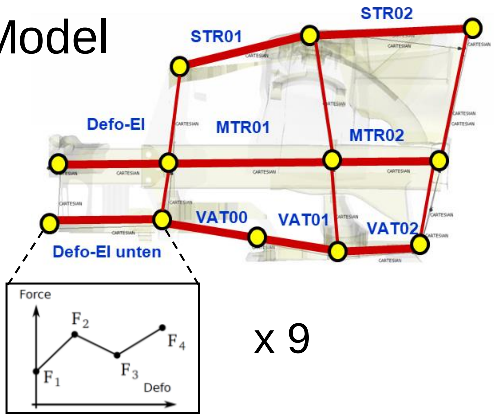
- $A, B = 1, \dots, N$ are sample indices
- Loop 1 over good designs
- Loop 2 over bad designs
- Loop 3 over design variables to find least painful boundary relocation (i.e. least loss of good sample points)

Modification step A: Trim



Stochastic Iteration – 36 design variables

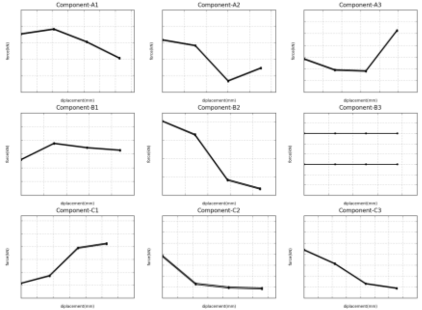
Model



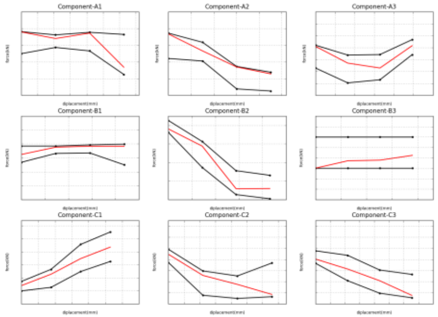
x 9

- Each force-deformation characteristic F(s) is represented by 4 support points
- Algorithm converges after 200 iterations
  → **~10⁵ function evaluations**

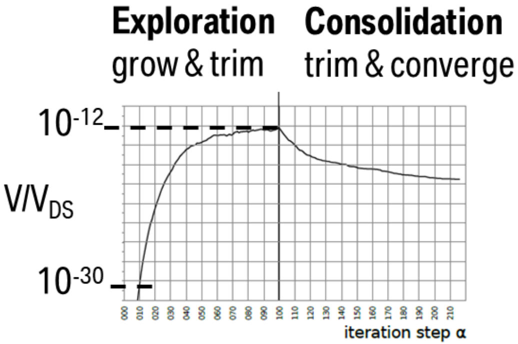
Exploration of the design space

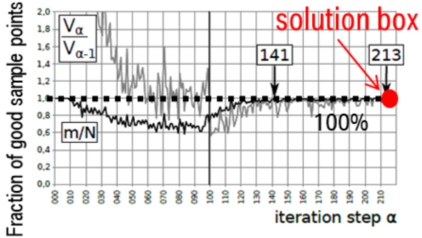


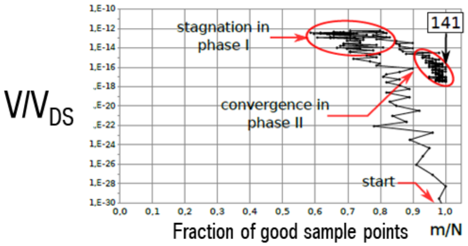
Sampling a candidate space



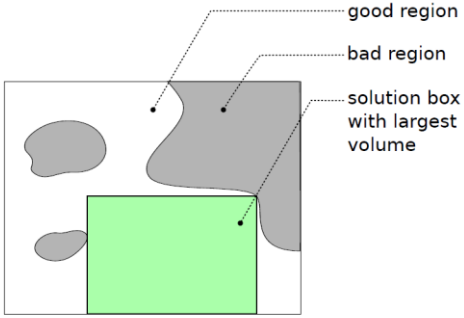
Candidate Solution Space / Test variants

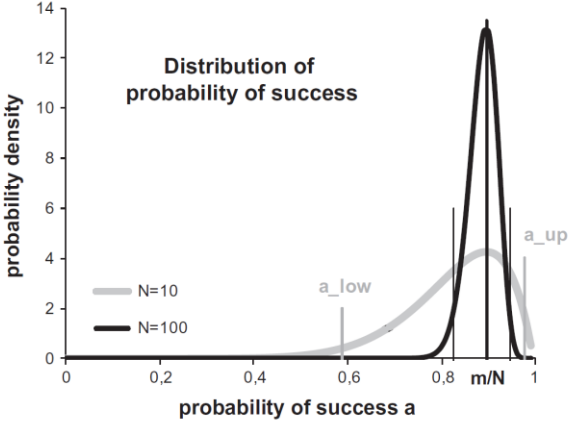


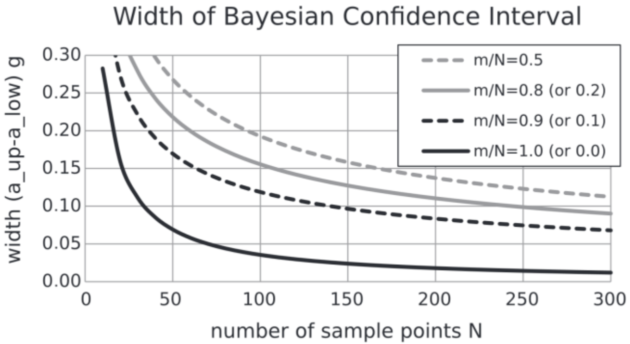




How to Detect Bad Designs in a Candidate Solution Space?





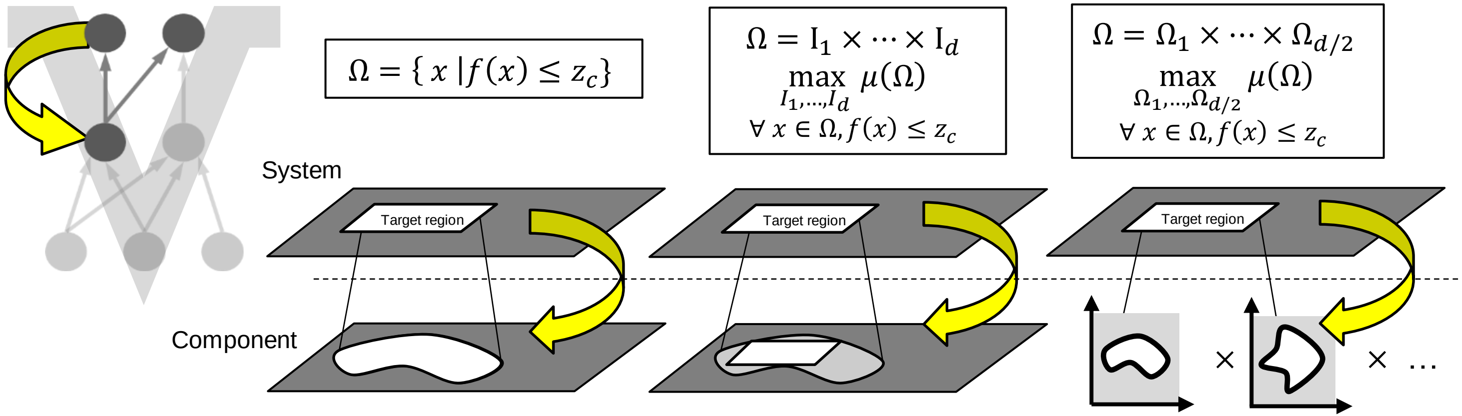


from [2]

- A candidate box is probed by Monte Carlo sampling with N~100.
- The probability density of the **true fraction of the good space a** is the beta distribution.
- m/N < 1: There are bad sample points in the box → modify.
- m/N = 1: Only good sample points $\boxed{\to\ \mathbf{P(97\% < a) = 95\%}}$
- This is independent of the number of dimensions!

Top-down Mappings

Complete solution space | **Solution box** | **2d-spaces**



+ No loss of solution space!

+ Complete decoupling

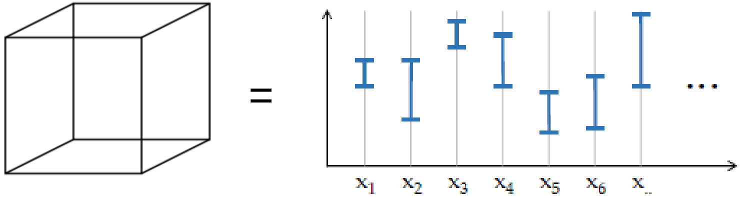
+ Selective decoupling to reduce loss of solution space

- Map permissible performance values onto regions of design variables = <u>many designs</u>.
- Need to carefully balance (1) decoupling and (2) loss of solution space.

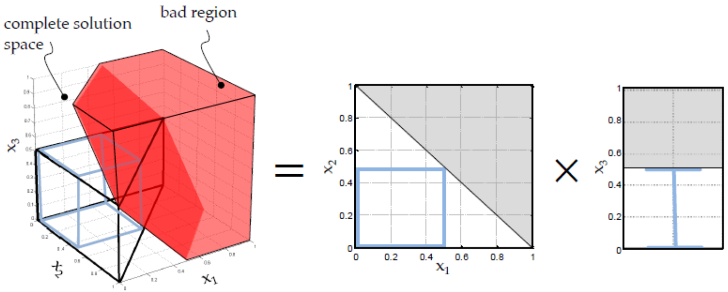
Courtesy of BMW

2d-Spaces – Underlying Idea

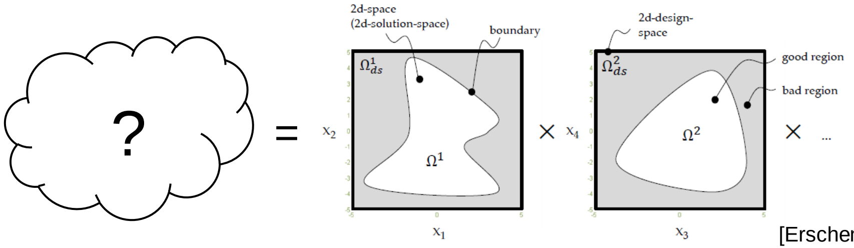
- Product of intervals = box



- Interval x 2d-space



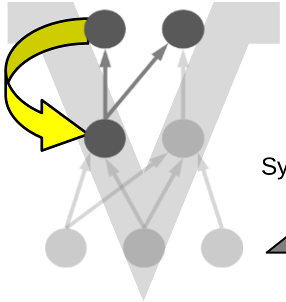
- Product of 2d-spaces



[Erschen]

Courtesy of BMW

Top-down Mappings



Complete solution space

$$\Omega = \{\, x \mid f(x) \le z_c \}$$

Solution box

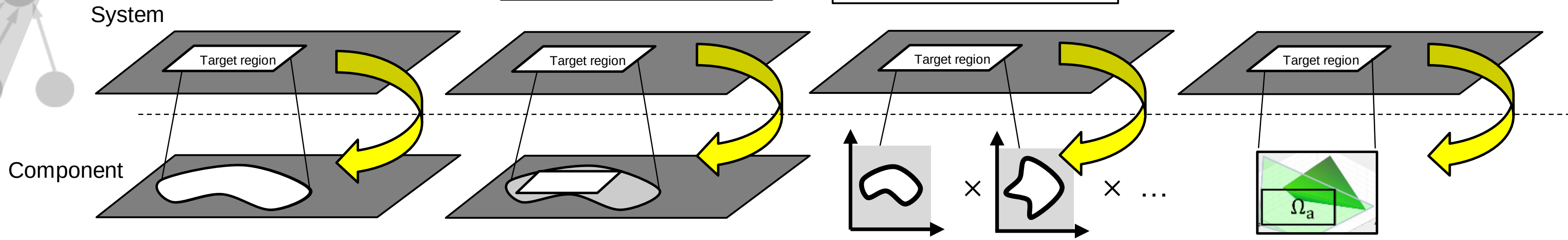
$$\Omega = \mathrm{I}_1 \times \cdots \times \mathrm{I}_d$$
$$\max_{I_1,\ldots,I_d} \ \mu(\Omega)$$
$$\forall\, x \in \Omega, f(x) \le z_c$$

2d-spaces

$$\Omega = \Omega_1 \times \cdots \times \Omega_{d/2}$$
$$\max_{\Omega_1,\ldots,\Omega_{d/2}} \ \mu(\Omega)$$
$$\forall\, x \in \Omega, f(x) \le z_c$$

Solution-Compensation Spaces

$$\Omega = \Omega_a \times \Omega_b$$
$$\max_{\Omega_a} \mu(\Omega)$$
$$\forall\, x_a \in \Omega_a, \exists x_b \in \Omega_b, f(x_a, x_b) \le z_c$$



+ No loss of solution space!

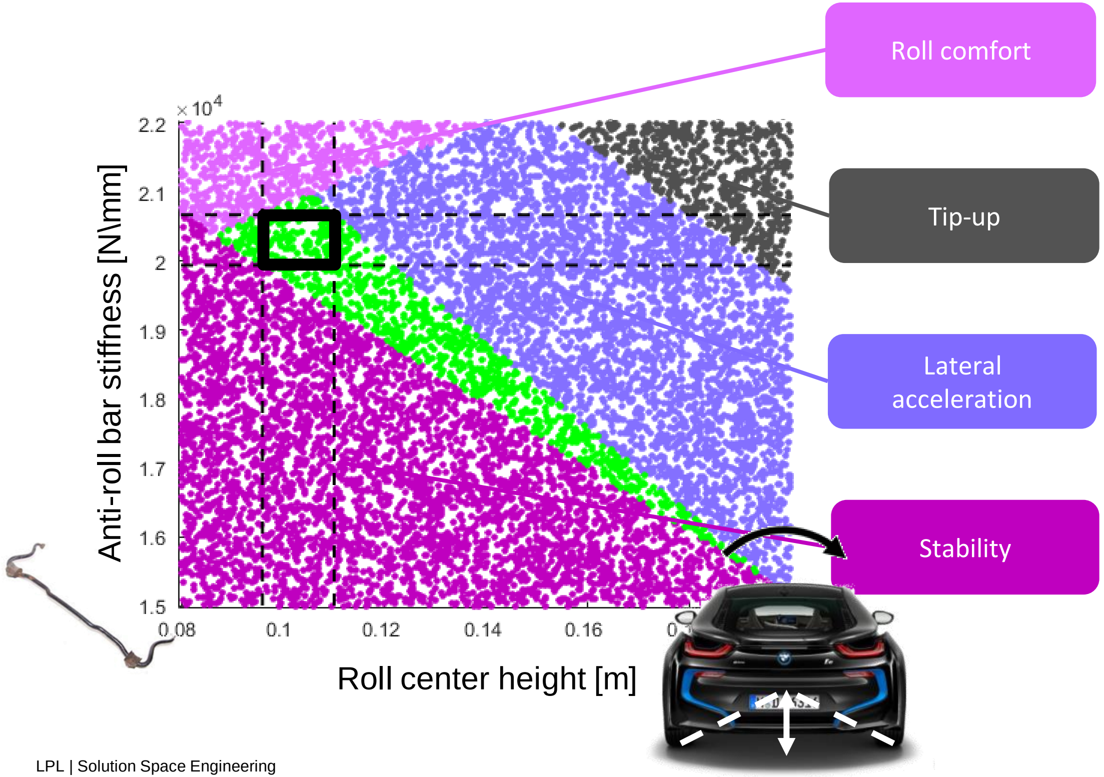
+ Complete decoupling

+ Selective decoupling to reduce loss of solution space

+ Decision order to reduce loss of solution space

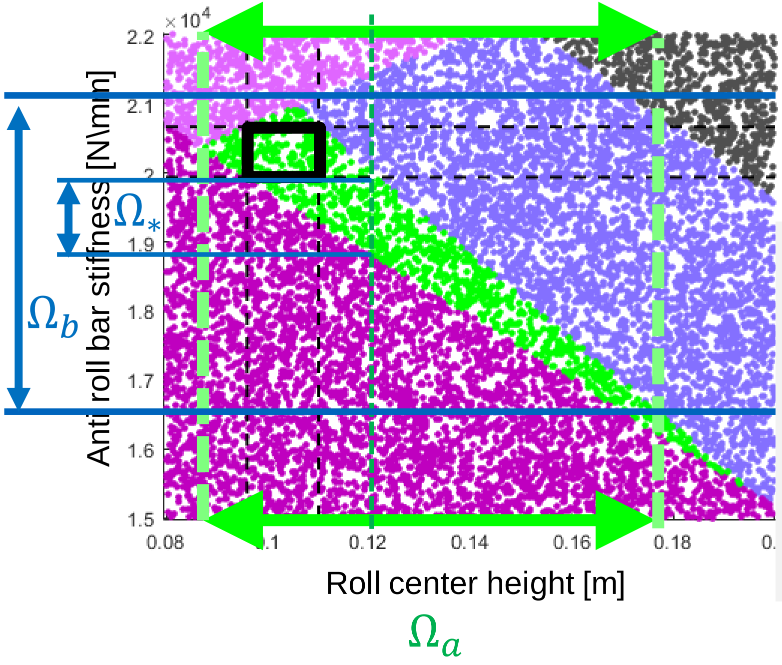
- Map permissible performance values onto regions of design variables = <u>many designs</u>.
- Need to carefully balance (1) decoupling and (2) loss of solution space.

Solution-Compensation Spaces



[courtesy of BMW, Vogt]

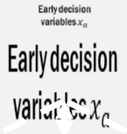
Solution-Compensation Spaces



Early decision variables x_a

Late decision variables x_b

May assume any value in their intervals



Have to be able to assume any value in their intervals

Problem statement

For given

(1) output functions $f_j(x_a, x_b)$ and

(2) associated requirements $f_j(x) \le y_{jc}$

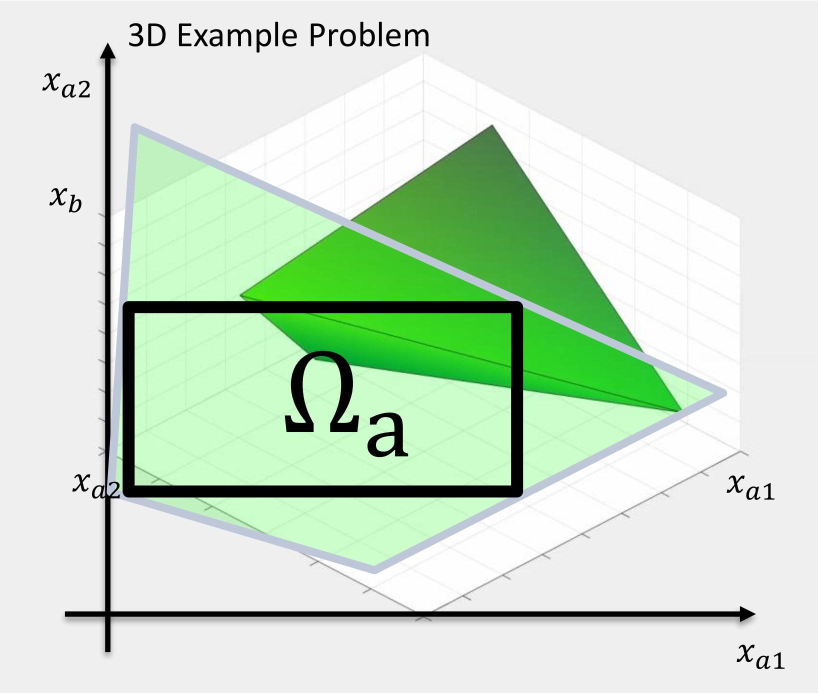
(3) compensation space Ω_b

$$\max_{\Omega_a} \mu(\Omega)$$
$$\forall\, x_a \in \Omega_a, \exists x_b \in \Omega_b, f_j(x_a, x_b) \le y_{jc}$$

$$\Omega = \Omega_a \times \Omega_b$$

[courtesy of BMW, Vogt]

Solution-Compensation Spaces – Projection



Problem statement
For given
(1) output functions $f_j(x_a, x_b)$ and
(2) associated requirements $f_j(x) \leq y_{jc}$
(3) compensation space Ω_b

$$\max_{\Omega_a} \mu(\Omega)$$
$$\forall\, x_a \in \Omega_a, \exists x_b \in \Omega_b, f_j(x_a, x_b) \leq y_{jc}$$

$$\Omega = \Omega_a \times \Omega_b$$

LINEARITY

$$f_j(\boldsymbol{x}) = \sum_i A_{ji} x_i \leq y_{jc}$$

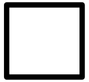
$$\max_{\Omega_a} \mu(\Omega_a)$$
$$\forall\, x_a \in \Omega_a, \sum_i A^*_{a,ji}\, x_{a,i} \leq y_{jc}$$

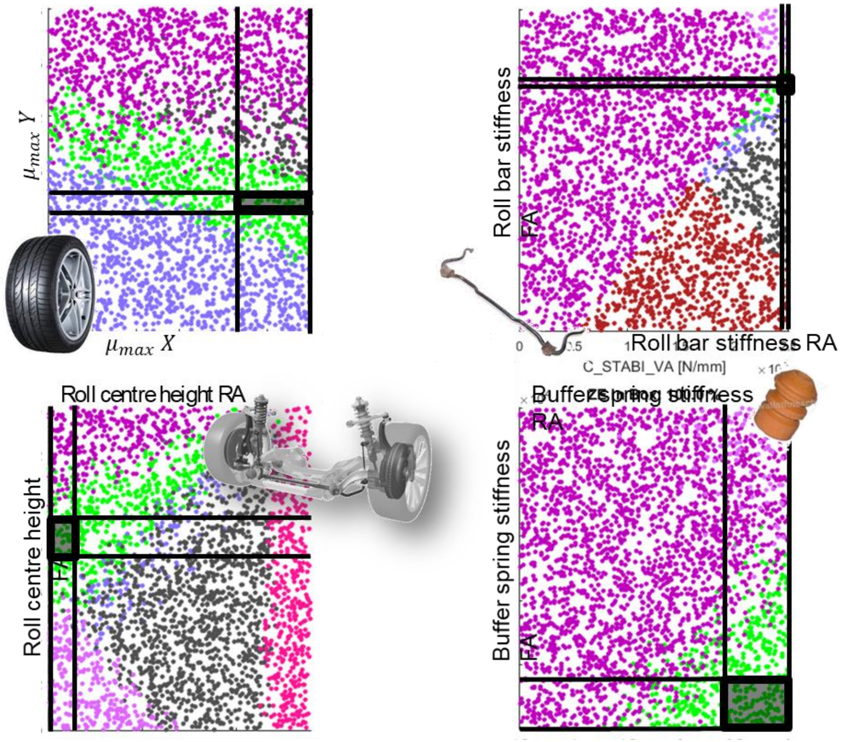
$$\Omega = I_1 \times \cdots \times I_d$$

[courtesy of BMW, Vogt]

Example 8: Vehicle Dynamics Design

Chassis design problem with
linear performance function
8 design variables
6 requirements


Initial box-shaped solution space



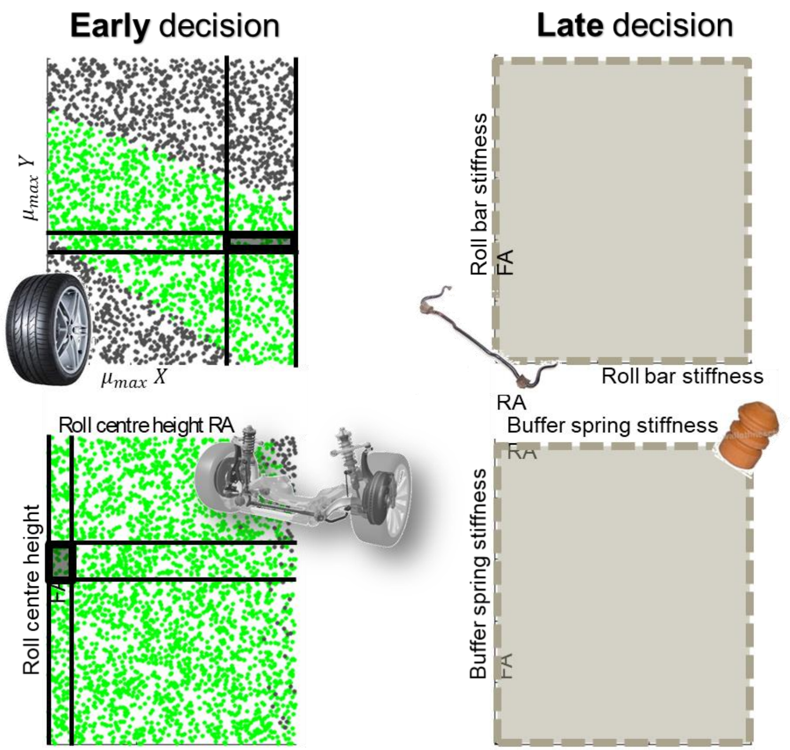
[courtesy of BMW, Vogt]

Example 8: Vehicle Dynamics Design

Chassis design problem with
linear performance function
8 design variables
6 requirements

Initial box-shaped solution space (type a)

Compensation Space (type b)



[courtesy of BMW, Vogt]

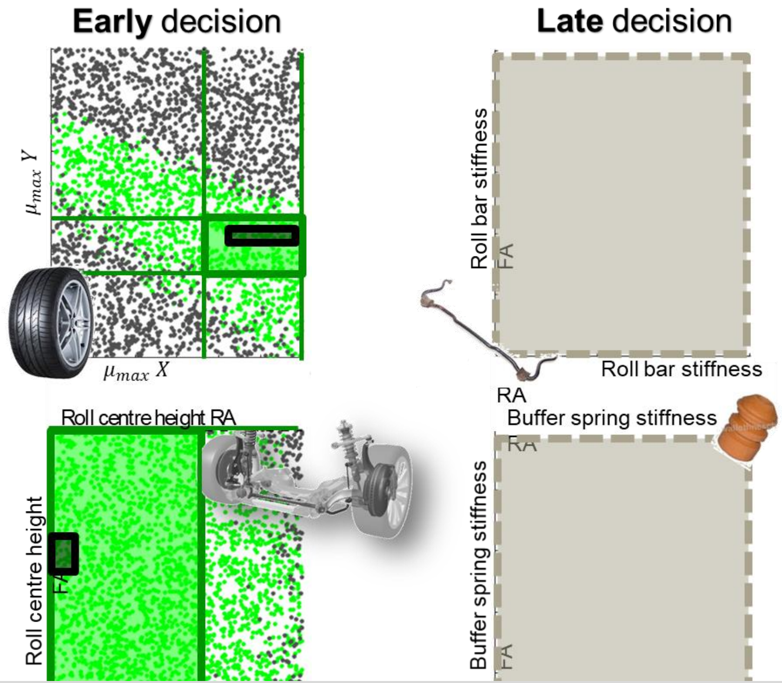
Example 8: Vehicle Dynamics Design

Chassis design problem with
linear performance function
8 design variables
6 requirements

Initial box-shaped solution space (type a)

Compensation Space (type b)

Box-shaped solution space (type a) in projected area



M. Vogt, F. Duddeck, M. Wahle, M. Zimmermann, **2018**. *Optimizing tolerance to uncertainty in systems design with early-and late-decision variables*. IMA Journal of Management Mathematics, DOI:10.1093/imaman/dpy003

Pictures

fotolia.com, shutterstock.com, iStockphoto.com
BMW, Munich