ESTECO
USERS' MEETING
INDIA

um 2023

Multi-objective Robust Design Optimization of Electric Vehicle Suspension System

Dr. Ranga Srinivas Gunti
General Manager -Tata Motors Ltd.
Adjunct Faculty – IIT Madras
Adjunct Faculty – Virginia Tech (US)

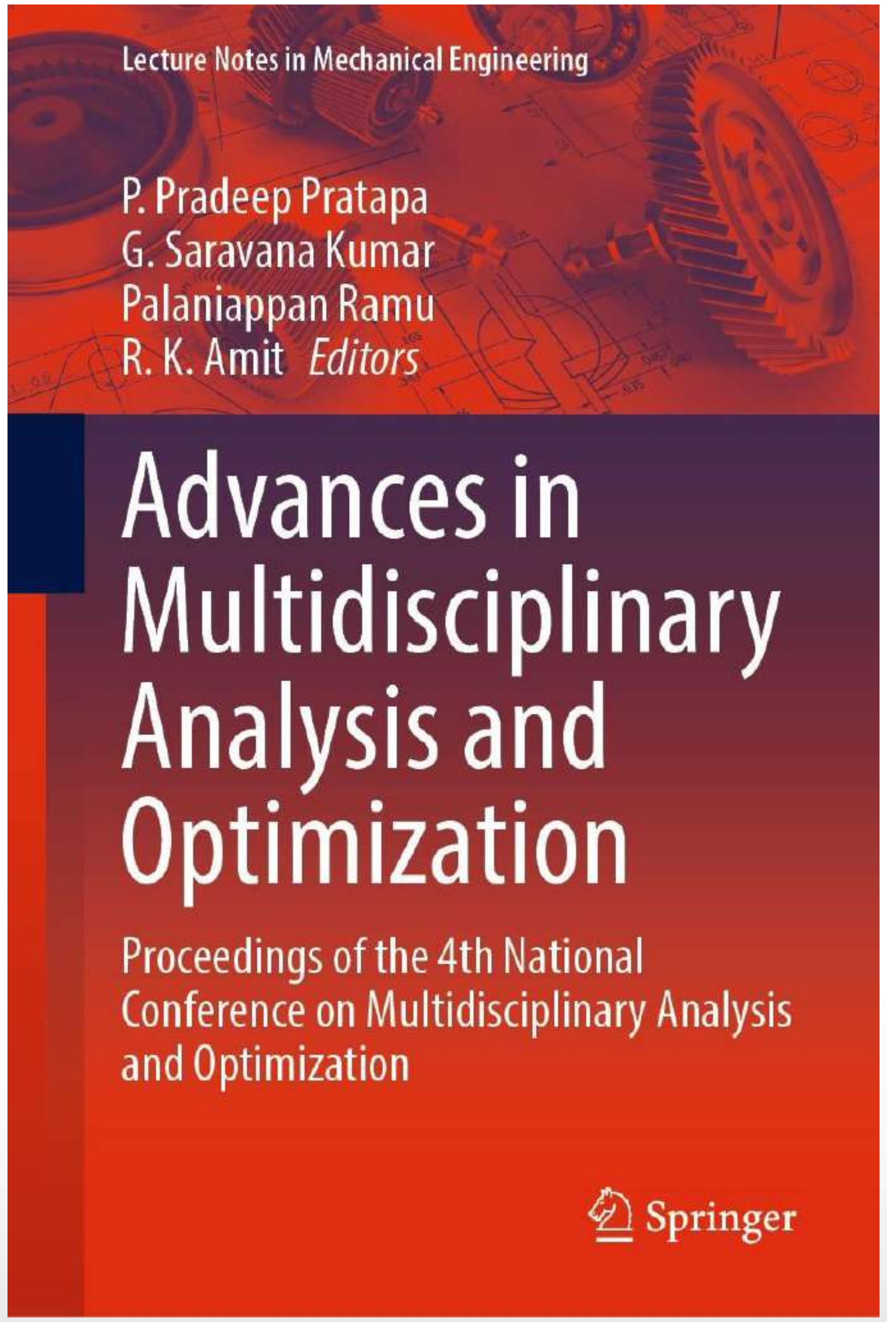
Mr. Anand Pitchaikani & Mr. Ajay Kumar
Modelon India

Topics Covered

- Introduction and the need for MDO – Automotive Suspension Systems
- Optimization Methodology
- Problem Statement and Definition
- Development of Calibrated – System Level Vehicle Dynamics Model
- Perform Multi-objective Design Optimization
- Engineering and Technology Recommendations
- Robust Design Optimization

ESTECO
USERS' MEETING
INDIA

A Methodology for Multi-objective Optimization of Vehicle Suspension Systems



Lingadalu Ganesh, Srinivas Gunti, N. Balaramakrishna,
Shankar Venugopal

Introduction and the need for MDO – Automotive Systems

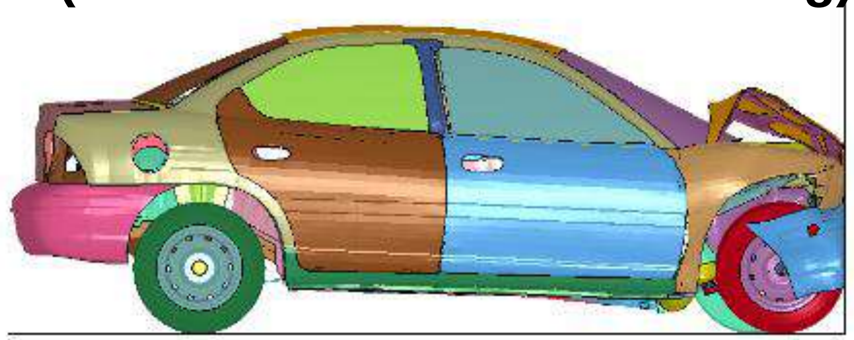
Vehicle Performance Integration – Multiple Attribute Requirements

ESTECO
USERS' MEETING
INDIA

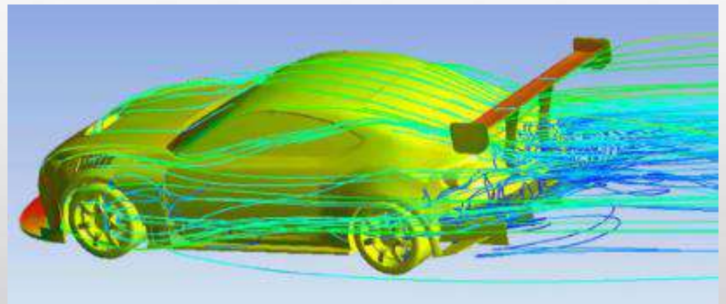
um 2023

Durability
(Fatigue Life > 1 lakh Cycles)

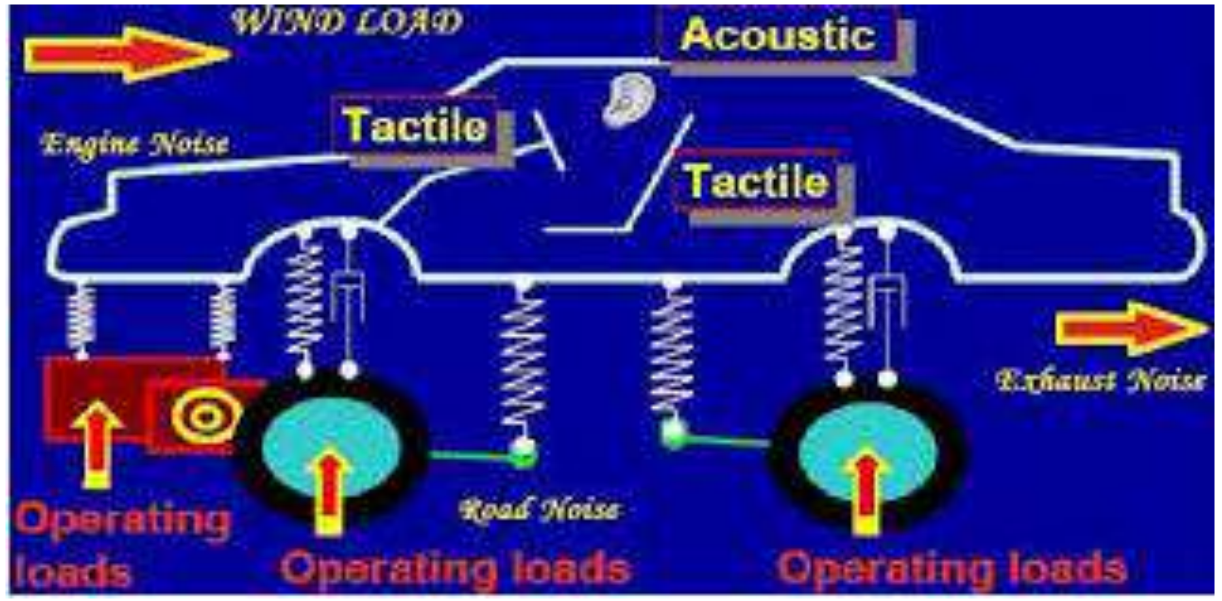
Crash
(GNCAP : 4 Star / 5 Star Rating)



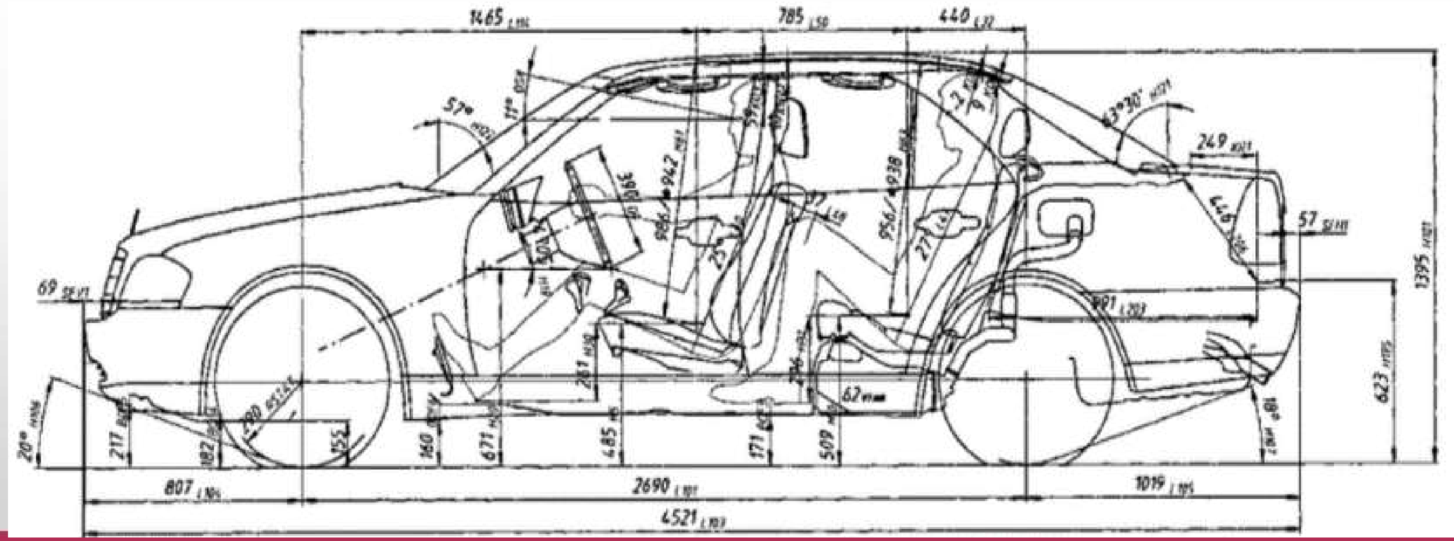
Fluid Dynamics
Cd < 0.32



Handling, Ride and NVH
Lateral, Longitudinal and Vertical Dynamics
Impact vertical shacks < 7 m/sec^2
Understeer Gradient 6 to 8 deg/m/sec^2
Cabin Noise – Driver ear < 45dBA



Ergonomics and Packaging Requirements



Ride

Handling

Development of a Methodology to Optimize Vehicle Systems Considering – Conflicting Requirements Pertaining to Ride and Handling

ESTECO

Introduction and the need for MDO – Automotive Systems

ESTECO USERS' MEETING INDIA

um 2023

Deep Learning based Customer Sentiment Analysis – Ride and Handling

Comfort

Design

Value_Money

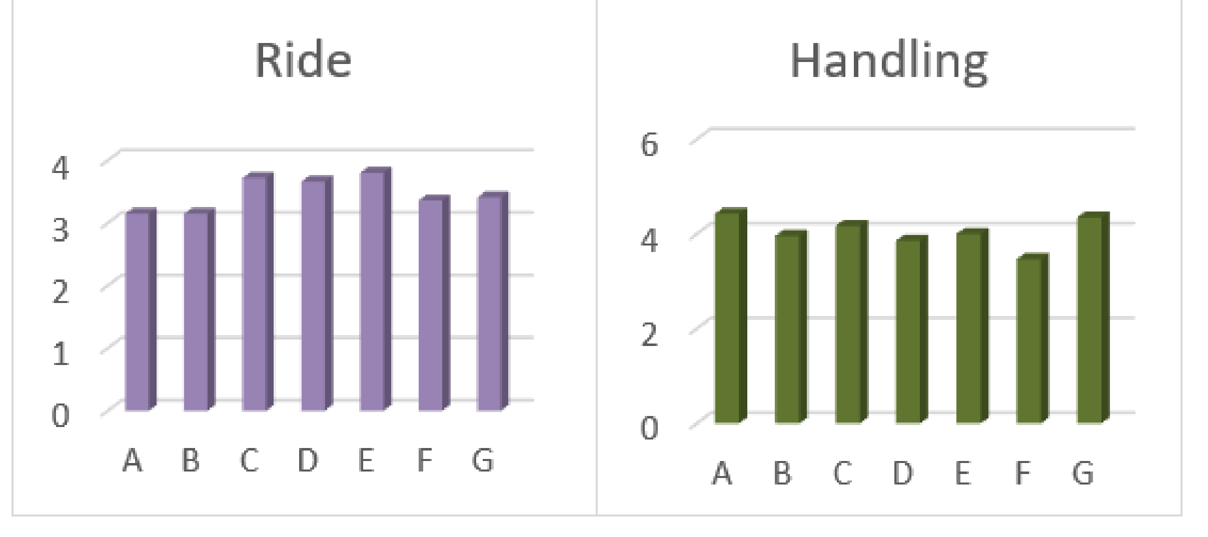
Safety

Infotainment

Connected

Mileage

Customer Sentiment		
Model	**Happy and Excited**	**Improvement Required**
A	Design, Handling	Storage, Ride
B	Infotainment, Storage	Connected, Ride
C	Ride, Ergonomic Comfort	Service, Safety
D	Safety, Ride	Connected, Infotainment
E	Ride, Ergonomic Comfort	Safety, Money
F	Infotainment, Service	Handling, Design
G	Design, Safety	Mileage, Storage



There is a need to
Optimize the Vehicle Systems Considering Both
Ride and Handling – Simultaneously

Methodology

Development of Vehicle Dynamics Model

↓

Model Calibration

↓

Selection of DVP for MDO Based on Customer Perception

↓

Formulation of Problem Definition

↓

DOE Analysis - Full Vehicle Analysis (in ADAMS Insight)

↓

Perform MDO

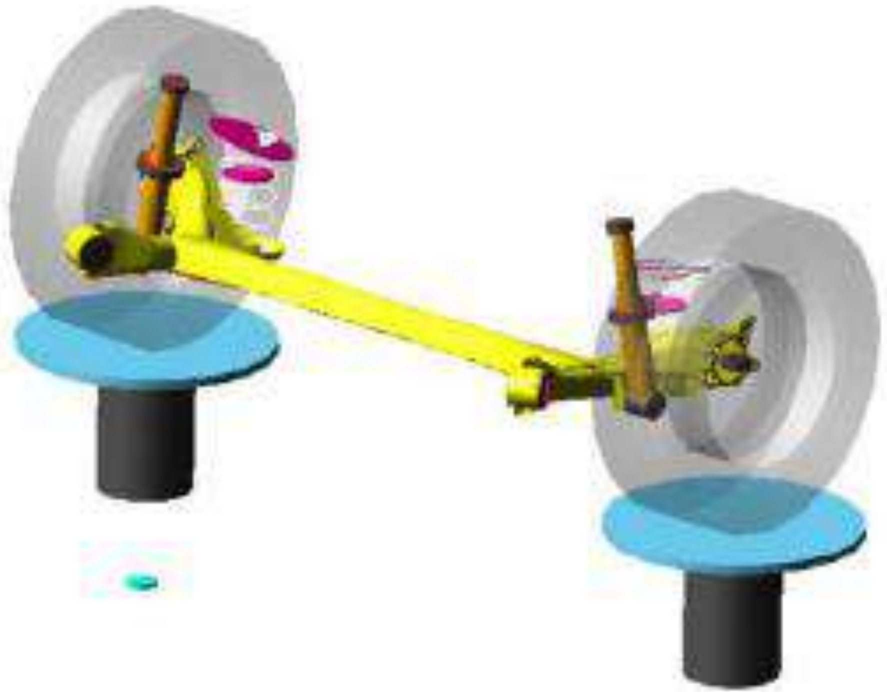
↓

Engineering and Technology Recommendations

Development of Vehicle Dynamics Model

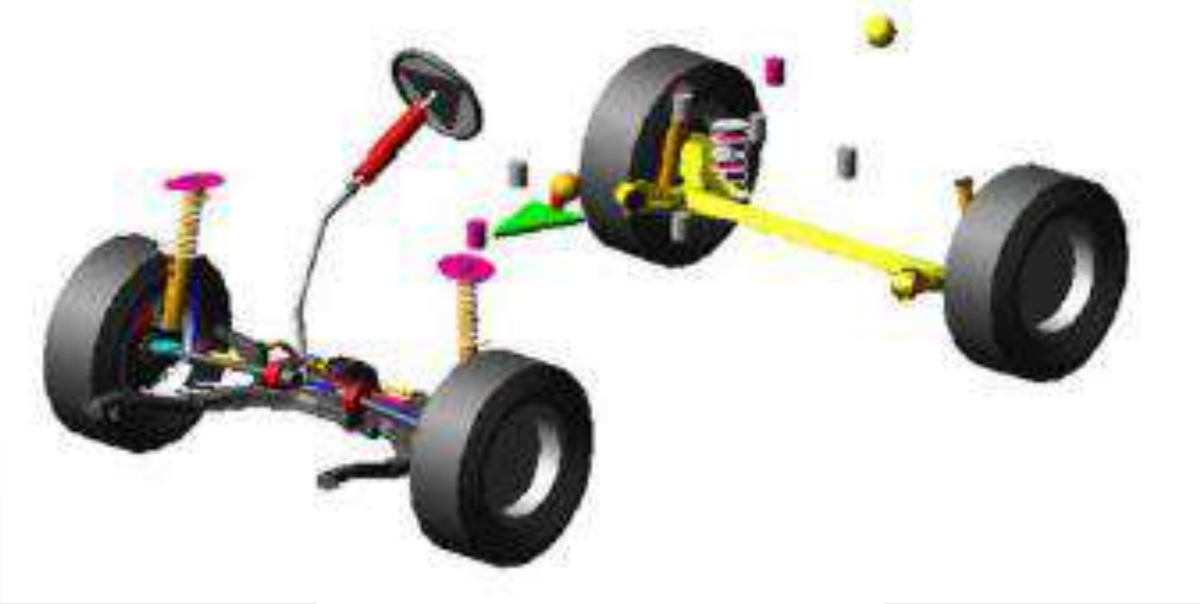
MacPherson Strut with Rack and Pinion Steering Assembly

Front - Suspension



Twist Beam Suspension

Rear - Suspension



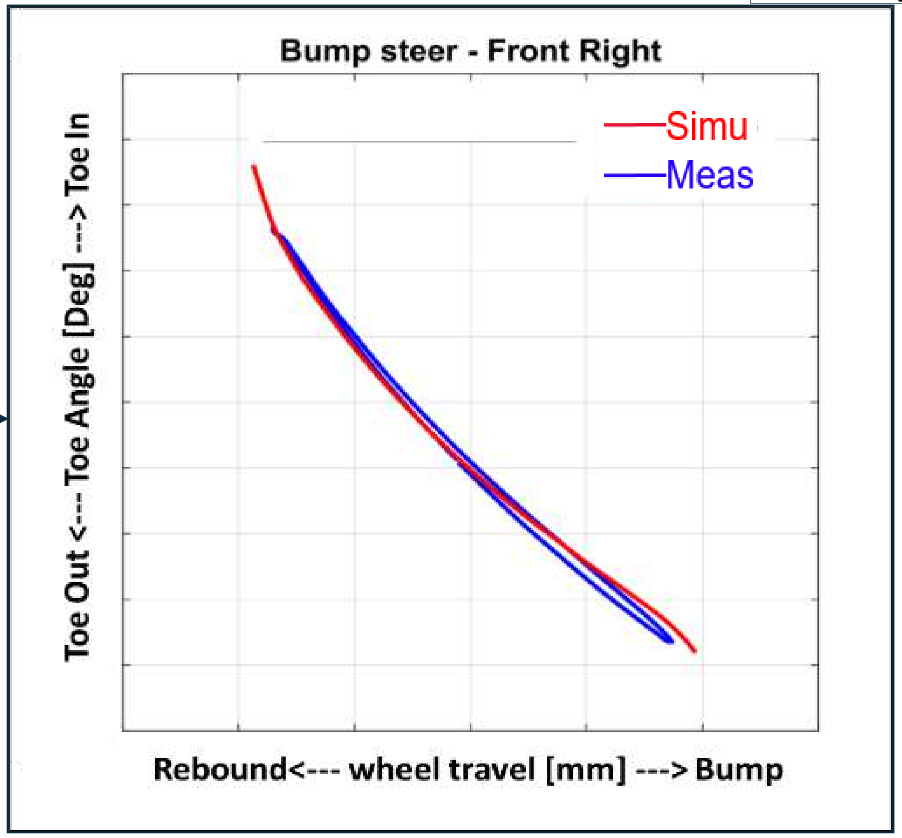
Full Vehicle Model

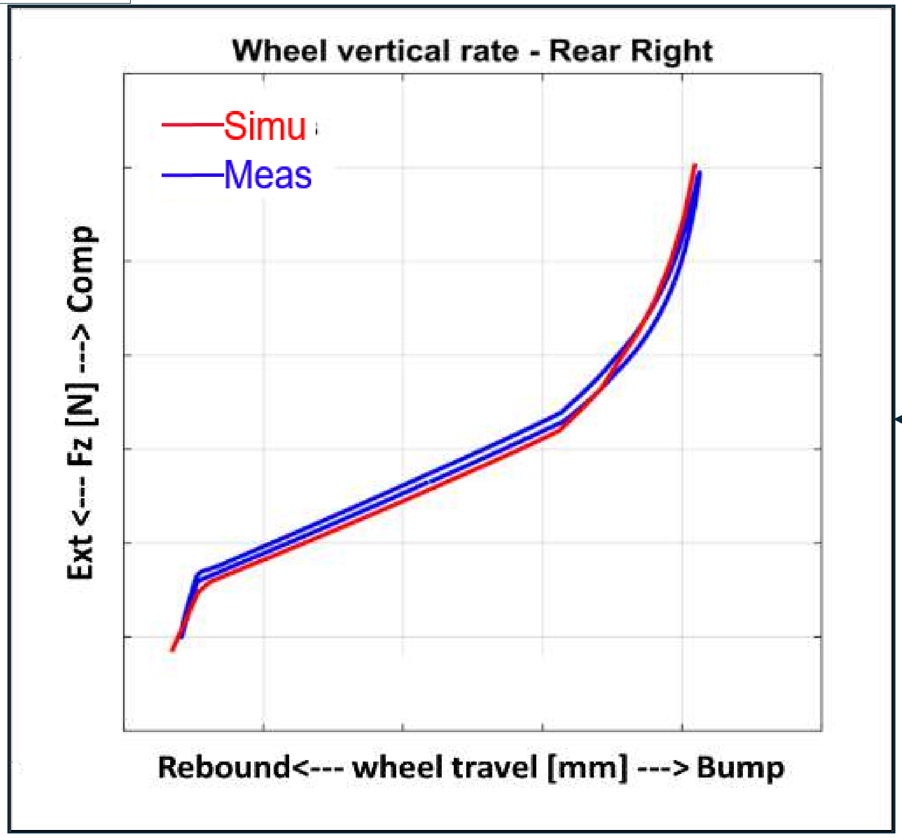
MBD Model Calibration

ESTECO USERS' MEETING INDIA

Subsystem level

Front Suspension Bump Steer correlation

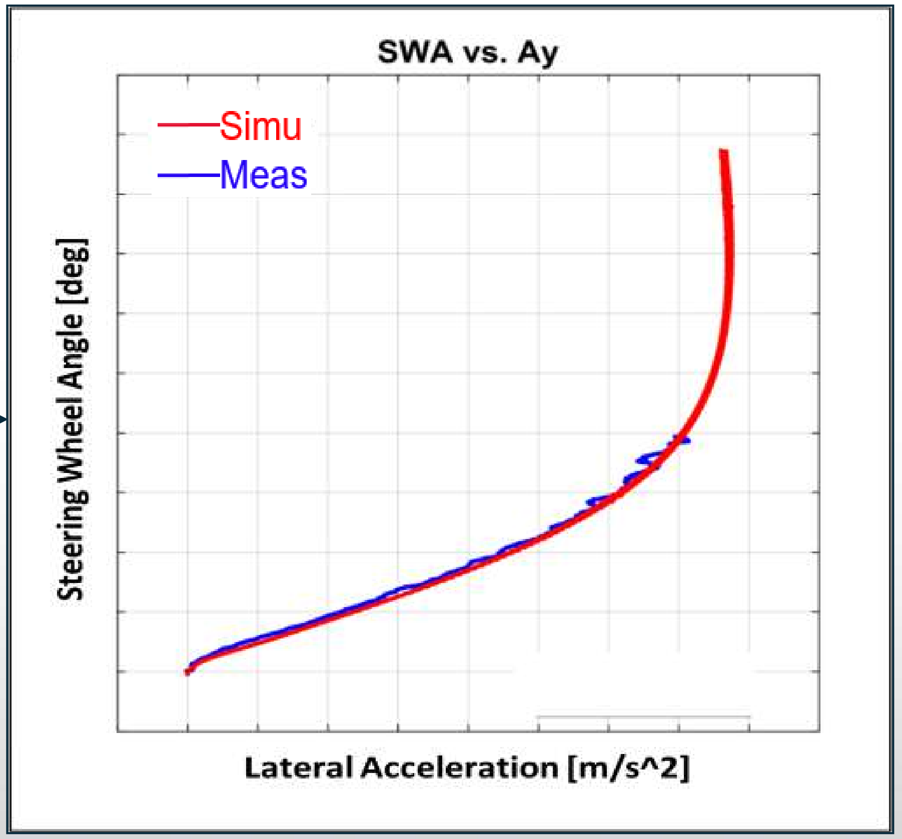


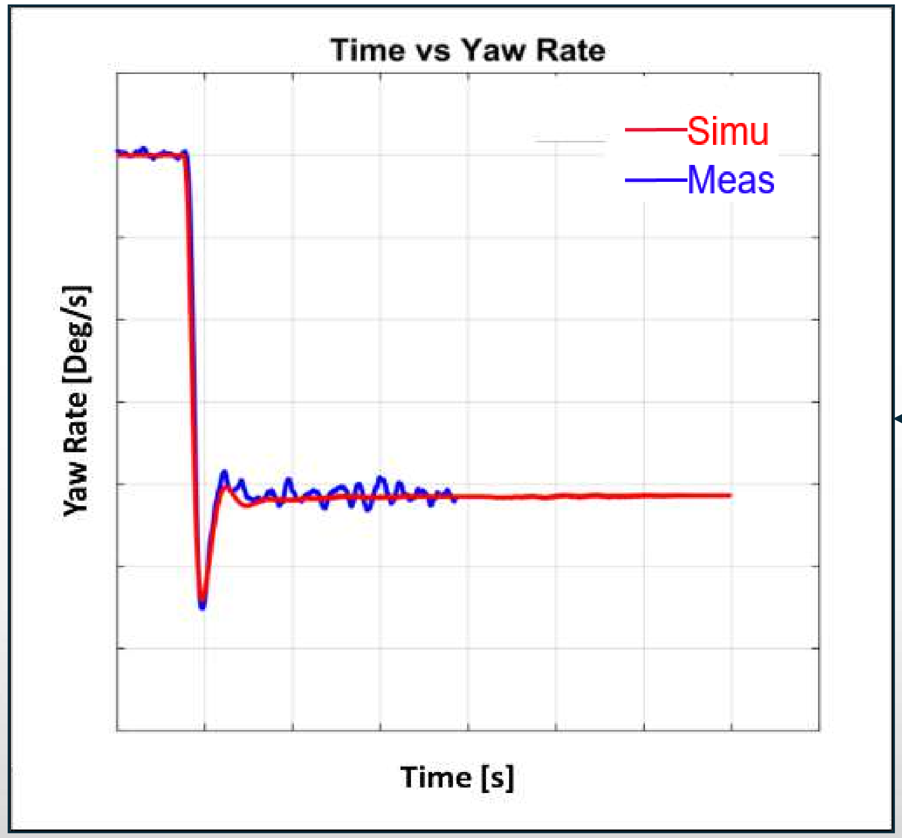


Rear Suspension Wheel Rate correlation

Full Vehicle level

SWA vs Lateral Acceleration correlation





Yaw rate vs Time correlation

Selection of DVP for MDO Based on Customer Perception

ISO	Test	Metrics	Units
ISO 4138	Steady-state cornering - Const. speed (continuous) -100kmph kmph	Steering wheel angle gradient	(deg/[m/s^2])
		Roll gradient	(deg/[m/s^2])
		Body side slip gradient	(deg/[m/s^2])
ISO 7401	Step Steer-100kmph (~+/- 4 [steady state])	Ay lateral 90% time	(sec)
		Ay Over shoot	(%)
		Yaw Rate 90% time	(sec)
		Yaw rate Over shoot	(%)
As per OEM	Straight line driving on deterministic inputs - 40 kmph	Seat Acceleration Ax	[m/s^2]
		Seat Acceleration Az	[m/s^2]

Evaluation Items	*Customer Perception*
Handling	Sporty/fun
Stability	confidence/Safety
Comfort	Ride

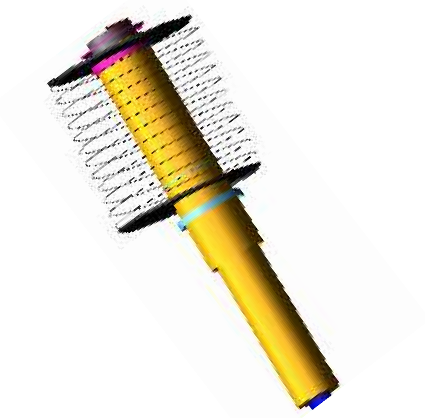
Subjective feel Mapping with Objective Data

Subjective feel	Objective metrics
Sporty/fun (Response, Agility)	Steering Wheel Angle Gradient Lateral acceleration Response time 90% Yaw Rate Response time 90%
Confidence/Safety (Body control/Yaw stability)	Roll gradient & Side slip gradient
Ride comfort (Deterministic impacts)	Seat Longitudinal and Vertical Acceleration

Source: Technical report from Idiada

DOE Analysis for Full Vehicle Analysis

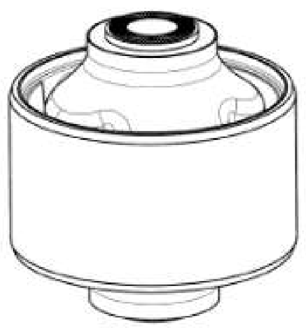
Design Variable for DOE Study



Front shock absorber

Rear Damper

Rear spring



Bush Stiffness

Number of input variables: 26

Workflow for DOE analysis

Common design space (Input Variables)

Load case 1
SSV

Load case 2
Step Steer

Load case 3
Ride

Compiled output in mF
(Input Variables & Responses)

RSM developed for the response

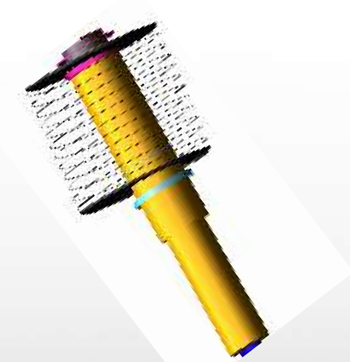
Number of DOE runs : 62

Design Variables – Feature Selection

TIEROD_HP_Z	TIEROD HARD POINT Z DIRECTION
LCA_OBJ_HP_Z	LCA OUTER BALL JOINT HARD POINT Z DIRECTION
RS_SPRING_STIFF	REAR SUSPENSION COIL SPRING STIFFNESS
TIRE_LKY	TIRE CORNERING STIFFNESS
FS_SPRING_STIFF	FORNT SUSPENSION COIL SPRING STIFFNESS
FS_ARB_DIA	FORNT SUSPENSION ANTI ROLL BAR STIFFNESS
RS_BS_CLR	REAR SUSPENSION BUMP STOPPER CLEARANCE
TB_TO_BIW_KX	TWIST BEAM TO BIW BUSH X-DIRECTION STIFFNESS
FS_DAMPER_STIFF_COMPR	FORNT SUSPENSION DAMPER STIFFNESS-COMPRESSION
FS_DAMPER_STIFF_REBOUND	FORNT SUSPENSION DAMPER STIFFNESS-REBOUND

Functional Attributes

- Roll gradient
- Side slip gradient
- Steering Wheel Angle Gradient
- Response time 90% of Ay
- Response time 90% of Yaw Rate
- Floor Vertical and Longitudinal impact Shocks

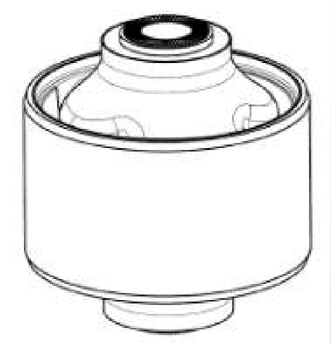


Front shock absorber

Rear Damper

Rear spring

Anti Roll Bar

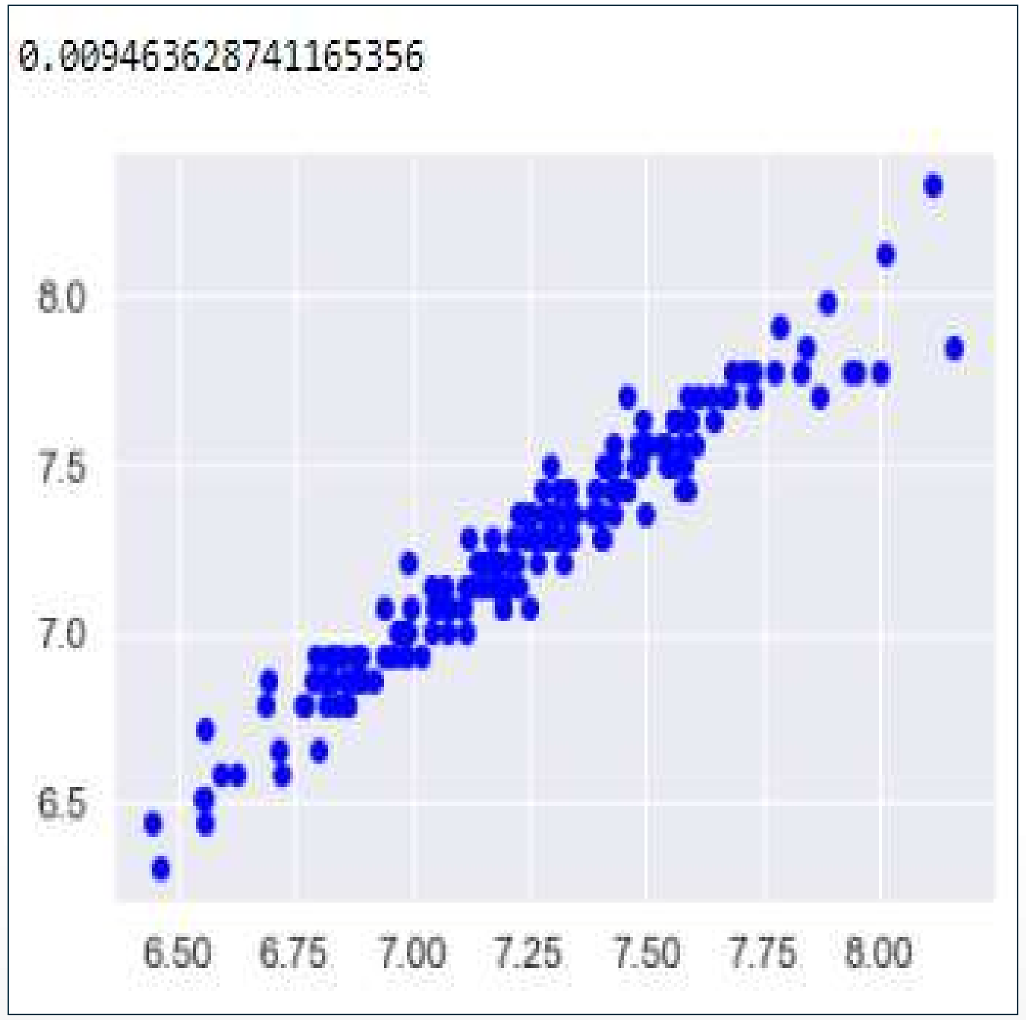


Bush Stiffness

Regression Based - Functional Forms **– Steady State Velocity**

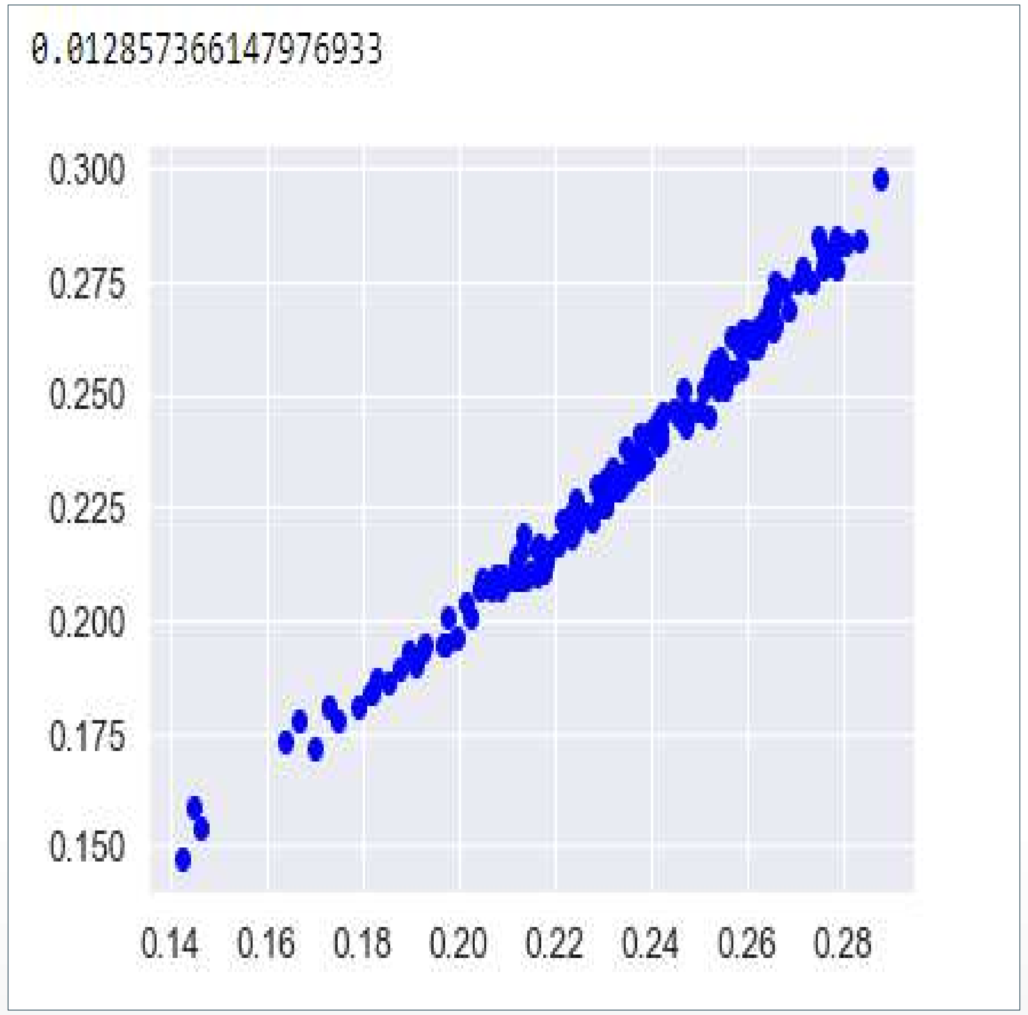
ESTECO USERS' MEETING INDIA

Steering Wheel gradient



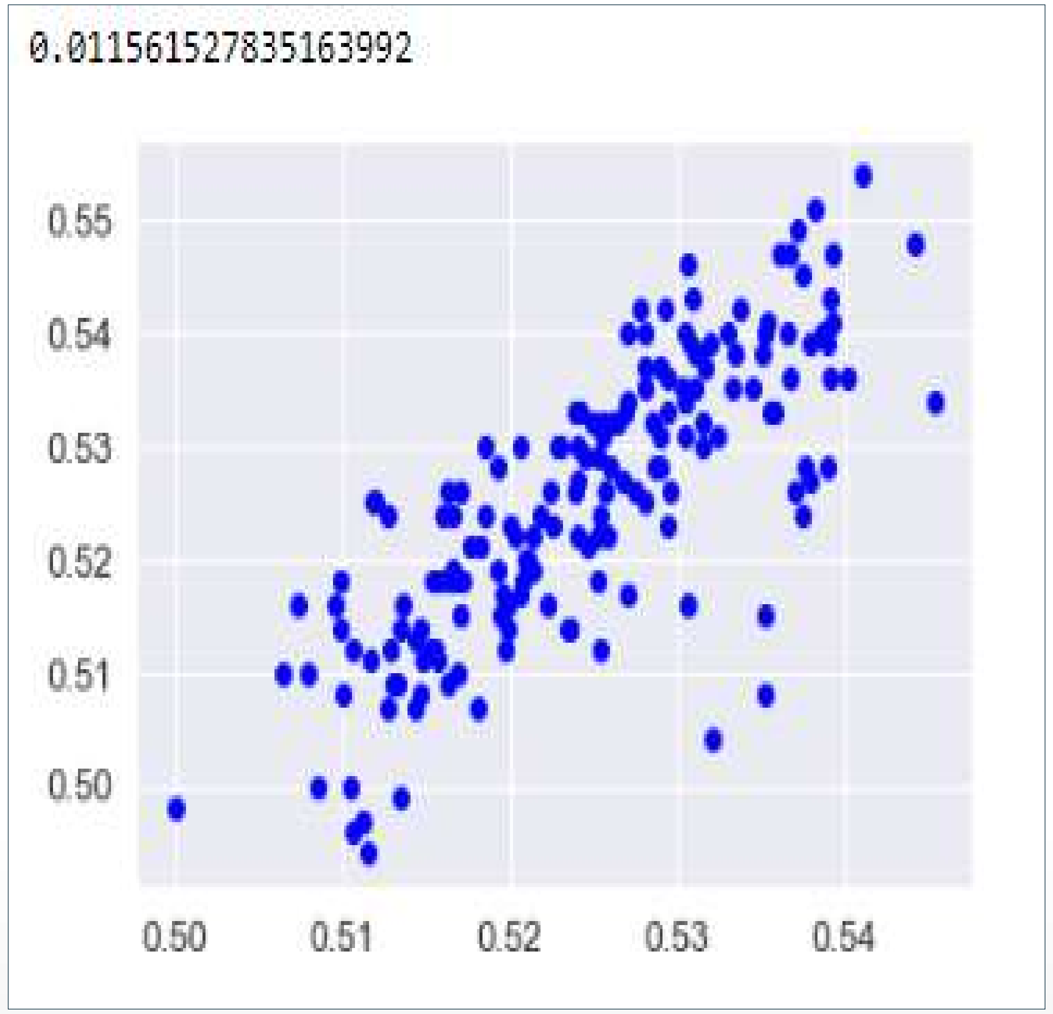
Accuracy: 99.05%

SIDE SLIP GRADIENT



Accuracy: 98.72%

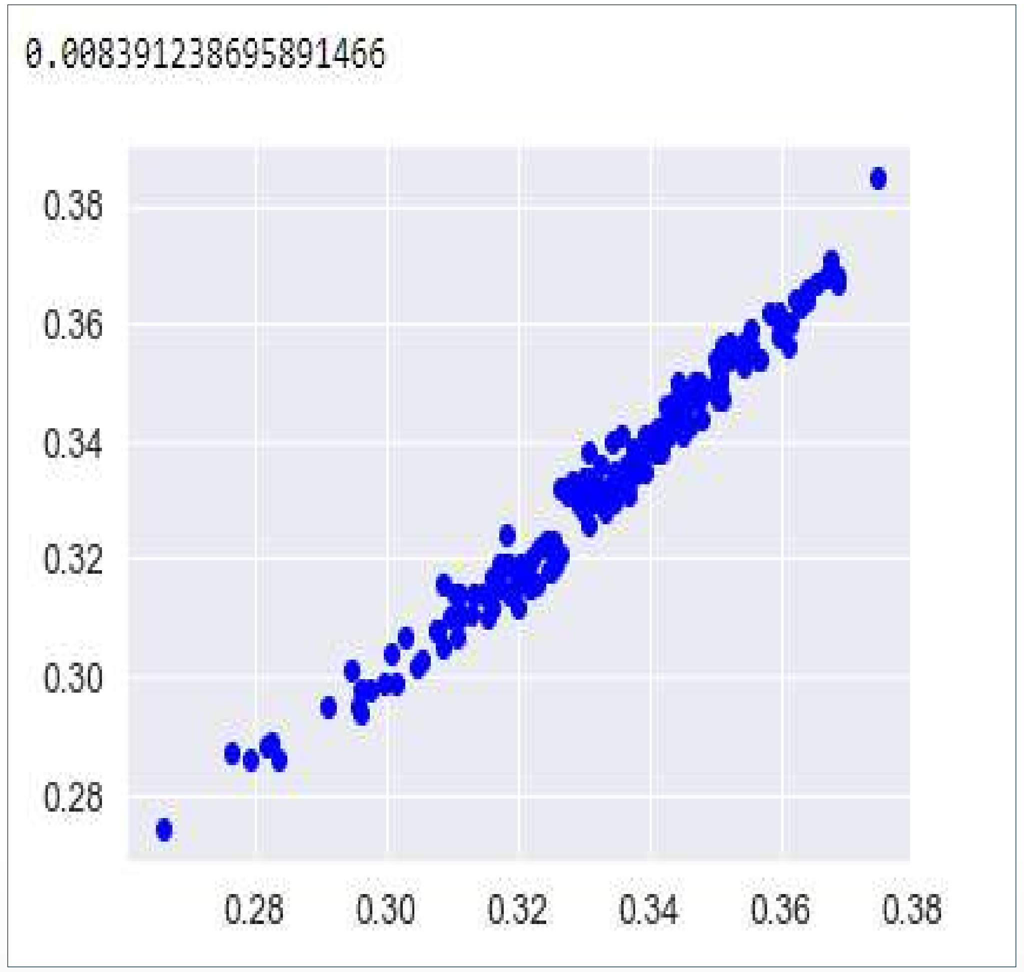
Roll GRADIENT



Accuracy: 98.84%

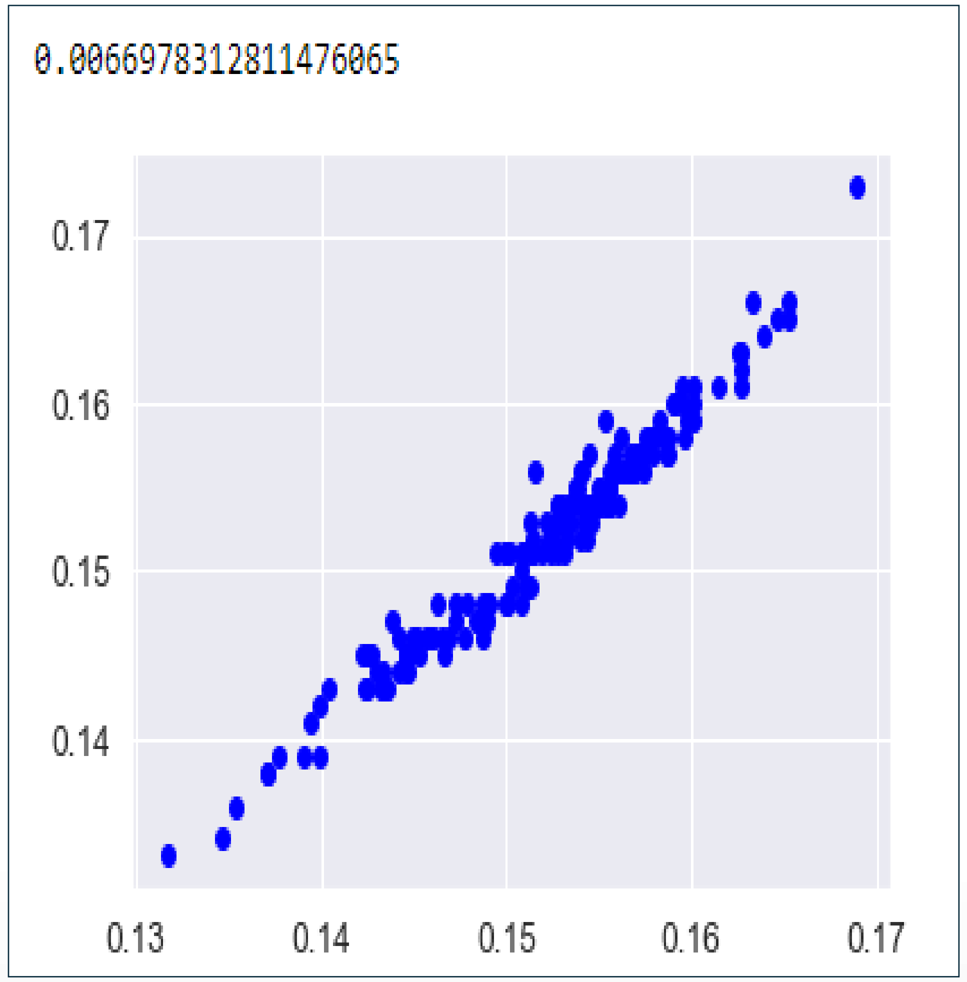
Regression Based - Functional Forms – **Step Steer**

Lateral Acc Response 90% time



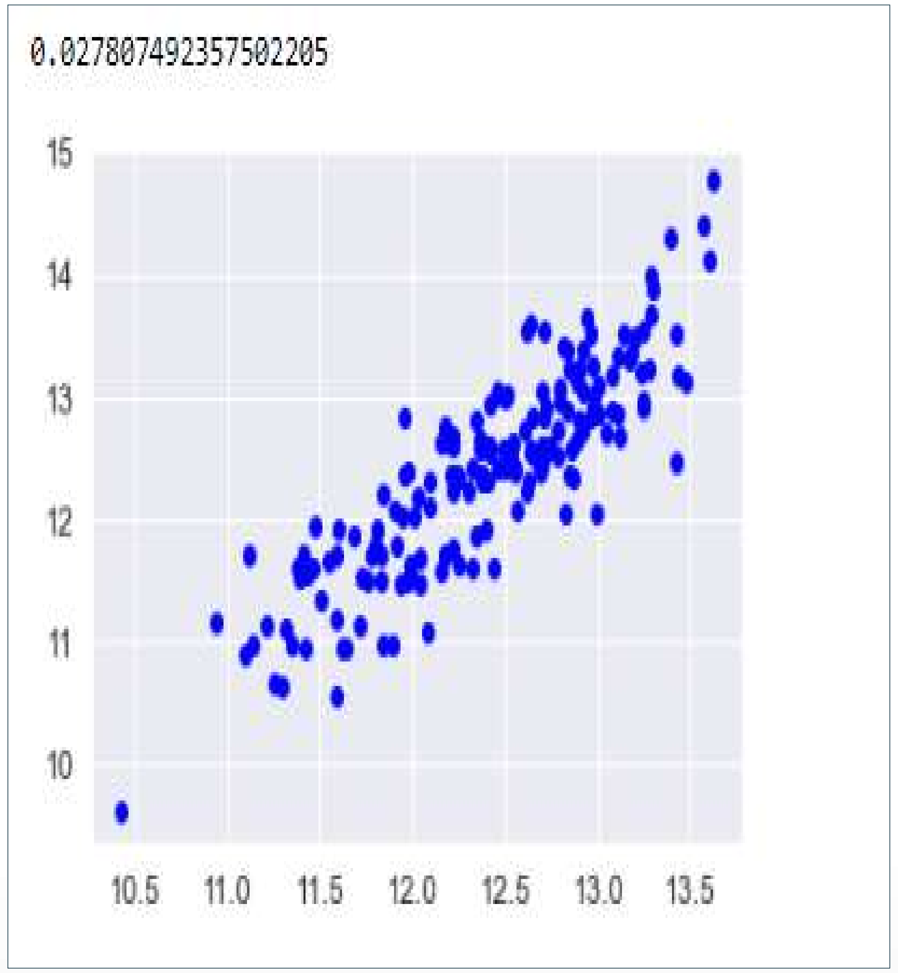
Accuracy: 99.2%

Yaw Rate Response 90% time



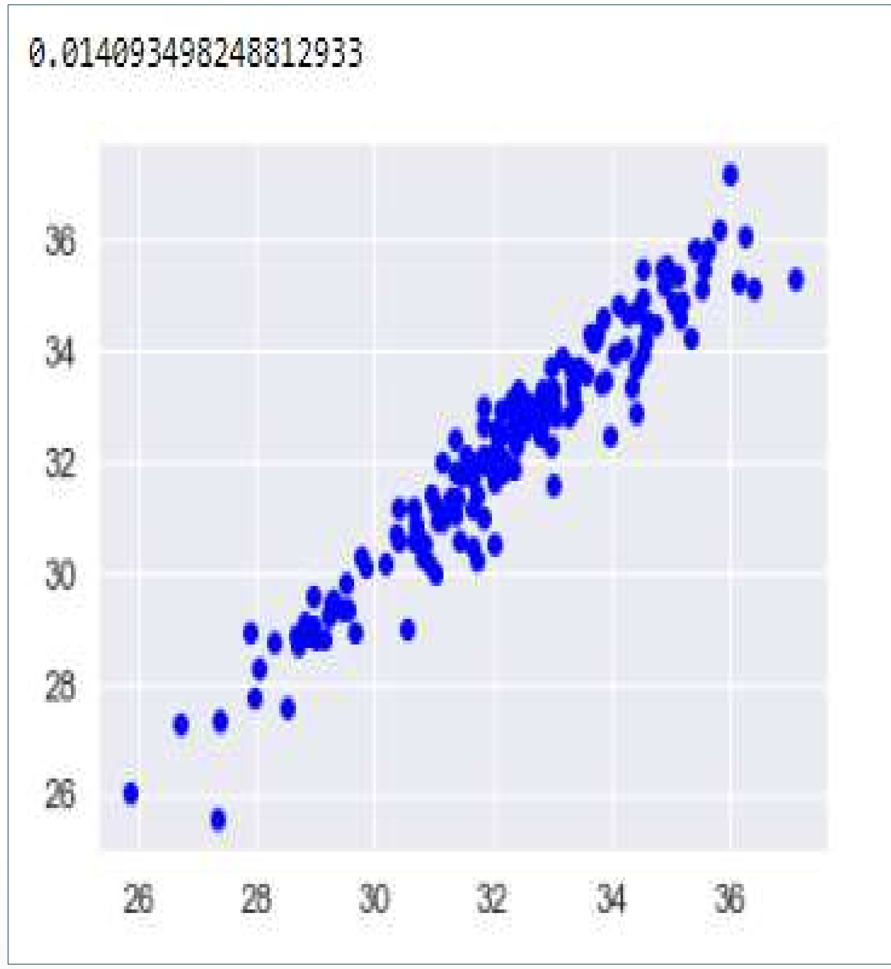
Accuracy: 99.3%

Lateral Acc Overshoot



Accuracy: 97.3%

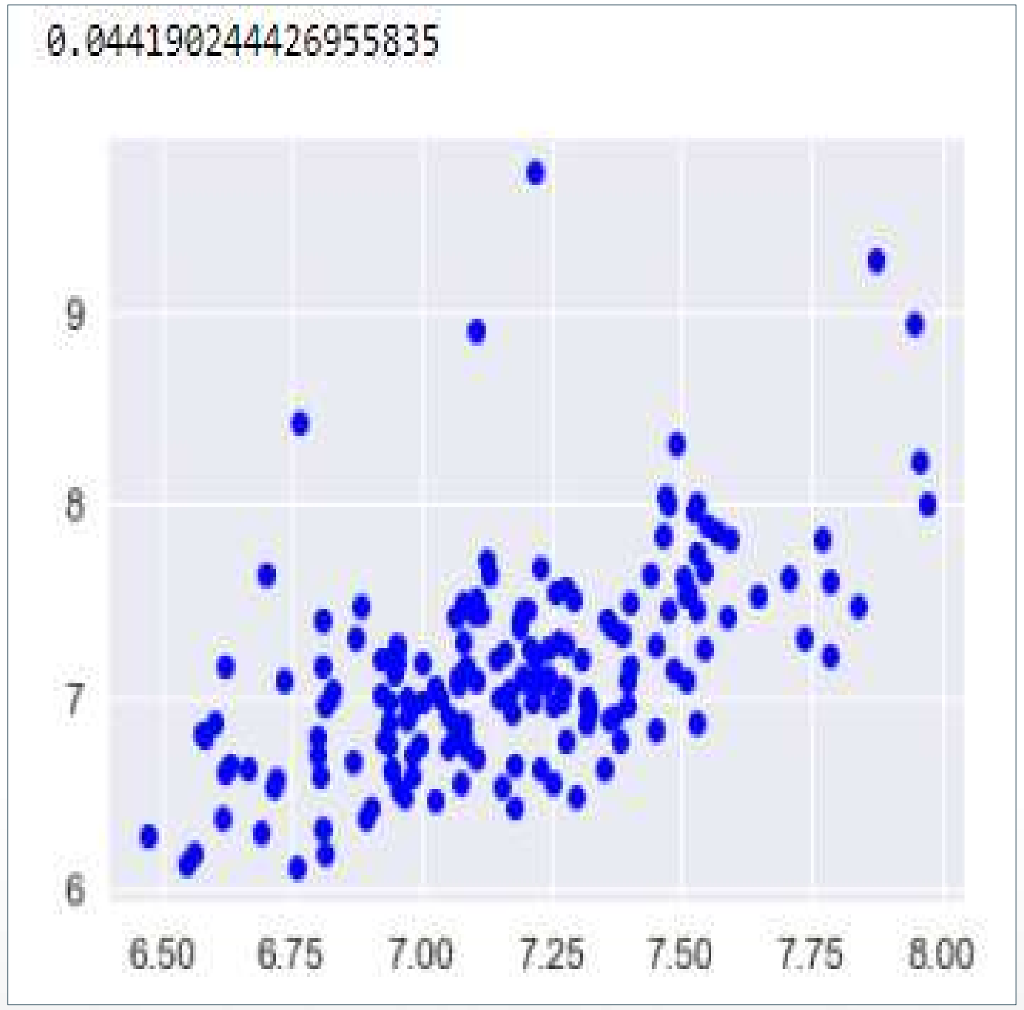
Yaw Rate Overshoot



Accuracy: 98.6%

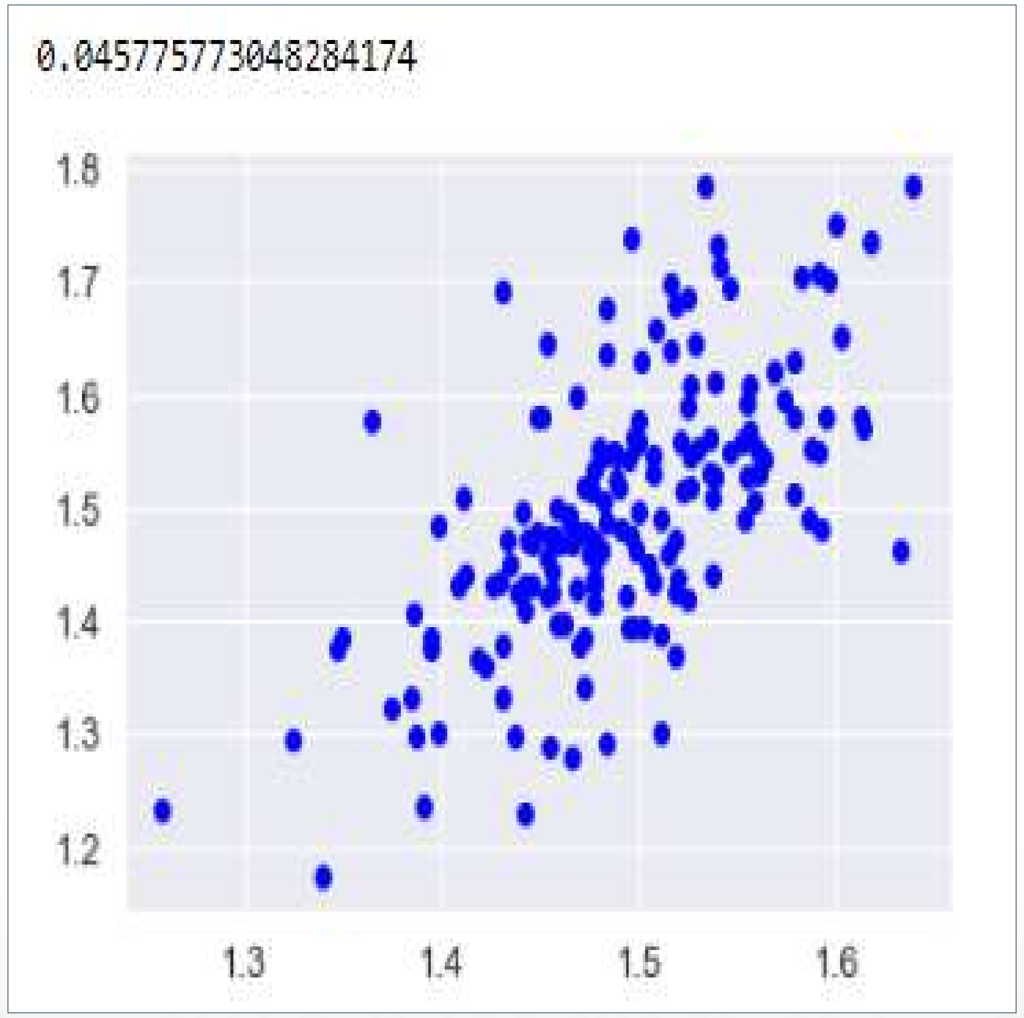
Regression Based Functional Forms – Ride

Driver Seat Acc Az



Accuracy: 95.6%

Driver Seat Acc Ax



Accuracy: 95.4%

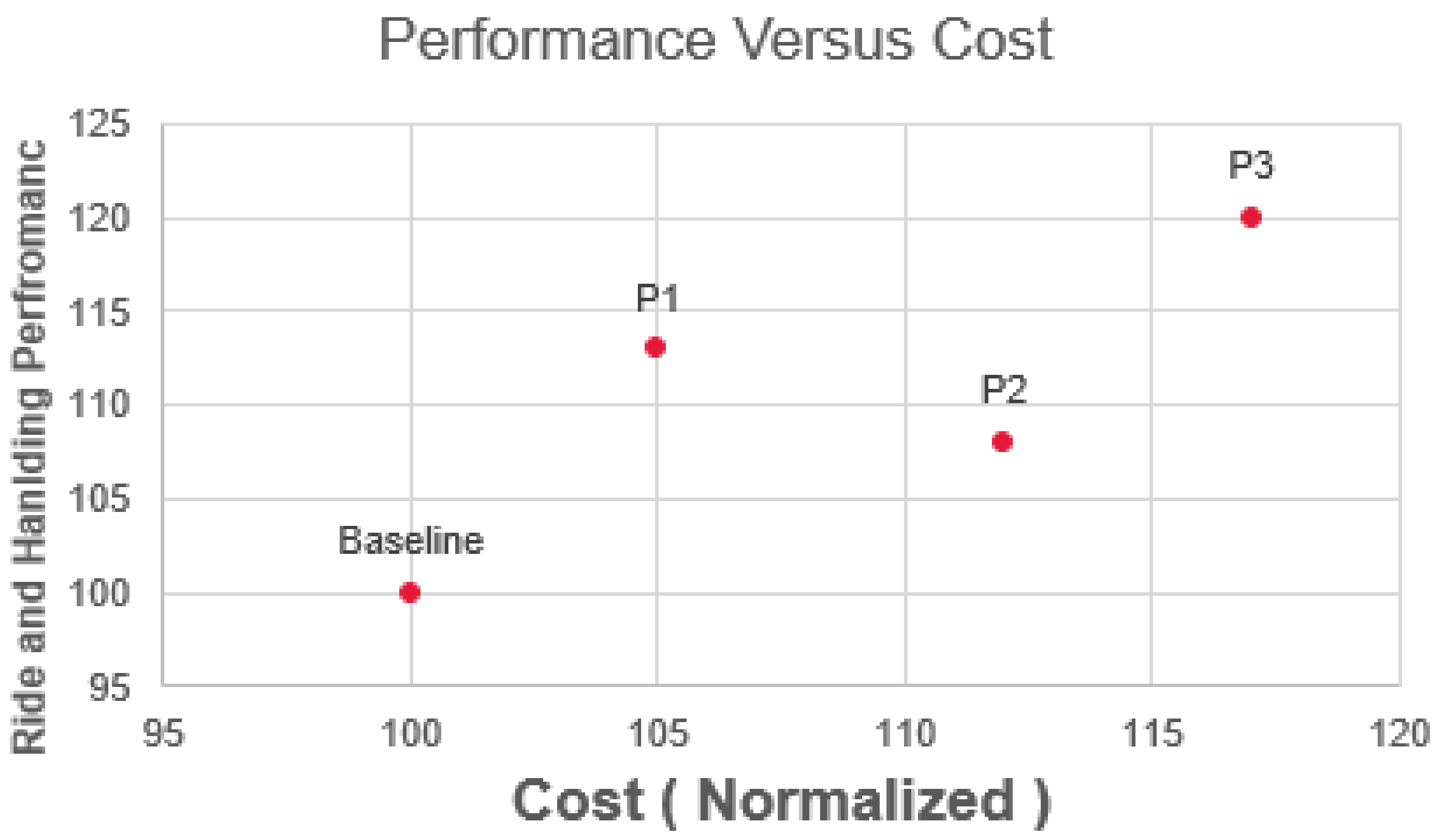
Multi-Objective Optimization – Python Code


```
In [398]: import pickle
          import numpy as np

In [399]: con1 = pickle.load(open('con1', 'rb'))
          con1_coef = con1.coef_
          con1_coef = np.append(con1_coef, con1.intercept_)
          con1_coef

Out[399]: array([ 0.11449349, -0.09910419, -1.41979803, -1.3516054 ,  4.93800272])

In [400]: con2 = pickle.load(open('con2', 'rb'))
          con2_coef = con2.coef_
          con2_coef = np.append(con2_coef, con2.intercept_)
          con2_coef

Out[400]: array([-0.23353778,  0.05795458, -0.03145311,  0.5465709 ])

In [401]: con3 = pickle.load(open('con3', 'rb'))
          con3_coef = con3.coef_
          con3_coef = np.append(con3_coef, con3.intercept_)
          con3_coef

Out[401]: array([-0.06901486, -0.0015768 ,  0.00159881,  0.33193544])

In [402]: res1 = pickle.load(open('res1', 'rb'))
          res1_coef = res1.coef_
          res1_coef = np.append(res1_coef, res1.intercept_)
          res1_coef

Out[402]: array([-7.81667412e-02, -4.87137958e-06,  2.96552367e-04,  7.28833393e-01])

In [403]: res2 = pickle.load(open('res2', 'rb'))
          res2_coef = res2.coef_
          res2_coef = np.append(res2_coef, res2.intercept_)
          res2_coef

Out[403]: array([-0.34982668,  0.06121437, -0.01858989,  0.55040695])

In [404]: res3 = pickle.load(open('res3', 'rb'))
          res3_coef = res3.coef_
          res3_coef = np.append(res3_coef, res3.intercept_)
```

```
In [445]: from pymoo.core.problem import ElementwiseProblem

          class OptimizationProblem(ElementwiseProblem):

              def __init__(self, con, res):
                  super().__init__(n_var = 12,
                                   n_obj = 3,
                                   n_constr = 4,
                                   xl = np.array([-100.3, -166.8, 0.8, 0.9, 0.8, 27877.1, 63.6, 0.8, 0.8, 0.8, 0.5, 0.5]),
                                   xu = np.array([-94.3, -160.9, 1.2, 1.3, 1.2, 32122.9, 77.7, 1.2, 1.2, 1.2, 1, 1]))

              def _evaluate(self, x, out, *args, **kwargs):
                  f1 = (res[0][0]*x[4] + res[0][1]*x[5] + res[0][2]*x[6] + res[0][3]) + (res[1][0]*x[3] + res[1][1]*x[2] + res[1][2]*x[7]

                  f2 = (res[4][0]*x[3] + res[4][1]*x[1] + res[4][2]*x[4] + res[4][3]) + (res[5][0]*x[9] + res[5][1]*x[8] + res[5][2]*x[3]

                  f3 = -(x[10]*f1 + x[11]*f2)

                  g1 = (con[0][0]*x[0] + con[0][1]*x[1] + con[0][2]*x[2] + con[0][3]*x[3] + con[0][4])  - 6.75
                  g2 = (con[1][0]*x[3] + con[1][1]*x[2] + con[1][2]*x[4] + con[1][3]) - 0.35
                  g3 = (con[2][0]*x[3] + con[2][1]*x[0] + con[2][2]*x[1] + con[2][3]) - 0.19
                  g4 = ((x[10] + x[11])) - 1

                  out["F"] = [f1, f2, f3]
                  out["G"] = [g1, g2, g3, g4]

In [446]: from pymoo.algorithms.moo.nsga2 import NSGA2
          from pymoo.factory import get_sampling, get_crossover, get_mutation

In [447]: problem = OptimizationProblem(con, res)

In [468]: algorithm = NSGA2(pop_size = 10,
                            n_offsprings = 5,
                            sampling = get_sampling("real_random"),
                            crossover = get_crossover("real_sbx", prob = 0.9, eta = 15),
                            mutation = get_mutation("real_pm", eta = 20),
                            eliminate_duplicates = True)

In [469]: from pymoo.optimize import minimize
          from pymoo.factory import get_termination

          termination = get_termination("n_gen", 400)
```

13 % improvement in Ride and Handling Performance is achieved w.r.t Baseline Design – Proposal 1 (P1)

Results



- **Proposal 1 (P1) – Optimal Solution**
- **Proposal 2 (P2) – Baseline + FSD**
- **Proposal 3 (P3) - Optimal Solution + FSD**

Baseline design can be upgraded with the Optimized Solution (P1)

Proposal 2 – Baseline with FSD can be dropped

Proposal 3 – Can be implemented for Luxury Segment

Need for Robust Design:
Variation in the Subjective Evaluation Metric

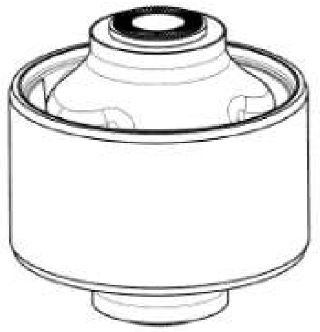
ESTECO
USERS' MEETING
INDIA

Front shock absorber

Rear Damper

Rear spring

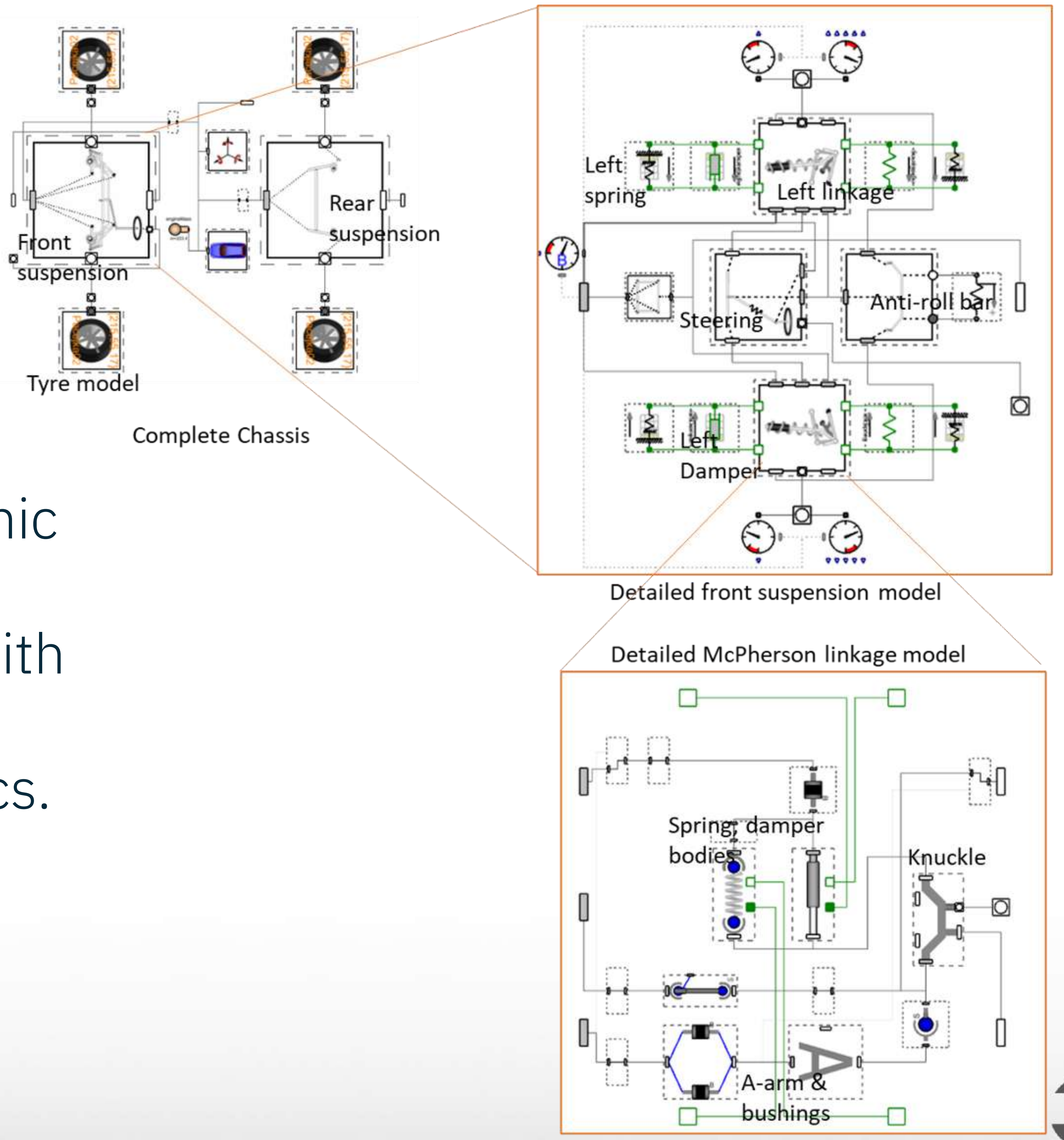
Anti Roll Bar



Bush Stiffness

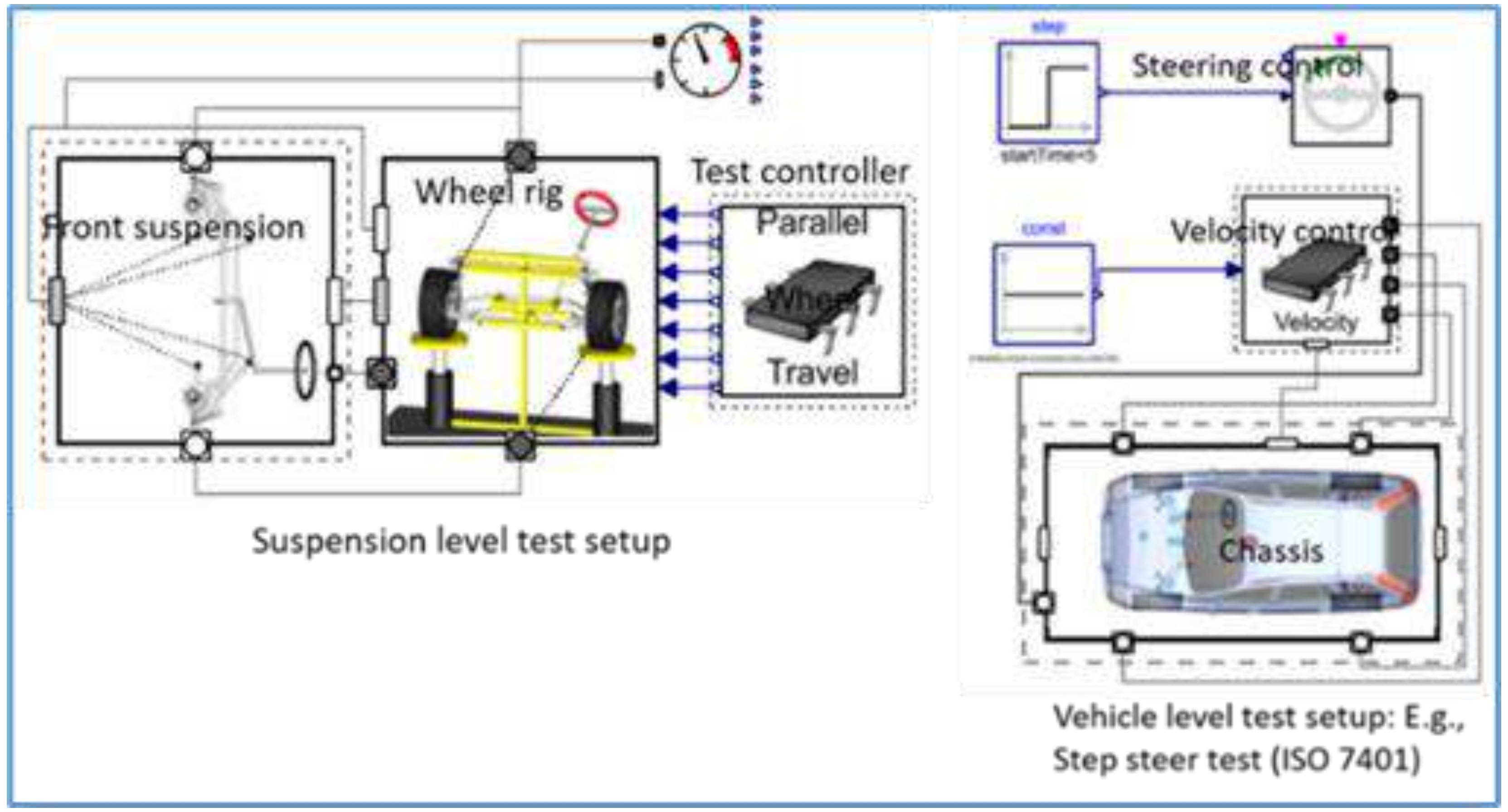
Model Setup

- A vehicle model has been setup on a Modelica framework tool called Modelon Impact and Vehicle Dynamics Library.
- The model incorporates detailed multibody based suspension linkages, spring/dampers, steering, antiroll bars, tyres, body & aerodynamic properties
- The model components were parameterised with parameters like hardpoints, mass, inertia properties, force splines and tyre characteristics. This vehicle is fit with front McPherson suspension and rear twist beam suspension models



Model Testing and Correlation

- The suspension models are unit tested using test rigs and correlated against reference model results. Similarly, actual chassis model is tested using popular handling manoeuvres and correlated
- The individual suspension models were tested using suspension test rigs for their Kinematics and Compliance performance correlation against a reference model
- After testing individual suspensions, vehicle level tests were performed by driving the chassis model using velocity and steer robots. No powertrain components were used in vehicle level tests
- The suspension and vehicle level relation achieved was deemed to be good and sufficient to proceed with equivalent electric vehicle model building and analysis



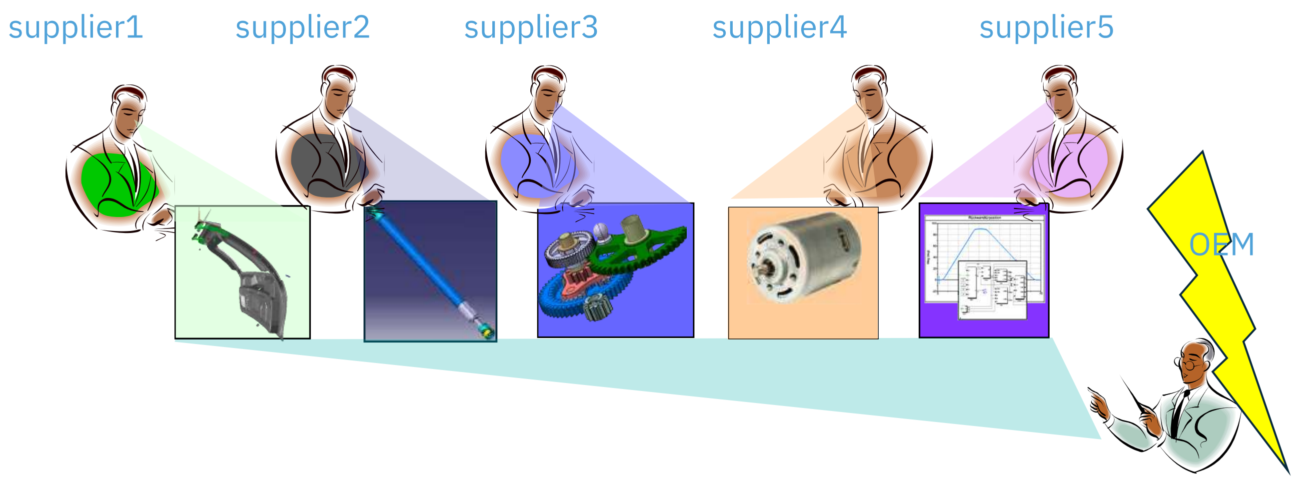
Suspension level test setup

Vehicle level test setup: E.g., Step steer test (ISO 7401)

Background to FMI

Problem

- Due to different applications, models of a system often must be developed using different programs (modeling and simulation environments)



- In order to simulate the system, the different programs must interact with each other
- This makes model exchange a necessity

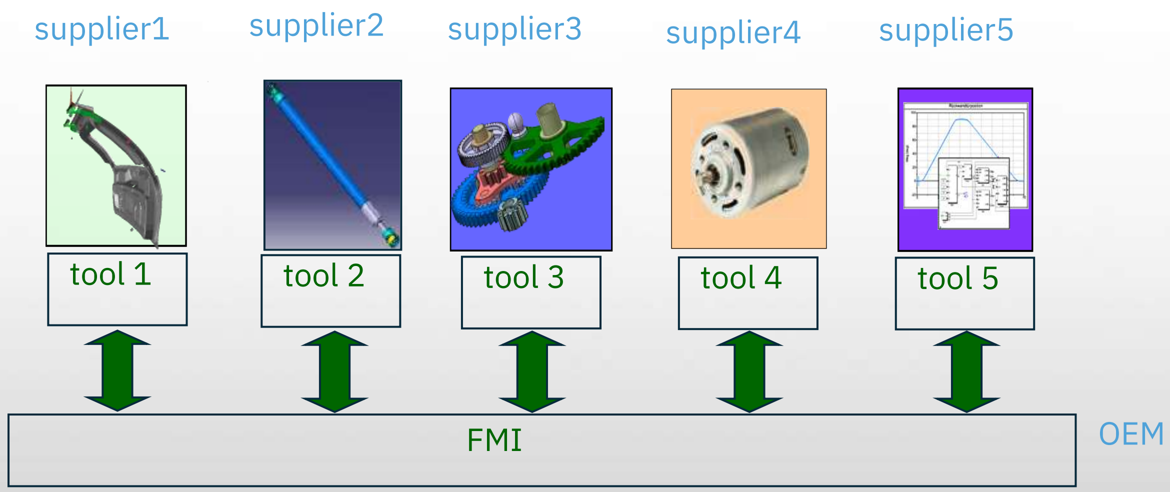
Background to FMI

Standardization

- Even though many tool suppliers provide their own specific solutions (interfaces) for the model exchange, a standardized "tool independent" approach is desirable
- Requirements:
  - The standard should be open
  - Easy to implement both in tools and for end users
  - Safe and seamless deployment - in-house and to suppliers
  - Allow for customization

Solution

- As a universal solution to this problem the **Functional Mock-up Interface** (FMI) was developed by MODELISAR



What is FMI?

FMI – Functional Mock-up Interface

Open interface standard for model exchange between different modeling and simulation environments.

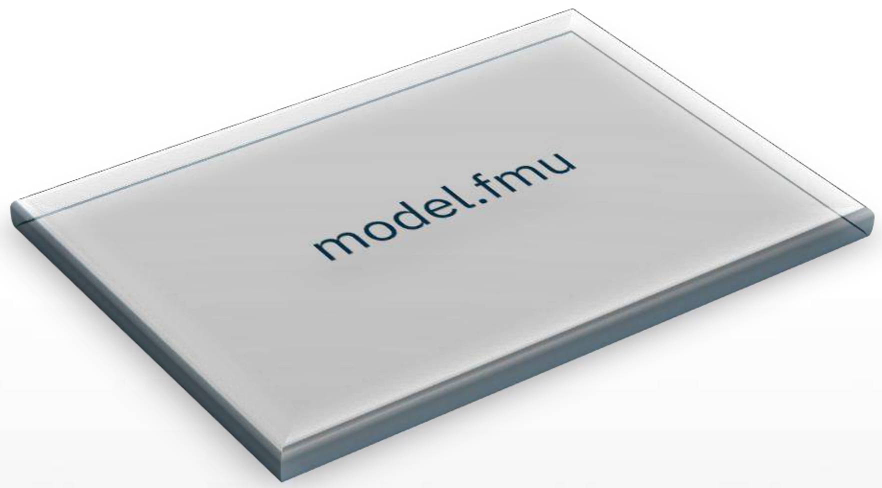
Consists of:

- Model Interface: Set of C functions for equation evaluation.
- Model Description Schema: XML Schema defining an XML file containing variable definitions and model meta data.
- Model File Distribution: The contents definition for a file (the FMU) that contains at a minimum the above two items.

What is AN FMU?

FMU – Functional Mock-up Unit

- Implementation of the FMI
- Zip-file with the file ending .fmu
- Can be run to get simulation results



FMU File Contents

- modelDescription.xml
  - A list of all the variables available during simulation
  - General meta data for how to run the FMU
- binaries/
  - FMI implementation for specific platforms. The FMU is run by calling these functions.
- sources/
  - C source code for the FMU
- resources/
  - Resources required by the FMU. Can be any format.
- documentation/
- model.png
  - Optional FMU Icon

BINARIES AND SOURCES

- The binary format differs between different platforms:
    - Windows binaries (.dll)
        - 32-bit, 64-bit
    - Linux binaries (.so)
        - 32-bit, 64-bit
- Make sure to use an FMU that contains the appropriate binary configuration.
    - Example: 64-bit MATLAB for Windows will only support 64-bit FMUs
    - Possible to have binaries for several configurations in the same FMU
    - More likely that a 32-bit FMU will work with the required applications than a 64-bit FMU (not all machines have support for 64-bit applications)
- Possible to also include the C source code in the FMU
    - Can be compiled to fit any configuration.
    - Compilation in the importing tool.
    - Not supported by all tools

FMU Integration in modeFRONTIER

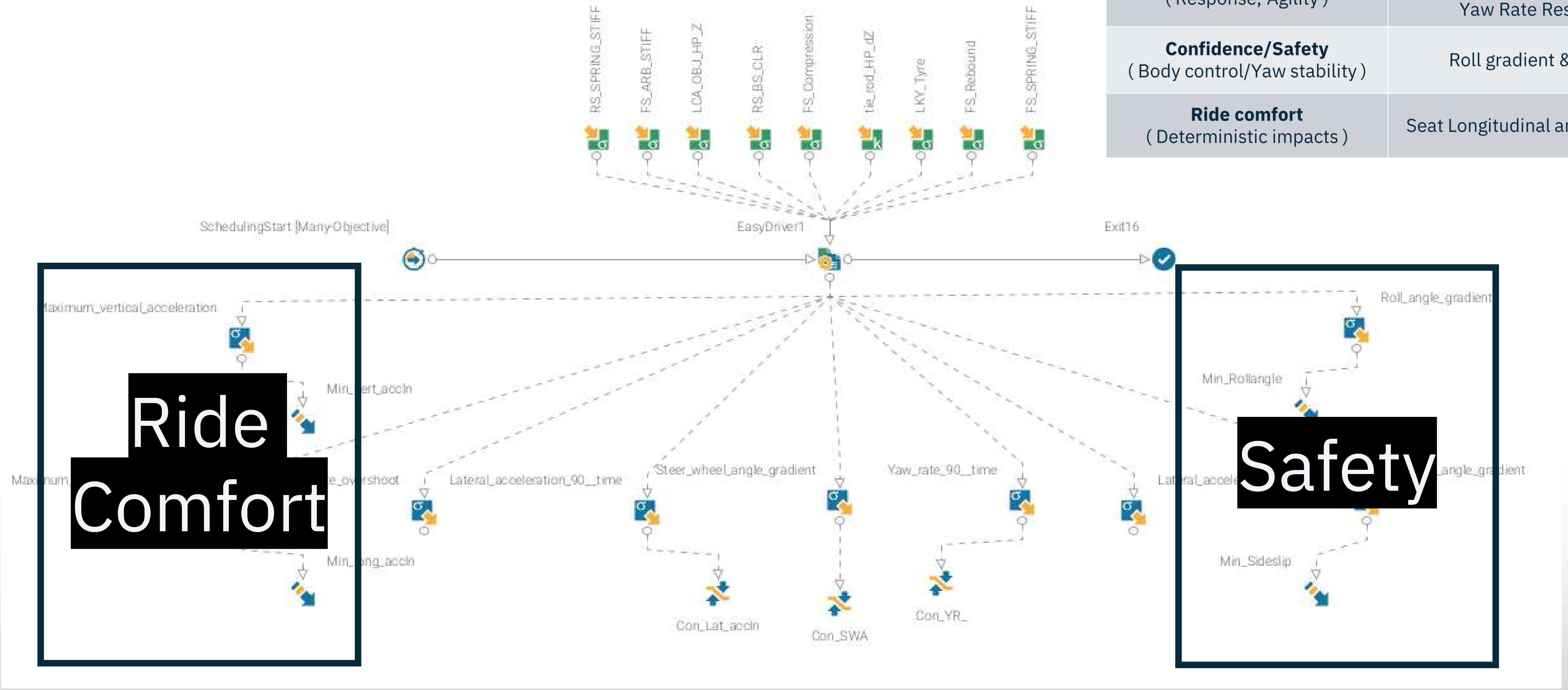
- Modelon Impact FMU is created for each test– Suspension level (Ride and Handling) and Vehicle level (Steep steer)
- Modelon Impact FMU is generated in Python format and added to the Easydriver Input template.
- The essential support files and the syntax to run the python file are added in the Driver tab.
- Output template writes the required outputs based on the python script.

ESTECO
USERS' MEETING
INDIA

um 2023

modeFRONTIER Workflow

Subjective feel	Objective metrics
Sporty/fun (Response, Agility)	Steering Wheel Angle Gradient Lateral acceleration Response time 90% Yaw Rate Response time 90%
Confidence/Safety (Body control/Yaw stability)	Roll gradient & Side slip gradient
Ride comfort (Deterministic impacts)	Seat Longitudinal and Vertical Acceleration



Optimization Formulation

Objective Function:
Minimize **Roll gradient, Side slip gradient, Seat Vertical and Longitudinal impact shacks**

Design Variables: -

RS_SPRING_STIFF	+/- 15 %
FS_ARB_STIFF	+/- 5 %
LCA_OBJ_HP_Z	+/- 5 %
RS_BS_CLR	+/- 5 %
FS_Compression	+/- 15 %
tie_rod_HP_dZ	Constant
LKY_Tyre	+/- 5 %
FS_Rebound	+/- 15 %
FS_SPRING_STIFF	+/- 15 %

Constraints: The following three constraints were used in the present study:

Steering Wheel Angle Gradient [deg/[m/s^2])

Lateral acceleration Response time 90% [s]

Yaw Rate Response time 90% [s]

Multi-Objective Optimization Process

1. DOE Method – Uniform Latin Hypercube (62 samples)
2. Direct Optimization using MANY Algorithm – 450 iterations (12 hours)
3. RSM is constructed over the generated iterations
4. Virtual Optimization using MANY Algorithm – 10000 iterations (Within minutes)
5. Results are compared and validated

Direct Runs – Pareto Solutions

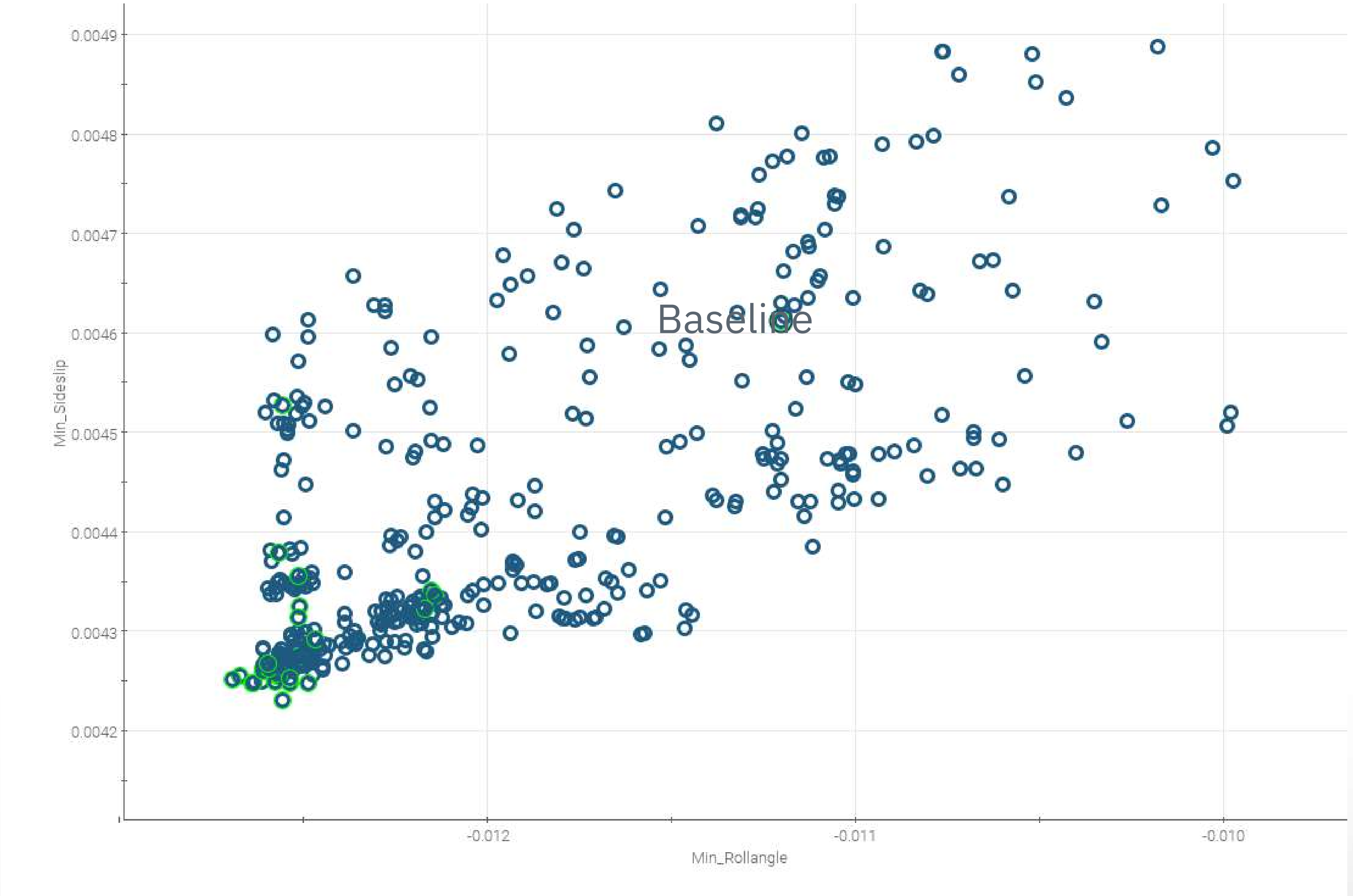


Fig 1. Min Roll Angle Gradient vs Min Side Slip Gradient

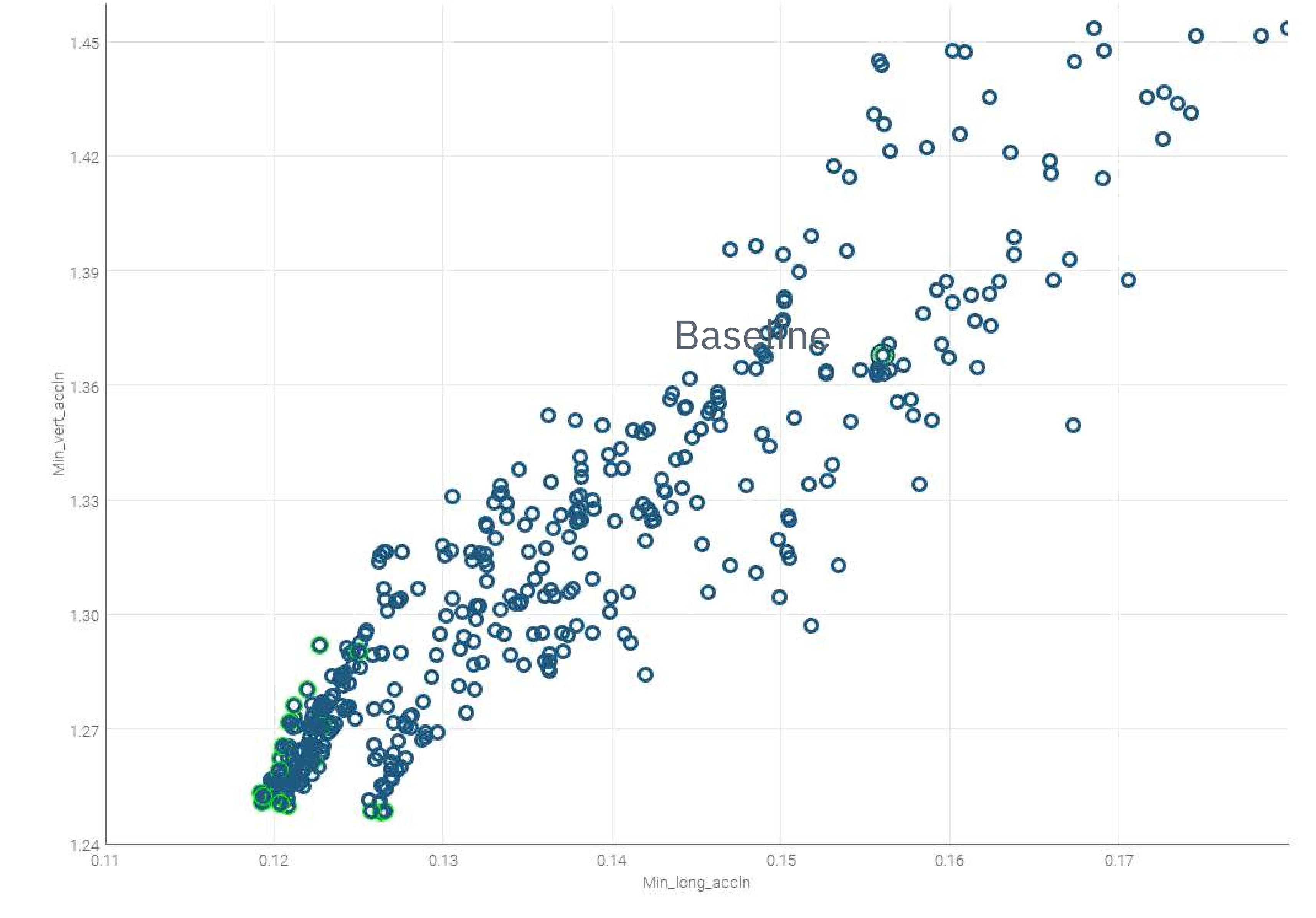


Fig 2. Min Longitudinal Acceleration vs Min Vertical Acceleration

Direct Runs – Bubble 4D chart

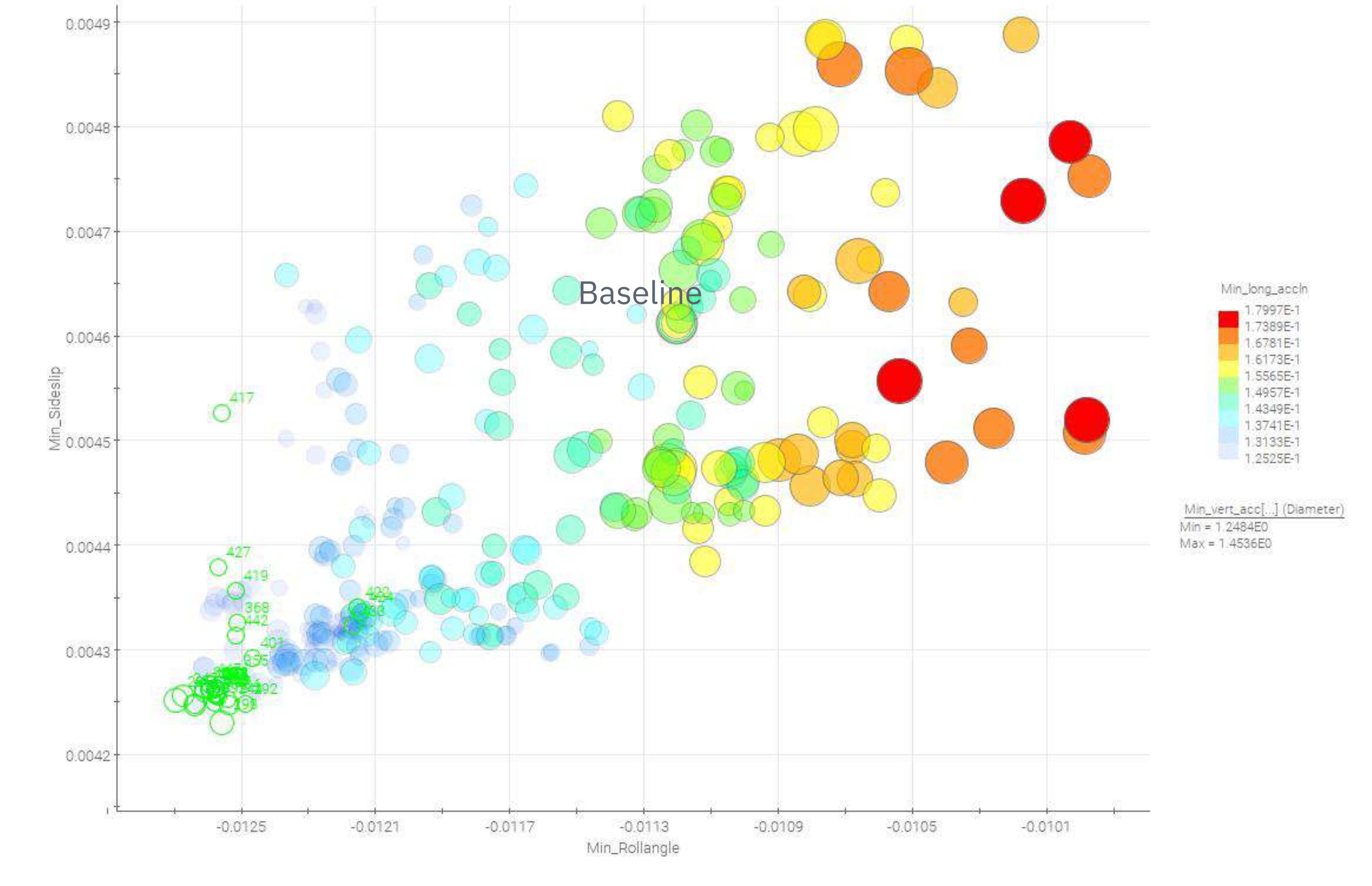
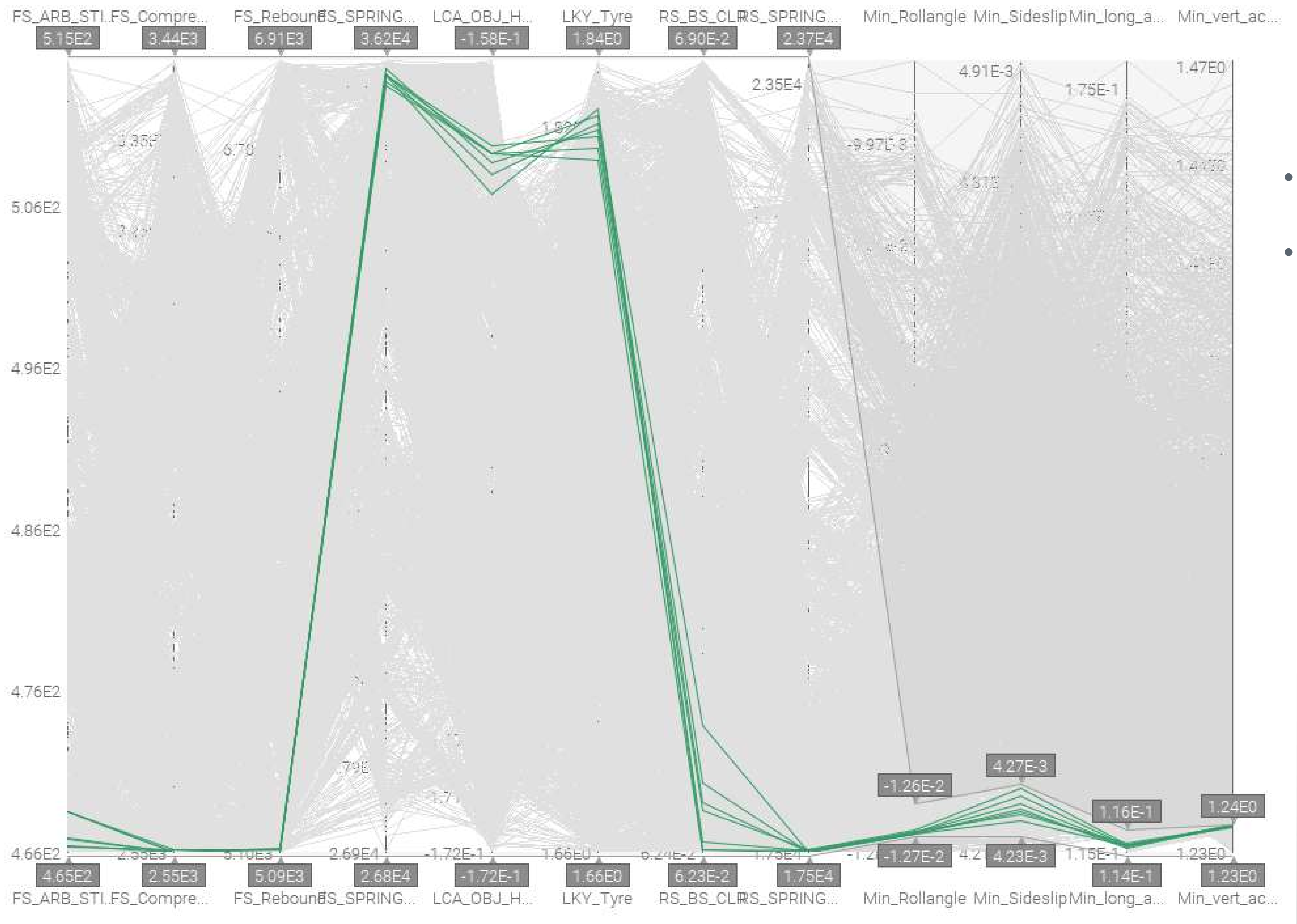


Fig 1. Min Roll Angle Gradient vs Min Side Slip Gradient vs Min Longitudinal Acceleration vs Min Vertical Acceleration

ESTECO
USERS' MEETING
INDIA

Direct Runs - Parallel Coordinates



- Parallel Coordinates chart applied on Input Variables and Objectives
- Filtered designs are taken ahead

Virtual Runs – Pareto Solutions

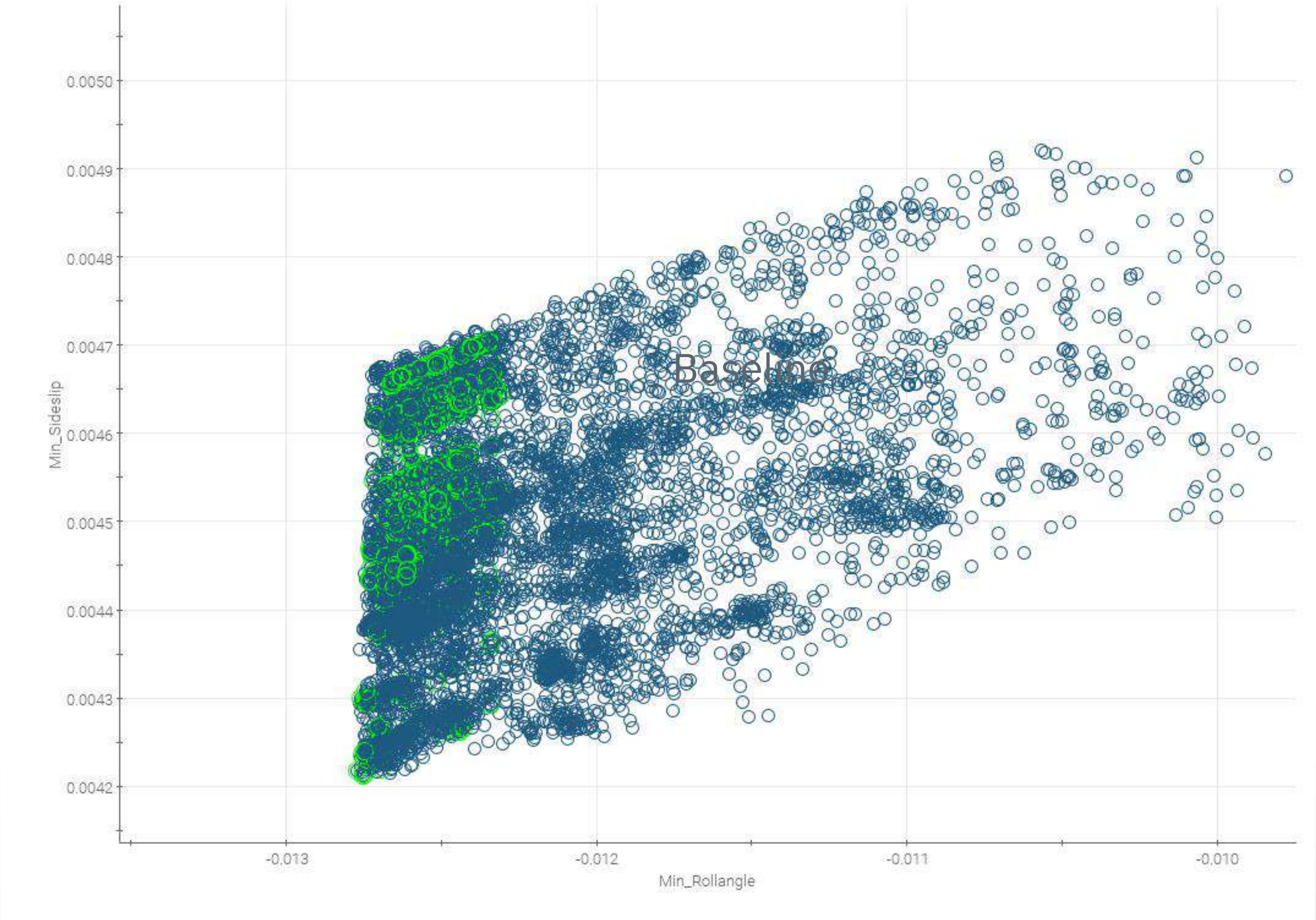


Fig 1. Min Roll Angle Gradient vs Min Side Slip Gradient

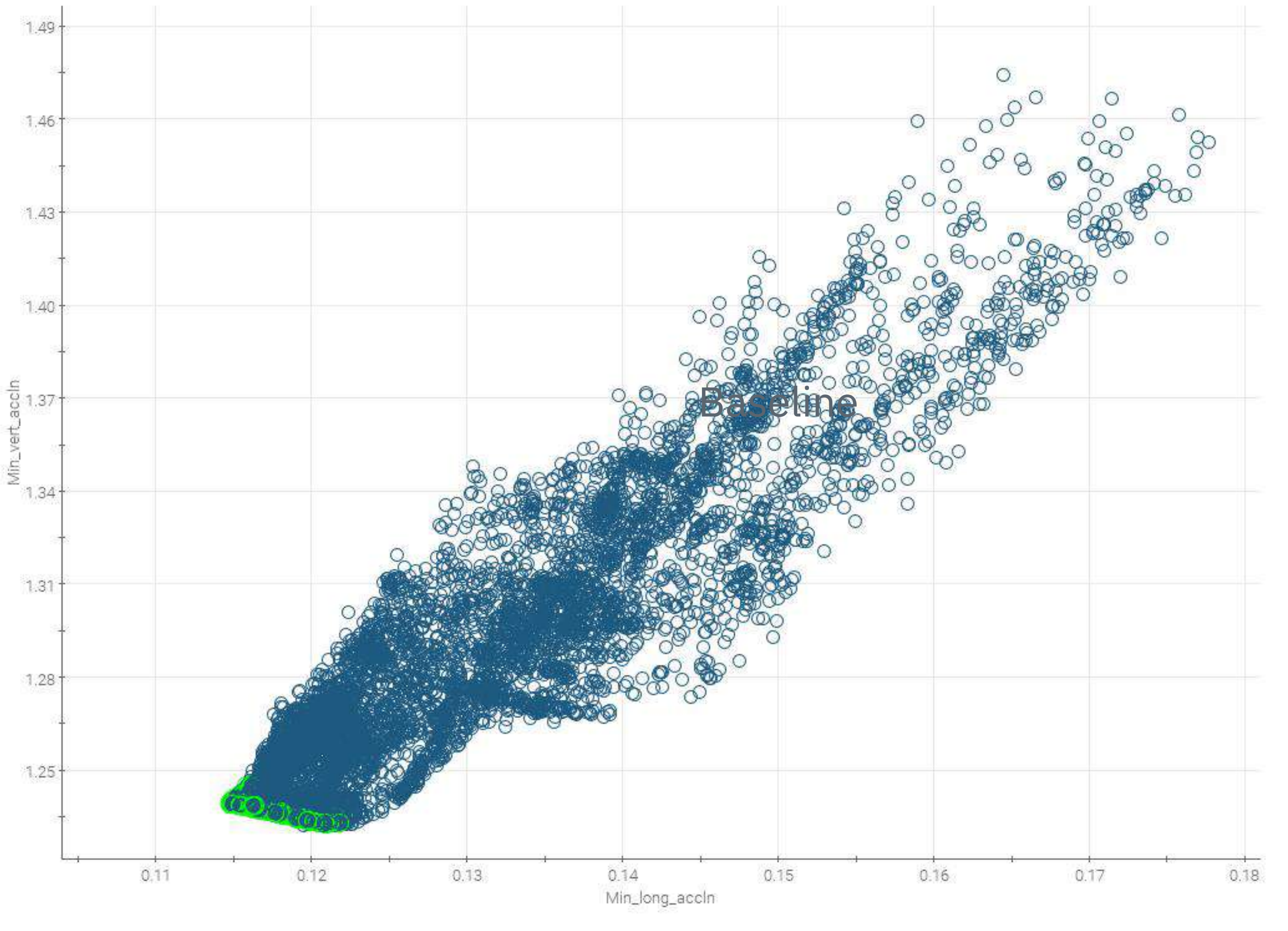


Fig 2. Min Longitudinal Acceleration vs Min Vertical Acceleration

Virtual Runs – Bubble 4D chart

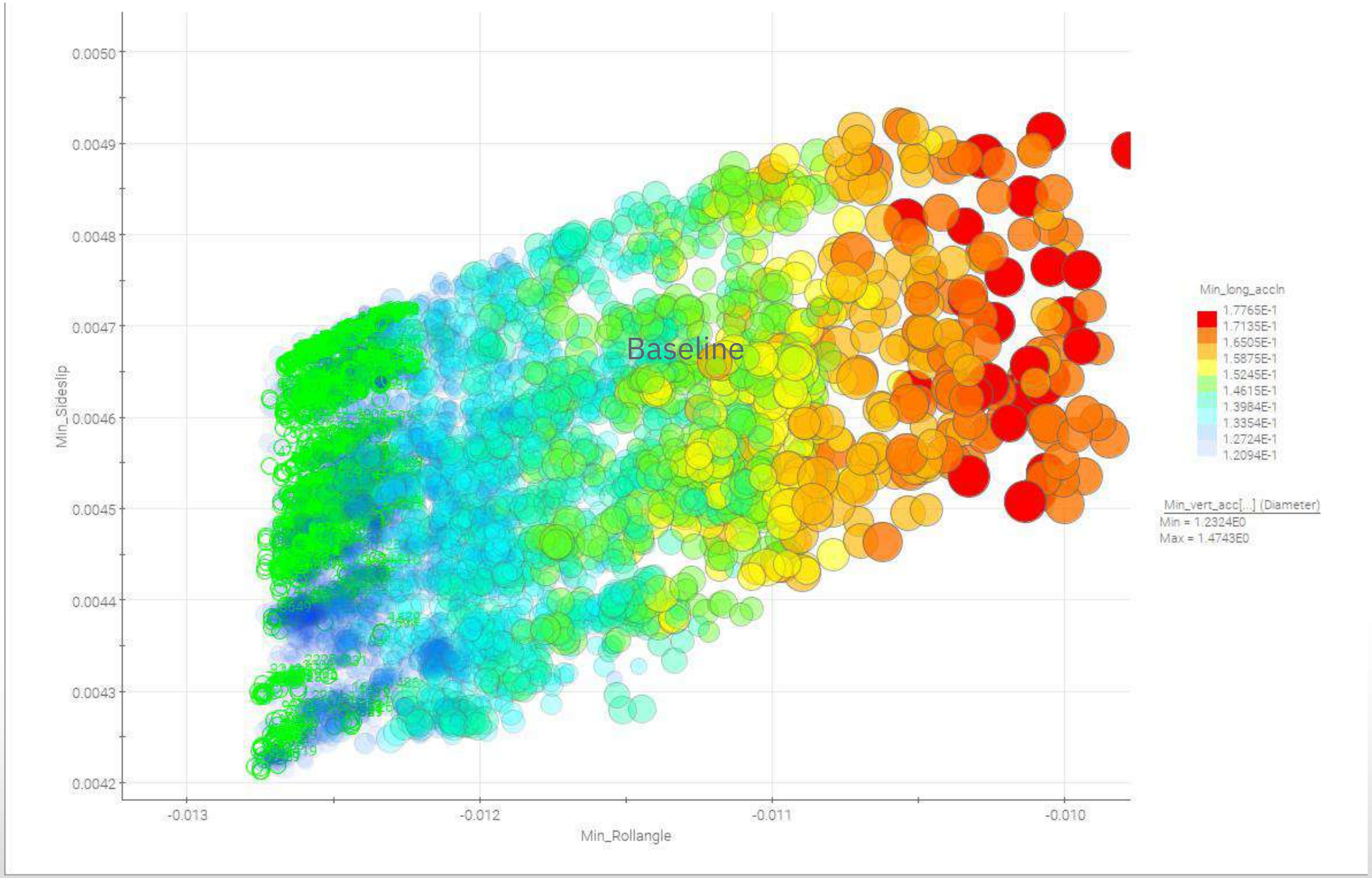
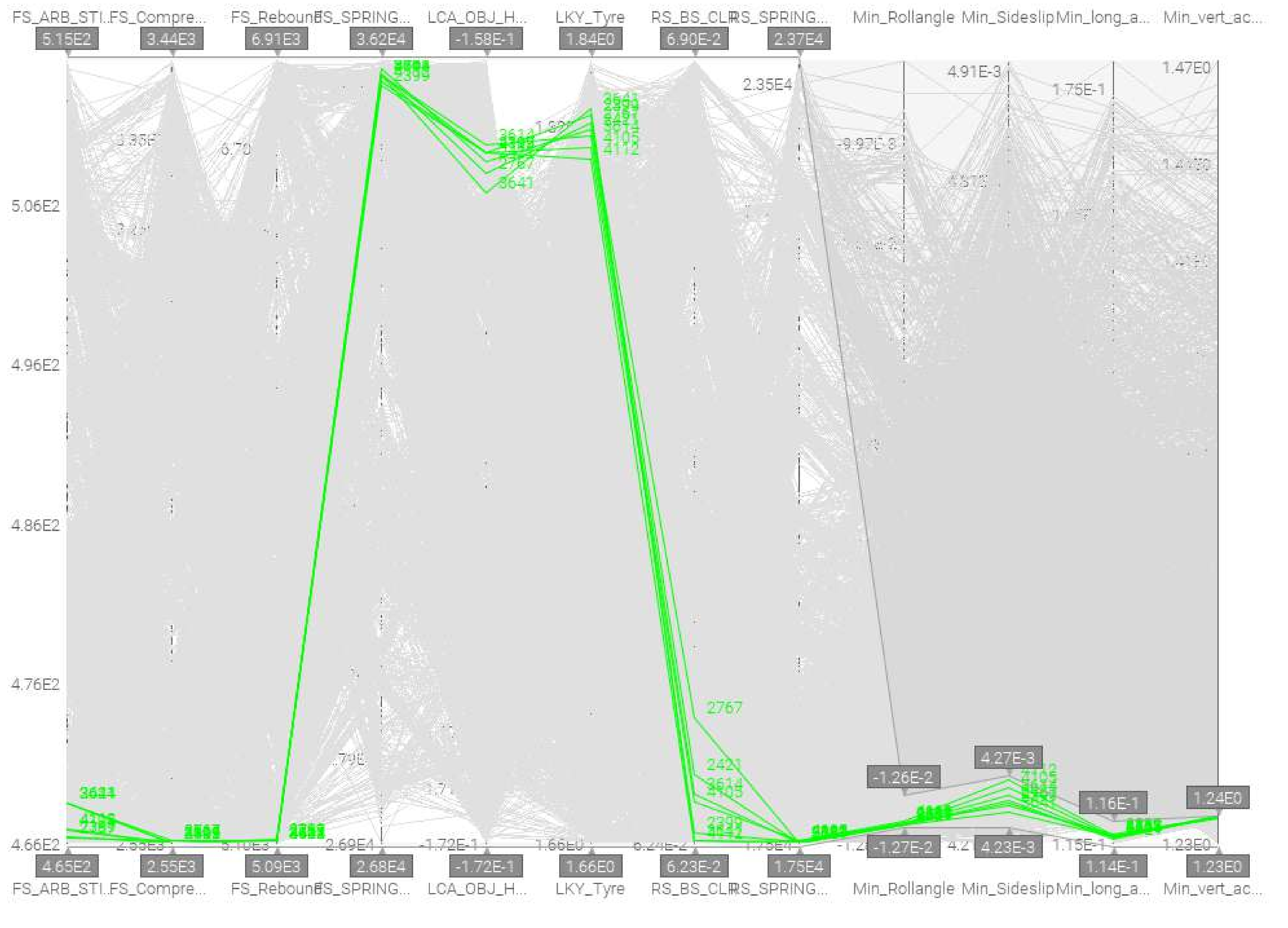


Fig 1. Min Roll Angle Gradient vs Min Side Slip Gradient vs Min Longitudinal Acceleration vs Min Vertical Acceleration

Virtual Runs - Parallel Coordinates



- Parallel Coordinates chart applied on Input Variables and Objectives
- Filtered designs are taken ahead

Direct vs Virtual - Results

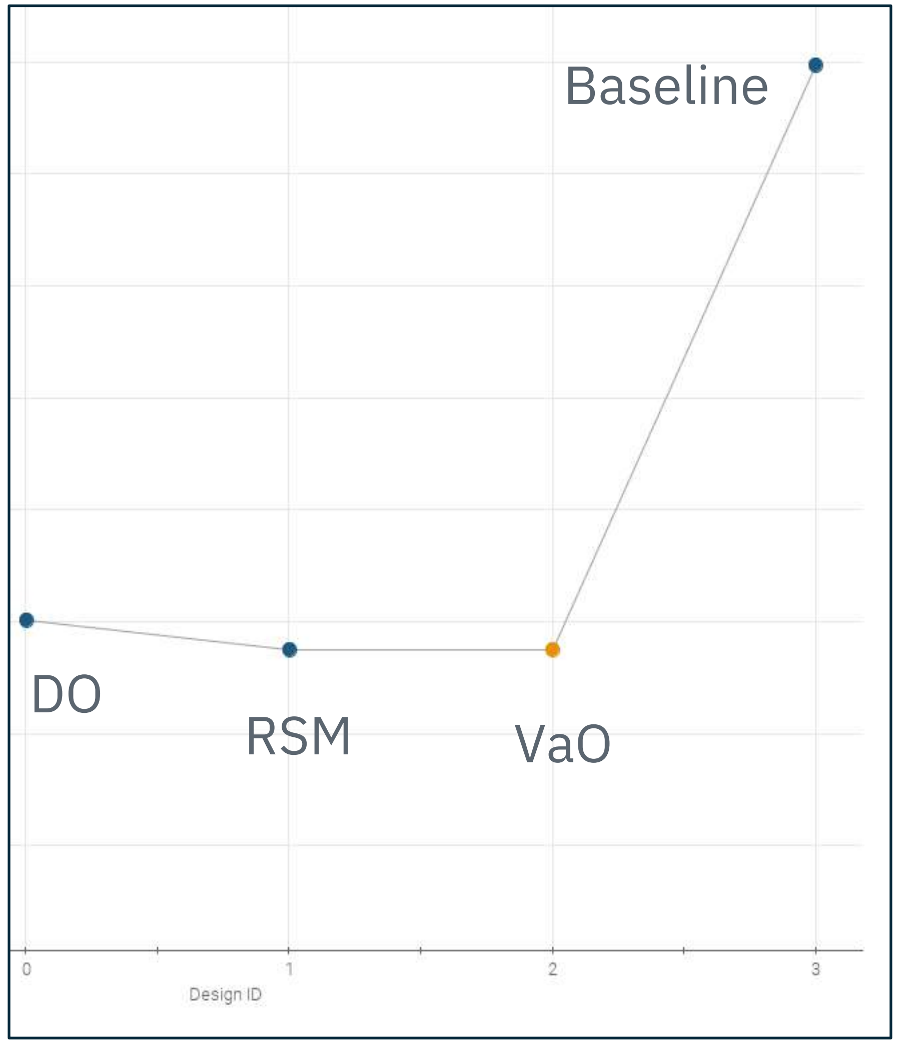


Fig 1. Optimal solutions for Roll Angle Gradient

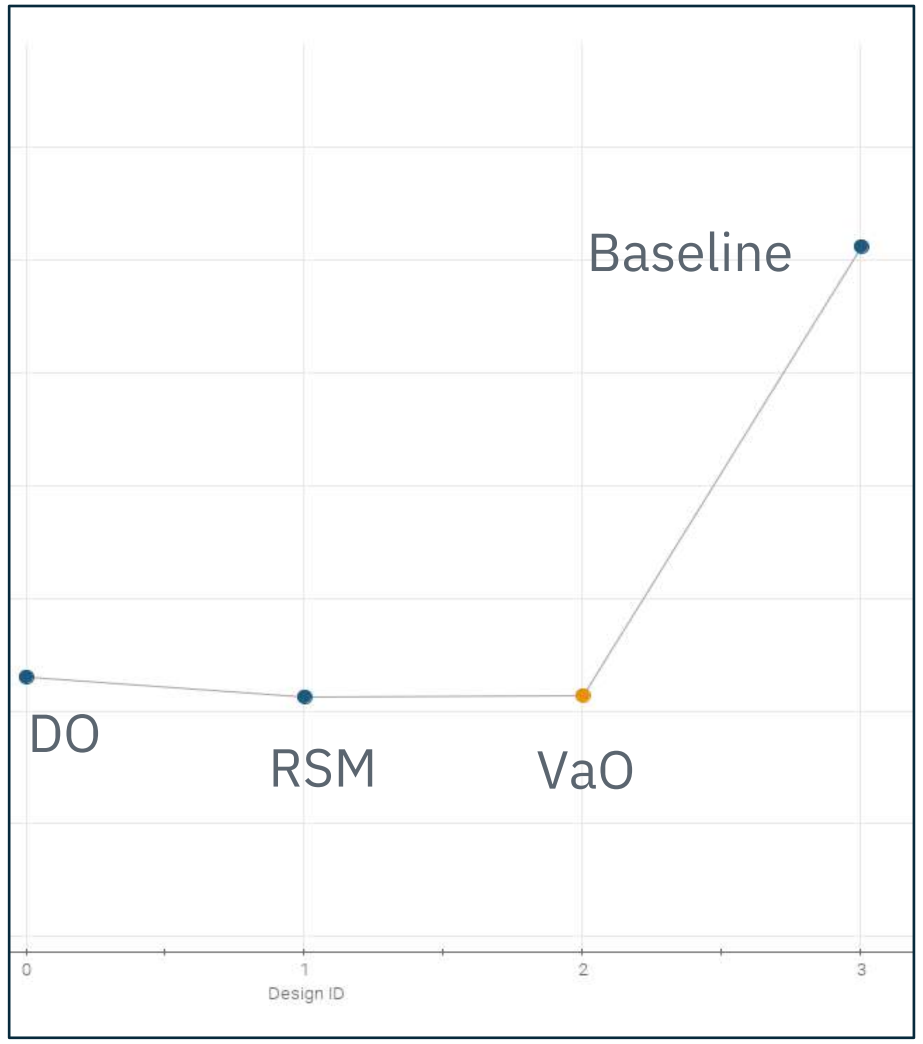


Fig 2. Optimal solutions for Side Slip Gradient

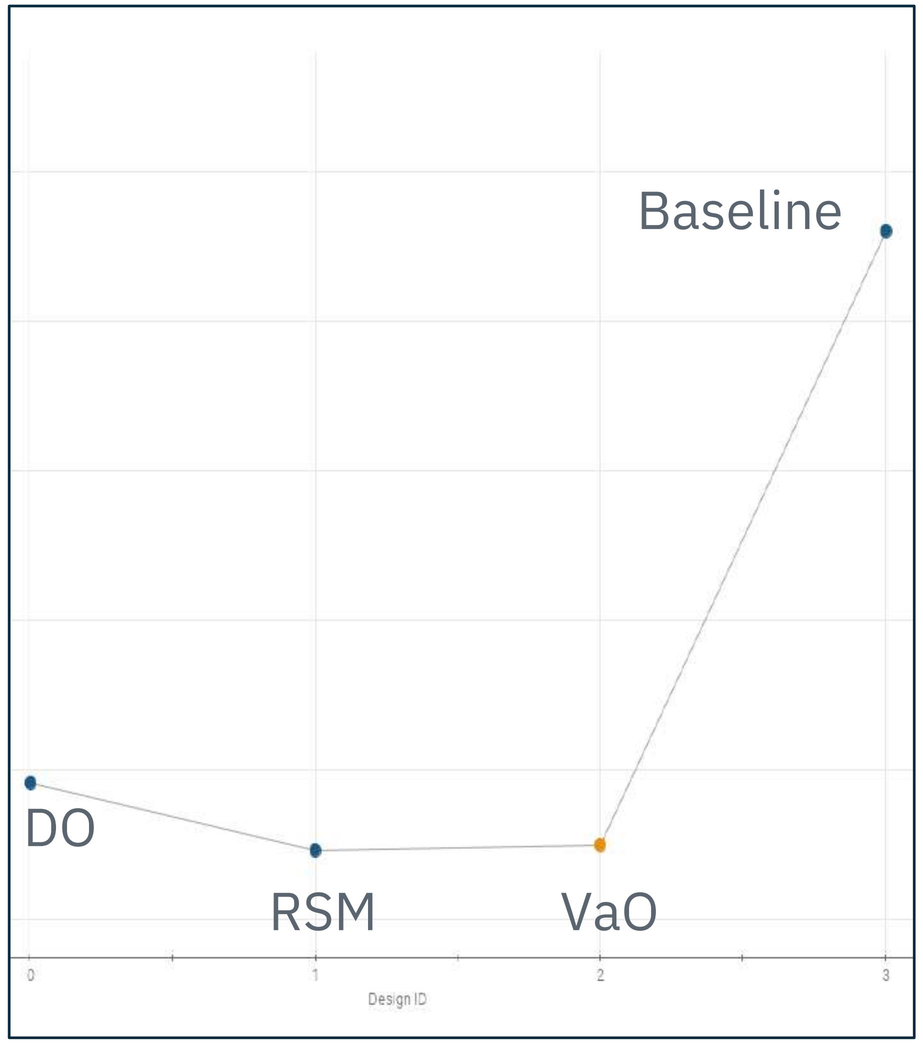


Fig 3. Optimal solutions for Longitudinal Acceleration

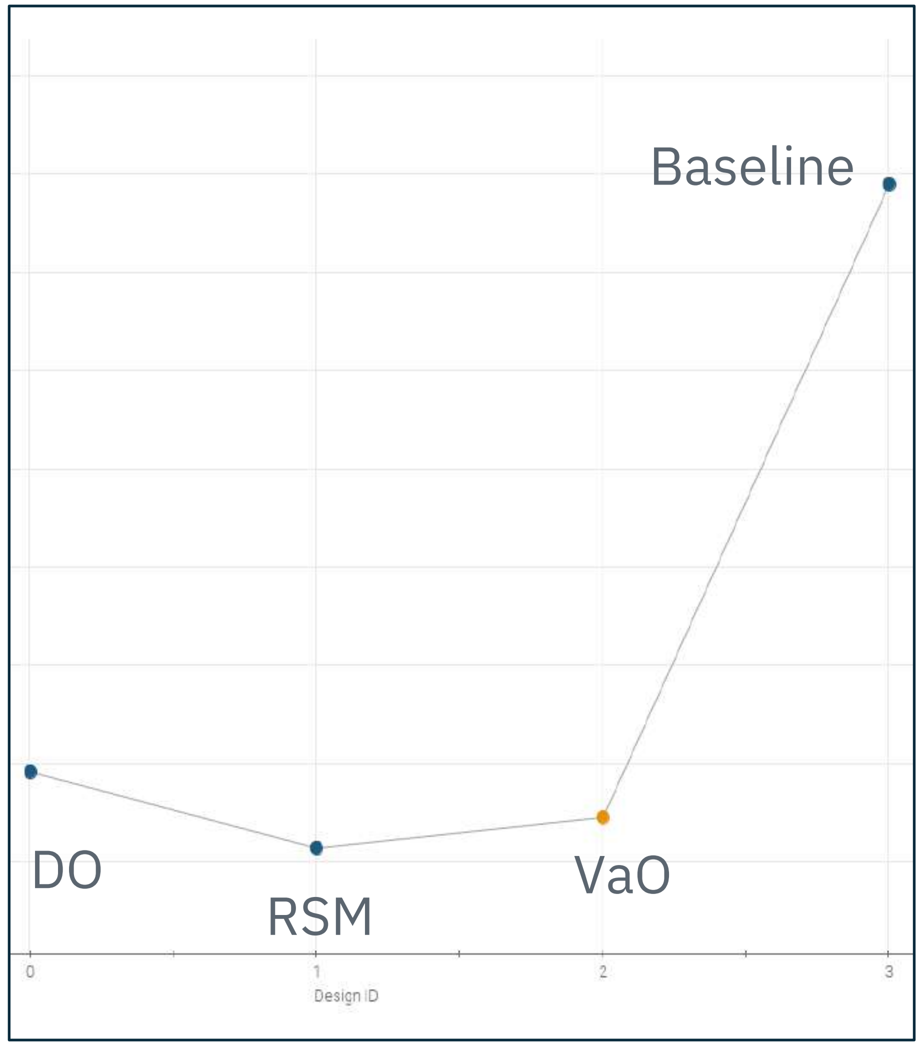


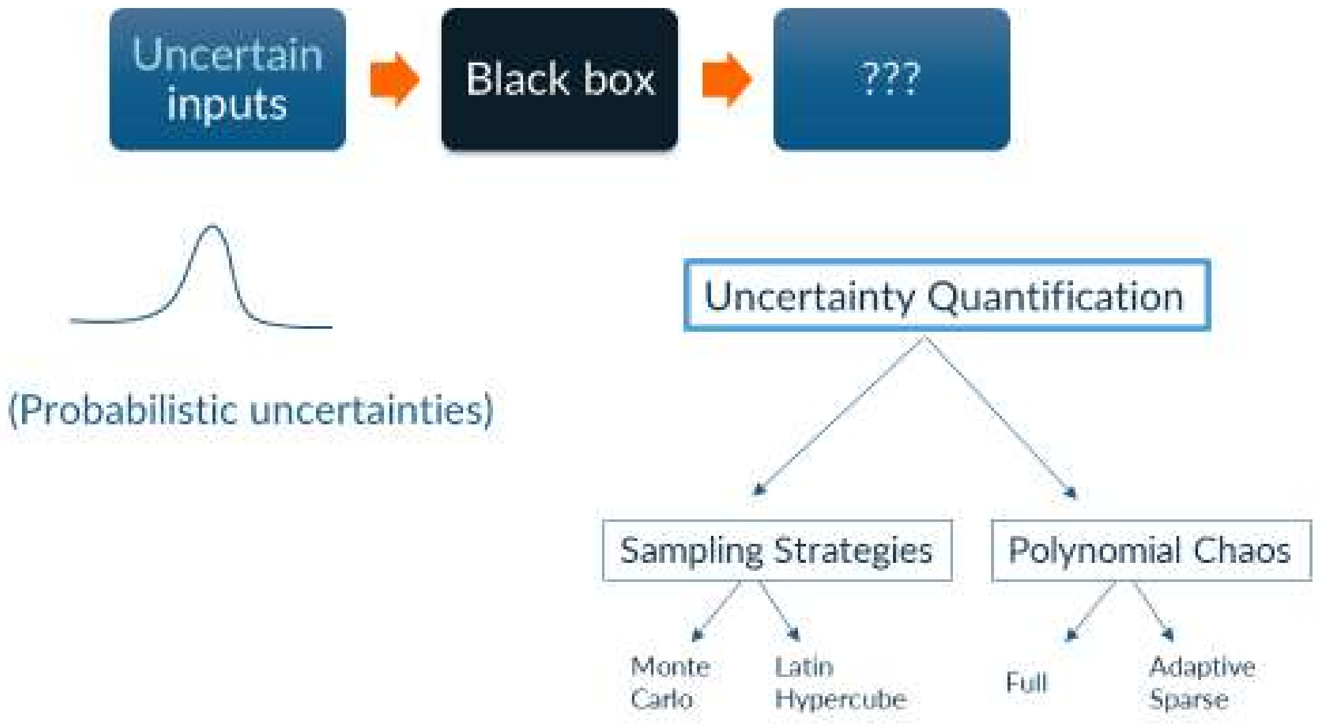
Fig 4. Optimal solutions for Vertical Acceleration

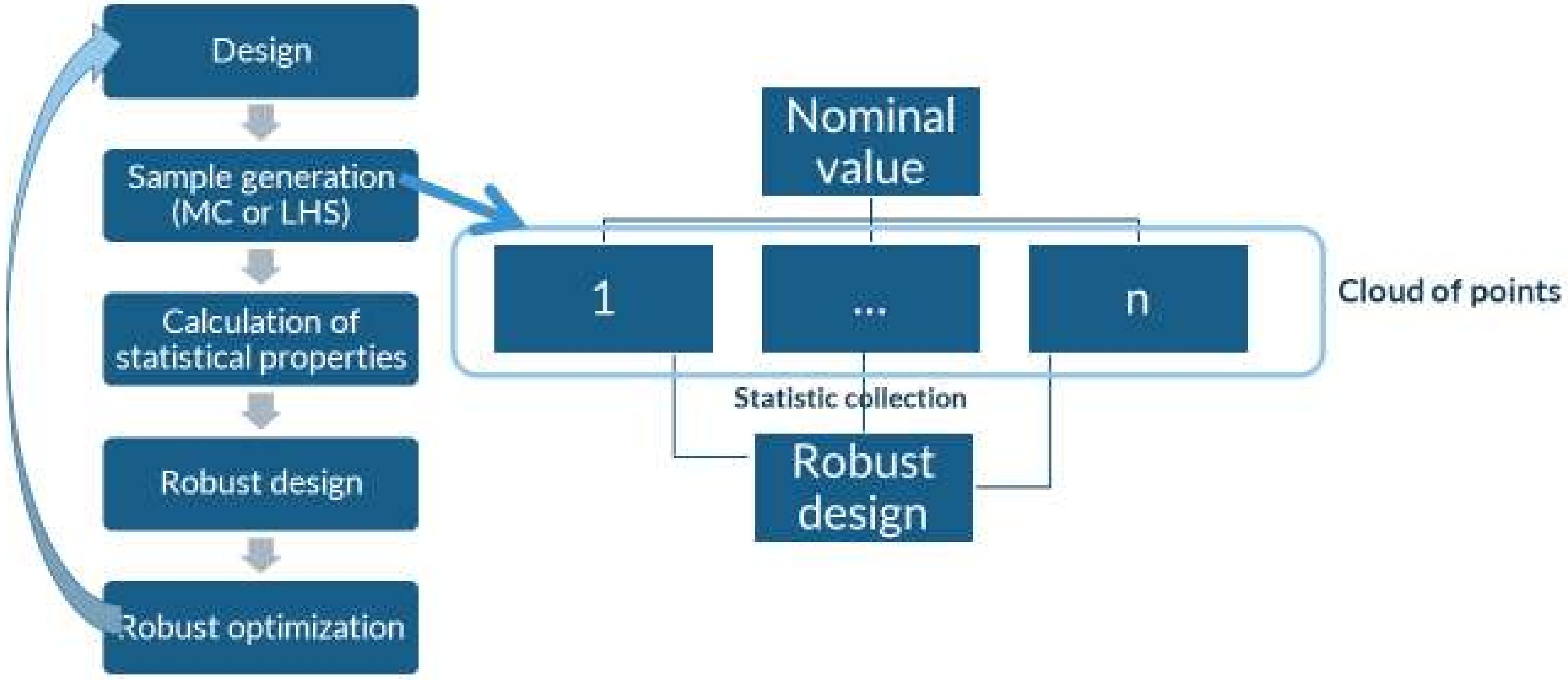
Direct and Virtual Optimization methods applied using modeFRONTIER have given better results than Baseline design

Baseline - Baseline design; DO – Direct Optimization ; RSM – RSM (Virtual)Optimization ; VaO – Validated RSM (Virtual) Optimization

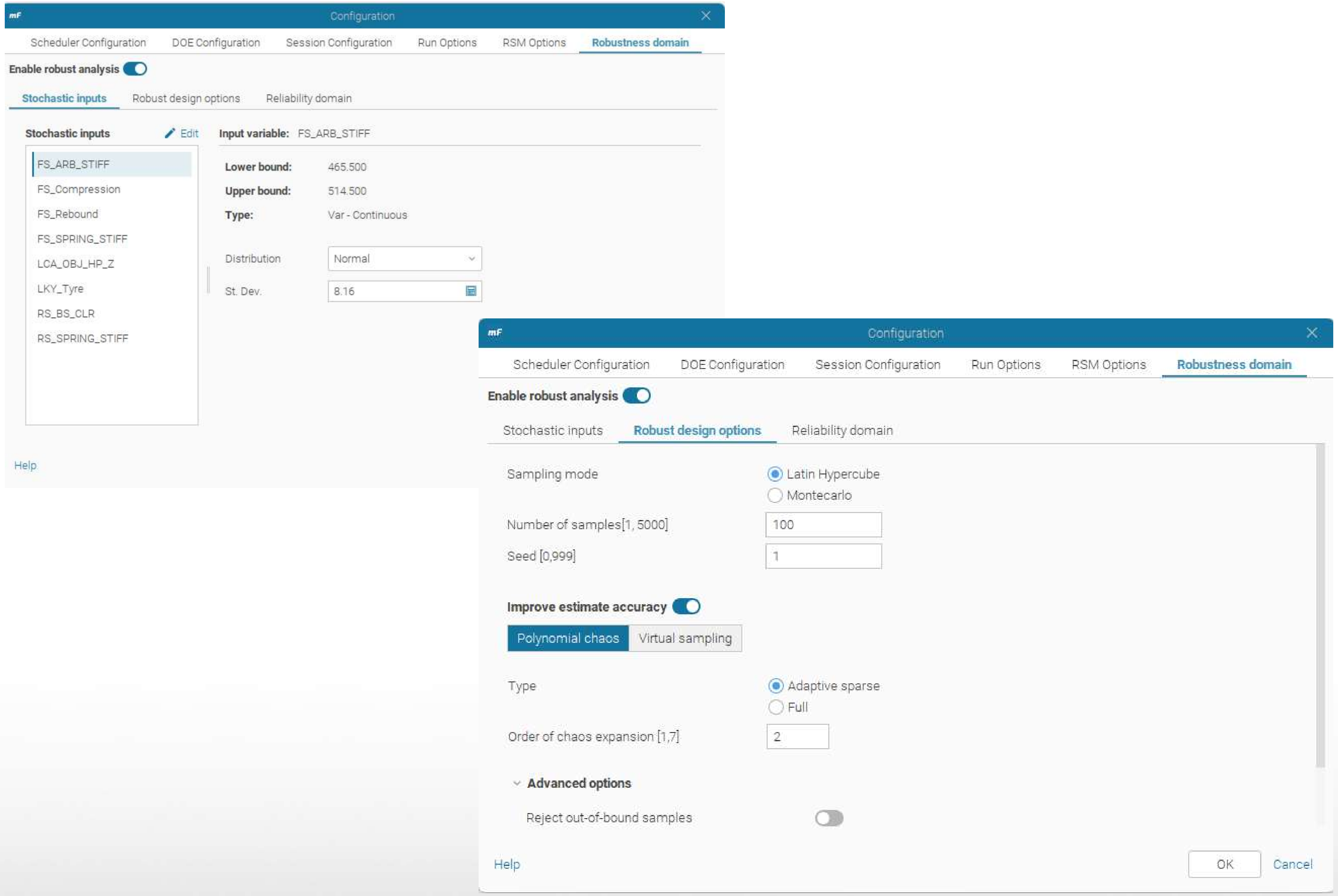
Uncertainty Quantification in modeFRONTIER

Uncertainties on Input variables may affect the responses and hence the performance. It is necessary to quantify the effect on the responses





Process in modeFRONTIER



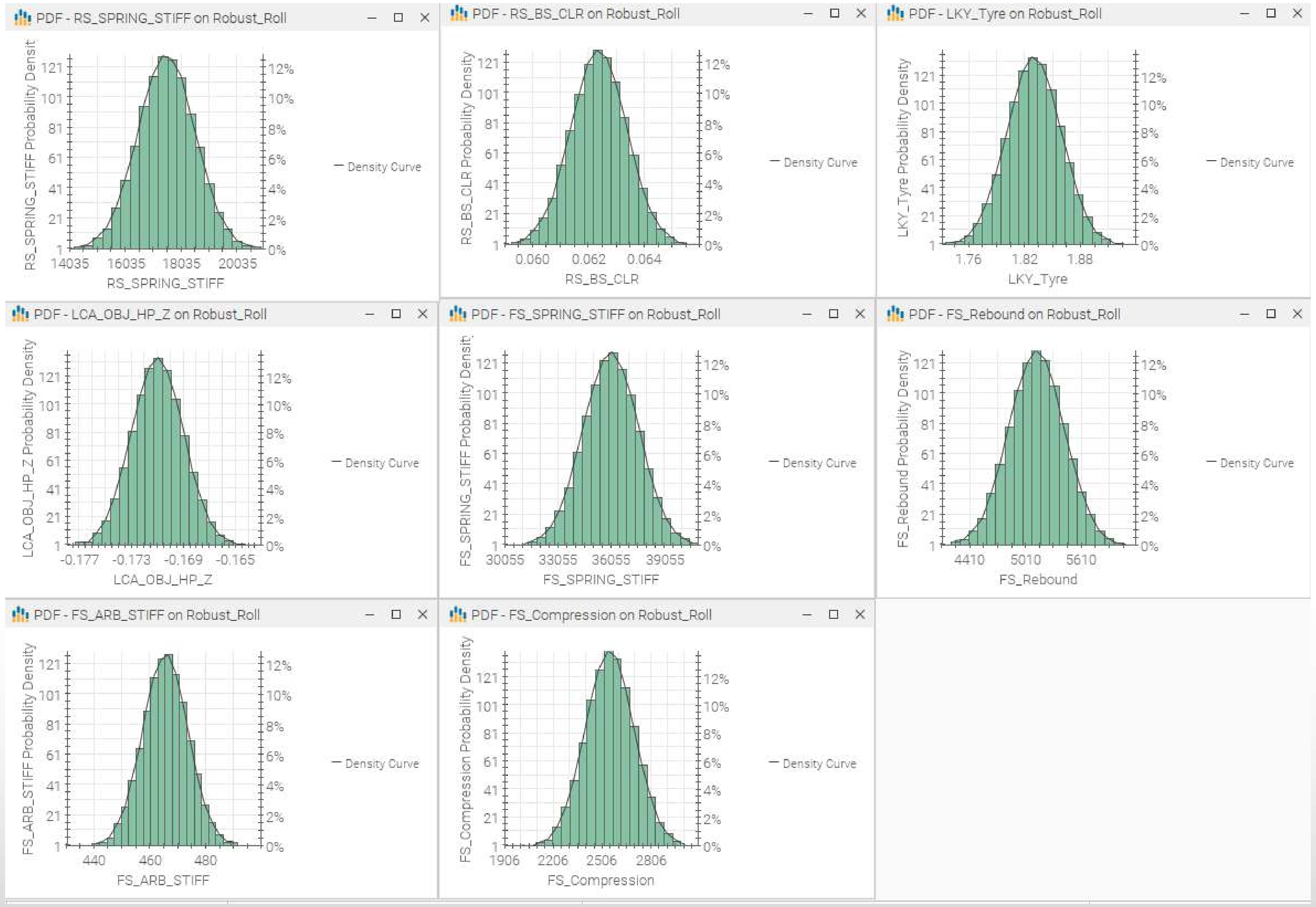
- The deterministic inputs are converted to stochastic inputs
- For each input, Normal distribution is chosen, and standard deviation is calculated based on Empirical formula:

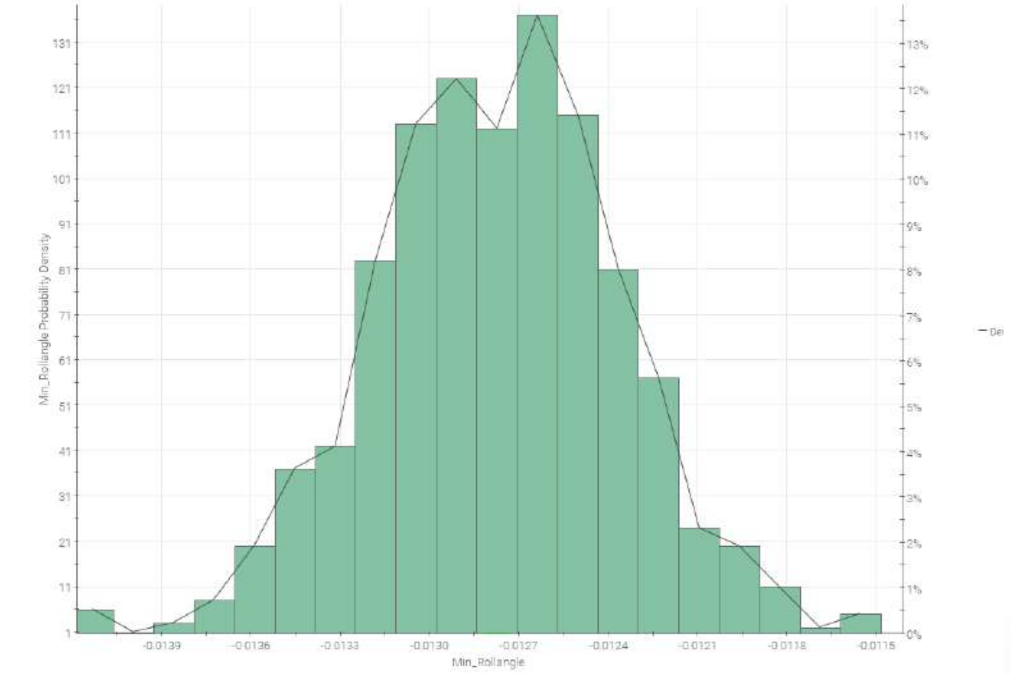
$$\frac{\textbf{Upper Bound} - \textbf{Lower Bound}}{\textbf{6}}$$

- 1000 Samples using Latin Hypercube following normal distribution were evaluated for the selected optimal design for each response

ESTECO
USERS' MEETING
INDIA

PDF charts - Inputs and output





Robust Sampling

ESTECO
USERS' MEETING
INDIA

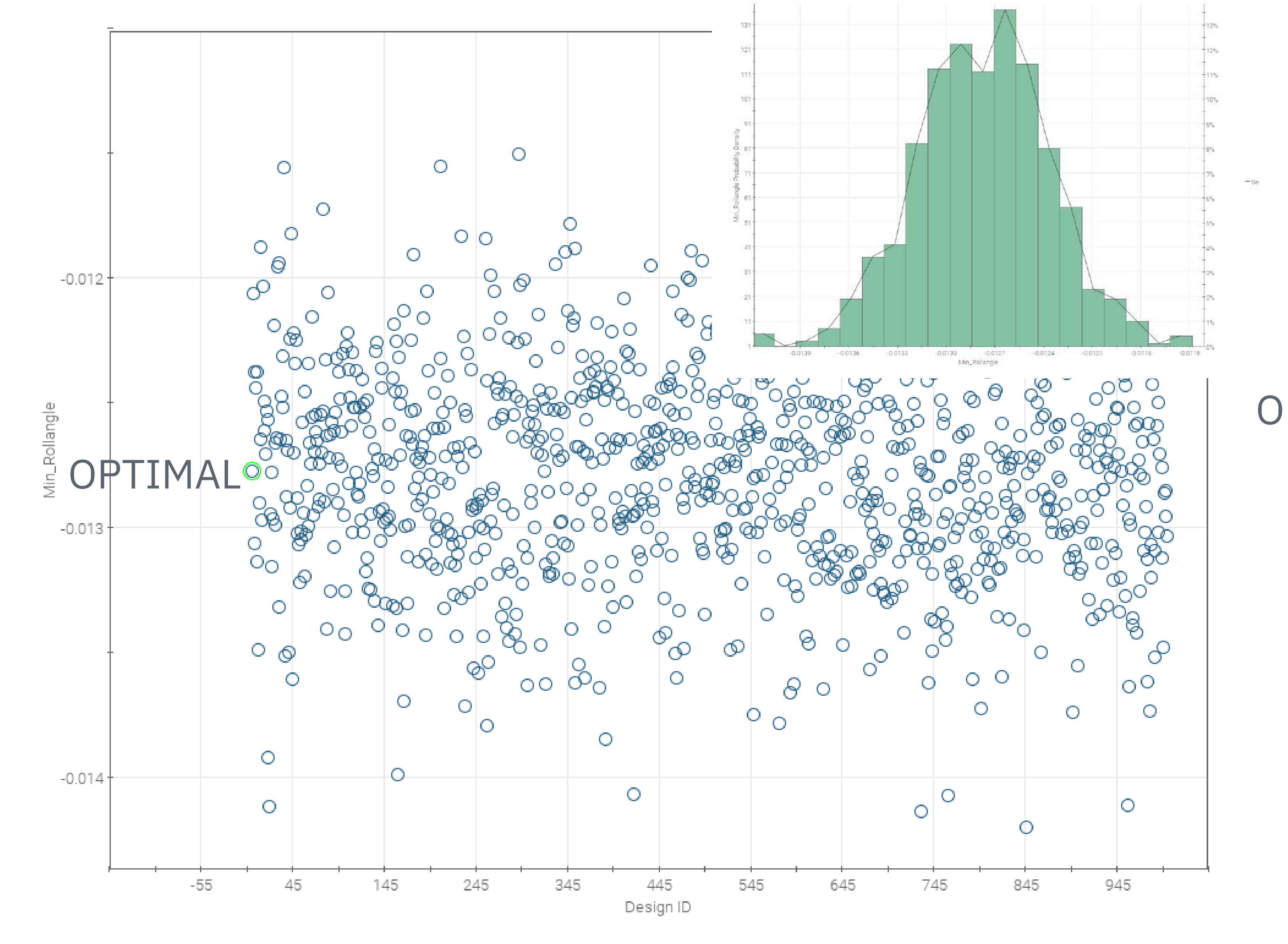


Fig 1. Robust Samples with RSM based Optimal solution for Roll Angle Gradient

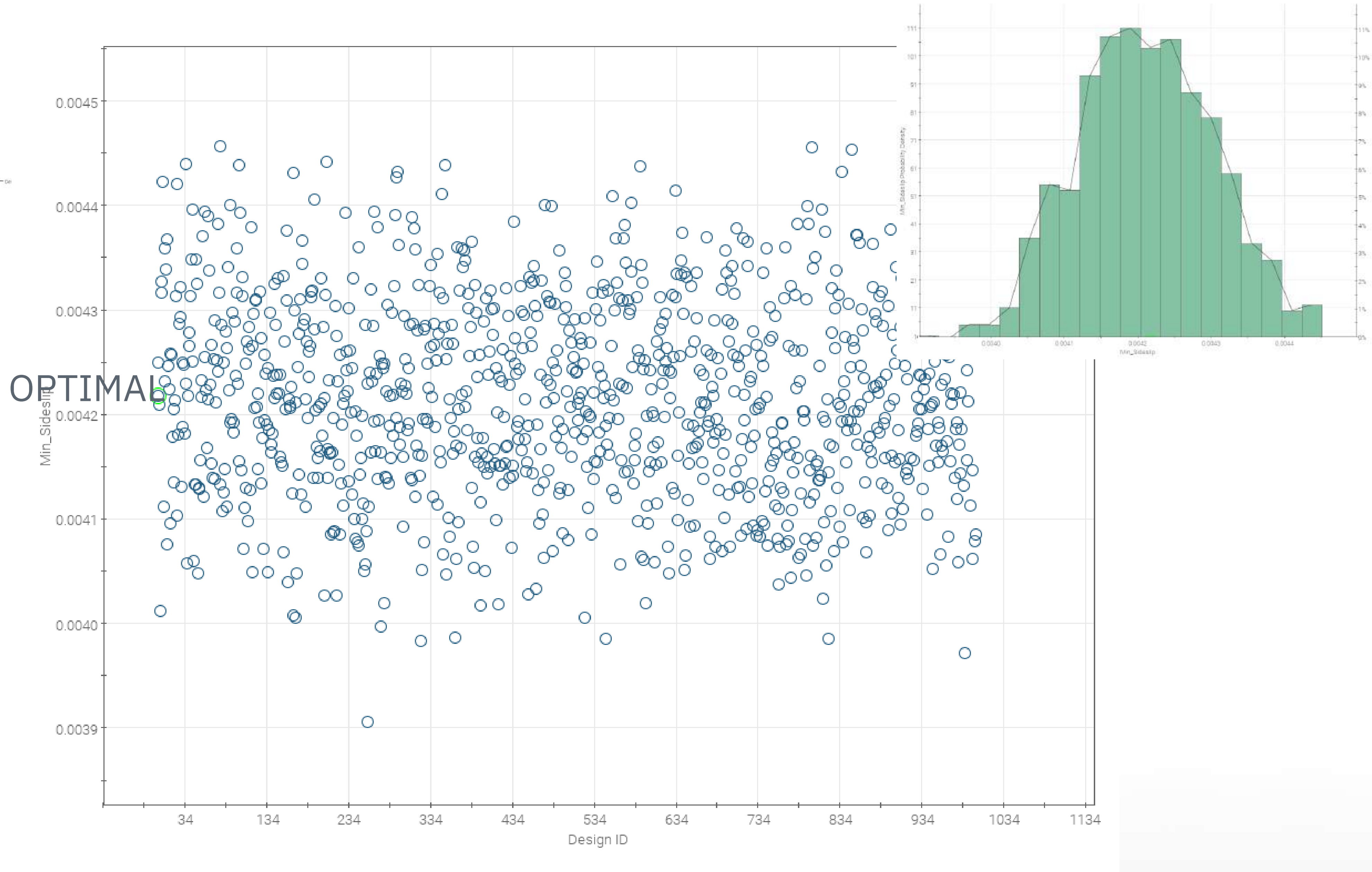


Fig 2. Robust Samples with RSM based Optimal solution for Side Slip Gradient

um 2023

Robust Sampling

ESTECO
USERS' MEETING
INDIA

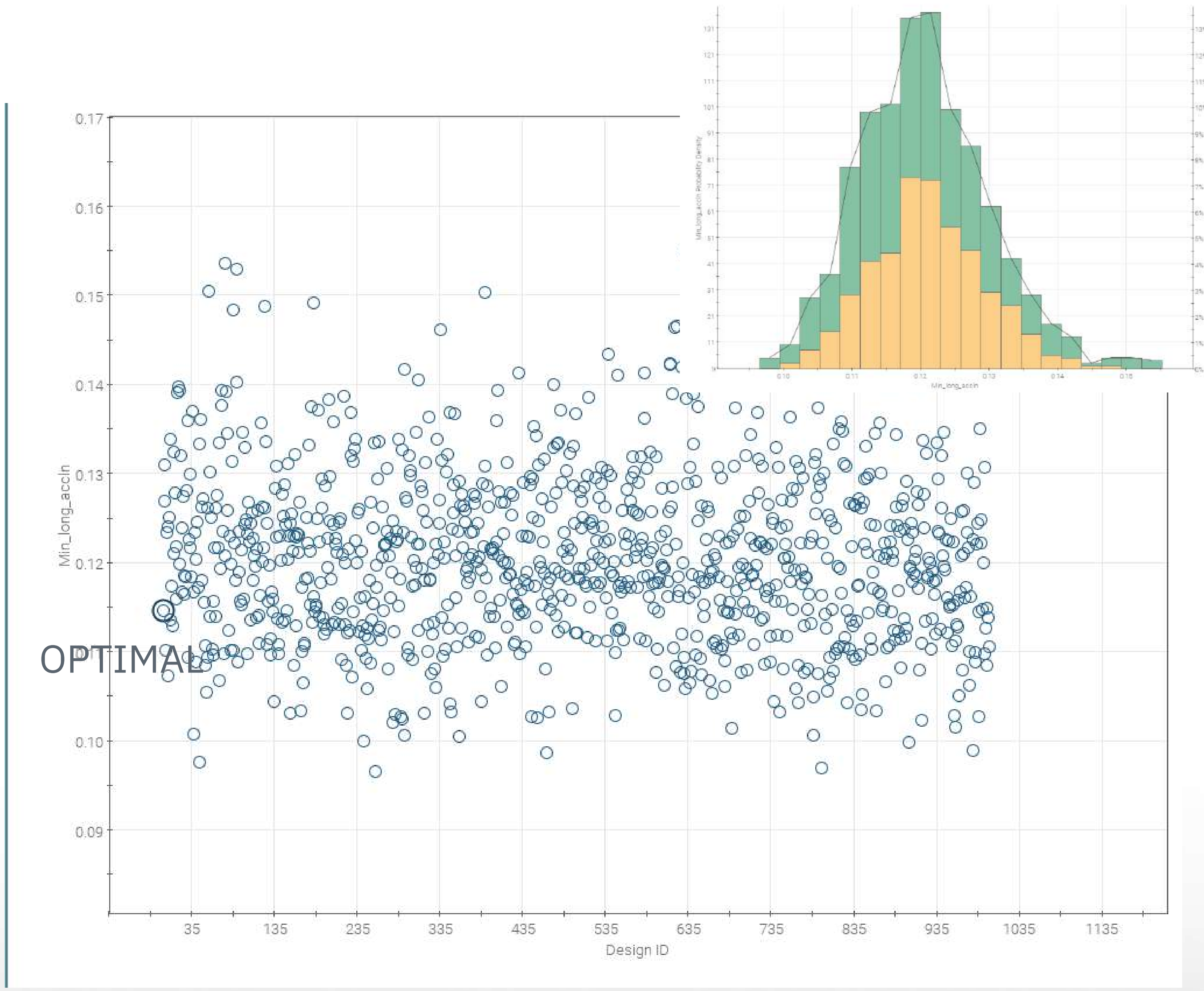


Fig 1. Robust Samples with RSM based Optimal solution for Longitudinal Acceleration

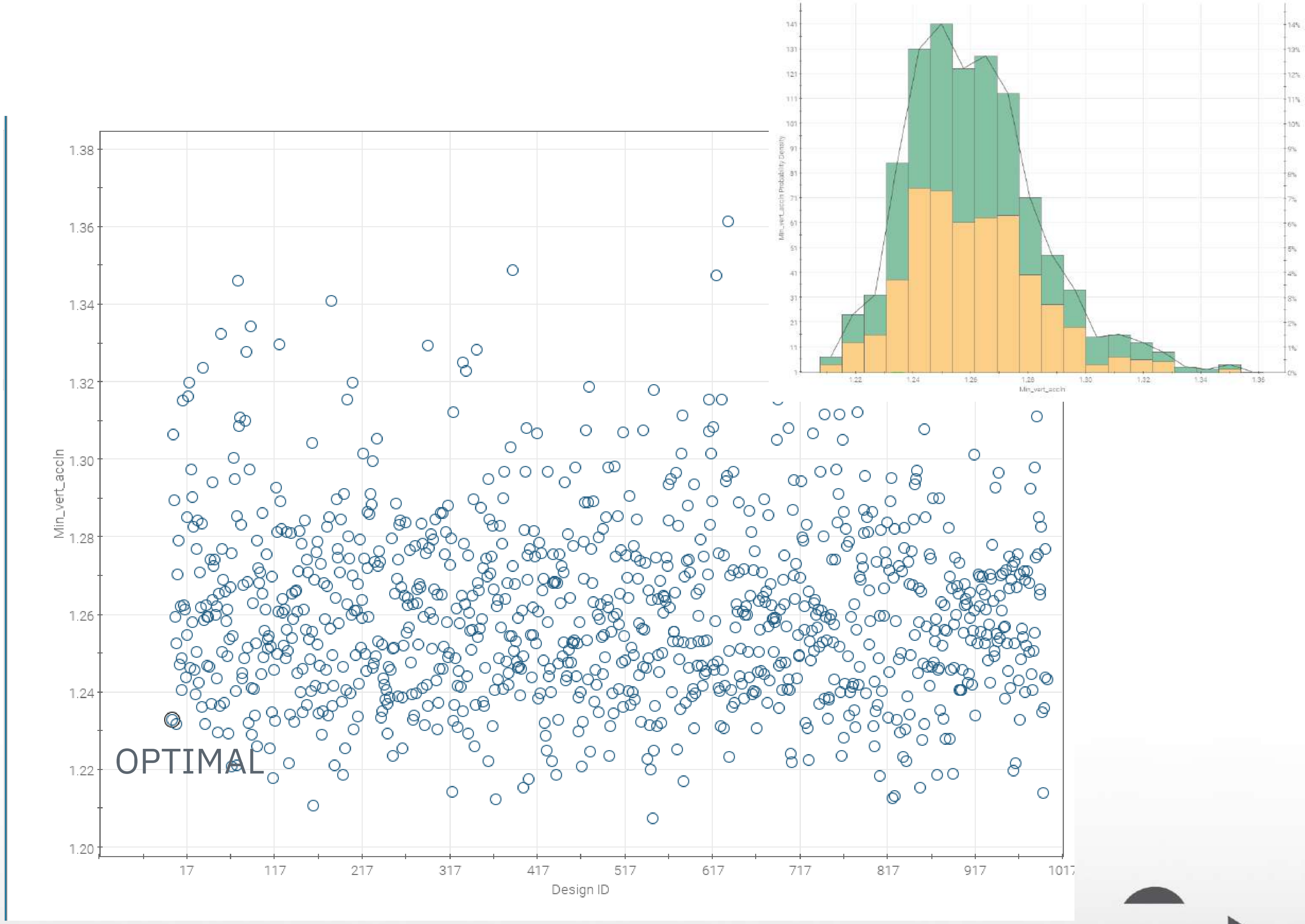


Fig 2. Robust Samples with RSM based Optimal solution for Vertical Acceleration

Inferences

- Acceptable Range of the DESIGN variables for Robust Design

ROII_RO

Designs: 1 (● 1)

		RID	Algorithm	Phase	
1	☐	0	● pilOPT	● MANY	[redacted]

		ID	Category
1	☐	0	○ pilOPT_MORDO
2	☐	1	○ pilOPT_MORDO
3	☐	2	○ pilOPT_MORDO
4	☐	3	○ pilOPT_MORDO
5	☐	4	○ pilOPT_MORDO
6	☐	5	○ pilOPT_MORDO

Conclusions

- A methodology for Multi-objective Optimization – considering both Ride and Handling Simultaneously is demonstrated.
- A methodology for Robust Design Optimization – considering 4 objectives pertaining to Ride and Handling is demonstrated
- A frame-work that establishes correlation between Subjective and Objective evaluation is In Progress.

ESTECO
USERS' MEETING
INDIA

Thank you!

esteco.com