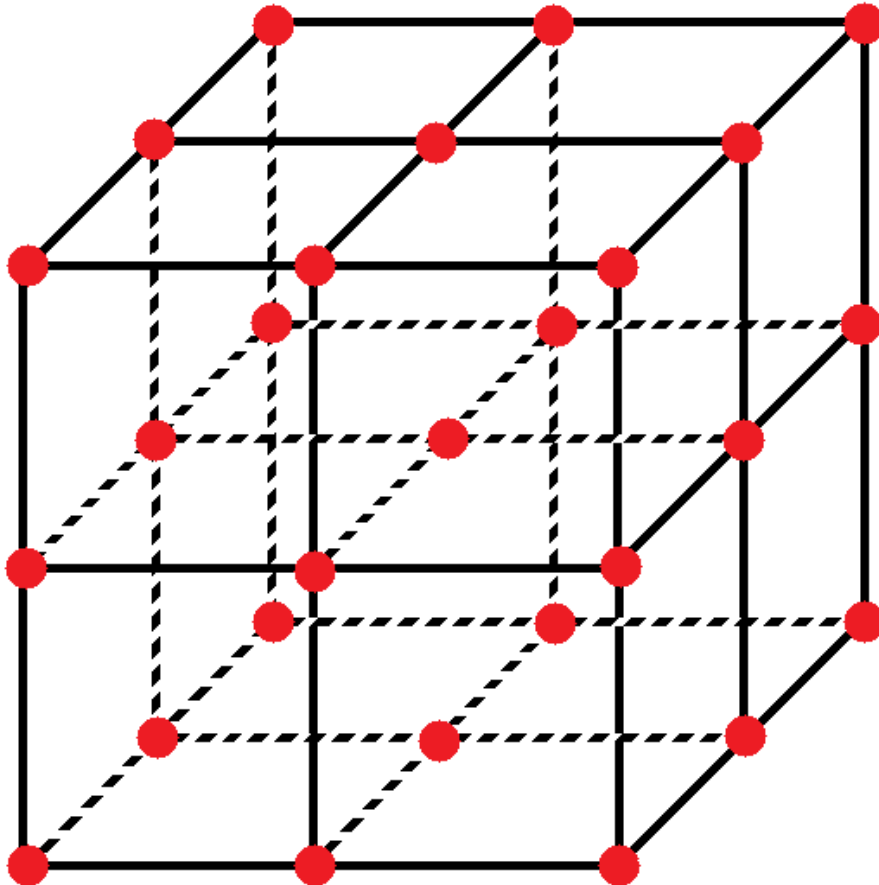


CHALMERS



Optimization of Modelica Models Utilizing Design of Experiments and Robust Design Methods

Master's thesis in Applied Mechanics

STEFAN BRÖYN

Department of Applied Mechanics
Division of Vehicle Engineering and Autonomous System
Vehicle Dynamics Group
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2014
Master's thesis 2014:04

MASTER'S THESIS IN APPLIED MECHANICS

Optimization of Modelica Models Utilizing Design of
Experiments and Robust Design Methods

STEFAN BRÖYN

Department of Applied Mechanics
Division of Vehicle Engineering and Autonomous System
Vehicle Dynamics Group

CHALMERS UNIVERSITY OF TECHNOLOGY

Gothenburg, Sweden 2014

Optimization of Modelica Models Utilizing Design of Experiments and Robust
Design Methods
STEFAN BRÖYN

© STEFAN BRÖYN, 2014

Master's thesis 2014:04
ISSN 1652-8557
Department of Applied Mechanics
Division of Vehicle Engineering and Autonomous System
Vehicle Dynamics Group
Chalmers University of Technology
SE-412 96 Gothenburg
Sweden
Telephone: +46 (0)31-772 1000

Chalmers Reproservice
Gothenburg, Sweden 2014

Optimization of Modelica Models Utilizing Design of Experiments and Robust Design Methods
Master's thesis in Applied Mechanics
STEFAN BRÖYN
Department of Applied Mechanics
Division of Vehicle Engineering and Autonomous System
Vehicle Dynamics Group
Chalmers University of Technology

ABSTRACT

This thesis describes the implementation of Robust Design and Design of Experiments methodologies with the FMI Microsoft Excel Add-in (FMIE). FMIE allows engineers to use Microsoft Excel in collaboration with simulation models created in other software and exported as FMUs.

A choice of pre- and post processing methods are implemented as Excel macros for use with the FMIE. Their functionality are demonstrated with a non trivial parameter optimization of a truck model created with Dymola. The development of the truck model and integral parameters are explained within this thesis.

The Excel macros are used to find a design point that is sufficiently good in aspect to two response factors. The stability of this design point with aspect to a noise factor and small variance of the design factors is determined.

Keywords: Robust Design, Design of Experiments, Modelica, Dymola, Microsoft Excel, Simulation, FMI, FMU

ACKNOWLEDGEMENTS

I would like to thank Modelon AB for making it possible to perform this master thesis.

I would especially like to thank Iakov Nakhimovski and Ivar Torstensson for their advice and discussions during the spring and summer.

Finally I also would like to thank my examiner professor Bengt Jacobson for his help during the thesis.

CONTENTS

Abstract	i
Acknowledgements	i
Contents	iii
1 Introduction	1
1.1 Scope and Purpose	2
1.2 Delimitations	2
2 Modelling and Simulation with Dymola and FMI	3
2.1 Modelica	3
2.2 Dymola	3
2.3 FMI Add-In for Excel	4
3 Robust Design	5
3.1 Design of Experiments	6
3.1.1 Factorial Designs	6
3.1.2 Space-Filling Algorithms	6
3.2 Post-processing	8
3.2.1 Scatter Plot	8
3.2.2 Pareto Front	9
3.2.3 Pearson's Correlation	9
4 Implementation within Microsoft Excel	10
4.1 Existing functionality, FMI Add-in for Excel	10
4.2 Explanation of DOE macros	10
4.2.1 Screening	10
4.2.2 Sensitivity analysis	11
4.2.3 Creating results	12
4.2.4 Presenting Results	13
5 Analysis of a transport problem	14
5.1 Model description	15
5.1.1 Truck Model	16
5.1.2 Driver Model	18
5.1.3 Road Model	19
5.2 Screening	19
5.3 Sensitivity analysis	21
5.3.1 Evaluation of Robustness Within the Control Factors	22
5.3.2 Analysis of Robustness due to Wind Conditions	23
5.4 Results from the Analysis	23
6 Recommendations for Future Work	25
A Additional Plots	27
B Pareto Data	28

1 Introduction

The cost of making changes to a product increases with the time spent in the development process. If it is discovered that a change has to be made in a full scale prototype of, for example, a car it most certainly will be both time consuming and expensive. Early gathering of information about how the product works reduces the risk of making faulty decisions early in the process that will have to be changed later. One way to gather this information is through computer models. Many products can be described by means of dynamic models, and then the use of the model is often referred to as a simulation or dynamic simulation.

When analysing a model the difficulties of understanding how each parameter affects the output arises with the models complexity. Optimally one would be able to describe the product behaviour with a invertible mathematical function, for complex products it is sadly closer to an exception than a regularity that this is possible. As a result of this the developer is forced to test the model repeatedly with different input. Depending on the experiment performed the time this takes varies, but as a rule of thumb it is more than desired.

As information gathering easily becomes both time consuming and expensive, methods have been developed to maximize the information gathered while minimizing the cost of the experiments. This thesis focuses on:

- Applying the methods of Robust Design (RD) and Design of Experiments (DOE)[10] to simulations performed with the FMI add-in for Excel (FMIE).
- Demonstrate the functionality obtained when implementing RD and DOE with the FMI add-in for Excel. This is done with a model of a truck performing a transport task created in Dymola.

The truck model is exported from Dymola using the FMI standard and simulated with the FMIE. The analysis is divided into two main parts, *screening* and *sensitivity analysis*, both of these two are then divided into the subparts *case set up*, *simulation* and *post-processing*. Before performing a sensitivity analysis the designer must decide on a design point, this is often done with the use of optimization. The subject of optimization is considered too large to be included within this thesis and is therefore excluded. The design process is displayed in Figure 1.0.1.

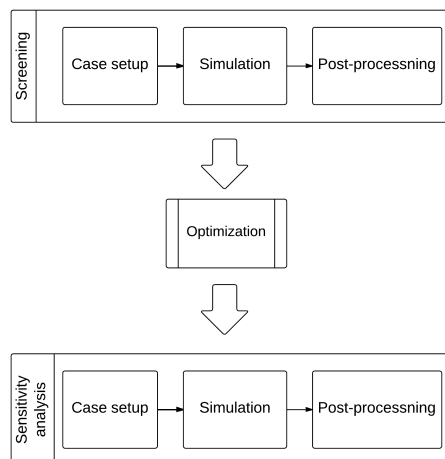


Figure 1.0.1: *The process of analysis applied within this thesis.*

1.1 Scope and Purpose

The results from this thesis are macros implemented within Microsoft Excel utilizing RD and DOE theories. The macros acts as a suggestion on how to continue the development of FMIE. The thesis includes an example as verification of that the implementation works for one relevant and nontrivial design.

1.2 Delimitations

The chosen DOE and RD methods will be implemented within Microsoft Excel and written in the Visual Basic for Applications (VBA) language.

The scope of RD and DOE is large and not all methods for product optimization are well suited for computer simulations. It is therefore important to identify the “right” methods for this thesis. The DOE and RD methods in this thesis are applied with computer simulations. They should however be applicable on other types of experiments as well.

The theories of DOE and RD are a great asset when working with optimization. Within this thesis the optimization is carried out with the use of manual selection. For a more mathematical approach to optimization the author recommends the use of other sources.

The truck model created in Dymola will not be based on any particular existing truck. It will however strive to use and produce as realistic values as possible.

2 Modelling and Simulation with Dymola and FMI

There are many different programming languages and software used for simulations. Choosing the right tool for the problem in question is important. A suboptimal simulation software or programming language may make it difficult to obtain good results. The model created for this thesis was built in Dymola a simulation software based on the Modelica programming language.

2.1 Modelica

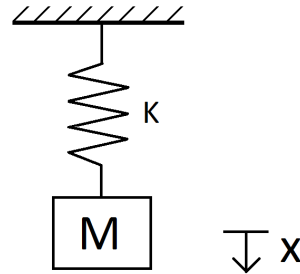
Modelica [1] is a language focused on modelling of multi domain physical systems. It is owned and developed by the Modelica Association. Being equation based and acausal the user does not have to define the order variables are calculated as long as the system is properly defined. The free Modelica Standard Library already includes a large number of ready to use components for areas as mechanics, electrical, thermodynamics etc. In addition it is easy to create your own libraries with custom components.

The Modelica code in Figure 2.1.1a contains the equations needed to describe the mass-spring system in Figure 2.1.1b.

```
model massSpring
  import SI =
    Modelica.SIunits;

  constant Real k = 2;
  SI.Acceleration a;
  SI.Velocity v(start=0);
  SI.Distance x(start=0);
  constant SI.Mass m = 1;
  constant SI.Acceleration g
    = 9.81;

  equation
    a = der(v);
    v = der(x);
    a*m = m*g-k*x;
end massSpring;
```



(a) Modelica code of mass-spring system.

(b) Mass-spring system

Figure 2.1.1: Mass-spring system

2.2 Dymola

Dymola [2] is a program developed by Dassault Systèmes. The modelling part of Dymola is based on the Modelica Language. Apart from text based modelling Dymola provides a graphical interface that helps the user create models and displays them in a pedagogical way. After a model is created Dymola also contains environments for simulation and illustration of results. A model of the same mass-spring system as in Figure 2.1.1a is viewed in Figure 2.2.1 but this time created with existing blocks from the Modelica Standard Library.

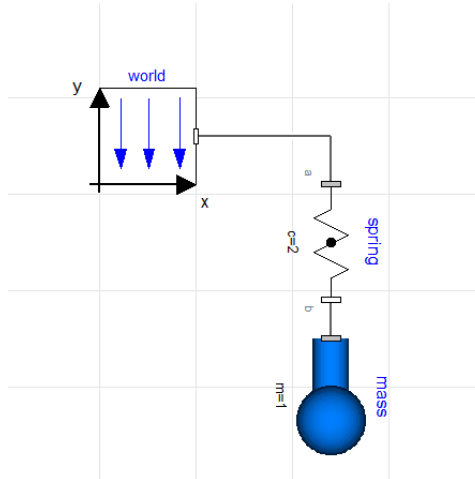


Figure 2.2.1: *Dymola model of mass-spring system.*

2.3 FMI Add-In for Excel

The Functional-Mock-up Interface [3] is a standard for exchange of compiled dynamic models, and is intended to promote model reuse and tool interoperability. Several tools provide export of Functional Mock-up Units (FMUs), all of which can be used with the FMI Add-In for Excel (FMIE). FMI provides two different formats for exchange of models:

- FMI for Model Exchange (FMI-ME). The FMI-ME specification is based on a continuous-time hybrid Ordinary Differential Equation (ODE) representation. The FMU-ME provides inputs and outputs and exposes functions for setting parameters and computing the derivatives of the ODE. FMIE supports initialization of FMU-MEs.
- FMI for Co-Simulation (FMI-CS). The FMI-CS specification provides a model representation where both the model and an integrator (ODE solver) are encapsulated inside the FMU-CS. Similar to the FMI-ME, the FMUCS provides inputs and outputs and means to set model parameters. It also provides a function to integrate the dynamics of the model for a specified time interval. FMIE supports dynamic simulation and initialization of FMU-CS.

An FMU is a zip archive that contains a platform specific DLL(s) and metadata in XML files and optionally additionally resource data. The metadata file specifies general model information such as the name of the model and when it was generated. It also specifies all the variables and parameters and how they may be used.

The amount of programmes supporting FMU export and import are steady growing, more information about the programmes and their specific support may be found on the FMI web page [4]. This thesis will make use of Co-Simulation FMUs created with Dymola.

3 Robust Design

When developing a product the choice of design process has large influence on the end result, especially for complex products where many engineers and companies/suppliers are involved over long time. During development decisions are made that will influence the performance of the product. As most developers would like the value of their product to be as high as possible they want to make good decisions. But how do one decide what is a good decision? What is attractive will differ from person to person and we cannot rely upon the single designer to know what is right and make the best decisions by heart. During the history of product development methods have been developed to quantify our results and eliminate personal judgement.

It is not sufficient to just maximize the performance and minimize the cost of the product. The developer must also ensure the quality of the product. Design parameters often deviate from their nominal values, this may be a result of tolerances within manufacturing, ageing or change of operating environment. If a products performance is inconsistent, no matter how high, it will be perceived as quality loss for the end user.

Robust Design focuses on achieving a consistently high performing product. In other words, balancing the mean on its target while minimizing the variance. One of the key tools used with Robust design is the P-diagram, Figure 3.0.1, it is used to classify the variables within the design into four categories.

- Response factors, f — How performance is measured, product shall be designed so that this is as close as possible to the requirements on the function.
- Signal factors, m — Input to the product/process from the end user. The signal factors often differ from time to time and between users.
- Control factors, b — Design parameters, these are changed to influence the response. It is through the control factors the designer influences the product. The goal with the development process is to find the correct values.
- Noise factors, z — The reason Robust Design exists. All unknown influence on the response factors. Examples are: Tolerances, temperature differences, wear, plastic creep etc.

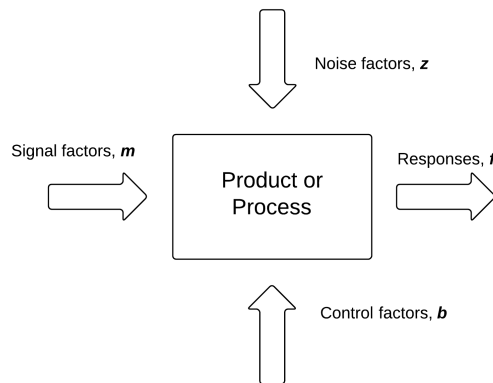


Figure 3.0.1: A P-diagram, commonly used with robust design methods.

3.1 Design of Experiments

The characteristics of a system is determined by the laws of nature, but for complex products it is difficult to intuitively predict its behaviour. Experiments therefore becomes a important tool when searching for a systems characteristics. As performing many experiment often becomes time consuming and expensive it is important to maximize the information gathered with each case. The goal here is to gain an understanding of if and to what magnitude control factors influence the mean value and variance. Design of Experiments describes how to select combinations of control variables to gain comprehensive information with a limited number of cases.

3.1.1 Factorial Designs

Assuming a experiment is performed to determine the influence three settings have on a system, the settings are either “high” or “low”. Creating a test case for each setting combination results in 2^3 cases, Table 3.1.1. This may seem affordable but as the number if factors grow the number of cases increases rapidly. It may also not be enough with two levels on each factors. The total number of cases are given by:

$$k = l^n \quad (3.1.1)$$

After performing the experiment the main effects of each factor may be determined.

Table 3.1.1: A full factorial design set up of a experiment with three control factors, A, B, C, and two levels, represented by +1 and -1.

A	B	C	Response
-1	-1	-1	r_1
+1	-1	-1	r_2
+1	+1	-1	r_3
-1	+1	-1	r_4
-1	-1	+1	r_5
+1	-1	+1	r_6
+1	+1	+1	r_7
-1	+1	+1	r_8

Using **A** as an example the mean response when **A** is set “low” is subtracted from the mean response when **A** is set “high”

$$\text{Main effect of A} = \frac{r_2 + r_3 + r_6 + r_7}{4} - \frac{r_1 + r_4 + r_5 + r_8}{4} \quad (3.1.2)$$

A full factorial design gives substantial information on each factor and even their interactions. But as the number of factors grow larger the advantage of limiting the experiment to the factors with a larger influence on the systems becomes apparent. This process is called screening. When screening the cases are limited to a set which is meant to provide enough information to determine which factors are trivial and may be removed from forthcoming studies.

In applications discussed in this report the factors are almost always scalars rather than switches with as clear states as “high” and “low”.^[12]

3.1.2 Space-Filling Algorithms

Monte Carlo Sampling (MCS)

When performing computer simulations the output of two runs with the same input will be identical (unless the model contains some random element). Reproducing

results is one of many strengths with computer simulation, but in reality results vary. The reasons for this are many and the designers goal is to make the products performance insensitive to unknown variation. If the designer wants to investigate the effect variations of a certain parameters has on the output it is often a good assumption to represent the uncertain variation with a normal distribution. The valuable information extracted from these experiments are not the results from a single simulation. The questions that need answers are rather of a form similar to “How does a distribution of the input affect the output”. [9]

When creating the Monte Carlo samples the most common way is to use a pseudorandom number generator. A drawback with this is that a very large sample is required to ensure a variance close to the theoretical value. For example, when simulating a single Gaussian variable, in order to ensure a 99% probability that the sample variance is within 1% of the correct value, 133 000 samples are required. [8]

Latin Hypercube Sampling

A Monte Carlo Sampling is effective when examining how variations of a parameter affects the model. But as the input is randomly distributed it is hard to control how the sample turns out. If a batch of simulations are run in the purpose of searching for a good parameter setting it is desired to use a sample of test points that covers the domain of interest in a effective manner. It is entirely possible that nearly identical values could be generated twice with MCS, which would be considered a waste of computing power.[7]

Latin hypercube sampling algorithms are constructed in such a way that they cover parameter ranges evenly. This is done by assuring that the one dimensional projections of the multidimensional domain are well distributed. Described with three stps:[8]

1. The range of each parameter is divided into intervals of equal size.
2. One value is then chosen from each interval.
3. The parameter subsamples is permuted to minimize correlation.

Many different algorithms for creating a Latin hypercube sample exists. The approach of the algorithm differs with their purpose. Examples are: “LHS method emphasizing multi dimensional uniformity” by Deutch[7] and an algorithm “to better enable the simulated variables to have the correct means and variances” by Huntington [8]. As the algorithms for creating samples become more complex the resources needed increase. If the quality of the results increase with a more complex algorithm this may be a cost well worth paying but otherwise simplicity often is to prefer.

Comparing Sampling Sets

The sets of data points created with the algorithms described above are on some level random. This makes it reasonable to assume that it is possible to compare sets of data points created by the same, or different, algorithms. In the end one of the data sets will be considered “better”. To do this we must first define a way to numerically compare sets and also what is “better”. One relatively simple and easy to understand way of doing this is to maximize the minimum point to point distance in the sets. By doing this the risk of simulating two points that provide the same information is reduced.

More complicated methods of comparing sample distributions is different forms of discrepancy. A common used is the star discrepancy[11], D^* , of a sequence of n points $x = \{x_0, \dots, x_n\}$ in the s -dimensional cube $I = [0, u_i)$. It is defined as

$$D_n^*(x) = \sup_{P \in Y^*} \left| \frac{A(P, x)}{n} - \lambda_s(P) \right| \quad (3.1.3)$$

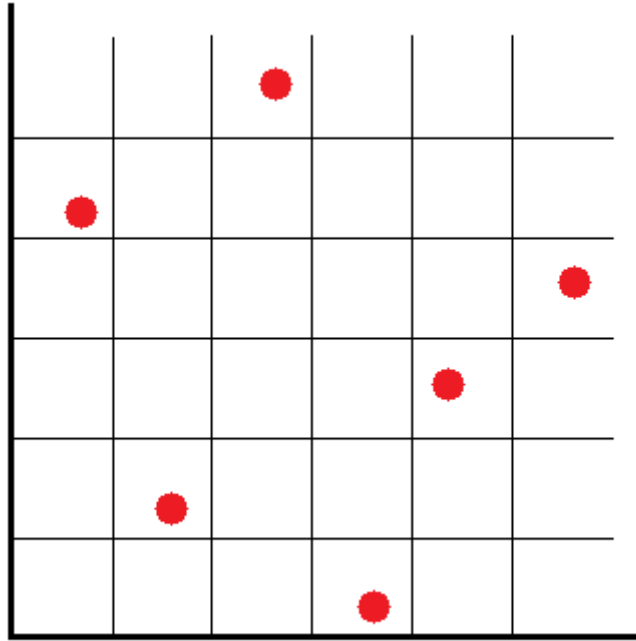


Figure 3.1.1: *LHS-cube of two design factors.*

where λ_S is the s -dimensional volume, $A(P, x)$ is the number of x_n in P and Y^* is the class of all s -dimensional subintervals P of I on the form

$$P \equiv \prod_{i=1}^s [0, u_i) \quad (3.1.4)$$

The star discrepancy is a potent measure for sample distribution in a multidimensional space but it has the drawback of being computational heavy. It is therefore suited for comparison between different sampling methods but regularly not recommended when optimizing a sample during an experiment.

3.2 Post-processing

How data obtained during simulations is presented has grate influence on its usefulness. It is therefore important that one uses the right methods of data representation and extracts the maximum amount of information from the data. In the following sections some methods of post-processing are explained.

3.2.1 Scatter Plot

After receiving the results from a batch simulation a good first choice of action is to “look” at the distribution, this may give the designer an idea of what other post-processing methods to use for continuing analysis. A scatter plot is a simple representation of data points in a graph, Figure 3.2.1. It is very intuitive in one, two ore three dimensions, versions exists for higher dimensions but this might be considered to be complicated.

3.2.2 Pareto Front

When optimizing a function with aspect to more than one variable one can not assume to always find an optima for all responses at the same control factor settings. A point is considered pareto optimal if no other point is strictly better in all aspects. That is, if one wishes to minimize two responses, A and B, a point is considered pareto optimal if no other point has a lower value of A without increasing B and vice versa. A pareto front is viewed in Figure 3.2.1.

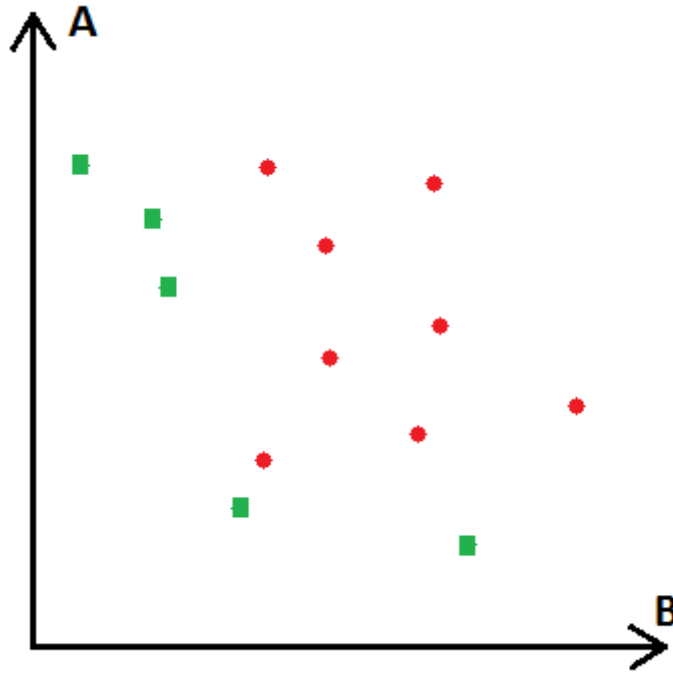


Figure 3.2.1: A 2D-Scatter plot with a pareto front. Green squares are considered pareto optimal.

3.2.3 Pearson's Correlation

Pearson's correlation coefficient[5] for a sample is defined as

$$r_{xy} = \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{(n-1) s_x s_y} \quad (3.2.1)$$
$$s_x = \left(\sum_{i=1}^n (x_i - \bar{x})^2 \right)^{\frac{1}{2}}$$

where n is the sample size. If y shows tendencies of increasing when x is increasing the correlation is positive. If y instead shows decreasing tendencies when x is increasing the correlation is negative. r takes values between -1 and 1. It reaches its extremes when the sample points are on a straight line, the correlation is then considered perfect.

4 Implementation within Microsoft Excel

Microsoft Excel is a widely used software and most people owning a computer has at least some experience working with it. While not used for advanced engineering computations Excel provides a wide range of tools for simpler calculations, organization and data presentation. Examples are histograms, filtering and pivot tables. It is also easy for the user to create nice looking charts with 3D-effects and shadowing.

With FMI Add-in for Excel the possibility arises to combine the easy to use Excel functionality with more complex models from other software. If the FMU imported into Excel is well specified it is possible to simulate and work both with input and output without possessing extensive knowledge of its origin or dynamics.

4.1 Existing functionality, FMI Add-in for Excel

This section describes the functionality in FMIE from which present thesis started from.

The FMI Add-in for Excel[3] comes with its own tab in the excel ribbon. From here the user can access functionalities as “Load FMU” and “Simulate”. The add-in supports simulation of FMUs generated by an FMI-compliant tool such as Dymola, OPTIMICA Studio by Modelon or SimulationX. After a FMU is loaded a new sheet called “FMU <name of FMU>” appears in the workbook. The user selects input and output variables that will be included in the batch simulation. The other main sheet used is called Experiment sheet and is named “EXP <name of FMU>”. When created it includes all variables earlier selected by the user in the FMU sheet. Here the different start and parameters values, input trajectories, simulation time, etc. defining the batch are entered. Also, the last value of each out data variable from the simulation results are written back to this sheet. A more complete result sheet is created if the option is selected with a check box in the ribbon before simulation.

4.2 Explanation of DOE macros

This section describes the functionality in developed for FMIE during present thesis.

The FMIE has support for batch simulations but the user has to create new cases and fill in input values by himself. The same goes for the simulation results. After a batch simulation is done all of the data is presented but it is up to the user to create a presentation of the data that is suitable for analysis. The macros written with VBA and presented below are tools to simplify the users handling with input and output from batch simulations. The general approach is that a new sheet is created by running a macro.

The FMU used in the following sections is called Furuta.fmu and is an example FMU included with the FMI Add-in for Excel. Attention should not be directed towards the numericals of the data input our output from the FMU as it is insignificant to the process.

4.2.1 Screening

The sheet designed for screening experiments is viewed in Figure 4.2.1, it consists of the following (numbers are found in Figure 4.2.1):

- Options for the sampling. The user selects an algorithm of choice in a drop down list, at the moment the two available are latin hyper cube and full factorial sampling. For LHS the number of points is set, this option is superfluous when running a full factorial design.
- Information about the sample and sheet, no user input.
 - Number of dimensions are a measure of the variable ranges spanning the sampling room.
 - Data generated changes to “yes” after the experiment sheet is updated with new data. At the moment the value **does not** change back to “no” when the data in the experiment sheet is no longer correlates with the current settings.
 - When data is generated its mean correlation is calculated. This is done by calculating the mean Pearson’s correlation coefficient, Section 3.2.3, for each indata variable combination. A variables correlation with itself is not taken into account as it is always equal to one and therefore redundant information.
- The indata from the experiment sheet. Here the user finds information about the indata of interest as well as input for the data generating algorithms. If the vary is set to “false” the value in the default column is used as input for every case. The levels option is only used for the full factorial sample.
- Button that creates the data sample based on sheet options.

1										
2	Sampling setup	1								4
3	Algorithm	Full Factorial								
4	Number of points (LHS)	25								
5										
6	Sampling information	2								
7	Number of Dimensions	3								
8	Data generated	No								
9	Sample mean correlation									
10	Related Experiment	Exp Furuta								
11										
12	Indata	3								
13	Name	Variability	Type	Unit	Default	Min	Max	Levels (FF)	Vary	Description
14	M	parameter	Real	kg	0,015				TRUE	
15	r	parameter	Real	m	0,245				TRUE	
16	m_pa	parameter	Real	kg	0,02				TRUE	
17										

Figure 4.2.1: Screening sheet created with excel macro.

4.2.2 Sensitivity analysis

The sheet designed for screening experiments is viewed in Figure 4.2.2, it consists of the following(numbers are found in Figure 4.2.2):

- The sensitivity analysis always uses Monte Carlo sampling, the user specifies the number of data points.
- Information about the sample and sheet, no user input, see Section 4.2.1.
- The indata from the experiment sheet. Each indata variable is represented by its name and useful information from the FMU. After follows user input options, the sheet will not generate a sample if input needed for the chosen distribution is absent, superfluous input is ignored.

4. Button that creates the data sample based on sheet options
5. Information about the arguments required to create the different kinds of distributions.

Sampling setup 1															
Algorithm	Monte Carlo Sampling	Generate Data 4													
Number of points	25														
Sampling information 2															
Number of Dimensions	2														
Data generated	No														
Data score															
Related Experiment	Exp Furuta														
Indata 3															
	Name	Variability	Type	Unit	Default	Distribution	Min	Max	Ar.Mean	St.Deviation	Peak	Vary	Description		
	M	parameter	Real	kg	0,015	Uniform						TRUE			
	r	parameter	Real	m	0,245	Uniform						TRUE			

Figure 4.2.2: Sensitivity sheet created with excel macro.

4.2.3 Creating results

Not all experiments demand the same methods of presentation, the design of the Post-processing sheet, Figure 4.2.3, is supposed to help the user customize the post-process according to the experiments needs. When created it contains four areas (numbers are found in Figure 4.2.3)

1. Information stating the related experiment and number of cases simulated.
2. The outdata. Basic information and statistical information about each variable.
3. A button used to add new methods of post-processing to the sheet. When pressed the form displayed in Figure 4.2.4 appears. From here the user may select and add desired methods to the sheet. After OK is pressed the Post-processing sheet changes to include selected methods, Figure 4.2.5. If the added methods require input is inserted by the user.
4. A button that creates a new sheets containing the post processing methods chosen by the user.

1														
2	Postprocessing 1													
3	Related Experiment	Exp Furuta												
4	Number of data points	3												
5														
6														
7														
8														
9														
10	Outdata 2													
11	Name	Variability	Type	Unit	Min	Max	Ar.Mean	St.Dev						
12	theta	continuous	Real	rad	5,50287	5,60036	5,554793333	0,049054872						
13	phi	continuous	Real	rad	0,473499	0,513054	0,494370667	0,019868093						

Figure 4.2.3: Post-processing sheet created with excel macro.

UserForm1

Select the methods that is to be added!

Screening

- Pareto Front
- 2D Scatter Plot

Design of Experiments

- Pearson Correlation

Sensitivity analysis

- Histogram

OK Cande

Figure 4.2.4: Form used to add post-processing methods.

1								
2	Postprocessing							
3	Related Experiment	Exp Furuta						
4	Number of data points	125						
5								
6								
7	2D Scatter Plot	X-axis	Y-axis					
8		theta	phi					
9								
10								
11	Pearson Correlation							
12			No user input					
13								
14								
15								
16								
17								
18	Outdata							
19	Name	Variability	Type	Unit	Min	Max	Ar.Mean	St.Dev
20	theta	continuous	Real	rad	5,41987	5,63215	5,5564264	0,057521717
21	phi	continuous	Real	rad	0,448553	0,542609	0,49598092	0,02552997
22								

Figure 4.2.5: Post-processing sheet with chosen methods.

4.2.4 Presenting Results

When the Create Results button in Figure 4.2.5 is pressed a result sheet, Figure 4.2.6, is created. It contains each of the post-processing methods chosen in the post sheet, Figure 4.2.5.

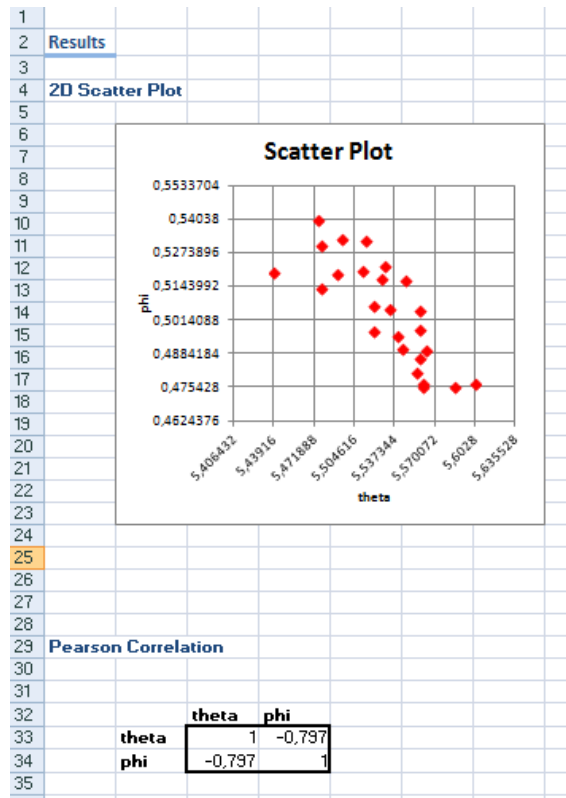


Figure 4.2.6: Result sheet displaying data as chosen in Figure 4.2.5.

5 Analysis of a transport problem

Trucks are one of the more common means of transportation when moving goods from A to B. Optimizing the transportation there are more than one response factor of interest. In this example the **response factors** are transportation time and fuel efficiency. A low transportation time and high fuel efficiency are desired, however the variables are most likely dependent on the same parameters. It may therefore not be possible to find a design point that is simultaneously optimal in both aspects. At the same time the design is sought to be robust.

This example will look at how driver behaviour affects the response factors above. More specific the **control factors** are gear shift strategy and maximum used acceleration pedal stroke, section 5.1.2. Some correlations may seem more obvious than others. For example it may be intuitive to say that a higher used acceleration pedal stroke decreases time spent to transport the goods, less intuitive is the effect it has on the fuel efficiency. The design point is also to be examined for stability with the aspect to a **noise factor**: wind condition. After the experiments are done the goal is to have a determined design point that is stable while satisfying both design aspects as well as possible.

The drivers behaviour is not actually decided in a development process, but the chosen control factors represents what could be automated/recommended to the driver through control of the vehicle. Gear shift pattern can be directly designed, if automated transmissions are used. Driver could be guided to a certain maximum accelerator pedal position, by means of active force feedback pedal.

5.1 Model description

The simulation model was created with Dymola. Most of the classes used to build the model was either a part of the Modelica Standard Library or written explicitly for this model. The exception is the design of the gearing strategy applied by the artificial driver, which was taken from the Vehicle Dynamics Library developed by Modelon AB. The model does not describe any particular truck, it was developed with the sole purpose to demonstrate the functionality of FMI for Excel and newly created macros. Hence there is no need to validate the model as such, but all parameters are set as realistic as possible, why the simulation results are assumed to be possible for some possible truck and driver.

The model used in the experiment consists of three major parts, truck, driver and road, Figure 5.1.1. A schematic view of the truck and driver components are presented in Figure 5.1.2 and Figure 5.1.4.

The drivers task is to set the truck and engine speed on target. As input it gets the target speed and the trucks momentary speed as well as the trucks engine speed. This information is used to derive a strategy for acceleration, braking and gearing.

The truck gets its input from the driver and returns the its current state as output, it is also affected by the inclination of the road. The key parts of the model will be described in this chapter, the report will not include a complete model description as it is considered of little use to the reader.

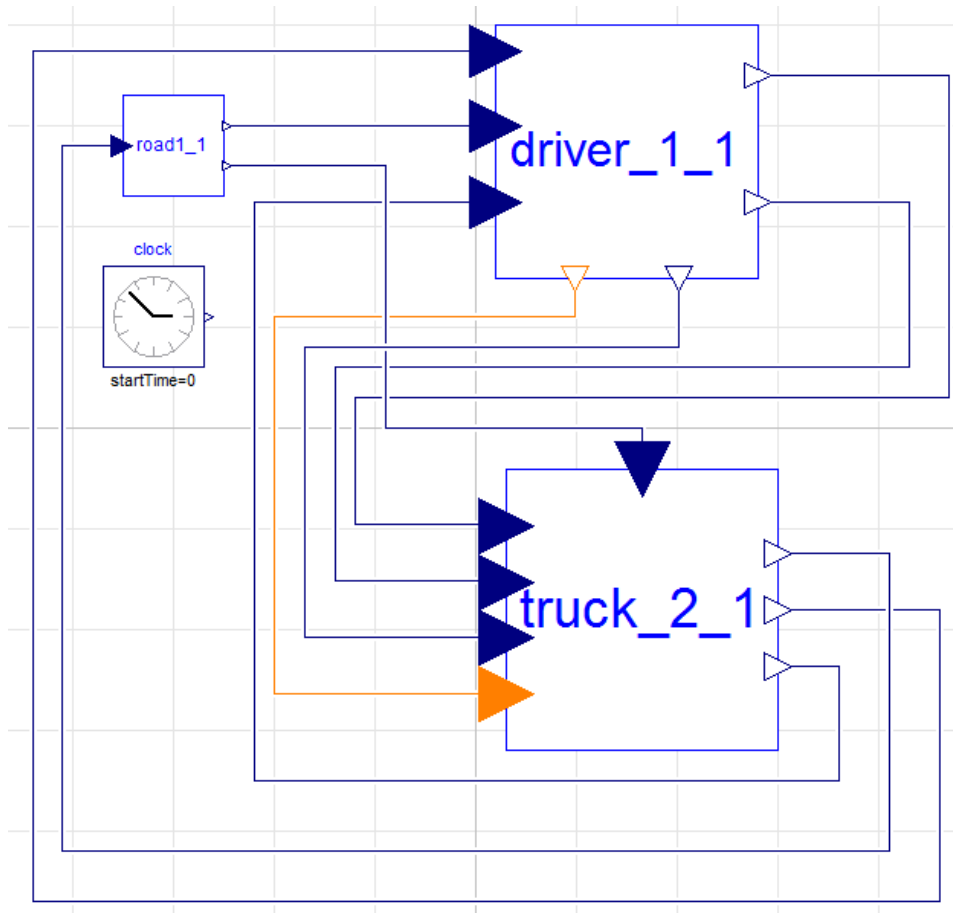


Figure 5.1.1: *The Dymola model at top level, the three main components and a clock component. The clock is only there too keep track of the time within simulation and is not crucial for model function.*

5.1.1 Truck Model

The truck model is a one dimensional model with translational movement. It consists of five parts: engine, transmission, driveline, chassis and a fuel tank, Figure 5.1.2. All the parts are independent of each other except from a few input/output variables. This makes it easy to replace single parts of the model if necessary.

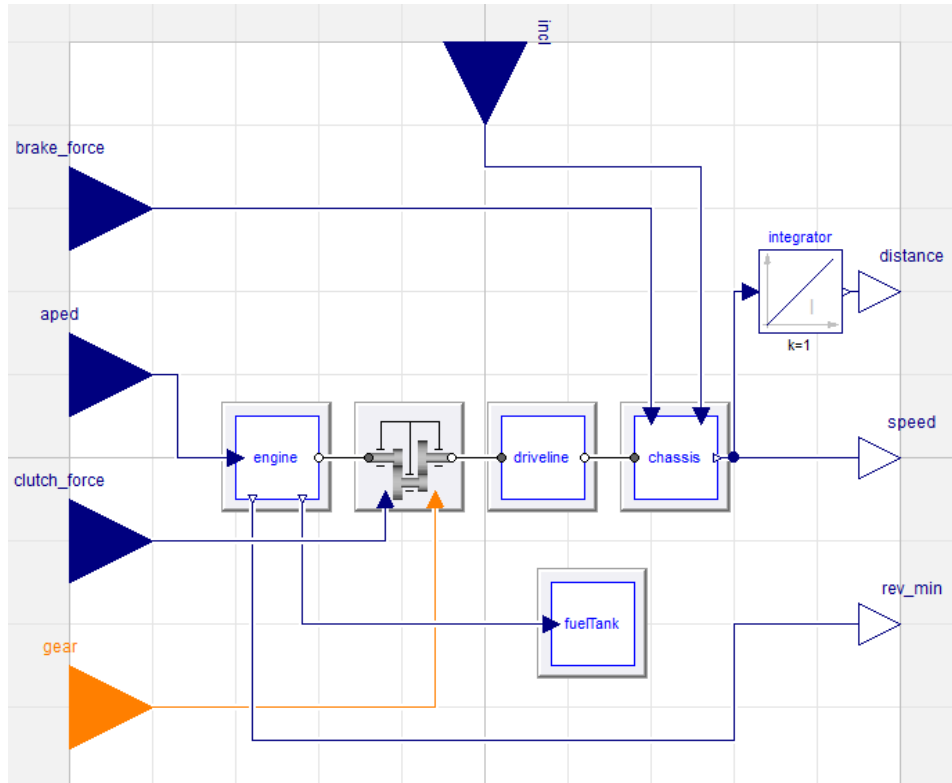


Figure 5.1.2: The truck model created in Dymola.

Engine

The engine is based on a engine map from an information booklet about the Volvo FM trucks. It has a maximum output of $1700Nm$ present within the range of $1150 - 1550rpm$. The output torque is an interpolation between two curves, one for maximum and one for minimum input from the accelerator pedal according to

$$T(\omega, ApedPos) = (1 - ApedPos) \times g(\omega) + ApedPos \times f(\omega) \quad (5.1.1)$$

$$0 \leq ApedPos \leq 1$$

where $ApedPos$ is the accelerator input and ω is the engine speed (rpm). f and g are functions for the output torque at maximum and minimum accelerator input respectively. The engine map is illustrated in Figure 5.1.3. The fuel consumption rate $[g/s]$ is interpolated from a table and dependent on current output torque and engine speed. Total amount of fuel consumed $[g]$ is stored within the fuel tank component.

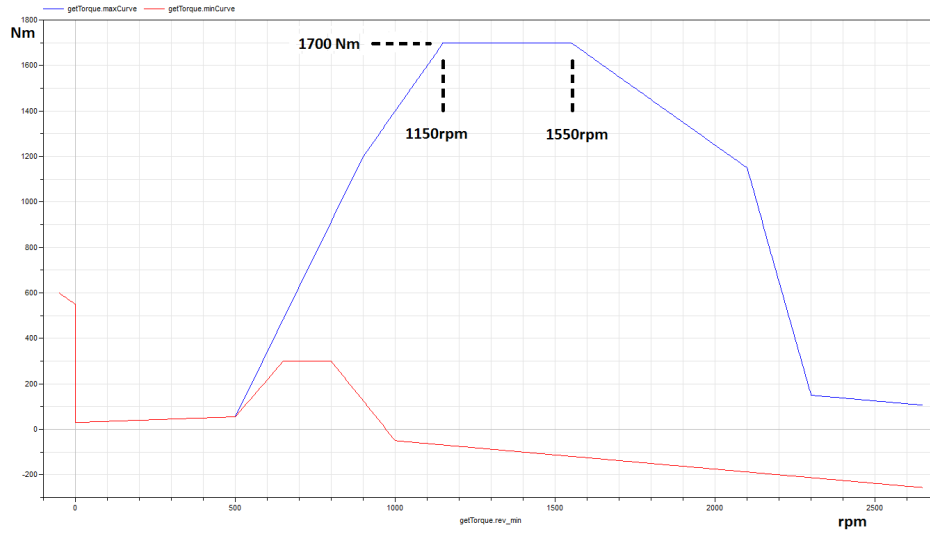


Figure 5.1.3: *Engine map.* The blue and red curves represents the engine torque output at $ApedPos=1$ and $ApedPos=0$ respectively.

Transmission and End Gear

The gear box has eight gears with the ratios according Table B.0.1. An end gear with ratio 4 was placed after the transmission.

Table 5.1.1: Ratios of each gear.

Gear	1	2	3	4	5	6	7	8
Ratio	10.18	7.16	5.04	3.75	2.72	1.91	1.35	1

Chassis

The interaction between the truck and its surroundings takes place within the chassis component. The torque on the driveline is transformed to driving forces. Resistance due to road inclination and air are also taken into account. The braking signal from the driver component is here connected to the wheels.

Wind

The air resistance is calculated according the following formula

$$F_{air} = \rho A C_d \frac{v^2}{2} \quad (5.1.2)$$

with ρ is the air density, A is the front area and C_d is the drag coefficient of the truck. To emulate different wind conditions the wind speed is changes according

$$v_{rel} = v_{truck} - v_{air} \quad (5.1.3)$$

it is important to notice that the formula in equation (5.1.2) does not take the force direction into account as the air velocity is squared. The air resistance must thus be directed in the right direction with the modification

$$F_{air} = \frac{v_{rel}}{|v_{rel}|} \rho A C_d \frac{v_{rel}^2}{2} \quad (5.1.4)$$

5.1.2 Driver Model

The components deciding the driver behaviour is a acceleration/brake regulator and a gear selector, Figure 5.1.4. The gear selector is based on components from the Vehicle Dynamics Library developed by Modelon AB[6].

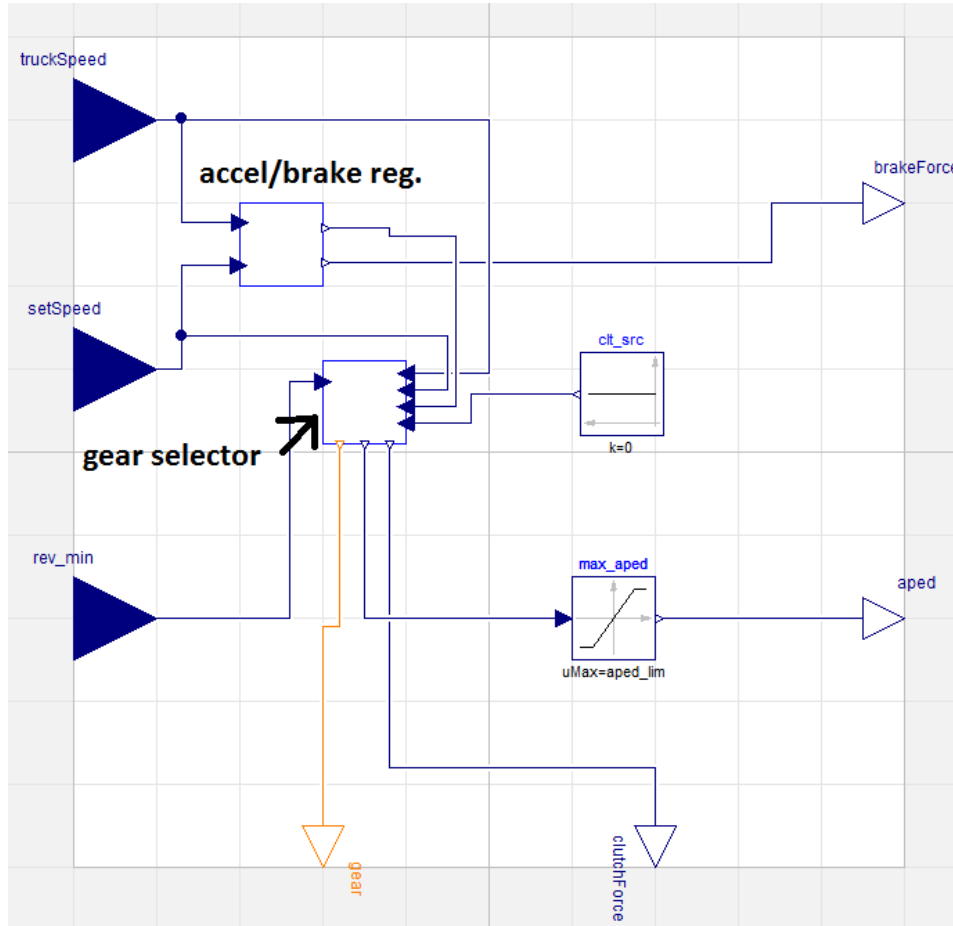


Figure 5.1.4: The driver model created in Dymola.

Accelerator and Brake

If the trucks momentary speed is slower than the target speed the driver component is supposed to increase the speed with the accelerator pedal, vice versa if the truck moves too fast the brake is applied. This is done by the use of a PI-regulator and a couple of help functions that limits the output to appropriate values and prevents simulation chattering in the transition between accelerator and brake.

Limitation of the Acceleration Pedal

The output to the accelerator pedal from the PI-regulator controlling its behaviour, u , ranges from zero to one. To control the maximal allowed input to the accelerator pedal a limiter is placed between the regulator and the acceleration pedal. The output of the limiter, y , is according to

$$y = \begin{cases} u & \text{if } u < \text{ApedMaxPos} \\ \text{ApedMaxPos} & \text{if } u \geq \text{ApedMaxPos} \end{cases} \quad (5.1.5)$$

where ApedMaxPos is the value of the limiter.

Gearing strategy

The truck is at any moment considered to be in one of the three following modes: accelerating, retarding or cruising. The gearing strategy of the truck is different for each of the three modes. For each mode an interval in engine speed exists where the gear is kept constant. If the engine speed exceeds or falls below this interval the gear is changed. Obviously the gear does not change above eight or below one. The engine speeds where the gears are changed is presented in Table 5.1.2. The values of the intervals are multiplied by a constant, α , representing the drivers gearing aggression. The intervals may be moved up or down by changing the value of α with default value one.

Table 5.1.2: Engine speeds for gear shift within different driving modes.

Mode	Gear down	Gear up
Accelerating	$\alpha 3000$	$\alpha 5000$
Retarding	$\alpha 3500$	$\alpha 4500$
Cruising	$\alpha 1800$	$\alpha 2500$

5.1.3 Road Model

During the simulation the trucks position changes in one direction. A road profile is included in the model where each position along the transportation direction has a given set speed and inclination, Figure 5.1.5 and Figure 5.1.6. The set speed is input to the driver and the inclination affects the trucks equation of motion. The Road used in these experiments was $520km$ long and the target velocity took values between 3 and $25m/s$ to represent driving both in towns and on highways.

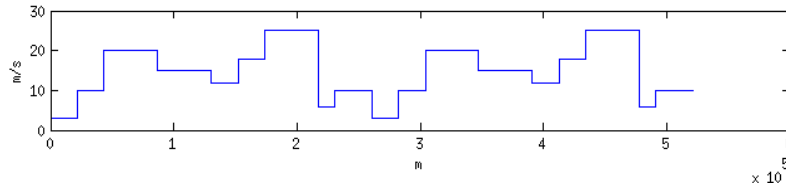


Figure 5.1.5: *The set speed as a function of distanced travelled.*

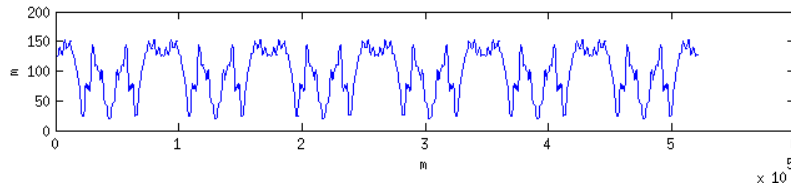


Figure 5.1.6: *The altitude of the road as a function of distanced travelled.*

5.2 Screening

The FMU was Loaded in Microsoft Excel with the use of FMI for Excel plug-in. The variables chosen for the following experiments are viewed in Table 5.2.1. An experiment sheet was created and from here a screening is performed in the variable

space to search for a design point with appropriate output. This was done by creating a Screening set-up sheet and filling in the desired variable limits. The values chosen for this run are viewed in Table 5.2.2, the variable airVelocity is constant zero, it will be used later in the process. A LHS algorithm created 500 data points in the two dimensions and the result was automatically filled into the experiment sheet. The simulations was run from the experiment sheet with the

Table 5.2.1: Variables included in the experiment.

Variable	Status	Usage
aggression	input & output	design factor
apedLim	input & output	design factor
endTime	output	response factor
fuelConsumption	output	response factor
airVelocity	input & output	noise factor

Table 5.2.2: Input variable ranges

Variable	Min	Max
aggression	0.75	1.25
apedLim	0.5	1

simulate FMU button on the FMI ribbon.

With the batch simulation finished, the result was analysed. A Post-processing sheet was created and suitable methods of result presentation were chosen. For this screening a pareto front, Section 3.2.2, and a 2D-scatter plot, Section 3.2.1, was chosen. For both figures endTime and fuelConsumption was chosen as x- and y-axis respectively. The options to include a table and input values with the pareto front were set as true. The results was created with the create results button.

The scatter plot, Figure 5.2.1, is a representation of all results from the batch simulation. For a lot of results the time it took for the truck to complete its travel are near the same. This is believed to be because the settings of the control factors are good enough for the truck to tightly follow the speed limits. As the design factor settings become worse the truck does not manage to keep up with the set speed limits of the road and the time used by the truck increases. Figures displaying how variations within a single design factor influences the results are found in Appendix A.

The pareto front, Figure 5.2.2, consists only of points from a small area of the scatter plot, this has made the shape of the graphs very different. As all the points in the pareto front, Figure 5.2.2 and Appendix B, are optimal in some aspect, and the scope of this thesis does not include a mathematical approach to optimization the choice of point along the front is up to the designer. In this experiment case 132 was chosen from the set for future analysis, it is located in within the cluster marked in Figure 5.2.2.

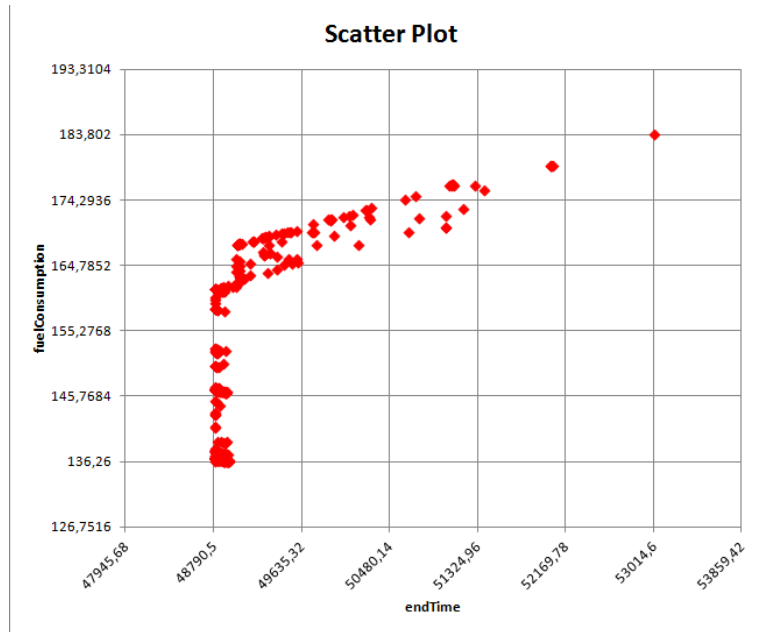


Figure 5.2.1: Screening 2D-scatter Plot. *endTime* is given in seconds and *fuelConsumption* in kg.

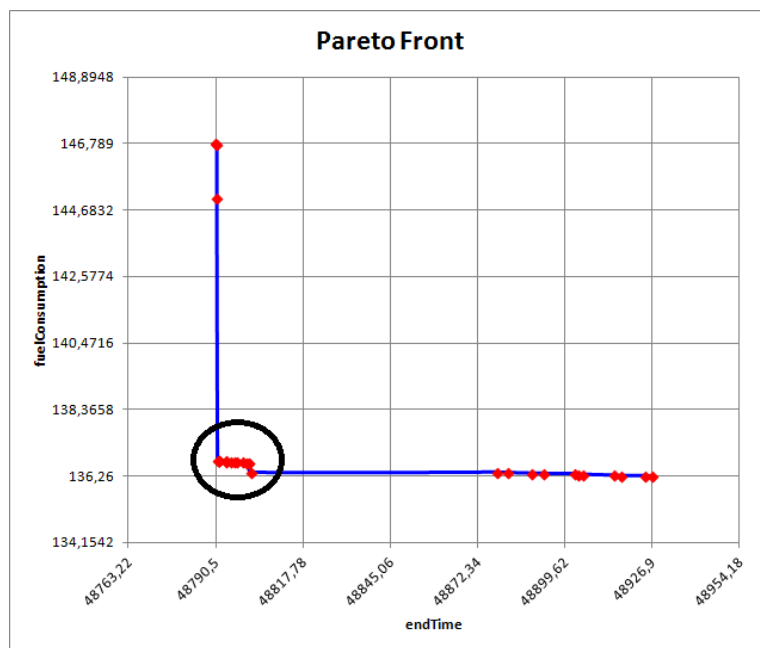


Figure 5.2.2: The pareto front of Figure 5.2.1. The data of all cases are viewed in Appendix B.

5.3 Sensitivity analysis

The simulations run in the sensitivity analysis may be run from the same experiment sheet as the screening, but this may be unsuitable as it comes with the loss of data and traceability backwards through the product development process. It is therefore recommended to create a new experiment sheet to use during the sensitivity analysis.

First thing done was to fill the input data from case 132 into the default column

(E19xE20) of the experiment sheet. Second, a set-up sheet for the analysis was created with *Create sensitivity sheet*. The data created for the sensitivity analysis is pseudo random and created according a chosen distribution. The intention, and often the case, is that the distribution of the noise is known when performing this analysis. During this experiment the distribution was not known, it was thus up to the writer to make adequate assumptions.

5.3.1 Evaluation of Robustness Within the Control Factors

Before analysing the noise factor a run was performed where the effect small changes in the design parameter has on the model was determined. This may represent the effect of for example end user variations or manufacturing tolerances. To accomplish this a set of input data were created where the design parameter values ranges from -10% to +10% of the values from case 132, previous run. The scatter plot of the results are created in a similar manner as previous plots and is viewed in Figure 5.3.1. The points does not stray far from the original output of case 132. It is therefore possible to say that the model is robust with aspect to small changes of the control factors. By performing this analysis the possibility to look at how

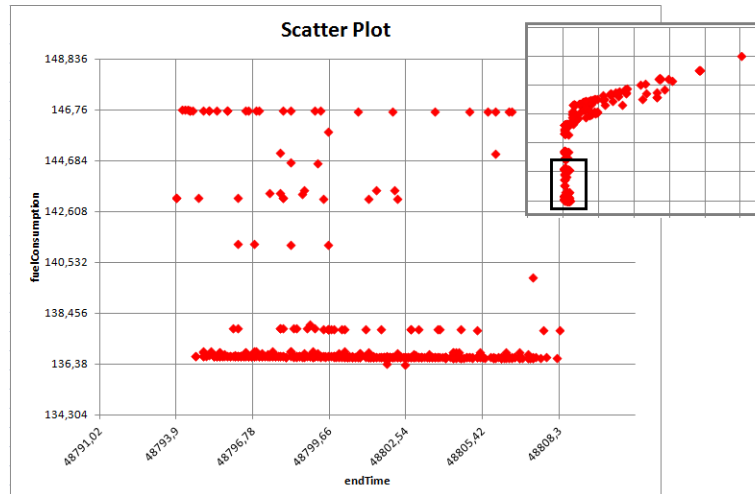


Figure 5.3.1: A scatter plot of simulations in the area around case 132. Design parameter values ranges from -10% to +10% of the values from case 132. The scatter plot from Figure 5.2.1 is in the upper right corner.

the design parameters affect the responses close to the design point arises. This is done with Pearson's correlation coefficient, Section 3.2.3. The coefficients retrieved during the simulations is presented in Table 5.3.1.

From the correlation coefficients it is possible to conclude that the relationship between apedLim and endTime is almost linear and that the aggression correlates to fuelConsumption the most.

Table 5.3.1: Pearson's correlation coefficients in the area around case 132.

	aggression	apedLim	endTime	fuelConsumption
aggression	1	0,017887132	-0,060239787	0,515192618
apedLim	0,017887132	1	-0,98245167	0,0912536
endTime	-0,060239787	-0,98245167	1	-0,171797984
fuelConsumption	0,515192618	0,0912536	-0,171797984	1

5.3.2 Analysis of Robustness due to Wind Conditions

To analyse the effect the wind has on the response factors an other sensitivity sheet was created from an experiment sheet. With constant design factor values from case 132 the noise factor airVelocity was varied. The wind velocity was here amused to be normal distributed with a mean value of 0 and a standard deviation of $4m/s$. The distribution was created with the generate data button and was then simulated from the experiment sheet.

Statistical data of the output distribution was obtained just by creating a post sheet. The data is displayed in Table 5.3.2 and it is clear that the wind distribution had almost no affect on endTime. However the min and max value of fuelConsumption differs a lot, about $60kg$ of fuel. A scatter plot of how airVelocity affects fuelConsumption was created, Figure 5.3.2. In the plot the relationship between airVelocity and fuelConsumption looks to be of second order which could be supported by equation 5.1.2.

Table 5.3.2: Statistical data of how the wind velocity distribution affects the response factors.

Response factor	Min	Max	Average	St.dev
endTime	48794	48832	48801	5.04
fuelConsumption	115.147	174.89	137.81	10.29

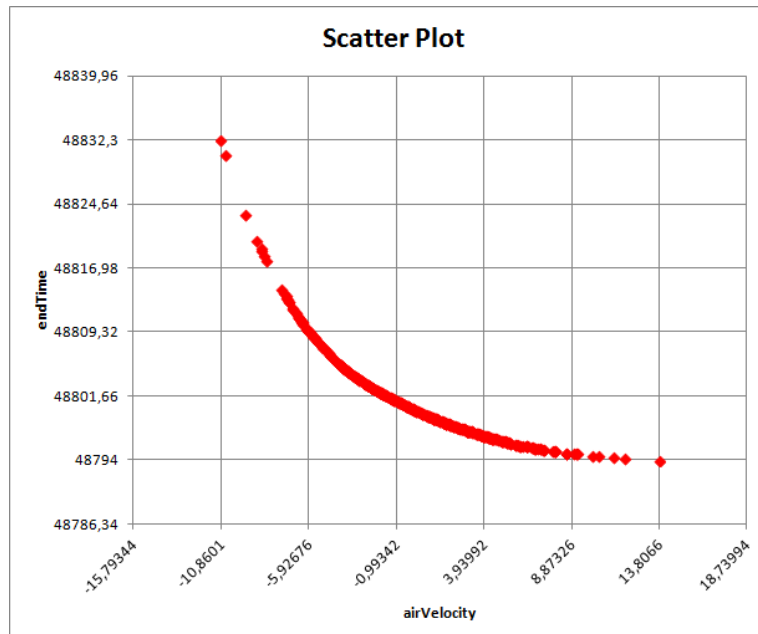


Figure 5.3.2: A scatter plot showing how the wind velocity affects the fuel consumption.

5.4 Results from the Analysis

From the screening a design point was identified, case 132, values in Table 5.4.1. It was determined in the sensitivity analysis that the design point is robust for small changes in the control factors. When taking the wind conditions into account the

design point still managed to deliver robust results of the endTime response factor but fuelConsumption was heavily affected. As the mean fuel consumed of the truck during the wind experiments is close to the value of case 132 it could be argued that the wind speed does not have a huge influence on the fuel consumption for the average case.

Table 5.4.1: Factor values of case 132

Case	aggression	apedLim	endTime	fuelConsumption
132	0.8948	0.7605	48800.1	136.698

6 Recommendations for Future Work

Solving an engineering problem with Excel as the main software leads to the discovery of both great features and lack of functionality present in more specialised engineering software. Excel is very interactive when working with pure numbers or simpler graphs but as the methods of presentation become more demanding Excel lags behind some other alternatives. A striking example of this is when creating a macro automating the creation of graphs. When a graph has a single set purpose it is easily created with Excel, but as it is a demanding procedure to change the graph when its done the importance of doing it right the first time becomes apparent. Doing this with a macro requires the that programmer has taken a numerous of possibilities and situations into account. If the end users wishes deviates from what the programmer had in mind and he/she has to make changes of his/her own some of the benefits of having pre written macros disappears. As macros are used in situations when manually creating a graph is not an option for example due to more complex algorithms they are still needed, but it is important to be aware of that the result may not be completely satisfying for a large amount of the end users.

It is nearly impossible to create a set of functions that satisfies all end users. But once the interface and work flow has been established it is easy to compliment the existing toolbox with additions. Recommendations are therefore to first evaluate and establish the interface and only after deciding on a design for the interface focus on function development.

When developing the toolbox remember to feature the strengths of Excel. The FMI standard gives users the fantastic opportunity to compliment their models with the interface of Excel. If the toolbox can communicate this feeling on the end user the product is definitively a success. On the other hand, if the end user experiences that the work could have been done easier with an other software he/she might not feel satisfied.

One of the biggest advantages with Excel is that it is widely known through the computer world and has a large user basis. The toolbox may be used as a way to give users not experienced with more complex software a chance to still make use of their powers. This leads to the dilemma that a user experienced with complexer software may be dissatisfied according above. It is therefore important to inform the end user by whom the functions are intended to be used.

When analysing a pareto front it may be useful to get a visual representation of the indata associated with the points in the pareto front. At the moment of writing, the macros explained in Chapter 4 are not able to produce such a plot. An example is provided in Figure 6.0.1. There it is possible to discover how the points of the pareto front form a few clusters. This plot may help the user to find suitable design point, as a data point in the middle of a cluster should have a higher chance of meeting the requirements of stability.

The existing macros functionality is not far from creating a plot of the type displayed in Figure 6.0.1 as it already produces (and outputs) all data needed. It is therefore recommended that this functionality is to be implemented if development continues.

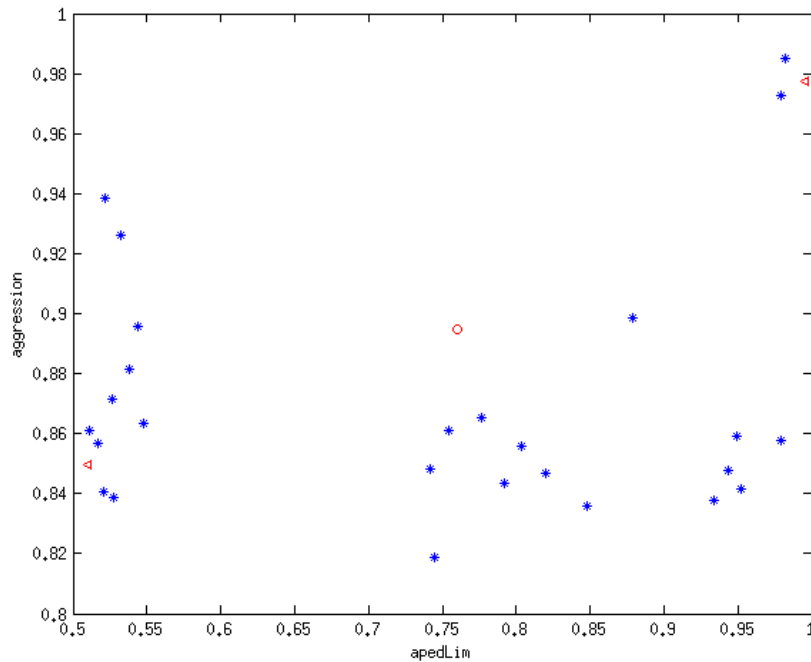


Figure 6.0.1: A scatter plot of the points in the pareto front created in Section 5.2 with the control factors on the axes (Data table found in Appendix B). The design point chosen in Section 5.2 (case 132) is displayed as a red circle. The end points of the pareto fronts are displayed as red triangles.

References

- [1] *Modelica - A Unified Object-Oriented Language for Systems Modelling*, 2012.
- [2] *Dymola Introduction Course*. Modelon AB, 2013.
- [3] *FMI Add-In for Excel User's Guide*, 2013.
- [4] Functional mock-up interface. <https://www.fmi-standard.org/>, 2013.
- [5] Social science statistics. <http://www.socscistatistics.com/>, 2013.
- [6] Modelon AB. "modelica libraries". <http://www.modelon.com/products/modelica-libraries/>, 2013.
- [7] Jared L. Deutsch and Clayton V. Deutsch. Latin hypercube sampling with multidimensional uniformity. 144:763–772, 2012.
- [8] D.E. Huntington and C.S. Lyrintzis. Improvements to and limitations of latin hypercube sampling. 13:245–253, 1998.
- [9] Christiane Lemieux. *Monte Carlo and Quasi-Monte Carlo Sampling*. Springer New York, 2009.
- [10] Gyung-Jin Park. *Analytic Methods for Design Practice*. Wiley, 2012.
- [11] Eric W. Weisstein. "star discrepancy." from mathworld—a wolfram web resource. <http://mathworld.wolfram.com/StarDiscrepancy.html>, 2013.
- [12] Öivind Andersson. *Experiment!* Springer London, 2007.

A Additional Plots

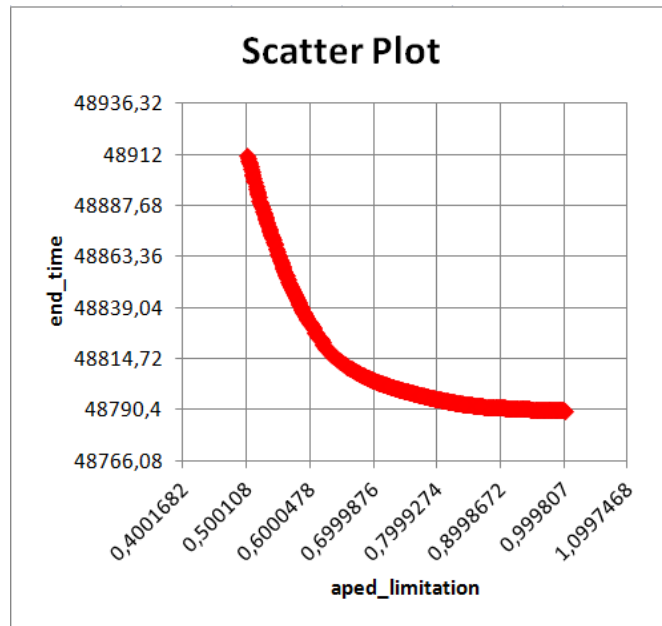


Figure A.0.1: Scatter plot of how the limitation of the accelerator pedal affects the end time. Data consists of complete set from the screening analysis performed in Section 5.2.

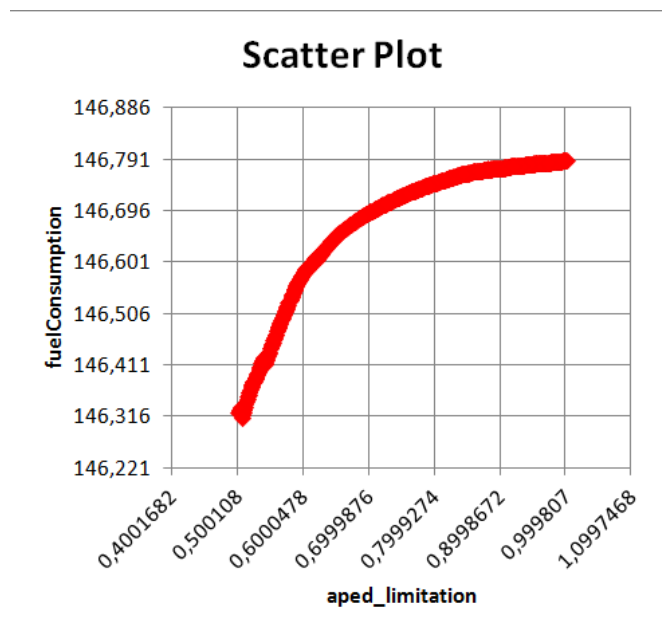


Figure A.0.2: Scatter plot of how the limitation of the accelerator pedal affects the fuel consumption. Data consists of complete set from the screening analysis performed in Section 5.2.

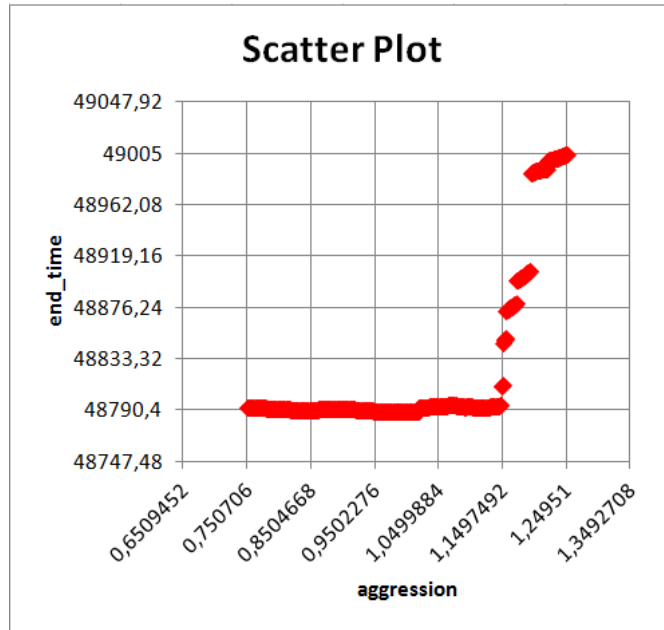


Figure A.0.3: Scatter plot of how the aggression affects the end time. Data consists of complete set from the screening analysis performed in Section 5.2.

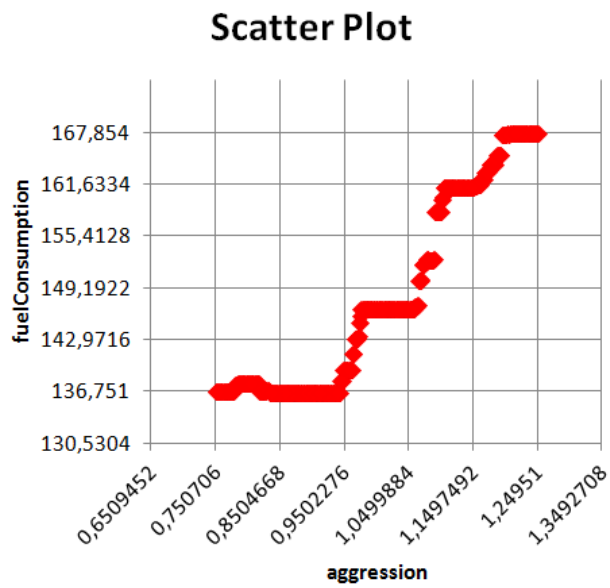


Figure A.0.4: Scatter plot of how the aggression affects the end time. Data consists of complete set from the screening analysis performed in Section 5.2.

B Pareto Data

Table B.0.1: Data of the points in Figure 5.2.2

Case	aggression	apedLim	endTime	fuelConsumption
267	0,977639961	0,995507752	48790,5	146,789
340	0,985214406	0,982549297	48790,7	146,788
474	0,972920806	0,979315082	48790,7	145,079
392	0,857645733	0,978837064	48790,9	136,751
89	0,859174126	0,94917094	48791,4	136,747
248	0,841779148	0,951874007	48791,4	136,747
483	0,847942304	0,94379966	48791,5	136,746
259	0,837568195	0,934192544	48791,6	136,743
119	0,898418513	0,878549802	48793,6	136,743
221	0,835980399	0,848206846	48793,7	136,726
190	0,846685384	0,820306511	48795,1	136,717
347	0,855595297	0,803624546	48796,1	136,71
271	0,843396339	0,792440253	48796,9	136,705
328	0,865511025	0,776508115	48799	136,703
132	0,894799061	0,760466908	48800,1	136,698
133	0,860900089	0,7546613	48800,8	136,692
203	0,848057713	0,741620081	48800,9	136,678
280	0,818881991	0,744750225	48801,6	136,386
285	0,863462599	0,547830403	48878,2	136,385
245	0,895678421	0,543810626	48881,7	136,372
273	0,881451799	0,537824654	48889	136,354
305	0,925917668	0,532027762	48892,9	136,345
44	0,938601043	0,522200441	48902,7	136,34
195	0,871540793	0,526457374	48903,6	136,318
255	0,838582262	0,527975393	48905,3	136,316
293	0,840628645	0,520604169	48915,1	136,288
102	0,856802348	0,517251576	48917,3	136,273
21	0,861099824	0,511050866	48924,6	136,273
161	0,849490479	0,510291266	48926,9	136,26