CHALMERS
UNIVERSITY OF TECHNOLOGY

Global Sensitivity Analysis of a Digital-Twin for Battery Electric Vehicles

Data Pipeline Development for Large-scale Simulations

Master's thesis in Complex Adaptive Systems

Hugo Claesson
Cassandra Svanbro

DEPARTMENT ELECTRICAL ENGINEERING

CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2026
www.chalmers.se

Master's thesis 2026

Global Sensitivity Analysis of a Digital-Twin for Battery Electric Vehicles

Data Pipeline Development for Large-scale Simulations

HUGO CLAESSON
CASSANDRA SVANBRO

CHALMERS
UNIVERSITY OF TECHNOLOGY

Department of Electrical Engineering
Division of Electric Power Engineering
Chalmers University of Technology
Gothenburg, Sweden 2026

Global Sensitivity Analysis of a Digital-Twin for Battery Electric Vehicles
Data Pipeline Development for Large-scale Simulations
HUGO CLAESSON
CASSANDRA SVANBRO

Supervisors:
Abdolreza Taheri, Volvo Group Trucks Technology & Industrial (TTI)
Fabio Santandrea, Volvo Group Trucks Technology & Industrial (TTI)

Examiner:
Torbjörn Thiringer, Department of Electrical Engineering

Master's Thesis 2026
Department of Electrical Engineering
Division of Electrical Power Engineering
Chalmers University of Technology
SE-412 96 Gothenburg
Telephone +46 31 772 1000

Cover: Digital rendition of a heavy duty BEV from Adobe Stock [1]

Typeset in LaTeX
Printed by Chalmers Reproservice
Gothenburg, Sweden 2026

Global Sensitivity Analysis of a Digital-Twin for Battery Electric Vehicles
Data Pipeline Development for Large-scale Simulations
HUGO CLAESSON
CASSANDRA SVANBRO
Department of Electrical Engineering
Chalmers University of Technology

Abstract

Modeling is a central tool in the development of battery electric vehicles, particularly in applications of validation and system optimization. This thesis aims to assist the development of simulation platforms through a large-scale global sensitivity analysis of a digital twin representing a heavy duty battery electric vehicle, using energy consumption as the primary output. The Elementary Effects Test was implemented to screen the model input space, after which a Sobol' sensitivity analysis was conducted on the most influential input parameters. The sensitivities were further explored through derivative-based measures, linear regression and Monte Carlo filtering. The results consistently identified the gross combination weight as the overwhelmingly most influential parameter. The Sobol' analysis further indicated that, depending on the drive cycle, this parameter accounted for between 25% and 77% of the total output variance. In addition the initial temperature, the aerodynamic drag area and the rolling resistance coefficient were concluded as influential input parameters but not in the same magnitude. The relative ranking of parameter importance was found to vary with the drive cycle used for the simulations. The linearity of the relation between input parameter and output was also investigated. Gross combination weight, rolling resistance coefficient and the aerodynamic drag area were found to have mostly linear effects on the output, thus suggesting that these relationships can be represented by simplified models in future work.

Keywords: global sensitivity analysis, data pipelines, digital twin, Sobol' method, Sobol' indices, elementary effects, battery electric vehicles, black-box model, high fidelity simulation, Monte Carlo simulation

Acknowledgements

We would like to express our heartfelt gratitude to everyone who supported throughout this journey and contributed in the completion of this thesis. A special thanks goes to our wonderful supervisors Abdolreza Taheri and Fabio Santandrea at Volvo Group TTI, for their guidance, encouragement and advice. We are equally grateful to our examiner Torbjörn Thiringer for making this process feel smooth and manageable, and for offering valuable feedback that helped strengthen our work. Thank you!

Hugo Claesson & Cassandra Svanbro, Gothenburg, May 2026

List of Acronyms

Below is the list of acronyms that have been used throughout this thesis listed in alphabetical order:

AD	Automatic Differentiation
BEV	Battery Electric Vehicle
CI	Confidence Interval
EE	Elementary Effects
EEM	Elementary Effects Method
ESS	Energy Storage System
GSA	Global Sensitivity Analysis
GSP	Global Simulation Platform
HPC	High-Performance Computing
KPI	Key Performance Indicator
LHS	Latin Hypercube Sampling
LR	Linear Regression
MAPE	Mean Absolute Percentage Error
MCF	Monte Carlo Filtering
OAT	One at a Time
RMSE	Root Mean Squared Error
SA	Sensitivity Analysis
VBSA	Variance-based Sensitivity Analysis

Nomenclature

Below is the nomenclature of indices, sets, parameters, and variables that have been used throughout this thesis.

Model Parameters

C_dA	Drag area of the vehicle
C_{rr}	Rolling resistance coefficient of the vehicle
T_A	Ambient temperature outside the vehicle
T_I	Initial temperature of vehicle components
$iSOC$	Initial battery-level of the vehicle
AUX	High voltage auxiliaries of the vehicle
FGR	Final gear ratio of the vehicle
GCW	Gross combined weight of the vehicle
PKS	Number of battery packs in the vehicle

Variables & Parameters

F_{drag}	Aerodynamic drag force
F_{roll}	Rolling resistance force
F_g	Grading resistance force
F_{acc}	Acceleration force
ρ	Air density
v	Velocity of the vehicle
m	Mass of the vehicle
g	Gravitational acceleration
θ	Road inclination
a	Acceleration of the vehicle

F_t	Traction force
P_t	Power from the engine
E_{tot}	Total energy consumed
t_0	Initial time of drive cycle
t_{end}	Final time of drive cycle
E_d	Energy discharged from the battery
E_r	Energy recuperated from regenerative breaking
$I(t)$	Current at time t
$V(t)$	Voltage at time t
E_c	Energy consumption from simulations
d	Distance of a given drive cycle
$\widetilde{E}_c$	Energy consumption from simulations normalized with distance
k	Number of inputs
$\mathbf{X}$	Vector of all inputs
$Y(X)$	Output of model given an input X
x_i	Input parameter i
Ω	Grid over input space
e_i	Unit vector along dimension i
EE_i^j	Elementary effect for input i and trajectory j
Δ	Difference between two points
F_i	Distribution of elementary effects for input i
r	Number of trajectories
μ_i^*	Absolute mean of the elementary effects for input i
σ_i	Standard deviation of the elementary effects for input i
S_i	First order Sobol index for input i
$X_{\sim i}$	Vector of all inputs except i
$\mathbb{E}_X(Y)$	Expected value of Y with respect to X
$V_X(Y)$	Variance of Y with respect to X
S_{T_i}	Total order Sobol index for input i
N	Number of sampling points for Monte Carlo simulations
$\mathbf{A}$	Simulation matrix for Sobol's method
$\mathbf{B}$	Simulation matrix for Sobol's method
$\mathbf{C}_i$	Simulation matrix for Sobol's method for input i
$x_i^{(j)}$	Value of input parameter i from sample point j

$y_{\mathbf{A}}$	Model value of matrix $\mathbf{A}$
$y_{\mathbf{B}}$	Model value of matrix $\mathbf{B}$
$y_{\mathbf{C}_i}$	Model value of matrix $\mathbf{C}_i$

Contents

Chapter 1

Introduction

Battery electric vehicles (BEVs) have become an increasingly important part of the global transition toward sustainable transport in recent years, with the demand expected to grow [2]. As a result of this, industries are becoming increasingly dependent on the performance and reliability of these systems. In order to meet the high demand while still maintaining high quality standards it is important that the development is efficient. Given the complex nature of modern vehicle systems it is not always clear which factors have the largest impact on performance. This is where sensitivity analysis (SA) is a useful tool. The aim of this project is to support the development of a digital twin model of a heavy duty BEV by applying large-scale sensitivity analysis.

1.1 Background

The drive for zero emissions in the transport sector and ambitious sustainability objectives across Europe is increasing the urgency and need for the next generation of simulation tools to support the development of BEVs. This is an area of development where conditions are changing rapidly, where the fast paced advancement of batteries, electric powertrains and charging infrastructure is reshaping the design space. This has resulted in a need to deliver scalable and data-driven solutions that meet industry standard.

To help further this advancement, vehicle testing through simulation is a critical supportive tool to physical testing. Using a digital twin where BEVs can be tested and refined in a virtual environment, reduces both costs and time compared to having to build physical prototypes. It is a necessary tool for meeting the current performance, quality and sustainability targets. However, for simulations to be meaningful, the underlying model has to be robust, reliable and interpretable. This requires not only validation but also a system level understanding of its behavior.

To ensure the practical usefulness of a model, it should accurately reproduce the behavior of the real components used and simulate real use conditions across a wide parameter space. Further, the accuracy of the model should be aligned with its intended application. For this reason, a deep understanding of the model is necessary

to trust it to make real life decisions, and understanding the model's sensitivities to inputs and parameters is an important step in this process. Through large-scale sensitivity analysis, implementing techniques such as Elementary Effects Method (EEM) and Sobol' indices, one can explore high dimensional parameter spaces, identify key drivers, and quantify uncertainty [3]. This exposes both capabilities and limitations of the model, which is crucial to gaining confidence in its performance. The main focus of this application is BEVs, but the methodology and analysis can be generalized to fit many types of models and simulation platforms.

1.2 Aim

The aim of this project is to bring theoretical and practical advancements to data-driven simulation and analytics of heavy duty BEVs at Volvo Group through a global sensitivity analysis (GSA), enabling insights into the accuracy of simulation models, estimating sensitivity of models on a large scale, and optimizing models for the simulation platform. By using sensitivity analysis one can better understand accuracies, robustness and uncertainties of the simulations.

The quantities of interest will be determined and the goal is to find which input parameters are most significant in simulations and lead to large or small output response when perturbation are introduced. The reasonable scale of perturbations will also be investigated.

1.3 Problem Description

This project will consider multiple key performance indicators (KPIs), with a greater focus on energy consumption per km, which plays a central role in assessing the performance of BEVs. Through looking at which of the relevant inputs that exhibit the strongest influence on different outputs and evaluating the overall variation of the output, the robust and sensitive parts of the model can be assessed. In addition, which inputs are the most important for assessing the performance can be identified and this information can be used in further development of the model.

To do this, the EEM will be used to screen which input parameters that are most important, both individual and combined through interaction effects. The most influential input parameters will be further analyzed using the variance-based sensitivity analysis method known as Sobol' method or Sobol' indices. In addition, the resolution of perturbations will be investigated to find which are suitable for the modeling platform, so that sensitivities are still recognizable but not misleading. The goal is to demonstrate the potential of sensitivity analysis tools in providing a better understanding of the models and the real systems they represent.

1.4 Limitations

Due to the complexity of the model and large scale of data, some limitations had to be made in order to keep the scope of the thesis in a feasible realm. This is done to lower the simulations needed and focuses the analysis on the most interesting input parameters for BEVs specifically.

One such limitation is that only a few relevant inputs, out of the hundreds available, will be considered for screening, these can be seen in Table 3.3. Furthermore, no input signals (time-varying inputs) will be considered and only single-valued inputs will be investigated. Additionally, they are all assumed to be independent. In a similar fashion, not all outputs will be considered. Instead the report will focus on single-valued KPIs that are good for comparing the performance of different vehicles on different routes, such as energy consumption.

The main methods are also limited. The only screening method is EEM, since it is deemed sufficient in order to draw conclusions and this is not where the focus of the project lies. Furthermore, this method do not provide an absolute quantitative ranking of the importance of the inputs and its results need to be analyzed further. For the variance based methods, Sobol' analysis will be used.

1.5 Ethical and Sustainability Considerations

The simulation of heavy-duty BEVs involves multiple ethical considerations. Compared with combustion engines, electrical engines produce lower green house gas emissions during operations, which gives BEVs a substantial environmental advantage [4]. From an ethical perspective, simulation can further support this advantage by replacing real test drives with simulations. This can have large positive effects, as it uses less resources, limits safety risks and avoids unnecessary energy consumption. Thus, the development of simulation platforms can be linked with contributions to a more sustainable vehicle development.

At the same time, BEVs are associated with larger emissions at construction [4] compared to combustion vehicles and there are additional ethical challenges associated with the battery production, especially the extraction of metals. These extraction processes are often linked to environmental harm in surrounding areas, health risks for affected workers and communities, and problematic labor conditions [5]. So, the development and thereby increased production could have negative consequences. In addition, when real-world driving data is used to support simulation models, it is important to protect the privacy of test drivers and ensure that they cannot be personally identified. In this thesis no data was used in such a way that could identify test drivers.

Chapter 2

Theory

Sensitivity analysis is a crucial step in modeling and simulation, as it enables researchers to assess how variations in input parameters influence model outputs. This chapter presents the theoretical background necessary for understanding the remainder of the thesis. It begins with the fundamental theory underlying the vehicle models, followed by an overview of various sensitivity analysis methods and their limitations.

2.1 Vehicle Models

To be able to simulate a vehicle and analyze the results a deep understanding of the different components and their connections is needed. Heavy-duty vehicles can be analyzed from multiple perspectives. Two such approaches include examining force-based interactions between the vehicle and its environment and considering the signals and energy flow between individual components.

2.1.1 Force-based Model

The longitudinal dynamics of a moving vehicle can be simplified to a force-based model. The relevant forces are aerodynamic drag, F_{drag}, rolling resistance, F_{roll}, grading resistance, F_g, and inertial force due to acceleration, F_{acc}, each defined as

$$\begin{aligned} F_{drag} &= \frac{1}{2}\rho C_d A v^2, \\ F_{roll} &= C_{rr} m g \cos(\theta), \\ F_g &= mg \sin(\theta), \\ F_{acc} &= ma, \end{aligned} \tag{2.1}$$

where ρ is the air density (kg/m^3), C_d is the drag coefficient, A is the frontal area (m^2), v is the speed of the vehicle (m/s), C_{rr} is the rolling resistance coefficient, m is the mass of the vehicle (kg), g is the gravitational acceleration (m/s^2), θ is the inclination of the road ($^\circ$), and a is the acceleration of the vehicle (m/s^2) [6] [7]. These forces are used to calculate the traction force, F_t, i.e. the force needed

from the engine. The traction force is used to determine the power needed, P_t, both defined as

$$\begin{aligned} F_t &= F_{drag} + F_{roll} + F_g + F_{acc}, \\ P_t &= F_t v. \end{aligned} \tag{2.2}$$

Figure 2.1 illustrates a schematic representation of the forces acting on a truck in motion.

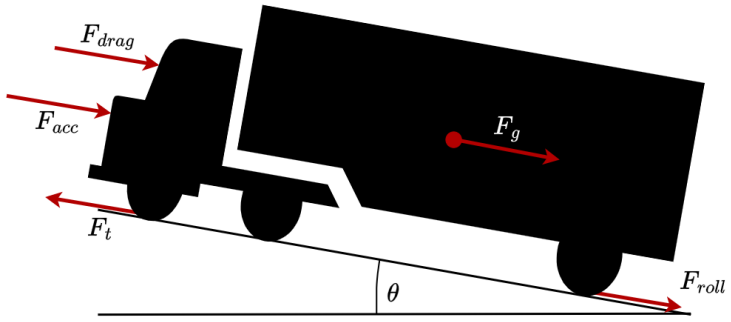


Figure 2.1: Schematic representation of the forces acting on a truck in motion.

The total energy consumption, E_{tot}, is obtained by integrating the required power over the considered time interval,

$$E_{tot} = \frac{1}{1000 \cdot 3600} \int_{t_0}^{t_{end}} P_t \, dt, \tag{2.3}$$

where t_0 and t_{end} are the initial and final times respectively. E_{tot} have the unit kWh.

2.1.2 Component-based Model

The vehicle may also be modeled as a system of interacting components [7]. In this version, each component is described by its own set of equations and transmits electrical output signals to connected components, thereby triggering responses. In this model, energy consumption is defined in terms of the resulting energy flows. Figure 2.2 illustrates a typical energy flow configuration for a BEV, where the arrows depict the flow of energy during propulsion and regenerative braking.

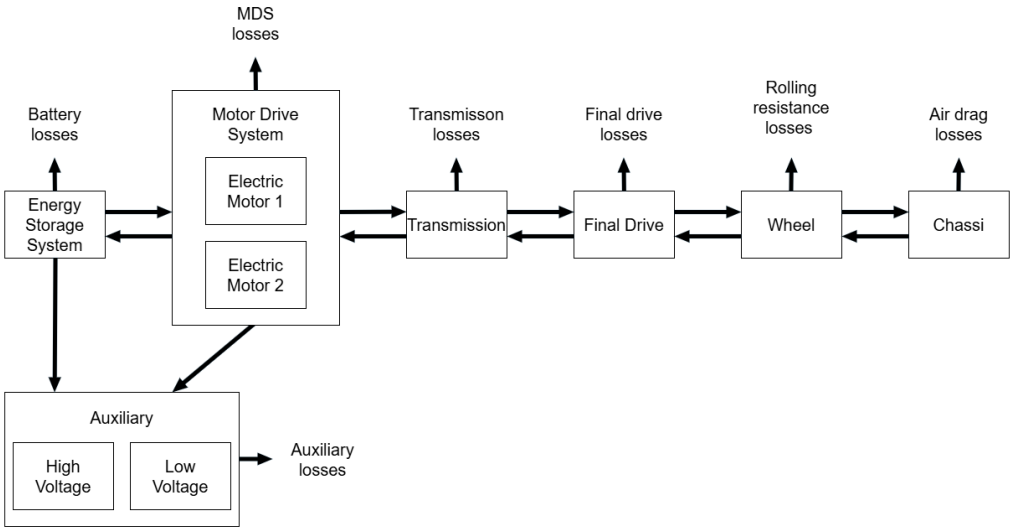


Figure 2.2: Schematics of the energy flow for a BEV.

2.1.2.1 Energy Consumption

To determine the total energy consumption of the vehicle, the energy storage system (ESS) which includes the components responsible for storing energy and delivering it to the vehicle's electrical systems is considered. The net energy consumption is calculated based on the energy discharged from the ESS, E_d, and the energy recuperated through regenerative braking, E_r. These quantities are calculated as

$$E_r = \frac{1}{1000 \cdot 3600} \int_{t_0}^{t_{end}} \max(0, I(t))V(t)\, dt, \tag{2.4}$$

$$E_d = \frac{1}{1000 \cdot 3600} \int_{t_0}^{t_{end}} \min(0, I(t))V(t)\, dt, \tag{2.5}$$

where $I(t)$ is the current from the ESS at time t, and $V(t)$ is the voltage from the ESS at time t [7]. The unit of each quantity is kWh. Thereafter, the energy consumption of the vehicle E_c is the difference between the discharged energy and the recuperated energy, calculated according to the relation

$$E_c = E_d + 0.94E_r, \tag{2.6}$$

where the factor is the efficiency of the electric motor, which is assumed to be 0.94 in this application. To be able to compare the energy consumption between different setups of the model, we normalize the energy consumption by the total distance, d, the vehicle traveled between t_0 and t_{end} as

$$\widetilde{E}_c = \frac{E_c}{d}. \tag{2.7}$$

The distance, d, have the unit kilometer km resulting in $\widetilde{E}_c$ having the unit kWh/km.

2.2 Gradient-based Sensitivity Analysis

The concept of a derivative is a fundamental part in calculus and it describes the rate of change of a function with respect to a variable. For a given function $f(x)$, the derivative can be defined as the limit

$$\frac{d}{dx}f(x) = \lim_{h \to 0} \frac{f(x+h) - f(x)}{h}. \tag{2.8}$$

This definition forms the basis for both analytical and numerical approaches of calculating sensitivities [8].

2.2.1 Analytical Approach

For a differentiable function $f(x)$, local sensitivity at a point x^* can be precisely calculated as its first order derivative, $\frac{d}{dx}f(x^*)$. This provides the exact rate of change of $f(x)$ at x^*. When a closed-form analytical expression for the derivative exists, it can be used to evaluate sensitivity across the entire input space, eliminating the need for any other technique. However, for very complex functions or functions of data points deriving such analytical expressions can be impossible. In such cases, alternative approaches can be considered.

2.2.2 Numerical Approach

For functions where closed-form derivatives are not an alternative, the derivative can usually be approximated by using finite difference methods. One such is the forward difference approximation

$$\frac{d}{dx}f(x) \approx \frac{f(x+h) - f(x)}{h}, \tag{2.9}$$

where h is a small, but non-zero perturbation. The choice of h is critical, if it is too large approximation errors increase, but if it is too small the numerical precision might create rounding errors [9]. Therefore, capturing the behavior of the function can be challenging, especially for complex and highly non-linear functions. Despite these issues, the finite difference method is valuable in scenarios involving black-box models and models that only have forward-pass, since there the are no analytical gradients.

2.2.2.1 Automatic Differentiation

For a limited subset of problems, analytical expressions are known but their derivatives are infeasible to calculate either due to complexity or discontinuity. In such cases, automatic differentiation (AD) provides an alternative to the analytical method. Through this method, one can use a numerical approach, but still get a point-wise exact gradient.

AD works on the basis that every function consists of a combination of a set of elementary functions with known derivatives. The first step is called forward-pass and goes through the function or model and computes the given elementary functions, but as it does so it also saves information about each parameters derivatives in each step and the structure of the computation. This creates a graph where the nodes are variables and edges represent the relations between these [9]. A simple example of this can be seen in Figure 2.3. This graph is then used for the second step, called backward-pass.

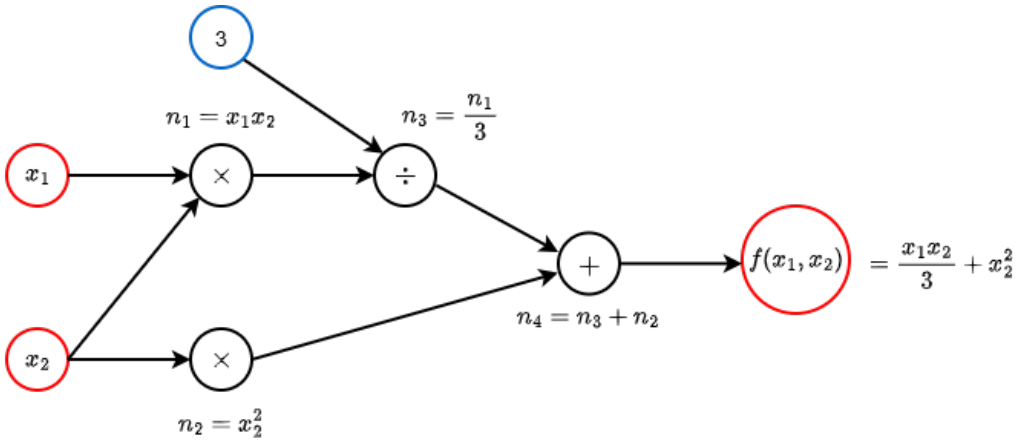


Figure 2.3: An example of how the graph created by the forward pass might look like.

To start the backward-pass, the output is assigned the derivative value one. Then the derivative is propagated backwards through the graph and in each step the derivative of the output with respect to that node is calculated. Then this is multiplied with the upstream derivative in accordance with the chain-rule, and all contributions are summed. At the end, the exact derivative of the output for each input parameter in one specific point is obtained [9].

2.3 Screening for Important Input Parameters

In high-dimensional spaces, a full SA is often computationally prohibitive due to the curse of dimensionality, whereby the number of input combinations scales exponentially with the number of dimensions [10]. In such instances, screening methods are used as an efficient and inexpensive technique to identify and rank the most influential input parameters. This provides a fast way to reduce the dimensionality by using only the most important input parameters, making the SA computationally viable [3]. It can also be implemented to screen black-box or forward pass-only models.

2.3.1 Latin Hypercube Sampling

Latin hypercube sampling (LHS) is a type of stratified sampling technique. The method works by taking each one of the k dimensions subspace D_i and to par-

tition it into M non-overlapping exhaustive strata $\{D_{i,j}\}_{j\in\{1,...,M\}}$ with the common probability of $1/M$. That is, for every dimension $i \in \{1, ..., k\}$ the subspace $D_i = \cup_{j\in\{1,...,M\}} D_{i,j}$ and $D_{i,j} \cap D_{i,l} = \emptyset$ for all $j \neq l \in \{1, ..., M\}$. Thereafter, for each dimension and each stratum one sample is drawn. The sample from the jth stratum are drawn from the density distribution $f_i(x)\mathbb{1}(x \in D_{i,j})/M$, where $f_i(x)$ is the marginal probability density function of the ith dimension. Afterwards, we can assemble a random vector X by sampling components by strata in each dimension [11].

2.3.2 Elementary Effects Method

The EEM takes an average of derivatives evaluated at different points of the input space and can it be used to screen out the parameters with the highest relative importance for a given output. It can also determine whether the input, X, affect the output, Y, linearly or non-linearly/interactively with other inputs [3]. This effectively reduces the number of iterations required for further statistical analysis by limiting the number of input parameters considered. It is a global sensitivity analysis method that captures the sensitivity of the entire input space in two sensitivity indices.

To do this, assume that we have a model with k independent input parameters, x_i, where $i = 1, ..., k$. These parameters varies over a k-dimensional unit cube, discretized into p levels, forming a grid denoted Ω. The grid Ω is constructed such that all adjacent points differ in only one dimension. In other words, if $X^{(m)}$ is an arbitrary grid point with adjacent points in j directions, then these are defined as $X^{(m)} + \Delta e_j$, $\forall j$ [3]. Because each point in Ω differs from its neighbors in only one dimension, this sampling strategy is referred to as one-at-a-time (OAT) sampling. A key design choice is how to select the grid points for computing the elementary effect.

A common approach is to use trajectories, which enhance the efficiency of OAT sampling. A trajectory is constructed as follows: the initial trajectory point, $X^{(1)}$, is randomly sampled from Ω. The second trajectory point, $X^{(2)}$, differs from $X^{(1)}$ in only dimension, i.e. $X^{(2)} = X^{(1)} + \Delta e_i$ where i is randomly selected in the set $\{1, 2, \cdots, k\}$ and Δ is a perturbation from the sampling. Subsequent points are generated by changing one dimension at a time, without using a previous dimension. For example, the third trajectory point, $X^{(3)}$, satisfies $X^{(3)} = X^{(2)} + \Delta e_j$ where $j \in \{1, 2, \cdots, k\} \setminus \{i\}$. This process is repeated until $k + 1$ trajectory points has been selected and the trajectory is complete [3].

Assume that $X^{(m)}$ and $X^{(m+1)}$ are two points in trajectory j that differ in dimension i. Then, the elementary effect (EE) for input i and trajectory j is defined as

$$EE_i^j = \frac{Y(X^{(m)}) - Y(X^{(m+1)})}{\Delta}. \tag{2.10}$$

The distribution of elementary effects for each dimension i, denoted by F_i, is obtained by sampling trajectories over Ω.

Assume that the sampling is done using r trajectories, resulting in r elementary effects for each input parameter. Then for each parameter i the mean can be calculated by

$$\mu_i^* = \frac{1}{r}\sum_{j=1}^{r}|EE_i^j|. \tag{2.11}$$

In addition to this one can also calculate the standard deviation by

$$\sigma_i = \sqrt{\frac{1}{r-1}\sum_{j=1}^{r}(EE_i^j - \overline{EE_i})^2}. \tag{2.12}$$

The two sensitivity indices, μ^* and σ, are estimates of the mean and standard deviations of F_i, and can thus be used to describe the relations between the input i and the output. The mean is used as a way to compare the overall effect, whereas the standard deviation is linked to the ensemble of effect, through both non linearity and interactive effects with other inputs. If one input has a relatively higher μ^* and σ than other inputs, it is implied that changes in that input give larger deviation in output than other inputs [3]. Both the sensitivity parameters should only be considered relative to other inputs and high values signal importance. EEM is a computationally effective method as it only requires $r(k+1)$ model evaluations, compared to the r^k evaluation needed for a full factorial design.

2.4 Variance-based Sensitivity Analysis

Variance-based sensitivity analysis (VBSA) is used to quantify how much an input affects a given output. In this approach, assuming independent inputs, the output variance is decomposed into contributions made by each input. In contrast to EEM, these methods provide clearly interpretable results compared to relative rankings. It is a global sensitivity analysis method that uses statistical quantities to estimate sensitivity.

2.4.1 Monte Carlo for Computing Sobol' Indices

VBSA requires accurate knowledge about the probability distribution of the outputs. However, this is usually not known for a complex model and therefore needs to be estimated by sampling the probability distributions of the inputs and propagating them through the model. A popular measure of sensitivity in VBSA is represented by Sobol' indices [3]. The method decomposes the variance of the model output into fractions attributed to individual input parameters and their interactions. These contributions are specified by order, corresponding to first order (single parameter), second order (pairwise interactions), and higher order interactions. Assuming that $\mathbb{E}_x(Y)$ is the expected value of Y with respect to x and $V_x(Y)$ is the variance of Y with respect to x, the first order Sobol' index is defined as

$$S_i = \frac{V_{x_i}(\mathbb{E}_{X_{\sim i}}(Y|x_i))}{V(Y)}, \tag{2.13}$$

and the total order Sobol' index is defined as

$$S_{T_i} = \frac{\mathbb{E}_{X_{\sim i}}(V_{x_i}(Y|X_{\sim i}))}{V(Y)}, \tag{2.14}$$

where x_i is the ith parameter and $X_{\sim i}$ denotes the vector of all parameters except x_i [12]. The total order could also be calculated by adding every other order Sobol' index. An intuitive interpretation for the numerator for the first order Sobol' index is that it is the expected reduction in variance if x_i could be fixed. The interpretation for the total Sobol' index is that it is the expected variance that would be left if all other factors but x_i could be fixed [12].

A standard approach to compute the conditional variances in (2.13) and (2.14) is to use Monte Carlo simulations. To illustrate the Monte Carlo method, consider a model with k input parameters, x_i, and output $Y(X)$, where $X = [x_1, \ldots, x_k]$. First, N samples of X are drawn from the assumed marginal probability distribution of each input parameter, i.e., $x_i \sim P_{x_i}$, where P_{x_i} denotes the marginal distribution of x_i. The model is then evaluated at each sampled point X, yielding an empirical approximation of the output distribution [13]. From this empirical distribution, statistical quantities such as the mean and variance can be estimated, providing the basis for computing the Sobol' indices [3].

However, this can be computationally heavy when computing Sobol' indices. Because, for the first order Sobol' index the inner conditional expectation is estimated through a set of Monte Carlo simulations and then the outer variance has to be estimated through an additional set of Monte Carlo simulations, and vice versa for total Sobol' index [3]. This requires N^2 simulations, which is computationally expensive. To lower the computational demand Saltelli [14] introduced a shortcut to computing these variances.

The approach is as follows: using two simulation matrices, $\mathbf{A}$ and $\mathbf{B}$, together with one matrix $\mathbf{C}_i$ for every input parameter. Firstly, generate a $(N, 2k)$ matrix of random numbers and define $\mathbf{A}$ and $\mathbf{B}$ as each half of the generated matrix, see (2.15) [3]. Afterwards, $\mathbf{C}_i$ is constructed by copying $\mathbf{A}$ and replacing the ith column with that from $\mathbf{B}$. Suppose we generate $\mathbf{A}$ and $\mathbf{B}$ as

$$\mathbf{A} = \begin{bmatrix} x_1^{(1)} & x_2^{(1)} & \cdots & x_i^{(1)} & \cdots & x_k^{(1)} \\ x_1^{(2)} & x_2^{(2)} & \cdots & x_i^{(2)} & \cdots & x_k^{(2)} \\ \vdots & \vdots & \ddots & \vdots & & \vdots \\ x_1^{(N-1)} & x_2^{(N-1)} & \cdots & x_i^{(N-1)} & \cdots & x_k^{(N-1)} \\ x_1^{(N)} & x_2^{(N)} & \cdots & x_i^{(N)} & \cdots & x_k^{(N)} \end{bmatrix},$$

$$\mathbf{B} = \begin{bmatrix} x_{k+1}^{(1)} & x_{k+2}^{(1)} & \cdots & x_{k+i}^{(1)} & \cdots & x_{2k}^{(1)} \\ x_{k+1}^{(2)} & x_{k+2}^{(2)} & \cdots & x_{k+i}^{(2)} & \cdots & x_{2k}^{(2)} \\ \vdots & \vdots & \ddots & \vdots & & \vdots \\ x_{k+1}^{(N-1)} & x_{k+2}^{(N-1)} & \cdots & x_{k+i}^{(N-1)} & \cdots & x_{2k}^{(N-1)} \\ x_{k+1}^{(N)} & x_{k+2}^{(N)} & \cdots & x_{k+i}^{(N)} & \cdots & x_{2k}^{(N)} \end{bmatrix}, \tag{2.15}$$

then $\mathbf{C}_i$ is constructed as described above, giving

$$\mathbf{C}_i = \begin{bmatrix} x_1^{(1)} & x_2^{(1)} & \cdots & x_{k+i}^{(1)} & \cdots & x_k^{(1)} \\ x_1^{(2)} & x_2^{(2)} & \cdots & x_{k+i}^{(2)} & \cdots & x_k^{(2)} \\ \vdots & \vdots & \ddots & \vdots & & \vdots \\ x_1^{(N-1)} & x_2^{(N-1)} & \cdots & x_{k+i}^{(N-1)} & \cdots & x_k^{(N-1)} \\ x_1^{(N)} & x_2^{(N)} & \cdots & x_{k+i}^{(N)} & \cdots & x_k^{(N)} \end{bmatrix} \tag{2.16}$$

where $x_i^{(j)}$ is the value of input parameter i from sample point j. Thereafter, the model output can be computed for all values in the sample matrices $\mathbf{A}, \mathbf{B}$ and all $\mathbf{C}_i$, resulting in the vectors $y_{\mathbf{A}} = Y(\mathbf{A})$, $y_{\mathbf{B}} = Y(\mathbf{B})$ and $y_{\mathbf{C}_i} = Y(\mathbf{C}_i)$. Saltelli's method estimate the first order Sobol' indices as,

$$S_i \approx \frac{\frac{1}{N}\sum_{j=1}^{N} y_{\mathbf{B}}^{(j)}(y_{\mathbf{C}_i}^{(j)} - y_{\mathbf{A}}^{(j)})}{\frac{1}{2N}\sum_{j=1}^{N}((y_{\mathbf{A}}^{(j)})^2 + (y_{\mathbf{B}}^{(j)})^2) - \frac{1}{2N^2}((\sum_{j=1}^{N} y_{\mathbf{A}}^{(j)})^2 + (\sum_{j=1}^{N} y_{\mathbf{B}}^{(j)})^2)}, \tag{2.17}$$

and the total order Sobol' indices as,

$$S_{T_i} \approx \frac{\frac{1}{2N}\sum_{j=1}^{N}(y_{\mathbf{A}}^{(j)} - y_{\mathbf{C}_i}^{(j)})^2}{\frac{1}{2N}\sum_{j=1}^{N}((y_{\mathbf{A}}^{(j)})^2 + (y_{\mathbf{B}}^{(j)})^2) - \frac{1}{2N^2}((\sum_{j=1}^{N} y_{\mathbf{A}}^{(j)})^2 + (\sum_{j=1}^{N} y_{\mathbf{B}}^{(j)})^2)}. \tag{2.18}$$

This approach lowers the number of simulations needed to compute Sobol' indices from N^2 to $N(k+2)$, which is helpful when $N >> k$ [15].

The first order and total order Sobol' indices satisfy the following inequalities:

$$\begin{aligned} 0 \leq S_i &\leq 1, \\ 0 \leq S_{T_i} &\leq 1, \\ S_i &\leq S_{T_i}, \\ \sum_{i=1}^{k} S_i &\leq 1. \end{aligned} \tag{2.19}$$

However, since the indices are estimated using Monte Carlo sampling, the estimators are subject to statistical error, and insufficient sampling may lead to non-converged or unstable estimates [15]. The number of simulations is therefore considered adequate when the indices the stated bounds and the relations while remaining stable with increasing sample size, such that further increases in sample size produce negligible change in both S_i and S_{T_i}.

2.5 Bootstrapping for Confidence Intervals

Bootstrapping is a method to estimating sampling variance, confidence intervals and other statistical properties from an original sampled data set. It works by resampling the obtained data to extract more information about its statistical properties. This is done with replacement, such that individual samples can be chosen multiple times. The size of the bootstrap samples can vary but is usually set to the size of the original data [16]. Figure 2.4 shows an example of how bootstrap samples can be generated.

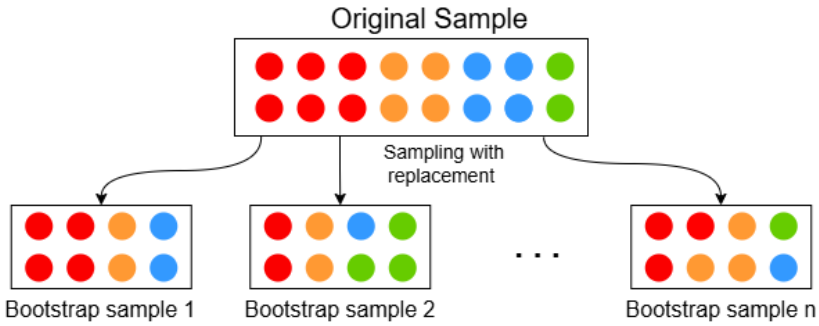


Figure 2.4: Bootstrap method to resample original data.

One property that can be extracted from the bootstrap is the original sample's estimated confidence intervals. The 95% bootstrap percentile can be found by considering the interval that contains 95% of the mean values for the different boot samples. So we can infer in which region the mean of future samples should lie. This is a very effective way of estimating an estimator without having to make assumptions about its distribution [16].

2.6 Linear Regression with Least Square Estimation

Linear regression (LR) is a fundamental method for modeling the relationship between a dependent variable and one or more independent variables. LR can be done using several techniques, however one of the most common techniques are the least square estimation. In linear regression the model output y is assumed to be linear with respect to the k input parameters $x_1, \cdots, x_k$ as

$$y = \beta_0 + x_1\beta_1 + \cdots + x_k\beta_k, \tag{2.20}$$

where $\beta_0, \cdots, \beta_k$ are the regression coefficients. In vector notation using $X = [1, x_1, \cdots, x_k]$ and $B = [\beta_0, \beta_1, \cdots, \beta_k]$ the LR would become

$$y = B \cdot X. \tag{2.21}$$

The goal of the LR method is to determine the B that best represents some given observations [17]. Assume n observations y_i respective to input $X^{(i)}$, where $i = [1, \cdots, n]$, using the least square technique, the optimum regression coefficients are

$$\hat{B} = \arg\min_B \sum_{i=1}^{n} (B \cdot X^{(i)} - y_i)^2. \tag{2.22}$$

Using the fact that the input parameters are independent and the notation $Y = [y_1, \cdots, y_n]$ and $\mathbf{X} = [X^{(1)}, \cdots, X^{(n)}]$ there exists a unique solution to (2.22) given by

$$\hat{B} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T Y. \tag{2.23}$$

2.7 Monte Carlo Filtering for Input Analysis

Monte Carlo filtering (MCF) for input analysis is a filtering method that works by defining a set of constraints that target the desired characteristics of a model, and check which samples uphold the constraints. Using the Monte Carlo method described in 2.4.1 with k input parameters and N simulations, each output gets a categorization of either behavioral, B, or non-behavioral, $\bar{B}$, depending on whether it is inside or outside of the target region described by the constraints. The categorization gives two subsets for each input parameter i, $(x_i|B)$ of size n and $(x_i|\bar{B})$ of size $\bar{n}$, where $n + \bar{n} = N$. These subsets are then used to calculate probability density functions, $f_n(x_i|B)$ and $f_{\bar{n}}(x_i|\bar{B})$ for each input parameter i. Afterwards, to identify and analyze the input parameters that drive the targeted behavior, the probability density functions for each input parameter are compared using statistical tests. When comparing the functions for a given input parameter x_i and they are significantly different it indicates that x_i is a driving factor for the behavior and if they are not significantly different it indicates that x_i is unimportant for the behavior [3].

Chapter 3

Methods

In this section all the procedures and approaches for the sensitivity analysis will be explained. This includes how the simulations were run, how parameters were defined, how the sensitivity indices for both EEM and Sobol' method were calculated, as well as any other practice deemed relevant. The methods used are general and can be applied to various fields outside of vehicle simulations. However in this case, the subject was heavy duty BEVs and therefore the specification of the relevant vehicle configurations will also be presented.

3.1 Vehicle to be Studied

The vehicle considered in this application is an electric truck of type tractor, which is a heavy duty vehicle with a tractor unit that a separate trailer can be attached to, a graphic representation can be seen in Figure 3.1. The axle configuration is 4×2 indicating two different axles, where only the rear axle is driven. Each battery pack has a nominal capacity of 90 kWh and the vehicle usually is equipped with six battery packs, leading to a total ESS capacity of 540 kWh. The vehicle has two electric motors, each capable of generating 270 kW of power and 500 Nm of torque. It also has a PIE-shift gearbox.

Figure 3.1: Graphic representation of the reference BEV [18].

3.2 Drive Cycles to be Studied

The simulations were run based on drive cycle data from physical test drives. A drive cycle contains all the necessary data needed to be able to run a simulation. Parameters dictated by the drive cycle are velocity, inclination, gear, air density, wind speed, and altitude etc. However, to limit the scope only six such drive cycles were considered. These were chosen to have variation in their inclination and labeled in three different categories: *Hilly*, *Medium* and *Flat*.

The cumulative elevation against the distance can be seen in Figure 3.2. The total elevation gain for *Medium 1* is more than double that of any other drive cycle, which is why it is labeled as a medium road instead of flat despite its rather constant inclination. The *Hilly 2* drive cycle has many small changes in elevation and is the only drive cycle that ends at a lower altitude than at the start. Other factors could have been used to decide which drive cycles to use such as speed profiles, however we deemed inclination to be the most important and interesting.

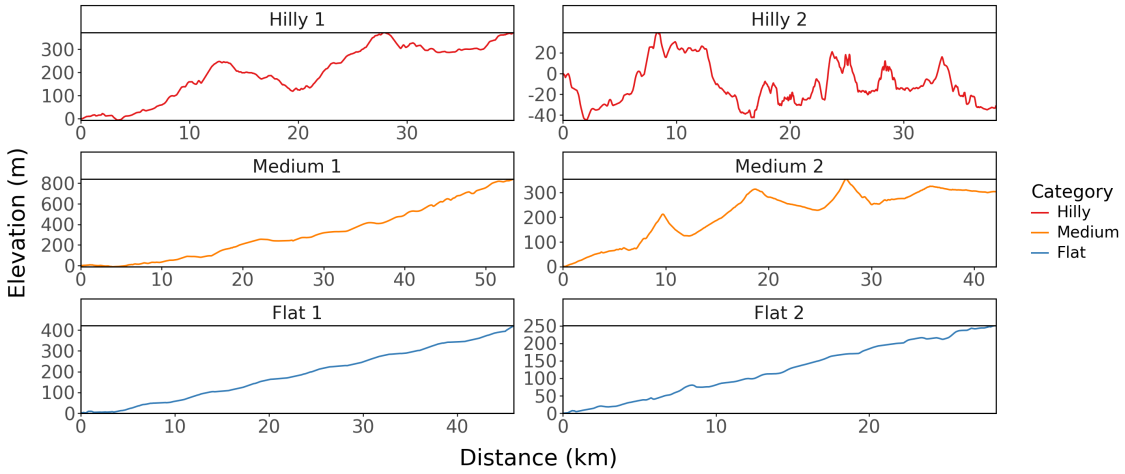


Figure 3.2: The cumulative elevation against the cumulative distance traveled for each of the six drive cycles used, categorized into three types: hilly, medium and flat.

Each drive cycle was chosen to be around 30-50 km long, since this was deemed sufficiently long to give representative results, but still guarantee that the simulations finish in a reasonable time. These can be seen in Table 3.1 along with the net elevation gain and the average velocity, discounting all zero velocities for each of the drive cycles.

Table 3.1: Total length of each drive cycle, their net elevation gain and their average velocity.

Drive Cycle	Hilly 1	Hilly 2	Medium 1	Medium 2	Flat 1	Flat 2
Length (km)	40	38	53	42	46	28
Net elevation gain (m)	370	-30	840	300	420	250
Average velocity (km/h)	58	46	68	58	66	70

3.3 Force-based Model

The force-based model of a heavy duty vehicle was the first to be implemented, see chapter 2.1.1. The purpose of this was to create a interpretable model that could be easily understood and the results could create a basis for further analysis. The model is not specific for BEVs or even heavy duty vehicles, instead it can be used to describe any moving vehicle. The model was implemented using (2.1) and (2.2) and was defined to have weight (GCW), drag area (C_dA) and rolling resistance coefficient (C_{rr}) as variable input parameters. The air density ρ was defined to be $1.22\,\mathrm{kg/m^3}$ and the gravity constant was set to $9.81\,\mathrm{m/s^2}$. To be able to run simulations with the model, a velocity profile and a slope profile were needed, with a sampling rate of 10 Hz. Both drive cycles from test drives and custom ones with constant inclination and speed were used to define the profiles. All the six drive cycles were used and the custom drive cycles had both constant velocity and inclination, but for varying values. From each simulation E_{tot} and the total distance d was extracted, to calculate the normalized energy consumption as in (2.7).

The model was implemented using `PyTorch` so that exact gradients could be calculated using its AD-tool `auto-grad` [19]. For each of the vehicle input parameters described above, that is GCW, C_{rr}, and C_dA, the exact local gradients were calculated in a OAT design. This what done for all five input parameters when not using the drive cycles and for GCW, C_{rr}, and C_dA when drive cycles were used. That is, for each of the five/three input parameters only one is changed at a time and the other were kept at nominal values. Thereafter each gradient were calculated as a function of the varying parameter. The nominal values and the ranges for the input parameters can be seen in Table 3.2.

Table 3.2: Nominal values and ranges for all the input parameters.

Parameter	C_dA ($\mathrm{m^2}$)	C_{rr}	GCW (tonne)	Inclination (%)	Velocity (km/h)
Nominal value	6	0.004	33	0	80
Range (Lo,Hi)	(2, 10)	(0.002, 0.006)	(16, 50)	(-15, 15)	(20, 100)

3.4 Global Sensitivity Analysis

A global sensitivity analysis was then done on the more complex digital twin BEV-model, Global Simulation Platform (GSP). Since GSP is a forward model, it maps input parameters to an output and does not provide a direct way to infer the inputs

from the output. Because of this, methods other than AD or analytical approaches are required to investigate the sensitivity of the model. Therefore, the statistical methods Elementary Effects Test and Sobol' indices were used. In the following sections both GSP and the chosen input space will be defined.

3.4.1 Global Simulation Platform

GSP is a simulation platform where different heavy duty vehicles can be simulated over various drive cycles. It is a component-based model, meaning it models each individual component of the vehicle and connects these using signals. It is implemented in Simulink and combines models of hardware and software to create a realistic simulation of a given vehicle and its subsystems [20]. The model simulates everything in the vehicle such as the electric motors, transmission box, batteries, wheels, chassis, cooling systems, charging systems, auxiliaries, and more.

The specification of the simulated vehicle all the way down to its subsystems is defined through configuration files. The configuration defines for example axle ratios, initial battery temperatures, wheel radius and so on. For each simulation a new configuration file has to be created so that the given input parameters are correctly inserted into the simulation. To vary the circumstances in which the vehicle is simulated, there is also a road setup, which defines every drive cycle as a time series of parameters such as inclination, speed, wind speed. These are often collected from test drives and thus mirrors actual driving experiences in order to make the simulation as similar to real life as possible. Only the input parameters for the vehicle configuration were investigated through sensitivity analysis in the thesis.

GSP simulations return numerous output signals from every component, nonetheless for our purpose only the energy consumption E_c calculated as (2.6) was considered. However, to compare the different routes accurately, the energy consumption was normalized by the length of the drive cycle, d as in (2.7) to get $\widetilde{E}_c$. This normalization accounts for the fact that the different drive cycles vary in length. Comparing only E_c would therefore not be representative, as a longer drive cycle would naturally result in a higher energy consumption.

3.4.2 Choosing Input Space

Before the GSA could be done a relevant input space had to be decided. To enable comparison of EEM results across different input parameters, the parameters had to be defined numerically with a meaningful and consistent distance between values. This is because EEM approximates and averages local derivatives of the model response with respect to each parameter. Therefore, each parameter needed to have well-defined derivatives and thus interpretable distances. This eliminated many input parameters, and therefore only the nine shown in Tab. 3.3 were chosen. The input parameters for Sobol' method were also chosen from this set of input parameters. The table shows the unit, abbreviation used throughout the report, the

minimum and maximum values (input range) and the variable type for each input parameter. All inputs were assumed to have uniform distribution.

Table 3.3: Different input parameters used in the screening process along with their variable type, their units and abbreviations used in the rest of the report.

Input parameters	**Unit**	**Abbrev.**	**Min**	**Max**	**Variable Type**
Drag area	m^2	C_dA	2	10	Continuous
Rolling resistance coefficient	-	C_{rr}	0.002	0.006	Continuous
Ambient temperature	°C	T_A	-20	40	Continuous
Initial temperature	°C	T_I	-20	40	Continuous
Initial state of charge	%	$iSOC$	40	95	Continuous
High voltage auxiliary	W	AUX	400	2000	Continuous
Final gear ratio	-	FGR	1	4	Continuous
Gross combination weight	kg	GCW	16000	50000	Continuous
Number of battery packs	-	PKS	4	8	Discrete

The ranges for the input parameters were selected based on default and reasonable values, and then extended slightly toward more extreme limits to evaluate the system's sensitivity even at the boundaries. The following list describes each parameter further.

- **Drag area** (C_dA): The drag area is also known as C_dA, it is the product of the aerodynamic drag coefficient, C_d and the frontal area of the vehicle. In this application, these were varied as a combined variable.

- **Rolling resistance coefficient** (C_{rr}): The rolling resistance coefficient is a dimensionless parameter that quantifies the energy losses associated with rolling.

- **Ambient temperature** (T_A): This is the temperature of the surroundings during the simulation. Note, that it is separate than the initial temperature. It is defined between -20 and 40°C in order to represent both the cold and hot climates where Volvo's heavy duty vehicles are in use.

- **Initial temperature** (T_I): This parameter represent the starting conditions of the vehicle, separate from the ambient temperature. One way to interpret the distinction between the initial temperature and the ambient temperature is to assume that the vehicle begins the drive cycle from a garage. The starting temperature functions as a collection of parameters that change simultaneously. When the starting temperature was varied, the following input parameters were set to the corresponding temperature value:
  - *Initial electric motor temperature*
  - *Initial electric motor inlet coolant temperature*
  - *Motor drive system inlet coolant temperature*
  - *Initial gearbox oil temperature*

For the real vehicles, it is essential that the battery temperatures are not too low when starting the drive. Therefore, the following were initialized as the maximum of starting temperature and 10°C:

- *Initial battery temperature*
- *Initial battery inlet coolant temperature*

Just as for the ambient temperature, the range was defined to reflect different climates.

- **Initial state of charge (*iSOC*):** *iSOC* is the initial battery percentage for all batteries at the start of the simulation. To ensure that the simulation is completed it could not be lower than 40%, since this caused issues when combined with a low number of battery packs. Since the SOC refers to the internal battery cell's true SOC, which is not always charged to 100% for safety reasons the maximum is set to 95% for *iSOC*.

- **High voltage auxiliary (*AUX*)**: This is the high-voltage power drawn from batteries to support auxiliaries such as thermal management systems, lights, air conditioning systems and compressors etc.

- **Final gear ratio (*FGR*)**: The final gear ratio is the number of turns the driveshaft have to make in order to rotate the wheels once. In reality it can only have certain discrete values, but for the analysis it was assumed to be continuous.

- **Gross combination weight (*GCW*)**: *GCW* is the weight of the vehicle combined with the load it is carrying, resulting in a final weight that is to be transported. The minimum is determined by the vehicle weight without any external load.

- **Number of battery packs (*PKS*)**: This is a discrete parameter that defines how many battery packs the vehicle has. Currently, the vehicle supports only 2 to 6 battery packs, but to make the analysis more comprehensible we increased the maximum to 8. As mentioned for the *iSOC*, to ensure that simulation could be completed the lowest number of packs was 4. The batteries are always dispersed evenly on the vehicle.

3.5 Elementary Effects Method

The first GSA-method that was implemented were the EEM. It was used to screen the input space to draw conclusions of which input parameters could be fixed beforehand and which should be investigated further.

3.5.1 Implementation in Python

The process of obtaining the EEM results was divided into two pipelines. In the first pipeline, the sampling was and the simulations were requested and executed in a high-performance computing (HPC) cluster. After the simulations had completed, the second pipeline was initiated, in which the results were retrieved from a database and the corresponding simulation results were processed.

3.5.1.1 Data Pipeline for the Simulation Matrix

To be able to calculate the elementary effects for each input parameter, the simulation grid had to be defined. This was done by using LHS to fill the space as best as possible. The design type chosen was trajectory design and the number of trajectories were $r = 200$ to ensure good resolution of the input space. This is equivalent to 2000 simulations per drive cycle. The sampling is defined through a matrix, called simulation matrix, where each row represent the input parameters for one simulation, and each column correspond to the values of one input parameter. To create the simulation matrix, samples were randomly drawn using the OAT strategy for each slot defined by LHS. The distribution functions `randint` and `uniform` from the python package `SciPy.stats` were used [21], depending on whether the input parameters were discrete or continuous, see Tab. 3.3. All settings to the sampling can be seen in table 3.4.

Table 3.4: The input parameter used for sampling of the simulation matrix for EEM.

Parameter	**r**	**Sampling strategy**	**Design type**	**Number of repetitions**
Value	200	Latin hypercube	Trajectory	5

To handle the large volume of simulation results and enable systematic organization, databases were used for storage. This approach capacitates efficient retrieval and processing of the results in downstream pipelines. Once the simulation matrix had been completed, it could be uploaded to a database together with the corresponding simulation requests. Each individual simulation as well as each simulation matrix was assigned a unique ID, to distinguish between different simulations and simulation batches.

3.5.1.2 Data Pipeline for Processing Results

When all the simulations had completed, the second pipeline could be used to calculate the EEs for each parameter and output. The pipeline started by retrieving the results from every simulation and calculating the normalized energy consumption $\widetilde{E}_c$, according to (2.7) for each simulation. If any simulation for some reason did not finish, the energy consumption was replaced with the mean of the remaining simulations in order to not distort the results. In addition, the other KPIs were also calculated through an existing pipeline. Then, the EEs could be computed according to (2.10), where $Y(X^{(m+1)}) - Y(X^{(m)})$ is the change in output response for the

given KPI when only changing one input parameter. From the EEs, the sensitivity indices μ^* and σ were calculated according to (2.11) and (2.12). To this end, the `SAFE Toolbox` package was used in the implementation [22].

This process was repeated for multiple drive cycles of different characterization, so that performance could be measured for all types of terrain as well as for high and low speeds. To enable comparison between different routes and different KPIs, the μ^* and σ were normalized using their maximum value for the corresponding drive cycle. We can do this since the explicit values of the sensitivity indices do not carry any information about their sensitivity, it is the relative comparison between the sensitivity indices of the input parameters that is important. After normalization, the results were analyzed for each route and route type, also the mean μ^* and σ were calculated and used for final decision-making of which input parameters to consider further in the sensitivity analysis.

For SA of this scale, with thousands of simulations and a large number of output variables, the resulting files can become very large and may exceed the available system memory during processing, causing out-of-memory failures. This can be managed through efficient columnar storage formats, such as parquet-files, ad through processing tools like the python package `polars` [23], which is especially useful since it can extract only the relevant columns from the files, and thereby reduce the computational load. Therefore, both `polars` and parquet files were used for handling the results.

3.6 Sobol' Indices on Important Inputs

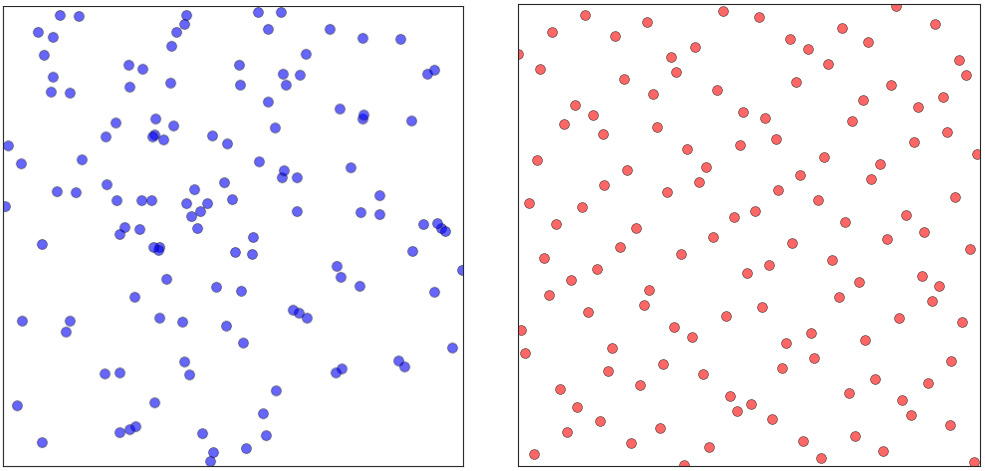
To further analyze the important inputs, Sobol' method was selected as the second GSA-method. The method decomposes the output variance with respect to the input parameters and therefore gives an understanding on the importance of each input parameter. This is the only method used that is not an OAT approach, but rather all at the same time. Two pipelines was used to perform the sensitivity analysis, the first one to generate a simulation matrix and request simulations, and the second one to calculate the Sobol' indices for the different parameters.

3.6.1 Data Pipeline for Initializing Sobol' Method

To begin the further analysis of the input parameter space, the EEM had to be performed first. From the EEM the four input parameters with highest relative μ^* and σ were chosen to be continually studied with Sobol' method. The reason for not considering all input parameters is that, that would be extremely computationally heavy. EEM requires $r(k+1)$ simulations and Sobol' method require $N(k+2)$ simulations, but in practice the N needed is much larger than r, so it is an effective approach to screen the parameters. Therefore, only the four most important input parameters from EEM were considered for Sobol' method.

Just as for the EEM, the first step to the pipeline was to create a simulation matrix. To compute the Sobol' indices, see (2.13) and (2.14), the expected variances in the numerator had to be approximated. This was done using Saltelli's shortcut. Thus, the three matrices: A, B and C, see (2.15) and (2.16), had to be defined in order to run the correct simulations.

The sampling for the matrices can be done in multiple ways. However, in this application a method called Sobol' quasi-random sequences was used. This is a low-discrepancy sequence in base 2 that generates points in the unit cube $[0, 1)$ [24]. Despite the name, there is little randomness to the points created, but since we are mostly interested in filling the input space in an efficient manner, it was deemed a good method to use [3]. An example of how a truly random sampling compares to a sampling using the Sobol' sequence can be seen in Figure 3.3. The quasi-random Sobol' sequence avoids clustering and fills the empty spaces much better than the pure random sampling, and thus were considered a suitable method for covering an entire input space. To implement this in Python the function `qmc.Sobol` from the `sciPy-stats` package was used [21], and for each matrix $2^{11} = 2048$ samples for each input parameter was used.



(a) Random sampling. **(b)** Sobol' quasi-random sampling.

Figure 3.3: Sampling of 128 points using purely random sampling and Sobol' quasi-random sampling. The latter fills the space more evenly.

Afterwards, simulations were ordered just as for EEM, using modified configuration files to implement the changes in the simulation matrices. This was repeated for the six different drive cycles to get a broader understanding.

3.6.2 Data Pipeline for Calculating Sobol' Indices

The second pipeline was initialized when all the simulations had completed and been uploaded to the database. Just as for the EEM, it retrieved the simulation results from all simulation with `polars` and calculated the normalized energy consumption $\widetilde{E}_c$, according to (2.7). Afterwards, bootstrapping was used to create 1000 bootstrap

samples to get an approximate distribution of the Sobol' indices. These samples were then used to calculate the first order Sobol' index, S_i, according to (2.17) and the total order Sobol' index, S_{T_i}, was calculated according to (2.18). This was done for each drive cycle separately and afterwards analyzed. To implement this in Python the function `sobol_indices` from the `sciPy-stats` package was used [21].

3.7 Post-Processing

After the EEM and Sobol' methods were completed, some further post-processing was done on the available simulations to investigate if there were any further behaviors that could be identified and further information extracted from the analysis.

3.7.1 Linear Regression

To examine the parameters further, a linear regression using the least square estimation was also done on the existing data. It is a way to extract further information about the parameters without having to run any new costly simulations. This includes both EEM and Sobol' method's data, to give as best as possible resolution and make the plot representative. This was implemented using `numpy` polyfit [25].

3.7.2 Monte Carlo Filtering

Another way the data was used after the initial tests were done was through implementing an input analysis using MCF. Since the input space used was the EEM simulation, it can be assumed to be equivalent Monte Carlo simulations, even if there is only a varying degree of randomness to LHS. The analysis was done using all available EEM simulations, since the analysis was done for all inputs.

The analysis was done for two behaviors, high and low $\widetilde{E}_c$. For both cases the top ten percent of simulation inputs with the investigated behavior were identified and marked as behavioral. The rest were labeled non-behavioral. Before the labeling all outputs were normalized by the largest $\widetilde{E}_c$ for the specific drive cycle, so the results evenly consider high energy consumption for each drive cycle. These two distributions were then illustrated for each input parameter to examine their driving force.

Chapter 4

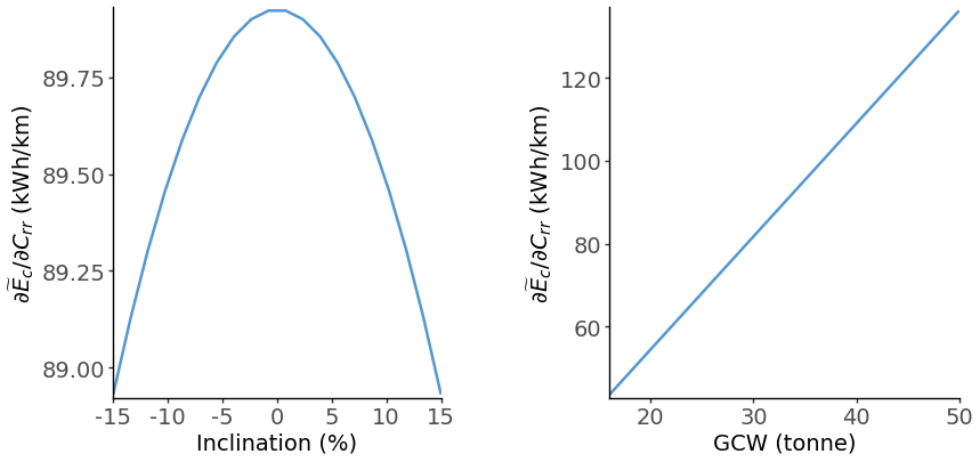
Results and Discussion

This chapter intends to present the results yielded from the work done. This includes the result for the force-based model, the EEM, the Sobol' method, linear regression, MCF as well as some further investigations.

4.1 Force-based Model

The AD analysis was conducted on the force-based model with an OAT design using both custom drive cycles with constant inclination and velocity, as well as the six drive cycles. For the first case the derivatives for all input parameters were examined, specifically their dependence on other parameter as well as inclination and velocity was investigated. For majority of the combinations the derivatives did not depend on any of the inputs, the cases where there were dependency can be seen in Figures 4.1, 4.2 and 4.3.

Figure 4.1 shows the partial derivative of $\widetilde{E}_c$ with respect to rolling resistance, $\frac{\partial \widetilde{E}_c}{\partial C_{rr}}$ and how its sensitive to the road inclination and GCW. The contribution of GCW to the derivative is much larger than that of the inclination, which changes the value of the derivative with less than one over the entire span, whereas for GCW the change is on the scale of 100. This means that the inclination of the road setup for which simulations are run, does not affect the results, instead the combination of parameters is important in order to evaluate the sensitivity of C_{rr}.

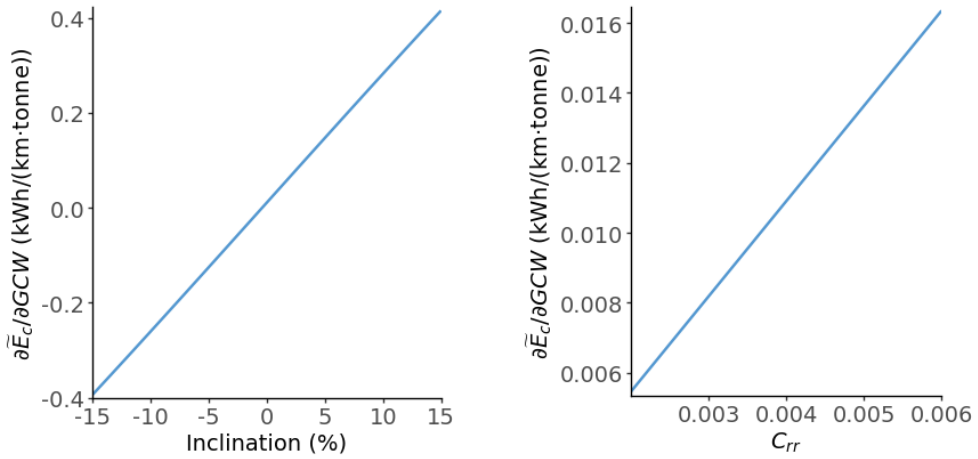


(a) C_{rr} partial derivative as a function of road inclination.

(b) C_{rr} partial derivative as a function of GCW.

Figure 4.1: The partial derivative of $\widetilde{E}_c$ (kWh/km) with respect to C_{rr} as a function of road inclination and GCW, where the other parameters are kept at fixed values.

Figure 4.2 shows the contributors to $\frac{\partial\widetilde{E}_c}{\partial GCW}$, note that the y-axis are on different scales. In this case we instead observe that the inclination is the input with largest impact on the derivative. The contribution of C_{rr} is negligible in comparison. This means that the interaction between GCW and C_{rr} is important for the sensitivity of C_{rr} but not for GCW. This can be attributed to the fact that the weight is involved with multiple forces beyond rolling resistance, and these additional forces have a greater contribution to the overall energy consumption.



(a) GCW partial derivative as a function of road inclination.

(b) GCW partial derivative as a function of C_{rr}.

Figure 4.2: The partial derivative of $\widetilde{E}_c$ (kWh/km) with respect to GCW $(tonne)$ as a function of road inclination (%) and C_{rr}, where the other parameters are kept at fixed values.

The final partial derivative can be seen in Figure 4.3. It is $\frac{\partial \widetilde{E}_c}{\partial C_d A}$ and the only dependency found for this partial derivative is the velocity. This is a quadratic dependency stemming from the quadratic velocity in the force for aerodynamic drag, F_{drag}.

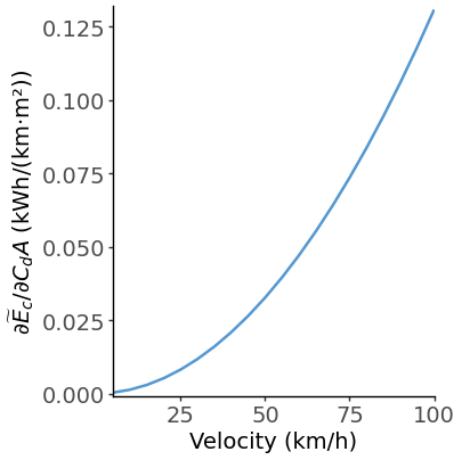


Figure 4.3: The partial derivative of $\widetilde{E}_c$ (kWh/km) with respect to $C_d A$ (m^2) as a function of velocity (km/h), where the other parameters are kept at fixed values.

To investigate the force-based model applied to the drive cycles, the energy consumption was first plotted and then all partial derivatives were calculated. Figure 4.4 shows how the energy consumption changes as each individual parameter is varied, and the remaining are kept at their nominal values. The x-axis has been normalized so that each parameter can be visualized in the same plot. From the figure you can see that all parameters has positive sensitivities and the GCW consistently has the largest sensitivity across all drive cycles. The results for C_{rr} and $C_d A$ are more similar.

The partial derivatives can be seen in Figure A.1 in Appendix A. The corresponding slopes are consistent to those shown in Figure 4.1 and 4.2. These results therefore indicate that the observed trends are representative of the model behavior also when applied to drive cycles, even though the exact numerical values differ. Since, the magnitude $\frac{\partial \widetilde{E}_c}{\partial C_{rr}}$ does not show any dependency on road inclination or vehicle velocity, thus it can be concluded that the results are very similar for the different drive cycles as well as the constant inclination and velocity.

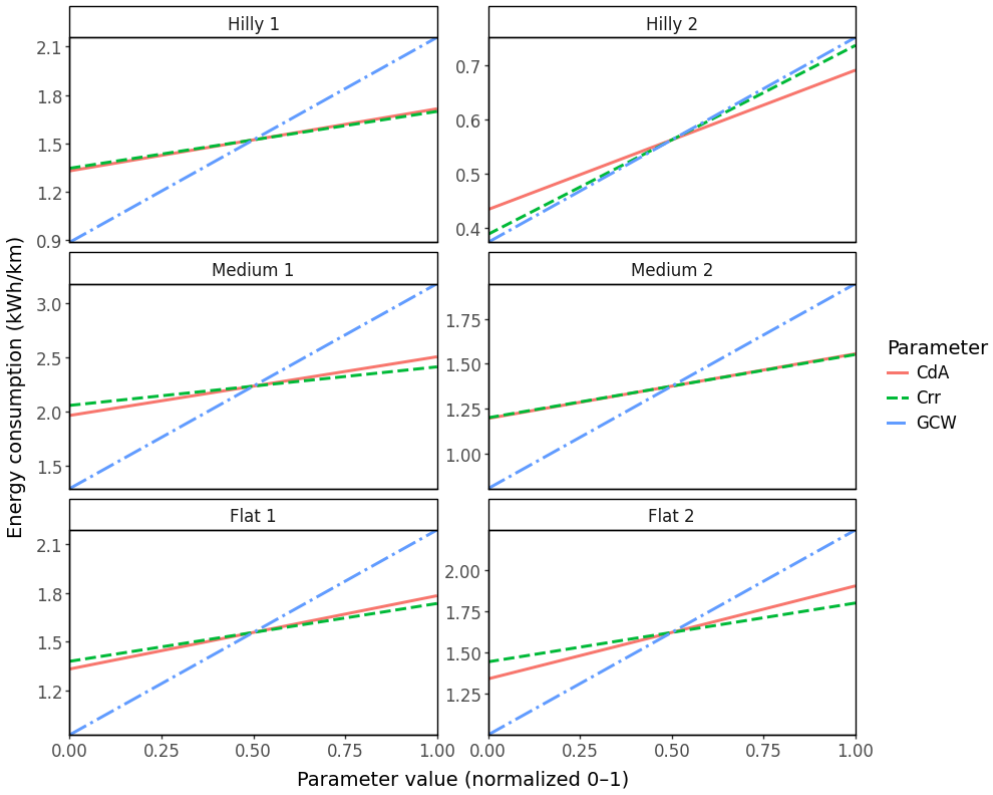


Figure 4.4: OAT parameter sweep of energy consumption across six routes. Each parameter (GCW, C_{rr}, C_dA) is varied individually over its range while the others are held fixed at nominal values, with the x-axis normalized to 0–1.

4.2 Elementary Effects

The results of the EEM include close investigation of the normalized energy consumption, $\widetilde{E}_c$, as well as an overview over the μ^* for 51 different KPIs. For all results, multiple routes of different characterizations were considered to make the results as general as possible.

4.2.1 Energy Consumption

Figure 4.5 shows the μ^* and σ calculated using (2.11) and (2.12) for six different drive cycles of different characterizations. Note, that the sensitivity indices are normalized for each of the drive cycles. The boxes represent the 95% confidence intervals (CI) for each parameter, obtained through bootstrapping over the data with 1000 repetitions. Thus, a large box suggests that there is more uncertainty in the result, but the probability of getting results outside of the box are very low. For some of the parameters the CIs are so small that they are hidden behind the point in the figure.

The fact that the CIs are rather large for some of the input parameters suggests that there might be some convergence issues. This is mostly the case for GCW,

iSOC, and T_I. However, it is important to note that even the lower part of the CI suggests the $\widetilde{E}_c$ has some of its largest sensitivities towards GCW and T_I. Another thing to note about the confidence intervals is that the σ seem to have much larger uncertainties than μ^* for four out of six drive cycles, excluding *Medium 2* and *Flat 2*. This implies that the measure of non-linear and interaction effects are less precise than the main effect. Suggesting that μ^* is a more trustworthy sensitivity index in this application. One possible explanation is that the model contains outliers or heavy tails that affect σ more strongly than μ^*. This is not unexpected, since the model is highly complex, with multiple levels and hundreds of parameters.

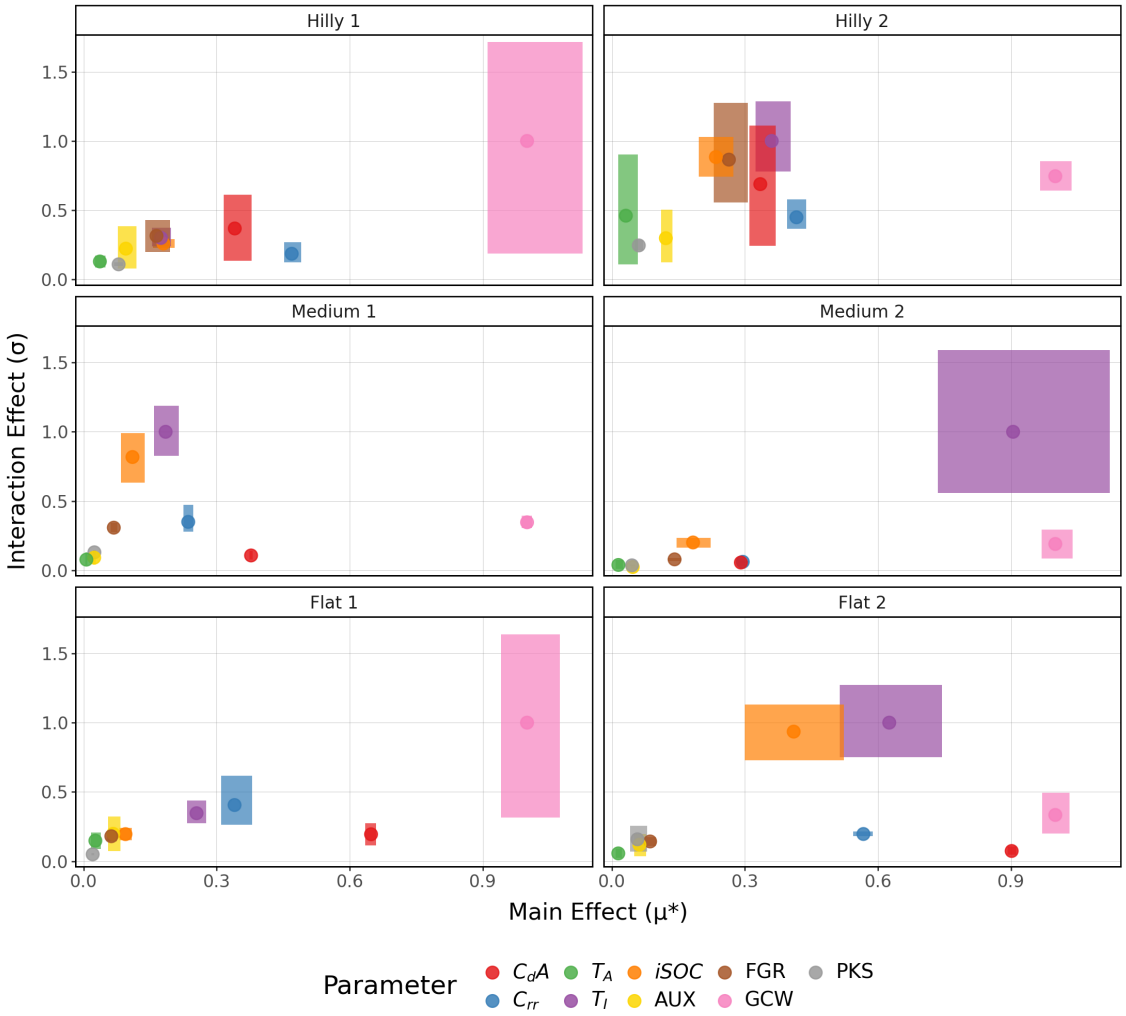


Figure 4.5: Mean effect (μ^*) and standard deviation (σ) of elementary effects for each input parameter, including their bootstrap 95% confidence intervals, for multiple routes.

As seen the input parameters with the comparably largest impact on energy consumption are GCW, T_I, C_dA and C_{rr}. As these are the top four input parameters in regards to μ^* for all routes, while also having large σ. The input parameters that seem to have the consequentially lowest impact are T_A, PKS and AUX, this can be concluded since they have the lowest μ^* for all drive cycles, and usually some of the

lowest σ as well. In addition, they have small confidence intervals, so the probability of this changing is very low. The exact values are listed in Table 4.1. Note that the results obtained are dependent on the sampling strategy and input ranges chosen, so it is not given that the results would be the same if these were changed.

Table 4.1: Normalized mean (μ^*) and standard deviation (σ) for each parameter across all routes.

	Index	C_dA	C_{rr}	T_A	T_I	*iSOC*	*AUX*	*FGR*	*GCW*	*PKS*
Hilly 1	μ^*	0.34	0.47	0.04	0.17	0.18	0.10	0.16	1.00	0.08
	σ	0.37	0.19	0.13	0.30	0.26	0.22	0.31	1.00	0.11
Hilly 2	μ^*	0.33	0.42	0.03	0.36	0.23	0.12	0.26	1.00	0.06
	σ	0.69	0.45	0.46	1.00	0.88	0.30	0.87	0.75	0.25
Medium 1	μ^*	0.38	0.24	0.01	0.19	0.11	0.02	0.07	1.00	0.02
	σ	0.11	0.35	0.08	1.00	0.82	0.09	0.31	0.35	0.13
Medium 2	μ^*	0.29	0.30	0.01	0.90	0.18	0.05	0.14	1.00	0.05
	σ	0.06	0.06	0.04	1.00	0.20	0.02	0.08	0.19	0.04
Flat 1	μ^*	0.65	0.34	0.03	0.25	0.09	0.07	0.06	1.00	0.02
	σ	0.19	0.41	0.15	0.35	0.20	0.20	0.18	1.00	0.05
Flat 2	μ^*	0.90	0.57	0.01	0.63	0.41	0.06	0.09	1.00	0.06
	σ	0.07	0.20	0.06	1.00	0.94	0.12	0.14	0.33	0.16

The relative rankings remained largely unchanged across all drive cycles. This suggests that the results are likely to hold for many drive cycles. However, there is considerable variation among possible drive profiles, and individual drive cycles may therefore produce substantially different results. Generalization across drive cycles is extremely difficult and thus the results should mainly be considered for the specific drive cycles, rather than as general truth for all drive cycles. This is especially important because all chosen drive cycles have approximately the same length, around $30 - 50$km which may downplay input parameters that influence $\widetilde{E}_c$ through smaller effects acting over longer periods of time.

Another way one can use the sensitivity indices from the elementary effects method is to judge the behavioral patterns of the input parameters. Figure 4.6 shows the relationship $\frac{\mu^*}{\sigma}$ calculated for the normalized energy consumption, $\widetilde{E}_c$, for six routes, including error bars derived from 95% confidence intervals calculated through bootstrapping with 1000 resamples. The figure illustrates how the parameters are categorized and whether they exhibit mostly linear, non-linear/interactive or mixed behaviors. When μ^* is much larger than σ, that indicates that the interactions and non-linearity are minimal and thus that the parameter is linear. If instead σ is larger than μ^* , making the ratio small, then the parameter is mostly affecting the output through interactions or non-linearity. A value close to one suggests that there is a mix of interaction and mean effect.

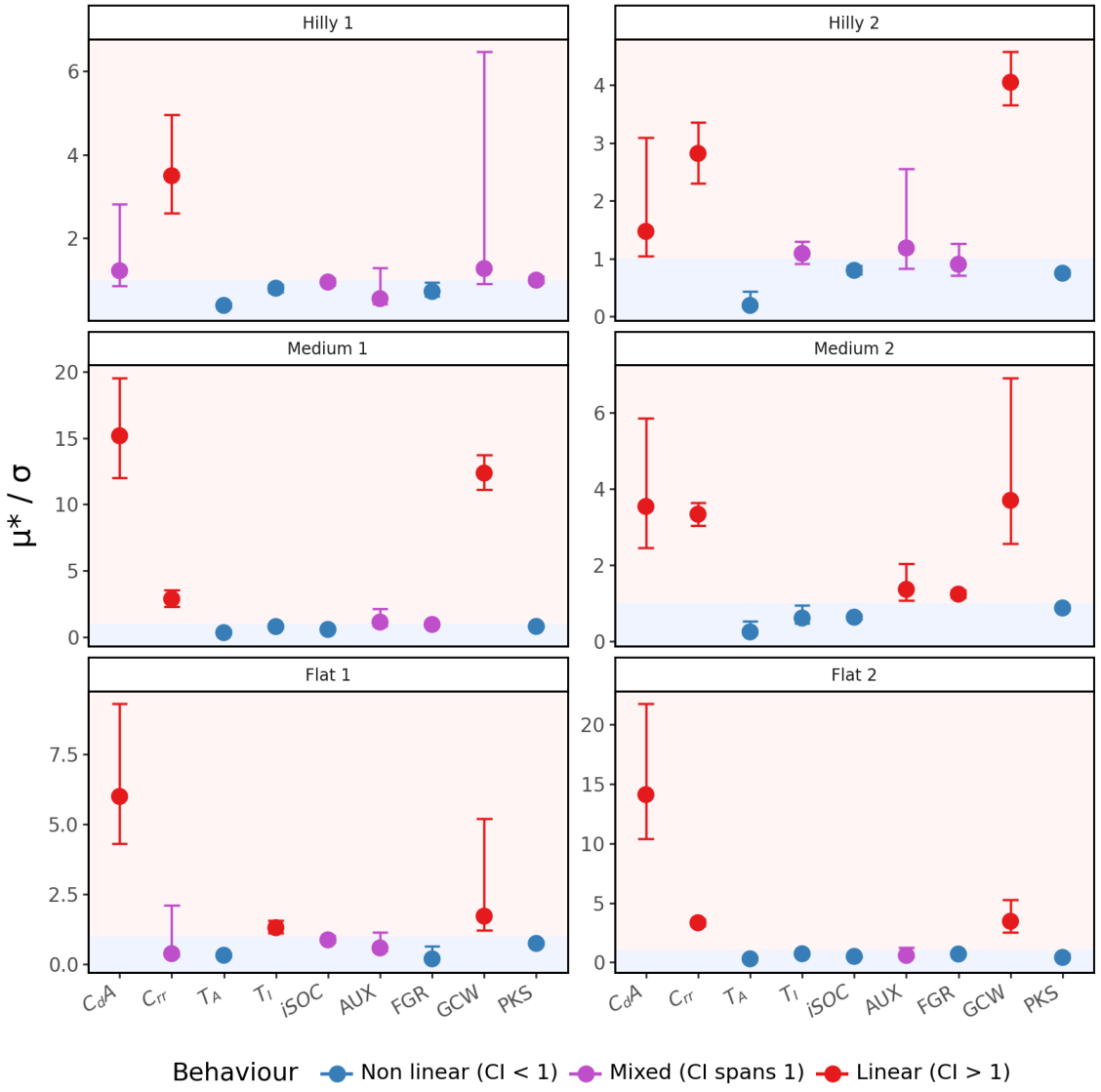


Figure 4.6: The ratio $\frac{\mu^*}{\sigma}$ calculated for the normalized energy consumption, for six drive cycles of different characterization, with error bars derived from 95% confidence intervals calculated using bootstrapping with 1000 resamples. This is a measure of the parameter behavior in the model.

As seen in figure 4.6 the parameters C_dA, C_{rr} and GCW are all repeatedly labeled linear, whereas the other parameters have more mixed results. These results are not absolute and the uncertainty presented by the confidence intervals in Figure 4.5 can largely affect the perceived behavior, this is seen through the error bars. Although there are large uncertainties, there is some indication that the three input parameters mentioned above mostly have linear effect. This is because even for the drive cycles where they span both the linear and non-linear values, the error bar covers the linear values to a larger extent, for example both GCW and C_dA for *Hilly 1* exhibit this behavior.

4.2.2 Aggregation of Multiple KPIs

The EEM was also implemented on an aggregated list of KPIs, resulting in the mean effects shown in Figure 4.7. The figure shows a normalized μ^* for multiple KPIs averaged over six drive cycles. The mean effects have also been binned to distinguish the importance of each input parameters. In general, the impactful input parameters vary depending on which output is considered. For a majority of the KPIs there are few $(2-5)$ higher order effects and then the rest of the input parameters have small or no effect on the output. This is despite the fact that the medium effect encompasses the entire middle half of the scale, implying that, for each KPI, only a limited number of input parameters need to be considered. This is very useful to know for further applications, since any further analysis into these KPIs can be limited to include only the parameters with high sensitivity.

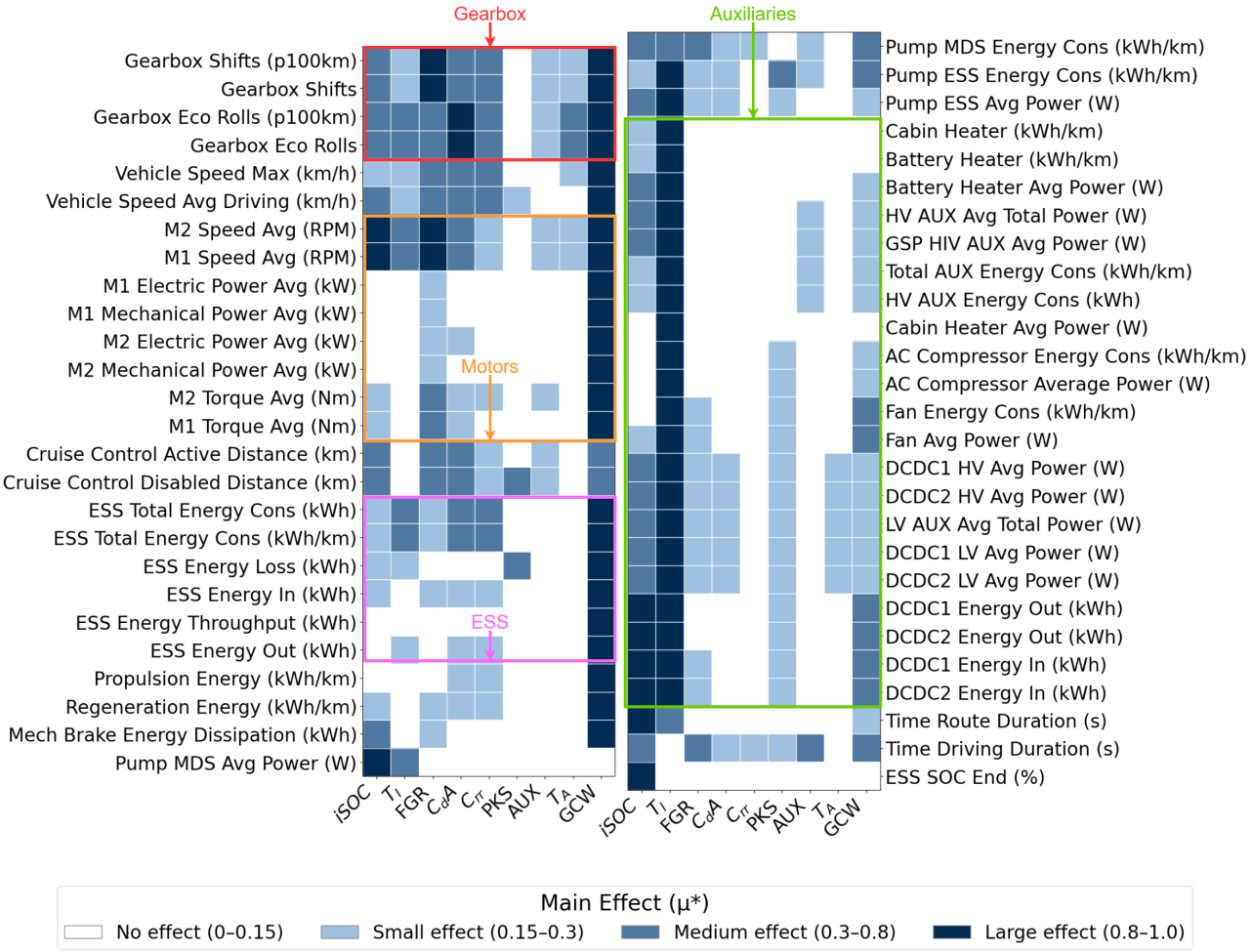


Figure 4.7: Normalized mean effect (μ^*) for multiple KPIs and input parameters, averaged over the six drive cycles. Three components subsystems are marked in color: electric motors (orange), ESS (pink), gearbox (red) and auxiliaries (green). The mean effect is divided into four categories: no effect, small effect, medium effect and large effect, and show which input parameters are reasonable to consider for each KPI.

Notably, both GCW and the initial temperature, T_I were found to have a large effect

on the same number of KPIs (23), which also was the largest, followed by *iSOC* (9). The initial temperature have the largest impact for auxiliaries and heating related KPIs, whereas *GCW* mostly affects KPIs related to the gearbox, the motors, and to the ESS. The *GCW* is found to be the single most important input parameter in the EEM, through this analysis it seems that this is in large because of its effect on the powertrain. A higher *GCW* requires a higher torque to move, this puts stress on the entire drive line, including gearbox the and motors. So that for example the motor energy consumption has a large sensitivity towards *GCW* is not surprising. The ESS KPIs are very closely related to $\widetilde{E}_c$ as it is the basis of calculating E_c, see (2.6), therefore it is reasonable that the μ^* for the aggregated ESS-KPIs reflect the results from the EEM where *GCW* is most important. For *iSOC*, the high impact areas are more spread through different parts of the vehicle.

Another thing evident from the figure is that there are certain KPIs that have larger sensitivity towards multiple input parameters, suggesting that they might have a large sensitivity overall. One such area is the gearbox, that is easily affected by *GCW*, *FGR*, C_dA, C_{rr} and *iSOC*. The same is true for average motor speed (RPM) where *FGR*, *GCW* and *iSOC* all have a large effect. It might be that these are generally sensitive or simply sensitive to the parameters considered in this application.

4.2.3 Output Distribution

The output distribution for each drive cycle for the EEM can be seen in Figure 4.8. The flat road cycles have a lower $\widetilde{E}_c$ than the medium and hilly ones. However there is one exception, *Hilly 2* has the second lowest mean energy consumption. This is likely because, despite substantial variation in inclination, the drive cycle has a net negative elevation change, whereas the others are characterized by a positive overall elevation gain. This means it has more opportunities for regenerative braking, which considerable impacts the overall energy consumption.

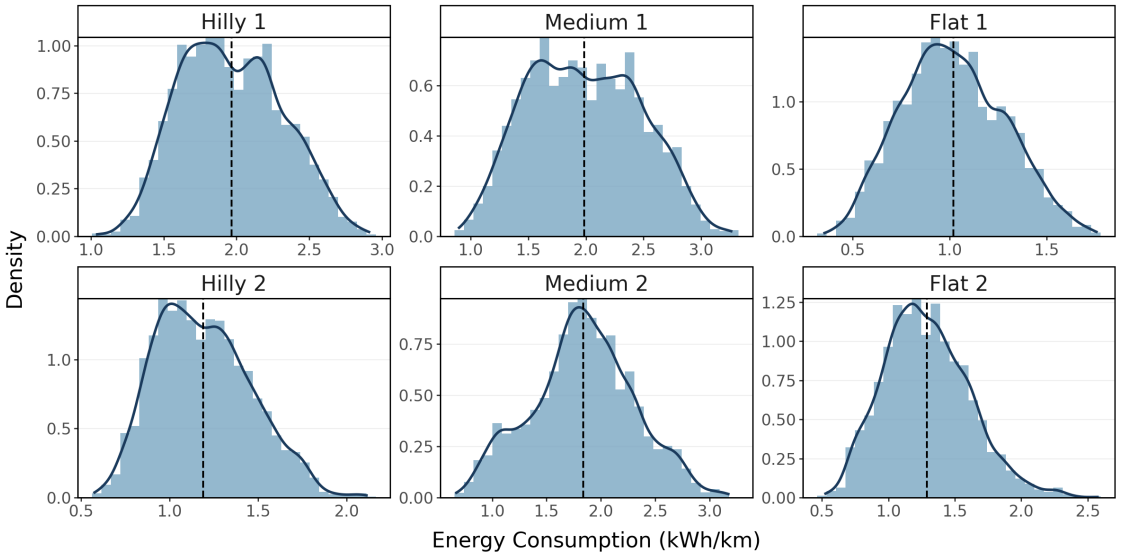


Figure 4.8: Distribution of energy consumption for each drive cycle, the dotted line shows the mean.

Figure 4.9 expands on how the distribution changes as the inclination varies. The middle shows $\widetilde{E}_c$ in segments were the road has more than three degree uphill slope and the right shows the same for downhill slopes. The flat drive cycles have no segments where they have a ± 3 degree slope so they are only shown in the subplot for all segments. As seen the energy consumption is considerably higher when inclination exceed three degrees and then stretches into the negatives for the segments with steep downhills.

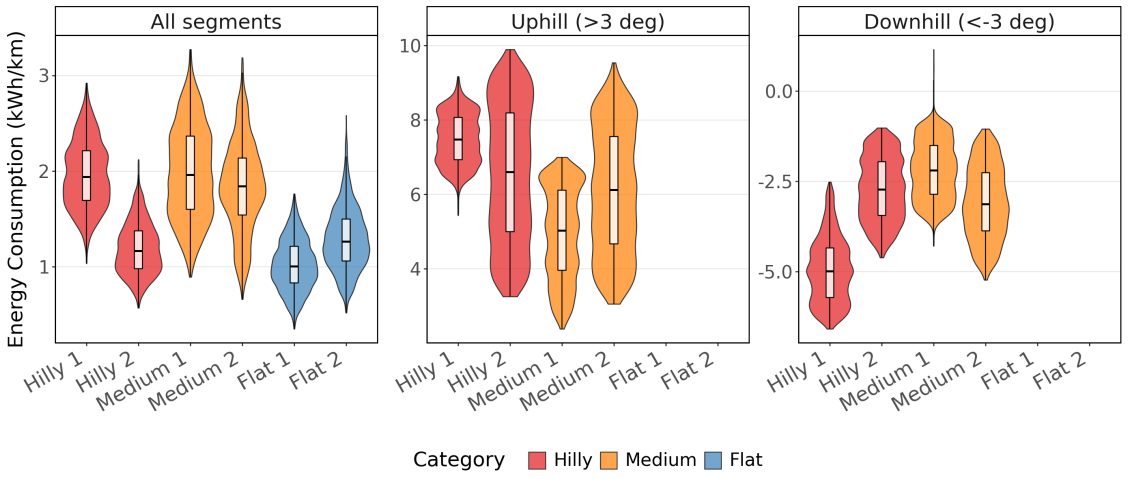


Figure 4.9: Violin plots with box plot of the density distribution of $\widetilde{E}_c$ for six drive cycles with three types: hilly, medium and flat. This is shown for all segments as well as for only uphill/downhill segments.

In general, it is evident from Figure 4.9 that the energy consumption is much higher in uphill segments, but it is not as clear that a hilly drive cycle results in an overall

higher energy consumption. Drive cycles are more complex and can not be easily categorized according to their road inclination distributions, instead accumulated small variations can lead to large changes in the total energy consumed. For example, *Hilly 2* have the second highest mean energy consumption for the uphill sections, however as mentioned have the second lowest mean energy consumption over all segments. In addition to this, there are large variation of $\widetilde{E}_c$ within the same drive cycle, further demonstrating their complexity.

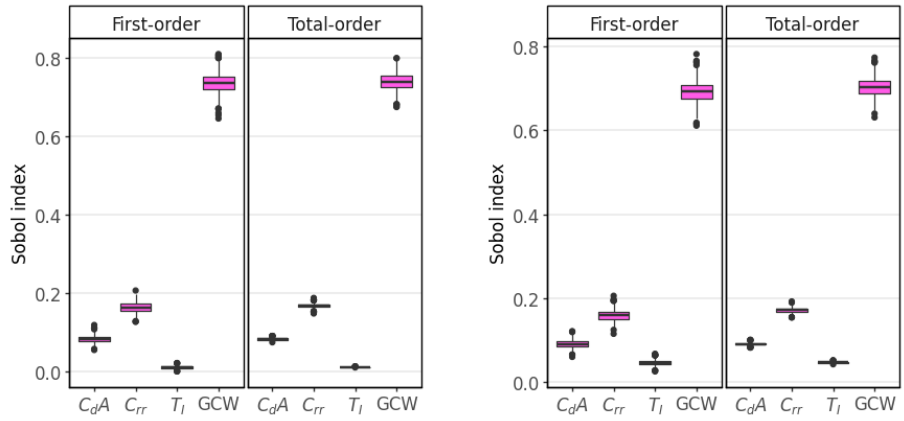
4.3 Variance Decomposition

The first order Sobol' index, S_i, and the total order Sobol' index, S_{T_i}, for the six drive cycles are listed in Table 4.2. They describe the expected reduction in output variance if x_i could be fixed and the expected variance left if all other factors but x_i were fixed, respectively. A boxplot using a 1000 bootstrapped samples for all drive cycle pairs, *Hilly*, *Medium*, and *Flat*, can be seen in Fig 4.10, 4.11, and 4.12 respectively. Table 4.3 shows the 95% confidence intervals for the bootstrap.

To verify that the Sobol' indices were computed with a sufficiently large number of simulations, the checks (2.19) were carried out on the results. Except for some minor decimal errors that could be due to numerical errors in the calculations, all the drive cycles got passing results. To further support this another analysis with Sobol's method using half the points, $N = 1024$, was carried out, giving similar results. This validates that $N = 2048$ is sufficiently large and the resulting indices can be trusted.

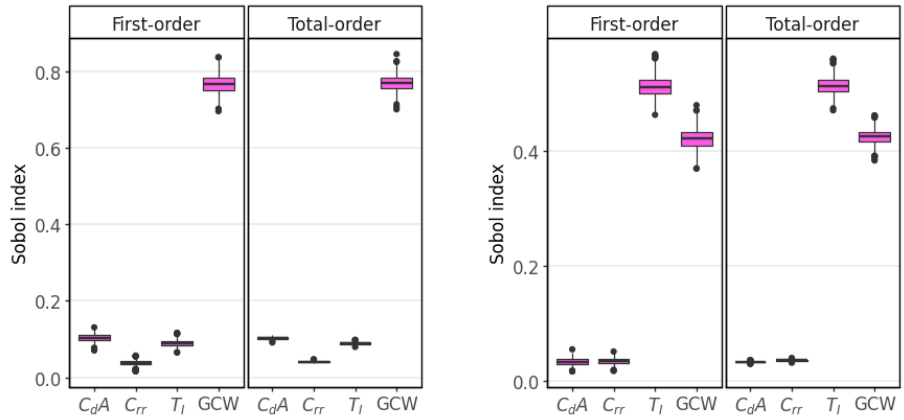
Table 4.2: First order Sobol' index (S_i) and total order Sobol' index (S_{T_i}) for each parameter across all routes.

Parameter	C_dA		C_{rr}		T_I		GCW	
Sobol' Index	S_i	S_{T_i}	S_i	S_{T_i}	S_i	S_{T_i}	S_i	S_{T_i}
Hilly 1	0.084	0.084	0.165	0.169	0.013	0.013	0.737	0.742
Hilly 2	0.091	0.091	0.159	0.171	0.046	0.047	0.692	0.703
Medium 1	0.104	0.104	0.038	0.042	0.090	0.090	0.765	0.768
Medium 2	0.032	0.032	0.034	0.035	0.511	0.512	0.420	0.424
Flat 1	0.266	0.265	0.065	0.077	0.041	0.041	0.617	0.630
Flat 2	0.193	0.193	0.077	0.082	0.480	0.480	0.245	0.250



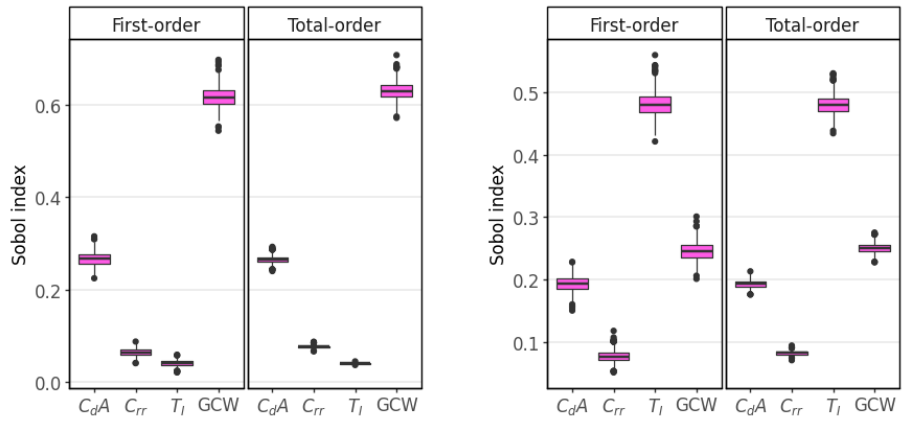
(a) *Hilly 1* **(b)** *Hilly 2*

Figure 4.10: Box plot of the bootstrapped S_i and S_{T_i} for *Hilly* drive cycles.



(a) *Medium 1* **(b)** *Medium 2*

Figure 4.11: Box plot of the bootstrapped S_i and S_{T_i} for *Medium* drive cycles.



(a) *Flat 1* **(b)** *Flat 2*

Figure 4.12: Box plot of the bootstrapped S_i and S_{T_i} for *Flat* drive cycles.

Table 4.3: Lower and upper 95% confidence interval bounds, calculated though bootstrapping with 1000 resamples, for the first order S_i and total order S_{T_i} Sobol' indices across the six drive cycles.

(a) C_dA

Parameter	C_dA			
Sobol' Index	S_i		S_{T_i}	
CI 95%	Low	High	Low	High
Hilly 1	0.066	0.099	0.079	0.089
Hilly 2	0.074	0.109	0.086	0.097
Medium 1	0.084	0.122	0.097	0.111
Medium 2	0.021	0.043	0.030	0.034
Flat 1	0.234	0.294	0.250	0.281
Flat 2	0.167	0.218	0.182	0.205

(b) C_{rr}

Parameter	C_{rr}			
Sobol' Index	S_i		S_{T_i}	
CI 95%	Low	High	Low	High
Hilly 1	0.141	0.188	0.158	0.179
Hilly 2	0.136	0.184	0.161	0.184
Medium 1	0.026	0.052	0.039	0.044
Medium 2	0.022	0.045	0.033	0.038
Flat 1	0.049	0.081	0.071	0.083
Flat 2	0.060	0.093	0.076	0.087

(c) T_I

Parameter	GCW			
Sobol' Index	S_i		S_{T_i}	
CI 95%	Low	High	Low	High
Hilly 1	0.006	0.020	0.012	0.014
Hilly 2	0.034	0.060	0.044	0.050
Medium 1	0.073	0.106	0.084	0.095
Medium 2	0.474	0.546	0.486	0.543
Flat 1	0.029	0.055	0.039	0.044
Flat 2	0.444	0.520	0.452	0.508

(d) GCW

Parameter	T_I			
Sobol' Index	S_i		S_{T_i}	
CI 95%	Low	High	Low	High
Hilly 1	0.694	0.783	0.705	0.785
Hilly 2	0.652	0.740	0.666	0.747
Medium 1	0.718	0.805	0.728	0.810
Medium 2	0.387	0.453	0.399	0.447
Flat 1	0.570	0.661	0.596	0.670
Flat 2	0.218	0.275	0.234	0.267

Across four of the six drive cycles, GCW exhibits the highest Sobol' indices, for both the first and total order. It has a maximum first order Sobol' index of 0.765 for *Medium 1*, which is the highest achieved for any parameter and drive cycle combination. The lowest first order index for GCW is observed for *Flat 2* where it is 0.245. In contrast, for the remaining two cycles, *Medium 2* and *Flat 2*, T_I attains the largest Sobol' indices and thus is the parameter responsible for the largest portion of variance. Nevertheless, in both *Medium 2* and *Flat 2*, GCW ranks second, and the disparity between the first and secondly ranked input parameters for both indices is smaller than it is for the other drive cycles. This further suggests that GCW should be considered as the parameter that causes the largest output variance for $\widetilde{E}_c$ overall.

For the remaining parameters, the values of the Sobol' indices across the drive cycles varies a lot. For some of the drive cycles C_{rr} has the second highest indices, whereas for others, C_dA ranks second. The third and forth ranked input parameter is a mixture of C_dA, C_{rr}, and T_I depending on the drive cycle. However, for some of the drive cycles the fourth ranked input parameter have both S_i and S_{T_i} almost equal to zero, meaning that they have negligible contributions towards the variance.

One additional observation is the fact that for the drive cycles with the fastest average velocity, *Medium 1*, *Flat 1* and *Flat 2*, the relative influence of C_dA is the highest. This aligns with the results from the force-based model which shows the importance of C_dA is proportional to the velocity. Another observation is that both Sobol' indices for C_{rr} ranks second for both *Hilly* drive cycles, see Figure 4.10. One explanation for this could be that these drive cycles have the lowest average velocity,

however this applies to both *Hilly 1* and *Medium 2*, but the relative importance of C_{rr} differ a lot between them.

The confidence intervals given in Table 4.3, shows that generally the values for S_{T_i} have less uncertainty than S_i. This means that it is difficult to isolate the main effect from the more noisy interaction effects. It is a common effect when using Monte Carlo simulations to measure variance, as the uncertainty depends on the estimator and sampling used for calculations [3][13].

4.3.1 Comparison of S_i and S_{T_i}

For all drive cycles and input parameters considered, S_i and S_{T_i} are very similar, as seen in Table 4.2. The difference between the two indices corresponds to the sum of the second, third, and forth order Sobol' indices, which represents interaction effects between the input parameters. Because this difference is small, the interaction effects can be considered negligible or nonexistent. This suggests that the inputs act approximately additively; the effect on the variance of changing one input does not strongly depend on the value of another. This does not imply that the model response is linear in any individual input, only that the inputs' contributions to the variance of $\widetilde{E}_c$ are largely separable.

The difference between S_i and S_{T_i} is slightly larger for GCW and C_{rr} than for C_dA and T_I, for which the difference is approximately zero. For each drive cycle, the difference between S_i and S_{T_i} is also approximately the same for GCW and C_{rr}. This indicates a small second order interaction shared by these two parameters, which is consistent with their coupling in the force equation (2.1). At the same time, the Sobol' analysis shows that almost all of the contribution from C_{rr} to output variance is captured by its first order index alone. The interaction effect is small because it is proportional to the variance of GCW, while the first order effect is proportional to the square of GCW's mean, which is an order of magnitude larger. This means that knowing the rolling resistance value, without knowing the specific GCW, is already sufficient to explain most of its influence on energy consumption.

Given the complexity of the model, the role of parameter interactions warrants consideration. However, the present analysis is limited to the parameters included in the Sobol' study and more complex interaction effects may emerge from other parameters. This is especially relevant since the inputs for which Sobol' indices were evaluated, with exception of T_I, are force-based and their effects are generally more predictable than factors defined in other complex subsystems of the vehicle.

4.3.2 Importance of Initial Temperature

The Sobol' indices for T_I were a bit confounding, as the initial temperature had a large effect on $\widetilde{E}_c$ for only two of the drive cycles: *Medium 2* and *Flat 2*. An

explanation for this behavior, specifically considering *Flat 2* could be that it is both the shortest drive cycle and the one with the highest average velocity. This means it is the drive cycle with the lowest completion time which might inflate importance of the parameters that set the initial conditions of the simulation, compared to those that apply throughout the entire simulation, such as GCW, C_dA, C_{rr}, FGR etc. Note, that this does not apply for *Medium 2*, which has an average length and velocity compared to the other drive cycles.

The high Sobol' indices might also be a consequence of the short drive cycles used. If the initial temperature is very low, a large amount of energy will be used for heating at the beginning of the drive cycle. If the drive cycle is short, this constitutes to a large fraction of the consumed energy, whereas for a longer drive cycle, other input parameters have more time to contribute, leading to a more balanced sensitivity among the parameters.

Another reason for this might be that the simulation spends time pre-heating the coolants before starting the drive. To investigate this the battery and coolant temperatures were plotted against the simulated time. Figure 4.13 shows these signals for a simulation with initial temperature of $-10°C$ together with the cumulative distance on a separate axis. After an investigation of the output signals, it was determined that once the vehicle has traveled a cumulative distance of 2 km no substantial pre-heating demand should remain. This point in time is included in the figure as a black line.

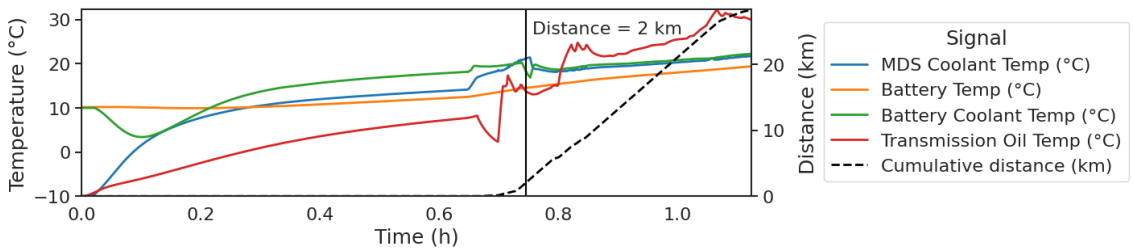


Figure 4.13: Different temperature signals over time for a simulation of *Flat 2* with $T_I = -10°C$ and the cumulative distance on a separate axis. The black line shows the time for which the vehicle has a cumulative distance over 2 km.

Because the simulation is initialized with such a low temperature, it spends over 40 minutes heating before it starts driving. This can be seen through the smooth lines in the figure, that then gets more jagged as the driving starts. The coolant temperatures starts rising whereas the battery temperature is pretty constant, up until the 40 minute mark. In figure 4.14 the same drive cycle was initialized at $T_I = 30°C$. In this case one can see that the time to reach 2 km is significantly less.

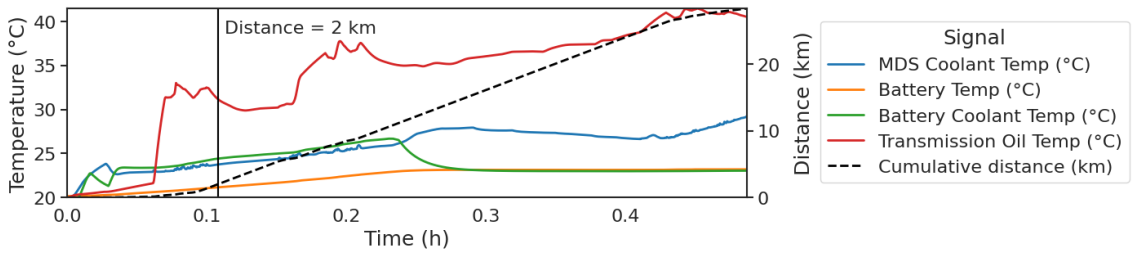
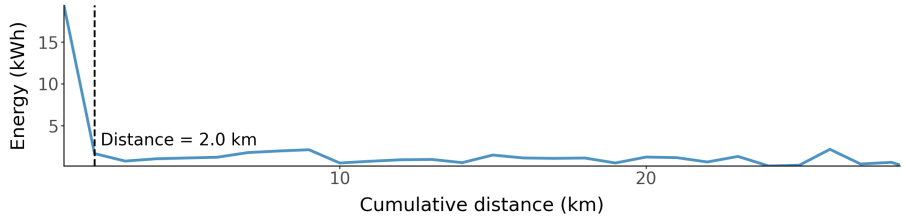
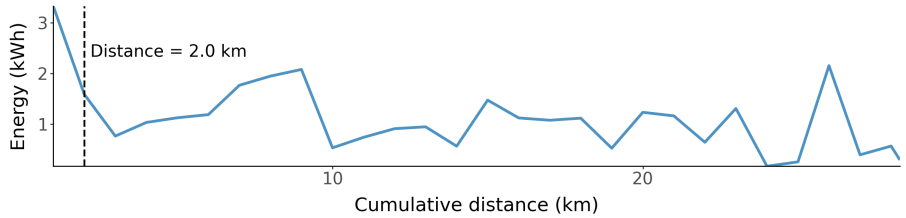


Figure 4.14: Different temperature signals over time for a simulation of *Flat 2* with $T_I = 20°C$ and the cumulative distance on a separate axis. The black line shows the time for which the vehicle has a cumulative distance over 2 km

Figure 4.15 shows the energy consumed as a function of cumulative distance for initial temperatures of -10°C and 20°C, aggregated in 1 km bins. The figure further illustrates that, at lower initial temperatures, a substantial share of the total energy consumption occurs at the beginning of the drive cycle. Although the simulation with an initial temperature of 20°C also exhibits its maximum energy consumption at the start of the drive cycle, the magnitude of this initial consumption is significantly lower.



(a) Initial temperature of -10°C.



(b) Initial temperature of 20°C.

Figure 4.15: Energy against the cumulative distance of drive cycle *Flat 2* binned per km, with the 2 km point marked as a dashed line.

Table 4.4 lists the percentage of total energy consumed before the threshold is reached across a range of different initial temperatures for all drive cycles. Notably, both *Medium 2* and *Flat 2* exhibit a substantial portion of its energy consumption prior to reaching 2 km when the initial temperature is $-10°C$. This might explains why T_I accounts for a large proportion of the variance in the Sobol' analysis, see Table 4.2, for these two drive cycles. Specifically, *Medium 2* consumes 20.5% of its total energy during the first 2 km, whereas *Flat 2* exhibits an even higher share of 43.4%. This percentage then decreases as the T_I increases. Similar trends are

observed across all drive cycles, with the lowest values generally occurring at initial temperatures of 20°C and 30°C, which indicates that these temperatures represent the optimal range for T_I.

Table 4.4: The fraction of energy consumed during the first 2 km of simulation for a range of initial temperatures. This is shown for all drive cycles.

Energy before 2 km (%)	−10°C	0°C	10°C	20°C	30°C	40°C
Hilly 1	16.9	12.5	13.7	10.2	10.6	13.7
Hilly 2	7.6	7.9	7.7	5.4	3.6	5.7
Medium 1	14.7	11.2	11.2	3.5	3.9	4.0
Medium 2	20.5	16.1	16.2	9.6	10.4	10.7
Flat 1	14.6	11.2	12.1	9.0	8.1	8.1
Flat 2	43.4	35.1	35.4	15.3	16.3	16.3

The energy consumed during the first 2 km of the simulation is disproportionately high relative to the distance traveled. Since 2 km corresponds to only 4 to 7 percent of the total drive cycle length, the substantially larger portion of energy consumption observed at the beginning of the simulations, particularly at lower temperatures, suggests that this effect can be attributed to pre-heating. Furthermore, this implies that $\widetilde{E}_c$ will become less sensitive to T_I as the drive cycle length increases, since the initial pre-heating effect will be progressively outweighed by parameters that influence energy consumption throughout the entire simulation.

4.3.3 Interaction Between GCW and C_{rr}

From a theoretical standpoint, rolling resistance depends directly on both mass and C_{rr}, see (2.1) and an interaction effect between these parameters would therefore be expected, leading to a higher S_{T_i} compared with S_i. However, this interaction is not reflected in the sensitivity indices, in contrast to the behavior observed with the force-based model. Although the underlying mechanism for this is understood, further investigation is justified.

Figure 4.16 shows the energy consumption over C_{rr} for different GCW calculated on five simulation of the *Flat 2* drive cycle. The slope, shown in the figure as β, increases as the mass increases. The slope values also follows the derivative from the force-based model that can be seen in Figure 4.1b. This eliminates the possibility that GSP does not follow the behavior presented by (2.1).

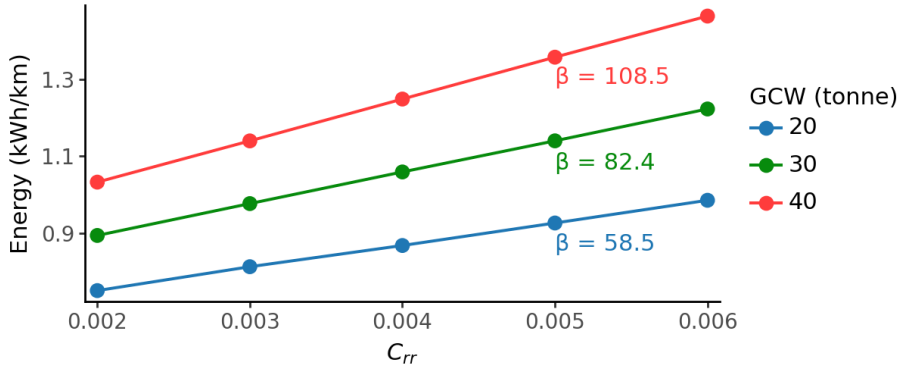


Figure 4.16: Energy consumption against the C_{rr} for different GCW, the slope of the lines are denoted as β. The drive cycle used is *Flat 2*.

One thing to keep in mind is that GCW impacts the energy consumption through multiple forces such as rolling resistance, aerodynamical drag and gravitational pull, whereas C_{rr} only is involved in the rolling resistance. The low impact the interaction has on the output variance suggests that rolling resistance is less influential than the other forces considered. One explanation is that the energy required to overcome aerodynamic drag exceeds that associated with rolling resistance, thereby contributing more significantly to the overall energy consumption and thus also its variation. The two types of energy consumptions can be seen in Figure 4.17 which shows that the energy losses due to aerodynamic drag is larger than those of rolling resistance. This is due to the high velocities that the vehicle is simulated with, which makes aerodynamic forces more significant than rolling resistance. For drive cycles with lower velocities, the results may differ.

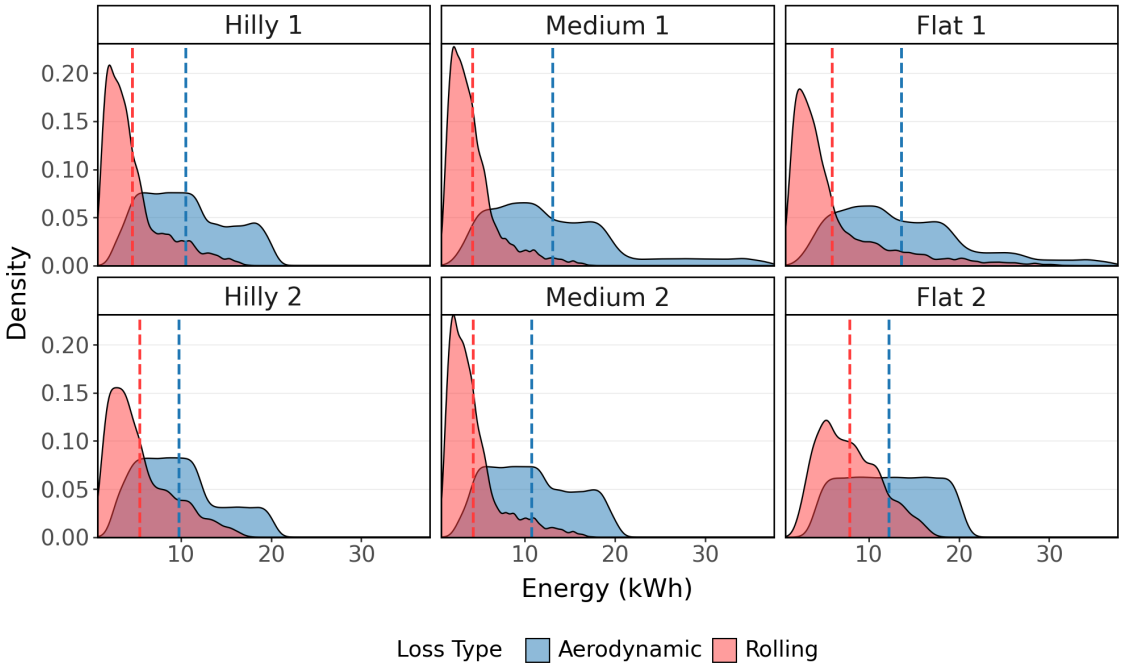


Figure 4.17: The distribution of losses from aerodynamic drag and rolling resistance for each drive cycle. Based on 30000 simulations

Considering both of these results, it can be assumed that the small difference between S_i and S_{T_i} for GCW and C_{rr} is accurate. This means that the coupling of these parameters leads to small variation in variance compared to their first order effects.

4.4 Linear Regression on Simulation Results

Figure 4.18 shows how $\widetilde{E}_c$ varies over the range of each input parameters for all EEM and Sobol' simulations. It shows the 95% and the 50% confidence intervals as well as a linear regression created with the least square method. The slopes of the regressions are shown in the figure as β and the change in output is displayed as ΔE. Note that since these are SA-simulations, all parameters are varied in each plot, accounting for the large spread in energy consumption. For each parameter 39000 EEM simulations were used and then for the four parameters considered for Sobol' indices, an additional 49152 Sobol' analysis simulations were plotted resulting in a total of 88152 simulations. This is why GCW, T_I, C_dA and C_{rr} have smoother confidence intervals than some other input parameters.

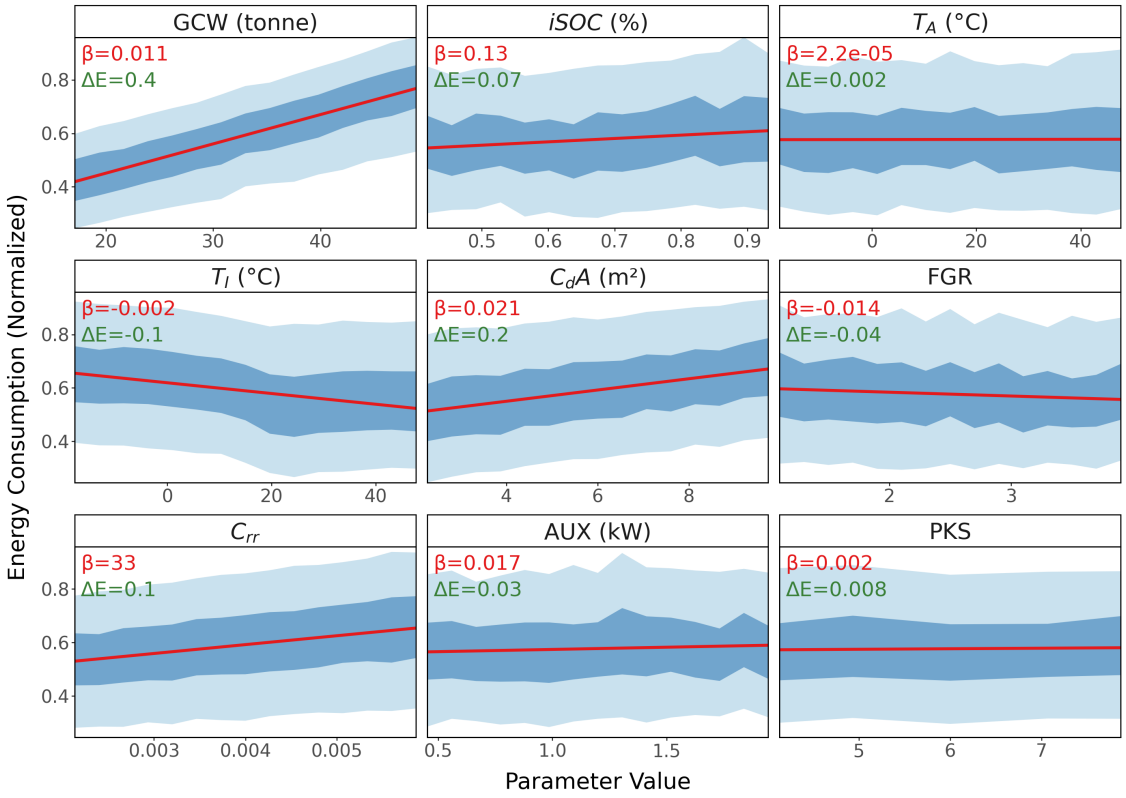


Figure 4.18: Normalized energy consumption for each input parameter with a fitted linear regression (red), 95% confidence interval (light blue) and 50% confidence interval (dark blue). The slope is denoted as β and the total change in energy as ΔE. Results include 39000 simulations spread over 6 drive cycles for $iSOC$, T_A, FGR, AUX and PKS, and 88152 simulations for GCW, T_I, C_dA and C_{rr}. Note, that all input parameters are varied in each subplot, resulting in the large spread.

When only looking at the effect shown as ΔE, we see that the GCW is the parameter with the largest change in $\widetilde{E}_c$ across it's input range. However it still has a small slope, with $k = 0.0103$ for one tonne, so while it is the parameter with potential of having the largest effect, it also requires large perturbations compared to other input parameters. The figure show that possible change over the input space are the largest for the four input parameters GCW, C_dA, C_{rr} and T_I, just as for the EEM.

Another thing shown in the figure, is that most of the input parameters seem to act linearly across the input space, with exception for T_I. Instead we see that T_I has a dip at roughly 25°C where $\widetilde{E}_c$ lowers in value, further strengthening the optimal range of temperature found for T_I in section 4.3.2. Nevertheless, since all other variables are varied equally throughout the plot, we can assume that this is either a case of nonlinearity or that there is only interaction at this part of the range for T_I. Either way it suggests something further than the linearity shown for other input parameters. If the ranges for the inputs such as $iSOC$, T_I and FGR were increased, they might show more nonlinearity since they seem to have some divergent behavior at their tails.

Table 4.5 shows the root means squared error (RMSE) and the mean absolute percentage error (MAPE) for the linear regression. Both errors are notably lower for GCW compared to other input parameters. Another thing to note in the table is that the MAPE errors are rather high, this is most likely because all parameters are varied, not just the one the regression is performed on. So the variation from the other input parameters show up as large confidence intervals and uncertainty.

Table 4.5: Errors for the linear regression.

Parameter	C_dA	C_{rr}	T_A	T_I	iSOC	AUX	FGR	GCW	PKS
RMSE	0.143	0.146	0.146	0.146	0.148	0.149	0.149	0.105	0.149
MAPE	0.224	0.230	0.228	0.228	0.235	0.237	0.236	0.163	0.237

The step size needed to make one percent change in $\widetilde{E}_c$, excluding nonlinear and interaction effects are listed in Table 4.6. The step size was calculated based on the linear regression. Because the input parameters have different scales the percentages are more representative of the parameter sensitivity. The model has negligible sensitivity to parameters T_A and PKS such that within the current limits of parameters the ΔE does not reach 1%. The parameters with higher μ^* and σ also have a lower percentage of change needed in order to achieve a noticeable change in $\widetilde{E}_c$. GCW has the lowest percentage, followed by C_dA, T_I and C_{rr}, whereas PKS, AUX and T_A have the highest.

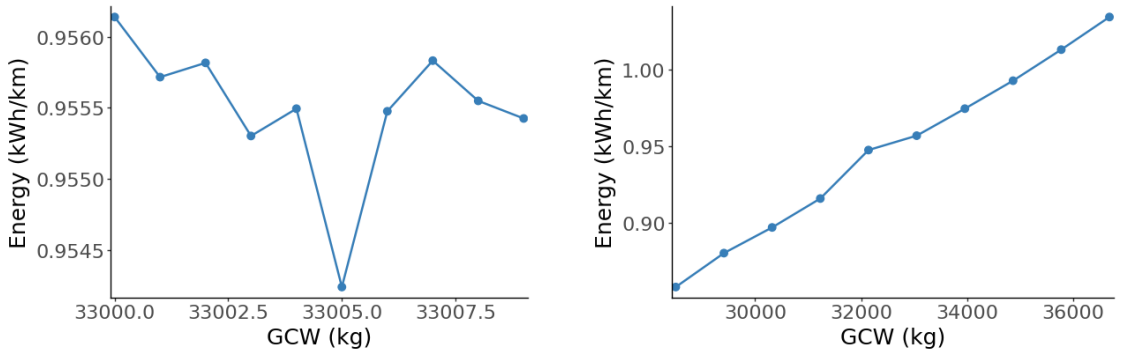
Table 4.6: Step size needed to take a one percent step increase in energy consumption for each input parameter and what percentage of the total range considered this represents. All based on the linear regression of 88152 simulation results.

Parameter	C_dA (m^2)	C_{rr}	T_A (°C)	T_I (°C)	*iSOC* (%)	*AUX* (kW)	*FGR*	*GCW* (tonne)	*PKS*
Step Size	0.48	0.0003	446	-5.0	8.0	0.6	-0.71	0.91	5.0
Percentage (%)	5.9	7.5	740	8.3	14	38	23	2.7	120

In many ways this linear regression confirms the behaviors identified with EEM and the Sobol' method. From the quantity $\frac{\mu^*}{\sigma}$ in Figure 4.6 we find that three input parameters are clearly linear, these are *GCW*, C_dA and C_{rr}, the rest have mixed behaviors. This is partially supported in the linear regression. From the Sobol' indices it can be concluded that the interactions and nonlinearity have low impact on the variance, this suggests mostly linear behavior for the sensitivities. Combining these results, one can conclude that these three input parameters can be considered to effect the energy consumption linearly.

4.4.1 Step Size Comparison

A further analysis of appropriate step length for *GCW* can be seen in Figure 4.19, where 10 simulations were run with fixed values for all other parameters but varied according to two different approaches for *GCW* on the *Flat 1* drive cycle. The initial approach involved varying the vehicle mass with a uniform step size of 1 kg. As shown in the figure, this step length did not produce any significant variation in the simulation output. GSP is a deterministic model so this behavior is likely caused by numerical precision limitations, leading to the resulting change in energy consumption to fall below the models effective resolution. Consequently, the output fluctuates around 0.955 for all simulations, rather than exhibiting the expected monotonic increase. For an ideal model, even a small increment would yield to a measurable increase of energy consumption, but this is not the case for GSP.



(a) Step length of 1kg, too small to estimate sensitivity.

(b) Step length of 910kg, suggested by the LR.

Figure 4.19: Energy consumption as the combined weight is varied according to different step lengths, one naive approach and the step length suggested by the linear regression. All simulations were run for *Flat 1*.

Figure 4.19b shows how the energy consumption varies when the step size suggested in Table 4.6 is used to vary GCW (910 kg). Then, a clear linear relationship is observed, where as the vehicle mass increases, $\widetilde{E}_c$ increases correspondingly. This behavior is more consistent with the physical expectations and provides more useful results. It also highlight how critical the choice of step size is when doing SA on GSP, as it can substantially influence the observed outcome.

4.5 Monte Carlo Filtering

The results from the MCF can be seen in Figure 4.20 and Figure 4.21. They show the behavioral and non behavioral distributions for the 10th percentile lowest and highest $\widetilde{E}_c$ respectively. The distributions were derived from 30000 EEM simulations, therefore the total distribution are not completely uniform since LHS was used to create them. If the two distributions differ it is implied that the parameter drives behavior.

From Figure 4.20, the MCF on low values for $\widetilde{E}_c$ can be seen. In agreement with earlier results, a low GCW, C_{rr} or C_dA, and high T_I drives low energy consumption. In particular there is a peak for the initial temperature at around 25°C which is the same value that yielded a dip in energy consumption in the linear regression. For the other input parameters there are no large differences so we can conclude that these are not factors for low energy consumption.

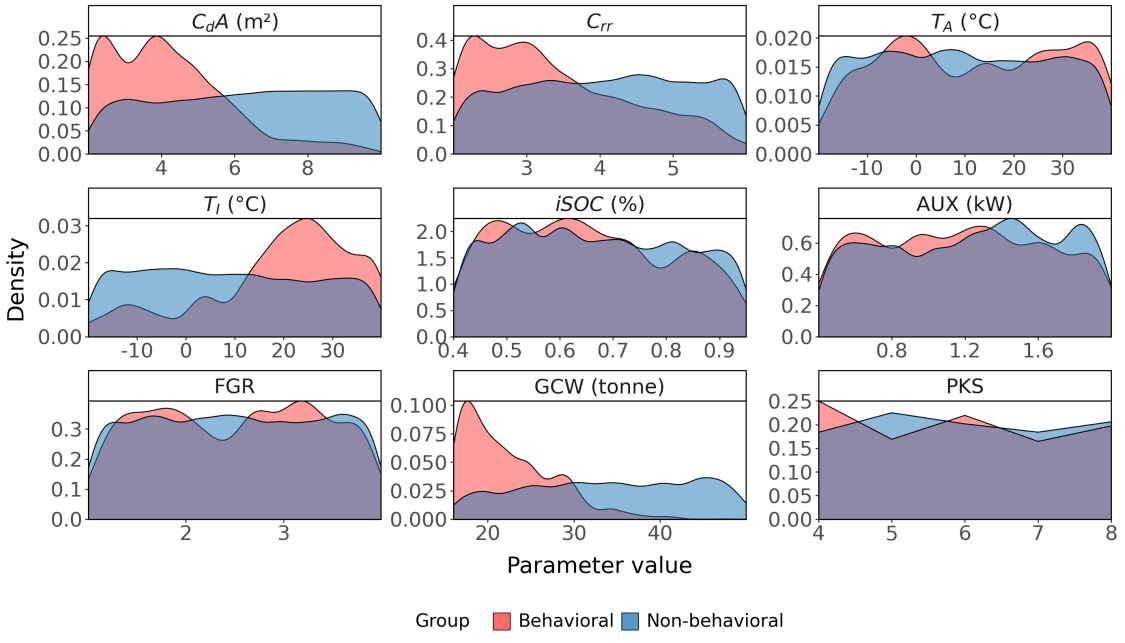


Figure 4.20: Factor mapping of the 10th percentile EEM simulations with the lowest $\widetilde{E}_c$ for each input parameter

In Figure 4.21 the behavioral and non behavioral distributions corresponding to the top 10% of outputs with high energy consumption can be seen. The observed relationships for the parameters are reversed compared to the case with low energy

consumption; a high GCW, C_{rr} or C_dA, and low T_I drives high $\widetilde{E}_c$. Furthermore, we can see some additional factors emerge. Specifically, a high $iSOC$ and low FGR are overrepresented in the behavioral distribution. This indicates that, while these parameters play a significant role in preventing high $\widetilde{E}_c$, their influence is less noticeable when considering the simulations with low energy consumption.

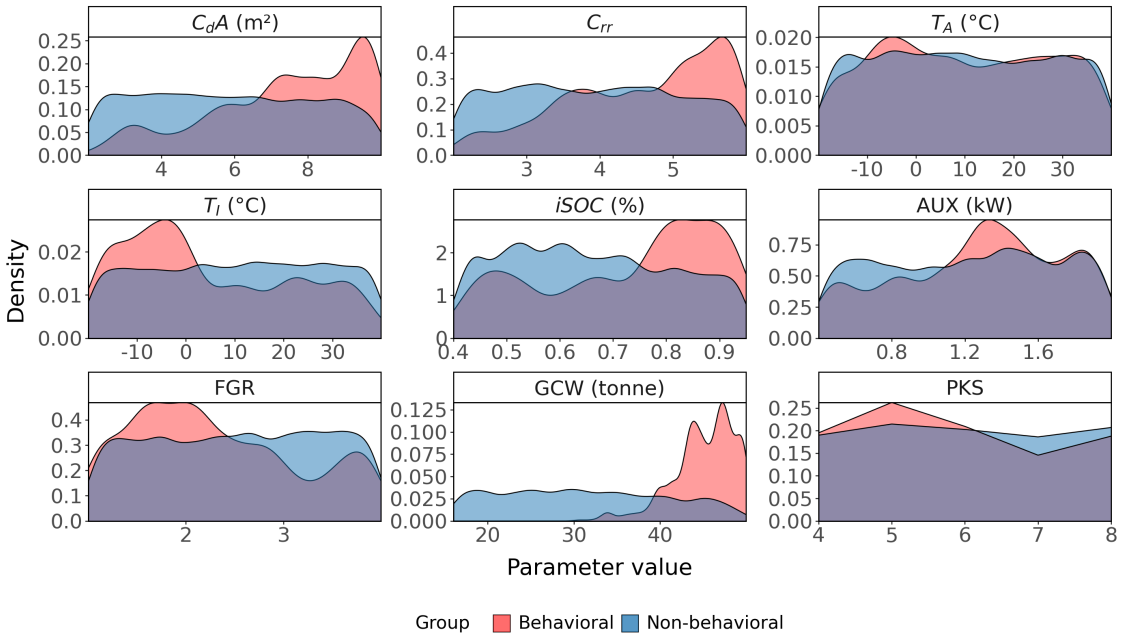


Figure 4.21: Factor mapping of the 10th percentile EEM simulations with the highest $\widetilde{E}_c$ for each input parameter

These results suggest that the importance of the input parameters depends on the selected output. In the previous sections, the SA considered only the overall influence on energy consumption, discounting whether the effect increased or decreased the output response. However, these results indicate that identifying the drivers of high energy consumption is not necessarily the same task as identifying the drivers of low energy consumption. Therefore, the purpose of the analysis must be clearly defined and carefully chosen in subsequent investigation, particularly in the context of MCF.

Chapter 5

Conclusion

This section will summarize the results to the most important conclusions as well as present some suggestions on how the work can be continued and which areas need more extensive research.

5.1 Energy Consumption

Most of the analysis was done using the normalized energy consumption, $\widetilde{E}_c$, as the output variable. The conclusions can be divided into two main themes: parameter importance and parameter behavior.

5.1.1 Parameter Importance

Both the EEM and Sobol' analysis showed that the results are largely drive cycle dependent, however conclusions could still be drawn. From the EEM it was found that the input parameters GCW, T_I, C_{rr} and C_dA had the largest sensitivities towards $\widetilde{E}_c$. In addition to this, the input parameters AUX, T_A and PKS were shown to have little to no impact on $\widetilde{E}_c$. This suggests that in future investigations of the energy consumption, these input parameters can be excluded from consideration. The $iSOC$ and FGR have a bit more mixed results, and should therefore not be completely excluded depending on circumstance, especially since they seem to be rather drive cycle dependent.

Through the Sobol' method, it was confirmed that GCW is the parameter that cause the largest variation in output for majority of the drive cycles. It's first order contribution ranged from 24.5% to 76.5% depending on drive cycle. It is not surprising that the combined weight is a large contributor to the energy consumption, since an increase in weight necessitates a larger energy to move the same distance through basic laws of physics.

The initial temperature was found to contribute to 51% and 48 % of variance through its first order Sobol' index for two of the drive cycles. However, this appears to be caused by excessive pre-heating necessary in the beginning of the drive cycles that is extra apparent for these two specific drive cycles. Therefore, the influence of T_I on

$\widetilde{E}_c$ should diminish if the drive-cycle length is increased, since the initial pre-heating would be progressively outweighed by parameters that influence energy consumption over the entire simulation.

Through the MCF and linear regression it was concluded that all input parameters, except for T_I and FGR affect $\widetilde{E}_c$ through positive sensitivities. Meaning, increasing their values also increase the output correspondingly. Most of these directions are natural, but in other applications this can be a productive way to investigate whether parameter values should be lowered or increased. It was also observed that the initial temperature had an optimal temperature of around 25°C.

5.1.2 Parameter Behavior

The ratio between μ^* and σ found through the EEM suggests that GCW, C_dA and C_{rr} primarily influence the energy consumption through linear contributions, meaning that the interactions and nonlinearities are negligible for these parameters. These observations were further supported by the linear regression, which showed only minor deviations from linear behavior for these variables. The remaining input parameters had mixed results in the ratio test for the EEM, making them hard to classify. In addition, T_I was found to have nonlinear behavior in the regression analysis, characterized by a local decrease in energy consumption for temperatures around 25°C.

Although the Sobol' indices do not directly quantify the extent to which nonlinearities influence the output, the similar values of S_i and S_{T_i} across all parameters and drive cycle combinations suggests that GCW, C_dA, C_{rr} and T_I all influence the output variance mainly through their first order effects with only limited contributions from higher order interactions. Combining these results, one can conclude that it is sufficient to consider GCW, C_dA and C_{rr} as predominately linear input parameters, whereas T_I requires more careful consideration due to its potential nonlinear and interaction effects.

5.2 Aggregated KPIs

The SA of the aggregated KPIs demonstrates that the gearbox-, ESS-, and motor-related KPIs mainly are sensitive towards GCW, whereas T_I predominantly affects auxiliary systems. This indicates that the input space can be reduced depending on the specific output of interest, enabling more efficient analysis. Furthermore, certain KPIs, for example those related to the gearbox, show consistently higher sensitivities for a large number of parameters. While this may stem from numerical artifacts in complex simulation models, it more likely reflects genuine sensitivities inherent to the modeled vehicle system. Overall, the EEM method proves to be an effective technique for identifying dominant parameters and providing a system level understanding of sensitivities in complex models.

5.3 Further Research

One of the main conclusions of the analysis in this thesis was that the results varied with the drive cycles used and the methods did not provide a clear way to handle these complexities. To investigate this further, it could be valuable to apply the GSA to simulations with drive cycles characterized by constant inclination and velocity in order to assess whether this affects the results and whether the resulting sensitivities become easier to interpret. Then, the inclination and velocity could also be varied or included as parameters in the EEM and the Sobol' method to investigate how the sensitivity depends on the road setup, similar to how they were studied in the force-based model. This would give further context to the results obtained in this study and thus might explain some of the discrepancies discussed.

An additional approach to reduce the influence of specific drive cycle characteristics and improve generalization is to simulate each cycle in both forward and reverse directions. By doing so, the net elevation change becomes zero, preventing any bias that could artificially inflate the importance of GCW. Furthermore, this ensures that uphill and downhill segments are equally represented in both directions, eliminating potential asymmetries associated with route profile. This could potentially yield more robust results without requiring modifications to the methods and existing framework.

Another direction further research could take is examining the sensitivity of the output as a time series instead of a single KPI. In this application we only considered one-dimensional outputs that quantify the vehicle performance in single number, but this is not the only way to handle time series data. One could also consider the output as a function dependent on time, $y(t)$. Then, the sensitivities also become time dependent and one can study the evolution of sensitivity throughout the drive cycle [26]. This makes it possible to analyse when each of the parameters are important and when they are not. For example, if a parameter dominates in the beginning of a drive cycle or only in steep uphill inclination etc. It becomes easier to explain the sensitivities and apply the results to the physical systems. This might be a good way to further clarify how the initial temperatures impacts energy consumption. This type of analysis could also be useful for charging or life time analysis for batteries of BEVs, as it then might reveal information about what parameters impact the vehicle's longevity in different stages of its life.

Bibliography

[1] Adobe Stock. https://stock.adobe.com. Accessed: 2026-05-08.

[2] Kersten Heineke, Philipp Kampshoff, and Timo Möller. A year of electric vehicle and mobility trends | mckinsey, 3 2024.

[3] Andrea Saltelli, Stefano Tarantola, Marco Ratto, Michaela Saisana, Fulvia Pennoni, Debora Gatelli, Jessica Cariboni, and Francesca Campolongo. *Global Sensitivity Analysis. The Primer*. John Wiley & Sons Inc, 11 2008.

[4] Volvo. Faq about electric trucks and sustainability. `https://www.volvotrucks.com/en-en/trucks/electric/FAQ/faq-sustainability-.html#accordion-3821ed63c2-item-9e9e939acc`. Accessed: 2026-05-18.

[5] Lucia Mancini, Nicolas A. Eslava, Marzia Traverso, and Fabrice Mathieux. Assessing impacts of responsible sourcing initiatives for cobalt: Insights from a case study. *Resources Policy*, 71:102015, 6 2021.

[6] Sai Krishna Vempalli, J. Ramprabhakar, S. Shankar, and G. Prabhakar. Electric vehicle designing, modelling and simulation. In *2018 4th International Conference for Convergence in Technology (I2CT)*, pages 1–6, 2018.

[7] Lino Guzzella and Antonio Sciarretta. *Vehicle Propulsion Systems Introduction to Modeling and Optimization*. Springer, third edition, 2012.

[8] Courant Richard and John Fritz. *Introduction to Calculus and Analysis I*. Springer Berlin, Heidelberg, 1999. Originally published in 1965 by Interscience.

[9] Atılım Güne¸s, Güne¸s Baydin, Barak A Pearlmutter, and Jeffrey Mark Siskind. Automatic differentiation in machine learning: a survey. *Journal of Machine Learning Research*, 18:1–43, 2018.

[10] Dan Gabriel Cacuci. *Motivation: Overcoming the Curse of Dimensionality in Sensitivity Analysis, Uncertainty Quantification, and Predictive Modeling*, pages 1–44. Springer International Publishing, Cham, 2022.

[11] Chenxiao Song and Reiichiro Kawai. Monte carlo and variance reduction methods for structural reliability analysis: A comprehensive review, 7 2023.

[12] Andrea Saltelli and Paola Annoni. How to avoid a perfunctory sensitivity analysis. *Environmental Modelling and Software*, 25:1508–1517, 12 2010.

[13] Ivano Azzini, Thierry A Mara, and Rossana Rosati. Monte carlo estimators of first-and total-orders sobol' indices. 2020.

[14] Andrea Saltelli. Making best use of model evaluations to compute sensitivity indices. *Computer Physics Communications*, 145:280–297, 5 2002.

[15] Andrea Saltelli, Paola Annoni, Ivano Azzini, Francesca Campolongo, Marco Ratto, and Stefano Tarantola. Variance based sensitivity analysis of model output. design and estimator for the total sensitivity index. *Computer Physics Communications*, 181:259–270, 2 2010.

[16] Robert Stine. An introduction to bootstrap methods. *Sociological Methods & Research*, 18:243–291, 1989.

[17] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer, 2009.

[18] Volvo Trucks. Image taken from front page. `https://www.volvotrucks.com/en-en/trucks.html`. Accessed: 2026-04-29.

[19] PyTorch. Automatic differentiation package - torch.autograd — pytorch 2.11 documentation. `https://docs.pytorch.org/docs/stable/autograd.html`. Accessed: 2026-04-29.

[20] Simulink. Simulation and model-based design. `https://se.mathworks.com/products/simulink.html`. Accessed: 2026-05-11.

[21] SciPy. Statistical functions (scipy.stats) — scipy v1.17.0 manual. `https://docs.scipy.org/doc/scipy/reference/stats.html`. Accessed: 2026-04-29.

[22] Francesca Pianosi, Fanny Sarrazin, and Thorsten Wagener. A matlab toolbox for global sensitivity analysis. *Environmental Modelling & Software*, 70:80–85, 8 2015.

[23] Polars. Lazyframe — polars documentation. `https://docs.pola.rs/api/python/stable/reference/lazyframe/index.html`. Accessed: 2026-04-29.

[24] I. M. Sobol'. On the distribution of points in a cube and the approximate evaluation of integrals. *USSR Computational Mathematics and Mathematical Physics*, 7:86–112, 1 1967.

[25] NumPy. numpy.polyfit — numpy v2.4 manual. `https://numpy.org/doc/stable/reference/generated/numpy.polyfit.html`. Accessed: 2026-04-29.

[26] Dhan Lord B. Fortela, Ashley P. Mikolajczyk, Rafael Hernandez, Emmanuel Revellame, Wayne Sharp, William Holmes, Daniel Gang, and Mark E. Zappi. Dynamic time warping as elementary effects metric for morris-based global sensitivity analysis of high-dimension dynamical models. *Mathematical and Computational Applications*, 29, 12 2024.

Appendix A

Figure A.1 shows the results from using AD to calculate partial derivatives for the force-based model applied to the six drive cycles.

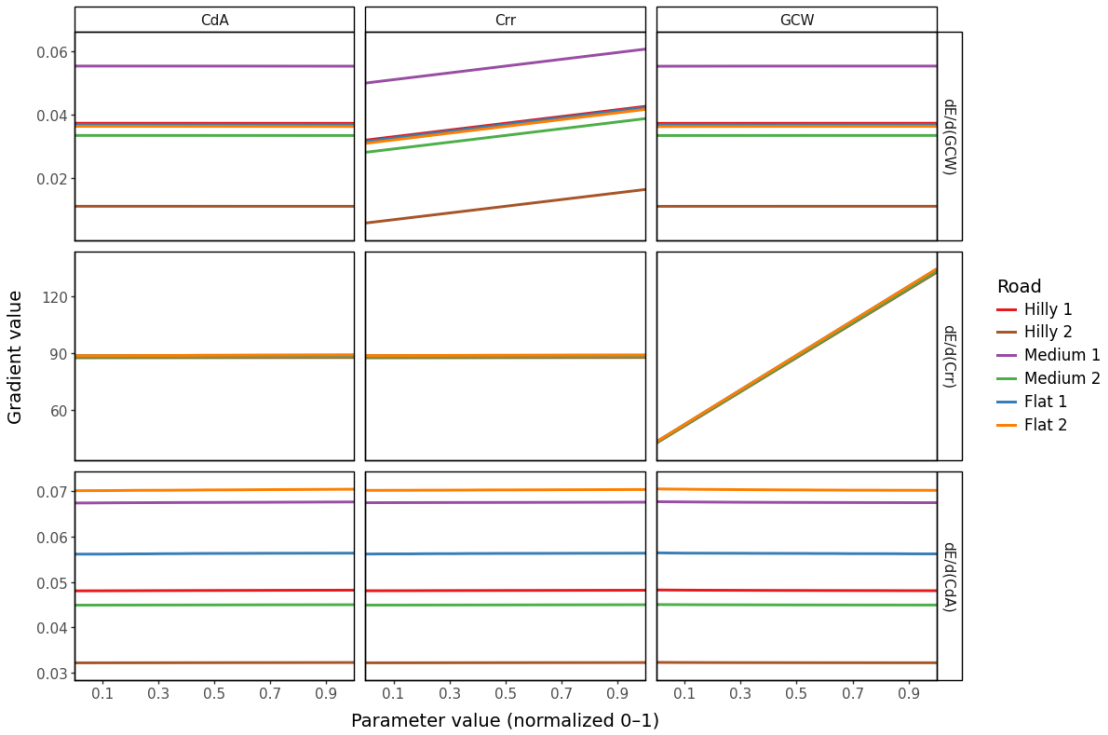


Figure A.1: Partial derivatives of energy consumption as a function of each swept parameter (columns), evaluated across six road profiles of varying hilliness. Non-constant gradients in off-diagonal subplots indicate cross-parameter interactions.

DEPARTMENT OF ELECTRICAL ENGINEERING
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden
www.chalmers.se

CHALMERS
UNIVERSITY OF TECHNOLOGY