Aalto University
School of Engineering

Anttila Joel

Uncertainty in electric bus driving cycles

Master's Thesis submitted in partial fulfilment of the requirements for the degree of Master of Science in Technology.

Espoo, Finland 12.12.2016
Supervisor: Kari Tammi, Professor
Advisor: Teemu Halmeaho, M.Sc. (Tech)

Aalto University
School of Engineering

Aalto University, P.O. BOX 11000, 00076 AALTO
www.aalto.fi

Abstract of master's thesis

Author Joel Anttila		
Title of thesis Uncertainty in electric bus driving cycles		
Degree programme Mechanical engineering		
Major/minor Engineering design		**Code** K3001
Thesis supervisor Kari Tammi, Professor		
Thesis advisor Teemu Halmeaho, M.Sc. (Tech)		
Date 12.12.2016	**Number of pages** 55+15	**Language** English

Abstract

Electric vehicles are in the brink of breakthrough in the automotive industry, and battery electric city buses are no exception. However, the costs are still high compared to conventional buses with internal combustion engines. A thorough electric city bus system planning is an effective way to curb the high costs, where, accurate knowledge about the specific energy consumption of the bus route is required.

This study utilized route energy consumption sensitivity analysis to create a reliable energy consumption forecast for a light traffic semi-urban bus route. The analysis was performed with the Monte Carlo method where the number of stops was addressed as the uncertainty. The uncertainty was modeled with a normal distributed probability density function.

The process of creating and utilizing the energy consumption sensitivity analysis is also presented. First, ten base driving cycles were measured in actual buses operating an actual bus line. Artificial driving cycles were synthesized from the measured cycles, based on the Monte Carlo sampling method. The fluctuating accelerations and speeds of different runs on the route are addressed by compiling the synthesized cycles from parts of the measured cycles. The synthesized driving cycles were simulated with a validated energy flow model of a battery electric bus. As a result, an energy consumption distribution for the route was acquired.

The energy consumption distribution includes all the outcomes obtained with the Monte Carlo method. The distribution also functions as an energy consumption forecast for the route. It was found out that the distance specific energy consumptions in the route fell between 0.483 kWh/km and 0.722 kWh/km. Which means, there is a 49.4 % variation between the extremes, when the number of bus stops varies. However, a closer result analysis proved that there is an 80 percent chance that the predicted consumption falls between 0.561 kWh/km and 0.642 kWh/km. Thus, it is denoted only 14.4 % variation is most likely present between the extremes. Additionally, the study suggested that lower accelerations result in lower consumptions. Driving cycles with less than average accelerations resulted in 5.6 % lower consumptions compared to cycles with higher than average accelerations.

Thus, it can be concluded that the number of bus stops greatly affected the energy consumption. In addition, Monte Carlo method proved to be worthwhile tool to clarify the effects of uncertainty.

Keywords Monte Carlo method, sensitivity analysis, battery electric city bus, energy flow model, simulation model, driving cycle, electrical consumption

Aalto-yliopisto
Insinööritieteiden korkeakoulu

Aalto-yliopisto, PL 11000, 00076 AALTO
www.aalto.fi

Diplomityön tiivistelmä

Tekijä Joel Anttila		
Työn nimi Epävarmuus sähköbussien ajosykleissä		
Koulutusohjelma Konetekniikka		
Pää-/sivuaine Koneensuunnittelu		**Koodi** K3001
Työn valvoja Professori Kari Tammi		
Työn ohjaaja(t) DI Teemu Halmeaho		
Päivämäärä 28.11.2016	**Sivumäärä** 55+15	**Kieli** Englanti

Tiivistelmä

Sähköajoneuvot ovat lyömässä läpi ajoneuvojen piirissä, eivätkä sähköbussit ole poikkeus. Siitä huolimatta, akkusähköbussien kustannukset ovat yhä korkeat perinteisiin polttomoottoroituihin busseihin verrattuna. Sähköbussisysteemien huolellinen suunnittelu on yksi tapa, jolla korkeita kustannuksia saadaan vähennettyä. Onnistunut suunnittelu tosin edellyttää, että bussireitin energiankulutus tunnetaan tarkoin.

Tässä tutkimuksessa suoritettiin herkkyysanalyysi bussireitin energiankulutuksesta, minkä avulla bussin kulutusta reitillä voitiin tarkoin kuvata ja ennustaa. Analyysi toteutettiin Monte Carlo menetelmän avulla, missä epävarmuutena käsiteltiin bussipysäkkien määrää. Epävarmuutta mallinnettiin normaalijakautuneella tiheysfunktiolla.

Tässä työssä esitellään prosessit herkkyysanalyysin tekemiseen. Ensiksi bussireitti mitattiin kymmenen kertaa. Mittausdatan pohjilta syntetisoitiin tuhansia ajosyklejä Monte Carlo samplausmenetelmän avulla. Syntetisoidut ajosyklit tehtiin yhdistelemällä osia mitatuista sykleistä, jolloin niissä huomioitiin ajosyklien vaihtelevat kiihtyvyydet ja nopeudet. Lopuksi ajosyklit simuloitiin validoidulla sähköbussin energiavirtamallilla. Tuloksena saatiin energiankulutusjakauma reitille.

Energiankulutusjakauma käsittää kaikki Monte Carlo menetelmällä saadut energiankulutuksen tulokset, mistä voidaan johtopäätöksenä tehdä energiankulutusennuste bussireitille. Saatiin selville, että kilometrikohtaiset energiankulutukset asettuvat 0.483 kWh/km ja 0.722 kWh/km välille. Tämä tarkoittaa, että kulutus voi vaihdella jopa 49.4 % ääripäiden välillä, kun pysähdysten määrä reitillä muuttuu. Siitä huolimatta, tulosanalyysi osoitti, että 80 % todennäköisyydellä kulutus pysyy arvojen 0.561 kWh/km ja 0.642 kWh/km välillä. Eli todennäköisimmin vaihtelua ääriarvojen välillä olisi vain 14.4 %.

Energiankulutusjakauman lisäksi tutkimus osoitti, että pienemmät kiihtyvyydet johtivat keskimäärin pienempiin kulutuksiin. Ajosykleillä, joissa positiiviset kiihtyvyydet olivat keskimääräistä pienempiä, saadut energiankulutukset olivat keskimäärin 5.6 % pienempiä kuin ajosykleillä, joissa kiihtyvyydet olivat keskimääräistä suurempia.

Lopuksi voidaan siis todeta, että bussipysäkkien määrä vaikuttaa suuresti energiankulutukseen. Monte Carlo menetelmä on hyödyllinen työkalu kuvaamaan epävarmuutta ja sen vaikutuksia.

Avainsanat Monte Carlo menetelmä, herkkyysanalyysi, akkusähköbussi, energiavirtamalli, simulaatiomalli, ajosykli, energiankulutus

Foreword

This thesis was written as a completion to the Master's Programme in Mechanical Engineering of Aalto University.

First, I would like to express my gratitude towards the thesis supervisor Professor Kari Tammi and thesis advisor M.Sc. (Tech) Teemu Halmeaho for their insightful guidance and encouraging support throughout the process of creating this dissertation. Additionally, I am much obliged for the "Henry Ford Säätiö" foundation for the vital financial aid which allowed me to fully concentrate on making the thesis. My appreciations also belong to Post-Doctoral Fellow (Tech) Antti Lajunen for sharing his simulation model, and his help in using the simulation tool.

Lastly, I would like to thank my friends and family for providing me with a firm base to work on. Especially my sweet girlfriend Jenny who has supplied me with endless love and care. With Your support, I feel I have the strength to take on anything.

Espoo 12.12.2016

_____

Joel Anttila

Table of contents

Symbols

A		Property A
A_{model}	[m^2]	Vehicle model frontal area
C_D		Vehicle model drag coefficient
F_D	[N]	Vehicle model aerodynamic drag force
$K(\cdot)$		Kernel smoothing function
N_{cycles}		Number of different synthesized cycles
N_{MC}		Number of points generated in the configuration space
S		Number of base cycles
T	[Nm]	Torque
T_{wheel}	[Nm]	Wheel torque
T_{in}	[Nm]	Torque transmitted from a final drive
T_{brake}	[Nm]	Brake torque
T_{roll}	[Nm]	Torque loss caused by rolling resistance
U	[N]	Potential energy
a_{decel}	[m/s^2]	Constant deceleration to a bus stop
C	[s]	Cutoff start time
d	[s]	Cutoff end time
$\widehat{f_h}(x)$		Kernel estimation function
h		Kernel estimation bandwidth
n_{stops}		Number of possible stops in the route
n		Number of bus stops
n_{kernel}		Kernel estimation sample size
r^N		Configuration of an N particle system
s_{comp}	[m]	Compensated distance
t_A	[s]	Time required to stop the vehicle
t_a	[s]	Time that has to be traveled at constant speed at point d
v	[m/s]	Velocity
$v(d)$	[m/s]	Velocity at point d
$v(i)$	[m/s]	Velocity at time i
v_{model}	[m/s]	Vehicle model velocity
v_0	[m/s]	Velocity at the end of stopping to a bus stop
v_1	[m/s]	Velocity at a bus stop in a driving cycle that passes the stop
β	[K^{-1}]	Reciprocal temperature
Δt	[s]	Time between samples in the driving cycles
ρ	[kg/m^3]	Air density
τ	[Nm]	Torque coupling torque
ω	[rad/s]	Angular velocity in torque coupling

Abbreviations and acronyms

AC	Alternating current
BEV	Battery electric vehicle
DC	Direct current
EV	Electric vehicle
GUI	Graphical user interface
ICE	Internal combustion engine
PDF	Probability density function
PI	Proportional-integral
SOC	State of charge
UNESC	United Nations Economic and Social Council
VPA	Vehicle propulsion architecture
VPC	Vehicle propulsion controller

1 Introduction

The energy crises of the last few decades and the risen environmental consciousness have fueled the interest in conserving energy and environment in both public and administrative sectors. This has led to uniform campaign to reduce the emissions and oil consumption. Thus, ever more efficient and environment-friendly technologies are being developed and applied. The same trend also goes for the automotive industry as road transport forms a great deal of the oil consumption. With continuously growing road traffic, the automotive industry is facing a tough challenge on how to reduce the environmental impact of motoring.

The use of fossil fuels in urban transport greatly contributes to both the local urban air pollution and global carbon dioxide levels. Thus, turning to CO2 neutral electric energy sources accompanied with electric vehicles (EVs) is an ideal solution as electric drive motors operate with better energy efficiencies compared to conventional internal combustion engines (ICEs). Moreover, EVs essentially emit no direct emissions and are not affected by the volatility of the fuel prices. Several times throughout the automotive history, electricity has tried to become the ruling vehicle energy medium. However, insufficient and expensive energy storing methods have prevented a true electric revolution.

For the first time in the automotive history, battery electric vehicles (BEVs) can match their ICE counterparts in public city transport operations [1]. Regardless of the technical advancements in electric propulsion and energy storing, diesel vehicles still dominate the public city bus traffic as diesel engines offer a low total cost of ownership yet to be unmatched by other means of propulsion [2]. In order to increase the attractiveness and competitiveness of battery electric city buses, the total costs must be further lowered. One key to lower costs is careful planning of electric bus transport route and schedule, where reliable and accurate simulation tools are required [3]. The aim of this dissertation is to examine the effects of bus route sensitivity to further advance the simulation tools used to optimize electric bus powertrains accordingly to bus routes.

1.1 Background

Computational simulations have become an important part of new vehicle development as actual prototyping and testing different vehicle designs is expensive and time consuming. Simulation models allow accurate analyses with increased flexibility and short time span. Consequently, it is evident that the simulation tools are also crucial in the optimization of electric city buses. As a result, a variety of simulation software have arose, of which, especially the Matlab-based tools are numerous; ADVISOR [4], CarSim [5], CSM HEV [6], HVEC [7], Janus [8], MARVEL [9], PSAT later known as Autonomie [10], PSIM [11], SIMPLEV [12], V-ELPH [13] and Virtual Test Bed [14].

Likewise, numerous are the simulation models created for electric city buses [2], [15]–[19] and other EVs [20]–[29]. However, the models used in the prior studies are based on one or more individual reference driving cycles derived from actual measured data or standardized transient driving schedules that simulate real cycles. It is proven [30] that driving cycle features, such as the number and magnitude of accelerations and decelerations and the average speed, have a definitive effect in the total energy efficiency. Naturally, the driving cycles of

bus routes can have significant variations, even between the same routes with different runs. Consequently, these variations result in uncertainty in the total energy consumption between computational simulation models and real life operations.

The uncertainty may lead to practical problems in the design of electric city bus systems as accurate information is vital in the parametric optimization of the buses. The models based on single cycles represent only the reference driving cycles and do not take in account the variable nature of actual bus routes. The unidentified uncertainty leads to potential oversizing in the city bus system design as sufficient performance must be ensured; the result is surplus costs. Thus, there is still room for improvement in the tools that are used in the research.

1.2 Purpose and scope

This thesis aims to create a novel energy consumption sensitivity analysis with the number of utilized bus stops in the route scrutinized as uncertainty. The sensitivity analysis clarifies the effects of uncertainties entailed in electric city bus routes, supporting the development of more efficient electric bus designs and operations. The result is a probability density function (PDF) of the electrical consumption with confidence intervals which forecasts the energy consumption with statistical justification.

However, certain limitations are implied in this study. First of all, the uncertainty of the number of bus stops is the only statistically analyzed factor. Other stops, such as stops due to traffic lights, are disregarded in the sensitivity analysis. Although, the effects of route features, such as pedestrian crossings and curves, are included in the measured base cycles of the route. Also, variable speed profiles of different runs on the route are modeled by combining the base cycles.

Furthermore, only one type of electric city bus configuration and Espoo bus line 11 are analyzed in this study. Other vehicle configurations and driving cycles are neglected. Thus, the driving cycle sensitivity analysis must be done individually for different routes in further studies. However, the sensitivity analysis indicates the scale of effects inflicted by route uncertainty in energy efficiency, which can give direction for other studies.

Moreover, only electrical consumption sensitivity analysis in conducted in this study. Thus, any possible applications and utilizations of the results are left for further studies. Additionally, environmental and socioeconomic analyses are also out of the scope of this study as only the energy consumption is considered.

Lastly, it must be acknowledged that some degree of imprecision is always involved in the simulations. In order to keep the simulation time sufficiently short, energy consumption simulations with extended time scale typically utilize less detailed models compared to transient dynamic models. Thus, certain threshold must be set for the inaccuracy. Additionally, only longitudinal vehicle dynamics are considered, hence, lateral effects and road elevations are excluded from this study.

1.3 Research design

The thesis consists of four main parts; literature review, the conduction of the sensitivity analysis, results and discussion. The second chapter of this study is composed of city bus classification and theoretical reviews of the electric bus drivetrain configurations, simulation models and Monte Carlo method. The sensitivity analysis along with the energy flow model and the model validation are presented in the third chapter. The fourth chapter reviews the results acquired with the sensitivity analysis. Key findings are recapitulated and evaluated in the final chapter, additionally, future prospects are also presented in the same chapter.

The energy flow model is assembled in the vehicle system simulation tool, Autonomie [10], and run in Matlab [31]. The validity of the simulation model is confirmed by comparing simulation results to actual measurements. If an acceptable precision is achieved with the simulation model, the model is approved and put to use. Concurrently, an algorithm that creates alternative driving cycles, based on the uncertainty of the number of bus stops, is presented.

The main results, electrical consumptions, are presented as a distribution. Confidence intervals are included in the distribution mark boundaries for the most probable outcomes. Additionally, the effect of the number of the stops to the energy consumption is reviewed along with the acceleration magnitude study.

2 Battery electric city buses and simulation models

In this chapter, the United Nations Economic and Social Council (UNESC) sanctioned city bus classification is presented. In addition, general EV drivetrains and computer vehicle simulation model types are reviewed and theoretical basis for Monte Carlo method presented.

2.1 City bus classifications

The UNESC [32] describes three main categories for power-driven vehicles solely utilized in passenger carriage. The three main category *M* vehicles are:

- M_1: Passenger carriage vehicles comprising maximum of 8 seats in addition to the driver's seat.
- M_2: Passenger carriage vehicles comprising more than 8 seats, in addition to the driver's seat, and maximum weight falls below 5 metric tons.
- M_3: Passenger carriage vehicles comprising more than 8 seats, in addition to the driver's seat, and maximum weight exceeds 5 metric tons.

In addition to the main categories, the passenger vehicles are further divided into subclasses depending on their structure and use. Category *M* vehicles with over 22 passengers, in addition to the driver, are separated into three classes:

- *Class I*: Vehicles that include seats and standing places for passengers, designed for frequent passenger movement.
- *Class II*: Vehicles that include seats and standing places in the gangway between the two double seats, designed primarily to transport seated passengers.
- *Class III*: Vehicles that include only seated places, designed solely to transport seated passengers.

While category *M* vehicles with a maximum of 22 passengers are separated into two categories:

- *Class A*: Vehicles that include seats and standing places for passengers, designed to carry standing passengers and allow frequent passenger movement.
- *Class B*: Vehicles that include only seated places, designed solely to transport seated passengers.

The most common type of vehicle for urban bus transport operations are low-floor M_3 *Class I* buses which are the main subject considered in this study.

2.2 Electric drivetrain configuration

In this section, a general EV drivetrain configuration layout, and its different, variations are presented. The functions and purpose of each subsystem and component of the drivetrain are reviewed. Also the features and benefits of different configuration variations are reviewed. The configurations presented in this section can also be applied in the battery electric city buses.

2.2.1 General electric drivetrain configuration

The general EV drivetrain configuration, illustrated in Figure 1, is divided into three main subsystems, which are the energy source, electric propulsion and auxiliary subsystems. The energy source subsystem is comprised of the energy source, energy refueling unit and energy management unit. The electric propulsion subsystem includes a vehicle controller, electronic power converter, electronic motor, mechanical transmission and wheels. The auxiliary subsystem contains an auxiliary power supply, cabin climate control unit and power steering unit. [33]

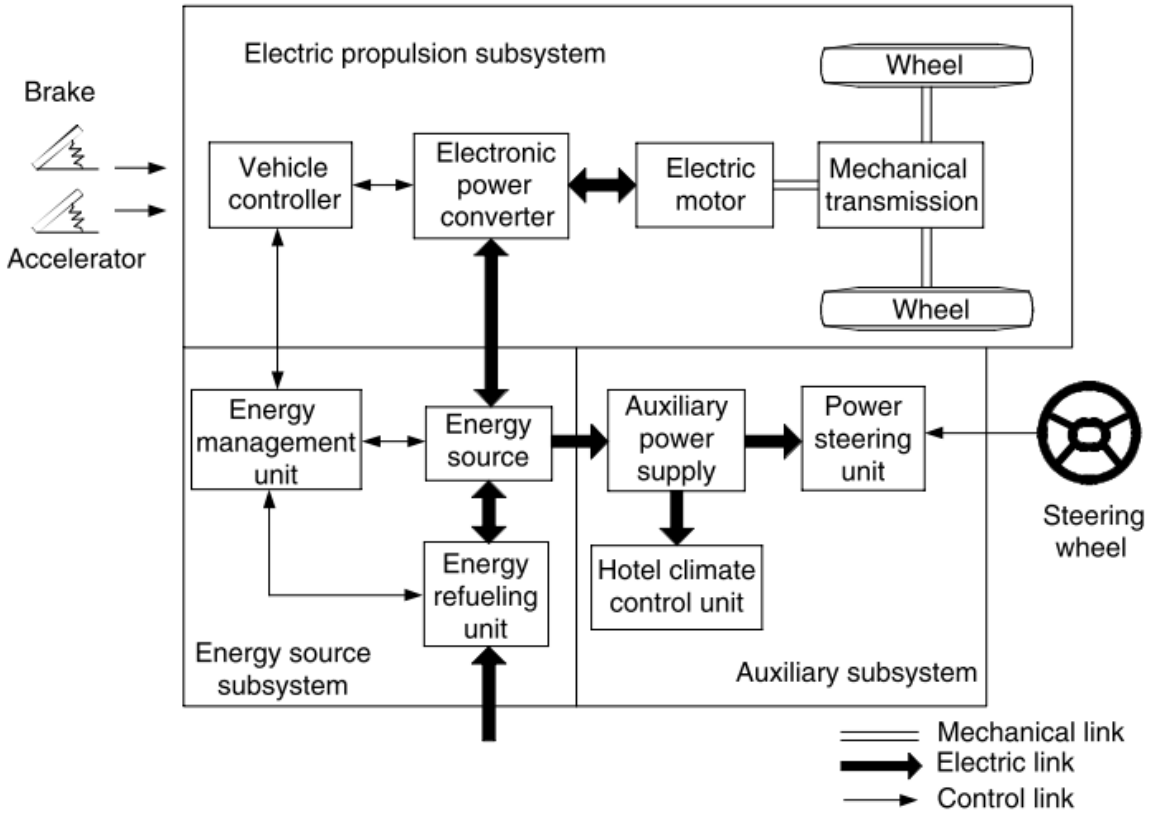

Figure 1. A general EV drivetrain configuration. [33]

The energy source subsystem provides the vehicle with the energy for the different operations. Typically, three modules form the energy source subsystem: the energy management unit, energy refueling unit and energy source. The energy management unit controls the energy flows from and to the energy source, while external energy can be introduced to the

system via the refueling unit. The energy source of a full-electric vehicle can be comprised of chemical batteries, fuel cells or ultracapacitors, or a combination of these components. [33]

Currently, lithium-ion batteries appear to be the most attractive energy storage solution for full electric city buses as fuel cells are too costly [18], [19], while ultracapacitors [17] can only be utilized as a supplement energy source in dual or multi energy source storage systems. Lithium-ion technologies already dominate the automotive industry, and the trend is growing. Moreover, the rapid progression of lithium-ion battery technologies and the explicit potential for further improvement bolster the use of the lithium-ion technology even more, making the technology irreplaceable in batteries for some time in the future. [34]

In the electric propulsion subsystem, the electric energy stored in the energy source is turned into thrust that propels the vehicle. The electronic power converter adjusts the voltage and current between the electric motor and energy source according to the vehicle controller, which is controlled by the pedals in human operated vehicles. Due to the fact that battery electric energy sources feed direct current (DC), the alternating current (AC) drive motors require a variable frequency and variable voltage DC-AC inverter. Both directions of power flow are possible with modern energy sources, such as ultracapacitors and lithium-ion batteries. These technologies can utilize the kinetic energy of the vehicle by regenerating energy back to the energy source during decelerations. [33]

The electric motor provides the mechanical torque for the transmission which delivers the torque to the wheels. Despite the maturity and simple control of DC motor technologies, DC motors are no longer used vehicle propulsion. Instead, frequency converter driven synchronous motors (mostly permanent magnet motors) and asynchronous induction motors prevail in the EV propulsion [35]. Modern commutatorless induction motors offer increased efficiency, power density and lower operational costs compared to DC motors. In addition, the induction motors are more reliable and maintenance-free. [33]

The auxiliary subsystem includes devices that are not directly part of the propulsion of the vehicle but are still necessary for vehicle operation. The necessary auxiliary devices include lights, horn and heating, while power steering, air conditioning, sound system and satellite navigation can be regarded as luxury yet important. Buses have additional auxiliary devices compared to passenger vehicles, such as door control and kneeling systems. In electric city buses, the auxiliary devices can form up to 20% of total energy consumption [2].

2.2.2 Configuration variations

A number of EV drivetrain configurations are presented in Figure 2. The configurations apply to rear wheel drive and all-wheel drive layouts, even though, the configurations in the figure are only for front wheel driven drivetrains. Electric city buses are typically rear wheel driven [35].

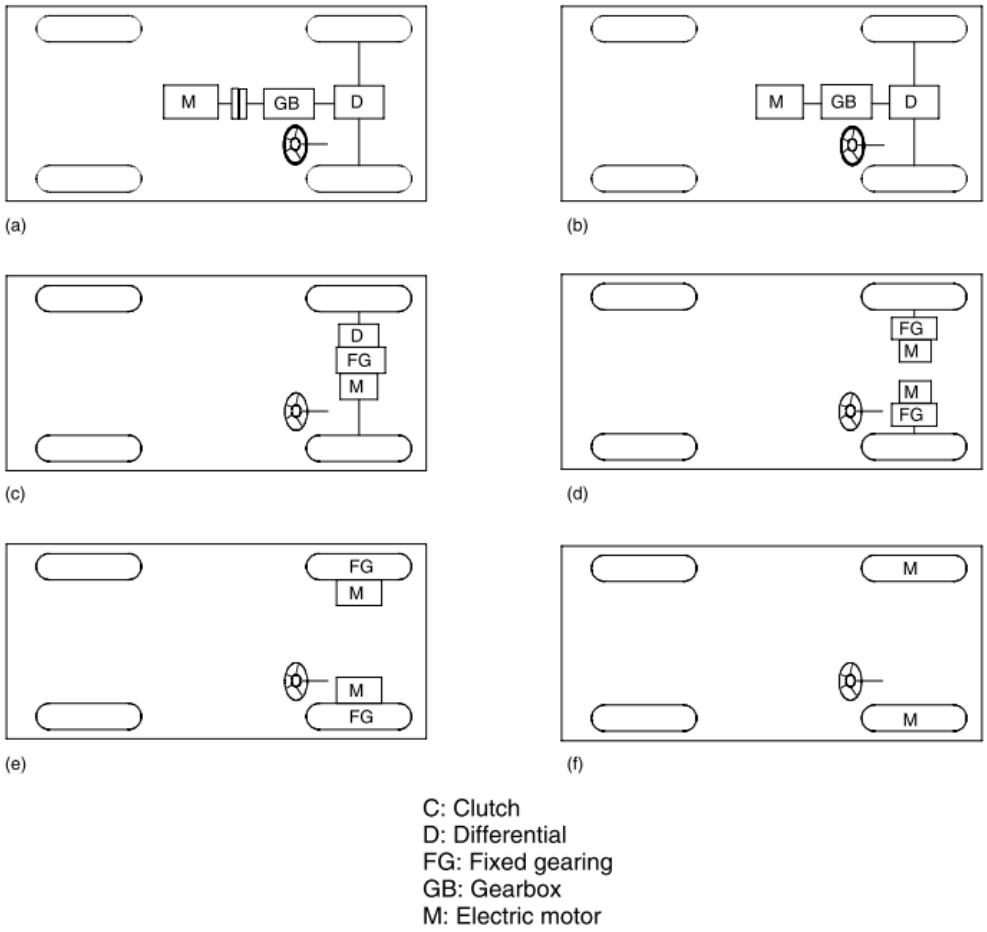

Figure 2. Different EV drivetrain configurations. [33]

(a) The electric drivetrain variation, presented in Figure 2(a), illustrates a similar configuration used with conventional internal combustion engines. Accordingly, the drivetrain consists of a motor, clutch, transmission and differential. This kind of configurations are typically created in conventional drivetrain electrification, where the ICE is replaced with an electric motor while other parts are held in place. [33]

(b) The second type of an electric drivetrain, illustrated in Figure 2(b), is a simplified version of the previous drivetrain configuration in (a). Electric motors tend to have very high torque at low engine speeds. Thus, the clutch is left out resulting in a simpler and lighter drivetrain configuration. However, some electric drives, such as the induction and peak permanent magnet motors, has limited constant power operating range. Consequently, a multi-speed gearbox is still needed. [33] For example, the Proterra Catalyst electric bus [36] utilizes a two-speed gearbox with a peak permanent magnet motor.

(c) Electric motors that have constant power in a wide range of speed, eliminate the need for a gradual torque transmission. As a result, the transmission can be replaced by a fixed gearing further simplifying the configuration. In Figure 2(c), the electric motor, fixed gearing and differential are integrated together in the same drive shaft. Hence, a more compact drivetrain configuration is achieved. [33]

(d) Dedicating each drive wheel with a separate electric motor permits advanced vehicle control, such as stability control and torque vectoring. Additionally, differential is rendered obsolete as each drive wheel can be controlled separately. The fixed gear between the motor and wheels, as in Figure 2(d), enables optimization between maximum motor speed and low speed torque. [33]

(e) In in-wheel electric drivetrain configurations, the electric motors and fixed gears are placed inside the wheel hub. In-wheel motors are utilized in passenger cars and city buses, however, in-wheel motors face challenges in difficult motor control and high costs [37]. The fixed gear can also be replaced with a thin planetary gear in order to gain the benefits of gradual torque transmission. [33]

(f) In some electric drivetrain configurations, all mechanical gearings between the electric motor and wheels are removed. This is the simplest solution as the vehicle speed is directly controlled by the electric motor. However, the removal of the transmission sets pronounced requirements for the motor. The motor must have high low-speed torque and extended speed range. [33]

2.3 Simulation models

Different types of simulation models are presented in this chapter. A simulation model is described as a computational representation that aims to imitate the main objectives or functions of a powertrain system or component [38]. As stated before, there is a variety of simulation tools available that can be utilized to create simulation models with different features and uses. Generally, simulation models used in the design of powertrain systems tend to be less accurate yet wider in scale. In contrast, high fidelity models suit for the research of individual system components as these models prove to be too heavy for simulations in the broader scale.

Models with extreme details and modeling precision are complex to construct and slow to run. For example, simulating couple of seconds in a detailed transient model with subsecond interval can take up to a day in simulation time. Furthermore, the creation of the model requires thorough knowledge of the subject which can prove to be an obstacle. On the other hand, the faster and simplified models have simulation run times of couple minutes. In addition to complexity and run time specifications, the simulation models also differ in flexibility, architecture and graphical interface. [39]

The aim of this chapter is to discuss the attributes of these models and find the most suitable type of model for the battery electric city bus simulation and the driving cycle analysis considered in this study.

2.3.1 Structural and functional models

Structural and functional categorizing refers to the structure of the model architecture. Most of the current simulation tools are based on the structural models where the model is comprised of interconnected blocks which resemble the physical structures of the actual devices.

The structural models are typically straightforward to use due to the physical resemblance to the actual system, the user connects the components to each other just as in the real system. In order to further simplify the use of structural models, most simulation tools include libraries of premade components that can be modified arbitrary by the user. [38]

On the other hand, functional models discard the physical structure of the system and model the system with interconnected mathematical functions [40]. This can make system analysis easier, although, structural models arguably fit better for energy flow models of vehicles due to the physical resemblance of the system.

2.3.2 Steady-state, dynamic and quasi-static models

Various models can be described and classified by the level of transient states in the model and the means of input data acquisition. Three categories are present in the literature, which are in the order of transiency; steady-state, quasi-static and dynamic models [25], [38]. Generally, models with higher degree of transiency are more complex, and thus, more accurate yet more difficult to construct and require more time to run.

Transient states are completely ignored in the steady-state models. Steady-state models are the simplest and fastest to run yet they lack the level of detail in comparison to transient models. These models depend on experimental data that is brought to the model in the form of lookup tables or simplified dynamic models [25], [38]. The data stored in the lookup tables is bound to the specific test setting, in addition, rough engineering assumptions prevail in the simplified models. Thus, the results acquired with the steady-state models are more directional than exact. Due to the fast yet less exact results, steady-state models are best suited for approximate analyses and real-time forecasts utilized in engine control, for instance. In addition, non-transient models can be used to create holistic models by simplifying complex components, such as engines, that would otherwise require excessively complicated models and increased computational performance. [25]

Opposite to steady-state models, dynamic models take in account the transient states. Instead of utilizing static data maps, full-fledged dynamic models aim to computationally imitate the actual functions and procedures in the powertrain system or component. In this way, more precise and general results can be obtained with the model since fewer experiment-based assumptions are made. For instance, power electronics devices, such as the electric motor, are graded by their peak voltage and average current. Even though the average current can be quite effectively approximated with steady-state maps, only rough estimation of the brief peak voltages is acquired. As a result, the accuracy of the model suffers. The transient peak voltages occur in gear shifts and fast accelerations shown in Figure 3, and thus, are evident in city bus driving cycle simulations. [23]

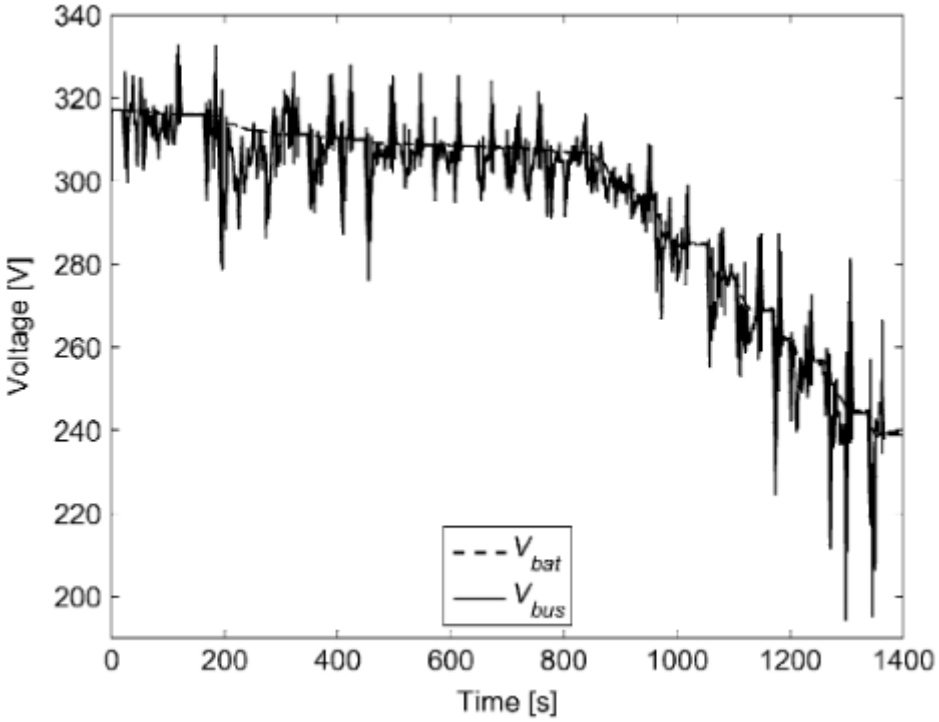

Figure 3. Dynamic behavior of battery voltage V_{bat} and dc-bus voltage V_{bus}. Transient spikes witnessed in the graph occur in gear shifts and fast accelerations. [23]

On the downside, the complexity of the models require thorough knowledge of the system and components which sometimes can be challenging to acquire. Especially with powertrain topologies and components that are still in the design phase. Additionally, computation time can be extremely high in the comprehensive dynamic models, depending on the depth of details. For example, a dynamic model with certain shortcuts [23], it took half a day to simulate a driving cycle similar to this study. Due to the sluggish nature of the transient models, full-dynamic models are mostly utilized in short-term simulations. [23]

Models that combine steady-state and dynamic components in the same model are known as quasi-static. The excessively complex components can be expressed with simplifying steady-state lookup tables while rest of the model consists of dynamic components [23], [25], [38]. Combining steady-state and dynamic models have two advantages. First, the modeler can emphasize a component based on the goal of the study. For example, if the goal is to study the battery, the battery can be an accurate high-fidelity model while rest of the model is simplified. Second, quasi-static models permit compromises between the steady-state and dynamic models. It is possible to create models that are close to dynamic models in accuracy but much faster.

In conclusion, full-fledged dynamic models are too heavy for complete city bus drivetrain models with complete bus routes. Thus, quasi-static models arguably suit best for this study as they offer improved accuracy compared to steady-state models with reasonable run times.

2.3.3 Forward and backward facing models

In addition to the division by transiency, models can also be classified by the direction of calculation. The two directions are forward and backward facing models [25], [38] that can

be utilized independent to the level of transiency. Although, transient components are typically used in the forward approach, while, the backward approach fits best for the static components [41]. Thus, forward facing models tend to be slower and accurate, whereas, backward facing models are faster and more approximate. It is also noted, that some simulation tools, such as the ADVISOR, can utilize simultaneous backward and forward approach in the simulation [42].

In the forward facing models, the energy and signal flows follow the actual flows of the powertrain systems of real vehicles. The calculation begins from the source of energy and advances towards the propulsion of the vehicle, as seen in Figure 4. If the model has capabilities for energy regeneration, the energy flows in the opposite direction. In practice, the model calculates resultant acceleration from the available force provided by the energy source divided by the total effective inertia of the drivetrain [42].

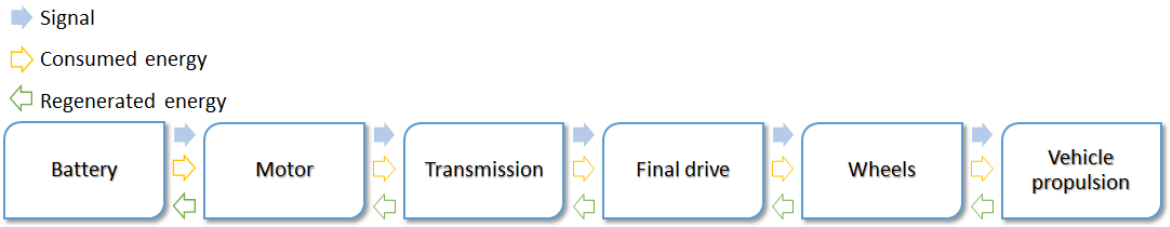

Figure 4. Illustration of the energy and signal flows in forward facing models. The simulation begins from the battery and travels towards the acceleration of the chassis.

Due to the realistic energy and signal flows, the forward facing models typically require driver models. The driver models consist of control signals, such as torque requirement or throttle pedal position, which are transmitted to the engine or engine control unit. The use of realistic driver models results in system operations very close to actual vehicles. Thus, forward facing models excel in system control studies like determining regenerative braking procedures.

Since motoring involves gear changes and other transient events, dynamic forward models are generally used in the design of powertrain controls [23], [41]. Just as in real life systems, forward facing models have a physical delay between the components. Thus, the computation times for the forward facing models are typically extended. Due to the fact that the forward facing models yield results without prior knowledge of the driving cycle, forward facing models excel in situations where there is uncertainty in the driving cycles [41]. It is evident that the irregularity of bus routes certainly poses uncertainty. As this study aims to account for the uncertainty in the driving cycles, the forward approach is essential in the energy flow model.

The backward facing models function in the reverse direction to the real tractive energy flow. The energy used in the upstream component is determined by the energy requirement of the component downstream. The driving cycle sets the first energy requirement for the component upstream of the input. From there on, component by component, the simulation proceeds until the energy source is reached. [41], [42]

The driver behavior is completely eliminated from the model, which implies, that the driving cycle is always precisely met [42]. Moreover, drivetrain capabilities are also neglected in the

backward approach. As by definition, backward facing models do not evaluate the drivetrain's ability to match the reference driving cycle and there is a possibility for the drivetrain to over perform. Hence, the driving cycle must be precisely tailored for the simulation model to avoid unreal drivetrain performance. Uncertainty in the driving cycle can abolish the credibility of the model.

The backward facing models are typically utilized in steady-state energy consumption studies and component design where the driving cycle and drivetrain components are thoroughly known. Additionally, steady-state maps are common in automotive drivetrain component testing, which supports the use of steady-state backward facing models in energy efficiency studies. Simple integration processes, such as the Euler method, supported by the steady-state backward approach allows relatively large time steps to be utilized in the simulation, further decreasing the simulation time. [42]

Some models, such as the ADVISOR, combines elements from approaches in both directions. However, ADVISOR is closely related to backward facing models, where forward approach is only used to adjust the torque according to the drivetrain component capabilities. [42]

All things considered, the forward approach suits best for this study as the driving cycle uncertainty can be addressed with in the forward approach. The emphasis on exact driving cycles, entailed in the backward approach, results in difficulties in driving cycle uncertainty approximation. Additionally, the backward models tend to produce less accurate results. The hybrid backward/forward models can offer some interesting features, such as faster computation, yet the advantages of forward models exceed the benefits acquired with the hybrid approach.

2.3.4 Causal and noncausal models

Causal simulation models are based on the principle of cause and effect; all the outputs in the model are integrated from the inputs. The causal models are asymmetric, thus, the inputs cannot be induced from the outputs [43]. In addition, causality in simulation models implies that there is a time domain between every input and output. Hence, causality is characteristic to forward looking models. [38]

In noncausal models, the causal link between the inputs and outputs is disregarded. Instead, the values of the components can be set based on how the component is linked to other components in the system. Thus, the inputs and outputs of the components in the model can be considered floating. However, floating inputs and outputs require that the component associations must be carefully considered as it is not defined by the model itself. [38]

As some components have fixed inputs and outputs in the causal models, it can lead to difficulties in the component associations. On the contrary, the floating quality of the components in noncausal models permits more unrestricted component association, and thus, noncausal models are typically utilized in structural models. [38]

2.3.5 Simulation tool of choice

Autonomie [10], developed by the Argonne National Laboratory, was chosen as the simulation tool for this study. Autonomie is a high-fidelity quasi-static simulation environment for Matlab/Simulink, based on causal forward facing approach, suitable for energy efficiency studies. A comprehensive set of ready-made, validated components are available in the component model library of Autonomie, which greatly expedites the model construction. In addition, Autonomie's open plug-and-play architecture allows the importation of expert tools from other simulation tools, further increasing the adaptability of Autonomie. [44]

2.4 Monte Carlo method

Monte Carlo simulation method is a tool widely used to simulate and create descriptive forecasts for complex and random processes or ensembles. The applications for the Monte Carlo calculations are numerous, from nuclear physics [45], [46] and biomedicine [47] to social sciences [48] and weather forecasts [49]. It has also been utilized in a parametric energy and cost sensitivity study on plug-in hybrid EVs, where the effects of different vehicle parameters to the energy consumption and system costs were studied [50].

Despite the wide range of applications, it is difficult to provide a single accurate definition of the Monte Carlo method due to the variations in the different applications. Thus, the generally accepted definition for the Monte Carlo method is rather open. Most researchers simply conclude that a Monte Carlo simulation involves creating a stochastic model, with one or more random variables, to represent the examined physical process [45]–[47], [51]–[53].

Although, some adjustments to the definition have been made. Lux and László [45] organize the Monte Carlo simulations into two categories: analog and nonanalog. Monte Carlo calculations that are numerical simulations of real events are referred as analog Monte Carlo simulations. These simulations aim to describe the events as they occur in real life. For example, in nuclear particle simulations, the history of every single particle is simulated precisely. In order to reduce simulation time and complexity, the physical processes can be resembled without the one-to-one simulations of the actual processes. Such Monte Carlo calculations are known as nonanalog. In practice, most of the Monte Carlo applications rarely are neither fully analog nor nonanalog, but rather something between the extremes. [45]

In the end, the Monte Carlo simulations can be generalized into the three step pattern illustrated in Figure 5. First, the model of the process is created, in which, uncertainty is modeled with PDFs. Second, the model is repeatedly sampled with the PDFs which result in outcomes of the process. Third and last, the outcomes are compiled into results which are typically illustrated in the form of histograms or outcome PDFs.

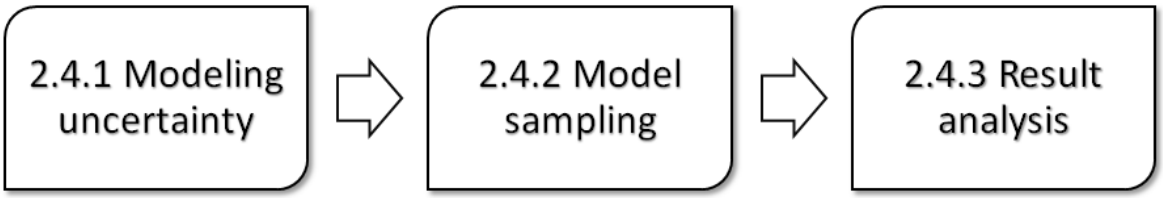

Figure 5. Three steps of Monte Carlo simulation. The first step is modeling the uncertainty with PDFs. The second step is sampling the model. The third step is the result analysis.

2.4.1 Modeling uncertainty

The first step is to create a model of the process and model the uncertain inputs with probability density functions (PDFs), which have a major influence on the results, as all the values of the uncertain input are included in them. Each uncertain inputs are modeled with an individual PDF. Thus, the Monte Carlo simulation can have multiple uncertainty PDFs.

The PDFs of the uncertain inputs represent all the possible values for the uncertain parameter, in addition, probability of occurrence for each parameter are included in the PDF. An exemplary PDF of an uncertain input N illustrated in Figure 6. The uncertain input can have values between 0 and 45, each with a corresponding probability of occurrence. For example, the probability for the uncertain input to occur with $N = 20$ is $P(N = 20) = 0.0255$.

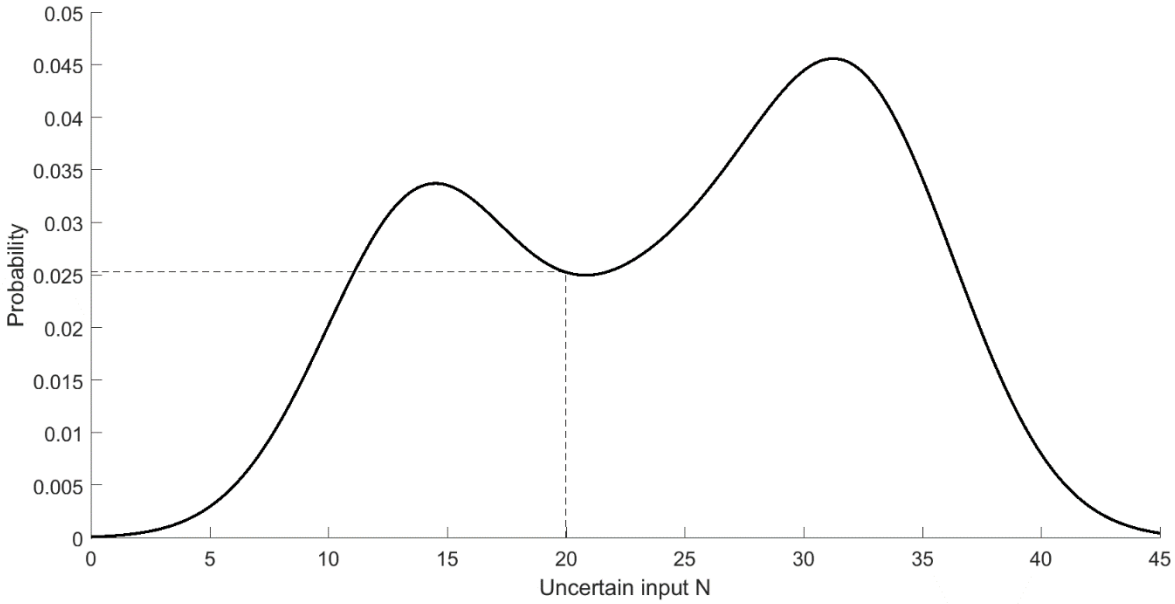

Figure 6. The solid line represents the PDF of an uncertain input parameter N. The probability for $N = 20$ to occur is illustrated with the dashed lines.

Sometimes, the PDFs created are based on prior knowledge of the uncertain inputs [49]. In these cases, the researcher has a clear understanding on what kind of uncertainties are involved and how the different values are distributed in the process. Although, other times the prior knowledge can be limited and it is impossible to form the PDF based on it. Consequently, a common probability distribution, such as the Gaussian distribution, can be utilized

[50]. For example, in a health risk assessment study [54] the population exposure frequency to a health risk was modeled with a normal distribution.

2.4.2 Model sampling

The second step involves a repeated sampling with the same PDFs to acquire the outcomes of the simulation. The uncertain inputs are first determined by the PDFs of the uncertain inputs, and then, run on the process model. Each time the process model is sampled, a new value for the uncertain input is acquired from the uncertain input PDF. Hence, the outcomes of the simulations have no history dependence between each other, instead, the results are derived only from the common initial conditions. In this fashion, the model is sampled until the iterative process reaches an eligibly stable point, convergence. [55]

There is a tradeoff between the convergence and simulation time. A higher number of sampled points results in better convergence, however, more time is also consumed during the computation. As a result, the target accuracy of the simulation must be carefully designated by setting an adequate number of sampled points. [50]

The amount of samples is usually chosen by examining the iterative accuracy of the sampling. For example, a common way to determine the amount of samples is based on the difference between the theoretical forecast mean and each point of sample. In the parametric vehicle study [50], the Monte Carlo simulation was first sampled 1000 times to gain information about the convergence. The total outcome average after 1000 simulation was set as the theoretical forecast mean. Then, the error at each sample point was computed by comparing the outcome to the theoretical forecast mean. In this fashion, it was found out that a somewhat sufficient accuracy was achieved after 100 simulation. However, it took 600 simulations to acquire results with suitable accuracy. Thus, it was concluded that 1000 simulations are more than sufficient for accurate results.

2.4.2.1 Theoretical Monte Carlo sampling example

In practice, the process of Monte Carlo simulation can be described as creating a K-dimensional hypercube with N points [50]. Earl and Deem [55] present a general computational example of Monte Carlo method which embodies the theory behind the tool. Their example exemplifies how the Monte Carlo simulation aims to calculate a state of balance for the system or process of interest.

In their example, an average value (A) is calculated for a property A:

$$(A) = \frac{\int \exp[-\beta U(\boldsymbol{r}^N)] A(\boldsymbol{r}^N)\, d\boldsymbol{r}^N}{\int \exp[-\beta U(\boldsymbol{r}^N)]\, d\boldsymbol{r}^N}, \tag{1}$$

where $\beta = 1/k_B T$ is the reciprocal temperature, U is the potential energy and $\boldsymbol{r}^N$ marks the configuration of a N particle system. The uncertainty is modeled with a probability density of finding the system in configuration $\boldsymbol{r}^N$:

$$p(\boldsymbol{r}^N) = \frac{\exp[-\beta U(\boldsymbol{r}^N)]}{\int \exp[-\beta U(\boldsymbol{r}^N)]\, d\boldsymbol{r}^N}. \tag{2}$$

As N_{MC} points are generated randomly in the configuration space based on the Eq. (2), the average (A) can then be expressed accordingly:

$$(A) \approx \frac{1}{N_{MC}} \sum_{i=1}^{N_{MC}} A(\boldsymbol{r}_i^N). \tag{3}$$

The average (A) in the Eq. (3) is calculated by sampling the function $A(\boldsymbol{r}_i^N)$ for N_{MC} points. Thus, based on the Eq. 3, the more samples are executed the more accurate results are acquired, as the errors in (A) scale with $1/N_{MC}$.

2.4.3 Result analysis

As mentioned above, the Monte Carlo simulation method is utilized to create descriptive forecasts of the examined process. Typically, the forecasts are presented with graphical illustrations, such as binned histograms, which provide distinct illustrations of all the possible outcomes. In most cases, the binned distributions are estimated with an uniform PDF [45], [46], [49], [50]. The illustrations are utilized to unambiguously point out the distribution of the outcomes, and can be supplemented with certain key figures, such as mode, mean and confidence intervals, to further describe the forecasts [50].

Binned distributions can be estimated with PDF estimations to provide more explicit descriptions of the results. In addition, key figure determination can be more straightforward for a PDF estimation. Although, if the outcome distribution does not correspond to a common distribution, such as the normal distribution, the PDF estimation may not be recommended [50]. Figure 7 illustrates the distribution for a parameter x acquired in the sampling phase of the Monte Carlo simulation. As the distribution closely resembles a normal distribution, it is further estimated with a normal PDF.

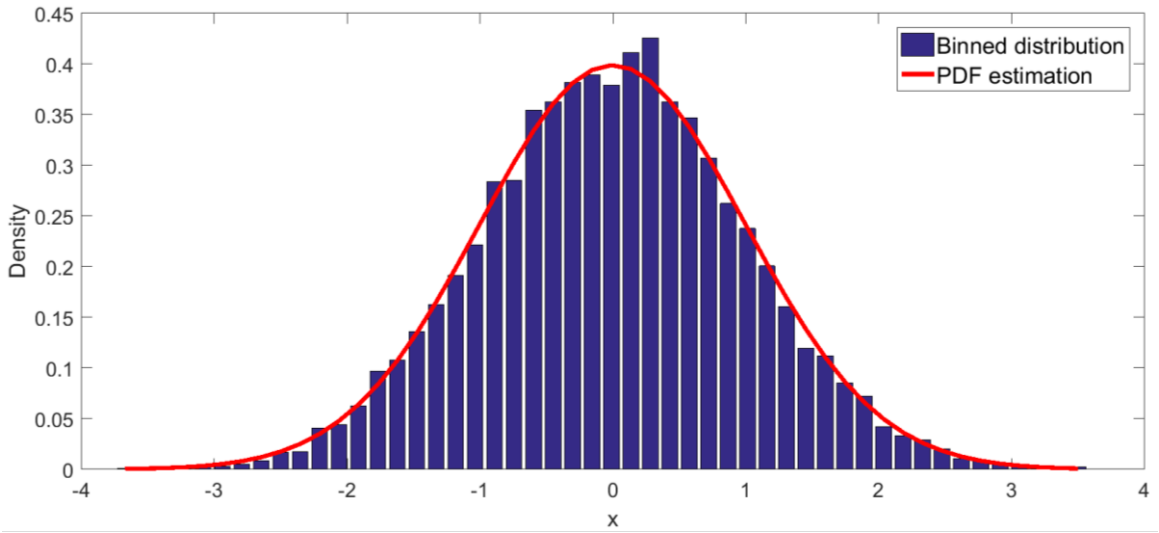

Figure 7. Binned distribution of outcome x and a normal estimation of the distribution.

Three key figures, illustrated in Figure 8, can be added to the outcome distributions. Mode is the single most probable outcome to occur, it is illustrated in binned outcome distributions with the tallest bar. Respectively, mode is the highest point in uniform PDFs. Mean is the overall average of all outcome values. Confidence interval is a range of values which occur with a specified probability. For example, 95 percent confidence interval implies there is a 95 percent chance that an outcome of the process is within the bounds of the interval. Alternative confidence interval ranges are also present in Monte Carlo applications, for example, the parametric energy and cost sensitivity study [50] utilized an 80% confidence interval.

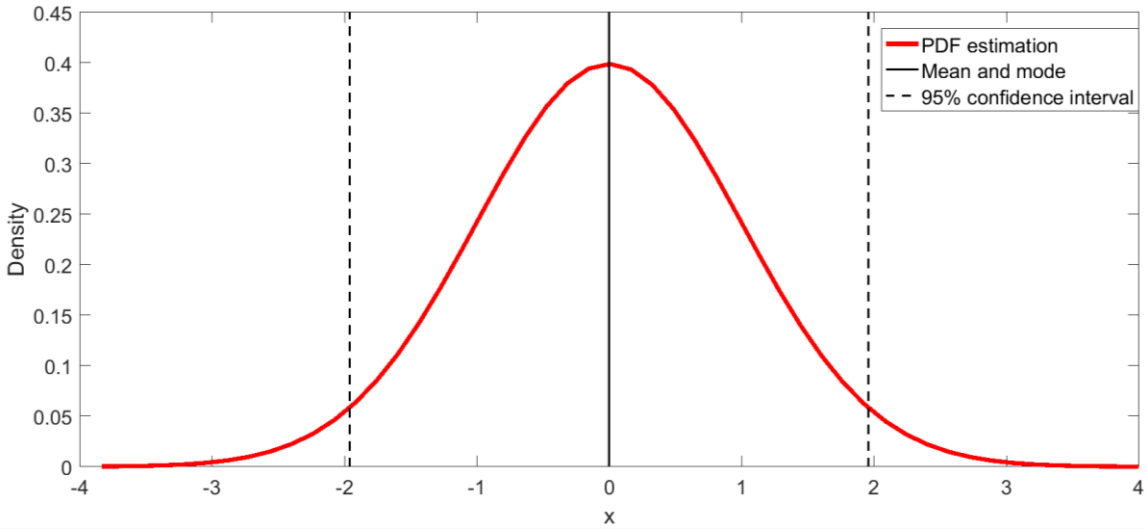

Figure 8. Key figures for the PDF estimation. Mean and mode, illustrated with the solid black line, are equal for a normal distribution. The black dashed lines represent 95% confidence interval, which means, 95% of the outcomes are included between the dashed lines.

The result distributions presents all the possible outcomes, considering the uncertainty of the model. Key figures can be added to the distributions to further clarify the probabilities of certain outcomes. In conclusion, with the aid of key figures, the outcome distributions of the Monte Carlo simulations provide statistically justified forecasts of the examined process.

3 Bus route electrical consumption sensitivity analysis

The process of creating the bus stop based electrical consumption sensitivity analysis is presented in this chapter. The bus route includes a number of random variations, for example, the route duration, velocities and number of stops vary between different runs. Consequently, utilizing only one reference driving cycle in the electrical consumption simulations does not provide a robust generalization of the bus route energy consumption. Moreover, measuring a vast number of different driving cycles in the route to include every deviating alternative of the bus route proves to be overwhelmingly burdensome and difficult, if not impossible.

Instead of measuring a massive set of variating driving cycles by hand, the Monte Carlo sampling method was be used to sample the uncertain inputs with desirable distributions and create the set of driving cycles accordingly. Hence, the uncertainty entailed in the number of stops is addressed with the driving cycle sensitivity analysis.

The analysis comprises of four stages, illustrated in Figure 9, which lead to statistical sensitivity analysis. The first stage consists of data logging and parsing, in which, the bus coordinate and speed traces were logged with a GPS logging device and then parsed into base driving cycles in Matlab. The second stage involves the synthetization of new driving cycles based on the base cycles measured in the first stage. The synthetization includes a Matlab built script that creates new driving cycles with randomized bus stops and speed profiles. In the third stage, the synthetization script is sampled to create a requisite number of different driving cycle samples. The sampling is performed with Monte Carlo sampling method where the uncertain inputs are modeled with a normal distribution. In the fourth and last stage, an electric city bus energy flow model is created and used to run the samples. The results acquired by this four stage process are analyzed in the next chapter.

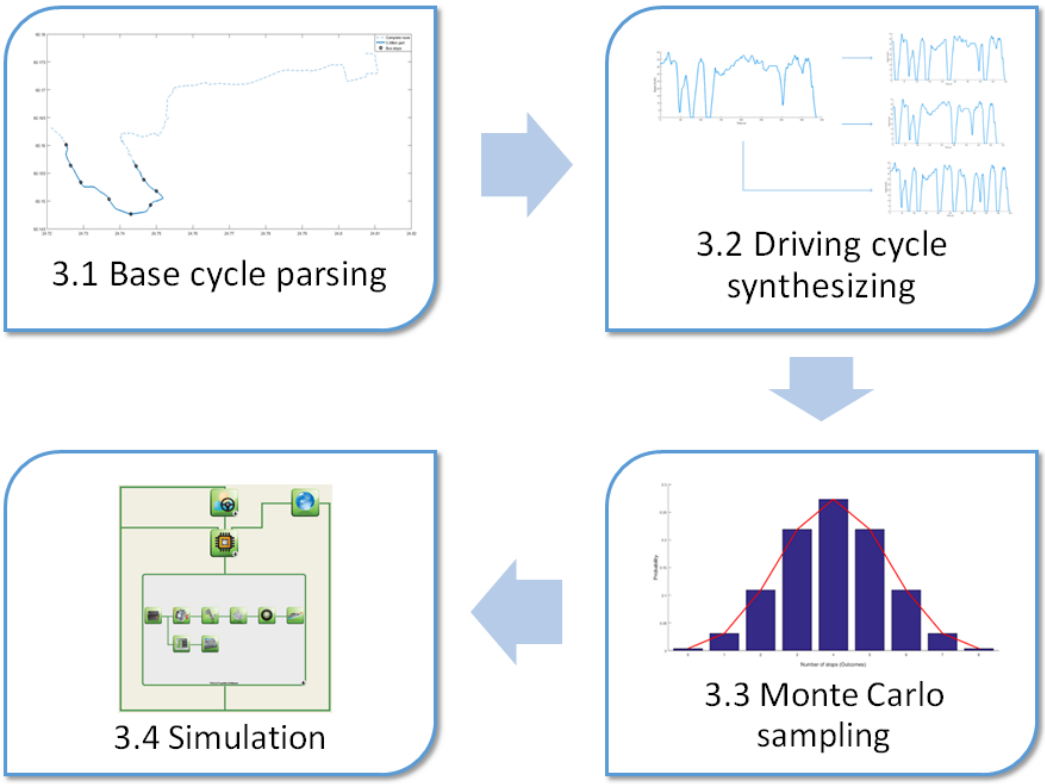

Figure 9. Four stages of bus route electrical consumption sensitivity analysis.

3.1 Base cycle parsing

The driving cycles are one of the key components in the sensitivity analysis, as the simulation results are derived from the cycles. There are several factors that need to be taken in account with the driving cycles to acquire valid simulation results. To begin with, the driving cycle has to resemble the actual driving route in every aspect, which means, the driving cycle has to have uniform the length and features with the actual roadway. Otherwise, the results do not describe the subject route. In addition, the driving cycle must be compatible with the vehicle used in the analysis or the vehicle is not able to follow the speed trace in the simulations accordingly. The subject vehicle sets certain constraints based on the performance of the vehicle, such as acceleration limits, which the driving cycle cannot exceed.

Hence, to meet with the requirements set previously, the driving cycles were logged with a GPS logging device in actual city buses operating a real bus route. The route was first mapped ten times, and then, the recorded base driving cycles were filtered in two phases and parsed shorter. In this part, the reference bus route is presented along with its speed trace logging, data parsing and filtration.

3.1.1 The route and GPS logging

The route analyzed in this study is based on an urban bus route operated by line 11 in Espoo, Finland. The direction from Friisilä to Tapiola is considered, which is roughly 10.1 kilometers in length. Ten measurements were made in the buses operating the line to account for the different speed profiles of the route. The measurements took place in three consecutive days between noon and evening (Table 1). In addition to the ICE counterparts, two electric city buses operated at the line at the time of the measurements. The operating vehicles, in which the measurements were performed, were chosen randomly. As a result, both ICE and electric driven city buses were included in the measurements.

Table 1. Time and duration of GPS logged bus route. Measurements with electric buses are marked with *(e)*.

No.	Date	Time	Total duration	3.3 km part duration
1	08/03/2016	14:31 – 14:59	28 min	6 min 38 s
2 *(e)*	08/03/2016	15:46 – 16:16	30 min	6 min 36 s
3	09/03/2016	12:37 – 13:04	27 min	7 min
4	09/03/2016	13:47 – 14:12	25 min	6 min 12 s
5	09/03/2016	16:15 – 16:44	29 min	7 min 6 s
6 *(e)*	09/03/2016	17:28 – 17:53	25 min	6 min 35 s
7 *(e)*	10/03/2016	11:52 – 12:18	26 min	5 min 43 s
8	10/03/2016	13:02 – 13:30	28 min	7 min 4 s
9 *(e)*	10/03/2016	14:12 – 14:42	30 min	7 min 17 s
10	10/03/2016	15:31 – 16:02	31 min	7 min 3 s

The speed trace was logged with a Racelogic VBOX mini GPS logger [56] with data sample frequency set to 10 Hz. Lest to lose any useful data, no pre-processing was utilized in the data acquisition system during the measurements.

However, due to some constant GPS signal interruptions present in certain parts of the route in every measurement and to lighten the calculations, only a 3.3 km part of the route was implemented in this study. The 3.3 km part of the route, illustrated in Figure 10, is a two lane road with nine bus stops, one roundabout, one intersection with a turn and a speed limit of 40 km/h.

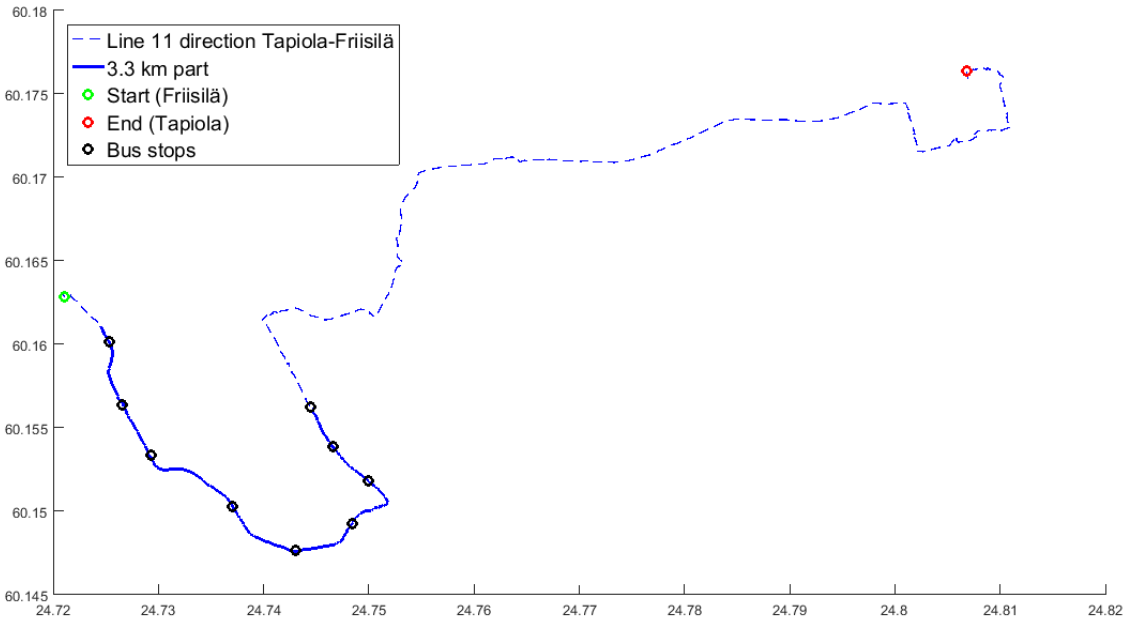

Figure 10. Line 11 route with 3.3km part, start, end and bus stops.

3.1.2 Filtration

Due to no preprocessing of the logged data, the raw data proved to be very noisy (Figure 11) with many interruptions and accelerations peaking up to 35 m/s^2 (Figure 13). Thus, the logged driving cycles had to be processed before further use. The raw data was filtered in two stages; low-pass filtration and acceleration filtration.

In the first stage, low-pass Butterworth IIR filtration, with filter order of 2 and half power frequency of 0.04, was applied to the speed-time profile to smooth out the high peak interferences and noise in the data. The Butterworth filtration resulted in considerably smoother speed profiles (Figure 11), which is apparent in the far less noisy acceleration profile (second image in the Figure 13). Although, despite the Butterworth filtration, the acceleration still remained to be excessively high in some parts. Hence, a second filtration was required to provide compatible driving cycle for the electric bus.

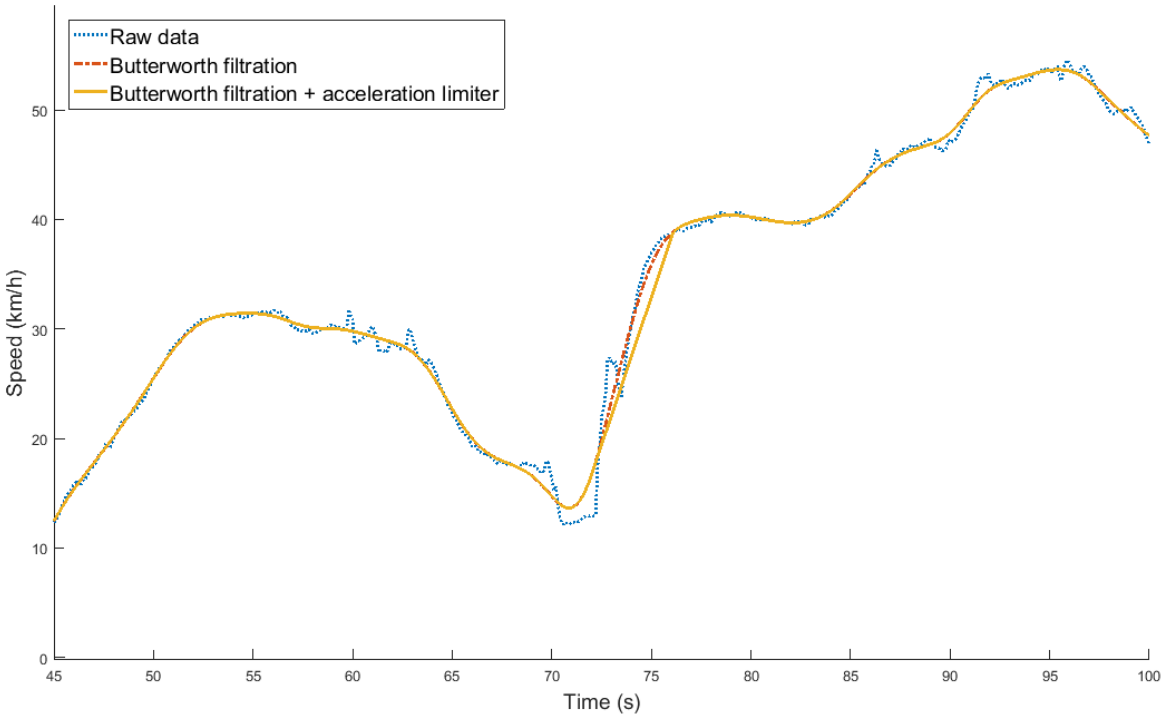

Figure 11. Speed profile samples of raw, Butterworth filtered and Butterworth filtered + acceleration limited data.

The second filtration was based on limiting the maximum acceleration of the vehicle. To find out the maximum acceleration, a maximum acceleration test was performed with the simulation model of the electric bus. In the test, a constant 320 km/h velocity demand is set for the driver. Even though the vehicle is not able travel at such high velocities, the velocity demand obliges the driver to meet the demand. Consequently, the driver sets full throttle for the vehicle.

The vehicle accelerates until an equilibrium, where motion opposing forces equal propulsive forces, is reached and the speed stays constant. Hence, the maximum acceleration and speed of the vehicle are discovered. The results of the acceleration test are illustrated in Figure 12, where the maximum constant acceleration is 2.2 m/s^2 roughly for the first 4 seconds. After the bus reaches 30 km/h, the acceleration starts to diminish gradually, until the maximum speed of 98 km/h is reached after 25 seconds from the beginning of the test.

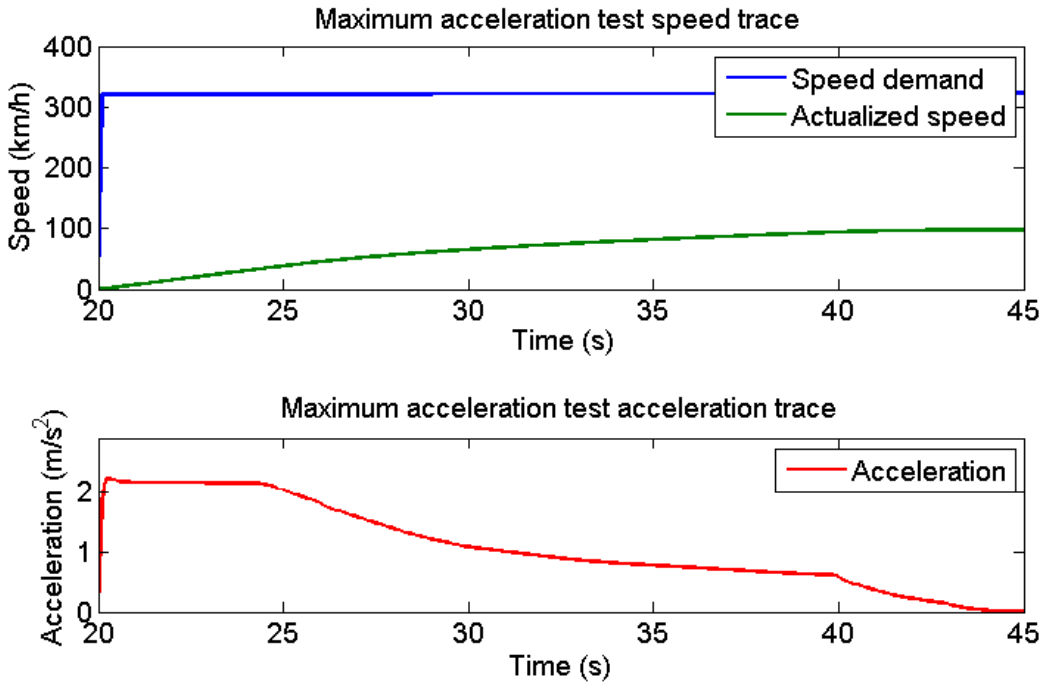

Figure 12. Speed and acceleration traces of the maximum acceleration test.

In addition to the maximum acceleration test, the maximum acceleration limit was also researched based on literature. A study carried out by Hoberock [57] suggests the maximum longitudinal acceleration of a ground transport vehicle should not exceed $0.11g \approx 1.1$ m/s^2. It is assumed, the drivers try not to exceed this boundary as higher accelerations tend to feel uncomfortable. However, it is noted the acceleration can surpass the suggested value due to different driving circumstances and the differences in the driving behavior of bus drivers. Therefore, to account for the possible transcending accelerations, the maximum possible acceleration a_{max} was set to 1.5 m/s^2.

If the acceleration between two consecutive points in the speed profile exceeded the a_{max}, the latter value of velocity was lowered to match acceleration of a_{max}. This change can be seen in the Figure 11 between 70 and 80 seconds, the slope of the orange curve is lower than in the red dashed curve. In addition, the limited maximum acceleration can be witnessed in the Figure 13 between the second and third picture.

Only longitudinal positive accelerations was considered in the acceleration limitation filtration. Deceleration was filtered only once in the low-pass filtration and left untouched in the acceleration filtration because higher decelerations seem to be more comfortable than accelerations, according to the Hoberock's study [57]. Additionally, higher magnitudes are achieved in the braking compared to acceleration with the bus. Thus, braking is not as restricted as acceleration.

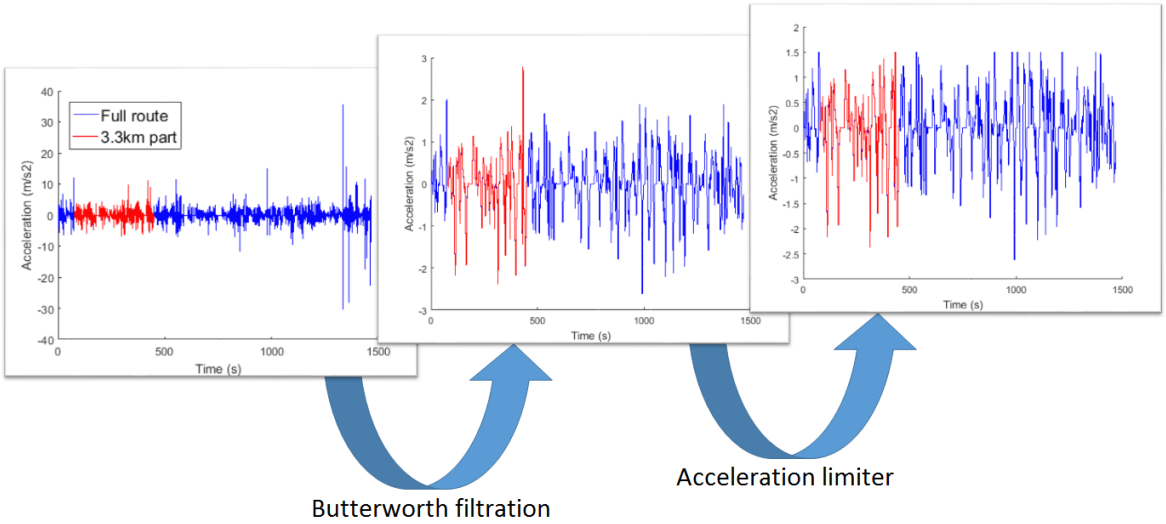

Figure 13. Acceleration profiles and two stage filtration. Blue line represents the full route, whereas, the red line represents the 3.3km part of the route. In the filtration process, low-pass filtration (Butterworth filtration) is performed first. After, acceleration filtration is performed.

As a result, ten unique driving cycles, illustrated in Figure 14, were obtained from the same 3.3 km part of the line 11. Each run varies in accelerations, velocities and number of stops, even if it was operated by the same driver with the same vehicle. Some of the driving cycles have higher top speeds and accelerations, while other cycles have milder driving characteristics. Based on the speed trace, no significant difference can be seen between ICE and electric driven speed profiles.

It is evident, that there is an endless amount of differences between each run of the route. Moreover, only a tiny portion of all the possible variations are covered with the measured cycles. Thus, in order to perform a comprehensive sensitivity analysis, more varying cycles are required.

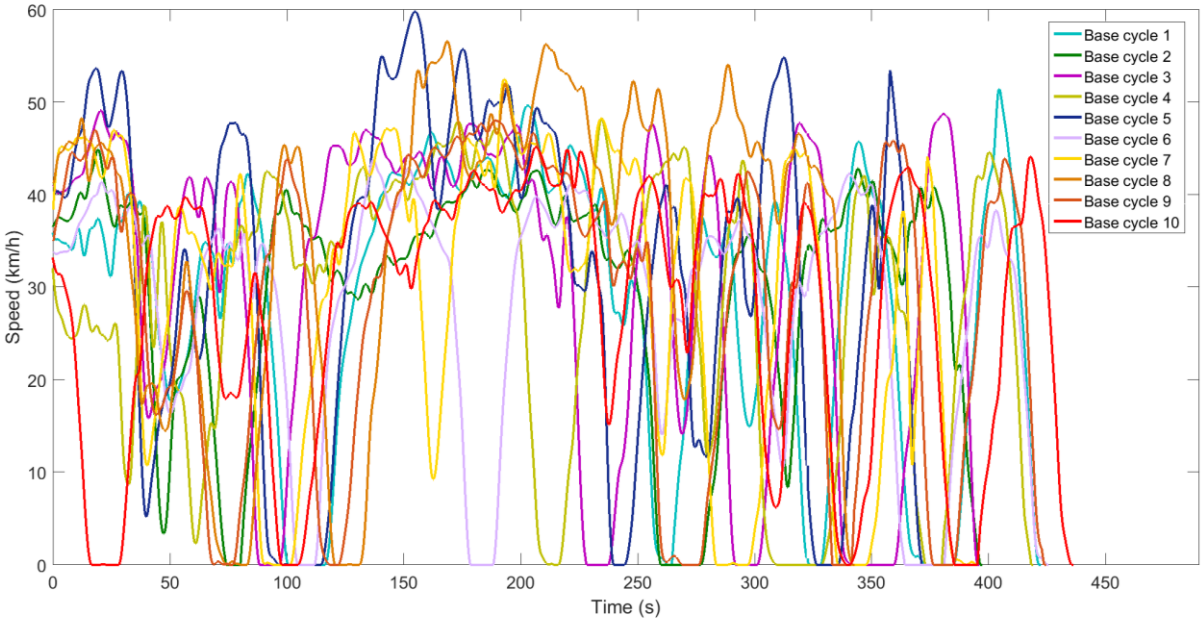

Figure 14. The ten measurements over 3.3km distance after the data parsing.

3.2 Driving cycle synthesizing

In order to cover the variations in the number of bus stops and speed profiles, the driving cycles utilized in the sensitivity analysis were artificially synthesized based on the ten measured driving cycles. The synthetization was performed with a Matlab script which creates new cycles with custom number of stops and random combination of the base cycles.

The synthetization process begins from choosing the individual bus stops. The stops can be set manually by choosing individual bus stops or leaving the choice to the random generator based on the pseudo-random number generator of Matlab. Then, the bus stops are either cut off or added in the driving cycle by function scripts presented in the next subchapters. As a result, the number of bus stops in the driving cycle is set accordingly.

Additionally, the speed profile variations evident in different cycles are addressed by combining different parts of the ten measured driving cycles. Combining the speed profiles of different driving cycles is a straightforward and efficient way to synthesize valid driving cycles. As all parts of the new driving cycles are based on actual measurements, the restrictions and characteristics of the vehicle and route are already included in the cycles. As a result, the cornering speeds, top speeds and accelerations precisely meet with the bus restrictions. In addition, the total length of the route and route characteristics, such as location of curves and intersections, are also correct in the new driving cycles.

Thus, there are two steps in the driving cycle synthetization. The first step is the bus stop determination, in which, the bus stops are set. In the following step, different parts of the measured driving cycles are combined. After these steps, a new driving cycle is synthesized.

The number of different driving cycles synthesized with this method can be calculated:

$$N_{cycles} = \sum_{i=1}^{n_{stops}} S^{i+1}, \tag{4}$$

where n_{stops} is the number of possible stops in the route and S is the number of base cycles. In this study, the number of stops n_{stops} is 8 and the number of base cycles S is 10. Thus, the amount of different possible driving cycles, according to the Eq. (4), is:

$$N_{cycles} = \sum_{i=1}^{8} 10^{i+1} = 1\,111\,111\,100. \tag{5}$$

In this fashion, as the Eq. (5) points out, it is possible to synthesize a massive number of different driving cycles, with different number of stops and different alternative speed profiles, based on the ten measurements. It must be noted, that the different driving cycles synthesized with this method are all based on actual measurements. Thus, all the parts of the driving cycles reflect real life situations in the route.

3.2.1 Altering the number of stops

The number of stops is altered by utilizing Matlab built function scripts to either cut off a bus stop from the driving cycle or adding one to it. As the ten measured driving cycles covered only a portion of the bus stops, artificial stoppings had to be implemented with the driving cycles. Utilizing the function scripts presented in the following subchapters, it is possible to arbitrary set the location and number of bus stops for the synthesized driving cycles.

3.2.1.1 Skipping a stop

A function script was generated in Matlab to remove bus stops from the driving cycles. This script, presented in the Appendix B, removes a stopping sequence from the driving cycle and creates a new, modified version accordingly, as presented in Figure 15. The script ensures the modified driving cycle still resembles the route by keeping the total distance equal and road characteristics similar with the unmodified cycle.

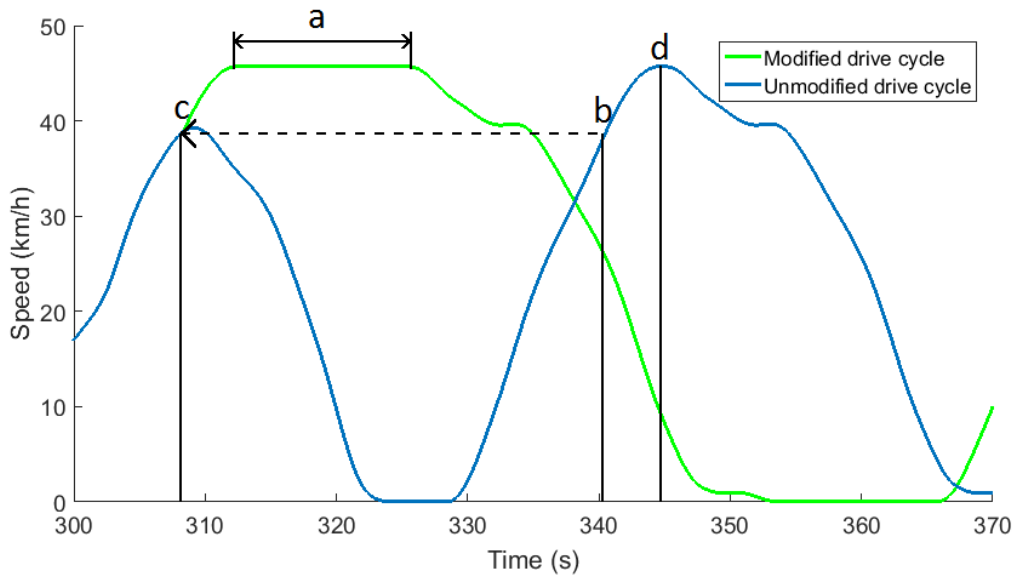

Figure 15. Cutting a stop from the driving cycle. The green line is the modified driving cycle where the bus stop has been completely cut off. The blue line is the original unmodified driving cycle, in which, the bus makes a stop between 322 and 329 seconds.
(a) is the distance compensation
(b) is the cutoff end point
(c) is the cutoff start point
(d) is the highest point of velocity next to the stop

Cutting a stop is based on two points in the driving cycle; the point before the deceleration to the stop (point c) and the highest velocity next to the original stop (point d). The deceleration to the stop, the time spent at the stop and the acceleration from the stop are removed from the driving cycle. Then, the cutoff end point (point b) is connected to the cutoff start point (point c). Thus, the bus does not slow down at the bus stop, instead it keeps on accelerating until it reaches the peak velocity which it would have reached if it had skipped the bus stop.

However, a distance has been traveled during the deceleration and acceleration in the original run. Hence, the distance (between the points d and c) must be compensated as otherwise the new driving cycle would be too short. The distance is compensated by travelling a constant speed for a required time (line a). The distance traveled during the deceleration and acceleration was calculated with the sum of distances between each measurement point from the start to the end of the removed section. Therefore, the compensated distance s_{comp} is:

$$s_{comp} = \sum_{i=c}^{d} v(i) * \Delta t, \tag{6}$$

where $v(i)$ is the velocity at time i, c is the cutoff start time, d is the cutoff end time and Δt is the time between two sample points in the driving cycle, which equals 0.1 s due to 10 Hz data log frequency. In conclusion, the duration t_a that has to be traveled with the constant speed at point d can be calculated by:

$$t_a = \frac{s_{comp}}{v(d)}, \tag{7}$$

where s_{comp} is the compensated distance calculated with the Eq. (6) and $v(d)$ is the velocity at point d.

3.2.1.2 Creating a stop

As the bus did not make all the eight stops in any of the measured runs, some the stops were required to be artificially added to the driving cycles. Thus, a function script, presented in the Appendix C, was generated for adding a stopping sequence to the driving cycle. This script modifies the driving cycle by adding a stop to the driving cycle, illustrated in Figure 16. The script ensures the modified driving cycle still resembles the route by keeping the total distance equal with the unmodified cycle.

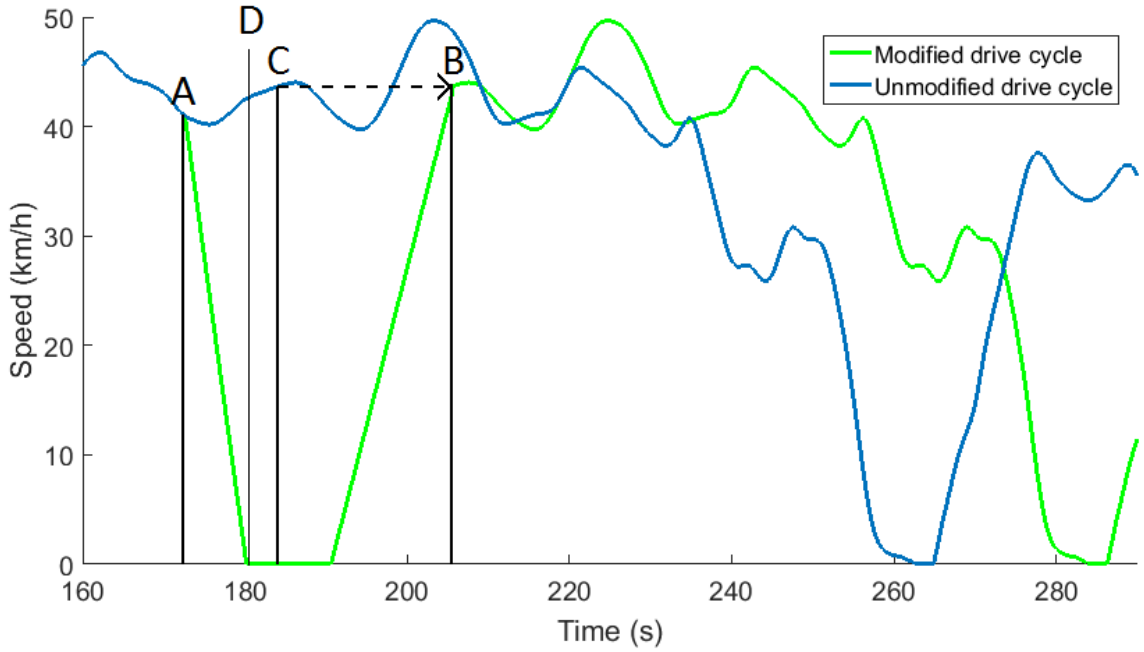

Figure 16. Creating a stop to a driving cycle. The green line is the modified driving cycle with a new stop between seconds 175 and 190 in the picture. The blue line is the original unmodified driving cycle which skipped the bus stop in examination.
(A) is the starting point of stopping sequence
(B) is the end point of the stopping sequence
(C) is the copied point in the original driving cycle from where the new driving cycle continues
(D) is the point of time of the bus stop in examination

Three phases can be determined for the stopping sequence. First, the bus has to decelerate to the bus stop. Then, the bus has stay halted for some time to allow passengers enter or exit the vehicle. After that, the bus accelerates to a speed based on the route and speed limit.

The first phase is pulling up to the bus stop. To ensure this with reasonable accuracy, the deceleration starting point (point A) is calculated by subtracting the deceleration time from the bus stop point of time (point D) of the original driving cycle. The time t_A required to stop the vehicle is calculated with the following equation:

$$t_A = \frac{v_1 - v_0}{a_{decel}} = \frac{-v_0}{a_{decel}}, \tag{8}$$

where v_1 is the velocity at the end of stopping to the bus stop and is equal to zero, v_0 is the velocity at the bus stop in the original driving cycle and a_{decel} is the deceleration which was set to constant (-) 1.5 m/s^2. Thus, the bus should land reasonably close to the bus stop.

The next phase is idling at the stop which simulates the passengers entering and exiting the vehicle. Based on the measured data, the stopping time at every created bus stop was set to ten seconds as it seems to resemble the typical time spent at the bus stop. The speed is simply set to zero for the duration of the idling.

After the idling, the last phase is the acceleration from the bus stop. A constant acceleration is added to the driving cycle until the original speed trace is reached. However, once again a distance is covered while decelerating and accelerating at the bus stop. Therefore, a part

has to be cut from the driving cycle to ensure the distance of the trip stays constant. The distance traveled during the acceleration and deceleration is calculated according to the Eq. (6). Then, the point C is determined by matching distance traveled between points A and C, in the original unmodified cycle, to the distance covered during the acceleration and deceleration of the modified cycle. Thus, the driving cycle continues from the point C and the part between points A and C is cut off.

3.2.2 Variating speed profiles

As pointed out earlier, speed profile variations are realized by combining different base driving cycle parts between the bus stops. The speed profile variation is included in the script that synthesizes new driving cycles and is executed after the bus stops are determined. The script is presented in the Appendix A.

The procedure of variant speed profile combination is illustrated in Figure 17. After the stops are chosen, the script chooses randomly one of the ten measurements and creates a base driving cycle according to the bus stop selection. Then, the script counts the number of stops made and creates an equal amount of additional driving cycles with the same bus stops as the base cycle. Next, the speed trace after the first stop is replaced with the speed trace of the first additional driving cycle. As a result, the new combined driving cycle follows the speed trace of the base cycle until the first stop, afterwards, the cycle follows the speed trace of the first additional driving cycle. The same procedure is repeated until the last stop. Consequently, the synthesized driving cycle is a combination of various parts of the different measured base driving cycles.

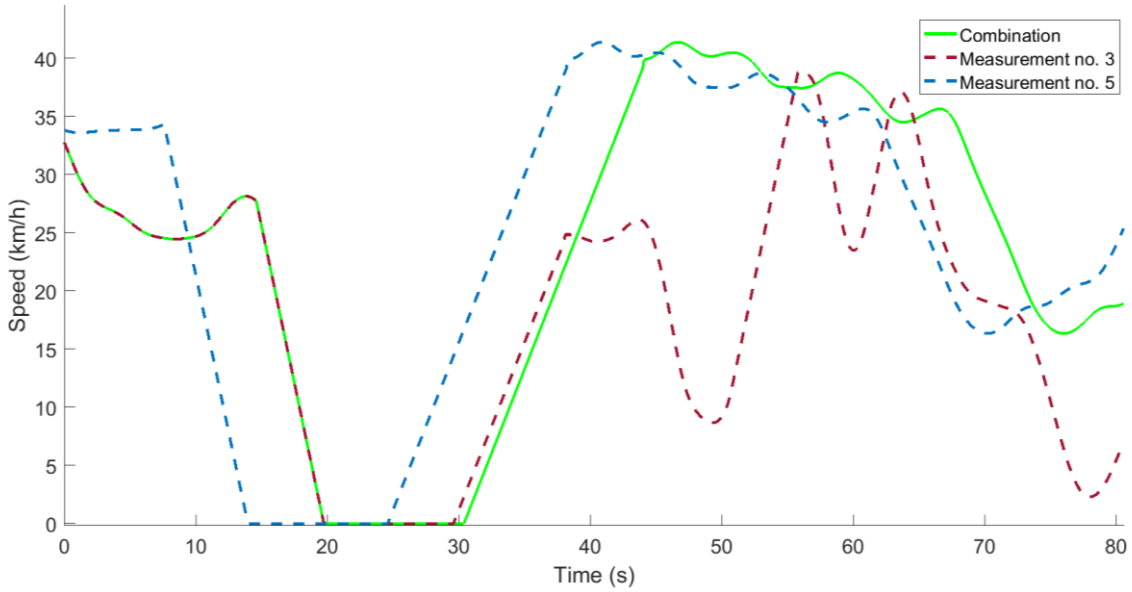

Figure 17. Combining two different driving cycles into one cycle. At first, the combined driving cycle follows the base measurement no. 3, after the first stop, it changes to the driving cycle corresponding the measurement no. 5.

3.3 Monte Carlo sampling

The Monte Carlo method is applied to address the sensitivity of the city bus driving cycles. Instead of measuring hundreds of cycles with different stops, the route was sampled with the Monte Carlo method to gain information about effects of bus stops to the energy consumption. Even though, there are many other sampling methods, such as hypercube sampling methods, the Monte Carlo sampling method was chosen as it proves to be reasonably accurate and straightforward tool to analyze models with low number of inputs [50].

This section consists of two dissections. First, the uncertain inputs are scrutinized by creating a PDF to model the uncertainty. Next, the convergence of the input sampling is examined by identifying adequate number of sampled points within reasonable the error bounds.

3.3.1 Uncertain inputs

The uncertain input models the uncertainty entailed in the process model. Individual inputs are sampled from the uncertainty PDFs which are formed to represent the uncertainty. In this study, the number of bus stops is regarded as the uncertainty. As the actual distribution of bus stops was unknown, the uncertainty is modeled with a discrete normal distribution, illustrated in Figure 18.

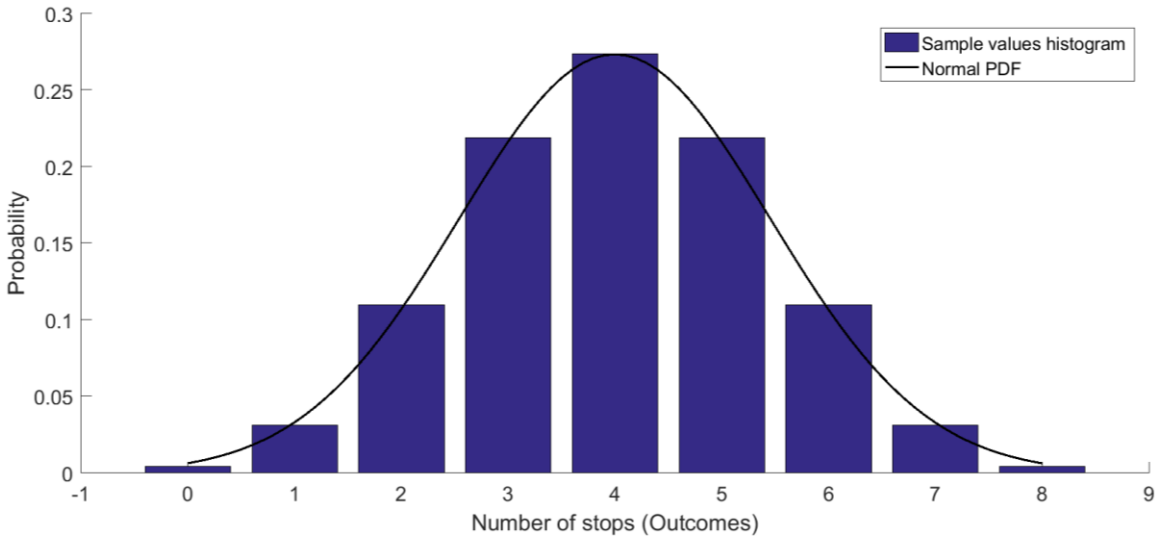

Figure 18. The number of bus stops is sampled from a normal distribution PDF.

As normal distribution is utilized to model the uncertainty, the probability of pulling up to a stop is assumed to be equal for each stop. In the simulation, the bus has a 50 percent chance to stop at any bus stop regardless of prior stops. As there are eight bus stops (n_{stops}) with two possible outcomes, stopping or passing, the total number of variations (n_{var}) is $2^8 = 256$. Therefore, the probability of stopping to n number of bus stops, $P_{stops}(x = n)$, is:

$$P_{stops}(x = n) = \frac{\binom{n_{stops}}{x}}{n_{var}} = \frac{\frac{8!}{n!(8-n)!}}{2^8}; n \in 0,1,2,\dots,9 \quad (9)$$

Consequently, the distribution of the number of stops made during the run matches with a normal distributed PDF. The outcomes of the uncertainty PDF determine how many stops the bus makes on a single run. The outcome is sampled from the PDF every simulation cycle with probabilities illustrated in Figure 18.

3.3.2 Input sampling

The simulation was run 1000 times to gain information about the convergence. The development of the convergence is illustrated in Figure 19. A reasonable convergence was acquired relatively fast, 0.5% error bound was achieved after 108 simulations. Whereas, a more accurate convergence took somewhat longer, 0.1% error bound was achieved after 450 simulations. In this study, relatively strict error bounds can be set for the convergence, because accurate estimation is acquired fairly fast, due to the fact that only one uncertain input was sampled.

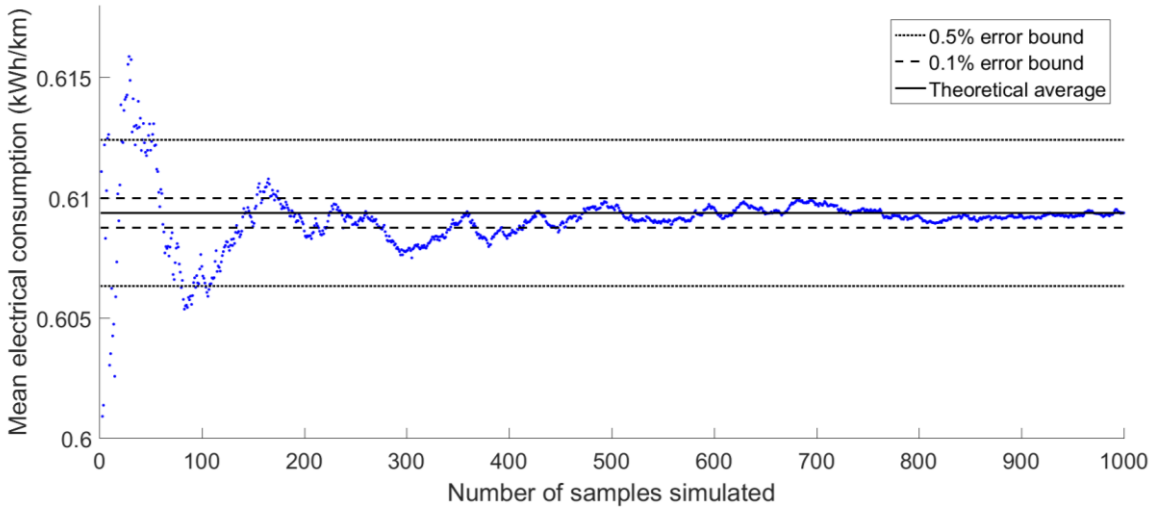

Figure 19. Impact of sample number on sampled input mean. The error is scrutinized by examining the convergence of the average outcome. Less than 0.5 % error was achieved after 108 simulations, while, less than 0.1 % error was achieved after 450 simulations.

Thus, based on the convergence analysis, at least 450 simulations, with one uncertain input, are required to provide accurate results with the Monte Carlo sampling method in this case. Simulation time-wise, 450 simulations should be manageable with the 3.3km part considered in this study, as the simulation time is proportional to the distance of the route. Simulation time for the 3.3km part should be roughly one third compared to the full route.

3.4 Electric city bus energy flow model and simulation

In this chapter, the energy flow model created to simulate the battery electric city bus is presented. First, the simulated battery electric city bus is introduced including all the main parameters of the bus. Then, the overall layout of the model is illustrated along with the functions and principles of every block in the model. The base of the model was created by Antti Lajunen in Autonomie and utilized in studies [2], [35]. The base model was modified to match a two-axle rear-axle-driven battery electric city bus validated in a recent study [58].

In the last chapter of this section, the modified base model was validated against dynamometer results of a full-scale electric city bus prototype measured in the previous study [58]. The details of the bus model and validation are presented below.

3.4.1 Battery electric city bus

The vehicle modeled in this study is a low-floor *M_3 Class I* battery electric city bus. It has two axles with tractive rear wheels. The curb weight of the vehicle is 9 345 kg, however, three different weight options were utilized in this study for the model validation. The three options include constant extra loads of 1 000kg, 3 000 kg and 5 655 kg. The bus has a permanent magnet electric motor with 207 kW nominal power, accompanied with a fixed reduction gear box and 56 kWh battery. Thus, the drivetrain configuration is essentially the same as illustrated in the Figure 2b of the previous chapter. The main parameters for the vehicle are listed in the Table 2.

Table 2. Main parameters of the vehicle model.

Subsystem	Parameter	Value
Vehicle	Curb weight	9 345 kg
	Total mass: Comparison 1 Comparison 2 Comparison 3	 10 345 kg 12 345 kg 15 000 kg
	Frontal area	6.2 m^2
	Drag coefficient	0.5
	Rolling resistance	0.0075
Drive motor	Nominal torque	1100 Nm
	Nominal speed	1800 rpm
	Nominal current	254 Arms
	Nominal voltage	500 V
	Efficiency in nominal point	96%
Energy storage	Capacity	56 kWh
	Nominal voltage	614.4 V
	Charging current	90 A, max 125 A
	Discharging current	180 A, max 270 A
	Resistance	0.192 Ω

3.4.2 Vehicle model overview

All the components and variables of the model are built in the graphical user interface (GUI) of the vehicle model in Autonomie, illustrated in Figure 20. The vehicle model is divided into four subsystems:

- Driver model
- Vehicle propulsion controller (VPC)
- Environment model
- Vehicle propulsion architecture (VPA)

The driver model imitates the driver of the vehicle and feeds the driver operated controls to the VPC. The VPC compiles signals from all the other subsystems and forms control parameters for the VPA accordingly. The environment model emulates the environment and bases its calculations on preset values and data from the VPA. The VPA is the most important subsystem of the energy flow model as it simulates the mechanical construct of the vehicle.

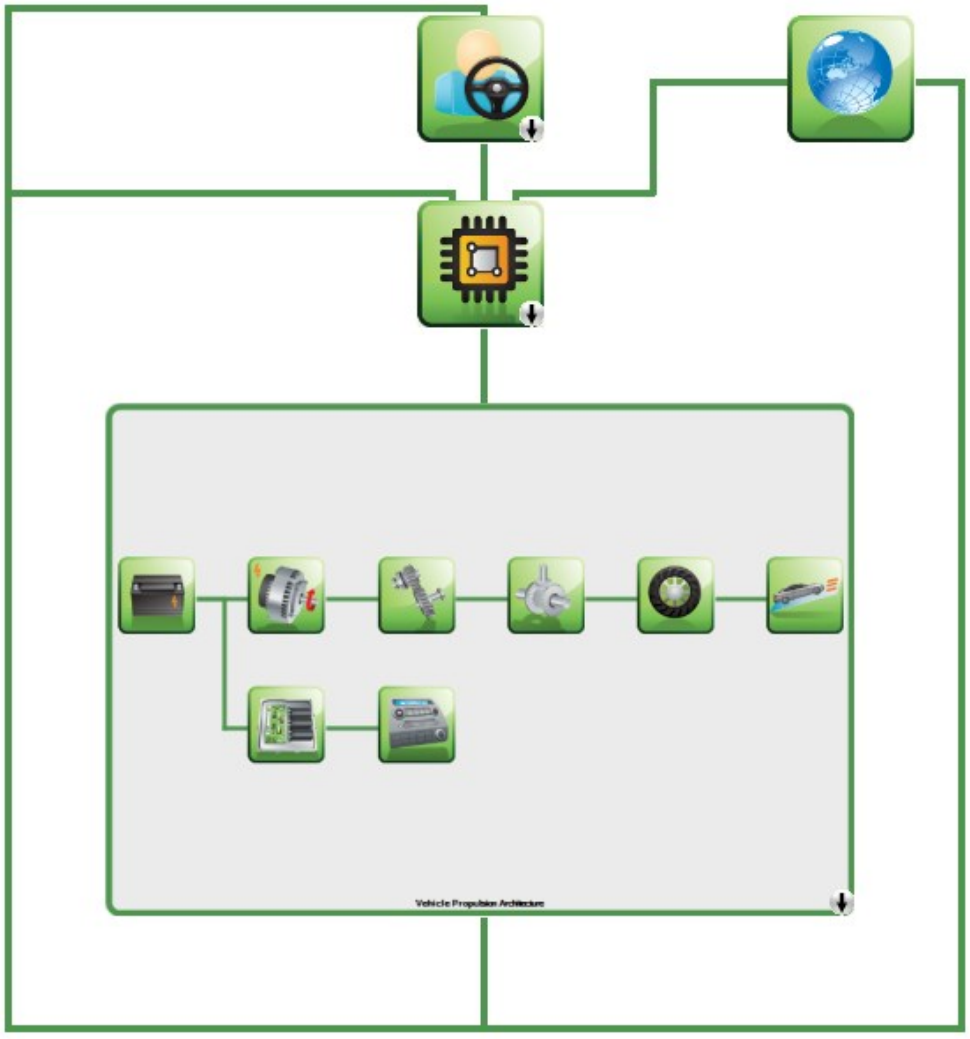

Figure 20. The GUI of the energy flow model in Autonomie consists of four main subsystems: The driver model, VPC model, environmental model and VPA model.

3.4.3 Driver model

In forward-facing models, the driver model sets acceleration and brake pedal positions according to the speed trace of the inputted driving cycle. There are two types of driver models in the pre-made component library of Autonomie: the proportional-integral (PI) controller and look-ahead based driver models. The PI controller based driver models rely on instantaneous PI control for trace following, which results in almost identical simulated vehicle speeds in comparison to the inputted driving cycle. However, the result cycle has a time lag compared to the inputted cycle due to the delay entailed in the system. To counter the time delays, look-ahead based driver models can be modified to react to the trace in advance with a desired weight coefficient. This should result in less fluctuations in the vehicle speed curves, and thus slightly lower energy consumptions. [59]

A look-ahead based driver model was utilized in this study with following parameters:

- Look-ahead anticipation time: 1 s
- Anticipation weight coefficient: 1.2

The effects of the look-ahead based drive model is illustrated in Figure 21. The driver anticipates the input driving cycle with one second time period and neglects sudden fluctuations of the driving cycle based on the anticipation weight coefficient. Thus, the anticipation weight coefficient virtually acts as a third filter for the driving cycle. All in all, a close correspondence is achieved between the input driving cycle and actualized driving cycle. In addition to the driver anticipation,

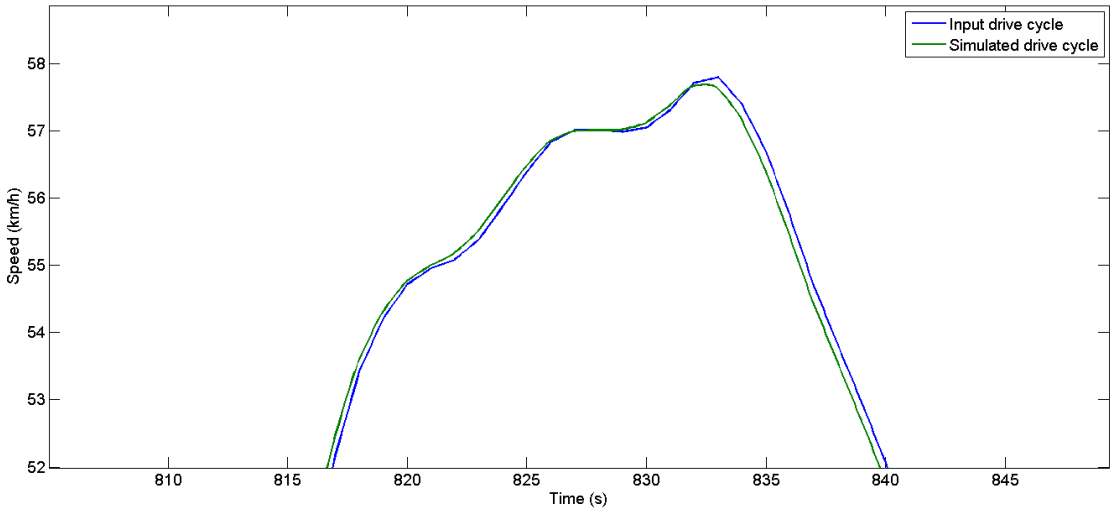

Figure 21. The difference between the input driving cycle and realized driving cycle with look-ahead based driver model. The realized driving cycle, in green, anticipates the input driving cycle with one second time period and smooths out sudden fluctuations.

3.4.4 Vehicle propulsion controller

The VPC distributes propulsion and braking efforts, based on the outputs of all the other main parts of the model, for the components in the vehicle propulsion architecture. For example, the torque demand that the VPC delivers to the drive motor is determined by the acceleration and brake pedal positions, vehicle speed, motor speed and battery state of charge. The VPC outputs also other control parameters such as engine on/off and regeneration mode.

In addition, the VPC also controls the regenerative braking conditions. There are three possible modes for the regeneration: no regeneration, partial regeneration and full regeneration. Each mode has a set of rules that determine the mode of regeneration, the rule are presented in the Table 3.

During the no regeneration mode, only mechanical braking is applied. This mode is regulated by three rules. First, decelerations above 4 m/s^2 are considered emergency braking, in which, only mechanical braking is applied. However, the deceleration will not exceed 4 m/s^2 in any part of the driving cycle, so this rule should not affect the results. Second, no regenerative braking is applied if the state of charge (SOC) of the energy storage is above 92%. Lastly, if the vehicle speed is less than 5.4 km/h, no regeneration is applied.

The partial regeneration mode combines mechanical braking at the wheels with the inertia of the electric drive. The vehicle utilizes partial regeneration mode if the deceleration is below 4 m/s^2, SOC below 90% and vehicle speed above 5.4 km/h.

In the full regeneration mode the whole braking effort is solely caused by the inertia of the electric drive. The vehicle enters this mode if the deceleration is less than 0.87 m/s^2 when the mass is 10 345kg, 0.72 m/s^2 when the mass is 12 345kg and 0.60 m/s^2 when the mass is 15 000 kg. The deceleration limits for each mass correspond a 90 kW regeneration power input limitation of the battery pack. In addition, the vehicle speed has to be over 10.8 km/h and SOC below 90%.

Table 3. Rules for different regeneration modes.

Mode	No regeneration	Partial regeneration	Full regeneration		
Rules	Decel. above 4 m/s^2 SOC above 92% Vehicle speed < 5.4 km/h	Decel. below 4 m/s^2 SOC below 90% Vehicle speed > 5.4 km/h	SOC below 90% Vehicle speed > 10.8 km/h		
			10345kg Decel. below 0.87m/s^2	12345kg Decel. below 0.72m/s^2	15000kg Decel. below 0.60m/s^2

3.4.5 Environment model

The environment model defines the parameters of the surrounding environment for the simulation. The parameters of the environment model are ambient air pressure, relative air humidity and ambient air temperature. Other environmental factors, such as wind speed and direction were not included in this simulation model.

In this study, the following parameters were set for the environment:

- Ambient pressure: 101.3 kPa
- Relative air humidity: 50 %
- Ambient temperature: 20 °C

3.4.6 Vehicle propulsion architecture

The VPA represents the actual drivetrain system configuration, and is built to match the real vehicle drivetrain layout. The system consists of models for energy storage, motor, torque coupling, final drive, wheels, chassis, and electrical accessory including power converter.

As with the forward-facing models in general, the energy storage supplies energy to other components in the model. Component by component, the energy flows from the energy storage unit to the electrical accessory and wheels. Additionally, energy regeneration is enabled in the model with reverse energy flows. Electrical energy is regenerated into the battery during braking when certain conditions, regulated by the VPC, are met. The energy flows are illustrated in Figure 22. Each component has a total efficiency that defines the amount of energy lost when energy travels through the component.

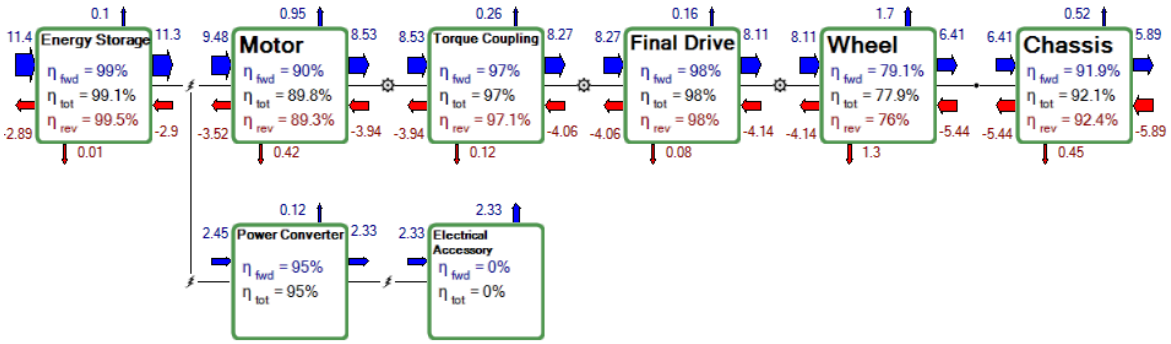

Figure 22. The vehicle propulsion architecture layout and energy flows. Arrows entering the blocks represent energy in, arrows exiting blocks energy out. Energy losses due to efficiency are on top and bottom of the blocks. The blue arrows represent propulsion energy flows and red arrows regenerating energy flows. Flash icons represent electric connections and cog wheel icons mechanical connections between the blocks. The values shown in the figure are representative.

3.4.6.1 Energy storage model

The drive battery is modeled in the energy storage block that consist of two blocks, controller and plant. The controller regulates the maximum charging and discharging power of the

energy storage depending on the transient SOC of the plant. While, the plant is the actual battery model, in which, the terminal voltage is based on an open circuit voltage and internal impedance per cell in charge and discharge.

The battery model has an internal thermal model which takes in account the effects of temperature in impedance and open circuit voltage. Voltage and current are provided to the drive motor and auxiliary power converter by the plant model. The energy storage model is initialized with following parameters:

- Cell mass: 1.25 kg
- Cell pack mass: 240 kg
- Energy storage system mass: 1 050 kg
- Number of cells in series: 168
- Number of cells in parallel: 4
- Initial SOC: 90 %
- Maximum SOC: 90 %
- Minimum SOC: 20 %

3.4.6.2 Auxiliary devices model

The electrical auxiliary system includes two models: a power converter and accessory block. The power converter is a steady-state efficiency model that provides the accessory block with electric voltage and current with following parameters:

- Constant efficiency: 95 %
- Output voltage: 150 V

The accessory block simulates the power consumption of auxiliary devices of the vehicle. The model is a constant power loss block that consumes energy at desired quantity over time. In this model, the accessory block total power consumption is estimated with a continuous mean power consumption. Each part of the accessory devices are included in the average power consumption estimation. The power consumption of the air compressor, used in the door operations, is estimated with a duty cycle percentage. The duty cycle percentage is the estimated door operation time over the whole test period.

In this study, following constant power loss was set for the accessory:

- Air compressor power: 6 kW
- Air compressor duty cycle: 23%
- Total accessory constant power loss: 1.5 kW

3.4.6.3 Drive motor model

The drive motor model consists of a controller subsystem, which has constraint and command blocks, and plant block. The constraints block of the controller subsystem regulates the positive and negative torques for the command block, which in turn, sends the motor

control signals to the electric drive based on the constraints, motor speed and the torque demand provided by the VPC.

The drive motor operations are modeled in the plant block. The drive motor is a permanent magnet synchronous motor, in which, model voltage and speed are inputted with command controller signals. Outputs are motor current and the rotor torque. Following parameters were used in the simulation:

- Maximum current of the motor: 1 241 A
- Motor inertia: 0.0621 kgm^2
- Motor mass: 266.92 kg
- Maximum torque: 1 100 Nm
- Slew rate time constant: 0.05
- Minimum voltage: 180 V
- Maximum motor power 315 kW

3.4.6.4 Torque coupling and final drive models

The torque coupling model is a model located between the electric motor and final drive, and used to scale the motor torque and angular velocity. The engine speed is multiplied by the torque coupling ratio which effects the engine speed and torque according to the Eq. (10), in which, the power is constant. As the engine speed increases, torque must decrease, and vice versa. In terms of engine torque and speed, the power is

$$P = \tau \cdot \omega, \tag{10}$$

where τ is torque and ω is angular velocity.

The final drive acts as a similarly to the torque coupling and is located between the torque coupler and wheels. In this model, neither the torque coupler nor final drive ratio have inertia, and thus, operate lossless. Following parameters were used in this study for the torque coupler and the final drive:

- Torque coupling ratio: 1.75
- Final drive ratio: 4.72

3.4.6.5 Wheel model

The wheels are modeled with two models, controller and plant models. The wheel controller model combines the braking commands from the brake pedal position in form of driver model brake demand and electric motor mechanical braking from the VPC brake demand. After forming the total brake demand from these two demands, the wheel controller model forwards them to the wheel plant model.

The wheel plant model simulates the wheels of the vehicle by calculating the angular velocity and the torque of the wheels which result in the traction. The equation for wheel torque T_{wheel} is

$$T_{wheel} = T_{in} - T_{brake} - T_{roll}, \tag{11}$$

where T_{in} is the torque transmitted from the final drive, T_{brake} is the torque applied by brakes and T_{roll} is the torque loss caused by rolling resistance.

Following wheel model parameters were used in this study:

- Rolling resistance coefficient: 0.0075
- Wheel inertia (per wheel): 15 kgm^2
- Total wheel mass: 390 kg
- Number of wheels: 4
- Wheel radius: 0.478 m
- Maximum braking torque 120 000 Nm

3.4.6.6 Chassis model

The final model in the forward facing simulation model is the chassis model. This model is used to calculate the impacts of road grade and aerodynamic drag in the vehicle dynamics. As the road grade is neglected in this study and it has no effect on the results, only aerodynamic drag causes a force opposing the vehicle movement in the chassis model. The equation for aerodynamic drag force in the vehicle model is

$$F_D = \frac{1}{2}\,\rho\, v_{model}^2\, C_D\, A_{model}, \tag{12}$$

where ρ is the air density, v_{model}^2 the velocity, C_D the drag coefficient and A_{model} the frontal area of the vehicle model.

Following chassis parameters were used in this study:

- Total mass: 12 345 kg
- Drag coefficient: 0.5
- Frontal area: 6.2 m^2
- Center of mass height: 0.77 m
- Weight distribution (rear %/front %): 60/40

3.5 Simulation model validation

The simulation model was validated by comparing the results to a prior study conducted by Halmeaho [58], in which, the same bus was validated based on the same bus line (Line 11). In the prior study, the results acquired with an energy flow model was created in Matlab/Simulink/Simscape environment, were compared to measurements made on a dynamometer with the actual bus.

Similar comparison was made in this study, the same driving cycle was run on the energy flow model of this study and then compared to the results acquired in the study of Halmeaho. Three comparisons were made, each time with different total weight. In the first run, the total mass of the vehicle was 10 345 kg which means the curb weight has an additional 1 000 kg weight load. The second run was made with 3 000 kg addition, which totals 12 345 kg. The third run had the highest total mass of 15 000 kg, or 5 655 kg added to the curb weight.

All the values of the comparison are presented in the Table 4. Close correspondences were achieved with the energy flow model in the first two comparisons, especially in the Comparison 2 where the difference between the simulated and measured values were minuscule. However, in the Comparison 3 the total error of the result rose close to 6 percent. This is probably caused by the increased braking energy recovery when the mass is increased.

Table 4. Comparisons between the energy consumptions of the energy flow model used in this study and values measured in [58] **.**

	Comparison 1 Line 11 – 10 345 kg	Comparison 2 Line 11 – 12 345 kg	Comparison 3 Line 11 – 15 000 kg
Simulated (kWh/km)	0.512	0.584	0.691
Measured (kWh/km)	0.518	0.581	0.734
Total error	-1.16 %	+0.52 %	-5.86 %

Due to the smallest error and representing the average cycle between the comparisons, the vehicle mass for the simulations was set to according to the Comparison 2, 12 345 kg.

4 Results

In the result section, the distribution of outcome samples of the sensitivity analysis, average electrical consumption by number of stops and the effect of acceleration amplitude to electrical consumption are presented.

4.1 Electrical consumption distribution and confidence intervals

All the electrical consumption outcomes obtained with the Monte Carlo simulation were collected in a distribution histogram, along with key values that further articulate the distribution. The key values include mode, mean and 80 percent confidence intervals. In addition, a kernel estimation was computed from the outcome values and compared to normal distribution to gain information on how well the results match a normal distribution.

The distribution histogram is presented in Figure 23. The modes, which are the most probable values to occur in this distribution, are represented with red dashed lines. The modes are located more on the left side of the graph, which implies, the outcomes seem to be slightly weighted below the mean. The mean is represented by blue dashed line.

In addition, confidence intervals, illustrated with green dashed lines in the Figure 23, were calculated for the distribution. The confidence intervals group the outcomes that occur with a selected probability. In this study, 80 percent confidence intervals were added to the figure. The lower bound of the interval represents the 10 % percentile of the distribution, while the upper bound is the 90% distribution percentile. Thus, a forecast can be formed which states that with an 80 % probability the electrical consumption will fall between the bounds of the confidence interval, when considering the uncertain inputs of this study.

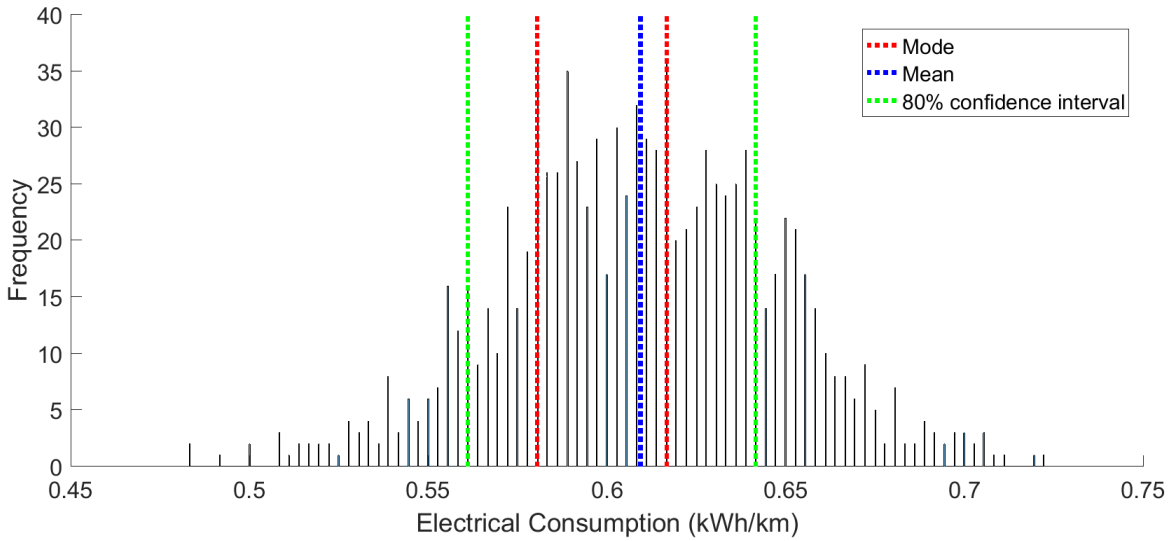

Figure 23. Electrical consumption distribution histogram, including key figures, for normal distributed uncertain PDF. The red dashed lines represent the modes of the distribution. The blue dashed line represent the average value of electrical consumption. The green dashed lines represent the boundaries of the 80 % confidence interval.

The modes, mean and confidence interval calculated for the outcome PDF are listed in the Table 5. In addition, the variation between the lowest and highest value in the confidence interval and the minimum and maximum are presented in the table.

Table 5. Mode, mean, confidence interval, minimum and maximum values for the outcome PDF.

Consumption variable	Electrical consumption (kWh/km)
Mode	0.581
	0.618
Mean	0.609
80% confidence interval (variation in %)	0.561 – 0.642 (14.4 % variation)
Minimum (difference to the mean %)	0.483 (-20.7 %)
Maximum (difference to the mean %)	0.722 (+18.6 %)
Total difference % between min and max	49.4 %

The distribution seems to closely resemble a normal distribution, although, a normal distribution should have only one mode equal to the mean. To further examine the resemblance to normal estimation, a kernel estimation PDF was created based on the outcome data and compared to a normal estimation. The kernel estimation was performed with the inbuilt kernel estimation tool in Matlab.

Kernel estimation is a tool used to form unbinned non-parametric presentation from a binned distribution such as the histogram [60]. The estimation is defined by a smoothing function and a bandwidth value which controls the smoothness of the kernel distribution. Following equation is given for the kernel estimation:

$$\widehat{f_h}(x) = \frac{1}{n_{kernel}h} \sum_{i=1}^{n_{kernel}} K\left(\frac{x - x_i}{h}\right);\ -\infty < x < \infty, \tag{13}$$

where n_{kernel} is the sample size, $K(\cdot)$ is the kernel smoothing function and h is the bandwidth.

Matlab has an inbuilt kernel estimation tool which estimates each column of a binned distribution with a continuous PDF. Then, the tool compiles the overall kernel distribution from the smaller component curves. The kernel estimation process is illustrated in Figure 24. [61]

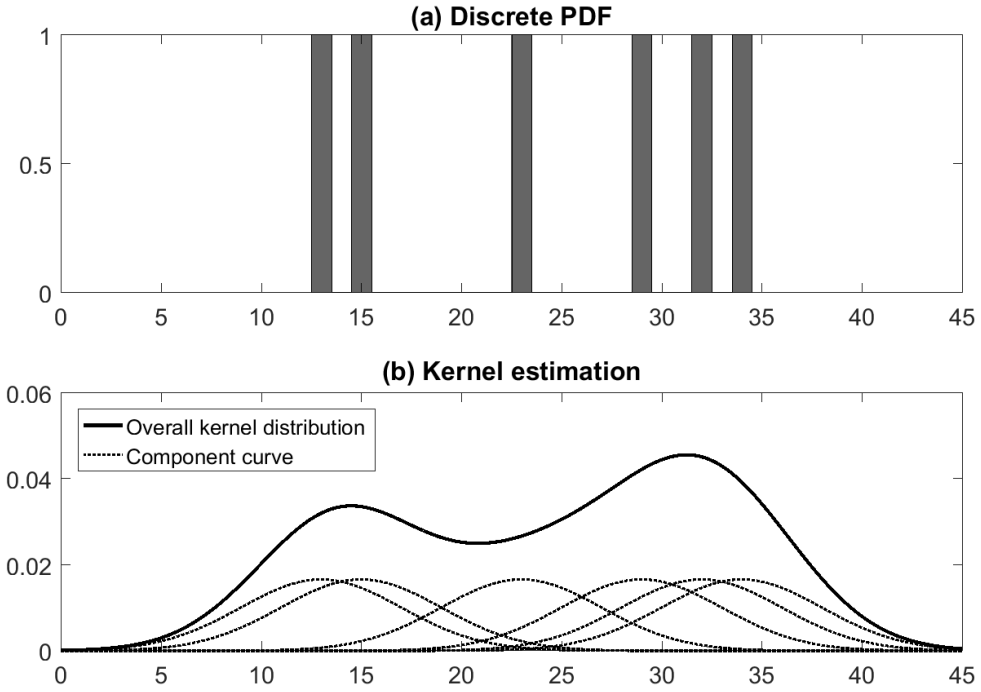

Figure 24. Formation of kernel estimation.
(a) Discrete PDF
(b) Kernel estimation created based on the discrete PDF. The overall kernel distribution is compiled of the smaller component curves which represent the columns of the PDF in (a).

Comparing the kernel estimation to a normal distribution confirms that the outcome histogram resembles the normal distribution closely yet not exactly, as seen in Figure 25. The reason, why an exact resemblance with the normal distribution was not achieved, probably lies in the different speed profiles compiled together to synthesize the driving cycles.

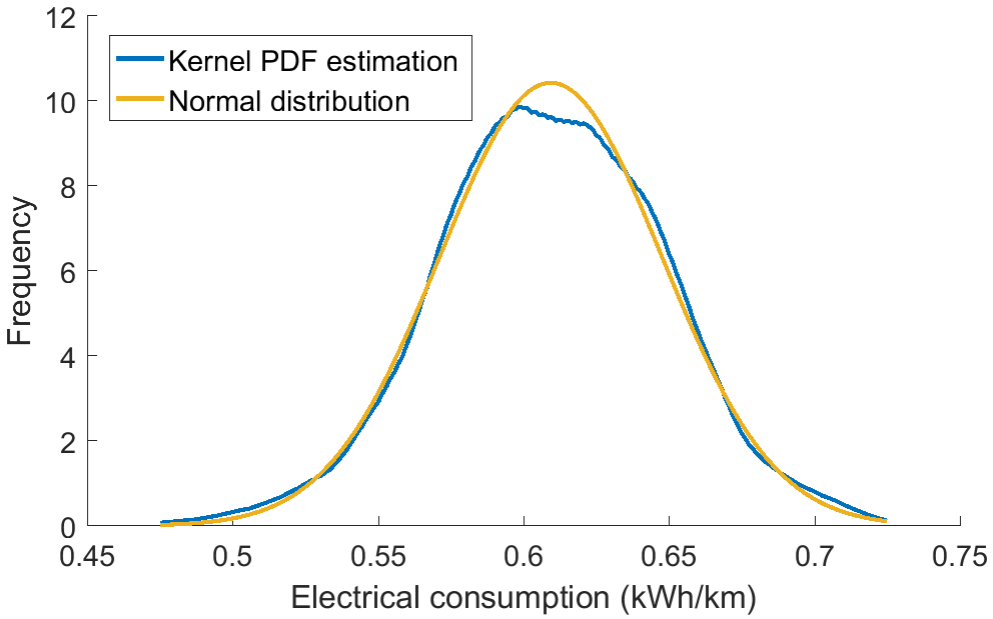

Figure 25. A comparison between the outcome kernel PDF estimation and a normal distribution.

4.2 Mean electric consumption by number of bus stops

The effect of the number of stops to the electrical consumption was also examined. As expected, the electrical consumption is directly proportional to the number of stops. The more stops the bus makes, the higher electrical consumption is. The electrical consumption in terms of the number of stops is presented in Figure 26. In the figure, the standard deviation at each bus stop is also presented.

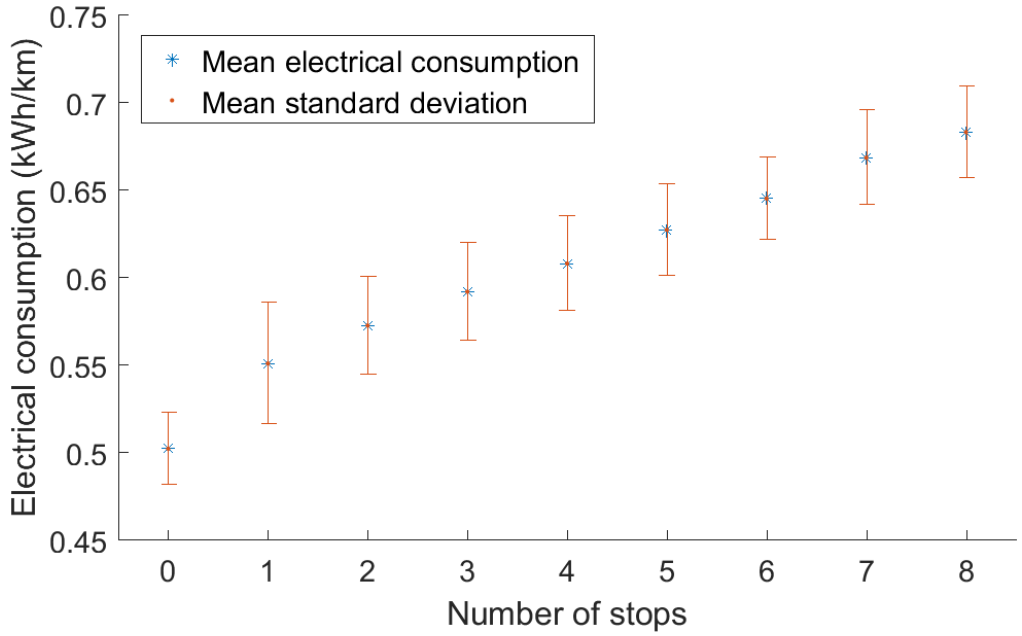

Figure 26. The mean electrical consumption by number of stops and mean standard deviations of each point.

Despite the standard deviations, a clear link between the number of stops and electrical consumption is witnessed. The deviation derives from the different speed profiles provided by the ten measured cycles and the speed characteristics of the cycle in the vicinity of the bus stop. For example, if the bus accelerates to a lower velocity after the bus stop, this would result in smaller deviation in speed, and thus, lower energy consumption. Driving cycles with this kind of bus stops would result in lower electrical consumptions. Furthermore, road characteristics, traffic and driver behavior also affect the speed profile in the driving cycles. In the next section, the effects of acceleration magnitudes are presented.

4.3 Impact of acceleration magnitude

The driver acceleration behavior was studied in terms of acceleration magnitudes. The impact of acceleration magnitude to the energy consumption was examined by comparing the positive average accelerations of different driving cycles. First, the driving cycles were divided into nine groups by the number of the stops made. Then, based on the average magnitude of acceleration at each number of stops, two groups were created. The first group includes driving cycles with more aggressive accelerations, cycles which had higher average positive acceleration compared to other driving cycles with the same amount of bus stops

were put in this group. Comparably, the second group was comprised of cycles with less than average accelerations.

Figure 27 illustrates how the acceleration magnitude affected the electrical consumptions in this study. The red circles represent driving cycles of the first group, in which, the accelerations were more aggressive than on average. Whereas, the green circles represent the less aggressive driving cycles of the second group. It is clear in the graphical representation that more aggressive driving cycles are more likely to result in higher energy consumptions. On average, the second group had roughly 5.6% smaller electrical consumption compared to the first group.

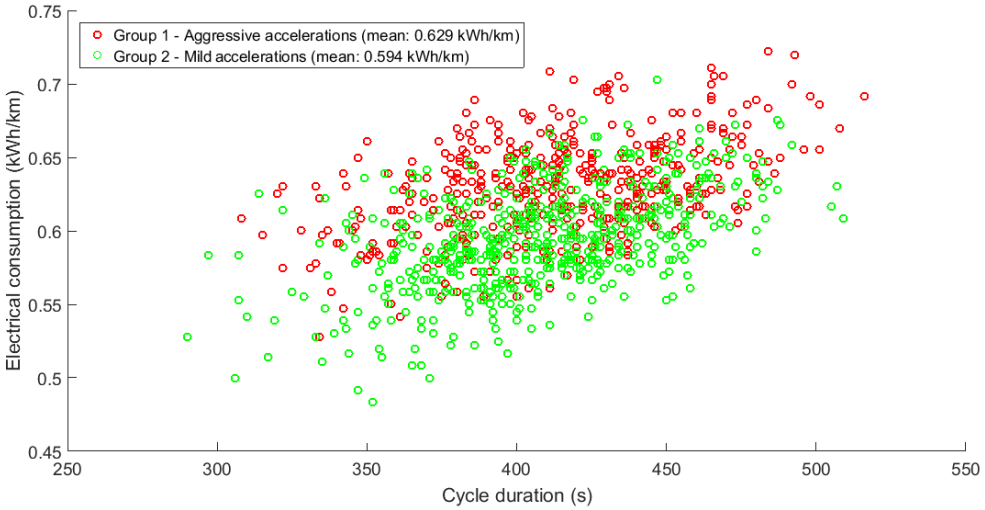

Figure 27. The effect of cycle duration and acceleration magnitudes on the electrical consumption. Red circles represent cycles with acceleration magnitudes more than average and green circles represent cycles with acceleration magnitudes less than average.

5 Discussion

This study utilized the Monte Carlo method to create an energy consumption sensitivity analysis based on the uncertainty in bus driving cycles. The Monte Carlo method provides useful information about the effects of the uncertain input in a process that entails uncertainty. When utilized correctly, the Monte Carlo method generates every possible outcome with a probability based on the uncertainty.

The Monte Carlo method results in a distribution of outcomes, which can be utilized to forecast future outcomes of the model. Adding confidence intervals to the results, provides a coherent, statistically justified statement of the likely results. This yields a more comprehensive and accurate description of the examined process when compared, for example, to a list of all possible outcomes or a single simulation which only presents one possible outcome.

The driving cycles were first synthesized with a Monte Carlo based algorithm and then simulated with an energy flow model of an electric city bus. As a result, an electrical consumption distribution, averages for electric consumption per number of stops and effects of acceleration amplitudes to the electrical consumption were acquired.

In this chapter, the conclusions are induced from the results. In addition, limitations entailed in the study are discussed and remarks for the future suggested.

5.1 Conclusions

The electrical consumption histogram illustrates the main results obtained with the Monte Carlo method, distribution of the outcomes. The distribution was further explicated with additional lines for mode, mean and confidence intervals. As a result, a descriptive forecast of the electrical consumption, based on the uncertainty considered in this study, was acquired. Based on the forecast, with an 80 percent probability the electrical consumption will be between 0.561 kWh/km and 0.642 kWh/km with an overall average of 0.609 kWh/km.

The accuracy of the Monte Carlo simulation increases with the number of points simulated, however, there is a compromise between the accuracy of the results and computation time. Hence, the optimal number of simulated points was evaluated. Based on the uncertain input of this study, at least 100 simulations were recommended in order to gain moderately accurate results and 450 simulations to gain accurate results. Nevertheless, the Monte Carlo simulation was performed with 1000 simulation points in this study, which resulted in eligible convergence. In conclusion, the number of simulations suggests that a decent accuracy was achieved with the simulations.

The results attained with the Monte Carlo simulations seemed to be in-line with the previous study [58], in which, the same vehicle and route were considered. The electrical consumption acquired in the previous study is included in the range of the outcomes of the Monte Carlo simulation performed in this study. In fact, the result of the previous study is close to the mode of the results.

Moreover, the benefits of statistical route sensitivity analysis are apparent when the electrical consumption variations of the route are scrutinized. It turned out that the difference between the minimum and maximum consumptions was 49.4%, while the sensitivity analysis expresses the electrical consumptions are most likely to vary with 14.4% deviation inside the confidence interval.

Two conclusions can be deducted from the results. First, this study demonstrates the impact of the number of bus stop to the electrical consumption. It is evident, that the number of bus stops has a notable effect on the consumption as the difference between the extremes is almost 50%. Hence, utilizing only one reference driving cycle in bus route consumption studies results in incomplete description of the possible energy consumptions in the route. It can be concluded that the route sensitivity is an important part of electric city bus electrical consumption analyses.

Second, the sensitivity analysis provides a descriptive forecast for the energy consumption. The distribution of possible consumptions offer useful data for electric bus system planning. In bus design parameter optimization, for example, if only one reference consumption was known for the route, higher margins would be required for the bus parameters. In order to make sure the bus performs reliably in the route, oversized design coefficients may be utilized. Likewise, it would be difficult to create valid electric bus route layouts if the magnitudes of energy consumption fluctuations were unknown. The results of this study indicate that there can be major differences between each run, thus, neglecting them can results in sub-optimal designs. To conclude, correctly utilized Monte Carlo simulation can be a promising tool in electric bus system planning as the tool provides the designers with encompassing prediction for the chosen parameter.

In addition to the electrical consumption distribution, energy consumption relation to the number of stops was studied. As expected, it was found out that the electrical consumption was proportional to the number of stops. On average, higher number of stops resulted in higher consumption. However, there was some deviation between the results, thus in some cases, it was possible that a higher consumption could be obtained with lower number of stops. Arguably driver acceleration behavior is one factor that interprets the deviations. Moreover, the simplifications in the auxiliary device modeling also play a role in the deviations.

The acceleration magnitude study provided insight to the driver acceleration behavior. The results were formed in groups for further analysis. First, the outcomes were first divided into nine groups based on the number of stops made during the driving cycle, and then, mean positive acceleration was calculated for each group. After, two separate groups were formed, group 1 with accelerations higher than average and group 2 with accelerations lower than average.

On average the energy consumptions in the group 1 were lower than in the group 2. Higher accelerations should result in higher consumptions as the electric motor efficiency drops at high current operations [62]. Although, on some occasions the electrical consumptions in group 1 were lower than in the group 2. This is most likely due to the division of the values, as the energy consumptions were scrutinized by the number of stops. As an example, an aggressive driving cycle with only one bus stop can result in lower energy consumption than

a run with light accelerations but many stops. All in all, the acceleration magnitude study suggests calm accelerations may result in consumptions.

In addition to the effects of the acceleration magnitude, the study revealed information about the effects of cycle duration to the electrical consumption. Longer runs tend to have higher energy consumptions per distance. This is probably due to two main reasons.

First, driving cycles with longer durations tend to have more stops in them. Increased number of stops results in more accelerations, which leads into higher energy consumptions. Driving cycles with shorter durations skip bus stops, thus, sparing the energy lost in the additional accelerations.

Second, the electrical consumption of the auxiliary devices was modeled with a constant power loss. In addition to the devices that are operated constantly during when driving the vehicle, such as power steering and climate control, some non-constant devices, such as the door control, were included in the auxiliary devices, This can result in inaccuracy in the energy flow model as driving cycles with slower speeds and longer durations lose more energy to the auxiliary devices than driving cycles with higher speeds.

All in all, the results of this dissertation suggest that route sensitivity analysis can be valuable in energy consumption studies. It was found out that the number of stops can have a significant effect on the distance specific consumption while the sensitivity analysis clarifies the distribution of consumptions. In addition, the other factors affect the consumptions even if the number of stops was kept constant. For example, the magnitude of acceleration was found to have an impact on the consumptions. Higher accelerations resulted in higher consumptions.

5.2 Limitations

Even though promising results were acquired in this study, certain limitations were present. First of all, only one uncertain input was sampled with the Monte Carlo method in this study. The only uncertain input, the number of stops, is not the only parameter which affects the energy consumption of the vehicle. For example, vehicle mass plays a role in the force required to accelerate an object. In this study, the mass of the vehicle was left constant, which is not the case with city buses in real life as the total mass changes every time passengers enter or leave the vehicle.

Secondly, the actual distribution of the number of bus stops was not known. Instead, the stops were randomly sampled from a normal distributed uncertainty PDF. In real life, it is possible that some bus stops are more frequently utilized than others. Ideally, the uncertainty PDF would be created based on the actual distribution of the stops. However, a generalization for the number of stops can be justified in some cases as it would require a great deal of effort to find out the stop distribution for every separate route. Additionally, the actual distribution might be unknown for new bus routes that have not been put to use. Nevertheless, it was not examined how accurately the normal distributed PDF modeled the bus stop distribution of the actual bus route. The uncertainty PDF affects the results as different uncertainty distributions will result in different outcome distributions with the Monte Carlo method.

Thirdly, the driver behavior was not comprehensively modeled, which limits the universality of the driving cycles. As only ten measurements of the route were performed, some possible speed profiles are inevitably left out from the driving cycles. Although, the limited number of measured driving cycles was addressed in the driving cycle synthetization by combining parts of the measured cycles. In addition, the measured driving cycles were based on both, ICE and electric driven vehicles. Even though, no striking differences were apparent in the driving cycles, some differences might be evident in closer examination of the driver behavior between ICE and electric driven city buses.

Fourthly, some simplifications were applied in the energy flow model and driving cycle models. For instance, the auxiliary devices were modeled as constant work loss blocks in the energy flow model. As a result, the auxiliary device energy consumption is comparatively higher in the slower cycles where run time is extended. Moreover, cabin heating devices were left out from the consideration, which results in lower consumptions. Additionally, the route was considered flat, only longitudinal movements and forces were considered and environmental factors, such as wind speed and direction, were left out of consideration.

Fifthly, the results are applicable only to the 3.3km part of the route. Other routes with different properties need to be examined separately. Although, it can be induced that the results generally apply to urban route with light traffic, the results may be different for routes with different amount of traffic.

5.3 Future remarks

This study pointed out the usefulness of the route sensitivity analysis in the electric bus energy consumption studies, such as electric bus system planning. The energy consumption forecast with confidence intervals indicate elaborately how much energy will be consumed in the examined route. The specifications of the bus, such as the battery capacity, can then be accurately optimized for the bus to uniquely match with the route. Likewise, the results can be utilized in electric bus route layout designing. For example, locations of charge stations can be placed based on the knowledge provided by the outcome distributions.

However, to attain a more accurate analysis of the route energy consumption sensitivity, more uncertain inputs are required in the Monte Carlo study. Two of the most important uncertain inputs would be a more accurate driver behavior sampling and vehicle mass sampling. In addition, analyzing the effects of traffic and other route features, such as traffic lights, would further expand the credibility of the route sensitivity analysis. More accurate forecasts would also require that the actual stop distributions were be defined and modeled accordingly.

Combining base cycles to synthesize additional driving cycles is a rather confined practice to model driver behavior. More accurate driver behavior modelling would require several additional measurements. One option would be to create a comprehensive algorithm to create completely artificial driving cycles from the route. In the fully artificial driving cycles, it would be more straightforward to define the magnitude of accelerations, top speeds and other speed profile characteristics. However, creating realistic driving cycles that precisely include all the route features is a challenging task and would require a study of its own.

Vehicle mass sampling requires different regenerative braking restrictions. Typically, the energy storage or the electric drive motor have limited input power which constrains the amount of energy regenerated. In this study, the regeneration energy was constrained by the deceleration and speed of the vehicle, which can be difficult to alter with different masses of the vehicle.

Traffic arguably affects the energy consumption, as traffic can create fluctuations to the driving cycles. There is no question that traffic is a complex factor and would require a separate study to find out its effects.

Lastly, the bus stoppings can be caused by other reasons than bus stops. For example, traffic lights and pedestrian crossings are typical reasons which result in stoppings. These factors could be modeled in similar fashion as the bus stops in this study. However, further study would be required as the stopping due to the traffic lights or pedestrian crossings is different to the bus stops. Traffic lights, for instance, tend to have longer space for braking which could result in longer decelerations with lower magnitudes. Additionally, the stopping time at the traffic lights could be longer compared to pedestrian crossings and bus stops.

References

[1] M. Rogge, S. Wollny, and D. Sauer, “Fast Charging Battery Buses for the Electrification of Urban Public Transport—A Feasibility Study Focusing on Charging Infrastructure and Energy Storage Requirements,” *Energies*, vol. 8, no. 5, pp. 4587–4606, May 2015.

[2] A. Lajunen, “Energy consumption and cost-benefit analysis of hybrid and electric city buses,” *Transportation Research Part C: Emerging Technologies*, vol. 38, pp. 1–15, Jan. 2014.

[3] G. J. Offer, D. Howey, M. Contestabile, R. Clague, and N. P. Brandon, “Comparative analysis of battery electric, hydrogen fuel cell and hybrid vehicles in a future sustainable road transport system,” *Energy Policy*, vol. 38, no. 1, pp. 24–29, 2010.

[4] National Renewable Energy Laboratory, “ADVISOR Advanced Vehicle Simulator,” 2013. [Online]. Available: http://adv-vehicle-sim.sourceforge.net/. [Accessed: 06-May-2016].

[5] Mechanical Simulation Corporation, “CarSim.” [Online]. Available: https://www.carsim.com/. [Accessed: 21-Apr-2016].

[6] C. G. Braun and D. Busse, “A Modular Simulink Model for Hybrid Electric Vehicles,” in *SAE Technical Paper*, 1996, no. 961659, p. 8.

[7] S. M. Aceves and R. J. Smith, “A Hybrid Vehicle Evaluation Code and Its Application to Vehicle Design,” in *SAE Technical Paper*, 1995, no. 950491, pp. 1–19.

[8] J. R. Bumby, P. H. Clarke, and I. Forster, “Computer modelling of the automotive energy requirements for internal combustion engine and battery electric-powered vehicles,” *IEE Proceedings A Physical Science, Measurement and Instrumentation, Management and Education, Reviews*, vol. 132, no. 5, p. 265, 1985.

[9] W. W. Marr and W. J. Walsh, “Life-cycle cost evaluations of electric/hybrid vehicles,” *Energy Conversion and Management*, vol. 33, no. 9, pp. 849–853, Sep. 1992.

[10] Argonne National Laboratory, “Autonomie.” [Online]. Available: http://www.autonomie.net/. [Accessed: 21-Apr-2016].

[11] Powersim, “PSIM Electronic Simulation Software.” [Online]. Available: https://powersimtech.com/. [Accessed: 06-May-2016].

[12] G. H. Cole, “SIMPLEV: A simple electric vehicle simulation program, Version 1.0,” Idaho Falls, ID, Jun. 1991.

[13] K. L. Butler, K. M. Stevens, and M. Ehsani, “A Versatile Computer Simulation Tool for Design and Analysis of Electric and Hybrid,” in *SAE Technical Paper*, 1997, no. 970199, pp. 19–25.

[14] University of South Carolina, “Virtual Test Bed.” [Online]. Available: http://vtb.engr.sc.edu/. [Accessed: 06-May-2016].

[15] T. Halmeaho, P. Rahkola, J. Pippuri, and K. Tammi, “Hybrid city bus design evaluation using system level simulations,” in *IEEE 23rd International Symposium*

on Industrial Electronics (ISIE), 2014, pp. 1671–1676.

[16] T. Halmeaho, P. Rahkola, A.-P. Pellikka, S. Ruotsalainen, and K. Tammi, "Electric City Bus Energy Flow Model and Its Validation by Dynamometer Test," in *IEEE Vehicle Power and Propulsion Conference (VPPC)*, 2015, pp. 1–6.

[17] P. Bubna, S. G. Advani, and A. K. Prasad, "Integration of batteries with ultracapacitors for a fuel cell hybrid transit bus," *Journal of Power Sources*, vol. 199, pp. 360–366, Feb. 2011.

[18] P. Bubna, D. Brunner, J. J. Gangloff, S. G. Advani, and A. K. Prasad, "Analysis, operation and maintenance of a fuel cell/battery series-hybrid bus for urban transit applications," *Journal of Power Sources*, vol. 195, no. 12, pp. 3939–3949, 2009.

[19] R. Ahluwalia, X. Wang, and R. Kumar, "Fuel Cell Transit Buses," *Report of the Argonne National Laboratory Argonne*, pp. 1–12, 2012.

[20] K. L. Butler, M. Ehsani, and P. Kamath, "A Matlab-based modeling and simulation package for electric and hybrid electric vehicle design," *IEEE Transactions on Vehicular Technology*, vol. 48, no. 6, pp. 1770–1778, 1999.

[21] T. Markel, A. Brooker, T. Hendricks, V. Johnson, K. Kelly, B. Kramer, M. O'Keefe, S. Sprik, and K. Wipke, "ADVISOR: a systems analysis tool for advanced vehicle modeling," *Journal of Power Sources*, vol. 110, no. 2, pp. 255–266, Aug. 2002.

[22] M. Panagiotidis, G. Delagrammatikas, and D. N. Assanis, "Development and Use of a Regenerative Braking Model for a Parallel Hybrid Electric Vehicle," in *SAE Technical Paper*, 2000, no. 2000–01–0995, p. 14.

[23] M. Amrhein and P. T. Krein, "Dynamic Simulation for Analysis of Hybrid Electric Vehicle System and Subsystem Interactions, Including Power Electronics," *IEEE Transactions on Vehicular Technology*, vol. 54, no. 3, pp. 825–836, May 2005.

[24] C. Lin, Z. Filipi, Y. Wang, L. Louca, H. Peng, D. N. Assanis, and J. Stein, "Integrated, Feed-Forward Hybrid Electric Vehicle Simulation in SIMULINK and its Use for Power Management Studies," in *SAE Technical Paper*, 2001, no. 2001–01–1334, p. 13.

[25] D. W. Gao, C. Mi, and A. Emadi, "Modeling and Simulation of Electric and Hybrid Vehicles," *Proceedings of the IEEE*, vol. 95, no. 4, pp. 729–745, Apr. 2007.

[26] B. K. Powell and K. E. Bailey, "Dynamic modeling and control of hybrid electric vehicle powertrain systems," *IEEE Control Systems Magazine*, vol. 18, no. 5, pp. 17–33, 1998.

[27] N. Jinrui, W. Zhifu, and R. Qinglian, "Simulation and Analysis of Performance of a Pure Electric Vehicle with a Super-capacitor," in *IEEE Vehicle Power and Propulsion Conference (VPPC)*, 2006, pp. 1–6.

[28] R. D. Senger, M. A. Merkle, and D. J. Nelson, "Validation of ADVISOR as a Simulation Tool for a Series Hybrid Electric Vehicle," in *SAE Technical Paper*, 1998, no. 981133, pp. 95–115.

[29] G. Rizzoni, L. Guzzella, and B. M. Baumann, "Unified modeling of hybrid electric vehicle drivetrains," *IEEE/ASME Transactions on Mechatronics*, vol. 4, no. 3, pp. 246–257, 1999.

[30] B. H. Wang, Y. G. Luo, and J. W. Zhang, “Simulation of city bus performance based on actual urban driving cycle in China,” *International Journal of Automotive Technology*, vol. 9, no. 4, pp. 501–507, Aug. 2008.

[31] MathWorks, “MATLAB.” [Online]. Available: http://se.mathworks.com/products/matlab/. [Accessed: 06-May-2016].

[32] United Nations Inland Transport Committee, “Consolidated Resolution on the Construction of Vehicles (R.E.3),” 2016.

[33] M. Ehsani, Y. Gao, S. Gay, and A. Emadi, *Modern Electric, Hybrid Electric, and Fuel Cell Vehicles*, vol. 6. CRC Press, 2004.

[34] B. Scrosati and J. Garche, “Lithium batteries: Status, prospects and future,” *Journal of Power Sources*, vol. 195, no. 9, pp. 2419–2430, 2010.

[35] A. Lajunen, “Improving the Energy Efficiency and Operating Performance of Heavy Vehicles by Powertrain Electrification,” Doctoral dissertation, Aalto University. School of Engineering. Helsinki: Unigrafia Oy. p. 51, 2014.

[36] Proterra, “The proterra® Catalyst Platform®.” [Online]. Available: http://www.proterra.com/wp-content/uploads/2015/05/Tearsheets_CatalystPlatform.pdf. [Accessed: 05-May-2016].

[37] K. Bullis, “Wheel Motors to Drive Dutch Buses,” *MIT Technology Review*, 2009. [Online]. Available: https://www.technologyreview.com/s/412656/wheel-motors-to-drive-dutch-buses/. [Accessed: 05-May-2016].

[38] C. C. Chan, A. Bouscayrol, and K. Chen, “Electric, hybrid, and fuel-cell vehicles: Architectures and modeling,” *IEEE Transactions on Vehicular Technology*, vol. 59, no. 2, pp. 589–598, 2010.

[39] A. Rousseau, P. Sharer, and F. Besnier, “Feasibility of Reusable Vehicle Modeling: Application to Hybrid Vehicles,” in *SAE Technical Paper*, 2004, no. 2004–01–1618, p. 12.

[40] R. Trigui, M. Desbois-Renaudin, B. Jeanneret, and F. Badin, “Global Forward-Backward Approach for a Systematic Analysis and Implementation of Hybrid Vehicle Management Laws. Application to a Two Clutches Parallel Hybrid Power Train.,” in *European Ele-Drive Conference*, 2004, p. 12.

[41] L. Horrein, A. Bouscayrol, P. Delarue, J. N. Verhille, and C. Mayet, “Forward and Backward simulations of a power propulsion system,” *IFAC Proceedings Volumes*, vol. 45, no. 21, pp. 441–446, Sep. 2012.

[42] K. B. Wipke, M. R. Cuddy, and S. D. Burch, “ADVISOR 2.1: A user-friendly advanced powertrain simulation using a combined backward/forward approach,” *IEEE Transactions on Vehicular Technology*, vol. 48, no. 6, pp. 1751–1761, 1999.

[43] Y. Iwasaki and H. A. Simon, “Causality and model abstraction,” *Artificial Intelligence*, vol. 67, no. 1, pp. 143–194, 1994.

[44] R. V.Gopal and A. Rousseau, “System Analysis Using Multiple Expert Tools,” in *SAE Technical Paper*, 2011, no. 2011–01–0754, p. 10.

[45] I. Lux and L. Koblinger, *Particle Transport Methods : Neutron and Photon Calculations Authors*. Boca Raton, FL: CRC Press, 1991.

[46] J. P. Biersack and L. G. Haggmark, “A Monte Carlo computer program for the transport of energetic ions in amorphous targets,” *Nuclear Instruments and Methods*, vol. 174, no. 1, pp. 257–269, 1980.

[47] L. Wang, S. L. Jacques, and L. Zheng, “MCML—Monte Carlo modeling of light transport in multi-layered tissues,” *Computer Methods and Programs in Biomedicine*, vol. 47, no. 2, pp. 131–146, Jul. 1995.

[48] T. Carsey and J. Harden, *Monte Carlo Simulation and Resampling Methods for Social Science*. Thousand Oaks, CA: SAGE Publications, 2014.

[49] R. E. Livezey and W. Y. Chen, “Statistical Field Significance and its Determination by Monte Carlo Techniques,” *Monthly Weather Review*, vol. 111, no. 1, pp. 46–59, Jan. 1983.

[50] G. Faron, S. Pagerit, and A. Rousseau, “Evaluation of PHEVs fuel efficiency and cost using Monte Carlo analysis,” in *EVS 24 Conference*, 2009, pp. 1–13.

[51] W. K. Hastings, “Monte Carlo Sampling Methods Using Markov Chains and Their Applications,” *Biometrika*, vol. 57, no. 1, pp. 97–109, 1970.

[52] C. Z. Mooney, *Monte Carlo Simulation*. Springfield, USA: SAGE Publications, 1997.

[53] R. L. Harrison, C. Granja, and C. Leroy, “Introduction to Monte Carlo Simulation,” in *AIP conference proceedings*, 2010, vol. 1204, pp. 17–21.

[54] E. Kentel and M. M. Aral, “2D Monte Carlo versus 2D Fuzzy Monte Carlo health risk assessment,” *Stochastic Environmental Research and Risk Assessment*, vol. 19, no. 1, pp. 86–96, Feb. 2005.

[55] D. J. Earl and M. W. Deem, “Monte Carlo Simulations,” *Methods in Molecular Biology*, vol. 443, pp. 25–36, 2006.

[56] VBOX Automotive, “VBOX Mini 10Hz GPS Data Logger (RLVBM01).” [Online]. Available: http://www.racelogic.co.uk/_downloads/vbox/Datasheets/Data_Loggers/RLVBM01_DATA.pdf. [Accessed: 07-Jul-2016].

[57] L. L. Hoberock, “A survey of longitudinal acceleration comfort studies in ground transportation vehicles,” *Journal of Dynamic Systems, Measurement, and Control*, vol. 99, no. 2, pp. 76–84, 1977.

[58] T. Halmeaho, P. Rahkola, K. Tammi, J. Pippuri, A.-P. Pellikka, A. Manninen, and S. Ruotsalainen, “Experimental validation of electric bus powertrain model under city driving cycles,” *IET Electrical Systems in Transportation*, Sep. 2016.

[59] Argonne National Laboratory, “Autonomie - Vehicle modeling approaches.” [Online]. Available: http://www.autonomie.net/references/vehicle_mods_25.html. [Accessed: 11-Jul-2016].

[60] K. Cranmer, “Kernel estimation in high-energy physics,” *Computer Physics Communications*, vol. 136, no. 3, pp. 198–207, May 2001.

[61] MathWorks Nordic, “Kernel Distribution - MATLAB & Simulink.” [Online]. Available: https://se.mathworks.com/help/stats/kernel-distribution.html. [Accessed: 02-Nov-2016].

[62] X. Wu, D. Freese, A. Cabrera, and W. A. Kitch, “Electric vehicles’ energy consumption measurement and estimation,” *Transportation Research Part D: Transport and Environment*, vol. 34, pp. 52–67, Jan. 2015.

Appendix

Appendix A. Matlab script used to synthesize new driving cycles
Appendix B. Matlab function used to skip a stop in the driving cycle
Appendix C. Matlab function used to create a stop in the driving cycle

Appendix A. Matlab script used to synthesize new driving cycles

Options and initial conditions

```
kys0='Luetaanko pohjatiedot 1=y 0=n \n';
kys1='Kuinka monta vaihtelua tehdään? \n';
kys3='Skaalataanko ajot? 1=y 0=n \n';
kys4='Viedäänkö matkat Autonomieen? 1=y 0=n \n';

luetaanko =0;
simulaatiot=input(kys1);

if luetaanko ==1
    load('ft_workspace_testi8_pohja2222.mat')
end

kys2=input('Pistetäänkö pysähdykset randomilla? 1=y 0=n \n');

random_pysahdykset=kys2;
kaikki_pysahdykset = 0; %pysähdytäänkö kaikilla pysäkeillä?
autonomieen =0;% input(kys4);
yhdistely_paalla=0; %yhdistelläänkö eri ajoja pysäkkiväleittäin? 1=y 0=n
%jos yhdistely pois päältä, tällä valitaan haluttu ajo:
ajovalinta_ei_yhdist = 3;

kmh_testiajo8={};
kmh_pohja={};
pysahtymiset_kok=0;
pysahtymiset_yhdist=0;
pysahdys_pysakit={};
matkat_yhdist=0;
matka_yhdist=0;
matkat_pysakeille=0;
matkat_pysakeille_kohta=0;
kmh_patka={};
kmh_testiajo8_alkup={};
pysahdykset_kok=0;

kiiht_yhdist_yli_011g_aika=zeros(1,simulaatiot);

kiiht_yhdist=zeros(1,simulaatiot);

ajot_testi8={};
valinnat_pysakki={};

for u = 1:10
figure(500+u);clf;hold on;
xlabel('Time (s)');
ylabel('Speed (km/h)');
tiitteli=sprintf('Route 11 - 3.36km part speed profile %.0f with bus stop
markers',u);
title(tiitteli);
end
```

```
for zz=1:simulaatiot
ajo =randi([1 10]);

if yhdistely_paalla==0
    %close(501:510)
    ajo=ajovalinta_ei_yhdist;
    %figure(500+ajo);hold on;
end

%figure(500+ajo);hold on;

aika_pysakki=0;
clear kmh_testiajo8{ajo,zz};
aika_pysakki_testi8=aika_pysakki;

aika_pysakki_pohja2= [...
    90,151,200,275,305,342,415,440;...
    90,137,168,220,265,306,371,425;...
    92,136,187,245,289,355,389,451;...
    92,136,179,239,275,317,374,405;...
    90,139,180,253,300,332,410,440;...
    89,137,165,232,295,360,412,473;...
    90,156,204,256,280,316,346,386;...
    90,148,196,263,300,342,415,457;...
    95,177,227,268,313,346,416,465;...
    90,143,180,255,295,335,399,449];

aika_pysakki_testi8=aika_pysakki_pohja;

aika_pysakki_testi8_alkup=aika_pysakki_testi8; %tarvitaan kun yhdistellään
ajoja
%tehdään niin, että aluksi pistetään ajo pysähtymään kaikilla pysähdyksillä
%sitten normaalijakauman todennäköisyyksien mukaan määritellään 1x8 matriisi,
%jossa 0 = ei pysähdy ja 1 = pysähtyy
%eli esim [1 0 0 0 0 0 0 0] pysähtyy vain ensimmäisellä pysäkillä
%pysahdys_pysakit arvoilla ajo PYSÄHTYY JOKAISELLA PYSÄKILLÄ
    pysahdys_pysakit{1,1}(1:8) = [1 0 1 1 1 0 1 1];
    pysahdys_pysakit{2,1}(1:8) = [1 1 0 1 1 0 0 0];
    pysahdys_pysakit{3,1}(1:8) = [1 1 1 1 0 1 0 0];
    pysahdys_pysakit{4,1}(1:8) = [1 1 0 1 1 0 1 0];
    pysahdys_pysakit{5,1}(1:8) = [1 1 0 0 1 1 1 0];
    pysahdys_pysakit{6,1}(1:8) = [1 1 0 1 1 1 0 0];
    pysahdys_pysakit{7,1}(1:8) = [1 0 0 1 1 1 1 1];
    pysahdys_pysakit{8,1}(1:8) = [1 0 0 1 1 0 0 0];
    pysahdys_pysakit{9,1}(1:8) = [0 0 1 1 1 1 0 0];
    pysahdys_pysakit{10,1}(1:8) = [1 1 0 1 1 0 0 0];

%otetaan ylempi pohjaksi/alustusarvoiksi:
pys_rnd_alustus = pysahdys_pysakit{ajo,1};

if kaikki_pysahdykset ==0

 %---%---%---%---%---%---%---%---%---%---%---%---%---%---%---%---%---%---%
```

```
%                                                                       %
%                                                                       %
```

```
%  %tämä on se 1x8 matriisi, jolla pysähtymiset voi itse määrätä,       %
%  %0=eipysähdy 1=pysähtyy                                              %
   pys_rnd_syote = [1 1 0 0 0 0 0 0];                                   %
%                                                                       %
%                                                                       %
%---%---%---%---%---%---%---%---%---%---%---%---%---%---%---%---%---%---%


    if random_pysahdykset == 1 %eli jos randomoidaan pysähdykset
        pys_rnd_syote = [randi([0 1]) randi([0 1]) randi([0 1]) randi([0
1])...
            randi([0 1]) randi([0 1]) randi([0 1]) randi([0 1])];
    end

    for u = 1:size(pys_rnd_syote,2)
        %jos syöte = 0, ajetaan pysäkin ohitse (eli muutetaan 1->0 tai 0->1
        %pys_rnd_alustus:ssa)
        if pys_rnd_syote(u) == 0
            if pys_rnd_alustus(u)==0
                pys_rnd_alustus(u)=1;
            else
                pys_rnd_alustus(u)=0;
            end
        end
        %jos syöte = 1, pysähdytään pysäkille (pidetään arvo pys_rnd_alus-
tus:ssa)
    end
end

%alustusarvoista tulee ns. oikeita arvoja:
pys_rnd = pys_rnd_alustus;
```

Base driving cycle definition

```
%määritetään kmh_testiajo8 jokaisella kierroksella "alkuperäiseksi" (eli
filtkmh_forceksi)
for uu = 1:10
    for j = 1:loppupiste_uus(uu) %size(filtkmh_force{1,uu},2)
        kmh_testiajo8{uu,zz}(j)=filtkmh_force{1,uu}(j);
        kmh_testiajo8_alkup{uu,zz}(j)=filtkmh_force{1,uu}(j);
    end
end

aika_pysakki_testi8=aika_pysakki_pohja;
 [kmh_testiajo8, aika_pysakki_testi8, aikaero_pohja] = ajoprofiilin_muodos-
tus...
    (pys_rnd, ajo, alkupiste_uus, loppupiste_uus, filtkmh_force, ...
    kmh_testiajo8, aika_pysakki_testi8, zz);
%Yhdistelyssä otetaan ensin satunnaisesti jokin kymmenestä ajosta pohja-
ajoksi,
%ja satunnoidaan pysäkit. Vertaamalla pys_rnd ja pysahdys_pysakit saadaan
%tietää millä pysäkeillä pohja-ajo pysähtyy. Sen jälkeen  satunnoidaan 9
%ajoa samoilla pysäkeillä kuin pohja-ajo (voi olla sama pohja-ajo uudestaan)
```

```
%jonka jälkeen leikataan pohja-ajo pysähdyksittäin katki (eli kun kmh=0).
%Yhdistetään pätkät järjestyksessä, jolloin saadaan yhdistelty ajo

kmh_pohja22={}; %väliaikainen, kmh_pohja = lopullinen
%muunnospohja eri ajoille
%määritetään kmh_testiajo8 jokaisella kierroksella "alkuperäiseksi" (eli
filtkmh_forceksi)
for j = 1:size(kmh_testiajo8{ajo,zz},2)
    kmh_pohja22{ajo,zz}(j)=kmh_testiajo8{ajo,zz}(j);
end

%poistetaan alkupistehuomiointi pois pohja-ajosta (eli se alkaa nyt pisteestä
1
for j = alkupiste_uus(ajo):size(kmh_pohja22{ajo,zz},2)
    kmh_pohja{ajo,zz}(j-alkupiste_uus(ajo)+1)=kmh_pohja22{ajo,zz}(j);
end
clear kmh_pohja22

%Vertaillaan pys_rnd ja pysahdys_pysakit, jotta tiedetään millä pysäkeillä
%pysähdyttiin, 1=pysähdys
for u = 1:8
    if pys_rnd(u) == pysahdys_pysakit{ajo,1}(u)
        pysahtymiset_yhdist(u,zz) = 1;
    end
    if pys_rnd(u) ~= pysahdys_pysakit{ajo,1}(u)
        pysahtymiset_yhdist(u,zz) = 0;
    end
end

%pysähdysten määrä
pysahdysten_maara = 0;
for u = 1:size(pys_rnd,2)
    if pysahtymiset_yhdist(u,zz)==1
        pysahdysten_maara=pysahdysten_maara+1;
    end
end

%pysäkkien numerot joissa pysähdytään:
pys_pysahd=0;
nn=1;
for u = 1:8
    if pysahtymiset_yhdist(u,zz)==1
        pys_pysahd(nn)=u;
        nn=nn+1;
    end
end
```

Defining additional driving cycles

```
pysahdys_uusi=0;

pysahdys_pohja=0;

pys_rnd_uusi2=0;
```

```
pys_rnd_uusi=0;

pysahtymiset_uusi=0;

%koska ensimmäinen ajo on jo tehty aiemmin, tarvitaan pysäkkien verran
%lisää pätkiä muista ajoista

pysahdykset_kok(zz) = pysahdysten_maara;

kulperi=1;
if kulperi ==1
for uusi_nro = 1:pysahdysten_maara
    ajo_uusi = randi([1 10]);

    if yhdistely_paalla ~= 1
        ajo_uusi = ajo;
    end

    %pys_rnd pitää olla niin, että uudessa ajossa pysähdykset oikeissa koh-
dissa
    %laitetaan aluksi pys_rnd niin, että pysähtyy jokaisessa pysäkissä

    pys_rnd_uusi(1:8) = pysahdys_pysakit{ajo_uusi,1}; %näillä pysähtyy kai-
kissa
    pys_rnd_uusi2(1:8) = pysahdys_pysakit{ajo_uusi,1};
    pysahtymiset_uusi(1:8) = 1; %kaikki =1 koska pysähtyy kaikissa

%jos pohja-ajossa ei pysähdytä, pitää pys_rnd muuttaa myös niin, ettei sekään
pysähdy
    for bb = 1:8
        if pysahtymiset_uusi(bb) ~= pysahtymiset_yhdist(bb,zz)
            if pys_rnd_uusi(bb) == 0
                pys_rnd_uusi2(bb)=1;
            end
            if pys_rnd_uusi(bb) == 1
                pys_rnd_uusi2(bb)=0;
            end
        end
    end

    %muodostetaan uusi ajoprofiili samoilla pysähdyksillä
    [kmh_testiajo8_uusi22, aika_pysakki_testi8_uusi, aikaero_uusi] = ...
        ajoprofiilin_muodostus(pys_rnd_uusi2, ajo_uusi, alkupiste_uus, ...
        loppupiste_uus, filtkmh_force, kmh_testiajo8_alkup, ...
        aika_pysakki_testi8_alkup, zz);

    %poistetaan uudesta ajoprofiilista alkupistesekoilu:
    kmh_testiajo8_uusi22; %väliaikaisprofiili
    kmh_testiajo8_uusi={};
    clear kmh_testiajo8_uusi
    for j = alkupiste_uus(ajo_uusi): size(kmh_testiajo8_uusi22{ajo_uusi,zz},2)
        kmh_testiajo8_uusi{ajo_uusi,zz}(j-alkupiste_uus(ajo_uusi)+1)=...
            kmh_testiajo8_uusi22{ajo_uusi,zz}(j);
    end
    clear kmh_testiajo8_uusi22
```

```
    uuusi(uusi_nro)=plot((0:0.1:(size(kmh_testiajo8_uusi{ajo_uusi,zz},2)...
        -1)/10), kmh_testiajo8_uusi{ajo_uusi,zz}(1:end).','--','linewidth',1);

    teksti_uusi = sprintf('Ajo %.0f',ajo_uusi);
    legend(uuusi(uusi_nro), teksti_uusi);

    %katsotaan missä kohdassa pohjaprofiili menee nollaan ja leikataan..
    %bb lähtee järjestyksessä ensin ekalta pysähdetyltä pysäkiltä, ja
    %sitten tokalta yms..

    if uusi_nro==1
        pys_alku=1;
    else
        %tässä lasketaan kohta, missä pysäkki loppuu (eli pysäkin jälkeinen
        %piste missä nopeus > 5 km/h)
        %ilman tätä, pysyttäisiin aina samalla ensimmäisellä pysäkillä
        pys_alku = round(pysahdys_pohja(uusi_nro-1))+1;
        for bb = pys_alku : size(kmh_pohja{ajo,zz},2)
            if kmh_pohja{ajo,zz}(bb) > 5
                pys_alku = bb+3;
                break
            end
        end
    end

    %tässä lasketaan pohja-ajon pysähtymiskohta (jonka perään uusi ajo liite-
tään)
    %pys_alku laskennassa (yllä) varmistetaan, että pysähtymiskohta on aina
    %seuraavasta pysäkistä
    for bb = pys_alku : size(kmh_pohja{ajo,zz},2)
        if kmh_pohja{ajo,zz}(bb) < 0.1
            pysahdys_pohja(uusi_nro) = bb+3;
            break
        end
    end

    %jos kyseessä viimeinen pysäkki, katsotaan pysähdyskohta taaksepäin,
    %eli loppukohdasta siihen kohtaan missä nopeus < 0.5 kmh
    if pys_pysahd(uusi_nro)==9
        for bb = 1: size(kmh_pohja{ajo,zz},2)
            if kmh_pohja{ajo,zz}(size(kmh_pohja{ajo,zz},2)+1-bb) < 0.1
                break
            end
        end
        for bbb = 1 : size(kmh_pohja{ajo,zz},2)-bb
            if kmh_pohja{ajo,zz}((size(kmh_pohja{ajo,zz},2)-bb)-bbb) > 0.1
                %(size(kmh_pohja{ajo,zz},2)+1-bbb);
                %nnnoppeus=kmh_pohja{ajo,zz}((size(kmh_pohja{ajo,zz},2)-bb)-
bbb);
                pysahdys_pohja(uusi_nro) = ((size(kmh_pohja{ajo,zz},2)-bb)-
bbb)+3;
                break
            end
        end
```

```
    end

    %missä uusi ajoprofiili pysähtyy
    ennen_pys = 120; %ennen_pys = kuinka monta pistettä ennen pysäkkiviivaa
    %aloitetaan nopeuden <0.1 etsiminen
    if pys_pysahd(uusi_nro)==1
        ennen_pys = 80; %jos kyseessä ensimmäinen pysäkki, ennen_pys pitää
        %olla <89 koska ekat pysäkit aikaisintaan 90 kohdalla
        %jos olisi >90 niin <0.1kmh katsominen aloitettaisiin ennen
        %nopeusprofiilia -> error
    end
    for bb = round((aika_pysakki_testi8_uusi(ajo_uusi,pys_pysahd(uusi_nro))...
            -alkupiste_uus(8)/10)*10 -ennen_pys) : size(kmh_testiajo8_uusi...
            {ajo_uusi,zz},2)
        if kmh_testiajo8_uusi{ajo_uusi,zz}(bb) < 0.1
            pysahdys_uusi(uusi_nro) = bb+3;
            break
        end
    end

    %uuden ajon pätkät
    %pätkä alusta ekaan, tokaan, kolkkiin yms. pysäkkiin
    kmh_patka{zz,uusi_nro} = kmh_testiajo8_uusi{ajo_uusi,zz}(alkupiste_uus:...
        aika_pysakki_testi8_uusi(uusi_nro));
    %pätkien matkat
    matka_patka=0;
    for y = 1:size(kmh_patka{zz,uusi_nro},2)
        matka_patka(y+1)=matka_patka(y) + (kmh_patka{zz,uusi_nro}(y)/3.6)*0.1;
    end
    matkat_patkat{zz,1}(uusi_nro)=round(matka_patka(end));

    %leikataan pohja-ajo pysäkin 1.1s pysähdyskohdan jälkeen
    %väliaikainen muokkaamiseen tarvittu profiili
    kmh_pohja22{ajo,zz} = kmh_pohja{ajo,zz};
    clear kmh_pohja
    for j = 1 : pysahdys_pohja(uusi_nro) +5 %1.1s pysähdyskohdan jälkeen
        kmh_pohja{ajo,zz}(j) = kmh_pohja22{ajo,zz}(j);
    end
    clear kmh_pohja22

  %yhdistetään uusi ajo pohjaan:
    for bb = pysahdys_pohja(uusi_nro) : size(kmh_testiajo8_uusi{ajo_uusi,...
            zz},2)-(pysahdys_uusi(uusi_nro)-pysahdys_pohja(uusi_nro))
        kmh_pohja{ajo,zz}(bb) = kmh_testiajo8_uusi{ajo_uusi,zz}(bb+...
            (pysahdys_uusi(uusi_nro)-pysahdys_pohja(uusi_nro)));
    end

end
end
```

Distance computation

```
%uuden pätkän matka
matka_testiajo8=0;
for y = alkupiste_uus(ajo):size(kmh_testiajo8{ajo,zz},2)
```

```
    matka_testiajo8(y+2-alkupiste_uus(ajo))=matka_testiajo8(y+1-...
        alkupiste_uus(ajo)) + (kmh_testiajo8{ajo,zz}(y)/3.6)*0.1;
```

```
end

matkat_testiajo8{ajo,1}(zz)=matka_testiajo8(end);

%muiden alkuperäisten ajojen matkat
matka_alkup={};
clear  matka_alkup
for muut_ajot=1:10
    matka_alkup{1,muut_ajot}(1)=0;
    for y=alkupiste_uus(muut_ajot):loppupiste_uus(muut_ajot)
        matka_alkup{1,muut_ajot}(y+2-alkupiste_uus(muut_ajot))=...
            matka_alkup{1,muut_ajot}(y+1-alkupiste_uus(muut_ajot)) +...
            (filtkmh_force{1,muut_ajot}(y)/3.6)*0.1;
    end
end

matka_yhdist=0;
%yhdistettyjen ajojen matkat
for y = 1:size(kmh_pohja{ajo,zz},2)
     matka_yhdist(y+1)=matka_yhdist(y) + (kmh_pohja{ajo,zz}(y)/3.6)*0.1;
end
matkat_yhdist(zz)=round(matka_yhdist(end));

%matkat pysäkeille
if yhdistely_paalla ==1
    matka_pysakille=0;
    for y = alkupiste_uus(ajo) : aika_pysakki_testi8(ajo,1)*10 -...
            alkupiste_uus(8) + alkupiste_uus(ajo)
        matka_pysakille(y+2-alkupiste_uus(ajo))=matka_pysakille(y+1-...
            alkupiste_uus(ajo)) + (kmh_testiajo8{ajo,zz}(y)/3.6)*0.1;
        if matka_pysakille(y+2-alkupiste_uus(ajo)) - 64.78 >= 0.5
            matkat_pysakeille_kohta(ajo,1) = y;
            break
        end
    end
    matkat_pysakeille(ajo,1) = matka_pysakille(end);
end
```

Acceleration computation

```
%yhdistetyn ajon kiihtyvyys
for y = 1 : size(kmh_pohja{ajo,zz},2)-1
    kiiht_yhdist(y,zz)=(kmh_pohja{ajo,zz}(y+1)-kmh_pohja{ajo,zz}(y))/3.6/0.1;
end

%monta sadasosaa ajetaan yli 0.11*g kiihtyvyydellä
for y = 1 : size(kiiht_yhdist(1:end,zz),1)
    if kiiht_yhdist(y,zz) >= 0.11*9.81
        kiiht_yhdist_yli_011g_aika(1,zz)=kiiht_yhdist_yli_011g_aika(1,zz)+0.1;
    end
end
```

Import to Autonomie

```
%lataa suoraan käsittelyssä olevan ajon autonomievalmiiksi tiedostoksi
if autonomieen ==1
    sch_cycle=0;

    clear sch_cycle;
    n=0;
    for u=1:size(kmh_pohja{ajo,zz},2)
        n=n+1;
        sch_cycle(n,2)=kmh_pohja{ajo,zz}(u)/3.6;
        sch_cycle(n,1)=(n+9)/10-1;
    end
    nimi=sprintf('Cycle_yhdist_nro%.0f',zz);

    sch_grade =  [0 0; size(kmh_pohja{ajo,zz},2) 0];
    sch_key_on = [0 1; size(kmh_pohja{ajo,zz},2) 1];
    sch_metadata.name = nimi;
    sch_metadata.proprietary = 'public';

    save (nimi,'sch_grade','sch_key_on','sch_cycle','sch_metadata')
end

    ladattu_prosentit=sprintf('Ladataan: %.0f %%',(zz*100/simulaatiot));
    disp(ladattu_prosentit);
end
```

Plotting

```
%muut ajot kuin käsiteltävä (piirto)
for muut_ajot=[1:10]
    m(muut_ajot)=plot(seconds(aika(alkupiste_uus(muut_ajot):...
        loppupiste_uus(muut_ajot)))-round(alkupiste_uus(muut_ajot)/10)+...
        0.1,filtkmh_force{1,muut_ajot}(alkupiste_uus(muut_ajot):...
        loppupiste_uus(muut_ajot)));

    teksti = sprintf('Cycle %.0f. Distance: %.0f m',muut_ajot,...
        matka_alkup{1,muut_ajot}(end));
    legend([m(muut_ajot)], teksti)
    for aaa=1:8
        p(muut_ajot,aaa)=plot([aika_pysakki_testi8(muut_ajot,aaa)-...
            alkupiste_uus(8) aika_pysakki_testi8(muut_ajot,aaa)-...
            alkupiste_uus(8)],[0 60],'r');
        teksti = sprintf('Cycle %.0f stop line, stop %.0f',muut_ajot,aaa);
        legend([p(muut_ajot,aaa)], teksti)
    end
end

%matkatiedot muille ajoille
 for muut_ajot=2:simulaatiot+1
    teksti = sprintf('Ajon %.0f matka: %.0f',muut_ajot-1,...
        matkat_testiajo8{ajo,1}(muut_ajot-1));
    legend([h(muut_ajot)], teksti);
 end

 %matkatiedot yhdistetyille ajoille:
 if yhdistely_paalla ==1
```

```
    nn=0;
    for bb = 1:zz
```

```
        teksti_yhdist = sprintf('Yhdistetyn ajon %.0f matka: %.0f',bb,...
            matkat_yhdist(bb));
        legend([plot_yhdist(bb)],teksti_yhdist);
        nn=nn+1;
        disp(sprintf('Ajo %.0f on alkup ajoa pidempi = %.0f m',bb,...
            matkat_yhdist(bb)-matka_alkup{1,nn}(end)))
        if nn == 10
            nn=0;
        end
    end
 end
 %alkuperäisen ja uuden matkan erotus
erotustxt=sprintf('Alkuperäinen on %.0fm pidempi', 3280 - ...
    (matkat_testiajo8{ajo,1}(muut_ajot-1)));
disp(erotustxt);

%keskinopeus
keskinopeus=0;
for u = alkupiste_uus(ajo):size(kmh_testiajo8{ajo,zz},2)
    keskinopeus = keskinopeus + kmh_testiajo8{ajo,zz}(u);
end
keskinopeus = keskinopeus / (size(kmh_testiajo8{ajo,zz},2) - alku-
piste_uus(ajo));
```

Appendix B. Matlab function used to skip a stop in the driving cycle

```
function [kmh_testiajo8,aikaero_kok,aika_pysakki_testi8] = ...
    pysakin_ohitus_nopeusjlkn(pysakin_numero,aikaero_kok,b,c,d,...
    aika_pysakki_testi8,kmh_testiajo8,lisattava_matka,loppupiste,alkupiste)

    %kohta josta jatkuu kun pysäkki skipataan (tässä pisteessä nopeus ~=
    %lakipisteessä ennen jarrutusta)
    skipattu_kohta=b+round(aikaero_kok)+alkupiste;

    %tästä pisteestä eteenpäin kiihdytys lakipisteeseen alkaa. Eli lähtee
    %jarrutusta edeltävän kiihdytyksen lakipisteen nopeudesta.
    lakipiste_kiihdytys=c+round(aikaero_kok)+alkupiste;

    %kohta jossa korkein nopeus jarrutuksen jälkeen (tätä nopeutta
    %mentäisiin, jos ei jarruteta, käytetään laskuissa kun pysäkki jätetään
    %välistä)
    lakipiste_joseijarruta=d+round(aikaero_kok)+alkupiste;

    %tämä on aika, joka "säästetään" kun ei pysähdytä tälle pysäkille. Tämä
    %aika pitää siis leikata pois kuvaajasta kun pysäkki ohitetaan
    %säästetty aika = (kohta josta ilman muutoksia jatkuisi) - (kohta jossa
    %korkein nopeus ennen jarrutusta)
    saastoaika=skipattu_kohta-lakipiste_kiihdytys;

    %puuttuvien jarrutuksien ja kiihdytyksien tuottama aikalisä kun pysäkki
    %ohitetaan pysähtymättä (vs. alkuperäisen pysähdys)
    ohiajolisa=round((lisattava_matka/ ...
        (kmh_testiajo8(lakipiste_joseijarruta)/3.6))*10);


    erotus = abs(lakipiste_joseijarruta-skipattu_kohta);

    %tässä piirretään jarrutuksen sijaan kiihdytys, mikä jatkuu
    %alkuperäisessä) kuvaajassa pysäkin jälkeen (jatketaan kohdasta mistä
    %jarrutus alkoi).
    %alkaa jarrutusta edeltävästä lakipisteestä, loppuu uuden kiihdytyksen
    %lakipisteeseen.
    for a = lakipiste_kiihdytys : lakipiste_kiihdytys + (erotus)
        kmh_testiajo8(a) = kmh_testiajo8(a+saastoaika);
    end

    %tässä lisätään tasaista nopeutta tarvittu määrä (ohiajolisan mukaan)
    %alkaa lakipisteestä ja loppuu muutoksenloppuun.
    for a =lakipiste_kiihdytys + (erotus) : lakipiste_kiihdytys + ...
            (erotus) + ohiajolisa
        kmh_testiajo8(a) = kmh_testiajo8(lakipiste_joseijarruta);
    end

    %tässä piirretään muutoskohdan jälkeisestä kuvaajan loppuun normaalisti
    for a = lakipiste_kiihdytys + (erotus) + ohiajolisa : loppupiste- ...
            saastoaika +ohiajolisa+round(aikaero_kok)-1
        kmh_testiajo8(a) = kmh_testiajo8(a+saastoaika-ohiajolisa);
```

```
    end

    aikaero = ohiajolisa - saastoaika;
    aikaero_kok = aikaero_kok +aikaero;
    %ylempi siis koko pysähtymisoperaation kuluttama aika (sekunneissa*10)

    %poistetaan turhat pisteet lopusta (ilman tätä ajo liian pitkä)
    kmh_testiajo8(loppupiste+round(aikaero_kok) : end) = [];

    %pysäkkien ajankohdat testiajo8:lle (muuttuu jos tulee uusia
    %pysähdyksiä/skipataan pysäkkejä)

    for a=pysakin_numero+1:9
        aika_pysakki_testi8(a)=aika_pysakki_testi8(a)+round(aikaero/10);
    end
```

Appendix C. Matlab function used to create a stop in the driving cycle

```
function [kmh_testiajo8,aikaero_kok,aika_pysakki_testi8] = ...
    pysahtyminen_pysakille(pysakin_numero,kmh_testiajo8, alkupiste,...
    loppupiste,aika_pysakki_testi8,pysahdysaika,aikaero_kok,filtkmh_force)

    %aikaero ennen tätä pysäkkiä
    aikaero_ennen=aikaero_kok/10;

    %jarrutukseen kuluva aika
    aika_jarrutus = (kmh_testiajo8(round(aika_pysakki_testi8...
        (pysakin_numero)*10)+alkupiste-773)/3.6 / 1.5)/2;

    %kiihdytykseen kuluva aika
    aika_kiihdytys = (kmh_testiajo8(round(aika_pysakki_testi8...
        (pysakin_numero)*10)+alkupiste-773)/3.6 / 0.8)/2;

    %koko pysähtymistapahtumaan kuluva aika
    hidastusaika(pysakin_numero) = aika_kiihdytys + aika_jarrutus;

    %tämän pysähtymisen aiheuttama aikaero
    aikaero= pysahdysaika + hidastusaika(pysakin_numero)*10;

    %aikaero muut pysäkit huomioiden
    aikaero_kok = aikaero_kok + aikaero;

    %lasketaan hidastus pysäkille (kiihtyvyydellä -1.5 m/s2)
    %aloittaa jarrutuksen pysäkkiviivan perusteella (vähentää siitä hidastus-
ajan)
    for a = alkupiste-773+round(aika_pysakki_testi8(pysakin_numero))*10-...
            round(2*aika_jarrutus)*10:alkupiste-773+round(...
            aika_pysakki_testi8(pysakin_numero))*10+pysahdysaika

        kmh_testiajo8(a) = (kmh_testiajo8(a-1)/3.6 - 1.5*0.1)*3.6;

        if kmh_testiajo8(a) <0
            kmh_testiajo8(a) = 0;
        end
    end

    %lasketaan kiihdytys takaisin edelliseen nopeuteen
    for a = alkupiste-773+round(aika_pysakki_testi8(pysakin_numero))*10+...
            pysahdysaika : loppupiste+round(aikaero_kok)

        kmh_testiajo8(a) = (kmh_testiajo8(a-1)/3.6 + 0.8*0.1)*3.6;

        if kmh_testiajo8(a) >= filtkmh_force(a-round(aikaero_kok))-0.5
            kmh_testiajo8(a) = filtkmh_force(a-round(aikaero_kok));
            %keskeytetään siihen kun ollaan törmätty alkuperäiseen nopeuteen
            aa=a;
            break
        end
```

```
    end

    %jatketaan keskeytyneestä kuvaajan loppuun
    for a = aa:loppupiste+round(aikaero_kok)
        kmh_testiajo8(a) = filtkmh_force(a-round(aikaero_kok));
    end

    %pysäkkien ajankohdat testiajo8:lle (muuttuu jos tulee uusia
    %pysähdyksiä/skipataan pysäkkejä)
    for a=pysakin_numero+1:9
        aika_pysakki_testi8(a)=aika_pysakki_testi8(a)+round(aikaero/10);
    end
```