

Small or medium-scale focused research project (STREP)



ICT Call 8

FP7-ICT-2011-8

**Cooperative Self-Organizing System for low Carbon Mobility at
low Penetration Rates**

COLOMBO

COLOMBO: Deliverable 4.2

**Extended Simulation Tool PHEM coupled to SUMO
with User Guide**

Document Information	
Title	Deliverable 4.2 - Extended Simulation Tool PHEM coupled to SUMO with User Guide
Dissemination Level	PU (Public)
Version	1.7
Date	28.02.2014
Status	final



Authors	Stefan Hausberger (TUG), Daniel Krajzewicz, (DLR)
---------	---

Table of contents

1	Introduction.....	5
1.1	Motivation	5
1.2	Objectives.....	6
1.3	Structure	6
2	PHEM Extension by Vehicle Models for the Year 2020	7
2.1	Engine Start/Stop.....	8
2.2	Hybrid Electric Vehicles (HEV)	9
2.2.1	The HEV model in PHEM	9
2.2.2	HEV passenger car.....	11
2.2.3	HEV city bus.....	12
2.3	Battery Electric Vehicles (BEV)	13
2.4	Compressed Natural Gas (CNG).....	15
3	Off-Line Interaction between PHEM and SUMO	17
3.1	Motivation	17
3.2	Implementation.....	17
3.2.1	PHEM Input Files	18
3.2.2	Converting SUMO-outputs to PHEM-inputs.....	19
3.2.3	Functionality Extensions.....	20
3.3	Performance Evaluation	21
3.4	Deployment	21
4	PHEMlight – PHEM Derivative embedded in SUMO.....	23
4.1	Outline.....	23
4.2	Implementation.....	24
4.2.1	Transient Computations	24
4.2.2	Emission and vehicle data handling.....	24
4.2.3	Implementation of the Interface to SUMO, Combining Data Handling and Transient Computation	26
4.2.4	Computation of Maximum Acceleration	27
4.2.5	Data Generation and Nomenclature Conventions.....	29
4.2.6	PHEM Emission Classes.....	31
4.2.7	Embedding into SUMO	34
4.2.8	User Interaction.....	35
4.3	Evaluation.....	36
4.4	Real Use Case: Evaluation of Traffic Management Practices	40
4.5	Deployment	43
5	Summary.....	45

Appendix A – References	46
Appendix B – Documentation for the Off-line Interaction between SUMO and PHEM	47
Appendix B.1 – TraceExporter Documentation	48
Appendix C – PHEMlight User Guide	54
Appendix C.1 – Page on Included/Supported Emission Models	55
Appendix C.2 – Page on PHEMlight	56
Appendix D – Tools for Emissions Evaluations	58

1 Introduction

The goal of the COLOMBO project is to deliver a set of modern traffic management applications that target at increasing traffic efficiency and at decreasing traffic's environmental footprint. All these applications are developed based on simulations. As such, they heavily rely of proper models for both traffic and vehicular emissions.

The consequences of the environmental pollution co-produced by traffic nowadays, especially the greenhouse effect, are well known by the society and political actors react to it. The directive that currently regulates the air condition in Europe is "2008/50/EC" and was passed by the European Parliament and the European Council on 21 may of 2008. It sets concentration limits to a set of air pollutants and forces parties responsible for road networks (often city councils) to perform administrative and executive actions to assure that these limits are not broken.

Besides actions done by traffic administration and management authorities, the improvement of vehicles by car manufacturers and the introduction of non-fossil engines, a large part of Intelligent Transportation Systems (ITS) aims on reducing pollutant emission and fuel consumption. ITS solutions work at different levels. On a coarse scale, logistical attempts are implemented that schedule vehicle trips for reducing the overall journey distances or for using time slots where only few other traffic participants can be expected – and a lower number of jams. Traffic lights or intelligent navigation systems interact with a subset of vehicles, often via wireless communication, trying to reduce the waiting times and/or accelerations. On a fine scale, methods for reducing a single vehicle's accelerations are under development.

In most cases, ITS solutions reduce emissions and consumption by realizing a smooth journey through the road network as accelerations are a major source of energy consumption. The solutions developed within the scope of COLOMBO – traffic lights controlling, emission-optimal driver behaviour – operate mostly following this basic dependency.

When investigating the impact of new ITS solutions on vehicular emissions, one should be aware of expectable changes in the vehicle fleet. Not only can a change in the vehicle fleet's pollution / consumption characteristics yield in a difference of the relative benefit of an investigated ITS solution. Electrical or hybrid vehicles may also need other treatments to reduce their environmental impact.

In its first steps, which outcomes are presented in this document, COLOMBO's work package 4 is responsible for supplying the project with an emission computation model that fulfils the project's needs. The model has to cover future vehicle fleets, in the means of including the possibility to compute emissions for modern combustion engines, as well as of electrical and hybrid vehicles. In addition, the emission model must be integrated into COLOMBO's overall simulation architecture (see also Deliverable 5.1 ([COLOMBO D5.1, 2013])). These requirements are fulfilled by extending PHEM¹ ("Passenger Car and Heavy Duty Emission Model") and by including a simplified model derived from PHEM into the used traffic simulation SUMO² ("Simulation of Urban MObility"). Both tasks are described within this document.

1.1 Motivation

One of COLOMBO's major targets is to develop applications that reduce the ecological footprint of traffic. Therefore, the COLOMBO project needs a software module that allows computing realistic emission values. This model will be mainly used within the development of traffic light algorithms within the work package 2 and within the development of driver-optimal behaviour performed in work package 4. In addition, the model will be used within further steps of WP4 itself, while

¹ <http://www.ivt.tugraz.at/de/forschung/emissionen.html>

² <http://sumo-sim.org/>

working on emission-optimal driver behaviour and the guideline on developing eco-friendly traffic light controls.

To use the state-of-the-art vehicular emissions model PHEM was decided during the project's preparation steps. The technical work consists of extending PHEM and embedding it within the COLOMBO simulation architecture.

1.2 Objectives

This document shall present two steps of the COLOMBO project. The first part shall cover the extensions performed on PHEM's database for modelling the vehicle fleet in the year 2020.

The second work item presented in this document was to allow using PHEM as an emission model within the COLOMBO overall system. Two possibilities were aimed, both described herein: an off-line connection for computing emission measurements using PHEM based on files SUMO generates in a first step and an on-line connection between both applications.

1.3 Structure

The document is structured as following. Chapter 2 describes extensions to PHEM performed in Task 4.1 towards modelling the vehicle fleet for the year 2020, including different types of electrical vehicles and vehicles powered by natural compressed gas. Chapter 3 presents on off-line coupling between SUMO and PHEM. In chapter 4, PHEMlight, a PHEM derivative that is directly implemented into SUMO is presented. The document closes with a summary.

2 PHEM Extension by Vehicle Models for the Year 2020

The simulation tool PHEM and the PHEMlight version (see chapter 4) are using two main data files per vehicle category with pre-defined input data to specify the vehicle characteristics relevant for fuel consumption and emissions. One file describes the vehicle properties (vehicle category, mass, loading, frontal area, Cd-value, tire properties etc.) and the other file the engine emission behaviour. The data necessary for all vehicle categories in PHEM are gained from measurement campaigns and corresponding evaluation work in the ERMES group³ and for the HBEFA⁴. From this work data on gasoline and diesel vehicles is available from EURO 0 to EURO 6 emission legislation (Figure 2.1).

Since hybrid electric vehicles (HEV), plug-in- hybrid electric vehicles (PHEV), battery electric vehicles (BEV) and vehicles using combustion engines but different fuels (e.g. CNG and E85) are entering the market, the emission simulation tool shall be extended to these vehicle categories. This seems to be especially important since SUMO and PHEM will be applied to simulate the effect of future traffic signal control strategies which shall lead to low fuel consumption and low pollutant emissions also for the future vehicle fleet. Thus the user shall have the option to test the control strategies for traffic signals also with a model describing the expected main characteristics of the future vehicle fleet. The user has following options to define the fleet composition in PHEM:

1. Load the generic “average” EU fleet composition from a predefined file
2. Define user defined shares of the available and pre-defined vehicle categories
3. Define own vehicle technologies by adapting data on vehicle properties (* .veh) files and/or by adapting the engine emission maps (* .mep and others depending on the technology) and defining then the share in vehicle mileage for these vehicle categories.

Options 1. and 2. are easy to handle in one file which contains the shares in the total vehicle mileage while option 3. needs a funded understanding of the simulation tool and of vehicle and engine technology to produce realistic vehicle models.

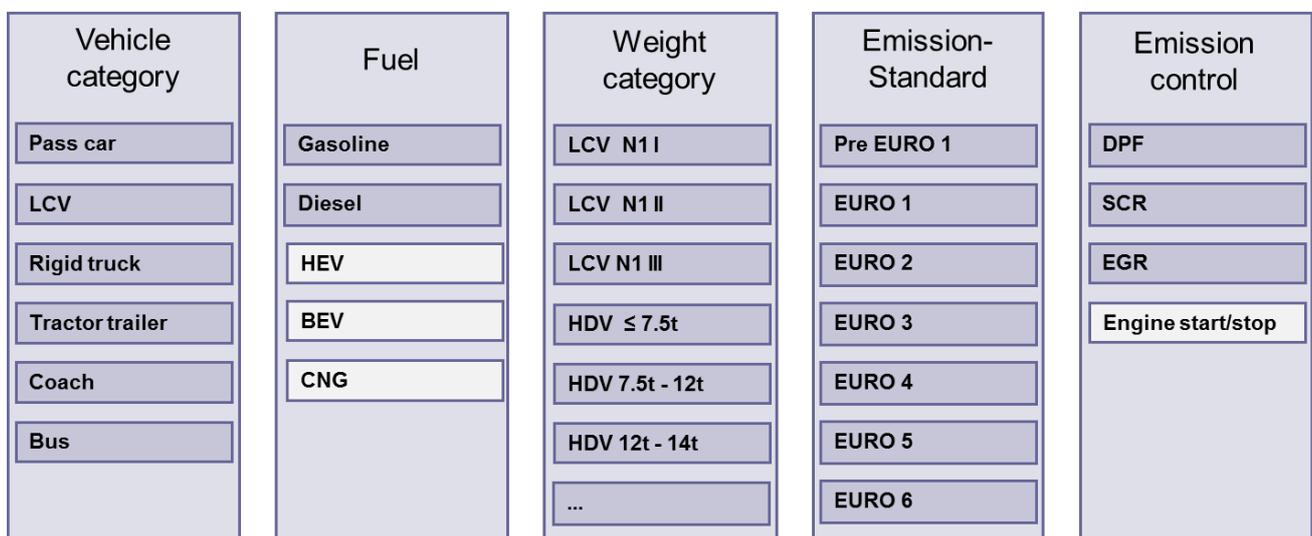


Figure 2.1: Structure of vehicle categories for emission calculation in PHEM and PHEM light (technologies added within COLOMBO are highlighted in light grey)

³ <http://ermes-group.eu/>

⁴ www.hbefa.net

Following technologies have been added to the pre-defined data sets in PHEM:

- Engine start/stop: PHEM shuts off the engine at vehicle standstill
- HEV: for passenger cars and for buses a model for parallel electric hybrids was set up which includes a subroutine for the electric motor, for the battery and for the controller, which decides under which conditions the vehicle drives electric, by combustion engine, by both engines or in electric energy generation mode. The HEV model can use gasoline or diesel engine maps.
- BEV: for passenger cars a model for an electric vehicle with a Lithium Ion battery was set up which includes a subroutine for the electric motor, for the battery and for the controller.
- CNG: for buses and for cars also CNG engine emission maps have been implemented.

The data source and the validation for each of these vehicle categories are described below.

2.1 Engine Start/Stop

Engine start/stop systems are a widespread technology for modern cars to reduce the fuel consumption and emissions. Start-stop systems switch off the engine automatically during vehicle standstill phases. Thus, emissions during engine idling are avoided. The total positive effect depends on the share of idling in a driving cycle (share of idling time) but also on the percentage of vehicles equipped with this technology and on the percentage of idling conditions where the engine is shut off at those vehicles equipped with a start/stop system (e.g. at low temperatures usually the engines keeps running at idling to ensure heating of the cabin and to reduce battery wear and also at low state-of-charge levels of the battery the engine is not switched off).

Market penetrations for start-stop systems are available from battery manufacturers since the performance of batteries has to be adapted to the electrical requirements of start-stop systems, e.g. increased cycling and deeper discharge ([Bosch, 2012], [Johnson, 2013]). The resulting market shares are included in the PHEM data as default values (Table 2.1). To consider effects of low battery charge in several driving conditions, cold temperatures etc. an average start-stop activation factor of 0.7 has been assumed for the simulation. The total start-stop-factor is simply the product from market penetration and activation factor. Table 2.1 gives market penetrations, activation factors and total start-stop factors. In addition in PHEM a threshold of 5 seconds minimum engine-on time before the stop-function can be activated again is set. Petrol and diesel vehicles are treated by the same method, HEV are in principle shutting off the combustion engine in idling and motoring conditions and for BEV the start/stop technology is not relevant.

Table 2.1: Market penetrations and activation factors of start-stop systems

	EURO 5	EURO 6	EURO 6c
Market penetration	50%	75%	90%
Activation factor	0.7	0.7	0.7
Total start-stop factor	0.35	0.525	0.63

To reflect the effects of start-stop technologies in the emission simulation the following method is applied: When PHEM simulates a driving cycle the emissions interpolated during idling phases are corrected by applying the total start-stop factors from Table 2.1 as correction factors, i.e. idle emission get multiplied by $(1. - total\ start-stop\ factor)$. A start stop factor of one for a vehicle category thus leads to zero emissions at idling (if reached more than 5 seconds after the engine was started). A stop factor of zero for a vehicle category leads to normal idling emissions (e.g. default settings for EURO 0 to EURO 4).

The settings for the start/stop system are defined in the vehicle data file (user can define the total start-stop factor in the *.veh input file).

2.2 Hybrid Electric Vehicles (HEV)

A first model for HEV was implemented in PHEM in a national research project dealing with the optimisation of the HEV design for diesel electric cars ([Luz, 2011]). This tool however is much too complex for application in the simulation approach for fleet emissions. Thus the HEV model was simplified and “typical” EURO 6 HEV vehicle models have been set up for a car and for a city bus. Both models have been validated by measurements of representative vehicles in real world test cycles.

2.2.1 The HEV model in PHEM

PHEM simulates for all vehicle categories the actual power demand from the propulsion system. This is the sum of engine power demand to overcome air and rolling resistance, of the power necessary for acceleration and road gradients and of the power necessary to run auxiliaries such as light, ventilation, air conditioning, etc. This calculation is valid for any propulsion technology (Figure 2.2). For HEV the “HEV-controller”, which is a subroutine in the PHEM-code, decides which mode is possible and which is overall the most fuel efficient option at current driving situation and battery state-of-charge (SOC):

- Electric driving (typically at lower speeds and high SOC)
- Assist (electric motor supports internal combustion engine (ICE), typically at full load)
- Generation (ICE increases torque to use the electric motor as generator to load the battery)
- Recuperation (kinetic energy of the vehicle is used to run the electric motor as generator to load the battery during deceleration phases, activated if the SOC is not already at the maximum tolerated SOC level)

With this decision the power and engine speed which has to be provided by the ICE can be computed. With this data emissions are simulated for the ICE as for the conventional vehicles using the engine map, transient correction functions and exhaust gas after-treatment models.

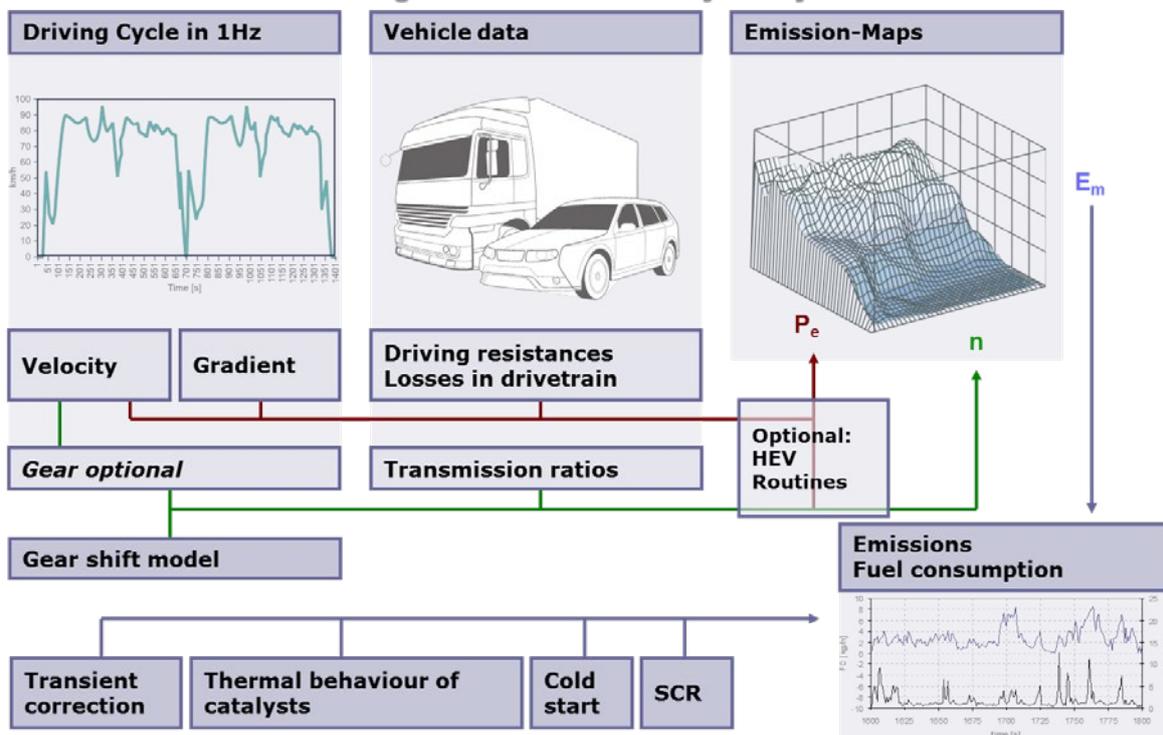


Figure 2.2: Schematic picture of the model PHEM with the HEV ad-on

For the electric system the efficiency of the electric motor is simulated also by an efficiency map as function of torque and rpm while the battery model uses the internal resistance to compute the losses during charging and discharging as function of the actual Current and Voltage. The battery SOC is also computed simply by integrating the electric energy going from and to the battery cells after all losses are applied. Figure 2.3 shows the design of the parallel HEV in PHEM.

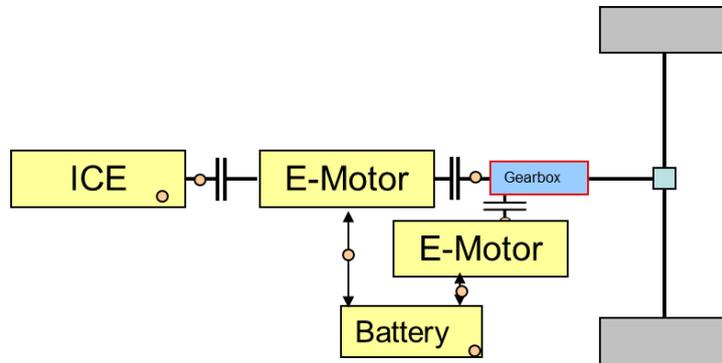


Figure 2.3: Structure of the HEV design in PHEM

The main HEV-control function in PHEM is based on a characteristic curve which defines if boost (or electric driving) or generating electric energy is the best option at a given SOC level at the actual power demand (Figure 2.4). PHEM calculates for every second in the driving cycle the kWh electric energy necessary to run the electric motor for electric driving or assisting and also the electric energy gained in the case of generation mode. Depending on the values computed for the actual load demand (power and rpm) one of these options has a higher distance to the characteristic controller curve and is selected as operation mode. The logic behind this controller is that on average over a total trip the fuel used to produce 1 kWh electric energy has to be lower than the fuel saved by electric driving (or assisting) per kWh electric energy consumption. Certainly the brake energy recuperated on average over trips is balanced here also but with zero fuel consumption per kWh electric energy produced. The characteristic line has been defined by running PHEM in many real world cycles and check where the balance level is. This level defines the horizontal line in the characteristic curve. Towards low SOC the curve drops to zero to ensure that under all driving situations electric energy is generated and no energy is taken from the battery for driving. Towards high SOC the characteristic curve has a steep increase to consume electric energy to largest possible extent to keep some capacity in the battery for a next braking event. With this controller the battery SOC is kept between the SOC levels which lead to a long battery life time and thus reflect real HEV systems. Several further controller details are necessary to gain a stable function. These are described in a diploma thesis of the Institute ([Smode, 2013]).

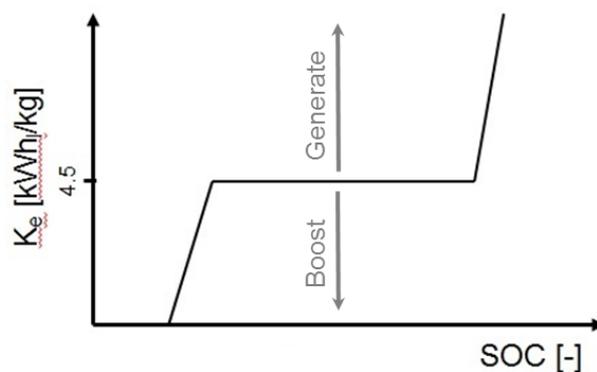


Figure 2.4: Schematic picture of the HEV control system applied in PHEM

2.2.2 HEV passenger car

The HEV model for passenger cars was adapted to a VW Jetta hybrid with a gasoline engine which has been measured in real world trips at TUG with on-board measurement equipment (PEMS). The main vehicle characteristics of the HEV car model in PHEM are shown in Table 2.2

Table 2.2: Vehicle data for the HEV car in PHEM

Information	Unit	Value
Vehicle mass	kg	1510
Motor		electro permanent synchronous
Nominal power	kW	28
at nominal engine speed	1/min	2500 - 8000
Combustion engine rated power	kW	110
at nominal engine speed	1/min	6000
Air resistance coefficient (Cw)		0.30
Cross sectional area	m ²	2.14
Transmission		6 gears manual
Axle drive ratio		3.88
Type of battery		Li-Ion
Battery voltage	V	222
Battery capacity	kWh	1.1
Battery mass	kg	53
Vehicle mass	kg	1510

Since the “average” HEV car has today higher electrical power (battery and motor) than the Jetta and reports also lower fuel consumption values, the PHEM model was set to give slightly lower fuel consumption values than measured in the Jetta. The validation shown in Figure 2.5 was done by recalculating the trips performed at the on-board tests with PHEM.

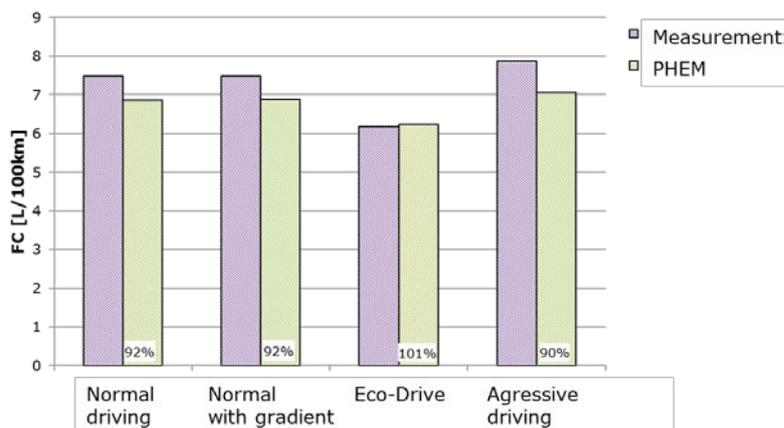


Figure 2.5: Results of the validation of the PHEM-HEV model for cars with measurements on a VW-Jetta hybrid in mixed real world driving

2.2.3 HEV city bus

The basis for the city bus model was a Volvo 7000 hybrid bus which was measured also at TUG. The design of power pack of the HEV bus is similar to the HEV car, i.e. a parallel hybrid with electric motor, a battery, a diesel combustion engine and a HEV controller. Certainly the control strategy is adapted to the city bus. In comparison to the conventional bus in PHEM the results for the hybrid version show approx. 20% less fuel consumption which is mainly due to the brake energy recuperation. The main vehicle characteristics of the HEV city bus model in PHEM are shown in Table 2.3.

Table 2.3: Vehicle data for the HEV city bus in PHEM

Information	Unit	Value
Vehicle mass	kg	12.200
Motor		electro permanent synchronous
Nominal power	kW	70
at nominal engine speed	1/min	2500 - 8000
Combustion engine rated power	kW	161
at nominal engine speed	1/min	22000
Air resistance coefficient (C _w)		0.6
Cross sectional area	m ²	6.5
Transmission		12 gears automated
Axle drive ratio		5.22
Type of battery		Li-Ion
Battery voltage	V	222
Battery capacity	kWh	1.1
Battery mass	kg	53
Vehicle mass	kg	12.200

In praxis quite different HEV bus designs can be found with different electric engine power and different battery capacity. Beside parallel hybrids also serial hybrid buses are on the market. Main findings of a national research project dealing with the analysis of the driver influence on the fuel saving potential of HEV was that the driver shall avoid braking with higher power than the maximum which the electric machine and the battery are capable of converting to electric energy stored in the battery ([Kies, 2013]). The HEV bus in PHEM has 70 kW nominal electric power and 161 kW rated power of the diesel engine. This leads to the effect that strong decelerations need much additional mechanical braking and the efficiency of brake energy recuperation drops significantly. This behaviour however seems to be typical for actual parallel HEV buses, since installing higher electric power significantly increases the costs of the system⁵. The validation of the HEV model in PHEM was made based on test data from the Volvo 7000 HEV bus measured in ([Kies, 2013]) and showed good agreement between simulated and measured fuel consumption and emissions (Figure 2.6). The test data included different driving styles where the tests on a test track

⁵ Serial hybrid buses have much higher nominal electric power but suffer from efficiency losses from the generator to the electric motor. Thus the overall efficiency of serial buses is not better than that of parallel systems.

covered different combinations of gas pedal position at acceleration (acc.+% in Figure 2.6) and brake pedal position (br.+% in Figure 2.6) to validate if the brake energy recuperation is simulated in a realistic way. Since realistic driving situations were met with less than 10% deviation the model seems to be sufficiently accurate since not a single bus model but “average HEV bus behaviour” shall be simulated in combination with SUMO. Meeting the test results from a single bus accurately would need the knowledge of the detailed HEV-control strategy of the bus. This information certainly is not published by the OEMs.

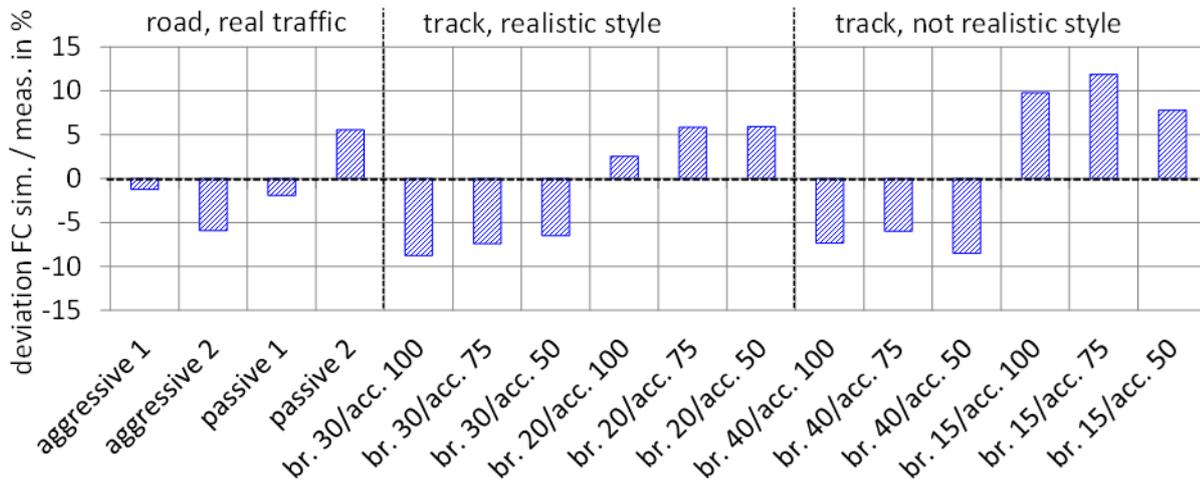


Figure 2.6: Results of the validation of the PHEM-HEV model for the city bus with measurements on a Volvo 7000 hybrid bus in urban bus operation and on a test track

The bus parameters can be changed by the user rather easily (mass, loading, rated power of the diesel engine and of the electric motor and battery capacity). However, to obtain a reasonably HEV bus model in PHEM after amendments in the input data needs some know how on HEV architecture and also testing of the resulting model.

2.3 Battery Electric Vehicles (BEV)

The battery electric vehicle model for PHEM was set up for rather small city car based on measurements at TUG at a Peugeot ION. The main characteristics of the car are given in Table 2.4. The same model structure for the electric motor and for the battery has been used as for the HEV. Certainly the input data is different for the BEV having a higher nominal power for the motor and a much higher battery capacity.

The quality of the BEV model depends very much on the accuracy of the motor efficiency map. The values of this map describe the relation between mechanical power output and electric power input (motor), resp. the relation between electric power output and mechanical power input (generator). The maximum theoretical efficiency thus is 100%. OEM specific values are typically not published, thus the BEV model uses a generic motor efficiency map (Figure 2.7). For this purpose data from motor manufacturers were collected and an averaged map normalised to maximum power and speed was generated. This ‘TUG standard map’ displays large areas with high efficiencies of more than 90%, but also some small areas with drastically declining efficiencies especially at low power in combination with very low or very high speed. It has to be noted that different motor technologies are applied at actual BEV and thus the real efficiency of single vehicles can differ significantly from the model produced here which shall represent an “average” BEV. Beside the motor also quite different gear boxes are mounted. Having a higher number of gears allows driving in favourable motor speed ranges more often than in the case of a fixed transmission. Especially very low and high motor speeds at low power demand result in rather low motor efficiencies. A disadvantage of an additional gear box are the additional costs.

Table 2.4: Vehicle Data for the BEV car in PHEM

Information	Unit	Value
Vehicle mass	kg	1100
Motor		electro permanent synchronous
Nominal power	kW	47
at nominal engine speed	1/min	2500 - 8000
Air resistance coefficient (C _w)		0.33
Cross sectional area	m ²	2.14
Transmission		fixed gear ratio
Axle drive ratio		6.066
Type of battery		Li-Ion
Battery voltage	V	330
Battery capacity	kWh	16
Battery mass	kg	230
Standard range	km	150

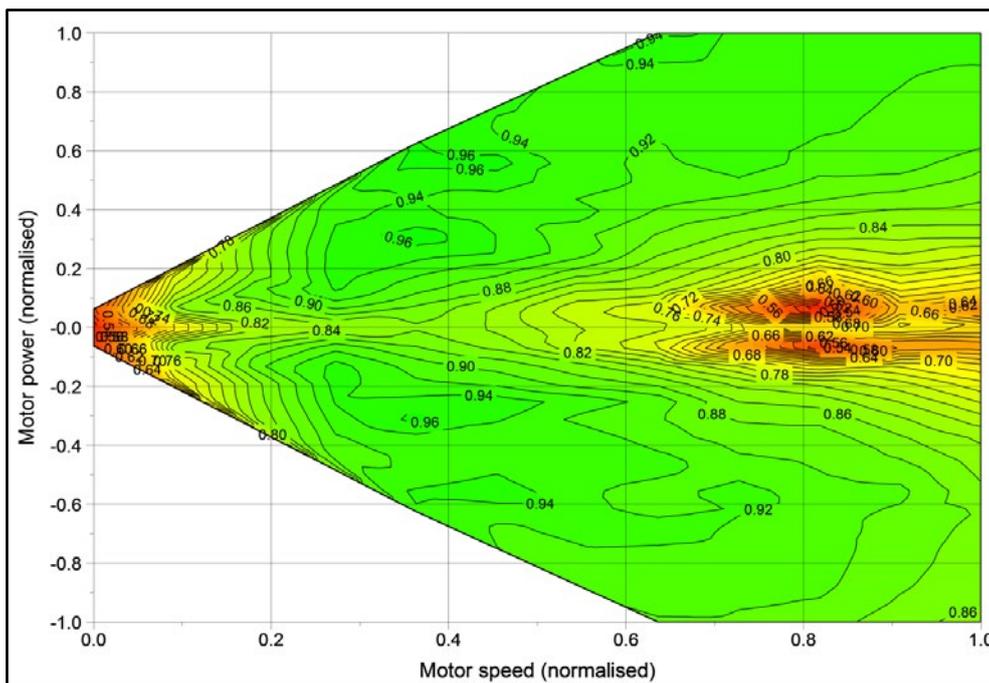


Figure 2.7: TUG standard efficiency map of an electric motor for passenger cars at 25 °C ambient temperature

As for the other new vehicle models also the BEV was validated against measured energy consumption values. The test vehicle was a Peugeot ION. The measured electric energy flows between the battery and the motor are compared to the model output values. In Figure 2.8 the total energy consumption values over the whole cycle are given in relation to the driven distance (kWh/km). Good agreements are found indicating that the used model approach and the motor efficiency map are reliable for depicting the electric power supply of the electric machine. The total

deviations are in the range between -4% and +1%. Only the CADC⁶ sub cycle for urban driving features a clear underestimation of -20%. A more detailed analysis of the modal energy flow shows, that the PHEM model overestimates the brake energy recuperated at higher deceleration rates in the urban cycle. This indicates that either mechanical braking is underestimated or that the efficiency of the generic motor is too optimistic at negative load and rather low rpm.

Since the BEV shall rather represent 2020 technology, we assume that a more efficient recuperation of electric energy is to be expected for future BEV and the model is not further adapted to the test results of the Peugeot ION.

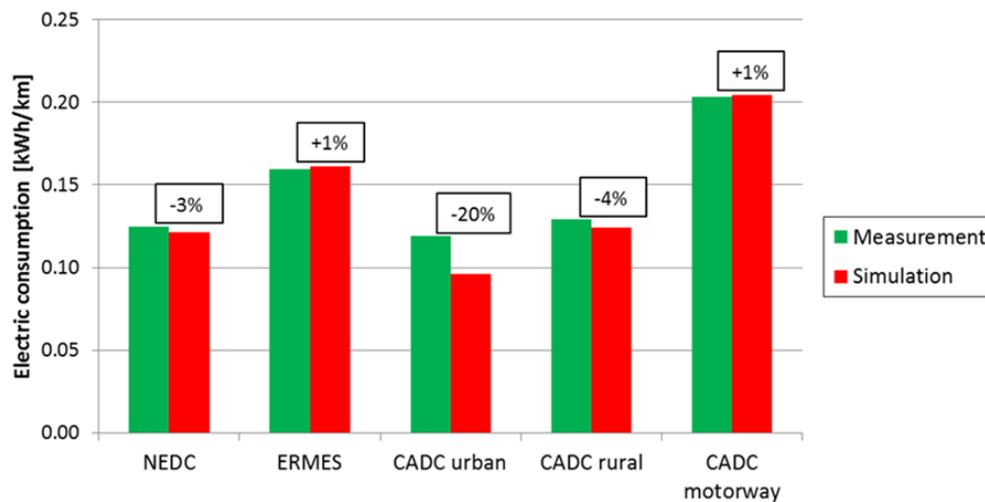


Figure 2.8: Comparison between measured and modelled electric power consumption of the motor during phases of positive mechanical output (25 °C)

2.4 Compressed Natural Gas (CNG)

In PHEM, the vehicles with combustion engines running with alternative fuels are simulated similar to the conventional gasoline or diesel vehicles. Only the engine fuel consumption and emission maps are exchanged to consider the different engine. Also additional weight for alternative tank systems is added where relevant.

The engine fuel consumption and emission maps for CNG are gained from the engine maps of conventional vehicles by applying “alternative fuel factors (AF)” to the map values. These AF define the ratio of a vehicle model running with alternative fuel to the same model when running with conventional fuel measured in the same test cycle on the chassis dynamometer. For CNG cars the AF is gained from the ratio of available measurements for CNG cars in relation to the same models running in the conventional gasoline version. Using the ratios instead of absolute values measured for CNG seems to be the better approach since the variability in pollutant emissions was found to be 3 orders of magnitude for conventional gasoline cars (however, all on very low absolute emission level, see www.hbefa.net). Since the budget for testing CNG models is limited, the small sample may result in not representative emission levels assuming a similar standard deviation within the CNG vehicles than found for gasoline cars. Thus the average AF was calculated from the available test data in the ERMES group on CNG cars and on CNG buses (AF in relation to the same bus type running with a diesel engine) by averaging the AF of the single vehicles.

⁶ CADC is the Common ARTEMIS Driving Cycle, a speed cycle over time developed for chassis dynamometer tests of cars representing typical real world driving situations. The cycle was developed in the FP5 project ARTEMIS and is the most established real world test cycle in the EU.

At the time being, still new test data on EURO 6 cars is analysed to validate the AF elaborated from the EURO 5 vehicle tests. Results shall be available end of 2013 and will be published in the next deliverable for this work package.

3 Off-Line Interaction between PHEM and SUMO

In a first step, an off-line interaction between SUMO and PHEM was implemented. “off-line” means here, that both applications do not communicate while being executed. Instead, the traffic simulation SUMO is executed, first. While being executed, it generates a dedicated output file. This output file is converted into a file PHEM can read. At last, the file obtained by this conversion is given to PHEM as input and PHEM generates the resulting emission values. This process can be performed manually, by calling the involved tools one-by-one, or can be embedded into a script.

The realization of the off-line coupling between SUMO and PHEM is described to a larger detail in the following sub-sections. The remainder of this chapter is structured as following: at first, we motivate the implementation of the off-line coupling. Then, the implementation is described, including a descriptions of the data to generate, the conversion process itself, as well as additional features that were added to the conversion script. Afterwards, some initial performance and usability tests are presented, followed by a section that covers how the tool is made available and how it is embedded into the SUMO software package. We close with a summary.

3.1 Motivation

The motivation to start with an off-line coupling system was to have a fall-back possibility to use PHEM even if an on-line coupling turns out to be technically too challenging, not portable, or turns out to be too slow or too memory consuming. It should allow continuing the work on other COLOMBO work packages even if an on-line coupling is not available or delayed. Technically, this approach is quite simple: post-process available SUMO outputs to generate input files PHEM can read.

In addition, this approach allows using the original PHEM implementation. This is a feature wished within COLOMBO’s work package 4 to compare the results of the simplified models developed herein against the original PHEM computations.

3.2 Implementation

The off-line coupling is realised by generating files using SUMO that PHEM can read. No feedback to SUMO is needed in this usage scenario. Still, two attempts exist for realising the off-line coupling. The files could either be generated directly by the simulation, or the simulation’s outputs could be converted into files of this type. The second option was chosen, as it allows the SUMO developers to keep XML be the only output type SUMO generates and avoids the introduction of additional, specific single-use outputs. Figure 3.1 shows the exporter script in context with the applications and data needed to generate the emissions for a given scenario.

In the context of “COLOMBO’s overall simulation system” (COSS), see also D5.1 as well as Figure 3.2, the conversion script is an additional application that has to be executed after the simulation run and PHEM has to be executed afterwards. At the time being, extending iCS to run the script and PHEM after a simulation does not make sense, as this tool chain is too project specific while iCS is used also by third parties. Instead, the conversion should be performed by an additional script that belongs to or realises the COSS-loop. It should be said that given the on-line coupling, described in chapter 4, the usage of the conversion script within the COLOMBO loop will probably be neglected any way.

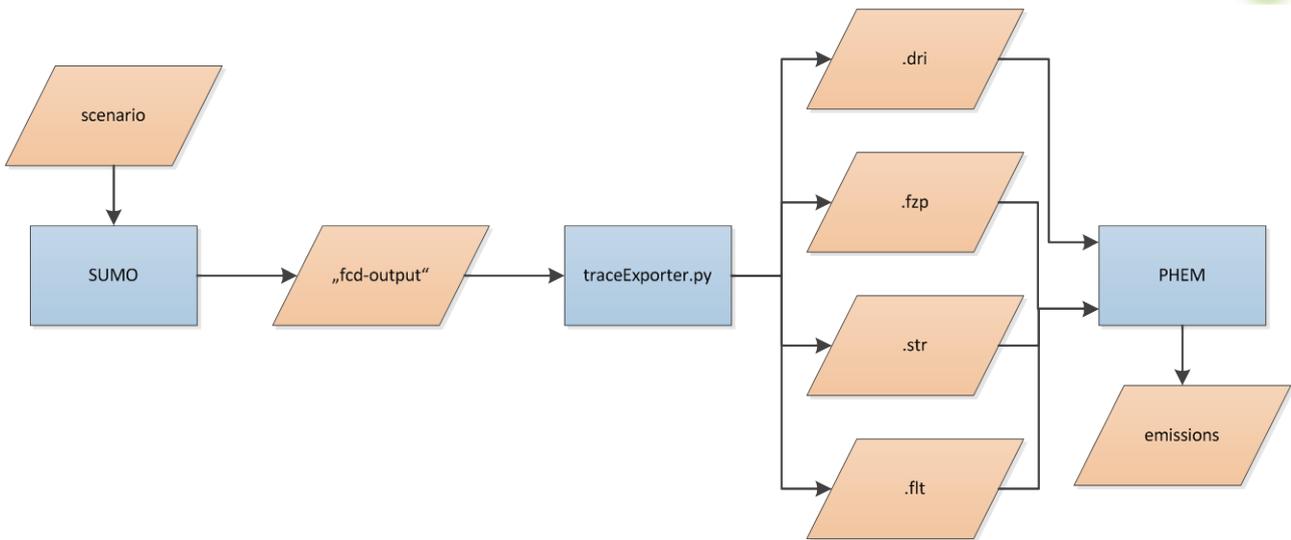


Figure 3.1: Data flow for off-line emissions computation

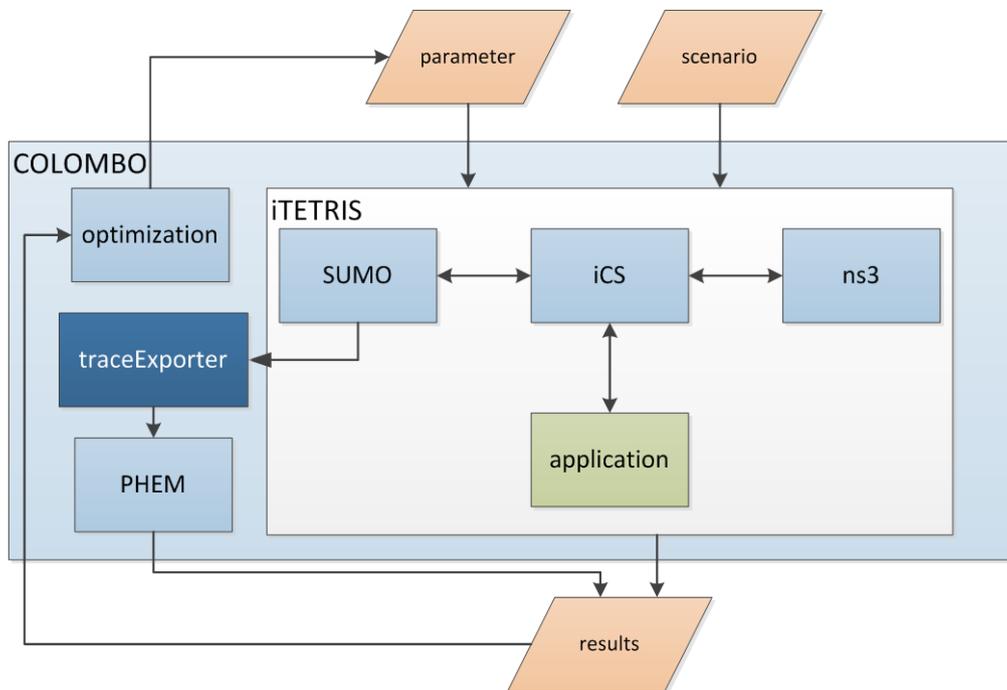


Figure 3.2: traceExporter in COSS context

The implementation of the off-line coupling between SUMO and PHEM is described in the following. At first, the file types read by PHEM are introduced. Then, a description of the conversion process is given. How the script is documented and how it is made available is described afterwards. This chapter ends with a short summary. The description covers the state of the script as given on the 17th of February, 2014.

3.2.1 PHEM Input Files

PHEM is capable to read two types of inputs:

- minimal mode: A single file with the extension “.dri” is used, which contains all vehicle speeds from a simulation run. It is encoded as a simple .csv file: it consists of single data lines, where each lines holds the attributes “tsim” (time, [s]), “v” (speed, [km/h]), “slope” (slope, [%])

“n” (revolutions, [1/min]). The first three data lines contain the version number (“v1”), the data names, and the measurements. Lines with a leading “c” are interpreted as comments.

- advanced mode: PHEM requires three files, all formatted as `.csv` files, namely:
 - `.str`: describes the road network’s streets. Each line contains values that describe a single street, namely: the street’s id (“Str-Id”), the number of lanes (“Sp”), the begin x-position (“SegAnX”), the end x-position (“SegEnX”), the begin y-position (“SegAnY”), and the end y-position (“SegEnY”);
 - `.fzp`: each line contains information about the vehicles’ progress through the road network, for each time step and all vehicles, namely: time (“t”), x-position (“WeltX”), y-position (“WeltY”), vehicle id (“Veh. No”), velocity (“v”), slope (“Gradient”), vehicle type id (“veh.Type-Id”), road id (“Str-Id”). The entries are sorted by time;
 - `.flt`: defines the vehicle population, by defining a vehicles class in each line using the following attributes: an abstract vehicle type (“Vehicle Type”), the name of the emissions parameters file (`.gen`) that shall be used (“gen file”), the percentage of the vehicles covered by the named `.gen`-definition within the named abstract vehicle type (“percentage”).

3.2.2 Converting SUMO-outputs to PHEM-inputs

A script written in Python⁷, version 2.7.x, has been implemented which reads information from files generated by SUMO into PHEM input files. All four file types read by PHEM are supported.

The main decision to take was which of the SUMO output files⁸ should be extended to retrieve all the needed values. The needed information about road geometries can be read from the simulated SUMO network, the vehicle types from the route files. To obtain the vehicle-dependent, simulated values that describe the movement of vehicles, the “fcd-output”⁹ was used, originally implemented by Mario Krumnow from TU Dresden. It had to be extended to include the information about the lane the vehicles use and the slope of the road at the vehicles’ current positions¹⁰.

To obtain files for PHEM’s “advanced mode”, the script works as following:

1. For building `.str` files, the script reads a SUMO road network¹¹. A map from SUMO road IDs to PHEM road ids is returned. It is later referenced by the other file types. One may note the following limitation: SUMO edges may have a “polyline”¹² geometry instead of being a straight line. This information is not passed further to PHEM.
2. The script reads the fcd-output, parses the single information, converting the vehicle’s and its type’s ids into a PHEM-representation, as well as the slope and the speed. For converting the road id, the map generated in step#1 is used. The values computed this way are stored in `.fzp` file format. The ids of both, vehicles and their types, are stored in maps which are returned.
3. The vehicle type map obtained from step#2 is used to build an according `.flt` file.

The process is depicted in Figure 3.3.

⁷ <http://www.python.org>

⁸ The complete list can be found at <http://sumo.sourceforge.net/doc/current/docs/userdoc/Simulation/Output.html>

⁹ <http://sumo.sourceforge.net/doc/current/docs/userdoc/Simulation/Output/FCDOOutput.html>

¹⁰ Only the slope at the vehicle’s current position is regarded. Changes in the slope within the vehicle’s movement during one simulation step are neglected. No evaluations of the errors that yield from this simplification have been performed.

¹¹ http://sumo.sourceforge.net/doc/current/docs/userdoc/Networks/SUMO_Road_Networks.html

¹² a polyline is a geometrical object consisting of connected points/positions

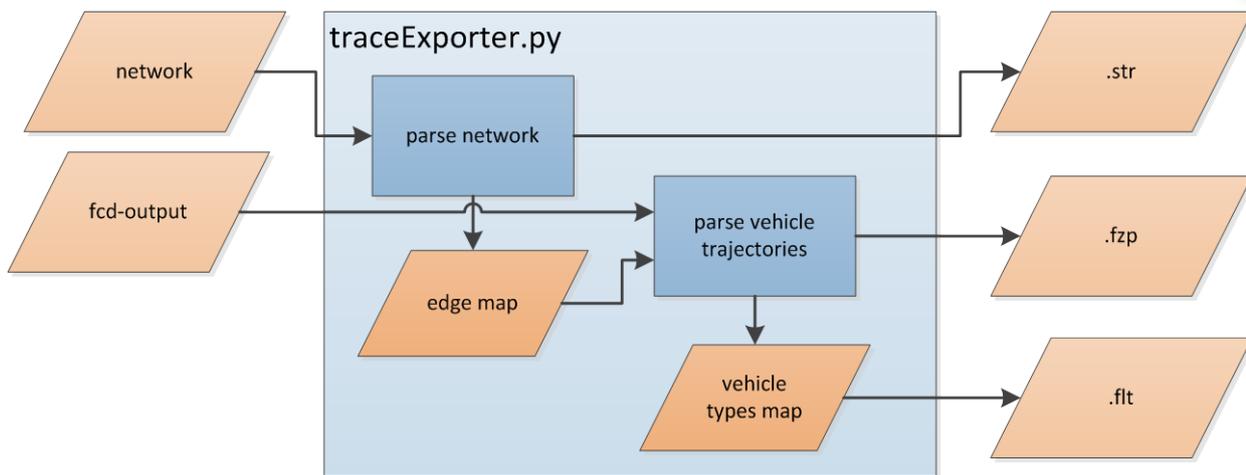


Figure 3.3: Data flow and dependencies within traceExporter.py when generating files for PHEM’s “advanced mode”

To obtain PHEM’s “minimal mode” files, the given fcd-output is parsed and written in the according format.

3.2.3 Functionality Extensions

Viewed at a coarse scale, PHEM retrieves speed and position updates of vehicles to regard per second. This type of information is similar to so-called “trace files” – position time-lines used as inputs for communication simulators. SUMO included a tool for generating such trace files by converting SUMO’s “net-dump” outputs into files that can be read by ns-2. The tool itself was outdated and due to being implemented in Java, only badly maintainable in SUMO context, as SUMO has no build infrastructure for Java.

As a logical step, the newly implemented script was extended by other export modules allowing to generate input files of the following type:

- OMNET: mobility-traces (.xml)
- Shawn: snapshot-files (.xml)
- ns2/ns3: trace-files, activity-files, and mobility-files (all *.tcl)
- PHEM: .dri-files, .str-files, .fzp-files, .flt-files
- GPSDAT (undefined extension)
- gpx (.gpx)
- SUMO’s points-of-interest (PoIs, .xml)

These export functionalities allowed to abandon the initial trace exporter, replacing it by the new implemented one, which is assumed to be easier maintainable and supports a larger number of target applications. The script has been extended by further functionalities, such as:

- `--penetration <AMOUNT>`: A float between 0 and 1, defining the probability of a vehicle to be included in the output
- `--gps-blur <METERS>`: A float defining a random offset applied on the vehicle positions before exporting them
- `--delta-t <TIME>`: Defines the time sampling; if given, only every <TIME> time step will be exported

3.3 Performance Evaluation

SUMO's "fcd-output"-files may get quite large. Table 3.1 shows some parameter of selected simulations. It also shows the computation times needed to obtain the PHEM files on a PowerEdge R420 TPM server (Intel Xeon E5-2470 2.30GHz CPU).

Table 3.1: Characteristics of the off-line Coupling in means of File sizes and Simulation Time

Scenario	Covered demand	Number of Vehicles	Size of FCD-Output	Plain run duration ¹³	Duration with writing FCD output ¹⁴
iTETRIS/joined	~1 h	~11,000	~744 MB	~11 s	~81 s
ORINOKO/VLS	1 day	~216,000	~43 GB	2673 s	6724 s

The numbers show that the off-line coupling increases the simulation time and that the generated files get very large very fast. For time critical tasks, using this kind of emissions computation should be avoided.

Of course, different possibilities to reduce the files sizes exist. Just to name a few: a) the information not needed as PHEM inputs could be removed from SUMO's fcd-output, b) PHEM input files could be generated directly in SUMO, c) one could think of using binary files to store the fcd-information. As there is currently no need for using the off-line coupling for time-consuming tasks, no attempts to improve this issue have been done.

3.4 Deployment

Availability

The conversion script named "traceExporter.py" was made public on 15th of January 2013 and will be included in the following SUMO-releases (under the GPL). The first SUMO-version including the script was 0.17.0, released on the 3rd of May, 2013.

System Requirements

Being written in Python 2.7.x, the script can be executed on all platforms for which Python is available. That is true for all common platforms, including MS Windows, Linux, and Mac OS. The script directly parses the entries step-by-step, so that no limits arising from memory consumption should be assumed.

User Documentation

The documentation on using the script was added to the SUMO documentation. The script itself is described at <http://sumo-sim.org/userdoc/Tools/TraceExporter.html>. Additionally, a tutorial on exporting trace files, including those readable by PHEM was added and is available at <http://sumo-sim.org/userdoc/Tutorials/Trace File Generation.html>. Both are included within this report in Appendix B.

¹³ Averaged over ten simulation runs

¹⁴ Averaged over ten simulation runs

Tests

The script is tested every night as a part of the SUMO test suite. The nightly test system builds the SUMO applications using the current SVN state, first. Then, acceptance tests are performed, where each test consists of executing an application with a certain set of input files and command line parameters. The resulting output on the command line, as well as the files generated by the application, are then compared to ones that were accepted while building the test case. The SUMO suite includes about 2500 such tests, extended by additional unit tests, complex tests that incorporate more than one application and some internal tests for data that cannot be given to the public due to licencing reasons.

The “traceExporter.py” script is tested for:

- properly formatting the files for
 - gpsdat (one test)
 - ns-2 (same as ns-3, four tests)
 - OMNET (one test)
 - Shawn (one test)
 - gpx (one test)
 - SUMO PoIs (one test)
 - PHEM (seven tests)
- pruning the traces by setting the begin, the end times, and time steps (seven tests)

4 PHEMlight – PHEM Derivative embedded in SUMO

In comparison to generating vehicular emissions by coupling SUMO and PHEM off-line, an on-line interaction is assumed to bring the following benefits:

- **reduction of the overall system’s complexity:** instead of executing multiple applications one-by-one, the simulation system directly returns the measurements to use
- **on-line retrieval of emission measurements:** the computed measurements can be read on-line, in each step of the simulation’s progress. This allows to implement a “reactive” use of those measurements: use them as an input to other applications that perform certain actions (vehicle rerouting, e.g.) immediately on the running simulation
- **lowering the amount of used storage:** no intermediate files must be stored, saving disc space (see section 3.3)
- **faster execution:** all values needed to compute vehicular emissions are passed using simple function calls, instead of the need to store and read values as done within the off-line coupling

The chosen approach, presented in the following, uses a simplified model of PHEM named “PHEMlight” that is directly included into SUMO’s simulation core. In the following, the implementation details, first evaluation and usage examples, as well as information about the implementation’s availability are given.

4.1 Outline

The main purpose of PHEMlight is to provide a simple but accurate simulation technique to derive modal emission quantities for given speed and acceleration demands. Besides, it should also offer functionality to assess driving trajectories whether they can be realistically achieved. To avoid the need for history driving trajectories (speed and acceleration) transient dynamic corrections, temperature influences on after-treatment-systems and, most important, the driver gear shift model to compute engine speeds were not included in the simulation. Hence, a quite simple and straight forward interface to SUMO has been implemented. This was indispensable to realize an on-line coupling with SUMO in the first place.

The interface to SUMO represents and combines the two basic capabilities of PHEMlight:

- Computing emission pollutants and fuel consumption
- Evaluating acceleration and power demands

For each emission pollutant (NO_x, CO₂, CO, HC and PM) and fuel consumption a method was implemented which takes the following parameters:

- SUMO emission class (SUMO emission classes are a superset to PHEM emission classes)
- Speed in [m/s]
- Acceleration in [m²/s]
- Slope [°]

The emission class comprises information on the shape, weight and default loading of a vehicle (passenger car, heavy duty vehicle, motorcycle...), as well as engine related properties, for example rated power, fuel type and most important the actual emission class (EU1, EU2 and so on). The handling of different loadings, especially for buses, coaches and trucks, is still an issue open for discussion. However, the principle methodology of PHEMlight allows simulating vehicles with different loadings, without the need of a new data base. Hence, this matter will affect implementations of interfaces between SUMO and PHEMlight without changing the design and algorithms presented here.

To determine whether a driving trajectory is realistic a method that computes the maximum acceleration for a given driving situation will be implemented. The driving situation is characterized by the same parameters as listed above; however, the acceleration denotes the current acceleration, whereas the maximum acceleration is returned by the method. Currently, two principles to compute acceleration bounds without the knowledge of the current engine speed are under investigation.

4.2 Implementation

The implementation of PHEMlight may also be divided in two separate parts, transient computations, such as maximum acceleration and power demand calculations, and the handling of emission and vehicle data. Both parts are then combined by the methods which implement the interface to SUMO.

4.2.1 Transient Computations

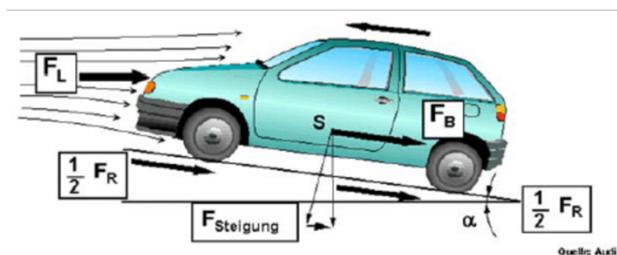
The implementation of the computation of maximum acceleration and power demands are relatively straight forward and, hence, nested directly in the “HelpersPHEMlight” class. The aggregate power demand for a specific driving situation depends on:

- Vehicle Parameters (road load coefficients, mass, loading...)
- Speed
- Acceleration
- Slope

Further, it can be portioned in four different demands:

- Power needed to overcome the rolling resistance
- Power needed to overcome the air resistance
- Power needed for desired acceleration
- Power to overcome the road gradient.

$$P_e = (P_{\text{rolling resistance}} + P_{\text{air resistance}} + P_{\text{acceleration}} + P_{\text{road gradient}}) / \eta_{\text{gearbox}}$$



$$P_R = (m_{\text{Vehicle}} + m_{\text{Load}}) \times g \times (Fr_0 + Fr_1 \times v + Fr_4 \times v^4) \times v$$

$$P_{\text{Air}} = (Cd \times A \times \frac{\rho}{2}) \times v^3$$

$$P_a = (m_{\text{Vehicle}} + m_{\text{Rot}} + m_{\text{Load}}) \times a \times v$$

$$P_{\text{grad}} = (m_{\text{Vehicle}} + m_{\text{Load}}) \times \text{Gradient} \times 0.01 \times v$$

$$\eta_{\text{gearbox}} = 0.95 \text{ (average efficiency)}$$

Figure 4.1: Calculation of power demand

Each individual demand can be computed by the formulas illustrated in Figure 4.1. The losses in the drive train are assumed to be constant at 5%.

4.2.2 Emission and vehicle data handling

Relevant information on vehicle parameters and emission behaviour are kept in so called CEP (Characteristic Emission curves over Power) classes (one per PHEM emission class). These classes compromise information on emission data files and vehicle files, which are read upon the first usage of the corresponding PHEM emission class. The CEP instances holding all information are themselves managed by the “PHEMCEPHandler”. The PHEMCEPHandler is designed as a Singleton and its only interface to PHEMlight is the GetCEP method. This method takes only the

PHEM emission class as parameter and returns the corresponding CEP instance. Internally it checks whether a request for the same emission class has been filed before, or loading of corresponding files is necessary. This ensures that files that are not used during the entire simulation are not loaded at all and, thereby, saves storage and computation time. Figure 4.2 show this basic principle.

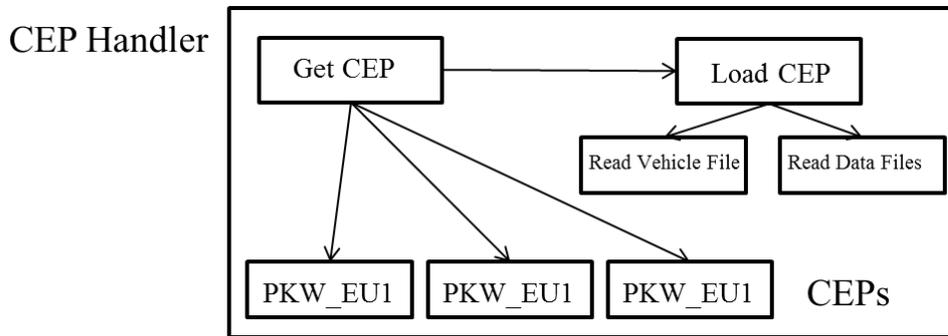


Figure 4.2: Structure of PHEMCEPHandler

In the latest version of PHEMlight each CEP instance combines the information read from three files:

- Fuel consumption data file
- Emission data file
- Vehicle data file

The vehicle data file contains all relevant information on the vehicle, as for example vehicle mass, engine rated power and road load coefficients. It has the same structure and form as vehicle files for the full version of PHEM, but is extended by one more parameter. In PHEMlight vehicles are additionally classified as light or heavy vehicles. Light vehicles are marked as LV at the end of a vehicle file and include passenger cars, motorcycles and light-duty vehicles. Heavy vehicle are basically heavy-duty vehicles and include buses, trucks and so on. They are marked as HV in the vehicle file.

The emission and fuel consumption data files contain emission and fuel consumption levels for the whole range of normalized power demands. Upon reading by the PHEMCEPHandler the normalized power demands are de-normalize to absolute power levels in [kW]. The normalization process is required to combine several vehicles and emission behaviours into one average vehicle per PHEM emission class. Further, it allows varying vehicle weight, rated power and driving resistances without adopting the characteristic emission curve, what makes PHEMlight easily extendable to varying vehicle parameters. One feature of PHEMlight is that this de-normalization process is done differently for heavy and light vehicles. This is even an extension to the full version of PHEM, where no diversification is implemented. The two different principles are:

1. Normalization over rated power (NRP)
2. Normalization over critical power (NCP)

Ad1.) Normalized power measurements over rated power are gained by dividing the actual power by the rated power of a vehicle and the other way around for de-normalization. The normalized power is obviously bounded by one. It can get negative while motoring. However, negative values are smaller than positive values and a lower bound of -0.4 can be assumed without the risk of omitting realistic power regions. This method is mainly applied for heavy vehicles, where type approval emission limits are given in [g/kWh].

Ad2.) For normalizing per critical power, measurements are divided by a vehicle specific critical power value. This value results from the power demand at 70 km/h speed and 0.45 m/s^2

acceleration. This critical power is dependent on vehicle parameters as mass, cross-sectional area, loading, cd value and others. On the other hand, it is fairly independent from engine parameters. For light vehicles emission behaviour has often a critical point in the power range. Above this point emissions can increase dramatically. This is a result of the type approval test cycle that results in a limited power demand. Manufactures design their vehicles to perform well (emit less) exactly up to this point. As this critical point is correlated more to vehicle parameters than to engine parameters it makes sense to use a normalization that reflects this behaviour and does not distinguish between the same vehicles with different engines. As the normalized values are not a function of the rated power any more they are not bounded by one. Typically, the normalized rated power is in the range of 5-8 and negative values smaller than -3 do not occur. Note that this normalization procedure has great benefits for light vehicle and emission pollutants other than CO₂ but does not fit other calculations. Table 4.1 shows which normalization method fits best different types of calculation.

Table 4.1: Usage of Normalization Techniques

	Fuel Consumption	CO ₂	Other Pollutants (NO _x , HC)
HV	NRP	NRP	NRP
LV	NRP	NRP	NCP

To avoid reading different power ranges and corresponding emission levels from one file, emission data is read from two separate files, one for fuel consumption and one for other pollutants. To ensure a consistent design this is also done for heavy vehicles where both files are generated by the same normalization procedure. Hence, the data loading process consists of the following steps:

1. Load vehicle file and assess if vehicle is light or heavy
2. Load fuel consumption file, de-normalize power using NRP
3. Load other emission pollutant file, de-normalize using either NRP or NRC
4. Create CEP object that holds emission and vehicle data

The CEP class is designed for holding emission and vehicle data. Emission data is hold in vector like structures, where entries represent emission levels for corresponding power levels. Due to the different power patterns, resulting from different de-normalization procedures, two separate structures for fuel consumption (FC) and other pollutants are held. An example for NO_x CEP data is given by Figure 4.8.

The CEP class provides the method “GetEmission” which serves as an interface. It takes as parameters the emission identifier (FC, CO₂, NO_x...) as a string and the power as a double. It returns the corresponding emission level. First, the corresponding emission data set is looked up. Second, the two bounding power levels (next bigger and last smaller level than the requested power) and the corresponding emission measurements are determined. Third, the emission level for the actual power demand is interpolated between the two bounding levels. In the very unrealistic case that the demanded power is smaller or bigger than the extreme values of the power pattern in the CEP, the two last, or first, values are extrapolated.

4.2.3 Implementation of the Interface to SUMO, Combining Data Handling and Transient Computation

As described above the interface to SUMO is a set of methods for retrieving emission levels and maximum acceleration for specific driving situations. The implementation of these methods (in the `HelpersPHEMLight` class) combines the two separate blocks of PHEMLight, data handling and transient calculations.

Upon a request for the emission level of a driving situation the power demand is computed using the described method. Further, the corresponding CEP instance is retrieved by the CEPHandler and used to return the emission level for the power demand.

The overall structure of PHEMlight is illustrated in Figure 4.3.

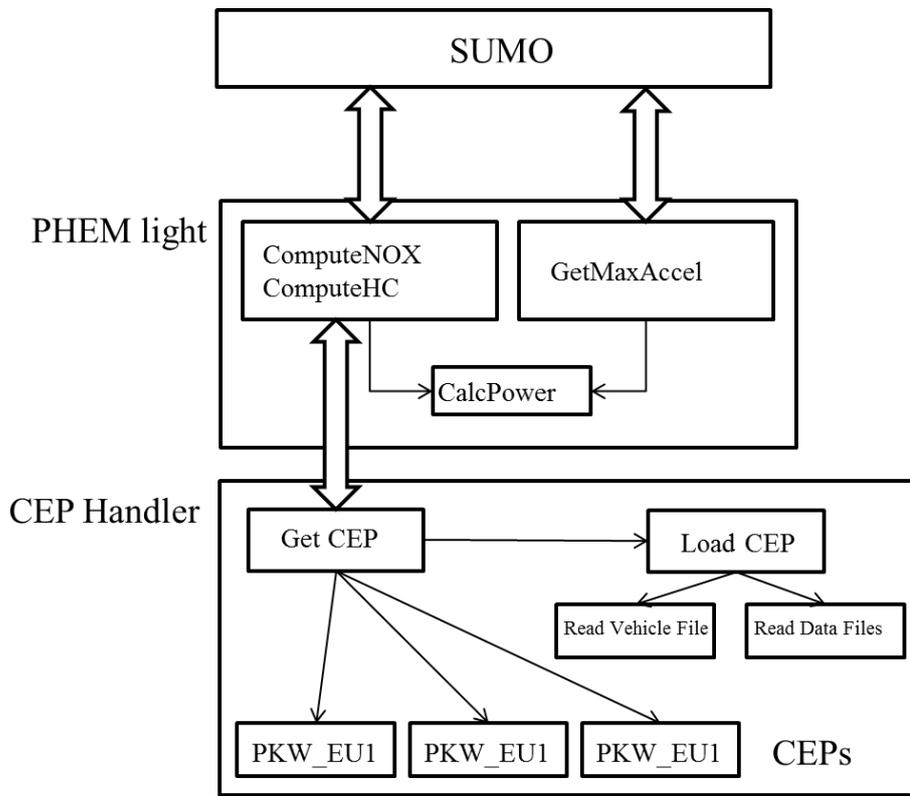


Figure 4.3: Structure of PHEMlight

4.2.4 Computation of Maximum Acceleration

Using the vehicle data from the data handling segment and the power calculating routine the following method to determine if a requested acceleration level can realistically be reached for a specific driving situation is proposed.

First, the power demand for acceleration is extracted from the method to compute the overall power demand:

$$P_a[kW] = (m_{curb} \cdot \Lambda + m_{loading} + m_{rot,wheels}) \cdot a \cdot v \cdot 0.001, \quad (1)$$

with:

a.....vehicle acceleration [m/s²]

v.....vehicle speed [m/s]

m_{curb}.....vehicle curb weight [kg]

m_{loading}.....vehicle loading [kg]

m_{rot,wheels}equivalent rotational mass of wheels [kg]

Λ.....rotating mass factor calculated by linear interpolation from a polygon $\Lambda = f(v)$ ¹⁵ [-]

¹⁵ In PHEM light the rotating mass factor is defined slightly different from literature and comprises only the rotational inertias from engine and drivetrain (but not of the wheels)

To define a polygonal representation for the rotating mass factor Λ a representative cycle (e.g. ERMES test cycle for passenger cars) was simulated by the full version of PHEM for each vehicle class. This data was used to calculate average speeds by gears for all vehicle classes. Further, the full load simulations of PHEM and formula (1) provide an approximation of rotational mass correction factor during maximum acceleration for each gear. Based on these value pairs a piecewise linear function was defined to approximate the rotating mass factor Λ .

Next, to get the maximum acceleration possible for a specific driving situation (speed and slope) the maximum power demand available for acceleration, $P_{a,max}$ has to be calculated. Therefore, the same methods to compute the overall power demand are used. The resulting demand for a 0-acceleration is subtracted from the maximum power available for the required speed level:

$$P_{a,max}[kW] = P_{rated} \cdot P_{max,norm} - P_{transm} - P_{roll} - P_{air} - P_{grd} \quad (2)$$

with:

- P_{rated} engine rated power [kW]
- $P_{max,norm}$ available share of rated power calculated by linear interpolation from a polygon $P_{max,norm} = f(v)$
- P_{aux} transmission losses [kW]
- P_{roll} power demand from rolling resistance [kW]
- P_{air} power demand from air drag [kW]
- P_{grd} power demand from road gradient [kW]

Thereby, we assume that the available share of rated power is a function over driving speed only, neglecting engine speeds. Certainly, this is a simplification, since the maximum available power depends on the actual engine speed (low max. power at low engine speed up to rated power at rated engine speed). Since PHEMlight excludes the simulation of the engine speed, average relations between vehicle speed and engine speed have been used to compute the actual maximum power. The actual power available is gained from the function shown in Figure 4.4 and depicts the lower level of engine speeds at full load acceleration. It has shown, that this “lower area of full load behaviour” leads to quite realistic results for maximum acceleration levels since in real driving torque interruptions during gear shifts occur and the drivers also rather seldom accelerate the engine up to the rated engine speed.

The “average maximum available share of rated power over driving speed” was approximated using the full version of PHEM that includes engine speed and gear shift models. The full version of PHEM attempts to follow a given speed profile as good as possible. Hence, by simulating a speed profile that jumps from zero speed instantly to a none-reachable speed level (e.g. 250km/h) provides to approximate available power values. These values were approximated by a linear function over an interval of relatively low driving speeds and constant above, as shown by Figure 4.4.

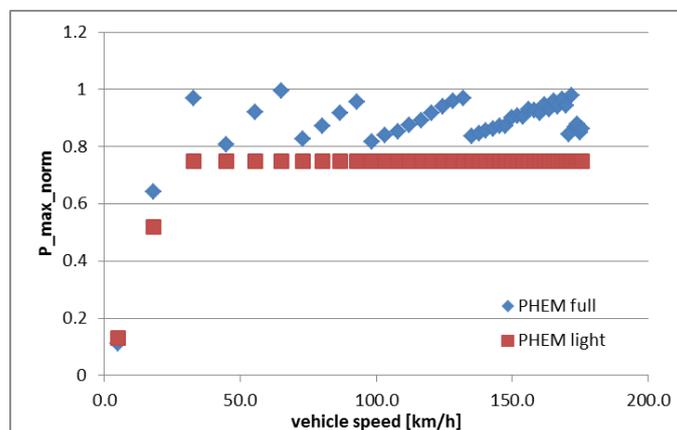


Figure 4.4: Maximum Normalized Rated Power

As one can see, the lack of an engine speed signal (in contrast to the full version of PHEM) leads to an underestimation of the power values available. However, since in real driving behaviour the aforementioned effects of torque interruption and early gear shifts reduce the maximum power during acceleration compared to the ideal full load, the result shall give a realistic picture of the acceleration levels a vehicle can achieve with a normal driver.

Finally, we can express the maximum acceleration in terms of $P_{a,max}[kW]$ and vehicle parameters using formula (1):

$$a_{max} \left[\frac{m}{s^2} \right] = \frac{P_{a,max} \cdot 1000}{(m_{curb} \cdot \Lambda + m_{loading} + m_{rot,wheels}) \cdot v}$$

Figure 4.5 shows the approximated maximum acceleration values by the full version of PHEM and the ones for a EU5 Diesel passenger car derived by the method described in this section. PHEM light produces slightly lower values, what is desirable as it is well known that the full version shows the ideal driver.

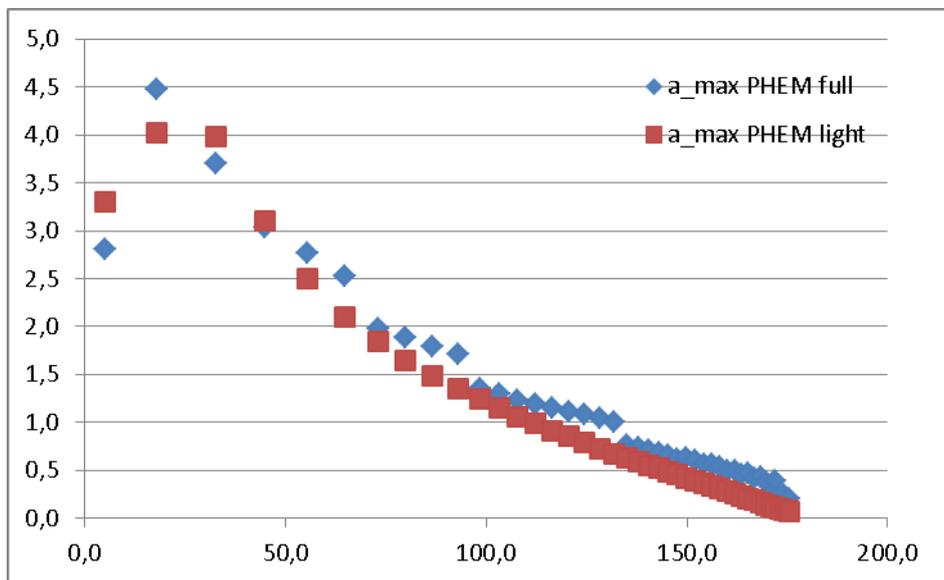


Figure 4.5: Comparison of maximum acceleration of PHEM light with full version of PHEM

4.2.5 Data Generation and Nomenclature Conventions

To produce emission curves (CEP curves) over power two things are necessary: the definition of average vehicles and the assessment of their emission behaviour. The definition of average vehicles could be directly imported from the full version of PHEM. In the near future it will also be extended by vehicles representing the fleet in 2020. Even formats of vehicle files are almost compatible, for PHEMlight they have to be extended by the HV/LV indicator.

For emission data new routines had to be implemented. The full version of PHEM uses engine maps that reflect emission behaviour over engine power and engine speed. To generate emission curves only over engine power one could transform these engine maps to CEP curves by computing distributions over engine speed. They could then be used as weighting factors to compromise different states of engine speed to a single measurement per engine power level. However, this approach appears to be cumbersome as it would require the definition of a representative cycle to generate engine speed distributions via simulation over this cycle and apply them to modify engine maps. Instead, one could use the PHEM-simulation results directly to rasterize power and emission signals and compute emission curves over engine power. This also ensures that only modal averages instead of averages over different cycles are used for CEP curve generation what fits the

needs of PHEMlight better. As mentioned above, this method requires the definition of representative test cycles to simulate average vehicles using the full version of PHEM. For light vehicle the ERMES cycle, developed by the TU Graz, was used. It consists of almost equal shares over time for urban, road and motorway driving and attempts to reflect reality better than the official type approval cycle NEDC or other cycles. Especially the NEDC cycle has been strongly criticized lately, as it omits motorway driving almost entirely and does not contain full load acceleration. However, the ERMES cycle covers all power regions of average light vehicles in real driving situations, has reasonable shares of representative driving conditions and therefore suits the needs of this project perfectly, ([Knörr, 2011]). Figure 4.6 shows the speed profile of the ERMES cycle.

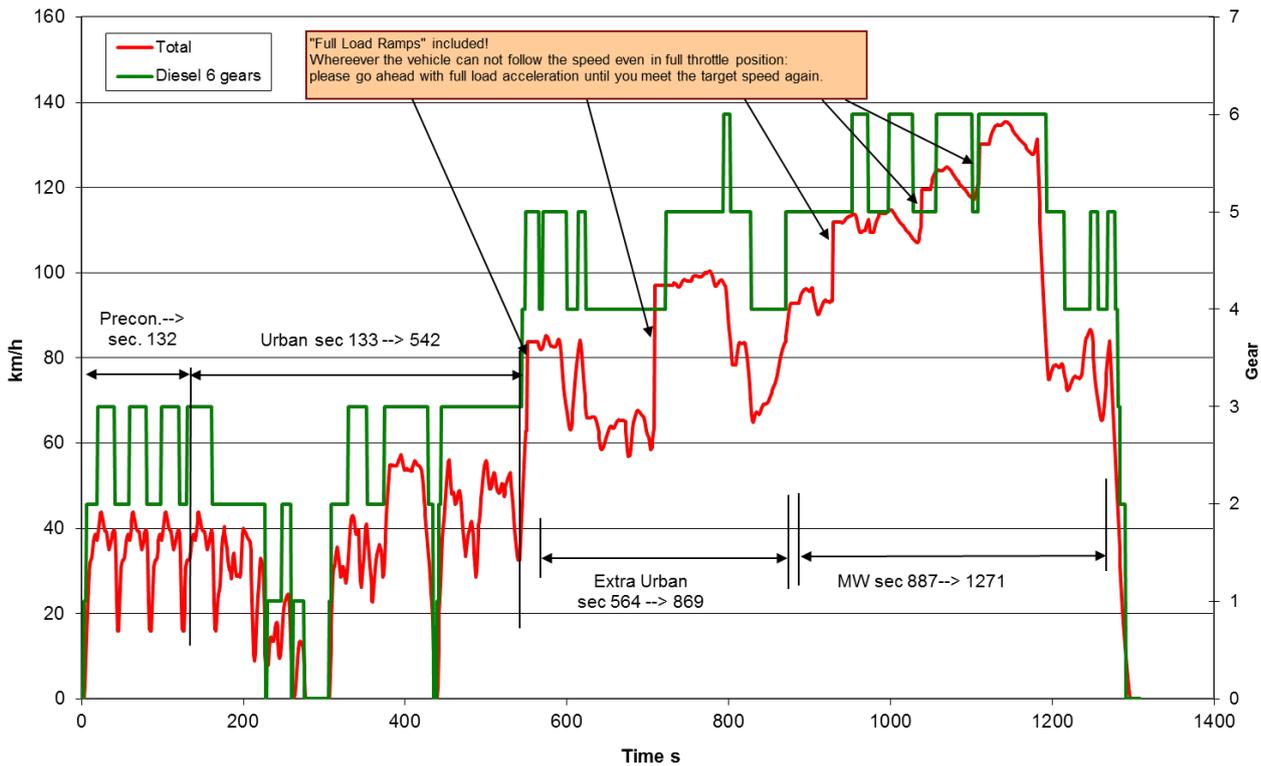


Figure 4.6: Speed Profile of the ERMES Test Cycle, gear position given for a diesel car with a 6 gear box

For heavy vehicles the ERMES cycle does not fit the requirements as they might not be able to perform the required acceleration to achieve the goal speeds at all times. Within in-use tests determining real world emission behaviour of heavy vehicles often the “TNO HD” speed profiles are used. These vehicle speed profiles have been elaborated by TNO Automotive (Netherlands) on basis of real world driving data and consist of three cycle parts (urban, rural and motorway). In this project the set of cycles fitting to a medium loaded heavy vehicle (ratio of rated power in kW to total vehicle mass in tons of 12.5) have been used and combined to a single cycle. This cycle has proven to give realistic fuel consumption and emission behaviour for typical heavy vehicles. Figure 4.7 shows the speed profile of the combined TNO test cycle.

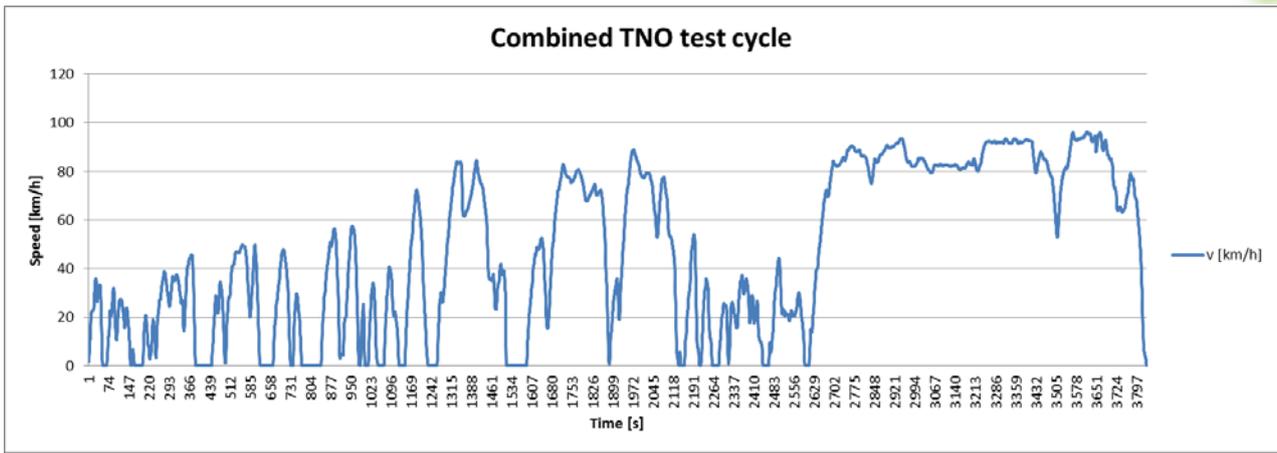


Figure 4.7: Speed Profile of the TNO Test Cycle

Both test cycles have been simulated with corresponding average vehicles using the full version of PHEM. Modal results have been normalized using the appropriate method and were partitioned into pre-defined normalized power intervals. Within these intervals average emissions were computed and used as a basis for the CEP curve generation. For intervals where no modal results could be assigned two cases were considered. First, if empty intervals were bounded by intervals with simulated data, average emission values were linearly interpolated. Second, if empty intervals occurred at the lower or upper end of the normalized power spectrum, the nearest (in terms of power regions) three simulated average emissions of intervals were used to perform a linear regression. The resulting linear function was used to extrapolate average emissions for empty power intervals. Figure 4.8 gives an example for the CEP data of diesel and gasoline passenger cars.

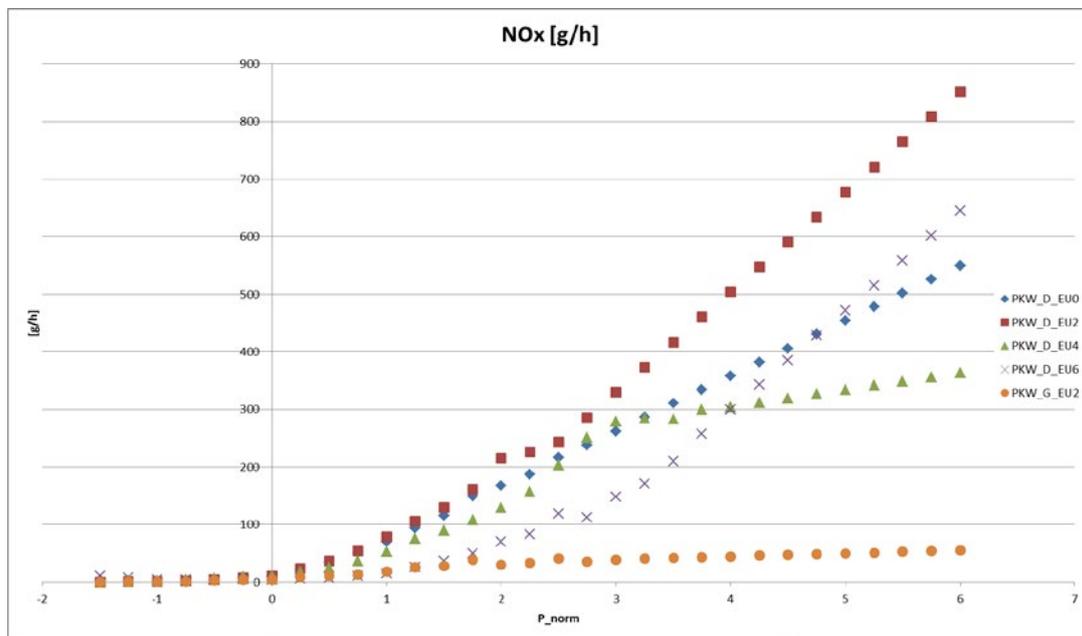


Figure 4.8: Example of CEP data

4.2.6 PHEM Emission Classes

In the following the different PHEM emission classes, which also consist of a classification for vehicle classes, are explained and the schemata for their nomenclature is introduced. The most general classification of vehicles considers the diversification between light and heavy vehicles.

These two classes use different methods for de-normalization, as mentioned earlier. The class of light vehicle consists of

- Passenger cars (PKW),
- Light duty vehicles (LNF),
- Motorcycles (MR),
- Scooters (KKR),
- Hybrid passenger cars (H_PKW)

whereas, the class of heavy vehicles consists of

- Tractor/Trailer (LSZ),
- Coaches (RB),
- Urban and inter-urban buses (LB),
- Trucks (Solo_LKW).

The second classification is done regarding the used fuel type, diesel (D) or gasoline (G). Third, the actual emission classes (not to be confused with the PHEM emission class) are considered, those are EU0 up to EU6. Last, for light duty vehicles three different size classes (I, II and III) and for trucks two different size classes (I and II) are defined. The size classes for light duty vehicles are taken from legislation and are defined as follows:

- I: $VRM \leq 1305kg$,
- II: $1305kg < VRM \leq 1760kg$,
- II: $1760g < VRM \leq 3500kg$,

where VRM denotes the vehicle reference mass and is the sum of the vehicle's empty weight and 100kg uniform loading. For trucks two classes, trucks with two (I) and three axes (II), were defined during the process of generating average vehicles for PHEM light. In terms of mass, weight classes with highest number of admissions have been chosen as representatives:

- I: 6821 [kg] vehicle mass and 1577 [kg] vehicle loading,
- II: 12143 [kg] vehicle mass and 6559 [kg] vehicle loading.

Note that this is a classification chosen by the TU Graz during the design process of PHEMlight. The classification reflects typical vehicle classes but does not reflect any legislation. In general the user can define the vehicle weights and loading to local conditions if better data is available than the predefined values. PHEM emission classes are combinations of the different subclasses, in particular their abbreviations. The nomenclature for PHEM emission classes consists of the following elements, separated by “_”:

- Vehicle class identifier (e.g. LNF),
- Fuel identifier (e.g. D),
- Emission class (e.g. EU4),
- Weight class (e.g. I), optional.

For example, this leads to the overall identifier “LNF_D_EU4_I” for light duty diesel vehicles of size class I and emission class EU4. Obviously, not all combinations exist, e.g. only diesel heavy duty vehicles exist and weight classes are only supported for trucks and light duty vehicles. A full list of all currently integrated PHEM emission classes is given in Table 4.2 and Table 4.3 for light vehicles and respectively Table 4.4 for heavy vehicles.

Table 4.2: PHEMlight Emission Classes for Light Vehicles (1)

Passenger Cars	Light Duty Vehicles		
PKW_D_EU0	LNF_D_EU0_I	LNF_D_EU0_II	LNF_D_EU0_III
PKW_D_EU1	LNF_D_EU1_I	LNF_D_EU1_II	LNF_D_EU1_III
PKW_D_EU2	LNF_D_EU2_I	LNF_D_EU2_II	LNF_D_EU2_III
PKW_D_EU3	LNF_D_EU3_I	LNF_D_EU3_II	LNF_D_EU3_III
PKW_D_EU4	LNF_D_EU4_I	LNF_D_EU4_II	LNF_D_EU4_III
PKW_D_EU5	LNF_D_EU5_I	LNF_D_EU5_II	LNF_D_EU5_III
PKW_D_EU6	LNF_D_EU6_I	LNF_D_EU6_II	LNF_D_EU6_III
PKW_G_EU0	LNF_G_EU0_I	LNF_G_EU0_II	LNF_G_EU0_III
PKW_G_EU1	LNF_G_EU1_I	LNF_G_EU1_II	LNF_G_EU1_III
PKW_G_EU2	LNF_G_EU2_I	LNF_G_EU2_II	LNF_G_EU2_III
PKW_G_EU3	LNF_G_EU3_I	LNF_G_EU3_II	LNF_G_EU3_III
PKW_G_EU4	LNF_G_EU4_I	LNF_G_EU4_II	LNF_G_EU4_III
PKW_G_EU5	LNF_G_EU5_I	LNF_G_EU5_II	LNF_G_EU5_III
PKW_G_EU6	LNF_G_EU6_I	LNF_G_EU6_II	LNF_G_EU6_III

Table 4.3: PHEMlight Emission Classes for Light Vehicles (2)

Two-Wheelers	Hybrid Passenger Vehicles	Motor Cycles	
KKR_G_EU0	H_PKW_D_EU5	MR_G_EU0_2T	MR_G_EU0_4T
KKR_G_EU1	H_PKW_D_EU6	MR_G_EU1_2T	MR_G_EU1_4T
KKR_G_EU2	H_PKW_G_EU5	MR_G_EU2_2T	MR_G_EU2_4T
KKR_G_EU3	H_PKW_G_EU5	MR_G_EU3_2T	MR_G_EU3_4T
KKR_G_EU4		MR_G_EU4_2T	MR_G_EU4_4T
		MR_G_EU5_2T	MR_G_EU5_4T

Table 4.4: PHEMlight Emission Classes for Heavy Vehicles

Tractor/ Trailer	Coach	Urban and Inter Urban Buses	Trucks	
LSZ_D_EU0	RB_D_EU0	LB_D_EU0	SOLO_LKW_D_EU0_I	SOLO_LKW_D_EU0_II
LSZ_D_EU1	RB_D_EU1	LB_D_EU1	SOLO_LKW_D_EU1_I	SOLO_LKW_D_EU1_II
LSZ_D_EU2	RB_D_EU2	LB_D_EU2	SOLO_LKW_D_EU2_I	SOLO_LKW_D_EU2_II
LSZ_D_EU3	RB_D_EU3	LB_D_EU3	SOLO_LKW_D_EU3_I	SOLO_LKW_D_EU3_II
LSZ_D_EU4	RB_D_EU4	LB_D_EU4	SOLO_LKW_D_EU4_I	SOLO_LKW_D_EU4_II
LSZ_D_EU5	RB_D_EU5	LB_D_EU5	SOLO_LKW_D_EU5_I	SOLO_LKW_D_EU5_II
LSZ_D_EU6	RB_D_EU6	LB_D_EU6	SOLO_LKW_D_EU6_I	SOLO_LKW_D_EU6_II

As mentioned in previous sections, for each PHEM emission class three files have been generated and are loaded by PHEMlight; a vehicle file, emission data and fuel consumption data files. They are named according to their PHEM emission class, as illustrated by the following example for a EU4 diesel passenger car:

- Vehicle file: PKW_D_EU4.veh,
- Emission data: PKW_D_EU4.csv,

- Fuel consumption data: PKW_D_EH4_FC.csv.

Vehicle files have the extension “.veh”, emission and fuel consumption data files are comma separated “.csv” files, where fuel consumption is additionally indicated by the “_FC” extension of the file name.

4.2.7 Embedding into SUMO

This section describes the implementation of PHEMlight from SUMO’s viewpoint, including the realisation of input and output interfaces.

Prerequisites

SUMO already included an emission model before COLOMBO has begun. The model has been implemented within the iTETRIS project and was based on a continuous reformulation of the HBEFA database [iTETRIS D3.1, 2009]. The implemented emission model allows writing aggregated values for edges (roads), lanes, and for each vehicle’s drive through the network. It is also possible to obtain these measurements from the last simulation step via the on-line interface “TraCI”¹⁶ (see [Wegener et al., 2008]). Additionally, the generated measurements can be visualised in SUMO’s graphical interface, again by colouring vehicles and / or lanes by the amount of a certain pollutant’s emission the vehicle produces / that are emitted on the lane.

Each vehicle in SUMO is assigned to a “vehicle type” (vType). The vehicle type contains information about the car-following (longitudinal) and lane-change (lateral) movement models, including physical parameters, such as the vehicle’s length, maximum velocity, etc. The vehicle class additionally describes properties, such as the visual appearance of the vehicle, an abstract vehicle class (“bus”, “taxi”, “passenger”) used to model lane or road restrictions. Finally, the vehicle type includes an – also abstract – vehicle emission class. Within the initial emission model implementation, this vehicle emission class points to a set of coefficients that parameterise the continuous emission function used by the HBEFA reformulation. It should be mentioned, that within iTETRIS’ HBEFA reformulation different emission types were clustered into abstract emission classes, see [iTETRIS D3.1, 2009].

Besides generating emission values by the simulation the package was extended within iTETRIS by the possibility to read the saved emission values as network weights within the routing module. This extension allowed a large set of investigations aiming on clarifying whether an emission-based “ecological” routing is possible and how it behaves ([Krajzewicz, Wagner, 2011], [Flötteröd et al., 2012]).

Embedding PHEMlight is assumed to improve the quality of available emission computation, but for having a freely available model (see section 4.5 on licensing), the HBEFA reformulation will be kept.

Interfacing different Emission Models within the Simulation

The very first idea was to split the way the user interacts with the simulation at the level of emission models. The interfaces of the initial model were duplicated and adapted to PHEMlight’s needs. This approach has shown to be not proper, mainly because a seamless “switch” between different models would require a similar granularity of implemented emission classes. Then, the user could switch from one model to a different one, keeping his initial assignment of vehicles to emission classes. In practice, such similarities cannot be found; as mentioned, the initial HBEFA reformulation used

¹⁶ <http://sumo-sim.org/userdoc/TraCI.html>

clustered, abstract emission classes. Even a projection (mapping) from one classification to a different one seems to be hardly possible.

For this reason, the implementation was reworked. Now, the distinction of the emission model to use is made based on vehicle emission class names only. When performing the simulation both, the emission model itself as well as its parameters that resemble the wanted emission class, are determined based on the vehicle’s emission class. Technically, SUMO chooses the model to use based on the vehicle’s emission class and calls this model’s computation methods. Instead of calling methods from either the `PHEMlightHelpers`-class or the `HBEFAHelpers`-class, depending on the emission model to use, methods from the added class “`PollutantsInterface`” are called, and the distinction which model to use is done herein. This is schematically shown in Figure 4.9. It should be noted, that even though an additional interface (`PollutantsInterface`) is added, this change is assumed to make SUMO more easily extensible in the means of adding further emission models and also the number of overall methods is reduced, as less lower-level distinctions have to be implemented within the core functionality.

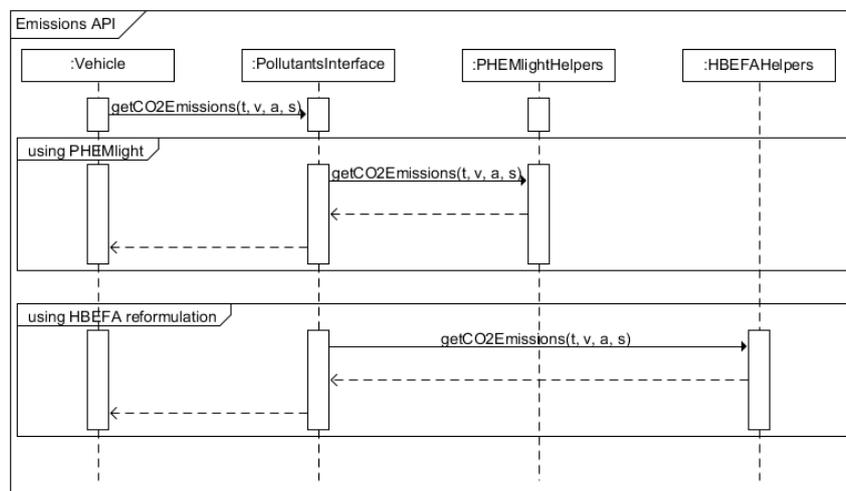


Figure 4.9: Distinguishing between the vehicle’s emission model to use in `PollutantsInterface`

This realisation has also a further advantage: as no high-level distinction of the emission model to use is performed, all other interfaces, such as the retrieval of emitted pollutants via TraCI or routing vehicles based on emissions can be kept as they are and work with all implemented models. In the initial PHEMlight embedding, all these interfaces would have to be reworked and extended as well.

4.2.8 User Interaction

PHEMlight is enabled by assigning one of its emission classes to a used vehicle type as mentioned before. The emission class names are same as given in Table 4.2, Table 4.3, and Table 4.4. A possible definition of a vehicle’s emission class may look like:

```
<vType id="myVehicleType" emissionClass="SOLO_LKW_D_EU6_II" ... further attributes .../>
```

Besides assigning PHEMlight emission classes to vehicle types, a path to read the PHEMlight definitions from must be configured. The usage of a variable path allows to store the used PHEMlight-CEPs wherever wanted and to compare different versions of those.

The path can be given to the applications by setting the environment variable “`PHEMLIGHT_PATH`” to the path. Additionally, it is possible to overwrite this information on the command line using the new option “`--phemlight-path`”. The option is available in both, the simulation (for computing the emissions of simulated vehicles) as well as within the routing modules (for computing emission-optimal paths).

It should be mentioned that this information is also given within the user documentation of the SUMO traffic simulation package (<http://sumo-sim.org>). In case of any changes, the user documentation will be kept updated.

4.3 Evaluation

The following assessments compare PHEMlight against the original PHEM. This has a two-fold motivation: On the one hand, an emission class (as given in PHEM and PHEMlight) covers a set of vehicles within a EURO exhaust gas legislation class with a defined engine type (diesel, gasoline, CNG, hybrid). Single vehicles can have very different emission behaviour compared to the class average (see e.g. www.hbefa.net). Therefore, a direct comparison to a certain vehicle’s emissions is not meaningful. In addition, per-vehicle emission measures are not available for all covered classes. On the other hand, the comparison against PHEM allows investigating how the simplifications done in PHEMlight change the simulation results for emission behaviour.

To obtain comparable data from both models, both were used to compute the amount of emitted pollutants by simulating commonly used emission driving cycles, namely the New European Driving Cycle (NEDC) and the Common Artemis Driving Cycle (CADC).

Aggregated View

In a first step, the aggregated emissions as the sum of a pollutant’s emissions during a whole driving cycle are compared. Figure 4.10 compares the emissions produced when driving the CADCs by both models. To stay brief, this figure shows only those vehicle classes that will be made publicly available.

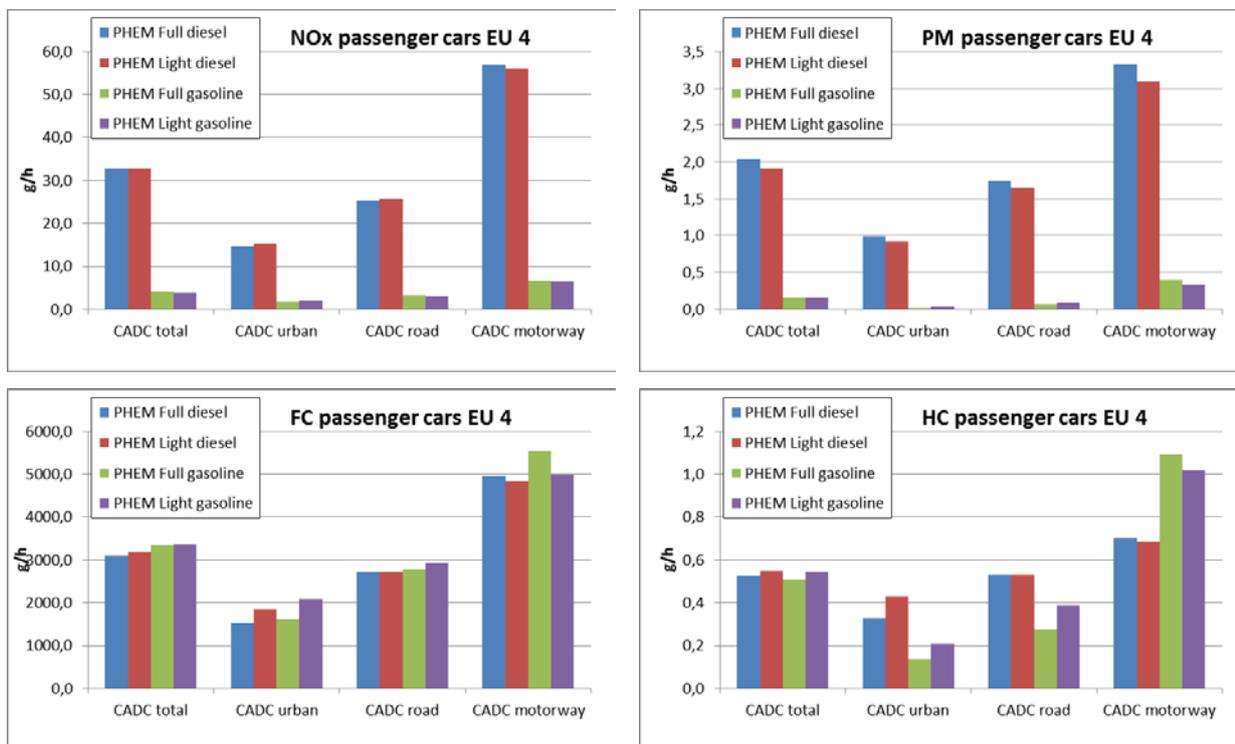


Figure 4.10: Comparison of emission behaviour as modelled in PHEM and in PHEMlight for passenger vehicles, Euro Norm 4, Diesel and Gasoline

Figure 4.11 puts the emission amounts computed by PHEM against those from PHEMlight including all emission classes for which a projection could be determined. One may note that driving cycles are not “normalized” to a certain behaviour or emission number. Therefore, Figure 4.11 contains the comparisons for three different driving cycles. Vehicle types can be distinguished within this Figure by the shape of the dots; passenger and light transport vehicles are shown as pentagons, busses as stars, all other vehicle types as triangles. The brightness of the dots indicates the Euro Norm the vehicle belongs to ranging from 0 (dark) to 6 (light). Vehicles using gasoline are coloured blue, Diesel-powered vehicles red.

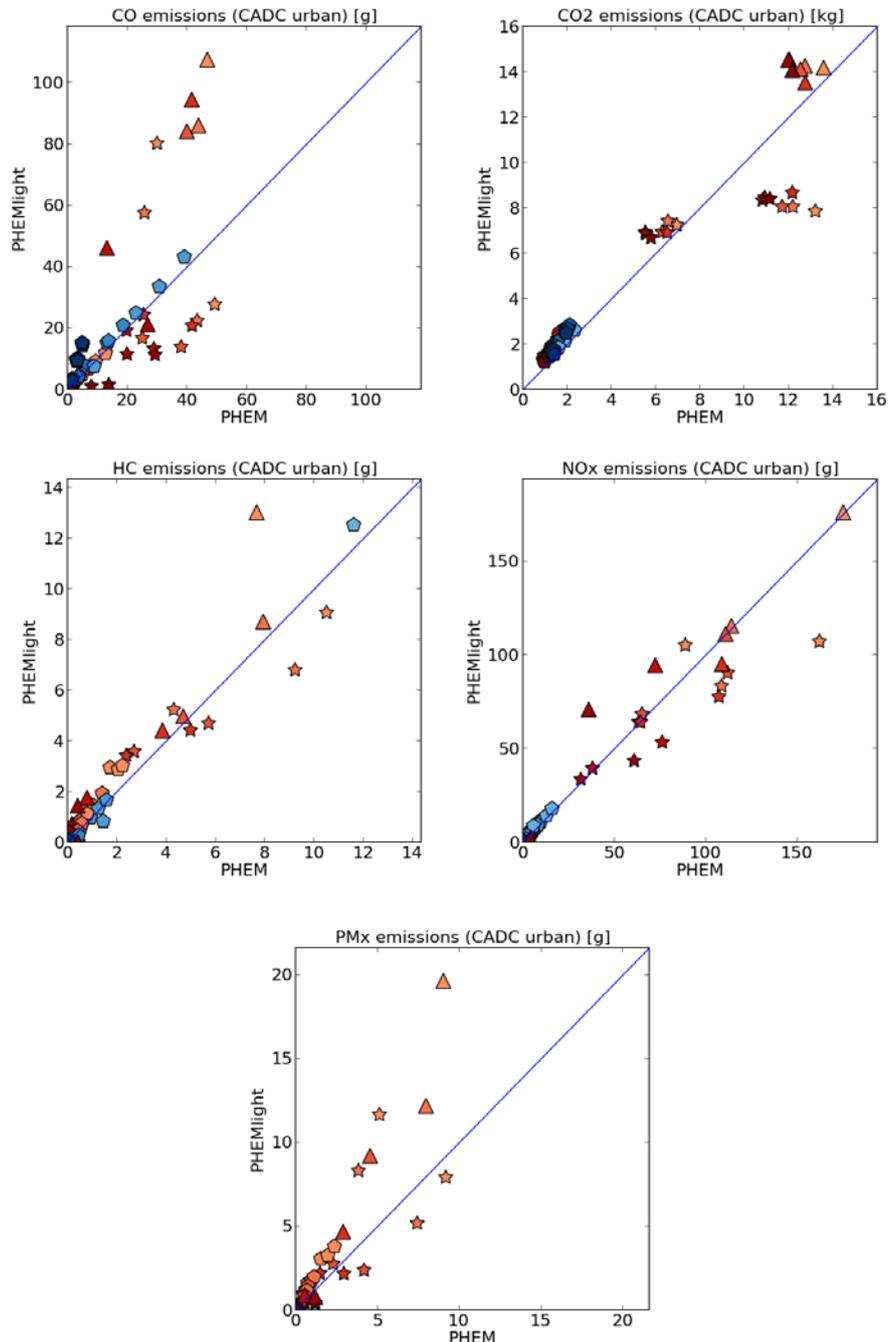


Figure 4.11: A comparison of emission amounts as computed in PHEM and PHEMlight; the explanation can be found in the text

Summarizing, one can state that most of the aggregated emission amounts obtained from PHEM are replicated by PHEMlight in a satisfying manner. CO and PM_x emissions for heavy duty vehicles (Diesel only) is overestimated by PHEMlight, the same counts for the CO emission of newer

(bigger Euro Norm) gasoline-powered light vehicles. The emission amounts of CO and CO₂ of some bus emission classes is underestimated by PHEMlight. Despite these outliers, the aggregated amounts could nonetheless be named fair and well usable.

Disaggregated View

While working on traffic management solutions that use per-second information from vehicles and control vehicles on a fine-grained scale, it is necessary to assure that the disaggregated emission behaviour is modelled correctly as well. A disaggregated view is given in Figure 4.12. Values correspond well between PHEM and PHEMlight, when taking into account the differences in the absolute amounts – the amounts of NO_x emissions produced by a gasoline vehicle are by a factor of ten smaller than the ones produces by a Diesel-powered vehicle.

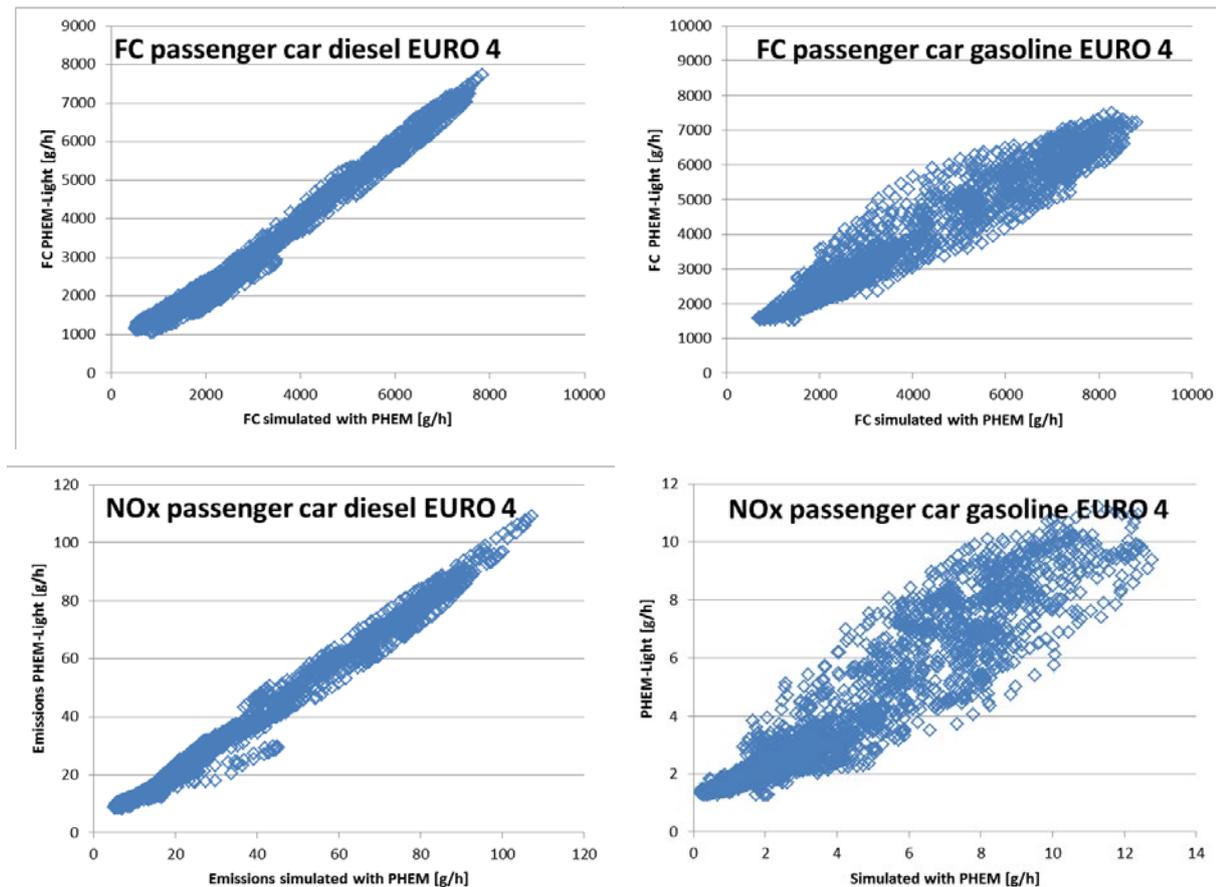


Figure 4.12: PHEM per-second emission measures put against same measures from PHEMlight (CADC)

Several other types of evaluations have been performed which compare all classes for which a projection between PHEM and PHEMlight exists. Figure 4.13 shows a comparison of CO₂ and NO_x emission behaviour for a gasoline-powered passenger vehicle, Euro Norm 5 (blue), and a heavy duty vehicle (Diesel) with the same norm (red). Both vehicles drive the urban CADC. Both, the time line of emissions computed by PHEM (solid line) as well as by PHEMlight (dashed line) are given. The figure is supposed to present two characteristics: the large span of emission amounts covered by the models (the passenger car’s NO_x emissions are even too close to zero to be visible) as well as the per-second relationship of the emission behaviour for both models.

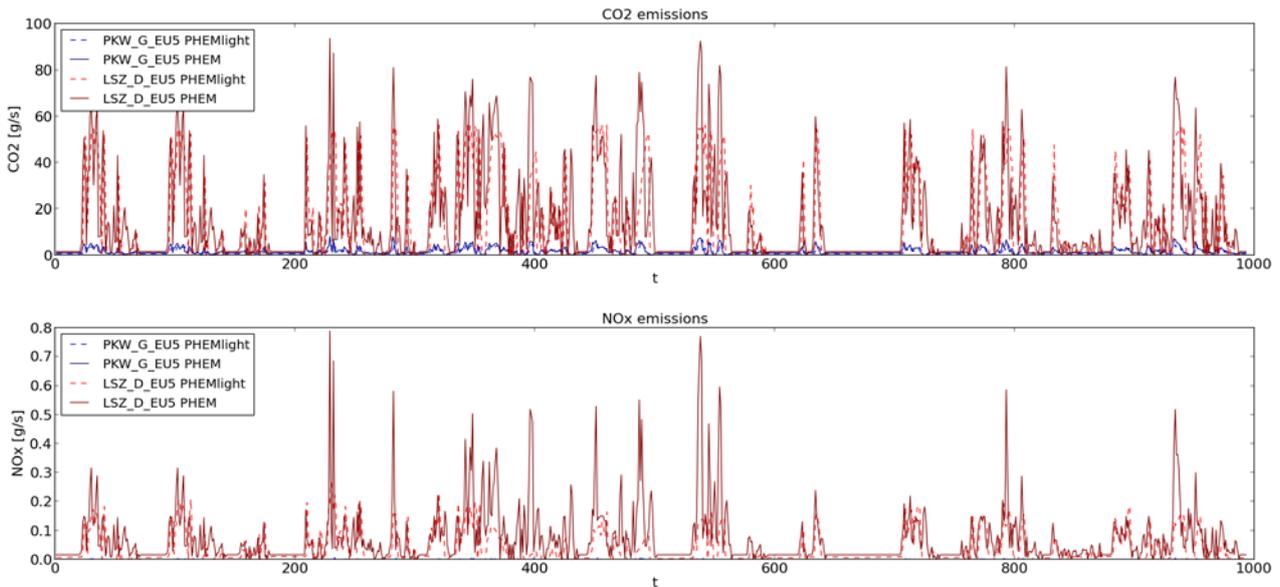


Figure 4.13: Per-second emission amount for a passenger vehicle (blue) and a heavy duty vehicle (red) for the urban CADC

In contrary to Figure 4.13, Figure 4.14 shows the same measurements but uses the urban part of the New European Driving Cycle (NEDC) and shows two different passenger vehicles as this cycle is not applicable to heavy duty vehicles.

Looking in per-step (1 s) results, larger deviations between the two models than for cycle aggregated numbers can be observed. This was expected, as certain driving conditions in PHEM are modelled in more detail (e.g. by driving in a particular gear) than in PHEM light. Nevertheless, PHEMlight is also able to predict the general characteristics of basic driving conditions (acceleration, constant speed driving at different speed levels, deceleration, standstill with idling) related to fuel consumption and emission behaviour. This capability is the main criteria for valid results of the application of PHEMlight in COLOMBO.

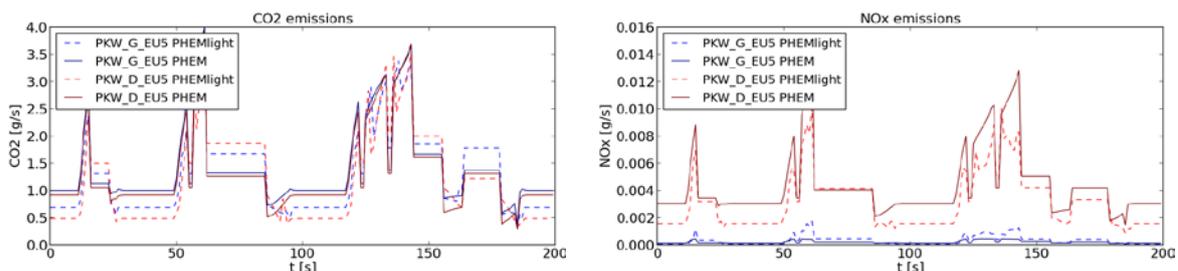


Figure 4.14: Per-second emission amount for a gasoline-powered (blue) and a Diesel-powered passenger vehicle (red) for the urban part of the NEDC

Summary

Given the fact that the model’s complexity has been reduced from a set of sub-models for gear choice, drive train losses, and catalyst behaviour to very simple look-up functions, PHEMlight’s results match the ones obtained from PHEM very well when comparing both on an aggregated scale.

The evaluation of per-step (1 s) emissions shows larger differences between PHEM and PHEMlight than cycle average results, however the main prerequisites for a valid application of PHEM light in COLOMBO were found to be guaranteed. Nevertheless, in the further course of the project,

conclusions made based on PHEM light will be benchmarked with PHEM full version results in order to guarantee the quality of the gained insights.

4.4 Real Use Case: Evaluation of Traffic Management Practices

PHEMlight has already been used within the Master thesis “Analysis and simulation of traffic management actions for traffic emission reduction” by Josep Tomàs Vergés ([Vergés, 2013]). The starting point of this thesis is MARLIS¹⁷, a database containing 3665 traffic management actions performed for reducing the ecological impact of traffic, mainly for fulfilling the regulations posed by the European Union in member states. Each traffic management action is described by naming the area, the type of the action (classified into topics, such as automotive engineering, fleet changes, infrastructure and construction actions, traffic management, public transport, etc.), the pollutants it should reduce, and, among others, the observed results.

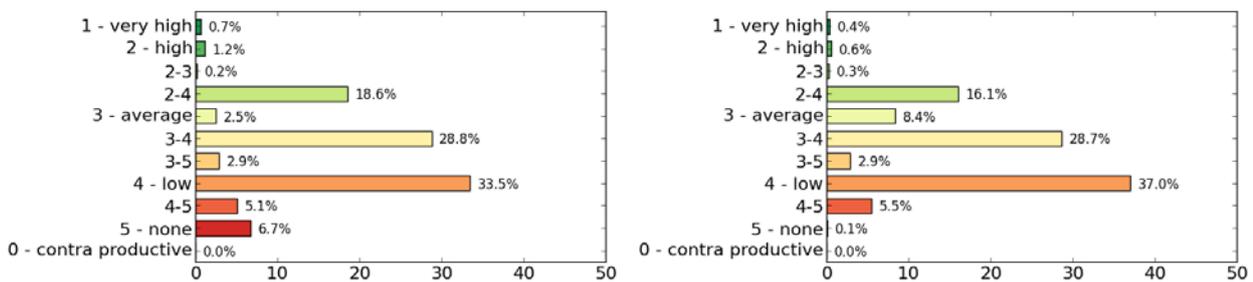


Figure 4.15: Reported effects of the traffic management actions listed in MARLIS on the emission of NO₂ (top) and PM₁₀ (bottom)

It is interesting that the majority of the performed actions (see Figure 4.15) has an effect on the environment below “medium” (further details can be found in the thesis or the reports on the MARLIS database). This motivated us to take a closer look at what happens after an action is implemented. For our simulations, we used SUMO, extended by a first version of PHEMlight and a simulation scenario that covers the city of Brunswick with the surrounding highway ring, see Figure 4.16. The scenario itself was generated within the DLR-project “AIM”¹⁸. It is based on a road network from NavTeq that was corrected based on aerial images and adapted to the simulation needs and the VALIDATE demand data from PTV AG, which covers about 600,000 passenger and 44,000 heavy duty trips, performed within a usual Tuesday-Thursday.

¹⁷ http://www.bast.de/nn_42544/DE/Publikationen/Datenbanken/MARLIS/MARLIS.html

¹⁸ http://www.dlr.de/fs/en/desktopdefault.aspx/tabid-6422/10597_read-37397/ (German)

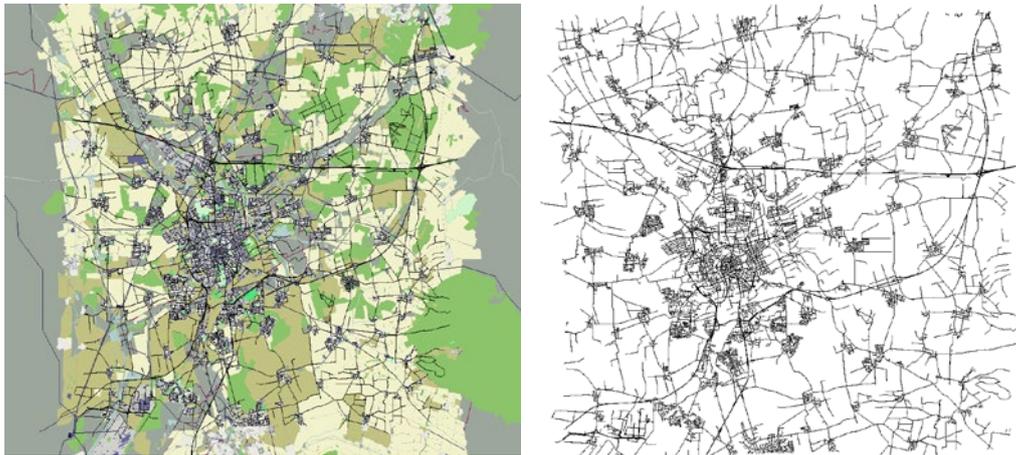


Figure 4.16: Brunswick scenario: simulated area (left: with additionally shown land use information; right: the road network only)

Within the thesis, three traffic management actions were implemented and evaluated: a reduction of the maximum velocity to 30 km/h in residential zones, a permissive and a restrictive environmental zone. The permissive environmental zone allows vehicles that fulfil the Euro 3 or Euro 2 + particle filter classes to enter while for entering the restrictive zone vehicles must at least fulfil the Euro 4 or Euro 3 + particle filter norm. The areas chosen for speed reduction and for the environmental zone are shown in Figure 4.17. The road network was accordingly modified, adapting the speed changes and the accessibility for certain vehicle types posed by the respective management action, respectively.

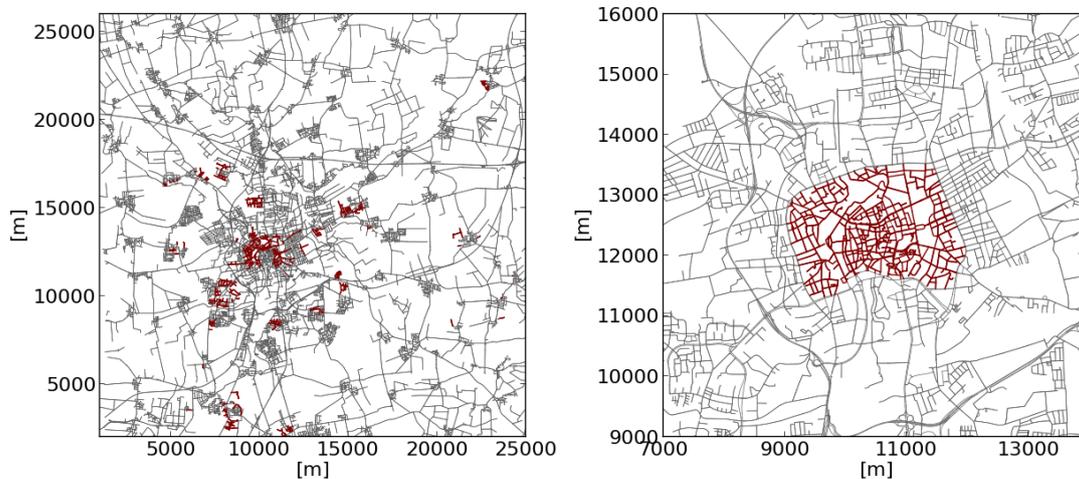


Figure 4.17: Influenced Areas; left: roads chosen as residential zones, limited to 30 km/h max, right: the chosen environmental zone

To accomplish this, the given demand had to be assigned to the current mix of emission classes on the roads, see Figure 4.18. This was performed by randomly assigning emission classes to vehicles using probabilities stored in HBEFA 3.1 ([INFRAS, 2010]).

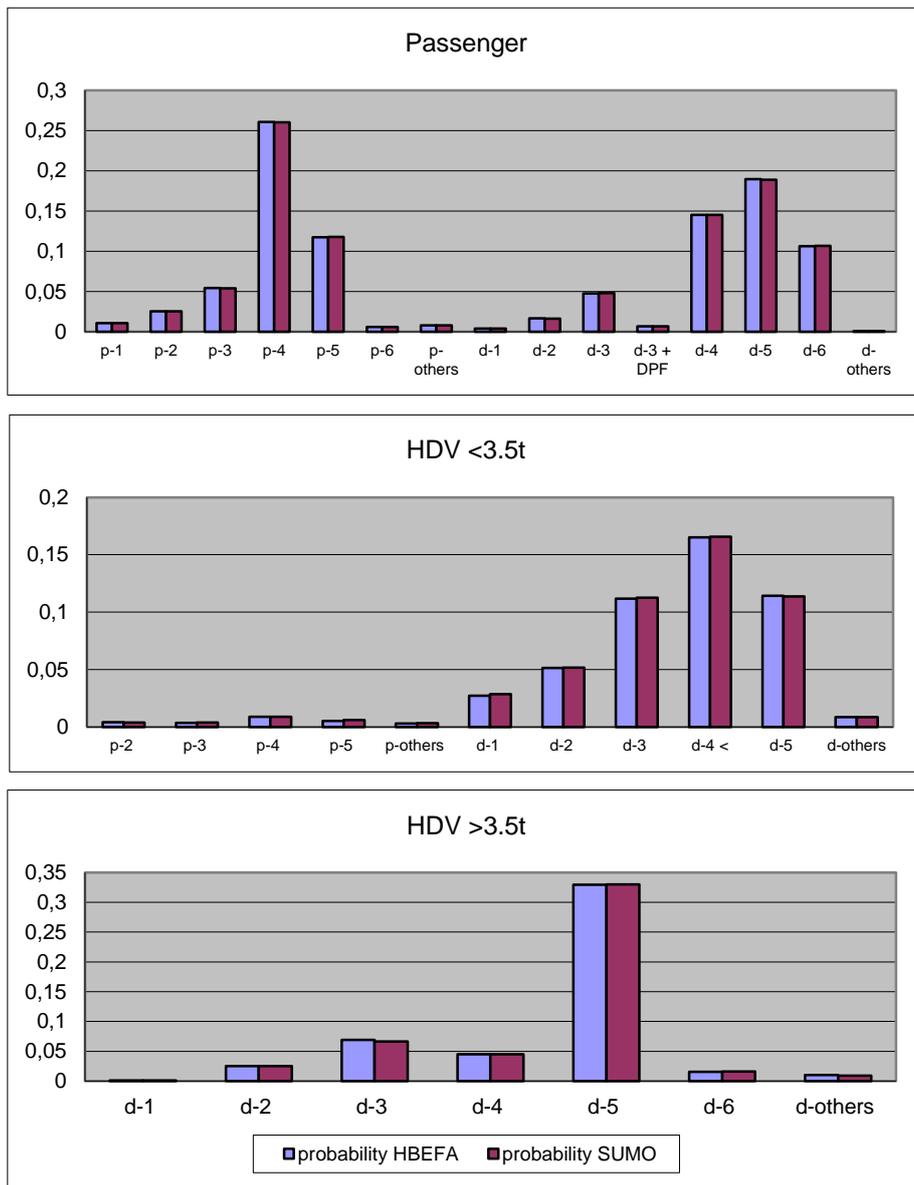


Figure 4.18: Vehicle classes, as found in HBEFA and as replicated within the scenario

For all four scenarios – the original scenarios and those that represent the traffic management actions – at first a traffic assignment was computed. Afterwards, a simulation run was performed, where different measurements were written.

It should be noted that the large number of measurements – about 650,000 vehicle trips, each generating emission and other values at 1 Hz – poses questions on their interpretation. SUMO is capable to aggregate measurements along the vehicles’ routes or along lanes or edges, but the results are nonetheless complex, covering different vehicle types, different area types that may be distinguished, different possible aggregation types, etc. At the current time, only the changes on roads were investigated, shown in Figure 4.19 and Figure 4.20 for the tempo restriction and the permissive environmental zone, respectively.



Figure 4.19: Differences in emissions for the tempo30 case against plain scenario (left: NO₂, right: PM₁₀; blue: less emissions, red: more emissions)



Figure 4.20: Differences in emissions for the permissive environmental zone against plain scenario (left: NO₂, right: PM₁₀; blue: less emissions, red: more emissions)

Albeit some questions – mainly on meaningful evaluation of the generated data – still persist, this usage example shows the possibilities a combination of a microscopic traffic simulation and a fine-grained emission model offers. We assume that a large-scale modelling of the impacts of traffic management actions on traffic’s environmental footprint could be a very valuable tool for fulfilling the air quality regulations posed by the European Union. Further extensions would be necessary for such an application. Currently, only the vehicular emissions are regarded, but immissions, not emissions are the measurement under regulation. The computation of immissions requires further models and tools, mainly for describing the pollutants’ dispersion and the existing background pollution.

4.5 Deployment

Licensing

PHEMlight’s functionality is divided into the model implementation (program) and the emission type specific data the implementation reads. The code, being included into SUMO is already “infected” by the GPL licence and will accordingly be released as a part of the usual SUMO release.

The data will be made only partially available for public use. Two vehicle types, a Diesel and a Gasoline passenger vehicle, both matching the Euro Norm 4, will be included into SUMO’s public

release. The remaining datasets will be distributed commercially by TU Graz. COLOMBO project partners have the access to all vehicle emission class definitions.

Availability

PHEMlight – the code and the two vehicle emission classes chosen to be made available for free – were included in SUMO’s release version on the 24th of February 2014. They will thereby be included within the next SUMO release, 0.20.0, currently scheduled to be put online at the begin of March 2014.

System Requirements

The PHEMlight methods are implemented within SUMO in a portable way and no additional libraries are needed. This means the requirements are the same as for standard SUMO.

User Documentation

SUMO’s user documentation on pollutant emissions had to be restructured, as it was very much built around the initially implemented HBEFA-based model. We decided to regroup the documentation around the topic “Models/Emissions”, for reducing the complexity of the overall SUMO documentation. As a result, the documentation on vehicular emissions – including both off-line coupling to PHEM as well as PHEMlight – is structured as shown in Figure 4.18. Links from the menu are not shown.

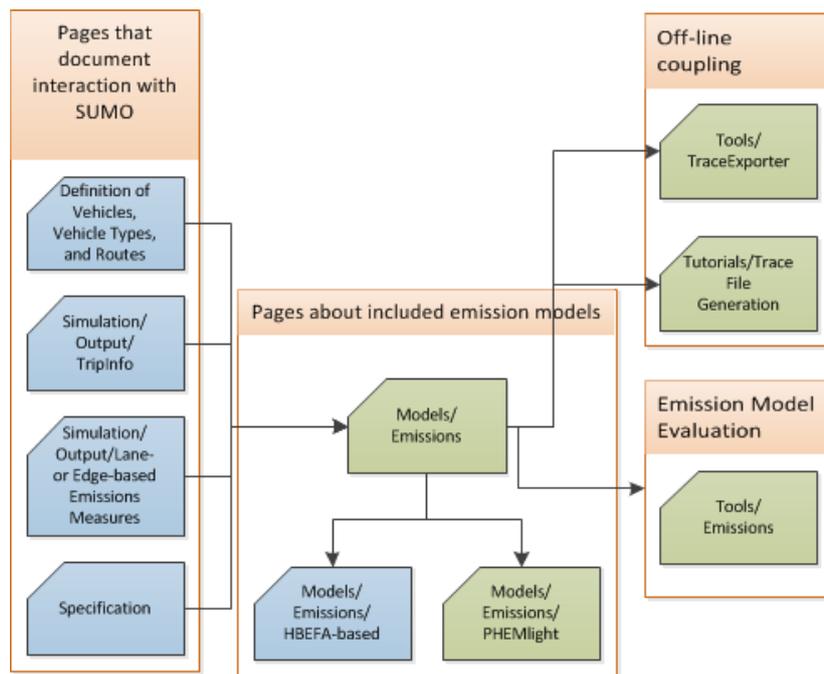


Figure 4.21: Documentation Structure (blue: old, updated pages; green: new pages)

The documentation focusses on using PHEMlight, not on its technical realization. The technical part may be included after it has been presented in scientific papers.

Tests

The functionality of PHEMlight is tested every night as a part of As PHEMlight is not included in SUMO’s public release completely, the test have been accordingly added to an internal test suite.

5 Summary

This deliverable summarizes the work performed in Tasks 4.1 and 4.2 of the “COLOMBO” project. Two topics were addressed within these work tasks: extending the PHEM model by vehicle emission classes that cover the vehicular population in the year 2020 and implementing connections between the microscopic traffic simulation SUMO and the pollutant emission model PHEM.

Within Task 4.1, PHEM was extended by emission factors of new vehicle types, including Hybrid Electrical Vehicles, Battery Electric Vehicles, and vehicles running on Compressed Natural Gas. Furthermore, methods that model engine start/stop technologies were implemented. The implemented models and data are based on measurements of existing vehicles’ behaviour.

Two connections between SUMO and PHEM were implemented and are described within this deliverable: a) an off-line connection where output files generated during a simulation run by SUMO are converted into inputs PHEM is capable to read and b) PHEMlight, a reduced version of PHEM that is embedded into SUMO.

The off-line connection was made available as a part of the open source version of SUMO and allows not only generating input files for PHEM, but also converts SUMO outputs to a large variety of trace files that can be read by other applications, mainly communication simulators. Being released about a year before finishing this report, it has already proven its functionality within the work of third-party SUMO users and first issues were already solved. The enormous file sizes and the time needed to generate and process them, but are not feasible when the simulation system shall be used in an optimization loop. Therefore, using PHEMlight should be preferred, even if the exactness of the obtained results proves to be worse.

Being directly embedded within SUMO, PHEMlight reduces the data handling and speeds up the computation of the simulated vehicles’ emissions. The performed evaluations show a good match against PHEM, but PHEMlight’s applicability to fine-grained vehicle behaviour has to be verified in next project steps by repeating comparisons against PHEM. PHEMlight will be commercial in nature, a sample data set containing emission definitions for two passenger vehicles will be provided with future versions of SUMO.

Appendix A – References

- [Bosch, 2012] Bosch. Profitieren Sie von umfassender Start / Stopp – Kompetenz: Teile, Diagnose und Services von Bosch. Brochure. http://aa-boschap-tr.resource.bosch.com/media/parts/service_parts__auto_parts/batteries/downloads_batteries/kompetenz_broschuere_de.pdf
- [COLOMBO D5.1, 2013] COLOMBO consortium. COLOMBO project’s Deliverable D5.1: “Prototype of overall System Architecture and Definition of Interfaces”, July 2013.
- [INFRAS, 2010] INFRAS. Handbuch der Emissionsfaktoren. 2010. Online: <http://www.hbefa.net/>, last visited on 21.10.2013
- [iTETRIS D3.1, 2009] iTETRIS Consortium iTETRIS Deliverable 3.1 – Traffic Modelling: Environmental Factors. 2009.
- [Flötteröd et al., 2012] Flötteröd, Y.-P., Wagner, P., Behrisch, M., Krajzewicz, D. Simulated-based Validity Analysis of Ecological User Equilibrium. In: Winter Simulation Conference Archive. 2012
- [Johnson, 2013] Johnson Controls Inc. Start-stop technology fact sheet. http://www.johnsoncontrols.com/content/us/en/products/power_solutions/power_solutions_presskit/fact-sheet/start-stop-technology.html
- [Kies et al., 2013] Kies A, Hausberger S., Silberholz G., et.al.: EHEV – Eco Drive for Hybrid Electric Vehicles; FFG project P-Nr. 829966; TU-Graz, October 2013
- [Knörr, Hausberger, 2011] Knörr W., Hausberger S.: Weiterentwicklung der Emissionsfaktoren für das Handbuch für Emissionsfaktoren (HBEFA); FKZ 370 952 142; Endbericht; Im Auftrag des Umweltbundesamtes; Heidelberg; 31. August 2011
- [Krajzewicz, Wagner, 2011] Krajzewicz, D., Wagner, P. Large-scale Vehicle Routing Scenarios based on Pollutant Emission. In: Advanced Microsystems for Automotive Applications 2011, pp. 237-246. Springer. AMAA 2011. ISBN 978-3-642-21380-9
- [Kühlwein et al., 2013] Kühlwein J, Rexeis M., Luz R., Hausberger S.: Update of Emission Factors for EURO 5 and EURO 6 Passenger Cars for the HBEFA Version 3.2. Report by order of the EUROPEAN COMMISSION DG JRC Contracts N°108919 and N°108921. Report No. I-25/2013/ Rex EM-I 2011/20 679 from 18.09.2013
- [Luz et al., 2011] Luz R., Hausberger S., Rexeis M., et al. Optimisation tool for Hybrid Electric Recuperation and Operation Strategies. Endbericht des Projektes „HERO“ finanziert im Rahmen der 2. Ausschreibung der Programmlinie a3plus des Forschungs- und Technologieprogramms iv2splus. Graz, 30.5.2011
- [Smode, 2013] Smode S. Simulation und Messung eines Hybrid PKW in Bezug auf ökonomischen Fahrstil, Diploma Thesis at the Institute for Internal Combustion Engines and Thermodynamics at the TU-Graz, October 2013
- [Vergés, 2013] Vergés, J. T. Analysis and simulation of traffic management actions for traffic emission reduction. Master thesis, TU Berlin, 2013.
- [Wegener et al., 2008] Wegener, A., Piorkowski, M., Raya, M., Hellbrueck, H., Fischer, S., Hubaux, J.P. TraCI: An Interface for Coupling Road Traffic and Network Simulators. In: Proceedings of the 11th Communications and Networking Simulation Symposium (CNS), 2008, pp. 155-163.

Appendix B – Documentation for the Off-line Interaction between SUMO and PHEM

The script is documented within the SUMO online documentation using two sections:

- a description of the script itself, listed below as Appendix B.1, originally located at <http://sumo-sim.org/userdoc/Tools/TraceExporter.html>
- a tutorial, listed below as Appendix B.2, originally located at http://sumo-sim.org/userdoc/Tutorials/Trace_File_Generation.html

Please note that the formatting is slightly different when viewed on-line and that the contents of the pages are frequently updated.

Appendix B.1 – TraceExporter Documentation

Tutorials/Trace File Generation

Contents

- [1 Introduction](#)
- [2 Simulation Scenario](#)
- [3 Generating a SUMO-trace](#)
- [4 Converting the Trace](#)
- [5 Further Possibilities](#)
 - [5.1 Output Options](#)
 - [5.2 Different Penetration Rates](#)
 - [5.3 Limiting the Time](#)
- [6 Further Information](#)

Introduction

This tutorial describes how to generate vehicular trace files. Besides SUMO, the [Tools/TraceExporter](#) will be used. It requires Python to be executed and currently allows to generate trace files for ns2/ns3, Shawn, OMNET, and PHEM.

Simulation Scenario

For generating vehicular traces, you need a simulation scenario, consisting at least of a road network and routes. You may visit [Tutorials/Hello Sumo](#) or [Tutorials/Quick Start](#) to learn how to get a scenario.

Generating a SUMO-trace

We assume You have the scenario and a configuration file named "myConfig.sumocfg". It may have a different name, of course. You should be able to start SUMO using Your configuration like this:

```
sumo -c myConfig.sumocfg
```

Now, we use the simulation to get an [fcd output](#), a trace file in a SUMO-format. This file will be later converted into a trace file for one of the applications supported by [Tools/TraceExporter](#). We generate an [fcd output](#) as following:

```
sumo -c myConfig.sumocfg --fcd-output sumoTrace.xml
```

And we will obtain the file "sumoTrace.xml".

Converting the Trace

We can now convert the vehicular traces in SUMO format into another format. For now, let's assume You would like to have a ns2 mobility file. We can generate one from the obtained [fcd output](#) using:

```
traceExporter.py --fcd-input sumoTrace.xml --ns2-mobilityoutput ns2mobility.tcl
```

We obtain the file "ns2mobility.tcl", which we can give ns2 as input.

Further Possibilities

Output Options

[Tools/TraceExporter](#) generates files for ns2/ns3, OMNET, Shawn, and PHEM. Please visit the [Tools/TraceExporter](#) manual page for further information about how to generate these files. Look for "output options".

Different Penetration Rates

[Tools/TraceExporter](#) allows to define different equipment / penetration rates using the `--penetration <FLOAT>`. `<FLOAT>` will be the amount of vehicles that are exported, where 1 is equal to 100 %. This means, if You want to have 10 % of vehicles to show up in Your ns2-trace, use:

```
traceExporter.py --penetration 0.1 \  
--fcd-input sumoTrace.xml --ns2-mobilityoutput ns2mobility.tcl
```

The `--seed <FLOAT>` allows You to give different random number initialisations for choosing the vehicles to export.

Limiting the Time

You may limit the time steps to export using the options `--begin <TIME>` and `--end <TIME>`. All time steps before `--begin` and all after `--end` will be dismissed.

Additionally, You may force [Tools/TraceExporter](#) to convert only every n^{th} time step using the option `--delta <TIME>`. Please note that the value should be a multiple of the time step length used within the simulation.

Further Information

- See the [Tools/TraceExporter](#) manual page
- See more information on SUMO's [fcd output](#)
- Visit tutorials describing how to set up a simulation: [Tutorials/Hello Sumo](#) or [Tutorials/Quick Start](#)

Tools/TraceExporter

Contents

- [1 From 30.000 feet](#)
- [2 Introduction](#)
- [3 Installation](#)
- [4 Usage Description](#)
 - [4.1 Input Options](#)
 - [4.2 Output Options](#)
 - [4.2.1 PHEM](#)
 - [4.2.2 OMNET](#)
 - [4.2.3 Shawn](#)
 - [4.2.4 ns2/ns3](#)
 - [4.2.5 GPSDAT](#)
 - [4.3 Processing Options](#)
 - [4.4 Processing Options](#)
- [5 See Also](#)

From 30.000 feet

traceExporter.py converts SUMO [fcd output](#) to different "trace file" formats: OMNET, Shawn, ns2/ns3, PHEM.

Purpose: trace file conversion/generation

System: portable (Linux/Windows is tested); runs on command line

Input (mandatory): SUMO fcd-output

Output: One or more "trace file(s)" and other complementary files

Programming Language: Python

Introduction

Different applications read "vehicular traces", files containing mainly positions of vehicles over time. **traceExporter.py** generates such files by converting SUMO's [fcd output](#) into several supported formats. Following files can be generated, sorted by application:

- OMNET: mobility-traces (.xml)
- Shawn: snapshot-files (.xml)
- ns2/ns3: trace-files, activity-files, and mobility-files (all *.tcl)
- PHEM: .dri-files, .str-files, .fzp-files, .flt-files
- GPSDAT (unknown)
- GPX
- POIs (can visualize GPS distortions and sampling in SUMO-GUI)

Installation

Tools/TraceExporter comes with SUMO and is located in `<SUMO_HOME>/tools/bin/`. Python must be installed.

Usage Description

You convert a given [fcd output](#) into a format using a command like this:

```
traceExporter.py --fcd-input myFCDoutput.xml --omnet-output myOMNETfile.xml
```

In the case given above, a file for [OMNET](#) will be built.

In the following, the command line options are described.

Input Options

traceExporter.py often needs only the [fcd output](#) as input. In some cases, the network used while obtaining the [fcd output](#) must be given. [traceExporter.py] gives an error message if the wanted output needs additional files, such as the network, but no such file is specified. The options for defining the input files are:

Option	Description
<code>--fcd-input <FILE></code> <code>-i <FILE></code>	The fcd-output - file to convert
<code>--net-input <FILE></code> <code>-n <FILE></code>	The network file used; Mainly used for exporting the network

Output Options

In the following, the output options of **traceExporter.py** are given, divided by the application,

PHEM

Option	Description
<code>--dri-output <FILE></code>	Generates a PHEM .dri-file named <FILE>
<code>--str-output <FILE></code>	Generates a PHEM .str-file named <FILE> ; note that the SUMO network used during the simulation must be given using the <code>--net-input</code> option
<code>--fzp-output <FILE></code>	Generates a PHEM .fzp-file named <FILE>
<code>--flt-output <FILE></code>	Generates a PHEM .flt-file named <FILE>

OMNET

Option	Description
--------	-------------

<code>--omnet-output <FILE></code>	Generates a OMNET mobility trace named <FILE>
--	---

Shawn

Option	Description
<code>--shawn-output <FILE></code>	Generates a Shawn snapshot file named <FILE>

ns2/ns3

Option	Description
<code>--ns2config-output <FILE></code>	Generates a ns2 configuration file named <FILE>
<code>--ns2activity-output <FILE></code>	Generates a ns2 activity file named <FILE>
<code>--ns2mobility-output <FILE></code>	Generates a ns2 mobility file named <FILE>

GPSDAT

Option	Description
<code>--gpsdat-output <FILE></code>	Generates a GPSDAT file named <FILE>

GPX

Option	Description
<code>--gpx-output <FILE></code>	Generates a GPX file named <FILE>

PoI

Option	Description
<code>--poi-output <FILE></code>	Generates a SUMO PoI file named <FILE>

Processing Options

Several options allow to fine-tune the processing.

Option	Description
<code>--penetration <FLOAT></code>	A float between 0 and 1, defining the probability of a vehicle to be included in the output; <i>default: 1</i>
<code>--gps-blur <FLOAT></code>	A float defining a random offset applied on the vehicle positions before exporting them; <i>default: 0</i>

--delta-t <u><TIME></u> -d <u><TIME></u>	Defines the time sampling; if given, only every <u><TIME></u> time step will be exported.
--begin <u><TIME></u> -b <u><TIME></u>	Time steps before <u><TIME></u> are not exported.
--end <u><TIME></u> -e <u><TIME></u>	Time steps after <u><TIME></u> are not exported. Please note that some issues are known, here.

Processing Options

Option	Default Value	Description
--help -h		Shows a help screen.

See Also

- This [tutorial on using TraceExporter](#)
- [working around bugs in version 0.19.0](#)

This part of SUMO was developed, reworked, or extended within the project "[COLOMBO](#)", co-funded by the European Commission



Appendix C – PHEMlight User Guide

As described in section 4.5, the current documentation covers information about how to use PHEMlight, not its technical details. Two pages were set up:

- one generic page about emission models in SUMO, listed below as Appendix C.1, originally located at <http://sumo-sim.org/wiki/Models/Emissions> (wiki)
- one page about PHEMlight, listed below as Appendix C.2, originally located at <http://sumo-sim.org/wiki/Models/Emissions/PHEMlight> (wiki)

Both pages can be found within SUMO's documentation wiki only. They will become a part of the official SUMO release within the next release, scheduled for the end of November 2013. Please note that the formatting is slightly different when viewed on-line.

Appendix C.1 – Page on Included/Supported Emission Models

Models/Emissions

SUMO includes the following emission models:

- [HBEFA v2.1-based](#): A continuous reformulation of the [HBEFA](#) v2.1 emissions data base (open source);
- [PHEMlight](#), a derivation of the original [PHEM](#) emission model (closed source, commercial).

Table: pollutants covered by models

model	pollutant / measurement					
	CO ₂	CO	HC	NO _x	PM _x	fuel consumption
HBEFA v2.1-based	x	x	x	x	x	x
PHEMlight	x	x	x	x	x	x

Pollutants emitted by the simulated vehicles can be visualised using [SUMO-GUI](#) or be written into output files, both by [SUMO](#) and [SUMO-GUI](#). The following output can be used:

- [trip information](#): In combination with the emissions device, the tripinfo output contains the sum of all pollutants emitted / fuel consumed during a vehicle's journey (see [trip information](#))
- [edgelane emissions](#): These output files contain the pollutants emitted at an edge / a lane, aggregated over a variable time span (see [edgelane emissions](#))

Both models implement different vehicle emission classes. These classes can be assigned to vehicles by using the vehicle type attribute "emissionClass". Available emission classes can be found within the emission model descriptions ([HBEFA v2.1-based](#), [PHEMlight](#)).

Further Interfaces

The tool [traceExporter.py](#) converts SUMO's [fcd-output](#) into files that can be directly read by the [PHEM](#) application. A [tutorial on generating trace files](#) using this tool is available.

Appendix C.2 – Page on PHEMlight

Models/Emissions/PHEMlight

[PHEM \(Passenger Car and Heavy Duty Emission Model\)](#) is an instantaneous vehicle emission model developed by the TU Graz since 1999. [PHEM](#) is based on an extensive European set of vehicle measurements and covers passenger cars, light duty vehicles and heavy duty vehicles from city buses up to 40 ton semi-trailers.

Within [COLOMBO](#), a simplified version of PHEM, embedded into SUMO, was developed. The PHEMlight model is the result of this work. An in-depth description can be found in [COLOMBO's deliverable D4.2](#).

PHEMlight uses data files that include the parameters of the modelled emission classes. At the time being, data sets for two emission classes are included in SUMO: PKW_D_EU4 and PKW_G_EU4, resembling passenger vehicles, Euro Norm 4, powered by Diesel (PKW_D_EU4) and gasoline (PKW_G_EU4).

To use PHEMlight, you have to set the path to these input files using the configuration option `--phemlight-path <FILE>` where `<FILE>` is the path. Alternatively, you may set the environment variable `PHEMLIGHT_PATH`.

In addition, your vehicles must be assigned to one of PHEMlight's vehicle emission classes, summarised in the following table.

PHEMlight Emission Classes for Light Vehicles (1)

Passenger Cars	Light Duty Vehicles		
PKW_D_EU0	LNF_D_EU0_I	LNF_D_EU0_II	LNF_D_EU0_III
PKW_D_EU1	LNF_D_EU1_I	LNF_D_EU1_II	LNF_D_EU1_III
PKW_D_EU2	LNF_D_EU2_I	LNF_D_EU2_II	LNF_D_EU2_III
PKW_D_EU3	LNF_D_EU3_I	LNF_D_EU3_II	LNF_D_EU3_III
PKW_D_EU4	LNF_D_EU4_I	LNF_D_EU4_II	LNF_D_EU4_III
PKW_D_EU5	LNF_D_EU5_I	LNF_D_EU5_II	LNF_D_EU5_III
PKW_D_EU6	LNF_D_EU6_I	LNF_D_EU6_II	LNF_D_EU6_III
PKW_G_EU0	LNF_G_EU0_I	LNF_G_EU0_II	LNF_G_EU0_III
PKW_G_EU1	LNF_G_EU1_I	LNF_G_EU1_II	LNF_G_EU1_III
PKW_G_EU2	LNF_G_EU2_I	LNF_G_EU2_II	LNF_G_EU2_III
PKW_G_EU3	LNF_G_EU3_I	LNF_G_EU3_II	LNF_G_EU3_III
PKW_G_EU4	LNF_G_EU4_I	LNF_G_EU4_II	LNF_G_EU4_III
PKW_G_EU5	LNF_G_EU5_I	LNF_G_EU5_II	LNF_G_EU5_III
PKW_G_EU6	LNF_G_EU6_I	LNF_G_EU6_II	LNF_G_EU6_III

PHEMlight Emission Classes for Light Vehicles (2)

Two-Wheelers	Hybrid Passenger Vehicles	Motor Cycles	
KKR_G_EU0	H_PKW_D_EU5	MR_G_EU0_2T	MR_G_EU0_4T
KKR_G_EU1	H_PKW_D_EU6	MR_G_EU1_2T	MR_G_EU1_4T
KKR_G_EU2	H_PKW_G_EU5	MR_G_EU2_2T	MR_G_EU2_4T
KKR_G_EU3	H_PKW_G_EU5	MR_G_EU3_2T	MR_G_EU3_4T
KKR_G_EU4		MR_G_EU4_2T	MR_G_EU4_4T
		MR_G_EU5_2T	MR_G_EU5_4T

PHEMlight Emission Classes for Heavy Vehicles

Tractor/Trailor	Coach	Urban and Inter Urban Buses	Trucks	
LSZ_D_EU0	RB_D_EU0	LB_D_EU0	SOLO_LKW_D_EU0_I	SOLO_LKW_D_EU0_II
LSZ_D_EU1	RB_D_EU1	LB_D_EU1	SOLO_LKW_D_EU1_I	SOLO_LKW_D_EU1_II
LSZ_D_EU2	RB_D_EU2	LB_D_EU2	SOLO_LKW_D_EU2_I	SOLO_LKW_D_EU2_II
LSZ_D_EU3	RB_D_EU3	LB_D_EU3	SOLO_LKW_D_EU3_I	SOLO_LKW_D_EU3_II
LSZ_D_EU4	RB_D_EU4	LB_D_EU4	SOLO_LKW_D_EU4_I	SOLO_LKW_D_EU4_II
LSZ_D_EU5	RB_D_EU5	LB_D_EU5	SOLO_LKW_D_EU5_I	SOLO_LKW_D_EU5_II
LSZ_D_EU6	RB_D_EU6	LB_D_EU6	SOLO_LKW_D_EU6_I	SOLO_LKW_D_EU6_II

Further Ressources

- [COLOMBO: Deliverable 4.2; Extended Simulation Tool PHEM coupled to SUMO with User Guide \(Draft\)](#); February, 2014

This part of SUMO was developed, reworked, or extended within the project "[COLOMBO](#)", co-funded by the European Commission



Appendix D – Tools for Emissions Evaluations

Tools/Emissions

Jump to: [navigation](#), [search](#)

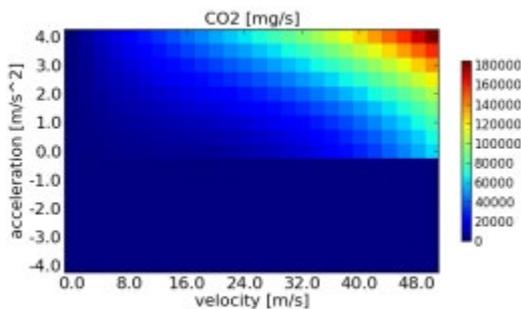
SUMO includes some small-sized tools and tests that help to develop and evaluate the implemented [emission models](#). These tools are presented in the following.

Contents

- [1 emissionsMap](#)
 - [1.1 Options](#)
 - [1.1.1 Processing](#)
 - [1.1.2 Output](#)
 - [1.1.3 Emissions](#)
 - [1.1.4 Report](#)
- [2 emissionsDrivingCycle](#)
 - [2.1 Options](#)
 - [2.1.1 Input](#)
 - [2.1.2 Processing](#)
 - [2.1.3 Output](#)
 - [2.1.4 Emissions](#)
 - [2.1.5 Report](#)

emissionsMap

This tool generates matrices of emissions for a given range of velocities, accelerations, and slopes, given the vehicle's emission class mainly.



The tool obtains parameter that describe the ranges and the emission classes to generate. If the option `--iterate <BOOL>` is set, emissionsMap iterates over all available emission classes. In this case, the option `--output-file <PATH>` should denote a folder the emission maps shall be written into. If `--iterate <BOOL>` is not set, `--output-file <FILE>` should name the file to generate and the emission class to write the map for must be defined using the option `--emission-class <STRING>`. If PHEMlight shall be used, one has to define the path to the CEP-files using the option `--phemlight-path <PATH>`.

The tool is located in `<SUMO_HOME>/bin`.

The image shows the visualisation of CO₂ emission of the [HBEFA v2.1-based](#) "P_7_6" emission class.

Options

Processing

Option	Description
-i <u><BOOL></u> --iterate <u><BOOL></u>	If set, maps for all available emissions are written
-e <u><STRING></u> --emission-class <u><STRING></u>	Defines the name of the emission class to generate the map for
--v-min <u><FLOAT></u>	Defines the minimum velocity boundary of the map to generate (in [m/s]); <i>default: 0</i>
--v-max <u><FLOAT></u>	Defines the maximum velocity boundary of the map to generate (in [m/s]); <i>default: 50</i>
--v-step <u><FLOAT></u>	Defines the velocity step size (in [m/s]); <i>default: 2</i>
--a-min <u><FLOAT></u>	Defines the minimum acceleration boundary of the map to generate (in [m/s ²]); <i>default: -4</i>
--a-max <u><FLOAT></u>	Defines the maximum acceleration boundary of the map to generate (in [m/s ²]); <i>default: 4</i>
--a-step <u><FLOAT></u>	Defines the acceleration step size (in [m/s ²]); <i>default: .5</i>
--s-min <u><FLOAT></u>	Defines the minimum slope boundary of the map to generate (in [°]); <i>default: -10</i>
--s-max <u><FLOAT></u>	Defines the maximum slope boundary of the map to generate (in [°]); <i>default: 10</i>
--s-step <u><FLOAT></u>	Defines the slope step size (in [°]); <i>default: 1</i>

Output

Option	Description
-o <u><FILE></u> --output-file <u><FILE></u>	Defines the file (or the path if --iterate was set) to write the map(s) into

Emissions

Option	Description
-p <u><PATH></u> --phemlight-path <u><PATH></u>	Determines where to load PHEMlight definitions from.

Report

Option	Description
-v <u><BOOL></u> --verbose <u><BOOL></u>	Switches to verbose output; <i>default: false</i>

emissionsDrivingCycle

This tool computes emissions given a timeline of speeds/accelerations for a single vehicle. The driving cycle definition must be given using the option **--timeline-file** [<FILE>](#). It must be a ';' separated .csv-file with the following columns:

- time (in [s]); must start at zero and be increasing in steps of 1 s
- velocity (in [m/s] unless **--kmh** is set, see below)
- acceleration (in [m/s²]); this column may be omitted using the option **--compute-a**
- optional: slope (in [°]); if given, one has to enable it using the option **--have-slope**

The tool generates a ';' separated .csv file which contains the following columns:

- time (in [s])
- velocity (in [m/s])
- acceleration (in [m/s²])
- slope (in [°])
- CO emissions (in [mg/s])
- CO2 emissions (in [mg/s])
- HC emissions (in [mg/s])
- PMx emissions (in [mg/s])
- NOx emissions (in [mg/s])
- fuel consumption (in [ml/s])

Additionally, the sums of the generated emissions / the consumed fuel are printed on the screen.

The tool is located in `<SUMO_HOME>/bin`.

Options

Input

Option	Description
-t <FILE> --timeline-file <FILE>	Defines the file to read the driving cycle from.
-e <STRING> --emission-class <STRING>	Defines for which emission class the emissions shall be generated.

Processing

Option	Description
-a <BOOL> --compute-a <BOOL>	If set, the acceleration is computed instead of being read from the file.
-s <BOOL> --skip-first <BOOL>	If set, the first line of the read file is skipped.

--kmh <u><BOOL></u>	If set, the given speed is interpreted as being given in km/h.
--have-slope <u><BOOL></u>	If set, the fourth column is read and used as slope (in [°]).
--slope <u><FLOAT></u>	Sets a global slope (in [°]) that is used if the file does not contain slope information.

Output

Option	Description
-o <u><FILE></u> --output-file <u><FILE></u>	Defines the file to write the emission cycle results into.

Emissions

Option	Description
-p <u><PATH></u> --phemlight-path <u><PATH></u>	Determines where to load PHEMlight definitions from.

Report

Option	Description
-v <u><BOOL></u> --verbose <u><BOOL></u>	Switches to verbose output; <i>default: false</i>

This part of SUMO was developed, reworked, or extended within the project "[COLOMBO](#)", co-funded by the European Commission

