Conference on Future Automotive Technology
Focus Electromobility
München, March, 18-19st 2013

# A modular and dynamic approach to predict the energy consumption of electric vehicles

Stefan Grubwinkler[1], Prof. Dr-Ing. Markus Lienkamp[1]
[1]*Institute of Automotive Technology, TU München, Garching, Germany*
*E-mail*: grubwinkler@ftm.mw.tum.de

**ABSTRACT:** An energy prediction along a specified route is necessary for various applications of electric vehicles (EVs). The energy consumption of EVs depends on several different impact factors. Because of their different dynamic behaviour and their interdependency, a server-based energy prediction system is used consisting of an in-vehicle part and a part realized on a backend server. This system architecture requires a modular and dynamic model to predict the energy consumption of EVs, which is introduced in this paper. Three modules are used to estimate the required propulsion energy. The average propulsion energy is predicted in a first step by an in-vehicle system that takes vehicle-specific influencing factors into account. This is then adapted in a second step to map attributes that occur along the chosen route. In a third module, the expected deviation from the predicted mean value is estimated using speed profiles that have been collected from a huge number of vehicles. The proposed system architecture is discussed with its requirements. Finally, first results of the prediction of the mean propulsion energy are presented.

***Keywords***: energy prediction system, electric vehicle, server-based system, propulsion power estimation

## 1. INTRODUCTION

Despite efforts of the vehicle industry to develop electric vehicles (EVs) their market penetration rate is still very low. One of the main reasons is the discrepancy between the range, charging time and prices of EVs offered on the market and the consumer expectations [1]. Due to the limited capacity and long recharging times of the currently available energy-storage technology, range anxiety is a limiting factor for market penetration. This is why a reliable and accurate EV range estimation is indispensable to increase consumer acceptance of EVs.

An accurate prediction of the energy consumption for a specified route is the basis for different important EV applications: The residual range can be calculated by considering the predicted future energy consumption of EVs and the estimated energy available in the battery. EV energy prediction is also used for vehicle navigation functions such as eco-routing [2] or for operational strategies in order to reduce the energy consumption needed to reach a destination [3]. Apart from in-vehicle applications, the energy prediction of EVs is required for charging strategies or travel planning applications [4].

Energy prediction is a complex problem since the energy consumption of vehicles depends on various factors [5]: Vehicle-specific parameters like mass, aerodynamic, rolling resistance and drivetrain efficiency affect the energy consumption. The operational strategy (e.g. recuperation during braking) or the energy needed for heating or air-conditioning due to different weather conditions also have an important influence on the energy consumption of electric vehicles. Similarly, different attributes of the chosen route (e.g. type of roads or topography) and attributes related to the prevailing traffic conditions (e.g. traffic congestion, traffic signals) also influence the energy consumption. Furthermore, the driver's individual driving style based on their skills and attitude can significantly influence the required energy consumption. Of course, these numerous impact factors interact strongly with each other. Besides, their temporal variability differs and various data sources are necessary in order to consider all relevant impact factors.

Cars will soon be able to exchange data about their position or speed via vehicle communication systems which can then be used by other vehicles. This prospective but realistic scenario may help improve energy predictions for EVs due to the described complexity of impact factors. As some of the varying influencing factors cannot be detected with in-vehicle sensors, up-to-date information from different data sources is needed. This is why we assume that parts of the energy prediction can be realized in a backend system connected to different vehicles.

This paper introduces an approach for a system to predict the necessary energy to reach a specified destination. The system is suitable for a server-based approach. The focus in this paper is on the general description of the energy prediction system and on the simulation test bed to develop the prediction system.

The motivation for a server-based approach is described in section 3. The comparability of energy consumption of different vehicles is a problem for these kinds of approaches. Collected speed profiles should be used for the proposed prediction system for the exchange of information among different vehicles. The speed profiles contain information about the individual driving style and the prevailing traffic situations. The consideration of the individual driving style and other effects on energy consumption will be developed in future and is not part of this paper. But these relevant impact factors on energy consumption of EVs are considered in the development of the system architecture of the whole prediction system.

## 2. RELATED WORK

This section introduces relevant existing systems for energy prediction of EVs.

In [6], a machine-learning engine (MLE) is used to predict the energy costs and travel time for road segments based on vehicle-specific data and context data such as time attributes or weather conditions. Additional machine-learning engines are needed to convert actual vehicle-specific energy costs per road segment into standardised values and vice versa. Different MLEs are necessary to compare data from different vehicles. The predicted energy consumption is used for energy efficient routing algorithms that are also based on machine-learning techniques.

Machine-learning algorithms, as used for e.g. in [6], are applicable for crowd-sourcing server-based systems like in the introduced project. Since collected speed profiles are the basis for our proposed prediction system different algorithms are necessary.

Other methods for energy prediction do not simply use the vehicles' energy costs but rather speed profiles to calculate the necessary energy for a specified trip. One possibility is the prediction of possible future speed profiles by using known geographical data along a specified route. In [7], a target speed profile is used to predict the energy consumption for every section along a trip. Geographical map data such as speed limits, slopes, traffic lights, road signs or traffic patterns are used to calculate the future speed profile. The future speed profiles are based only on map attributes, individual driving styles are not taken into account.

Features of collected speed profiles are used for the energy prediction based on statistical methods. Thus a prediction of speed profiles on the basis on map attributes is not intended in our project.

Another solution is the use of statistical methods to predict the energy consumption based on speed profiles. Paper [8] presents a method to quickly calculate the energy consumption for each link in a map for routing algorithms. The actual power of conventional vehicles depends on road gradient resistance, rolling resistance, air resistance, acceleration resistance, demanded auxiliary power and a factor for the engine-dependent transmission efficiency. An estimation function for the average acceleration and deceleration depending on traffic congestion, road type and gradient for every link is generated to calculate the necessary driving forces and engine power. Depending on traffic congestion, road type and gradient for each link; several estimation functions calculate the relative time share of different driving phases (cruising, idling, acceleration and deceleration phase). The estimation functions are calculated using a database of driving cycles. The influence of the different driving resistance forces varies depending on the driving phases (e.g. the rolling resistance has no influence when idling), so that they are weighted depending on the calculated share of time spent in each driving phase.

The aim of this solution is the fast calculation of energy consumption. Since in the proposed project a more detailed prediction is necessary a more detailed approach is intended. Nevertheless the described statistical methods can be used.

An eco-routing navigation system is presented in [9]. An emission model that predicts the necessary energy on the selected links in the roadway network is needed to calculate an eco-friendly route. Various parameters are thus used such as vehicle characteristics and roadway characteristics from a digital map. Different sources (vehicle sensors, point measurements, car floating data or traffic simulation models) provide real-time and historic traffic parameters. Furthermore, other parameters such as driver characteristics and environmental factors are also used. Multivariate regression techniques are employed to develop a prediction model. The described parameters, a vehicle-specific validated microscopic fuel consumption model and real-world vehicle velocity trajectories are used in the regression model. The routing engine uses the energy costs for every link estimated by using the vehicle-specific calibrated regression coefficients of the prediction model.

This approach can partly be used, since the model has to be recalibrated for different types of vehicles. Besides, the individual driving style is not considered. The described statistical methods can be used for our project.

Another statistical method is employed in [10]. A driving pattern recognition strategy uses different features from traffic and road conditions. A hilly-zone-clustering algorithm based on road gradient information along with a speed-zone-clustering method that employs speed limits and is based on future speed predictions are used to partition a complete route. As a result, every segment of the route has a different combination of features. A library of normalized driving force distributions is saved for every possible driving pattern. The corresponding force distribution is used for the energy prediction according to the sequence of feature zones with their identified driving patterns along a trip.

The method described in [10] does not use crowd-sourced collected information extracted from speed profiles, which we want to use in our project. Nevertheless the described driving patterns or features can be evaluated for possible usage.

## 3. SERVER-BASED SYSTEM FOR ENERGY PREDICTION

Studies show that there will be a great change in information and communications technology (ICT) in the automotive industry. A cloud-based ICT-architecture will replace likely today’s in-vehicle centralized ICT-architecture [11]. On the other hand, the major challenges of e-mobility are intelligent knowledge distribution, predicting EV travel time, and energy and journey planning [12]. Cloud-based ICT-architectures might help to solve these.

We are currently experiencing a rapid increase in the use of different crowdsourcing applications such as *Waze, OpenStreetMap* or *INRIX Traffics* [13], since users benefit from the huge amount and actuality of data collected by the community. Crowdsourcing applications will enter the automotive sector since the data can be collected easily by smart phone sensors and the number of vehicles with an UMTS or LTE connection will increase.

These trends enable us to develop a server-based system for EV energy prediction (Fig.1): The energy prediction system consists of an in-vehicle system and a backend part which is connected to the vehicle via a wireless communication link. Various up-to-date databases can be accessed in the backend and used for the energy prediction. All vehicles (even those not using the energy prediction system) collect the required information as rolling sensors and this is then stored in the backend on a server.

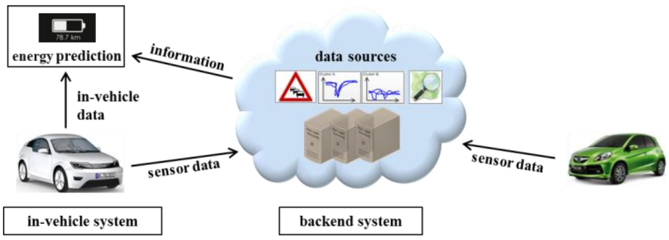


Figure 1: Server-based system for energy prediction

### 3.1 System requirements for server-based energy prediction

A server-based system has different and additional requirements for energy prediction models. Data collected for energy-related factors has to be used for all vehicles despite their different attributes. And it has to be remembered that the crowd sourced data is collected on different routes depending on the drivers’ choice. This is not made easier by the fact that the collected data varies greatly depending on the different personal driving styles and further conditions (e.g. weather or traffic flow) (see figure 2).

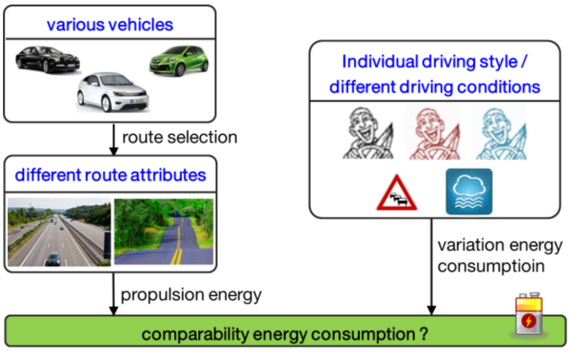

various vehicles
Individual driving style / different driving conditions
route selection
different route attributes
variation energy consumptioin
propulsion energy
comparability energy consumption ?

Figure 2: Problem for a server-based system for energy prediction

For this reason, the impact of driver, vehicle and route topography have to be considered separately, or at least appropriate interfaces have to be defined so that the database with collected energy relevant signals can be used for all types of vehicles. Separate systems would not be possible because the quantity of up-to-date data has to reach a critical size.

Following system requirements are identified:

- **Combination of an in-vehicle and server-based system**
  EVs’ energy consumption depends on different impact factors. By using a server-based energy prediction system it is possible to enhance in-vehicle data with different up-to-date online resources. Since the remaining range is relevant information when driving an EV, a stand-alone backend system is not possible if a connection to the backend is not available. In this case, an in-vehicle system has to predict the energy consumption. Besides, it is not practical to send all vehicle-specific data to the backend, so that a combination of an in-vehicle and server-based prediction system is necessary.
- **Modularity**
  Different usable data sources (e.g. attributes of digital maps, traffic information) are not always available for every road segment. Thus, a modular system architecture is required to realize an energy prediction with limited available data. Furthermore, modularity enables the energy prediction system to be used for various applications.
- **Independence of crowdsourcing-based and user- or vehicle-specific influencing factors**
  In community-based systems, a group of users exchange data. If the shared data is collected under equal conditions the data can

be exchanged easily. The exchange of energy consumption figures for EVs is more difficult since the data is very dependent on individual factors such as vehicle parameters or individual driving style. Besides, impact factors that are the same for all users exist to a limited extent (e.g. traffic conditions, weather), though they can vary a lot over time. Thus, crowdsourcing data can only be used for EVs' energy prediction if the collected and shared data is separated from individual influences. This calls for the definition of appropriate interfaces or the use of normalization processes.

- **Dynamic adaption**
  User-specific, in-vehicle parameters are not known in advance and can vary greatly during a trip. Some EVs have different operational modes (e.g. eco-mode, sport-mode) or different settings for the deceleration mode. Studies have shown, for example, that the deceleration mode in particular changes several times during a trip [14]. Because of the high number of impact factors and the changeable behaviour of some factors, no accurate prediction for an entire route is possible before starting a trip. Instead, a dynamic adaption to the current situation is necessary. As the prediction system consists of several modules, a dynamic adaption of every module is necessary.
- **Link-based approach**
  Since the system could be used for eco-routing and other navigational applications, the energy has to be predicted for every single link along a selected route. These link values can be used for known navigation algorithms as link weights to find an optimal route. A quick update of several link weights has to be possible during navigation, which has to be considered when the necessary algorithms for energy prediction are selected.

Furthermore, additional requirements such as calculation time are important for real-time applications. Since we are discussing an approach there will be no evaluation of the performance of the system, though a possible realization in an EV is taken into account in the system design process.

Database requirements such as privacy, security, redundancy or largest number of users have to be taken into account for server-based systems, but they are not discussed in this research.

## 3.2 System design

### 3.2.1 Energy consumption of EVs

The energy consumption of vehicles depends primarily on the energy needed to overcome the forces opposing the vehicle's motion. These include the power of the rolling friction $P_{roll}$, the aerodynamic drag force $P_{air}$, the inertia at slopes $P_{slope}$ and the acceleration force $P_{acc}$. $P_{wl}$ is the driveshaft output power on wheels depending on vehicle parameters.

$$P_{wl} = P_{air} + P_{acc} + P_{slope} + P_{roll} \tag{1}$$

The power needed to overcome the driving forces opposing the vehicle's motion is supplied by the battery. Losses occur in the driveline components (inverter, electric machine and transmission box). The transmission efficiency factor $\eta_{pos}$ represents all losses (2).

The electric machine is used in EVs for regenerative braking. Part of the driveshaft output power $P_{wl}$ is recuperated and helps charge the battery. The ratio between regenerative braking and mechanical braking depends on the operational strategy of the EV and the driver's requested deceleration. This ratio and the transmission losses are represented by the efficiency factor $\eta_{neg}$. The values for $\eta_{pos}$ and $\eta_{neg}$ depend a lot on the corresponding operating point of the electric machine.

$$P_{pt} = \begin{cases} \eta_{pos} * P_{wl} & \text{if } P_{wl} \geq 0 \\ \eta_{neg} * P_{wl} & \text{else} \end{cases} \tag{2}$$

The propulsion power $P_{pt}$ is mainly independent of the travel time $t_{trip}$ in EVs. What's more, auxiliary power $P_{aux}$ is needed for the on-board power supply and for heating and air conditioning. Their energy consumption depends only on travel time $t_{trip}$. $P_{aux}$ is more or less independent of the spatial domain *s*. Losses between the battery and the consumers are taken into account. The necessary total energy demand for an EV is supplied from the traction battery and can be calculated:

$$E_{ges} = \int_0^{t_{trip}} P_{Bat}\, dt = \int_0^{t_{trip}} P_{aux}\, dt + \int_0^{s} P_{pt}\, ds \tag{3}$$

### 3.2.2 Prediction of auxiliary energy

Compared to conventional vehicles, the power for heating or air conditioning has a higher ratio of the total energy consumption. As $P_{aux}$ is independent of the route and selected speed, the prediction for the necessary energy $\hat{E}_{aux}$ can be calculated independent of the necessary propulsion energy $\hat{E}_{pt}$. $\hat{E}_{aux}$ depends on vehicle parameters, user settings (e.g. desired interior temperature) and prevailing weather conditions, which can be measured by common in-vehicle sensors. The vehicle's navigation system provides the required trip time $t_{trip}$. Since $\hat{E}_{aux}$ depends mainly on vehicle und user-specific parameters, the potential for improving the prediction by using a server-based approach is not very high. Crowd based information only allows a more accurate estimation of $t_{trip}$. For this reason, we suggest an in-vehicle prediction system for $\hat{E}_{aux}$, which is updated dynamically according to user-settings and the progression of the temperature inside the vehicle. A dynamic update of the travel time estimation as in today's navigation systems is necessary.

### 3.2.3 Prediction of propulsion energy

Unlike $\hat{E}_{aux}$, the prediction of the necessary propulsion energy can be improved by using a server-based approach since the energy consumption of the EV is largely determined by driving patterns represented by variations of vehicle speed and inertial acceleration [10].

The first question to be discussed is the choice of variables to be exchanged between several users via the backend system. The exchange of energy values requires a normalisation process as in [6]. If the normalisation system is realized in the backend a continuous transmission of the numerous relevant in-vehicle signals is necessary. Another possibility is to exchange speed profiles or extracted features from speed profiles via the backend system. In this paper it was decided to use speed profiles to exchange information between different EVs because they offer several advantages:

- Speed profiles contain information about driving style, traffic conditions and road attributes and take into account most of the impact factors on propulsion energy consumption.
- A number of applications and technologies exist for collecting speed profiles such as the collection of floating car data (FCD) for intelligent transportation systems.
- Data from conventional vehicles can be used, which is very important since EVs do not yet have a high penetration.
- Information can be merged more easily with existing data bases (e.g. traffic information) since these are often based on speed profiles too.

By using speed profiles to exchange information via the backend, we can implement intelligent prediction models that do not depend on vehicle attributes. In comparison the algorithms introduced in [8] or [9] have to be adapted or relearned for different vehicles.

As discussed in section 3.1. the crowd-based influence has to be separated from the vehicle-specific one in the prediction of the propulsion energy $\hat{E}_{pt}$. The average energy consumption of the EV can be predicted by using vehicle attributes such as mass or the rolling resistance of the tire and selected road attributes such as grade or road type (e.g. specific average speed depending on speed limit). Because individual driving styles and traffic congestions vary greatly, the real energy consumption fluctuates around the average consumption. For this reason the propulsion energy $\hat{E}_{pt}$ is split into a mean value $Ø\hat{E}_{pt}$ and into the corresponding deviation $\Delta\hat{E}_{pt}$ from $Ø\hat{E}_{pt}$ in the introduced prediction model. The mean value $Ø\hat{E}_{pt}$ will be predicted using vehicle-specific and route-specific parameters, whereas the deviation $\Delta\hat{E}_{pt}$ will be predicted using driving patterns collected in the backend. The mean average power does not fluctuate that much compared to $\Delta\hat{E}_{pt}$ as it depends largely on near-constant values during single segments along a selected route.

Since the ratio of recuperation depends on both the driving forces on user-settings and the vehicle operational strategy (e.g. on the state of charge of the traction battery), the powertrain energy in acceleration mode $\hat{E}_{pt,pos}$ is predicted independent of that in deceleration mode $\hat{E}_{pt,neg}$.

$$\hat{E}_{pt,pos} = \text{Ø}\hat{E}_{pt,pos} * \Delta\hat{E}_{pt,pos} \quad (4)$$

$$\hat{E}_{pt,neg} = \text{Ø}\hat{E}_{pt,neg} * \Delta\hat{E}_{pt,neg} \quad (5)$$

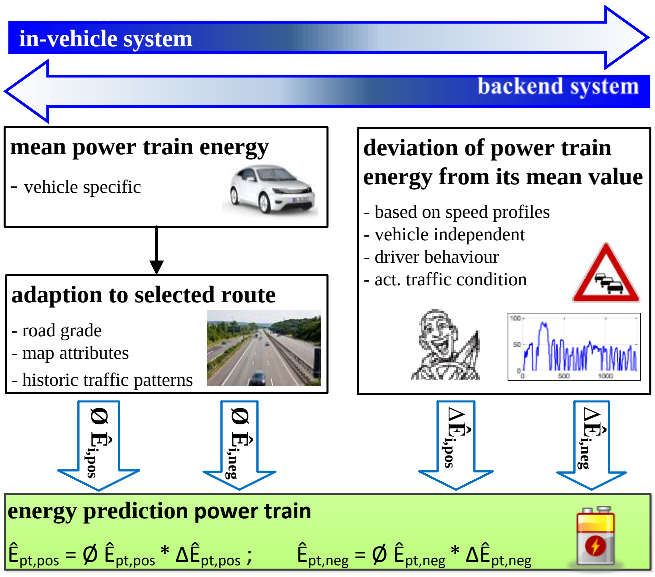

in-vehicle system
backend system
mean power train energy
- vehicle specific
adaption to selected route
- road grade
- map attributes
- historic traffic patterns
deviation of power train energy from its mean value
- based on speed profiles
- vehicle independent
- driver behaviour
- act. traffic condition
100
50
0
500
1000
$\text{Ø}\,\hat{E}_{i,pos}$
$\text{Ø}\,\hat{E}_{i,neg}$
$\Delta\hat{E}_{i,pos}$
$\Delta\hat{E}_{i,neg}$
energy prediction power train
$\hat{E}_{pt,pos} = \text{Ø}\,\hat{E}_{pt,pos} * \Delta\hat{E}_{pt,pos}$ ;
$\hat{E}_{pt,neg} = \text{Ø}\,\hat{E}_{pt,neg} * \Delta\hat{E}_{pt,neg}$

Figure 3: Energy prediction for power train

According to the requirements described in section 3.1., the energy should be predicted for every link *i* along the selected route. Thus, the mean energy $\text{Ø}\hat{E}_{pt}$ is predicted for every segment *i*. A function with specific energy values for existing road classes is used to predict $\text{Ø}\hat{E}_{pt,i}$ and the result is adapted according to additional road attributes such as slope or top speed. The calculation of the mean energy $\text{Ø}\hat{E}_{pt}$ is based on the prediction of the actual propulsion power of the vehicle. The prediction is adapted dynamically if the energy consumption of the vehicle varies.

Driving patterns that may influence emissions of conventional vehicles are described in [15]. Examples of these statistical driving patterns are average speed, standard deviation of speed or average acceleration. The same or adapted driving patterns can be used for EVs. A prediction model estimates the deviation of the propulsion energy $\Delta\hat{E}_{pt}$ by using a certain number of *j* driving patterns *F* (6). It is important that suitable driving patterns *F* are selected, which are independent of vehicle specific attributes, since the *F* has to be used for every type of EV.

$$F = (F_1, F_2, \dots, F_j) \quad (6)$$

The necessary driving patterns are extracted from the recorded speed profiles. The extraction process can be realized inside the vehicle so that specific, driver-related values for *F* can be stored in a local, in-vehicle database. This allows the influence of personal driving style in the actual vehicle to be used when predicting $\Delta\hat{E}_{pt}$.

The values for the driving patterns are anonymized and transmitted to the backend. The values *F* for all participating users are averaged and stored in an appropriate way in the backend.

The values *F* from both the in-vehicle driver-specific database and the crowd based database in the backend are used to predict $\Delta\hat{E}_{pt}$. Collected data only exists for frequently-travelled routes and might be outdated. The in-vehicle prediction system allows a verification of server-based data.

### 3.2.4 System architecture

If the prediction of the propulsion energy for every road segment *i* along the selected route and the prediction of the necessary auxiliary energy is combined, the total requested energy for a trip is (7):

$$\hat{E}_{ges} = \sum_{i=1}^{n} \left( \text{Ø}\hat{E}_{i,pt,pos} * \Delta\hat{E}_{i,pt,pos} + \text{Ø}\hat{E}_{i,pt,neg} * \Delta\hat{E}_{i,pt,neg} \right) + \\ + \hat{E}_{aux} \quad (7)$$

$\hat{E}_{aux}$ depends mainly on $t_{trip}$, but can be calculated for every segment *i* if the energy is requested for every link (e.g. for navigation applications). Figure 4 provides an overview of the whole system.

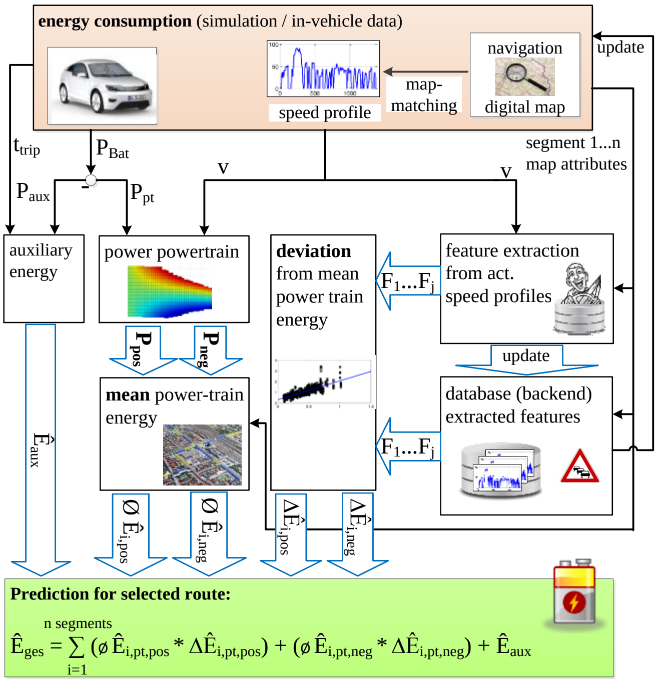

energy consumption (simulation / in-vehicle data)
speed profile
map-matching
navigation
digital map
update
$t_{trip}$
$P_{Bat}$
$P_{aux}$
-
$P_{pt}$
v
v
segment 1...n
map attributes
auxiliary energy
power powertrain
deviation from mean power train energy
feature extraction from act. speed profiles
$F_1 \dots F_j$
update
$P_{pos}$
$P_{neg}$
mean power-train energy
database (backend) extracted features
$F_1 \dots F_j$
$\hat{E}_{aux}$
$\text{Ø}\,\hat{E}_{i,pos}$
$\text{Ø}\,\hat{E}_{i,neg}$
$\Delta\hat{E}_{i,pos}$
$\Delta\hat{E}_{i,neg}$
Prediction for selected route:
$\hat{E}_{ges} = \sum_{i=1}^{n\ segments} (ø\,\hat{E}_{i,pt,pos} * \Delta\hat{E}_{i,pt,pos}) + (ø\,\hat{E}_{i,pt,neg} * \Delta\hat{E}_{i,pt,neg}) + \hat{E}_{aux}$

Figure 4: Server-based system for energy prediction

Several CAN-signals containing energy-relevant measured values are needed to realize the prediction system in an EV. Additionally, a navigation system with a digital map has to provide road attributes for the selected route. An appropriate communication device is necessary for receiving and transmitting data to the backend.

The database with collected speed profile features in the backend can be used to correct or update out-of-date map attributes or to enhance existing digital maps with new features as suggested in [16].

Since we chose a modular system architecture, different variants of realization are possible for the system, especially for the distribution of the modules between the backend and in-vehicle ECUs. For example, the calculation of the deviation of powertrain energy, the navigation system and the prediction of the necessary mean powertrain energy can be implemented in the backend. A discussion of these variants is not part of this paper.

## 4. SIMULATION TEST BED

Data recorded from a fleet of different electric vehicles is needed to develop the proposed server-based energy prediction system. Since the market penetration of electric vehicles is low and the costs of extensive fleet tests with a significantly large number of test persons are high, a different approach is used.

### 4.1. Data acquisition and system overview

In a system described in [17], smartphones are used to track the mobility behaviour of a group of participants in a fleet test. The vehicle speed, time of day and GPS position are recorded using the sensors in the smartphones, with a frequency of 1 Hz and the collected data is transmitted to a database in the backend. Since the selected test persons take their smartphones with them in the car, real world speed profiles are collected. A process that runs simultaneous to data acquisition process filters the raw data and cuts the data stream into segments to identify complete car rides, called tracks.

It is easy to collect a vast amount of real world driving cycles using this method. This is necessary because deviations that occur in reality in speed profiles depending on traffic congestion, time of day, route-specific impact factors or individual driving patterns have to be taken into account when developing the proposed system. The vehicle speed and the required trip time $t_{trip}$ are necessary for the energy prediction system; these can be extracted from the recorded tracks. In order to use attributes from digital maps, a map-matching algorithm is necessary (see section 4.2.).

The variance of the energy consumption of different EVs depending on their vehicle attributes or external conditions such as weather have to be considered too. Parameter variations that occur can be taken into account by using a simulation model (see section 4.3.). The real world speed profiles are used as input for the simulation model.

The developed system can be validated and optimized by comparing the predicted energy $\hat{E}$ and the simulated energy consumption $E_{sim}$. An overview of the complete simulation test bed is shown in Figure 5.

## 4.2. Map

A digital map has to be chosen in order to use route-specific attributes for the collected speed profiles. One possibility is the crowdsourced data of the OpenStreetMap (OSM) project, which is accessible under the Open Database Licence [18].

### 4.2.1 Use of OpenStreetMap data

Features such as road type, speed limit, number of lanes and length of the links in the road network are tagged in the OSM map. A brief evaluation is provided before the data is used for the energy prediction system.

OSM data is compared with TomTom's commercial datasets in [19]. Since OSM data contains a lot of roads for pedestrian and cycle navigation, the OSM dataset contains more data for the total street network but less data needed for car navigation. The data completeness depends on the selected region. In the Munich area, where most of the tracks were recorded, the OSM data contains around 97% of all objects compared to commercial datasets [19].

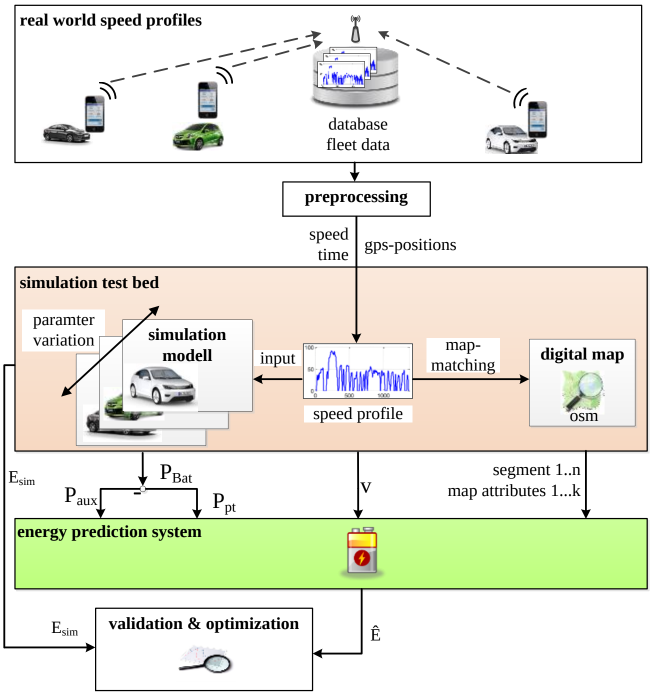


Figure 5: Simulation test bed for energy prediction system

Nevertheless, only 50% of all speed limits are tagged. However, if we focus on the area around Munich the speed limits can be updated using the collected car floating data. Compared to commercial data, the number of turn restrictions and one-way streets is much lower in OSM, but these disadvantages don't affect the energy prediction system as we use real world speed profiles. Another disadvantage is the missing traffic information, which has to be discussed later in this project.

OSM data can be used to develop our energy prediction system. But because OSM data is not routable, the *oms2po* script [20] is used to make the data set routable. The converted OSM data is stored in a *PostgresSQL/PostGIS* database.

Because OSM data is only two-dimensional, height data has to be added to the recorded speed profiles. A freely accessible source for a digital elevation model is the data collected by the Shuttle Radar Topography Mission (SRTM) [21]. The influence of different elevation models on the accuracy of eco-routing is analysed in [22]. The accuracy of the SRTM dataset is valid for eco-routing because the differences in energy consumption based on different elevation models are less than 1% on average for routes. Nevertheless, the deviation is up to 30% for a few single short segments, where large height differences occur. In the selected area around Munich the error is lower since a region with large height differences in the pre-Alps was selected in [22] for the analysis of elevation models. As suggested in [23], we use a two-dimensional, low-pass filter to filter the noise caused by detected buildings or trees during the radar mission. According to these results, the SRTM dataset can be used for our energy prediction system in the area around Munich.

Table 1: Road segments travelled in the selected area

| **Road type** | **Distance in road network [km]** | **Distance travelled [km]** |
|---|---|---|
| Motorway + trunk | 972 | 31412 |
| Primary + secondary | 3080 | 30210 |
| Tertiary + residential (intown) | 10449 | 5466 |
| Tertiary + residential (out of town) | 4125 | 1122 |

Most of the speed profiles were collected in the area around Munich, so that we focus on a specific area (11.2 < longitude < 12; 47.9 < longitude < 48.4). Table 1 shows the number of total distances in the road network in the specified area compared to the distance of collected data. The road types are distinguished according to the OSM definition. As seen in Table 1, we collected enough data for a statistical evaluation for a range prediction of EVs on the main roads in particular. The amount of data is not so high for side roads (in town and out of town), but the variation of road attributes is lower (e.g. inside a residential area) for these road types.

### 4.2.2 Map-Matching

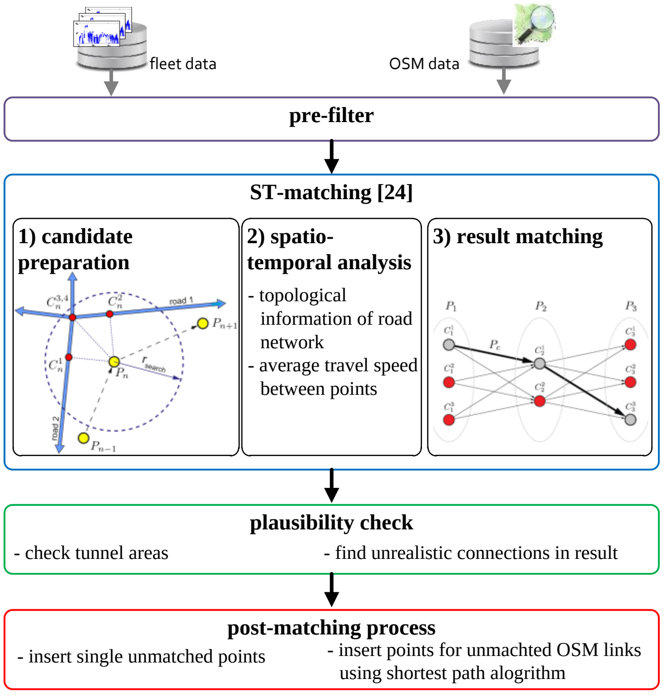
A map-matching algorithm is needed to use the OSM map features for the collected real world speed profiles. We define the following requirements for the map-matching algorithm in our simulation test bed of our prediction system:

- **Suitability for low-sampling rates with minimal input values**
  We are currently recording speed profiles at a sample rate of 1 Hz. As we plan to extend the database in future, the matching algorithm should work for speed profiles with lower sampling rates (e.g. traffic flow data, one point every 30s).
- **Matching accuracy**
  The main goal of the algorithm is matching accuracy. We want to use the data to develop intelligent prediction algorithms based on artificial intelligence (AI) or statistical methods, which require a high data accuracy. The quality of the recorded GPS tracks depends largely on the accuracy of GPS-sensors in the used smartphones. Furthermore, smartphones inside vehicles, especially in urban areas, often do not have a clear line-of-sight to GPS-satellites, leading to poor signal accuracy. The matching

accuracy has to be high for our system despite the described difficulties in recording speed profiles with smartphones.

- **Offline algorithm in the backend**
  The collected GPS-tracks should be matched in the backend. Problems such as the energy consumption of battery-powered smartphones are thus irrelevant. Besides, the entire track can be used for the matching process with offline algorithms, leading to a higher accuracy.

We chose the ST-matching algorithm based on a spatio-temporal analysis for our project [24] because of the execution time, mapping accuracy and flexibility for variable sampling rates. The algorithm works for a track consisting of a collection of GPS points $T = \{P_1, P_2, \dots P_n\}$ as follows (see figure 6):



fleet data
OSM data
pre-filter
ST-matching [24]
1) candidate preparation
$C_n^{3,4}$
$C_n^2$
road 1
$P_{n+1}$
$C_n^1$
$P_n$
$r_{search}$
road 2
$P_{n-1}$
2) spatio-temporal analysis
- topological information of road network
- average travel speed between points
3) result matching
$P_1$
$P_2$
$P_3$
$P_c$
plausibility check
- check tunnel areas
- find unrealistic connections in result
post-matching process
- insert single unmatched points
- insert points for unmachted OSM links using shortest path alogrithm

Figure 6: Overview of the map-matching algorithm

$c$ possible candidate points $(C_n^1, C_n^2, \dots C_n^c)$ are found on the road network within a defined neighbourhood of a measured GPS point $P_n$ . The topological information of the road network is taken into account in the spatial analysis and a so-called observation probability is calculated for each candidate point $C_n^c$. In the temporal analysis for all candidate points $C_n^c$ , a so-called transmission probability is calculated on the basis on the distance and travelled speed between two neighbouring candidate points $P_n$ and $P_{n-1}$. A graph consisting of the set of candidate points $( C_n^1, C_n^2, \dots )$ connecting all candidate points of neighboured GPS points $P_n$ and $P_{n-1}$is created after the spatio-temporal analysis. The best matching path sequence with the highest score of both probabilities is selected (e.g.$P_c$: $C_1^1 \rightarrow\ C_2^1 \rightarrow\ C_3^3 \rightarrow \cdots$) as a result.

Problems of GPS accuracy for automotive applications in urban canyons are known. We extend the known ST-matching algorithm to increase the mapping accuracy since we use smartphone sensors. As an offline algorithm is selected, not every data point of the recorded track has to be used for the matching algorithm, so that the accuracy of the matching result can be increased. Data points with low values for the calculated probabilities are filtered and a link-based plausibility check is carried out after map-matching. Unmatched points or segments are corrected in a post-matching process, so that in the end, all recorded data points are matched (see Figure 6).

GPS points of a poor quality are ignored for the ST-matching process. Thus, GPS points with high values for horizontal dilution of precision (HDOP) or with low GPS-speed (especially during standstill inaccuracy is high) are filtered. In addition, GPS points that differ greatly from the heading of the previous points are not taken into account.

The result of the ST-matching algorithm is checked for plausibility. To begin with, tunnels or long bridges often appear in urban areas where the GPS-signal reception is not possible or noisy. The ST-matching algorithm has problems in these areas because of the illogical GPS signals. Matched points in these areas are checked for plausibility according to their transmission probability; wrongly matched points are identified and the whole segment is set as unmatched so that it can be corrected in the post-matching process.

Next, the distance between matched points is compared to the distance calculated according to the average speed between the two points for each connection between neighbouring GPS points. In the event of any deviation or in case of low values for the transmission probability, all of the GPS points for the actual OSM link are set as unmatched. Since we use an offline-method, this plausibility check is carried out in both directions, from the beginning of the track to the end and vice versa.

The post-matching process inserts all unmatched points and OSM segments of a recorded track into the validated matching results. Single points are inserted according to their GPS-positions and speed-values. If points of complete OSM-links have to be inserted, the points are distributed along a probable route according to their speed-values and GPS-positions. The probable route is calculated using a shortest path algorithm.

850 randomly chosen tracks are used to evaluate the map matching system. The accuracy of the map matching algorithm is evaluated for every navigable OSM-segment by observing two criteria:

$$\Delta l_s = \frac{|\int p_{gps}(lat,lon)\; ds\ - l_{OSM}| - l_{tol}}{l_{OSM}} \qquad (8)$$

$$\Delta l_v = \frac{|\int v_{gps}\, dt - l_{OSM}| - l_{tol}}{l_{OSM}} \qquad (9)$$

In (8), the relative deviation $\Delta l_s$ is the distance calculated from the matched GPS points $p_{gps}(lat, lon)$ and defined segment length $l_{OSM}$. Since GPS points are recorded with 1 Hz, we allow a tolerance $l_{tol}$ of the travelled distance in one second. $\Delta l_s$ does not show if the points are matched on the correct street. This is why we use $\Delta l_v$, which is the relative deviation of the integrated GPS speed from the defined segment length $l_{OSM}$. The integrated value for the $v$ have to be consistent with the distance defined in OSM if the track is matched correctly.

The result of the standalone ST-matching algorithm (ST-M) as was introduced in [24] is compared with the result of complete map-matching system, including a plausibility check and post-matching process (P-M). About 2.8% of the points could not be matched at all for different reasons (e.g. different routes because of construction work, driving on parking lots, data recording problems). Figure 7 shows the cumulative probability density functions (CDF) of the defined evaluation criteria.

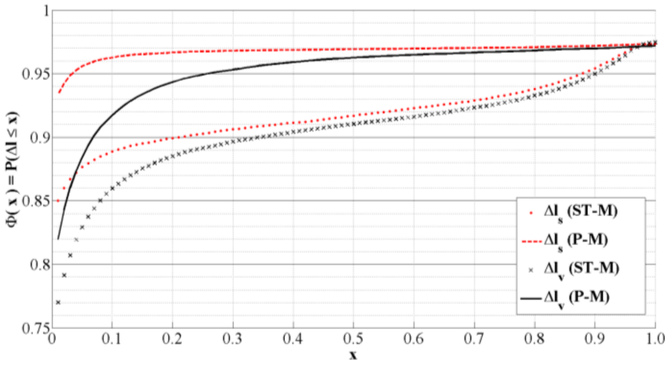


Figure 7: Evaluation of map matching

The results show that the post-matching process clearly improves the matching accuracy. The CDF for $\Delta l_v$ is lower than the CDF for $\Delta l_s$, which does not mean that the points are matched to undriven roads. This is because the GPS speed is not accurate enough so that the integrated GPS-speed of a segment does not correspond exactly to the defined length of the segment. All in all, the match quality is good enough for the development of the proposed approach.

### 4.3. Simulation model

A simulation model is used to calculate the energy consumption. The collected speed profiles are used as inputs in the model. A PI-controller is used as a driver model to follow the desired speed profile. Further input values are the slope calculated using the SRTM data and parameters for weather conditions, which are necessary for modelling the power consumption of the auxiliaries.

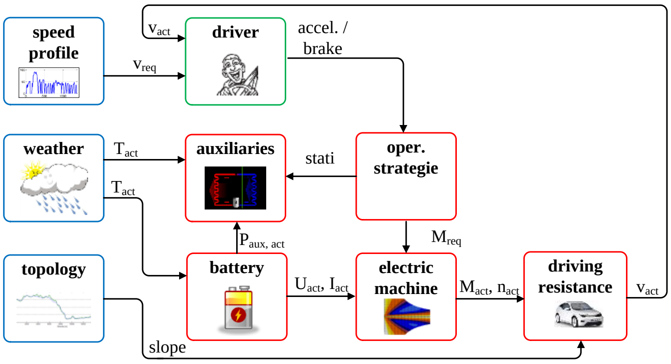


Figure 8: Simulation model

The vehicle model (see Figure 8) consists of various component models. We used an equivalent circuit model parameterized by the results of measured lithium cells for the battery. The electric machine is modelled with an efficiency map based on validated measurements. The efficiency map and battery parameters used are scaled for the adaption to vehicle parameters (e.g. max. acceleration torque). Parameters for the operational strategy during braking (e.g. max. deceleration for recuperation) can also be adapted.

Parts of the system have to be independent of vehicle parameters when developing a server-based energy prediction model. Simulations with varied vehicle parameters are needed to test these prediction systems. For this reason validated component models are used.

Eight different vehicle models are simulated for the estimation of propulsion power; the most important parameters are shown in Table 2. The efficiency map of the electric machine and battery parameters of vehicles 1 to 4 are different from the other vehicles. $\eta_{pt}$ is a constant powertrain efficiency factor (e.g. transmission).

Table 2: Most important parameters for simulated vehicles

| Vehicle parameter | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|---|---|---|---|---|---|---|---|---|
| Mass [kg] | 1200 | 900 | 900 | 1300 | 1900 | 1600 | 1600 | 1100 |
| cd value [-] | 0.33 | 0.28 | 0.28 | 0.28 | 0.32 | 0.32 | 0.32 | 0.28 |
| Cross-sect. Area [m$^2$] | 2.04 | 2.04 | 2.04 | 1.9 | 1.9 | 1.9 | 1.9 | 1.9 |
| $\eta_{pt}$ [-] | 0.92 | 0.92 | 0.81 | 0.92 | 0.98 | 0.98 | 0.98 | 0.92 |
| Rolling resist. [-] | 0.015 | 0.015 | 0.015 | 0.015 | 0.007 | 0.01 | 0.01 | 0.007 |
| max. deceleration for recuperation [ms$^{-2}$] | -0.8 | -0.8 | -0.5 | -0.6 | -0.6 | -0.6 | -0.6 | -0.4 |

## 5. ESTIMATION OF POWERTRAIN POWER

An appropriate clustering of the road segments based on energy consumption is necessary to predict the propulsion power since the average energy consumption varies according to different impact factors. The necessary mean energy is predicted for every road class along the chosen route.

### 5.1. Clustering of road segments based on energy consumption

A clustering method based on road attributes is used to define road classes based on energy consumption, since these are available in common for every link along a route. We analysed the influence of the following map attributes: road type (e.g. highway, primary, residential), maximum speed, integrated topographical gradient upwards $\Delta h_{pos}$ and downwards $\Delta h_{neg}$ per navigable link and the number of crossing roads per km. We simulated the propulsion energy $E_{pt,pos}$ and the recuperated energy $E_{pt,neg}$ for every track for a clustering based on energy consumption (see Table 1).

A Kruskal-Wallis test was used to rank the influence of the tested map attributes on $E_{pt,pos}$ and $E_{pt,neg}$ and to identify the most significant road attributes. This test is a non-parametric method to test for differences among two or more groups and does not require a normal distribution [25].

As a result, the road type and maximum speed are the most significant map attributes for $E_{pt,pos}$ and $E_{pt,neg}$. A clustering by one map attribute is not possible since the average energy consumption on a road type varies with different speed limits and the energy consumption of some different road types (e.g. for motorways and primary) is similar. The following method is used for clustering the road segments:

1) **Calculation of the mean energy consumption of different vehicles depending on speed limit and road type**
Since the road classes have to be used for all different types of vehicles, the necessary energy for the recorded tracks (see Table 1) is simulated for different vehicle attributes (see Table 2). Because the ratio of recuperation depends on vehicle attributes, the energy to overcome the driving resistances is used for clustering. According to the defined speed limit and road type, the calculated distance-based energy $E_{wl,pos}$ and $E_{wl,neg}$ of every navigable OSM-link is used to calculate a mean value for every combination of the two road attributes for every of the 8 simulated vehicles ($\emptyset E_{wl,pos}$ and $\emptyset E_{wl,neg}$). Only observations with a significantly high number of measurements are taken into account for clustering. Moreover, the median value is used instead of the mean value to filter out statistical outliers.
2) **Clustering the mean energy consumption for different vehicles**
A two-dimensional feature vector $x_n = [\emptyset E_{wl,pos}, \emptyset E_{wl,neg}]$ exists for every observation $n$ (every existing combination of speed limit and road type). These $n$ objects are split into $k$ clusters using the k-means algorithm, which is a simple unsupervised learning algorithm for solving clustering problems [26]. The number of clusters $k$ has to be chosen before starting the algorithm, which assigns each observation $n$ to one of the $k$ clusters by minimizing the metric distance to the centroids of the $k$ clusters.

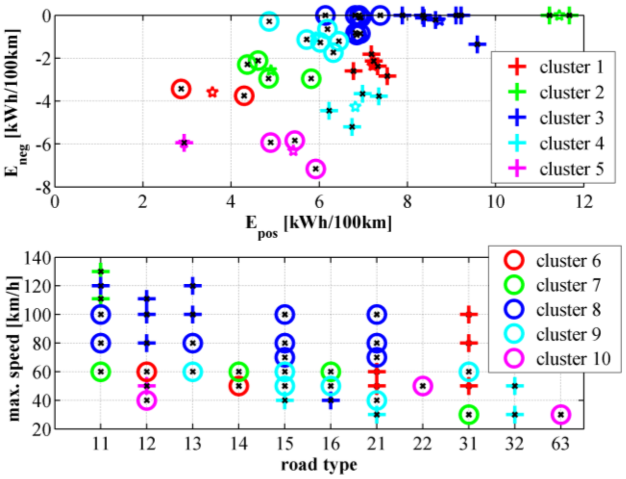


Figure 9: Road segment clustering based on simulated energy consumption for vehicle 7

An example of clustering $x_n = [\emptyset E_{wl,pos}, \emptyset E_{wl,neg}]$ for vehicle 7 is shown in Figure 9. The upper part shows the assignment to the 10 clusters. In the lower part, the result of clustering is arranged by maximum speed and road type. The nomenclature of the road types can be found in Figure 10; the names of the road types conform with the definition in OSM [27].

The results in Figure 9 show differences according to the different road types. On driveways in particular (e.g. road type 12, 14 or 16) and on roads in residential areas, the part of the energy potentially to be recuperated is higher than on arterial roads (e.g. road type 15).

After running the cluster algorithm with a different number of clusters we set the number of clusters to ten. On the one hand, fewer clusters lead to higher silhouette coefficients providing a representation how well each object lies within its cluster. On the other hand, more clusters do not seem to be necessary.

3) **Definition of road classes based on energy consumption**

The clusters for each vehicle are sorted ascending by energy consumption. The final road classes based on energy consumption are defined in a heuristic way. The vehicle-specific clustering results are combined to one final clustering. The clustered road segments are adapted by considering some additional requirements:

- separation into in-town and out-of-town segments
- join links to the corresponding road type (except road type 11, because of the length of motorway links)
- separation of completely different road types in terms of traffic (e.g. secondary and motorway)

These additional requirements are necessary since the energy consumption can be similar for complete different road classes with various road types and speed limits (e.g. same energy consumption on segments in in-town residential area and out-of town motorways with a speed limit). Finally twelve road classes are defined, which are shown in Table 3. If not enough recorded data exists for clustering a feature vector, the road class from similar feature vectors is used.

## 5.2. Estimation of powertrain energy

In a first step the module to estimate the powertrain energy is realized, which will be used for adaption according to route-specific impact factors.

Table 3: Road classes based on energy consumption

| speed limit [km/h] | road type | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | motorway | | trunk | | primary | | secondary | | tertiary | residential | living street |
| | 11 | 12 | 13 | 14 | 15 | 16 | 21 | 22 | 31 | 32 | 63 |
| 30 | *1* | *1* | *1* | *1* | *1* | *1* | **2** | *2* | **3** | **2** | **5** |
| 40 | *4* | **1** | *4* | *4* | **6** | **6** | *8* | *8* | *8* | *8* | *5* |
| 50 | *4* | **1** | *4* | *4* | **6** | **6** | **8** | **8** | **8** | **8** | *5* |
| 60 | **4** | **4** | **4** | **4** | **6** | **6** | **7** | *7* | **7** | *7* | |
| 70 | *10* | *10* | *10* | *10* | **7** | *7* | **7** | *7* | *7* | *7* | |
| 80 | **10** | **10** | **10** | *10* | **7** | *7* | **7** | *7* | **9** | | |
| 100 | **10** | **10** | **11** | *11* | **11** | *11* | **7** | *7* | *9* | | |
| 110 | **12** | **10** | *11* | *11* | *11* | *11* | | | | | |
| 120 | **12** | *11* | **11** | *11* | *11* | *11* | | | | | |
| 130 | *12* | *11* | *11* | *11* | | | | | | | |

**x** **result of clustering**

*x* *heuristic definition*

road type 12, 14, 16 and 22 are the links to the corresponding road types 11, 13, 15 and 21

### 5.2.1. System overview

We defined the following requirements for the prediction of the mean powertrain energy:

- consideration of all relevant vehicle-specific impact factors
- prediction of a mean value for every road class
- fast online adaption to changing vehicle parameters
- easy applicability to different electric vehicles

As physical modelling of the powertrain with nonlinear equations is complicated and depends on many varying parameters, we aim to use dynamic system identification methods to predict the necessary propulsion energy. Thus, the number of impact factors used can be reduced and a dynamic adaption to different vehicles or vehicle parameters is possible.

As shown in (1) and (2), the necessary propulsion power depends largely on the chosen speed $v$ and acceleration $a$. Apart from the gradient of the road in $P_{slope}$, the additional vehicle-specific impact factors on the propulsion power (e.g. drivetrain efficiency $\eta_{pos}/\eta_{neg}$ or the mass of the vehicle $m$) display a dynamic behaviour but do not vary as frequently as $a$ or $v$. For this reason, it is possible to describe the stationary behaviour of the nonlinear correlation between propulsion energy consumption and $a$ and $v$ with a look-up table, called the powertrain characteristic map (PCM). The influence of all relevant vehicle parameters such as the mass $m$ or the drivetrain efficiency is considered in the PCM. The influence of the road gradient is taken into account later, when the mean energy is adapted to road parameters (see Figure 10). The PCM can be estimated online in the vehicle by using system identification methods and is continuously adapted to parameter changes.

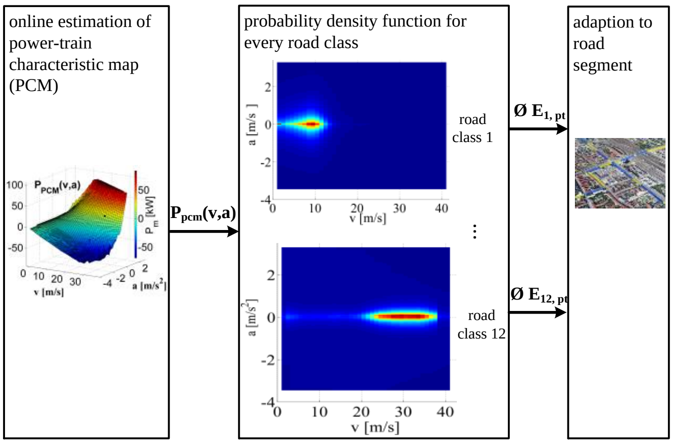


Figure 10: Estimation of the powertrain energy

The frequency of different values for the speed $v$ or the acceleration $a$ depend on different impact factors such as driving style or chosen road type. Because the influence of these factors is considered in subsequent modules in our system, the calculated mean propulsion energy consumption has to be independent of these factors, but should take into account the vehicle specific impact factors.

We use a two-dimensional probability density function (PDF) of the speed $v$ and acceleration $a$ for every road class for normalization. The PDF is created on the basis of the recorded tracks in the database. The road class-specific and distance-based propulsion energy consumption can be calculated with the PDF and the estimated powertrain characteristic map. This allows all relevant vehicle-specific parameters to be considered in the predicted propulsion energy.

### 5.2.2. Online identification of a powertrain characteristic map

The powertrain characteristic map describes the necessary power $P_{pcm}(v,a)$ to overcome the driving resistances depending on vehicle speed $v$ and acceleration $a$. An online parameter identification method is used to estimate $\hat{P}_{pcm}(v,a)$. One interpolation node of $P_{pcm}(v,a)$ according to the actual value of $a$ and $v$ can be updated at time $t$ with the actual measured power $P_m(t)$. $P_m(t)$ is calculated using the consumed power at the battery, less the consumption of the auxiliaries $P_{aux}$. As the influence of the road gradient is considered later, the corresponding driving force $P_{slope}$, depending on the gradient α and the vehicle mass $m$, has to be subtracted:

$$P_{m,\{v(t),a(t)\}}(t) = U_{bat}(t) * I_{bat}(t) - P_{aux}(t) - P_{slope}(t) \qquad (10)$$

The powertrain characteristic map $P_{pcm}(v,a)$ is defined as a two-dimensional, grid-based look-up-table of the size $r$ x $s$. It consists of interpolation nodes $y_{r,s}$ located on the grid lines $v_1,\dots,v_i,\dots,v_r$ and $a_1,\dots,a_j,\dots,a_s$ defined by the vehicle speed $v$ and acceleration $a$. As $a$, $v$ and $P_{m,\{v(t),a(t)\}}(t)$ are based on noisy measurements, we have a noisy measurement vector for every interpolation node $y_{v_i,a_j}$. Recursive least-square (RLS) methods are often applied to estimate the optimal value $\hat{y}_{v_i,a_j}$ by minimizing the sum of the squared error of the measurements [28] [29]. The estimation of all interpolation nodes forms the estimated powertrain characteristic map $\hat{P}_{pcm}$.

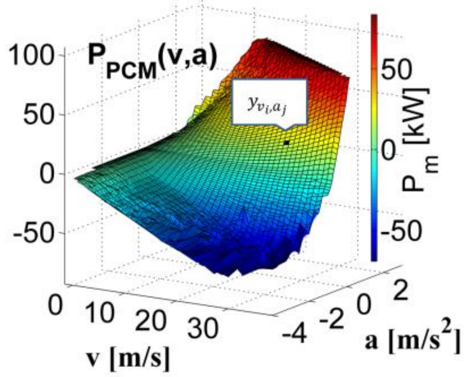


Figure 11: Powertrain characteristic map

Thus, we use an RLS-algorithm for every interpolation node $y_{v_i,a_j}$ of the powertrain characteristic map $P_{pcm}(v,a)$ (see Figure 11).

The RLS-algorithm of our system is can be simplified as we use a constant value for every interpolation node. We would like to explain the recursive least square algorithm in detail for the interpolation node $y_{v_i,a_j}$ with the measured input $y_m$ and the predicted output $\hat{y}$ (11):

$$y_m = P_m \qquad \hat{y} = \hat{P} \tag{11}$$

$$e(t) = y_m(t) - \hat{y}(t) \tag{12}$$

The RLS-algorithm is used for adaptive identification systems and only uses the prediction result of the previous time step *t-1*. Minimizing the square sum of error *e(t)* (12), the predicted value $\hat{y}(t)$ can be calculated as follows [29]:

$$\hat{y}\,(t) = \hat{y}\,(t-1) + \tfrac{1}{t}[y_m(t) - \hat{y}\,(t-1)] \tag{13}$$

The influence of the error on the adaption of the prediction becomes smaller with an increasing measurement time. Several RLS-algorithms exist with weighting factors or forgetting factors [28]. We use an RLS-algorithm with a dynamic weighting factor $\omega(t)$ for the dynamic adaption to changing vehicle parameters.

$$\hat{y}\,(t) = \hat{y}\,(t-1) + \frac{1}{\omega(t)}[y_m(t) - \hat{y}\,(t-1)] \tag{14}$$

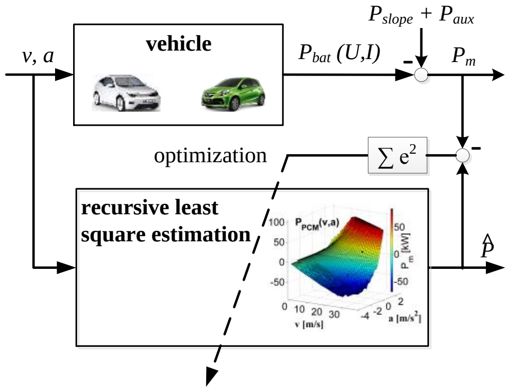


Figure 12: Recursive least-square estimation of powertrain characteristic map

A global weighting factor $\omega(t)$ for the whole powertrain characteristic map $P_{pcm}(v,a)$ is used. The value of $\omega(t)$ is set depending on the average error of a fixed number of the last appearing interpolation nodes. The weighting factor $\omega(t)$ is decreased with an increasing e in order to accelerate the dynamic adaption.

The prediction of $P_{pcm}(v,a)$ is evaluated by simulating a real world track with the parameters of vehicle 8 (see Table 2) . During the length of the track three different parameter changes are simulated, which lead to a significant change in the powertrain characteristic map:

- t=5 min: increase of mass (+300 *kg*)
- t=20 min: increase of rolling resistance (+0.002)
- t=45 min: decrease of maximum deceleration for recuperation (- 0.6 $ms^{-2}$)

The first results show that an RLS-algorithm with a dynamic weighting factor is possible for the prediction of $P_{pcm}(v,a)$. The powertrain characteristic map $P_{pcm}(v,a)$ respectively the estimated $\hat{P}_{pcm}(v,a)$ are calculated according the simulated power $P_m(t)$ along the selected track. Possible prevailing measuring inaccuracies (e.g. for the calculation of $P_{slope}$) are taken into account. $P_{pcm}(v,a)$ and $\hat{P}_{pcm}(v,a)$ are used to calculate $\emptyset E_{2,pt}$ respectively $\emptyset \hat{E}_{2,pt}$.

The error of the estimation for the calculated mean energy consumption of several road classes is shown in Figure 13. The error and the setting time of the powertrain characteristic map depends on the operating points of the vehicle. Since each node of $P_{pcm}(v,a)$ is corrected separately, a correction of the mean energy consumption is only possible if the significant nodes of the corresponding probability density function (*PDF*) appear in the track. This is the reason why $\emptyset E_{10,pt}$ is not corrected at all in the first 20 minutes of the track. The adaption to the parameter changes is fast, if the significant speed and acceleration values for a road class appear in the track (e.g. around *t=17 min* for road class 6). The time to correct the $P_{pcm}(v,a)$ due to impact factors on recuperation depends on the number of decelerations phases.

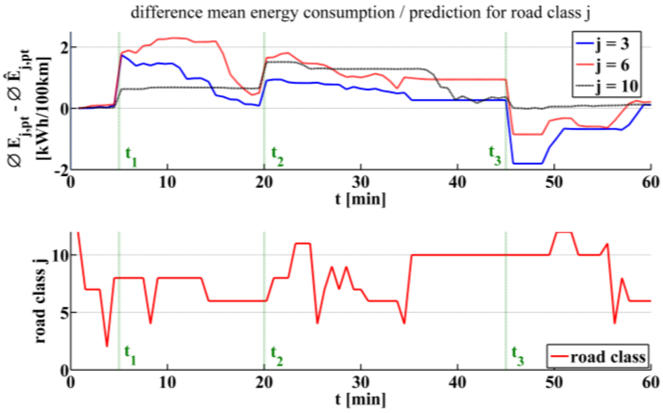


Figure 13: Estimation of powertrain energy for different road classes

The accuracy of the estimation is evaluated by analysing the total energy of the complete track. In Figure 14 the consumed energy is compared with the calculated energy by using the powertrain characteristic map $P_{pcm}(v,a)$ at the beginning of the trip and with the estimated energy according to the continuously corrected $\hat{P}_{pcm}(v,a)$. The difference between real energy consumption and the predicted energy is small. The results show that the correction of the $P_{pcm}(v,a)$ of driven road classes is fast enough.

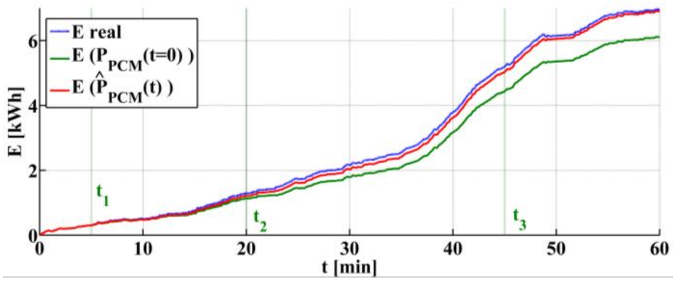


Figure 14: Estimation of powertrain energy

An accurate prediction of the energy consumption is also necessary for road classes, which are actually not driven, but will be driven later in the trip. Thus, a faster adaption is necessary for these operating areas of $P_{pcm}(v,a)$, for which no measurement values are available due to the actual vehicle speed or acceleration. One possibility is the use of extrapolation algorithms for these operating areas.

## 6. CONCLUSION

This paper has introduced a concept for predicting the energy consumption of EVs on a selected route. Due to the high number of impact factors and the complexity of their interdependence, a concept is proposed consisting of an in-vehicle part and a part realized on a server in the backend. Crowd sourced data from several vehicles is used for energy prediction in this kind of system.

Speed profiles are used to exchange data related to the energy consumption of EVs via the backend. The speed profiles are used to predict the deviation of the propulsion energy from its mean value. The model to predict the deviation is almost independent of the vehicle attributes. The necessary mean energy consumption of the powertrain is predicted in an in-vehicle system.

The first part to be introduced is the prediction of the mean energy consumption of the powertrain. We used a least-mean square algorithm to estimate a characteristic map of the drivetrain. The estimation is adapted online due to parameter variations and is based on only a few measured in-vehicle signals. The results show appropriate behaviour for the road classes, which are driven during the track. The estimation has to be expanded to correct the prediction of the mean energy consumption for the actual non driven road classes.

In future, the part of the system to predict the deviation from mean energy consumption will be realized. Intelligent algorithms will be used to consider the individual driving behaviour and the impact of prevailing traffic conditions.

## 7. ACKNOWLEDGMENTS

We would like to thank the German Federal Ministry of Transport, Building and Urban Development for funding the project "eflott", in which some of the speed profiles used this project were collected. The work described in this paper was conducted with basic research fund of the Institute of Automotive Technology. This article contains the results of student theses from Moritz Hann, Fabian Stoffels, Andreas Stroka and Konrad Vogt. The authors would like to thank them for their contributions.

## 8. REFERENCES

[1] Deloitte "Unplugged: Electric vehicles realities versus consumer expectations", Sep. 2011.

[2] Sachenbacher, M., Leucker, M., Artmeier, A. and Haselmayr, J. "Efficient Energy-Optimal Routing for Electric Vehicles" AAAI Conference on Artificial Intelligence Special Track on Computational Sustainability, San Francisco, Aug. 2011.

[3] Themann, P. and Eckstein, L. „Modular Approach To Energy Efficient Driver Assistance Incorporating Driver Acceptance" IEEE Intelligent Vehicles Symposium, Alcalá de Henares, Jun. 2012, pp.1023-1028.

[4] Hoch, N., Zemmer, K., Werther, B. and Siegwart R. "Electric Vehicle Travel Optimization – Customer Satisfaction Despite Resource Constraints", IEEE Intelligent Vehicles Symposium, Alcalá de Henares, Jun. 2012, pp.172-177.

[5] Dorrer, C. "Effizienzbestimmung von Fahrweisen und Fahrerassistenz zur Reduzierung des Kraftstoffverbrauchs unter Nutzung telematischer Informationen", Dissertation, Renningen, expert-Verlag, Renningen, 2004.

[6] Demestichaas, K., Masikos, M., Adamopoulou, E., Dreher, S. and Diaz de Arkaya, A. "Machine-Learning Methodology for Energy Efficient Routing", Proc. of the 19th ITS World Congress, Vienna, Oct. 2012.

[7] Karbowski, D., Pagerit, S. and Calkin A. "Energy Consumption Prediction of a Vehicle along a User-Specified Real-Word-Trip", Proc. from Electric Vehicle Symposium (EVS26), Los Angeles, May 2012.

[8] Kraschl-Hirschmann, K. and Fellendorf, M., „Estimation Energy Consumption for Routing Algorithms" IEEE Intelligent Vehicles Symposium, Alcalá de Henares, Jun. 2012, pp.258-263.

[9] Boriboonsomsin, K., Barth, M., Zhu, W. and Vu, A. "Eco-Routing Navigation System Based on Multisource Historical and Real-Time Traffic Information", IEEE Transactions on Intelligent Transportation Systems, vol.13, no.4, Dec. 2012, pp.1694-1704.

[10] Yu, H., Tseng, F. and McGee, R. "Driving Pattern Identification for EV Range Estimation", IEEE International Electric Vehicle Conference (IEVC), Greenville, Mar. 2012, pp.1-7.

[11] Buckl, C., Schätz, B., Fehling, M., Kuhn, K., Klein, C. et al. "Mehr Software (im) Wagen: Informations- und Kommunikationstechnik (IKT) als Motor der Elektromobilität der Zukunft" Abschlussbericht des BMBF-geförderten Verbundvorhabens "eCar-IKT-Systemarchitektur für Elektromobilität", 2011.

[12] Werther, B. and Hoch, N. "E-Mobility as a Challenge for New ICT Solutions in the Car Industry", in TGC 20011, LNCS 7173, Springer Berlin/Heidelberg, 2012, pp46-57.

[13] Schön, S., Steinmann, R., Friesenecker, M., Hackl, R. and Rehrl, K. „Crowdsourcing bei Geo- und Reisedatenprojekten – was macht OpenStreetMap, Waze & Co. erfolgreich?" Salzburg Research Forschungsgesellschaft m.b.H., Salzburg, 2012.

[14] Henning, H. "Der A1 e-tron im Flottenversuch", in Proc. for Conference on Future Automotive Technologies, Garching, Mar. 2012.

[15] Ericsson, E. "Independent driving pattern factors and their influence on fuel-use and exhaust emission factors" , Transp. Res. Part D: Transport and Environment, part.6, 2001, pp.325-345.

[16] Carisi, R., Giordano, E., Pau, G. and Gerla M. "Enhancing in Vehicle Digital Maps via GPS Crowdsourcing", in IEEE Int. Conference on Wireless On-Demand Network Systems and Services, Bardonecchia, Jan. 2011, pp.27-34.

[17] Ritzer, J., Pesce, T., Schickram, S. and Lienkamp, M. „Einsatz von Smartphones zur Erfassung von mobilitätsrelevanten Daten in Flottenversuchen – Mobilitätsuntersuchungen mit Elektrofahrzeugen", Automobiltechnisches Kolloquium, München, Mar. 2011.

[18] OpenStreetMap, http://www.openstreetmap.org/, accessed: 2013-01-30.

[19] Neis P., Zielstra D. and Zipf A. "The Street Network Evolution of Crowdsourced Maps: OpenStreetMap in Germany 2007–2011", in Future Internet vol.4, no.1, 2012, pp.1-21.

[20] OSM2PO: OpenStreetMap Converter And Routing Engine for Java, http://osm2po.de/, accessed: 2013-01-30.

[21] SRTM data, www2.jpl.nasa.gov/srtm/, accessed: 2013-01-30.

[22] Bachhofer F. "Einfluss der vertikalen Genauigkeit von DGM auf das EcoRouting von Elektrofahrzeugen" in Angewandte Geoinformatik 2011, Wichmann-Verlag, Berlin, 2011, pp.338-346.

[23] Smith B. and Sandwell D. "Accuracy and resolution of shuttle radar topography mission data", in Geophysical Research Letters vol.30, no.9, 2003.

[24] Lou, Y., Zhang, C., Zheng, Y., Xie, X.; Wang, W. and Huang, Y. „Map-matching for low-sampling GPS trajectories", Proc. of the 17th ACM SIGSPATIAL International Conference on Advances in Geograhic Information Systems, Seattle, 2009, pp.352-361.

[25] Kruskal, W. and Wallis W., "Use of Ranks in One-Critereon Variance Analysis" In: Journal of the American Statstical Association 47, no. 260, 1952, pp.583-621.

[26] Lloyed, S. "Least square quantization in PCM" in IEEE Trans. on Information Theory, vol. 28, 1982, pp.129-137.

[27] Map-features, http://wiki.openstreetmap.org/wiki/Map_Features, accessed: 2013-02-05.

[28] Isermann, R. and Münchhof, M., „Identification of Dynamic Systems" Springer, Berlin, Heidelberg, 2011, ISBN: 978-3-540-78878-2, pp.203-316.

[29] Ljung, L. "System Identification: theory for the user 2." Prentice Hall, Upper Saddle River, New York, 1999, ISBN: 0-13-656695-2.