

# Model-Based Remaining Driving Range Prediction in Electric Vehicles by using Particle Filtering and Markov Chains

Javier A. Oliva<sup>1</sup>, Christoph Weihrauch<sup>1</sup> and Torsten Bertram<sup>1</sup>

<sup>1</sup>*Institute of Control Theory and Systems Engineering, Technische Universität Dortmund, Otto-Hahn-Str. 4, 44227 Dortmund, Germany (e-mail: javier.oliva@tu-dortmund.de)*

---

## Abstract

The remaining driving range (RDR) has been identified as one of the main obstacles for the success of electric vehicles. Offering the driver accurate information about the RDR reduces the range anxiety and increases the acceptance of electric vehicles. The RDR is a random variable that depends not only on deterministic factors like the vehicle's weight or the battery's capacity, but on stochastic factors such as the driving style or the traffic situation. A reliable RDR prediction algorithm must account the inherent uncertainty given by these factors. This paper introduces a model-based approach for predicting the RDR by combining a particle filter with Markov chains. The predicted RDR is represented as a probability distribution which is approximated by a set of weighted particles. Detailed models of the battery, the electric powertrain and the vehicle dynamics are implemented in order to test the prediction algorithm. The prediction is illustrated by means of simulation based experiments for different driving situations and an established prognostic metric is used to evaluate its accuracy. The presented approach aims to provide initial steps towards a solution for generating reliable information regarding the RDR which can be used by driving assistance systems in electric vehicles.

*Keywords: electric vehicle, remaining driving range, driving assistance system, particle filter, Markov chain*

---

## 1 Introduction

The remaining driving range (RDR) represents one of the main obstacles for the success of electric vehicles. The limited range together with the long charging time has been pointed out as the main technical factors affecting the acceptance of electric vehicles. A successful integration of electric vehicles into future mobility concepts requires not only the development of faster battery charging systems and facilities but also the application of advanced driving assistance systems that support the driver with reliable information regarding the vehicle's driving range. To accomplish this, algorithms that accurately model the driving load of the road ahead, and thereby better predict the RDR, are required. Unfortunately future driving conditions are difficult to predict. The driving style, road conditions or the traffic situation are some of the factors that stochastically affect the

RDR. The randomness of these factors makes the RDR prediction problem difficult.

To our knowledge, few works have been published on the field of the RDR prediction. In [1] a driving pattern identification based approach was introduced, where a library of driving patterns is used for predicting the driving load of an electric vehicle in dependence of the road ahead. In [2] a two-step prediction algorithm is applied to determine the RDR. Nine factors in total for determining the RDR are considered. First, a rough RDR prediction is done if the remaining battery energy is higher than a preset critical battery energy threshold. If the threshold is crossed, a precise range estimation takes place. In [3] a method that combines the use of a web server, a digital map and a mobile application is presented. The mobile device sends the position of the vehicle and the current state of charge (SOC) of the battery to the web server, which

first estimates the energy consumption along all possible routes and then, based on the SOC, it calculates the maximum driving range. The main drawback with these approaches is that the RDR is treated as a deterministic quantity and the uncertainties given by the driving behavior or by the traffic situation are not taken into account.

Our aim is to introduce an approach that accounts for these sources of uncertainty. In this paper the RDR prediction problem is approached by combining particle filtering with Markov chains. A two-step algorithm, adapted from a prognostics framework [4], is used to this aim. In the first step, the battery states are estimated using a particle filter. In the second step, the probability distribution of the RDR is determined by using the estimated states and by propagating the particles through a driving profile generated stochastically via Markov chains. The remaining of this paper is organized as follows. Section 2 formulates the problem of the RDR prediction and introduces the proposed prediction architecture. In section 3 the model of the electric vehicle is described. Section 4 briefly discusses the particle filter used for the estimation of the states of the model. Section 5 discusses the proposed approach for the RDR prediction. Section 6 presents simulation results for demonstrating and validating the approach. Section 7 concludes the paper and provides an overview of the future work.

## 2 Prediction Methodology

The RDR is formally defined as the actual distance an electric vehicle can cover with the energy stored in the battery at given time  $k$ . The RDR prediction problem can be formally formulated by considering the electric vehicle as a nonlinear system represented, in a discrete-time form, by

$$\begin{aligned} \mathbf{x}_k &= \mathbf{f}(\mathbf{x}_{k-1}, \mathbf{u}_k, \mathbf{v}_k, \mathbf{w}_k) \\ \mathbf{y}_k &= \mathbf{h}(\mathbf{x}_k, \mathbf{u}_k, \mathbf{n}_k, \mathbf{w}_k), \end{aligned} \quad (1)$$

where  $\mathbf{x}_k$  is the state vector,  $\mathbf{w}_k$  is the parameter vector,  $\mathbf{v}_k$  is the process noise vector,  $\mathbf{u}_k$  is the input vector,  $\mathbf{y}_k$  is the output vector and  $\mathbf{n}_k$  is the measurement noise vector.  $\mathbf{f}(\cdot)$  and  $\mathbf{h}(\cdot)$  represent the state and output function respectively. The RDR prediction is concerned with forecasting the energy consumption of the electric vehicle along the road ahead and identifying the point at which new recharging is required. This point can be mathematically determined by defining a threshold as follows

$$T(\mathbf{y}_k) = \begin{cases} 1 \\ 0 \end{cases} \quad (2)$$

with  $T = 1$  if new charge is required and  $T = 0$  otherwise. It is important to notice that the threshold is defined as  $T(\mathbf{y}_k)$ . The reason for this is that the battery management system of

most electric vehicles prevent the battery cells to discharge below the cell's cut-off voltage, which marks the point of total charge depletion. Since the terminal voltage is an indicator for total charge depletion and is a measurable quantity, it represents the output  $\mathbf{y}_k$  of the system.

In practice there are many sources of uncertainty that influence the prediction of the RDR, e.g., the lack of knowledge about the system states and parameters, the noise presented in the measurements or the ignorance about the future driving conditions. Given these sources of uncertainty, it would be wrong to consider the RDR as a deterministic quantity. Thus, instead of predicting single RDR values, we compute  $p(\text{RDR}_{k_p} | \mathbf{y}_{0:k_p})$ , i.e., the probability distribution of the RDR. Here  $k_p$  is the time at which the prediction takes place.

Fig. 1 depicts the adopted architecture for predicting the RDR. The RDR prediction proceeds in two steps. In the first step, the state estimation module uses  $\mathbf{u}_k$  and  $\mathbf{y}_k$  to compute the posterior estimate  $p(\mathbf{x}_k | \mathbf{y}_{0:k})$ , i.e., the most up to date approximation of the states of the battery based on measurements acquired up to time  $k$ . In the second step, the RDR prediction module uses, at time  $k_p$ , the current posterior state estimate  $p(\mathbf{x}_{k_p} | \mathbf{y}_{0:k_p})$  together with an hypothesized future driving profile  $\{\mathbf{u}_{k_p}, \mathbf{u}_{k_p+1}, \dots, \mathbf{u}_m\}$  to compute  $p(\text{RDR}_{k_p} | \mathbf{y}_{0:k_p})$ . For the sake of better understanding, a driving profile is characterized by the speed ( $v$ ) and acceleration ( $a$ ) of the vehicle and by the slope ( $\alpha$ ) of the road.

## 3 System Modeling

### 3.1 Modeling Approach

To predict the RDR a detailed model that determines the power demand of the electric vehicle and describes the dynamic behavior of the battery is needed.

This work employs a quasi-static model, for modeling the chassis, the driveline and the electric motor. The quasi-static approach is computationally efficient since it assumes that the vehicle moves exactly with the predicted speed. This assumption is convenient because no differential equations have to be solved and the power requirements can be easily computed by solving algebraic equations.

Nevertheless, the battery cannot be modeled using this approach since, as already mentioned, the terminal voltage determines the threshold of the prediction algorithm. It is then necessary to model the battery in such a way, that the non-linear effects, in the capacity and in the terminal voltage, are taken into account. Fig. 2 shows the implemented model of the electric vehicle. We combine the quasi-static model with a dynamic

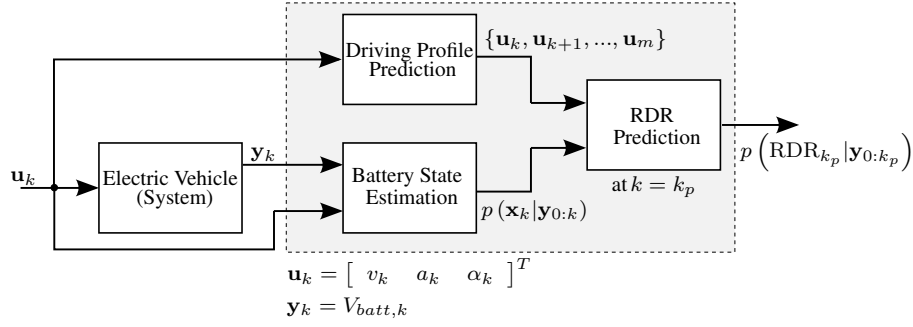


Figure 1: RDR prediction architecture.

model to account for the aforementioned nonlinearities presented in the battery.

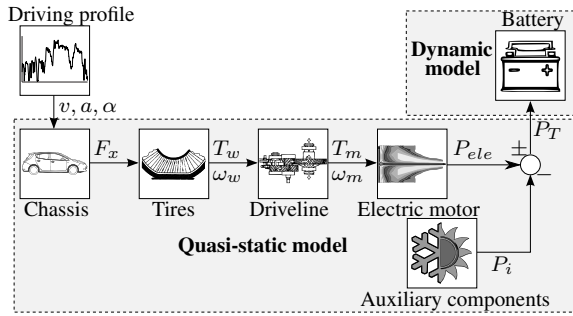


Figure 2: Combined quasi-static/dynamic electric vehicle model.

In the following two sections both parts of the model are explained in detail. For the sake of better understanding, we omit expressing the variables of the quasi-static model as time dependent, since this model is described just by a set of algebraic equations. The differential equations of the dynamic model, instead, are expressed in a discrete-time form, since both, the state estimation and the RDR prediction modules, require a discrete-time representation of the battery model.

### 3.2 Quasi-static Model

An electric vehicle is composed of many components which, for simplification purposes, can be considered to move uniformly. Thus, the electric vehicle can be represented as one lumped mass. As it can be seen in Fig. 3, the force  $F_x$  needed for moving the vehicle forward is given by

$$F_x = F_{air} + F_g + F_r + F_i, \quad (3)$$

where  $F_{air} = \frac{1}{2} \rho_{air} c_w A v^2$  is the aerodynamic drag force,  $F_g = mg \sin(\alpha)$  is the hill climbing force,  $F_r = mg K_r$  is the rolling resistance and  $F_i = ma$  is the force needed to accelerate/decelerate the electric vehicle. Here  $v$  represents the vehicle speed,  $\rho_{air}$  is the air density,  $c_w$  is the aerodynamic drag coefficient,  $A$  and  $m$  are the frontal area and the mass of the vehicle,  $g$  is the gravitational acceleration,  $K_r$  is the rolling

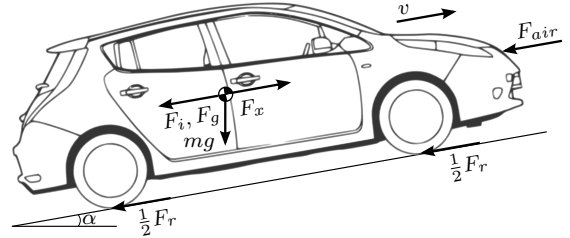


Figure 3: Forces acting during the motion of a vehicle.

resistance coefficient and  $\alpha$  is the slope of the road.

The mechanical power demand  $P_{mec}$  is calculated from the definition of mechanical power  $P_{mec} = F_x v$  as follows

$$P_{mec} = \frac{1}{2} \rho_{air} c_w A v^3 + mg \sin(\alpha) v + mg K_r v + mav. \quad (4)$$

This model accurately calculates the mechanical power demand of a vehicle with a very low computational cost.

To properly employ Eq.(4) in the RDR prediction algorithm, we need to differentiate between input variables and parameters. As already mentioned, in this paper  $a$ ,  $v$  and  $\alpha$  are considered as input variables. All the other terms of Eq.(4), namely,  $\rho_{air}$ ,  $c_w$ ,  $A$ ,  $m$ ,  $g$ , and  $K_r$  are assumed to remain constant, since they rarely change or change slowly during a trip. Accordingly, the input vector for the electric vehicle model is given by

$$\mathbf{u} = [v \ a \ \alpha]^T. \quad (5)$$

The power of the electric motor is calculated from the torque demand  $T_m$  and from the rotational speed  $\omega_m$  at the rotor. As stated above, the dynamic behavior of the driveline components is not modeled. Therefore,  $T_m$  and  $\omega_m$  can be easily determined by

$$T_m = \frac{T_w}{i_d} = \frac{F_x r_{tire}}{i_d}, \quad (6)$$

$$\omega_m = \omega_w i_d = \frac{v i_d}{r_{tire}}, \quad (7)$$

where  $r_{tire}$  and  $i_d$  are the tire's radius and the gear ratio of the driveline respectively.

The relationship between the mechanical power  $P_{mec}$  and the electrical power  $P_{ele}$  can be calculated without a detailed model by using a stationary map of the electric motor's efficiency  $\eta_m$  as a function of  $\omega_m$  and  $T_m$ . In this case the electrical power is calculated by

$$P_{ele} = \frac{P_{mec}}{\eta_m(\omega_m, T_m)}. \quad (8)$$

One main feature of electric vehicles is that a certain amount of the kinetic energy can be recovered by means of regenerative braking. During regenerative braking, the motor works as a generator and delivers power back to the battery. In such a case, the generated power is given by

$$P_{ele} = P_{mec}\eta_g(\omega_m, T_m). \quad (9)$$

It is worth mentioning that the efficiency  $\eta_g$  in generator mode differs from the efficiency  $\eta_m$  in motor mode. The total power  $P_T$ , that is obtained or supplied to the battery, is given by both the electric motor and the auxiliary components, as depicted in Fig. 2. The total power demand is then calculated as

$$P_T = P_{ele} + \sum_{i=1}^n P_i, \quad (10)$$

where  $P_i$  represents the power consumed by both the main and the secondary auxiliary components such as the compressor of the air conditioning system or the lights. For the sake of simplicity the power demand of all auxiliary components is considered to be constant.

### 3.3 Dynamic Battery Model

We employ a model of a Li-ion cell, as shown in Fig. 4. The model combines the Kinetic Battery Model (KiBaM) [5] for capturing the nonlinear effects in the battery capacity, such as the recovery and the rate capacity effect, with a second order equivalent circuit based model which captures the dynamic response of the Li-ion cell. Furthermore, the combined model demands low computational effort, which makes it suitable for real-time applications. Even though the KiBaM was initially developed for lead acid batteries, it has shown to be suitable for modeling the capacity behavior of Li-ion cells [6].

The Kinetic Battery Model abstracts the chemical processes of the battery discharge to its kinetic properties. The model assumes that the total charge of the battery is distributed with a capacity ratio  $0 < c < 1$  between two charge wells. On the one hand, the first well has the available charge and delivers it directly to the load. The second well, on the other hand, can

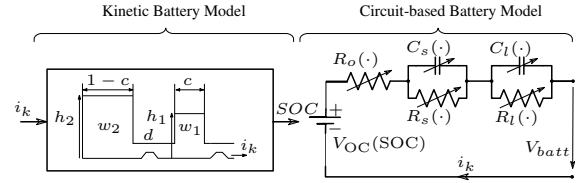


Figure 4: Combined battery model.

supply charge only to the first well by means of the parameter  $d$ . The rate of charge that flows from the second well to the first well depends on both  $d$  and on the height difference between the wells ( $h_2 - h_1$ ). If the first well is empty, then the battery is considered to be fully discharged.

By applying load to the battery, the charge in the first well is reduced, which leads to an increment in the height difference between both wells. After removing the load, certain amount of charge flows from the second well to the first well until the height of both wells is the same. In this way the recovery effect is taken into account by the model. The rate capacity effect is also considered in this model. For high discharge currents, the charge in the first well is delivered faster to the load in comparison to the charge that flows from the second well. In this scenario there is an amount of charge that remains unused. The consideration of this effect is especially important for applications in electric vehicles, since the unused charge can eventually increase the driving range.

The KiBaM yields two differential equations which describe the change of capacity in both wells in dependence of the load  $i_k$ , the conductance  $d$  and the capacity ratio  $c$ :

$$w_{1,k+1} = a_1 w_{1,k} + a_2 w_{2,k} + b_1 i_k, \quad (11)$$

$$w_{2,k+1} = a_3 w_{1,k} + a_4 w_{2,k} + b_2 i_k, \quad (12)$$

where

$$\begin{pmatrix} a_1 & a_2 \\ a_3 & a_4 \end{pmatrix} = e^{\begin{pmatrix} -\frac{k}{c} & \frac{k}{1-c} \\ \frac{k}{c} & -\frac{k}{1-c} \end{pmatrix} \Delta t},$$

$$\begin{pmatrix} b_1 \\ b_2 \end{pmatrix} = \int_0^{\Delta t} e^{\begin{pmatrix} -\frac{k}{c} & \frac{k}{1-c} \\ \frac{k}{c} & -\frac{k}{1-c} \end{pmatrix} \vartheta} d\vartheta \begin{pmatrix} 1 \\ 0 \end{pmatrix}.$$

The term  $\Delta t$  is the sampling time used in the discretization. The battery SOC is then given by

$$\text{SOC}_k = \frac{w_{1,k}}{cC_n 3600}, \quad (13)$$

where  $C_n$  is the nominal capacity of the battery. The right-hand-side equivalent circuit of Fig. 4 is compound of three parts, namely, the open circuit voltage  $V_{OC}$ , a resistance  $R_o$  and two RC networks.

The voltage  $V_{OC}$  changes at different SOC levels, as depicted in Fig. 5. The ohmic resistance  $R_o$  captures the I-R drop, i.e., the instantaneous voltage drop due to a step load current event. The  $R_s C_s$  and  $R_l C_l$  networks capture the voltage drops due to the electrochemical and the concentration polarization, respectively. In Fig. 4 the dependency of these parameters on the temperature and on the SOC is represented by the term  $(\cdot)$ .

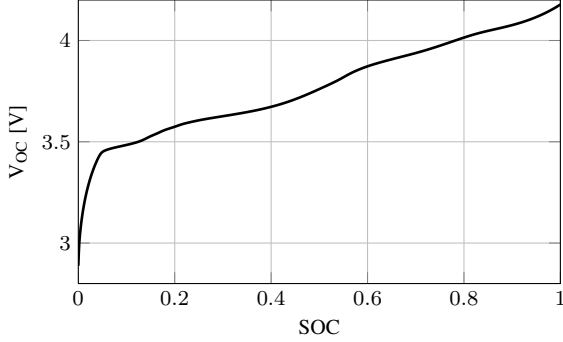


Figure 5:  $V_{OC} - SOC$  relationship.

This part of the model yields two differential equations which describe the transient response of the battery:

$$v_{s,k+1} = e^{-\frac{\Delta t}{R_s C_s}} v_{s,k} + \left( -R_s e^{-\frac{\Delta t}{R_s C_s}} + R_s \right) i_k, \quad (14)$$

$$v_{l,k+1} = e^{-\frac{\Delta t}{R_l C_l}} v_{l,k} + \left( -R_l e^{-\frac{\Delta t}{R_l C_l}} + R_l \right) i_k. \quad (15)$$

Accordingly, the state vector of the battery model is given by

$$\mathbf{x}_k = [ w_{1,k} \ w_{2,k} \ v_{s,k} \ v_{l,k} ]^T. \quad (16)$$

The battery terminal voltage  $V_{batt,k}$  is given by the sum of the open circuit voltage, the voltage drop at  $R_o$  and the transient voltages  $v_{s,k}$  and  $v_{l,k}$  as follows

$$V_{batt,k}(SOC) = V_{OC}(SOC) + R_o i_k + v_{l,k} + v_{s,k}. \quad (17)$$

As presented in the previous section, the quasi-static part of the electric vehicle model computes the total electrical power demand  $P_T$ . Nevertheless, the battery model requires the load current  $i_k$  as the input variable. Therefore, it is necessary to express  $i_k$  in terms of  $P_T$ . The load current  $i_k$  can be obtained from the definition of electrical power  $P = IV$ . Considering  $P = P_T$  and  $V = V_{batt}$  the terminal voltage can be expressed as

$$V_{batt} = \frac{P_T}{i}. \quad (18)$$

By substituting Eq.(18) into Eq.(17) and solving it for  $i$ , the current at time  $k$  is given by

$$i_k = -\frac{C - \sqrt{C^2 - 4P_T(\mathbf{u}_k)R_o}}{2R_o}, \quad (19)$$

where

$$C = (V_{OC}(SOC) + v_{s,k} + v_{l,k}).$$

$P_T(\mathbf{u}_k)$  expresses the dependency of the total electrical power demand on the input vector given by Eq.(5). The solution with the positive part in the square root term of Eq.(19) is neglected, since its consideration would cause some current to be supplied by the battery in the case of no load, i.e., when  $P_T = 0$ , which is physically not meaningful.

## 4 Battery State Estimation

As explained in section 2, the states of the battery have to be estimated before the prediction module computes the RDR. The state estimation establishes a starting point for the prediction step. The task of the estimation module is to compute  $p(\mathbf{x}_k | \mathbf{y}_{0:k})$ , i.e., to represent the most up-to-date knowledge of the system states (shown in Eq.(16)) at given time  $k$  based the current and on all past measurements.

To address this estimation, recursive Bayesian tracking techniques such as particle filter (PF) [7], the extended Kalman filter (EKF) or the unscented Kalman filter (UKF) [8] have emerged as very promising solutions. This work uses a particle filter for estimating the states of the battery. The advantage of the particle filter over other approaches is that it does not require linearizing the system model and no Gaussian distribution of the system states has to be assumed.

The particle filter approximates the posterior probability distribution  $p(\mathbf{x}_k | \mathbf{y}_{0:k})$  by a set of  $N_x$  weighted particles  $\mathbf{S}_k = \{ \mathbf{x}_k^i, w_k^i \}_{i=1}^{N_x}$ . Here  $\mathbf{x}_k^i$  is the set of particles representing the state space and  $w_k^i$  are the associated importance weights. Each particle is sampled from an *a priori* estimation of the state space and it is propagated through the function  $\mathbf{f}(\cdot)$ . The value of each particle is recursively updated from measurements through the output function  $\mathbf{h}(\cdot)$ .

Then, the probability distribution of the state variables at time  $k$  is approximated by

$$p(\mathbf{x}_k | \mathbf{y}_{0:k}) \approx \frac{1}{N_x} \sum_{i=1}^{N_x} w_k^i \delta(\mathbf{x}_k - \mathbf{x}_k^i) \quad (20)$$

where  $\delta(\cdot)$  describes the Dirac delta function located at  $\mathbf{x}_k^i$ .

An important issue with the application of the particle filter is the so called *particle degeneracy*, i.e., all but few particles have negligible

weights. Particle degeneracy leads to a poor approximation of the state variables and, since most weights are close to zero, valuable computational effort is wasted by updating insignificant particles. To overcome this issue the *sequential importance resampling* (SIR) [9] is employed. The idea behind the SIR is to duplicate particles with large weights and to eliminate those with small weights.

## 5 RDR Prediction

The second step applies the particle filter for predicting the RDR at given time  $k_p$ . To this aim the RDR prediction module uses the posterior estimate  $p(\mathbf{x}_{k_p} | \mathbf{y}_{0:k_p})$  as initial condition.

By assuming that the set of particles  $\{\mathbf{x}_{k_p}^i, w_{k_p}^i\}_{i=1}^{N_x}$  accurately represents the unknown states at the time of prediction, it is possible to approximate the probability density function of system states at any time  $k_p + m$  in the future by means of the law of total probabilities [10]

$$\hat{p}(\mathbf{x}_{k_p+m} | \hat{\mathbf{x}}_{k_p:k_p+m-1}) \approx \sum_{i=1}^{n_x} w_{k_p+m-1}^i \hat{p}(\hat{\mathbf{x}}_{k_p+m}^i | \hat{\mathbf{x}}_{k_p+m-1}^i). \quad (21)$$

To account for the fact, that during the prediction the shape of the states probability distribution may change, due to noise and process nonlinearities, Eq.(21) requires the set of weights to be updated at each iteration. However, during the prediction step no new measurements, which could serve for updating the weights, can be acquired. This implies that an update procedure for the particle weights, as it would happen in a typical filtering problem, cannot be carried out. This issue is addressed by assuming the weights as invariant from  $k_p$  to  $k_p + m$ . This assumption is justified by considering the uncertainty added by model inaccuracies or by the ignorance about future driving conditions to be large in comparison to the uncertainty which comes from considering constant particle weights. In this way, the set of weighted particles  $\{\mathbf{x}_{k_p}^i, w_{k_p}^i\}_{i=1}^{N_x}$  is simply propagated forward into the future by simulating the behavior of the electric vehicle as reaction to a future driving profile, until minimum allowable battery terminal voltage is reached.

Once all particles have reached the cut-off battery voltage, i.e.,  $T_{k_p}^i = 1$ , the traveled distance  $\text{RDR}_{k_p}^i$  of each particle is determined and combined with its weight  $w_{k_p}^i$  to approximate

$p(\text{RDR}_{k_p} | \mathbf{y}_{0:k_p})$  as follows

$$p(\text{RDR}_{k_p} | \mathbf{y}_{0:k_p}) \approx \sum_{i=1}^{N_x} w_{k_p}^i \text{RDR}_{k_p}^i. \quad (22)$$

As already mentioned, Eq.(22) requires an hypothesized driving profile, which serves as the reference for the propagation of particles. We employ a stochastic approach to predict the driving profile in such a way, that real-world driving patterns are captured. The following section introduces the approach for predicting driving profiles.

### 5.1 Driving Profile Prediction

It has been shown that driving profiles can be modeled as a discrete-time Markov chain [11]. For predicting the entire driving profile two Markov chains are used. First, future values of speed and acceleration are generated by a 2D chain. Second, the slope profile is predicted by means of a 1D Markov chain independent of the speed and the acceleration. To apply a Markov chain the input space is quantized for the speed/acceleration pair and for the slope in such a way that each input variable takes a finite number of values. The input space is then given by  $\{\mathbf{u}_1^{va}, \mathbf{u}_2^{va}, \dots, \mathbf{u}_m^{va}\}$  and by  $\{u_1^\alpha, u_2^\alpha, \dots, u_m^\alpha\}$ , where  $\mathbf{u}^{va} = \{v, a\}$  and  $u^\alpha = \alpha$  represent parts of the input vector given by the speed/acceleration pair and by the slope respectively, with  $m$  as the horizon length of the predicted profiles.

For the sake of brevity, here we just explain the generation of the speed/acceleration profiles. The prediction of the slope profile proceeds in a similar fashion.

The Markov chain assumes that the transition probability from a state  $\mathbf{u}_k^{va}$  to a state  $\mathbf{u}_{k+1}^{va}$ , where  $k$  is a discrete time instant ( $k = t_0, t_1, t_2, \dots$ ), only depends on the current state and not on the sequence of states that precede it. The transition probabilities between all possible states are grouped in a transition probability matrix (TPM)  $\mathbf{P} \in \mathbb{R}^{n \times l}$  such that

$$p_{ij} = \mathbf{P}(\mathbf{u}_{k+1}^{va} = j | \mathbf{u}_k^{va} = i), \quad (23)$$

where  $p_{ij}$  is the  $ij^{\text{th}}$  element of  $\mathbf{P}$ . In this paper the transition probabilities are estimated from historical driving data and from standard driving cycles. The maximum likelihood estimation method [12] is applied for estimating the TPM. The transition probability  $p_{ij}$  is computed by

$$p_{ij} = \frac{n_{ij}}{\sum_{j=1}^s n_{ij}} = \frac{n_{ij}}{n_i}, \quad (24)$$

where  $n_{ij}$  is the number of times a transition from  $\mathbf{u}_i^{va}$  to  $\mathbf{u}_j^{va}$  has occurred, and  $n_i$  is the

total number of occurrences of  $\mathbf{u}_i^{va}$ . Fig. 6 depicts an example of the TPM used for the speed/acceleration profile prediction. The matrix is composed of  $n$  columns and  $l$  rows, which represent the finite states of speed and acceleration respectively.

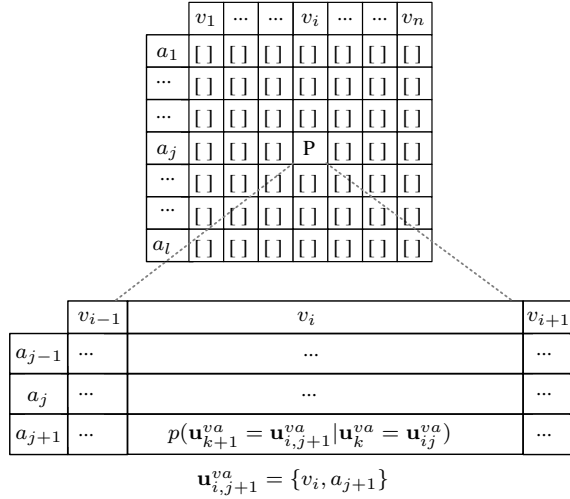


Figure 6: Transition probability matrix for  $\{v_i, a_j\}$  at time  $k$ .

The chain is initialized at  $k = 0$  with the pair  $\{v_k = 0, a_k = 0\}$ . Then, we randomly draw for the next state  $\{v_{k+1}, a_{k+1}\}$  according to the probability distribution represented by the discrete transition probabilities located at  $\{v_k, a_k\}$ . Once the next state  $\{v_{k+1}, a_{k+1}\}$  has been determined, the process is repeated until the desired size of the chain, i.e., the desired length of the profile is reached.

## 5.2 Characterization of the RDR

Until now the RDR prediction, as formulated in Eq.(22), requires propagating the set of particles through a single predicted driving profile. However, such a propagation accounts just for the uncertainty introduced in the state estimation step but it does not consider the uncertainty related to the predicted driving profile. Taking this uncertainty into account would require propagating the set of particles through multiple predicted driving profiles, and not through a single one. The computational complexity of such a prediction becomes a function of  $N_x \times N_u$  [13], where  $N_x$  is the number of weighted particles used in the state estimation and  $N_u$  is the number of predicted driving profiles. The set of weighted particles is then propagated through multiple driving profiles until all particles, along all predicted profiles, have reached the cut-off voltage, i.e.,  $T_{k_p}^{ij} = 1$ . Here  $j$  represents each generated driving profile. The probability distribution  $p(\text{RDR}_{k_p} | \mathbf{y}_{0:k_p})$  is

then approximated by

$$p(\text{RDR}_{k_p} | \mathbf{y}_{0:k_p}) \approx \frac{1}{N_u} \sum_{j=1}^{N_u} \sum_{i=1}^{N_x} w_{k_p}^i \text{RDR}_{k_p}^{ij}. \quad (25)$$

It must be noted that all predicted profiles are equally weighted by means of  $\frac{1}{N_u}$ .

## 6 Results and Discussions

This section presents the results obtained from predicting the RDR for different driving scenarios. The  $\alpha - \lambda$  and the relative accuracy (RA) metrics [14] are applied for evaluating the performance of the approach.

### 6.1 Simulation Results

To validate the performance of the proposed approach, a number of simulations describing different driving situations is performed. The RDR prediction is tested by letting the electric vehicle follow repeatedly the standard drive cycles shown in Fig. 7, namely, the UDSS and the ARTEMIS rural and motorway drive cycles [15]. These drive cycles are representative of typical driving patterns shown in the city, in rural areas and on the highway.

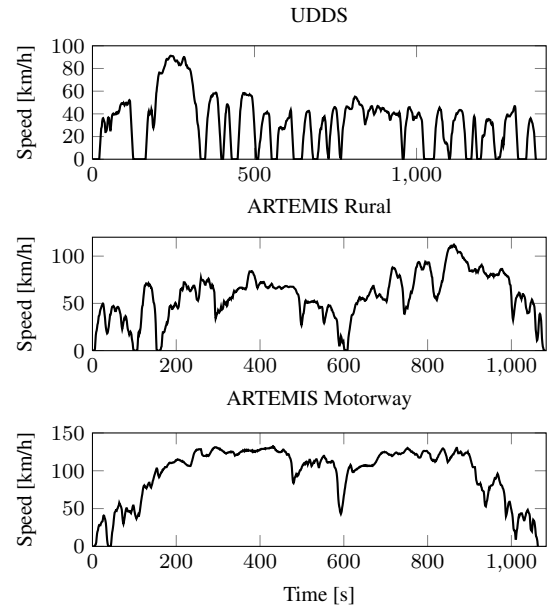


Figure 7: Standard drive cycles used in the simulation.

The parameters of the quasi-static model are chosen according to the parameters of the Nissan Leaf, as shown in Table 1. In this case the parameter are obtained from manufacturer's data sheets. The battery model is tuned using experimental data of a 2.15 Ah Li-ion cell. The battery

pack of the Nissan Leaf is emulated by scaling up the battery capacity to 24 kWh and the voltage to 403.2 V. Table 2 presents the model parameters of the identified Li-ion cell.

Table 1: Parameters of the quasi-static part of the electric vehicle model.

Parameter	Value
$A$	2.29 m <sup>2</sup>
$c_w$	0.28
$m$	1520 kg
$K_r$	0.7
$T_{m,\max}$	280 Nm
$P_{ele,\max}$	80 kW
$r_{tire}$	0.3 m

Table 2: Model parameters of the Li-ion cell.

Parameter	Value
$C_n$	2.15 Ah
$V_{nom}$	4.2 V
$V_{lim}$	2.8 V
$d$	$1.4 \times 10^{-5}$
$c$	0.96

At each experiment, the electric vehicle follows the speed profile imposed by the driving cycle. For the sake of analysis, at the beginning of each simulation the battery is assumed to be fully charged. In this way the maximum driving range that the electric vehicle can reach under different driving situations is compared. During the simulation, the battery states are estimated by the particle filter at each iteration step and then this estimate is used for predicting the RDR. To reduce the computational burden of the simulation, a RDR prediction is performed every 1000 seconds if the battery SOC is larger than 30%. After this point the RDR is predicted every 500 seconds.

### 6.1.1 RDR Prediction Performance

In this section, we evaluate the prediction performance for the different scenarios. For a given prediction time  $k_p$ , metrics of accuracy and spread are computed. Since the estimate  $p(\text{RDR}_{k_p} | \mathbf{y}_{0:k_p})$  is usually non Gaussian, we rely on the median for estimating the RDR and on quantiles as the measure of spread [16]. The RA metric for evaluating the accuracy of the prediction is computed by

$$\text{RA}_{k_p} = 100 \left( 1 - \frac{|\text{RDR}_{k_p}^* - \text{RDR}_{k_p}|}{\text{RDR}_{k_p}^*} \right), \quad (26)$$

where  $\text{RDR}_{k_p}^*$  represents the ground truth RDR, at time  $k_p$ , obtained later after the simulation finishes and  $\text{RDR}_{k_p}$  is the median of the predicted RDR at that time.

The approach has shown similar performance for each driving scenario. Table 3 summarizes the relative accuracy calculated for the prediction at different prediction times. Fig. 8 shows the predictions for the UDDS drive cycle. As it can be seen, for most part of the time, the predicted RDR lies within the boundaries given by the  $\alpha - \lambda$  metric ( $\alpha = 0.15$ ). Also the quantiles  $Q_5$  and  $Q_{95}$  lie within the boundaries for most of the time. It can be observed that, even though the accuracy of the prediction at  $k_p = 1$  is high, the uncertainty in the prediction is large. This is caused by the uncertainty related to the state estimation. At the beginning of the simulation the particle filter does not accurately track the battery states. A certain time is needed for the filter to converge. Therefore, the ignorance about the initial state of the battery causes the RDR prediction to be highly uncertain. Fig. 9 depicts the simulation results for the ARTEMIS rural drive cycle. The results are similar to those obtained with the UDDS. It is important to notice that fewer predictions have been carried out. This is because the average speed of the driving profile is higher and therefore the time each particle needs to drive until  $T_{k_p}^{ij} = 1$  is lower. Fig. 10 shows the prediction carried out with the ARTEMIS motorway driving profile. This case shows the best performance of all driving scenarios. The high accuracy in the prediction is due to the few drastic changes presented in the drive cycle. In this case a transition probability matrix of the highway for predicting the driving profiles is used and therefore the generated profiles have very similar characteristics to the chosen drive cycle.

Table 3: RDR prediction performance.

$k_p$	$RA_{\text{UDDS}}$	$RA_{\text{RURAL}}$	$RA_{\text{MOTORWAY}}$
1	99.32	94.50	99.66
2	97.65	93.55	95.18
4	92.82	97.60	99.60
6	91.73	97.39	99.79
8	92.54	96.09	98.61
10	90.92	77.27	94.89
12	91.25	--	--
14	88.75	--	--
16	88.19	--	--
18	91.71	--	--

## 7 Conclusions and Future Work

In this paper, a model-based approach for predicting the remaining driving range in electric vehicles is implemented. The proposed approach takes into account different sources of uncer-



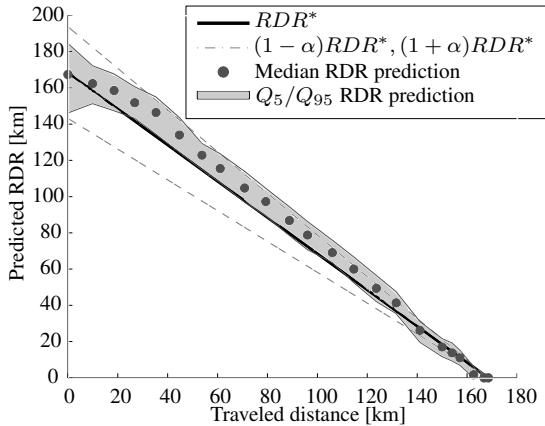


Figure 8: RDR prediction results for the city drive cycle UDDS.

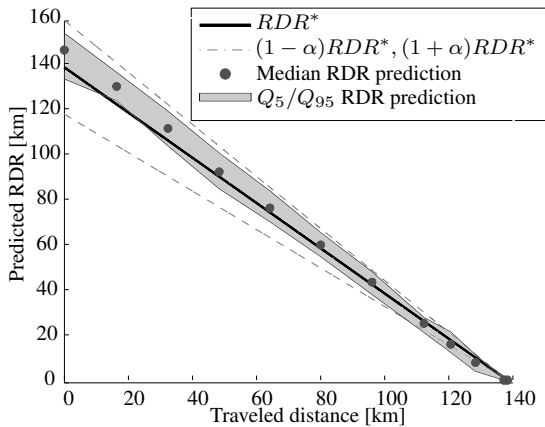


Figure 9: RDR prediction results for the ARTEMIS rural drive cycle.

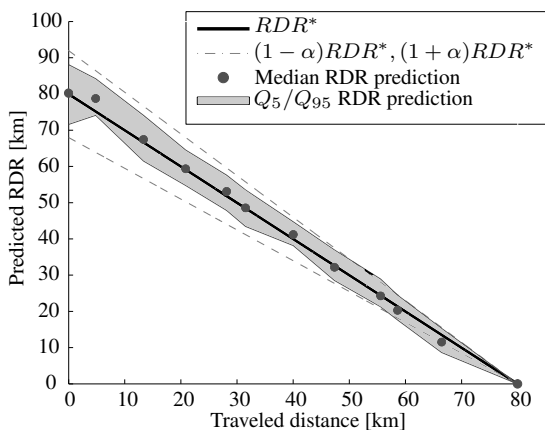


Figure 10: RDR prediction results for the ARTEMIS motorway drive cycle.

tainty such as the one related to variability of the driving profile. Also the inherent uncertainty caused by measurements noise and by the estimation of the battery states is considered in the prediction.

The state estimation is carried out by a particle filter. Then, a set of weighted particles, which approximate the posterior estimate of the battery states, is propagated forward in time through predicted driving profiles until all particles reach the minimum allowable battery terminal voltage. The driving profiles are modeled as stochastic processes and they are predicted with the help of Markov chains. The RDR is then computed as a probability density function approximated by the distribution of the propagated particles.

For demonstration and validation purposes, the approach is tested in the simulation. Three representative standard drive cycles have been used as reference. The simulations have shown that the proposed approach predicts the RDR for all driving scenarios with an acceptable accuracy and computational effort.

In the future, we aim to investigate new methods for speeding up the prediction time so that a real time application can be integrated in our test vehicle. It is also of high importance to investigate methods for describing the driving profile in a parametric form. In this way probability distributions for the parameters can be identified, with the help of experimental data, so that analytical methods for uncertainty propagation can be employed. The use of analytical methods would further increase the efficiency of the prediction algorithm, on the one hand, and would produce repeatable calculations, on the other hand.

## Acknowledgments

The funding for this work was provided by the federal state of North Rhine-Westphalia (NRW) in frame of the project "Technology and test platform for a competence center for interoperable electromobility, infrastructure and networks" (TIE-IN).

## References

- [1] H. Yu, F. Tseng, and R. McGee, *Driving pattern identification for EV range estimation.*, Electric Vehicle Conference (IEVC), IEEE International, (2012).
- [2] Y. Zhang, W. Wang, Y. Kobayashi and K. Shirai, *Remaining driving range estimation of electric vehicle*, Electric Vehicle Conference (IEVC), IEEE International, (2012).
- [3] P. Conradi and S. Hanssen, *Dynamic cruising range prediction for electric vehicles*, in Advanced Microsystems for Automotive Applications, Springer-Verlag Berlin Heidelberg, (2011), pp. 269-276.
- [4] M. Daigle and K. Goebel, *A model-based prognostics approach applied to pneumatic valves*,

International Journal of Prognostics and Health Management, ISSN 2153-2648, (2011).

- [5] J. F. Manwell and J. G. McGowan, *Extension of the Kinetic Battery Model for wind-hybrid power systems*, Proceedings of EWEC, (1994).
- [6] M. R. Jongerden and B. R. Haverkort, *Which battery model to use?*, Software, IET, (2009), vol. 3, no. 6, pp. 445-457.
- [7] G. G. Rigatos, *Particle filtering for state estimation in nonlinear industrial systems*, Instrumentation and Measurement, IEEE Transactions on, (2009), vol. 58, no. 11, pp. 3885-3900.
- [8] S. J. Julier and J. K. Uhlmann, *Unscented filtering and nonlinear estimation*, Proceedings of the IEEE, (2004), vol. 92, no. 3 pp. 401-422.
- [9] R. Douc and O. Cappé, *Comparison of resampling schemes for particle filtering*, Image and Signal Processing and Analysis, 2005. ISPA 2005. Proceedings of the 4th International Symposium, (2005).
- [10] M. E. Orchard and G. J. Vachtsevanos, *A particle-filtering approach for on-line fault diagnosis and failure prognosis*, Transactions of the Institute of Measurement and Control, (1983), vol. 31, pp. 221-246.
- [11] T. K. Lee and Z. S. Filipi, *Representative real-world driving cycles in Midwestern US*, Les Rencontres Scientifiques d'IFP Energies nouvelles - RHEVE, (2011).
- [12] T. C. Lee, G. C. Judge and A. Zellner, *Estimating the parameters of the Markov probability model from aggregate time series data*, North-Holland, 2nd edition, (1970).
- [13] M. Daigle, A. Saxena and K. Goebel, *An efficient deterministic approach to model-based prediction uncertainty estimation*, International Journal of Prognostics and Health Management, (2012).
- [14] A. Saxena et.al., *On applying the prognostics performance metrics*, Annual conference of the prognostics and health management society, (2009).
- [15] Michel Andre, *The ARTEMIS European driving cycles for measuring car pollutant emissions*, Science of the Total Environment, (2004), vol. 334-334, pp. 73-84.
- [16] D. C. Hoaglin, F. Moesteller and J. C. Tukey, *Understanding robust and exploratory data analysis*, Wiley, (1983).

## Authors



**Javier A. Oliva** graduated in 2010 from the Technische Universität Dortmund with the Master's degree in Automation and Robotics. He is currently working as researcher at the Institute of Control Theory and Systems Engineering from the TU Dortmund in the area of driver assistance systems for electric vehicles.



**Christoph Weihrauch** graduated in 2012 from the Ruhr-Universität Bochum with the Master's degree in Electrical Engineering. He is currently working as researcher at the Institute of Control Theory and Systems Engineering from the TU Dortmund in the area of charging systems for electric vehicles.



**Torsten Bertram** is Professor at the Technische Universität Dortmund and is the person in charge of Institute of Control Theory and Systems Engineering. He is active in the research areas vehicle and drive systems, service robotics and development methodology.