Real-Time Prediction of Battery Power Requirements for Electric Vehicles

Eugene Kim, Jinkyu Lee, and Kang G. Shin

Department of Electrical Engineering and Computer Science
The University of Michigan, Ann Arbor, MI 48109-2121
{kimsun, jinkyul, kgshin}@umich.edu

ABSTRACT

A battery management system (BMS) is responsible for protecting the battery from damage, predicting battery life, and maintaining the battery in an operational condition.

In this paper, we propose an efficient way of predicting the power requirements of electric vehicles (EVs) based on a history of their power consumption, speed, and acceleration, as well as the road information from a pre-downloaded map. The predicted power requirement is then used by the BMS to prevent the damage of battery cells that might result from high discharge rates. This prediction also helps BMS efficiently schedule and allocate battery cells in real time to meet an EV's power demands. For accurate prediction of power requirements, we need an accurate model for the power requirement of each given application. We generate this model *in real time* by collecting and using historical data of power consumption, speed, acceleration, and road information such as slope and speed limit. By using this information and the operator's driving pattern, the model extracts the vehicle's history of speed and acceleration, which, in turn, enables the prediction of the vehicle's (immediate) future power requirements. That is, the power requirement prediction is achieved by combining a real-time power requirement model and the estimation of the vehicle's acceleration and speed. The proposed approach predicts closer to the actual required power than a widely-used heuristic approach that uses measured power demand, by up to 69.2%.

Categories and Subject Descriptors

C.3 [**Special-Purpose and Application-Based Systems**]: Real-time and embedded systems; D.4.1 [**Operating Systems**]: Process management—*Scheduling*; I.6.3 [**Simulation and Modeling**]: Applications

ICCPS'13, April 08–11, 2013, Philadelphia, PA, USA.
Copyright 2013 ACM 978-1-4503-1996-6/13/04 ...$15.00

General Terms

Design, Human factors, Management, Measurement, Performance, Reliability

Keywords

Prediction of battery power requirement, acceleration prediction, battery management system (BMS), electric vehicles (EVs)

1. INTRODUCTION

Electric vehicles (EVs) powered by batteries are the key to reducing global warming and rising fuel cost. A recent survey [1] reported that EVs must be a good fit for most Americans' driving and commute patterns, and 40% of the respondents indicated strong interests in purchasing EVs. Despite the increasing popularity of EVs, they have not yet dominated the internal combustion vehicle market for three reasons. First, EVs are expensive due mainly to the large number of battery cells required to power them; for example, 6,800 lithium-ion battery cells are required for Tesla Motors' EVs and 288 cells for GM 2010 Chevy Volt. Second, batteries pose a risk of explosion under extreme conditions, such as high temperature, high charging and discharging rates. Finally, the infrastructure for EVs is not yet ready; while more than 120,000 (90,000) gas stations in the US (China) have been deployed, there are only 8,000 (168) battery charging stations for EVs [2, 3].

An effective battery management system (BMS) is a must in addressing the above-mentioned challenges of EVs; efficient charging/discharging can reduce the required number of battery cells and the charging frequency (hence battery cost); robust cell protection/balancing prevents battery explosion (hence battery protection) [4, 5]. To do this efficiently, processing units in the BMSes need accurate and appropriate physical state information around/in the battery cells. While existing approaches assume that the physical states, such as the cell discharging rate (i.e., the required EV load), are constant or change slowly with time [5, 6], this assumption is not realistic as EVs' power requirements usually change abruptly and significantly. That is, simple measurement of past power consumption cannot accurately predict power demands in future for efficient battery management. Furthermore, the calculation of power requirement

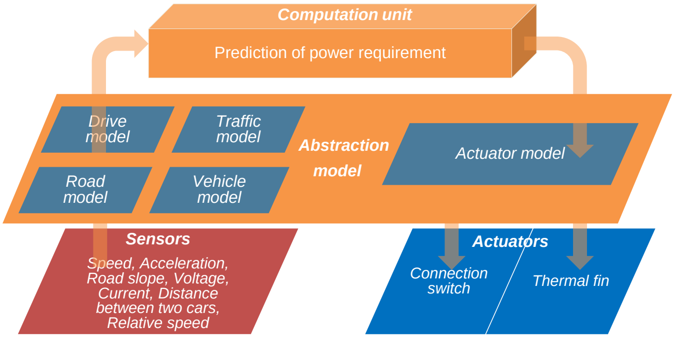


Figure 1: Cyber-Physical System for PRP

should be done fast and used to (re)configure the connection of battery cells; otherwise, the calculated power requirement may become obsolete at the time of its use for battery management. It is therefore necessary for the BMS to predict the power requirement accurately and in real time.

The main goal of this paper is to develop a systematic way of accurately predicting an EV's power requirement in *real time*, which will then enable the BMS to keep cells within their operating limits and schedule battery cells to be discharged and recharged more effectively while balancing them. To show the utility of our power requirement prediction, we also propose an advanced BMS equipped with the power requirement predictor (PRP). Note, however, that this prediction is difficult since the power requirement depends on many *physical elements* such as traffic conditions, traffic regulations, the operator's driving pattern, the vehicle's state and the road condition. Therefore, *cyber elements* in PRP that compute the power requirements should capture their coupling with the physical elements. To achieve this, we add an interface that abstracts the physical elements as shown in Fig. 1; the input and output of the interface are directly connected to physical components like sensors and actuators.

To facilitate the prediction of power requirements, we introduce a power requirement model with parameters that dictate the required power, and predict each parameter from the corresponding *physical* model in the abstraction layer. Although many power requirement models have been proposed, such as [7, 8, 9], we use the simple and widely-used model with real-time adaptation (i.e., history-based energy-usage model in [10]) to different vehicles and operational environments. With the model, our PRP operates as follows. First, we update the parameters in the power requirement model according to the performance of the vehicle's physical components and structure by periodically estimating the parameters based on the driving pattern and the power demand data. Second, we pre-calculate changes in the parameters related to the road information using a pre-downloaded map. Finally, to predict the changes of acceleration, the most challenging step, we make three key observations that affect the acceleration pattern: the traffic flow, the operator's driving behavior, and the traffic regulations. Then, we develop acceleration prediction methods that address these observations, and propose a unified method that exploits the prediction methods.

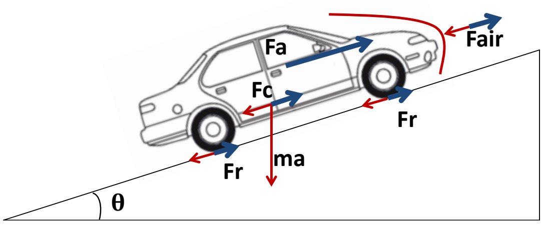


Figure 2: Forces in the power requirement model

Our evaluation with actual experimentation and realistic simulation demonstrates that the proposed PRP with the unified acceleration prediction method yields the actual required power more accurately than a widely-used heuristic approach by up to 69.2%.

The proposed PRP differs from other approaches in that it

- predicts the operator's driving pattern in real time;
- estimates parameters adaptively for the prediction of power requirements;
- reduces the complexity of computation; and
- reflects traffic flows, the driver's behavior, and traffic regulations.

The paper is organized as follows. Section 2 presents our PRP, and Section 3 details the acceleration prediction schemes developed for the PRP. Section 4 evaluates our acceleration prediction using experimentation and simulation. Finally, the paper concludes with Section 5.

2. POWER REQUIREMENT PREDICTION

We first present an existing power requirement model we adopt for the power requirement predictor (PRP), and then describe how to predict all the parameters in the model. To this end, we identify the parameters' characteristics, and then use them to design the PRP. Finally, we present how the PRP is placed in an advanced BMS.

2.1 Power requirement model

We use a polynomial power requirement model [11] that accounts for power demand according to physical laws. Fig. 2 illustrates the model where F_a, F_{air}, F_c, and F_r denote, respectively, the force for acceleration, the aerodynamic force, a load related to steepness, and rolling resistance. The total power requirement (P_{total}) is then calculated as the sum of all forces multiplied by the vehicle's forward speed (V):

$$P_{total} = F_a \cdot V + F_{air} \cdot V + F_c \cdot V + F_r \cdot V$$

$$= m \cdot a \cdot V + \frac{1}{2} \cdot \rho_{air} \cdot C_d \cdot A \cdot V^3 + m \cdot g \cdot \sin\theta \cdot V + m \cdot g \cdot K_r \cdot V, \quad (1)$$

	a (m/s^2)	V (m/s)	m (kg)	g (m/s^2)	K_r	α (degree)	ρ_{air} (kg/m^2)	C_d	A (m^2)
Dynamics	Extremely high	High	Extremely low	Extremely low	High	High	Low	Extremely low	Extremely low
Dependency	Driver, Road, Traffic	Driver, Road, Traffic	Vehicle structure	Altitude	Tire, Road	Road	Altitude	Vehicle structure	Vehicle structure

Table 1: Characteristics of parameters in the power requirement model

where θ denotes the slope angle, and all other parameters will be detailed in Section 2.2.

Simplicity is the primary reason for adopting this model. Since the model depends only on some physical parameters, we can obtain their actual values from measurement or calculation. Moreover, the calculation of the power requirement in Eq. (1) requires only basic operations, facilitating the real-time prediction of power requirements. In Section 4, we will show that the model, albeit simple, yields good accuracy in estimating the required power.

2.2 Characteristics of the model parameters

To estimate the amount of power that the BMS should provide during the next time interval, we need to predict all the parameters in Eq. (1). Such prediction requires the knowledge of (i) the frequency of change in each parameter and (ii) the information necessary to predict each parameter. One can then tailor prediction schemes to this knowledge. Table 1 summarizes the parameter characteristics which are then used to categorize the parameters as

C1. *stable*: mass of the vehicle (m), gravitational acceleration (g), air density (ρ_{air}), drag coefficients (C_d) and frontal area (A);

C2. *dynamic but easy to predict*: rolling resistance coefficient (K_r) and road slope (θ); and

C3. *dynamic and difficult to predict*: acceleration (a) and speed (V).

The parameters in C1, such as m, C_d and A, rarely change as they depend only on the vehicle itself. Also, g and ρ_{air} change very slowly. The parameters in C2 depend on the car's location on the road. We can thus accurately predict the parameters by using offline information of the roads (e.g., Google Map [12]) and real-time location information (e.g., GPS). For example, we can get the information of road type and slope *a priori* from a pre-downloaded map, thus acquiring K_r and θ for a given location. By incorporating the static information into real-time location information, we can easily predict the parameters in C2 despite their dynamic variation.

In contrast, the two parameters a and V in C3 dynamically change and are difficult to predict, since they depend not only on road information, such as the road type and slope, but also on the operator's driving pattern, which are difficult to predict and analyze. In particular, drivers have their own driving patterns related to complex human decision-making processes, and the pattern might change with their mood or physical condition. Besides, the driver should obey traffic regulations and consider traffic conditions, which significantly affect the two parameters.

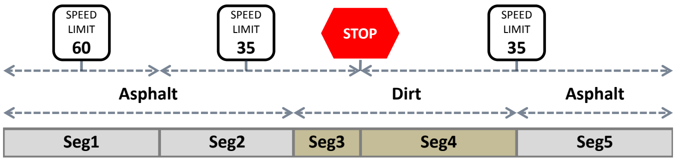


Figure 4: Example of road segmentation

As a result, we need two different approaches in predicting the parameters in C1, C2, and C3 as detailed in the next subsection.

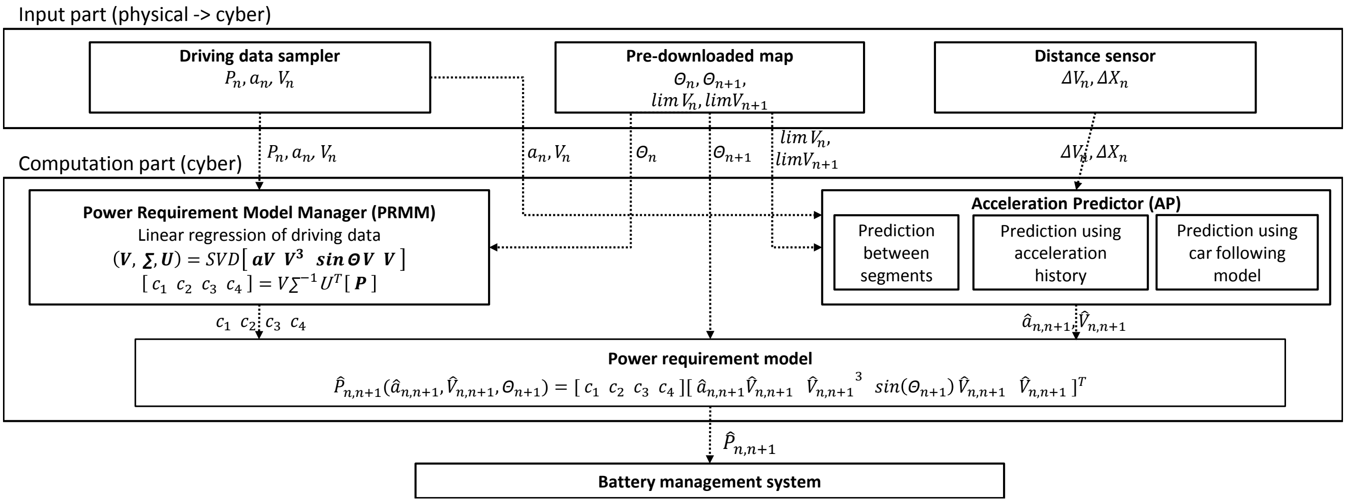
2.3 Design of the power requirement predictor

Here we present the design of PRP. As shown in Fig. 3, PRP estimates the next (step $n+1$) power requirement during the current time interval (step n), and consists of input and computation parts. The input part provides all measured data of the current step and all pre-calculated data for the next step. Using this data, the computation part estimates all the parameters in C1–C3, and then outputs the estimated power requirement for the next step (i.e., $\hat{P}_{n,n+1}$) using the power requirement model in Eq. (1).

Before detailing these two parts, we first define a *road segment*, which is used for the computation part. A road segment is a continuous road interval in which road conditions and traffic regulations are uniform. Fig. 4 shows an example. The concept of road segment is important, since some parameters in the power requirement model vary with road segment.

The computation part estimates the parameters in two different ways. As discussed in Section 2.2, the parameters in C1 and C2 are either slowly changing or predictable, but the parameters in C3 are difficult to estimate. Considering the different parameter characteristics, the power requirement model manager (PRMM) and the acceleration predictor (AP) predict the parameters in C1 and C2, and those of C3, respectively, as detailed next.

The parameters in C1 change slowly, and K_r in C2 is constant within a road segment. So, after segmentation of the


Input part (physical -> cyber)
Driving data sampler
P_n, a_n, V_n
Pre-downloaded map
Θ_n, Θ_{n+1}, $lim V_n, lim V_{n+1}$
Distance sensor
$\Delta V_n, \Delta X_n$
Computation part (cyber)
P_n, a_n, V_n
a_n, V_n
Θ_n
Θ_{n+1}
$lim V_n$, $lim V_{n+1}$
$\Delta V_n, \Delta X_n$
Power Requirement Model Manager (PRMM)
Linear regression of driving data
$(\boldsymbol{V}, \boldsymbol{\Sigma}, \boldsymbol{U}) = SVD[\, \boldsymbol{aV} \;\; \boldsymbol{V^3} \;\; \boldsymbol{sin\Theta V} \;\; \boldsymbol{V}\,]$
$[\, c_1 \;\; c_2 \;\; c_3 \;\; c_4 \,] = V\Sigma^{-1}U^T[\, \boldsymbol{P} \,]$
Acceleration Predictor (AP)
Prediction between segments
Prediction using acceleration history
Prediction using car following model
$c_1 \;\; c_2 \;\; c_3 \;\; c_4$
$\hat{a}_{n,n+1}, \hat{V}_{n,n+1}$
Power requirement model
$\hat{P}_{n,n+1}(\hat{a}_{n,n+1}, \hat{V}_{n,n+1}, \Theta_{n+1}) = [\, c_1 \;\; c_2 \;\; c_3 \;\; c_4 \,][\, \hat{a}_{n,n+1}\hat{V}_{n,n+1} \;\; \hat{V}_{n,n+1}{}^3 \;\; sin(\Theta_{n+1})\,\hat{V}_{n,n+1} \;\; \hat{V}_{n,n+1} \,]^T$
$\hat{P}_{n,n+1}$
Battery management system

<Notations> P_n, a_n, V_n: power demand, acceleration, speed *measured* at step n

$\hat{P}_{n,n+1}$, $\hat{a}_{n,n+1}$, $\hat{V}_{n,n+1}$: power demand, acceleration, speed at step $n+1$ *estimated* at step n

$\theta_n, limV_n$: road slope, speed limit at step n obtained from a pre-downloaded map

ΔV_n, ΔX_n: relative speed and distance between the leading and my cars *measured* at step n

Figure 3: Overall structure of the power requirement predictor (PRP)

road, we can abstract K_r and the parameters in C1 into four coefficients, thus reducing the number of input parameters for the power requirement model as follows.

$$P_{total}(a, V, m, g, K_r, \theta, \rho, C_d, A) = \text{the RHS of Eq. (1)}$$

$$\Rightarrow P_{total}(a, V, \theta) = c_1 \cdot a \cdot V + c_2 \cdot V^3 + c_3 \cdot sin\theta \cdot V + c_4 \cdot V. \quad (2)$$

Then, Eq. (2) depends only on the coefficients c_1—c_4, acceleration (a), speed (V) and road slope (θ). Since θ is given by the pre-downloaded map, we only need to estimate the coefficients, and acceleration and speed. For coefficients, PRMM adopts a popular history-based estimation—linear regression, calculating the coefficients based on the past data of all parameters in Eq. (2). For this, PRMM receives the current data of power requirement (P_n), acceleration (a_n), speed (V_n), and rode slope (θ_n) as shown in Fig. 3, and estimates the four coefficients for prediction at the next interval.

On the other hand, AP outputs acceleration and speed for the next interval, and needs additional information on the relative speed and distance between the leader and the follower cars (ΔV_n and ΔX_n) as shown in Fig. 3, which abstract the traffic flow. Of the two parameters a and V, we will focus on the prediction of a since we can calculate V in the next interval once we know a and V in the current interval. To predict a in the next interval, we identify factors that affect the acceleration pattern, develop schemes to address the factors, and finally combine the schemes towards a unified, effective prediction. We will detail the acceleration prediction in Section 3.

As shown in Fig. 3, the input part in PRP provides measured or pre-calculated data to PRMM, AP, or the power requirement model directly, and consists of three components: (i) the driving data sampler for the power consumption (P_n), acceleration (a_n), and speed (V_n); (ii) the pre-downloaded map for the current and next road slopes (θ_n and θ_{n+1}), and the current and next speed limits ($limV_n$ and $limV_{n+1}$); and (iii) the distance sensor for the relative speed and distance between the leader and the follower cars (ΔV_n and ΔX_n).

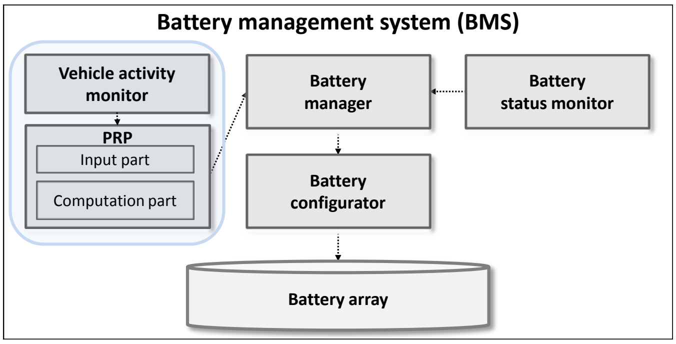


Figure 5: Structure of proposed BMS.

2.4 Advanced BMS

To manage large-scale batteries for EVs in the presence of nonlinear, complex battery dynamics [13], existing BMSes [4, 5, 6, 14, 15, 16] usually consist of the battery status monitor, the battery manager, and the battery configurator as shown in Fig. 5. The battery status monitor periodically reports each cell/pack's state, such as SoC (State of Charge), temperature, and open circuit voltage. Then, the battery manager calculates each cell's discharge rate and rest period, and then sends the results to the battery configurator. The configurator sets the connection of battery cells as instructed by the battery manager at the circuit level.

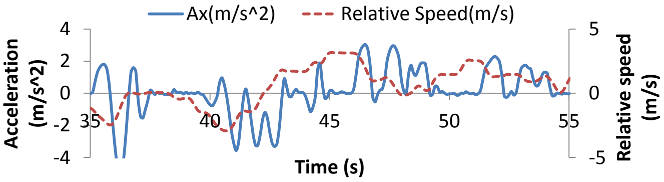


Figure 6: Acceleration of a car and relative speed between the car and the leader car; we chose a car and the car in front on the US-101 highway.

One of the key features of BMSes is to schedule and allocate battery cells to meet the EV's power demand in an effective and safe way. To do this, we have (in the previous subsection) developed the PRP, which forms additional parts of a BMS as shown in Fig. 5. That is, the advanced BMS contains the vehicle activity monitor, which reports the vehicle's driving information, including all the variables provided by the input part. Then, the PRP in Fig. 3 calculates the vehicle's next power demand, which is then sent to the battery manager for more efficient and safe battery (connection) arrangement. We will briefly discuss how the proposed PRP in this paper is placed with the advanced BMS in Section 5.

3. ACCELERATION PREDICTOR

As discussed in Section 2, it is difficult to predict a vehicle's acceleration since it depends not only on static information such as the road type and slope as well as traffic regulations, but also on dynamic one such as the operator's driving pattern and the traffic flow near the car. For accurate acceleration prediction, we first make three key observations that affect the acceleration pattern. Then, we develop acceleration prediction schemes that account for some of these observations. Finally, we combine these prediction schemes to devise a unified acceleration prediction method.

3.1 Observations of acceleration pattern

We make three key observations that determine the acceleration pattern: the traffic flow around the car, the operator's driving behavior, and the traffic regulations. We also corroborate the observations with experiments. The observations will be a basis for acceleration prediction schemes to be developed in Section 3.2.

O1. The leader car affects the follower's acceleration pattern.

As long as car A does not pass the front car B, the average speed of A cannot exceed that of B. Therefore, as shown in Fig. 6,[1] the average relative speed between A and B is almost zero. If the relative speed is negative (positive), A tries to catch up (distance itself from) with B by acceleration (deceleration). Therefore, the front car B determines an average behavior of the follower car A's acceleration.

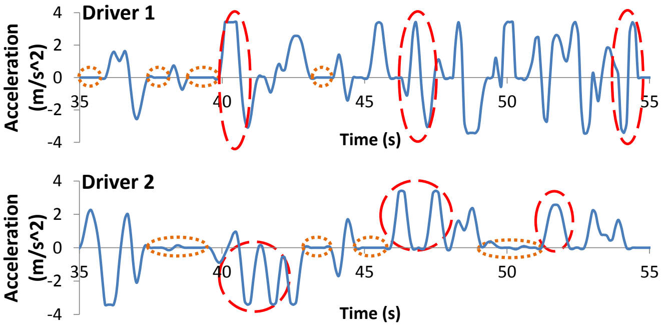


Figure 7: Reaction strength and time; we picked two cars on the US-101 highway.

O2. The operator's driving behavior affects the acceleration pattern.

We capture each operator's driving pattern with two variables: *reaction strength*—how quickly the operator/driver accelerates to achieve a desired speed, and *reaction time*—how long it takes for the driver to react to a given situation, which are represented in Fig. 7, respectively, by the slope of acceleration and the duration during which the acceleration is zero. For instance, some drivers press the acceleration pedal aggressively until the vehicle reaches the desired speed, but others gradually increase/decrease speed with a long acceleration time. Although the reaction strength varies with drivers, we found that the reaction strength of a driver remains similar all time. The reaction time also depends on the driver. While drivers with a long reaction time tend to maintain their vehicles' speed instead of acceleration or deceleration, others with a short reaction time immediately respond to a given situation, e.g., the front car's slowdown.

O3. The traffic regulations affect the acceleration pattern.

The traffic regulations include speed limit, traffic lights and other traffic signs, and have a great impact on acceleration. For example, every driver is supposed to decelerate at a stop sign or speed bump, and hence, this is independent of drivers as shown in Fig. 8. Using offline profiling of road information, we can predict the impact of traffic regulations on the acceleration pattern.

3.2 Acceleration prediction methods

To predict an operator's acceleration pattern, we present five methods, each with its own properties. The first three methods utilize the history of previous acceleration patterns, thus indirectly/directly addressing O2. On the other hand, the next two methods address O1 and O3, respectively, by using an existing traffic flow model and road segmentation based on traffic regulations. The five methods are summarized in Table 2. Considering the fact that the five methods are effective for different environments/situations, we combine them to devise a unified acceleration prediction method, exploiting advantages of the five individual methods.

[1] We obtained Figs. 6, 7 and 8 using the evaluation tools to be described in Section 4.

prediction methods	previous acceleration	previous acceleration with reaction strength	correlation with past accelerations	car following model	road segmentation
addressed observations	O2 (implicitly)	O2 (implicitly) and reaction strength in O2	O2 (implicitly)	O1	O3
limit to application	no limitation	no limitation	no limitation	no limitation	segment transfer
sampling rate	dependent	dependent	dependent	independent	independent
calculation cost	low	low	high	low	low
input data	a_n	a_n, reaction strength	$a_{n-q}, \cdots, a_n$	ΔV_n, ΔX_n	V_n, $limV_{n+1}$

Table 2: Properties of the acceleration prediction methods

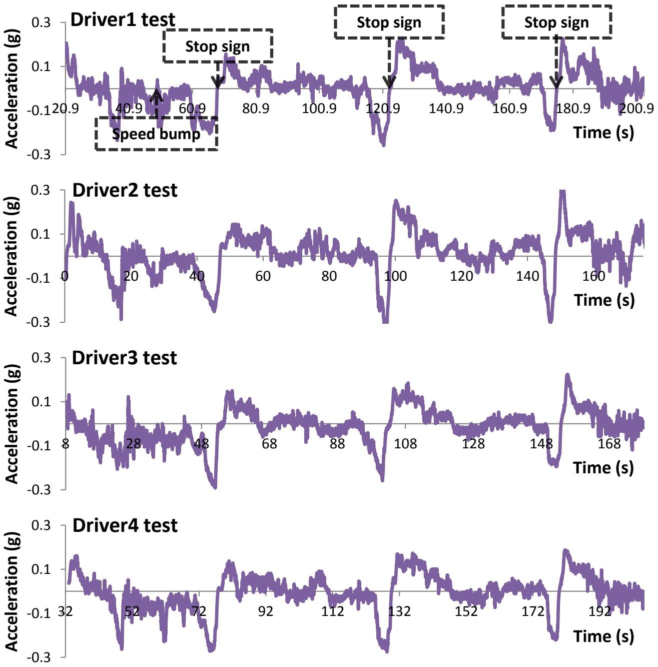


Figure 8: Acceleration pattern between road segments; four different drivers drove around the University of Michigan campus and the path includes three stop signs and a speed bump.

3.2.1 *Previous acceleration*

In this method, the current acceleration is used as the next acceleration prediction. Its advantage is simplicity. This method, although naive, results in accurate prediction in case of slow acceleration fluctuation, or high sampling frequency with an almost zero battery reconfiguration time. However, this method alone cannot cope with abrupt changes in acceleration in many practical situations.

3.2.2 *Previous acceleration with reaction strength*

To improve the simple acceleration method that uses the previous acceleration, we incorporate the reaction strength in O2. To do this, we calculate the acceleration slope when the driver starts acceleration, and use it for the next acceleration prediction as follows:

$$\hat{a}_n = a_{n-1} + k \cdot (a_{n-1} - a_{n-2}), \qquad (3)$$

where k is the sensitivity coefficient for the acceleration slope (reaction strength). This additional consideration yields more accurate prediction than the previous acceleration method while requiring only two more basic operations.

3.2.3 *Correlation with past accelerations*

While the two history-based methods focus on the previous acceleration, this method employs a linear filter that correlates a long history of earlier accelerations with the next acceleration. For this, we first measure and record timestamped accelerations, and then generate correlation factors based on the errors between recorded (input) and predicted accelerations. Finally, we predict the next acceleration by inner-product of the input and the correlation vectors. This method is useful when the next acceleration is highly correlated with earlier accelerations. It achieves higher accuracy than the two history-based methods at the expense of high computational complexity. Note that all history-based methods including this depend on the data sampling rate, because the higher rate can mask rapid changes of acceleration.

3.2.4 *Car following model*

Many transportation researchers studied microscopic driver behavior to describe a real traffic flow. The car following model [17, 18], a well-known microscopic traffic flow model, describes the vehicle's acceleration in terms of the relative speed and distance between a car and the leading car in front. Among existing car following models [17, 19, 20, 21], we choose the popular one in [20], which extends the classic model [17] based on the observation for optimal velocity. The main advantage of prediction with the car following model is its independence of the sampling frequency. Although this model provides which action is needed to maintain a relative speed of zero and a constant desired distance between the two cars, it does not directly provide the vehicle's acceleration, resulting in inaccurate prediction.

3.2.5 *Road segmentation*

To address the effect of traffic regulations, we segment the road based on traffic regulations, and then predict the acceleration pattern when a car transits from a segment to another. This method enhances the accuracy of acceleration prediction between road segments; for example, if the

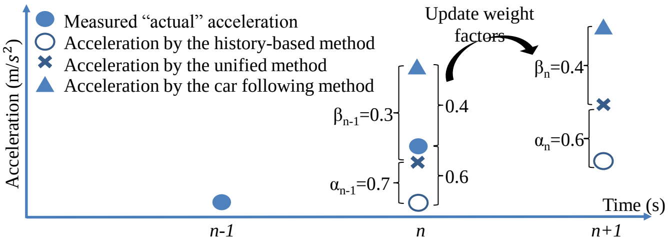


Figure 9: Update of the weight factors and calculation of the next acceleration using the weight factors

next segment's speed limit is higher or lower than the current one, we can predict the acceleration during the segment transition by calculating of the difference in speed (limits) between the current and next segments. Since the traffic regulation information can be extracted from a pre-downloaded map, it does not depend on the data sampling frequency.

3.2.6 *Unified prediction of acceleration*

As shown in Table 2, each of the above methods has its own characteristics and advantages: (i) the history-based (the first three) methods depend on the sampling rate and are effective when the rate is high; (ii) the method based on the car following model is independent of the sampling rate and thus more widely applicable; and (iii) the road segmentation method is specialized for segment transition.

Considering these characteristics, we propose a unified method that predicts the acceleration in the next interval as follows: (a) when the car crosses a segment boundary, it uses the road segmentation method to predict acceleration; (b) otherwise, it predicts acceleration using a combination of methods based on acceleration history and car-following model.

There are two variables that affect the decision of (b): the sampling rate and the reaction time. As to the sampling rate, the history-based methods (the method with the car following model) are more effective than the counterpart under a higher (lower) sampling rate, because the car following model guarantees the same level of accuracy even under a low sampling rate, while the prediction accuracy of the history-based methods is proportional to the sampling rate. As to the reaction time, the history-based methods are accurate for a long reaction time, because drivers with a long reaction time tend to maintain their speed. On the other hand, the acceleration by drivers with short reaction time can be described well by the car following model as the drivers react immediately to a given traffic situation.

Considering the two variables, the unified method handles the situation (b) via the weights α and β (such that $\alpha+\beta = 1$), which are respectively the weight of one of the history-based methods and that of the method with the car following model. The weights can be calculated adaptively so as to reduce the error in predicting the next acceleration. Fig. 9 shows an example of the process of updating the weights. Suppose that the weights at step $n-1$ are set to $\alpha_{n-1} = 0.7$ and $\beta_{n-1} = 0.3$, and the method uses them to predict the acceleration of step n. Then, at step n, the actual acceleration is different from the predicted; if the weights were $\alpha_{n-1} = 0.6$ and $\beta_{n-1} = 0.4$, the prediction would have no error. Using this information, the method updates the weights at step n for the acceleration prediction of step $n+1$ as shown in the figure.[2] In Section 4 we will compare the accuracies of acceleration prediction by the unified and the other methods.

4. EVALUATION

We now evaluate the proposed power requirement predictor. We first describe tools and settings used for the evaluation. Then, we demonstrate the accuracy of acceleration predictor (AP) and power requirement predictor (PRP).

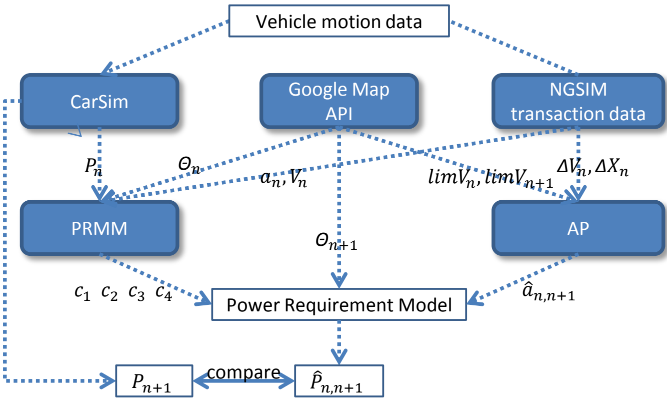


Figure 10: Block diagram of the PRP evaluation system

4.1 Evaluation tools and settings

To evaluate PRP, three types of offline/driving data are needed as inputs of PRP: (i) road information (θ, $limV$), (ii) real driving data (a, V, ΔX, ΔV), and (iii) the measured power consumption (P). We designed an emulator of PRP, which provides actual or realistic data from three popular tools as shown in Fig. 10: Google Maps API [12], NGSIM [22], and CarSim [23].

Google Maps API enables developers to build map-based applications which allow us to access the road information database including the road type, the speed limit, and the elevation along with the GPS data, thus providing road information (i).

NGSIM (next generation simulation program) was initiated by the US DOT, and accompanied by a core of open behavioral algorithms in support of traffic simulation with a primary focus on microscopic modeling. NGSIM collects high-quality primary traffic and trajectory data to test new algorithms, thus providing real driving data (ii). We use the vehicle trajectory data including vehicle identification number, instantaneous velocity and acceleration of the vehicle,

[2] While the unified method calculates average prediction error of x previous steps for the update of weights, we illustrate $x = 1$ for simplicity of presentation.

and distance between the front-center of a vehicle and that of the preceding vehicle.

CarSim is a vehicle model simulator that allows users to design, develop, test, and plan vehicle programs in a variety of environments. It can simulate the dynamic behavior of passenger cars, racecars, light trucks, and utility vehicles, under specified road type and slope. CarSim provides animations, and outputs the required power, which can be plotted, analyzed, and exported to other software such as MATLAB, Excel, and other optimization tools. In our emulator, CarSim receives the vehicle motion data from NGSIM, and then generates the required power, as shown in Fig. 10.

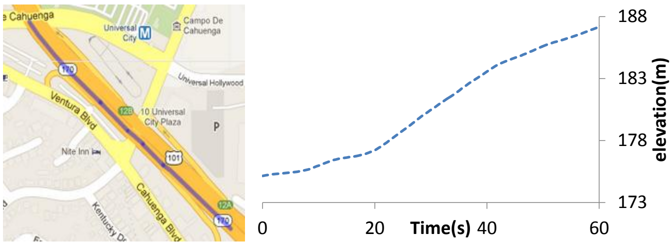


Figure 11: Road location and slope

For NGSIM and CarSim, we use a hatchback-type car with 205/55 R16 tires, assuming a paved road. We also assume that there is no regenerative break system, and therefore, the required power is always non-negative. For road information, we select the southbound US-101 highway in Los Angeles, CA from Google Maps API as shown in Fig. 11, and NGSIM tested the car on the road with traffic of June 15th, 2005.

4.2 Evaluation results

In this subsection, we evaluate the accuracy of AP and PRP. We first discuss hardware factors that may degrade accuracy, such as the sampling rate of the required data, the times for computing the required power and reconfiguring the connection of battery cells. All of these factors are related to manufacturing costs; if a BMS is equipped with high-performance sensors, processors, and quickly-reconfigurable systems, the undesirable effect of these factors on prediction accuracy will be reduced. However, BMS designers are required to use cheap and low-powered devices to reduce the cost and power consumption of a BMS, which will in turn lower accuracy of PRP. While the degree of inaccuracy caused by these factors varies with hardware, it is inevitable for a BMS to spend time for sensing physical data, calculating the required power, and incorporating them into the BMS. To address this latency, we abstract these factors as a sampling rate, and evaluate it with three different rates: 2Hz, 5Hz, and 10Hz.

Such a finite sampling rate causes prediction error as shown in Fig. 12. That is, even if PRP exactly predicted the required power for step n at step $n-1$, the required power between steps n and $n+1$ varies, resulting in some error. This also holds for acceleration prediction. So, let Opt-P (Opt-A) denote this perfect prediction of power (acceleration) with the inherent error due to the finite sampling rate.

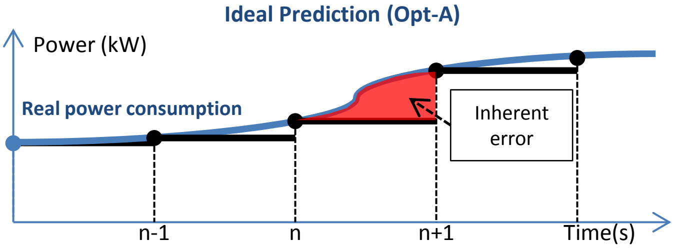


Figure 12: Inherent error due to finite sampling frequency

We now show the accuracy of AP and PRP with the following acceleration prediction methods, compared to Opt-A.

- Prev (previous acceleration) in Section 3.2.1,
- Prev+ (previous acceleration with reaction strength) in Section 3.2.2,
- Corr (correlation with past accelerations) in Section 3.2.3,
- CF (car following model) in Section 3.2.4,
- RS (road segmentation) in Section 3.2.5, and
- Uni(H) (the unified prediction of acceleration with one of the history-based methods, i.e., H is either Prev, Prev+, or Corr) in Section 3.2.6.

4.2.1 *Accuracy of AP*

We first compare the acceleration prediction methods when the car is within a single road segment. To evaluate the accuracy of acceleration prediction, we used the driving record produced by NGSIM, and adapted each prediction method to predict the vehicle's next acceleration. To compare the performance of each prediction method, we calculated the average of absolute prediction errors, as shown in Table 3. Overall, the higher sampling rate, the more accurate prediction because more information is available to predict the next acceleration. At a low sampling rate, prediction by CF shows less prediction error since CF is the only method that is independent of the sampling rate. On the other hand, history-based acceleration predictions, such as Prev, Prev+, and Corr, performed well at a high sampling rate. Uni(·), in general, yields more accurate acceleration prediction than other methods irrespective of the sampling rate. One exception is the prediction by CF at 2Hz, because there are a limited number of driving information samples at 2Hz and then errors of Uni(·) are likely to occur during the learning of the driver's reaction.

Now, we show how RS is useful when a car crosses segment boundaries. To predict the acceleration between road segments, we picked two drivers from NGSIM trajectory data. The leading car passes a traffic light just before the light turns to red, but the following car stops at the traffic light since it just turned to red. Then, we compare the accuracy of acceleration prediction with/without RS when CF is used as the baseline prediction method. Since the the following

	Average error (m/s^2)		
Prediction methods	2Hz	5Hz	10Hz
Opt-A	0.795	0.369	0.168
Prev	1.518	0.989	0.556
Prev+	1.566	1.128	0.461
Corr	1.465	1.016	0.444
CF	0.984	1.070	0.985
Uni(Prev+)	1.154	0.992	0.433
Uni(Corr)	1.162	0.948	0.448

Table 3: Average error of acceleration prediction

	Average error (m/s^2)		
Prediction methods	2 Hz	5 Hz	10 Hz
Opt-P	8.671	3.861	1.705
PRP with Opt-A	9.186	4.448	2.458
Actual previous power	15.884	10.461	5.904
PRP with Prev+	17.198	11.212	5.202
PRP with Corr	12.567	10.368	4.962
PRP with CF	10.464	11.006	10.630
PRP with Uni(Prev+)	12.138	10.425	4.916
PRP with Uni(Corr)	10.891	10.220	5.068

Table 4: Average error of power prediction

car cannot accelerate due to the traffic light, CF's prediction of the following car's acceleration shows a large average error as shown in Fig. 13. However, the road segmentation can provide a chance to predict acceleration accurately between different road segments as shown in the figure. The average errors of acceleration prediction are 1.912 and 1.262, respectively, for CF and CF with RS.

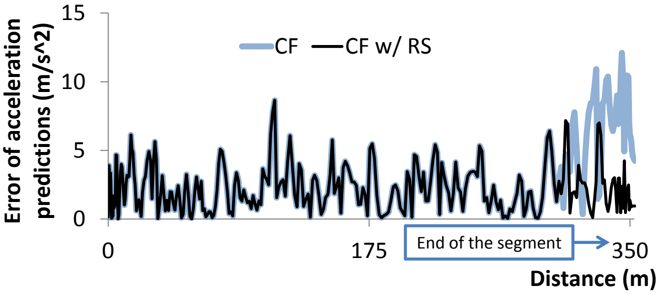


Figure 13: Acceleration prediction between segments

4.2.2 Accuracy of PRP

We first evaluate the power requirement model in Eq. (1). Since Opt-P and Opt-A are the most accurate power and acceleration predictions achievable for a given sampling frequency, we compare Opt-P, and PRP with Opt-A. As shown in Table 4, the difference in average error between the two is marginal, meaning that the power requirement model we employed yields accurate enough prediction of the required power.

Since Opt-P and Opt-A are ideal power and acceleration predictions (thus infeasible), we need to employ one of acceleration prediction methods with PRP; Table 4 summarizes the prediction errors, and Fig. 14 plots the prediction. We included one heuristic (actual previous power) that uses the current actual power consumption as the next power requirement. Note that this scheme does not use our PRP. Compared to the heuristic, PRP with Uni(Corr) shows an improvement of prediction accuracy by 69.2% at 2Hz, 3.7% at 5Hz, and 19.9% at 10Hz within a road segment. If we compare PRP with different acceleration prediction methods, the trend will be similar to the accuracy of AP, because the error in predicting the power requirements is dominated by that of acceleration prediction. For example, Uni(Corr) is in general (one of) the best methods for predicting acceleration that corresponds to the required power associated with PRP. In particular, compared to PRP with CF, PRP with Uni(Corr) predicts closer to the actual required power by 11.0% at 5Hz, and 62.3% at 10Hz.

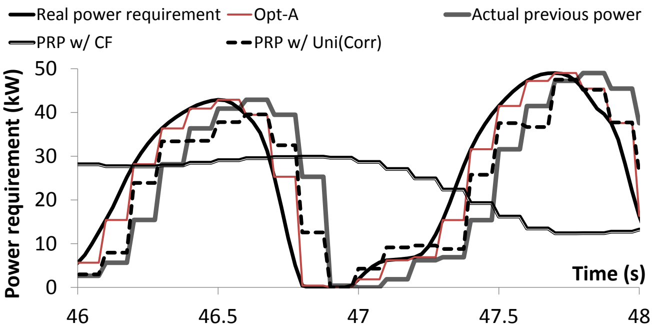


Figure 14: Prediction of power requirement with 10Hz sampling

5. CONCLUSION

Increasing demand to make EVs lighter and less expensive places stringent restrictions on battery size and capacity. It is thus important to devise advanced BMS extending battery capacity without increasing its size. A number of BMSes have been proposed to exploit nonlinear battery characteristics, including recovery and rate-capacity effects.

To introduce adaptability and efficiency into existing BMSes, we have proposed the battery power requirement predictor (PRP). It predicts the amount of required battery power that enables BMSes to allocate battery cells to power EVs efficiently and safely. While recording and linearly regressing vehicle activities such as speed, acceleration, road slope and power consumption, PRP updates the (near) static parameters of the battery power requirement model for EVs. In the meantime, PRP estimates the acceleration, perhaps the most important dynamic parameter of the model, in real time, by accounting for the driver's acceleration pattern, traffic flows and regulations. Our evaluation with actual experimentation and realistic simulation has shown that PRP makes a significant improvement in acceleration prediction over a simple heuristic method often seen in the existing BMSes.

Since PRP was developed to be used in an advanced BMS in Fig. 5, we plan to develop a new battery manager that exploits the predicted power by PRP. We can then find how much PRP with the new battery manager can improve the life and operation time of existing BMSes. It would also be interesting to explore ways of improving PRP. For example, we can enhance the accuracy of the acceleration prediction by sensing the driver's motion on the acceleration pedal. Besides, we may employ more accurate power requirement models, such as those in [7, 8, 9]. Since these models are complex, it would be interesting to investigate how to adapt them for real-time prediction.

Acknowledgement

The work reported in this paper was supported in part by the NSF under grants 0930813 and 1138200.

6. REFERENCES

[1] C. Vyas, D. Hurst, and J. Gartner, "Electric vehicle consumer survey," Pike Research, Tech. Rep., 2011.

[2] "China publishes plan to boost fuel-efficient and new energy vehicles and domestic auto industry," http://www.greencarcongress.com/2012/07/china-20120709.html, 2012, [Online; accessed 15-September-2012].

[3] U. EIA, "Annual energy outlook 2012 with projections to 2035," U.S. Energy Information Administration, Tech. Rep., 2012.

[4] L. Benini, A. Macii, E. Macii, M. Poncino, and R. Scarsi, "Scheduling battery usage in mobile systems," *IEEE Transactions on VLSI systems*, vol. 11, no. 6, pp. 1136–1143, 2003.

[5] H. Kim and K. G. Shin, "Scheduling of battery charge, discharge, and rest," in *Proceedings of the 30th IEEE Real-Time Systems Symposium (RTSS)*, 2009, pp. 13–22.

[6] K. Li, J. Wu, Y. Jiang, Z. Hassan, Q. Lv, L. Shang, and D. Maksimovic, "Large-scale battery system modeling and analysis for emerging electric-drive vehicles," in *Proceedings of the 16th ACM/IEEE International Symposium on Low Power Electronics and Design (ISLPED)*, 2010, pp. 277–282.

[7] P. D. Blair, "Modeling energy and power requirements of electric vehicles," *Energy Conversion*, vol. 18, no. 3, pp. 127–134, 1978.

[8] K. Post, J. Kent, J. Tomlin, and N. Carruthers, "Fuel consumption and emission modelling by power demand and a comparison with other models," *Transportation Research Part A: General*, vol. 18, no. 3, pp. 191–213, 1984.

[9] K. Ahn, H. Rakha, A. Trani, and M. V. Aerde, "Estimating vehicle fuel consumption and emissions based on instantaneous speed and acceleration levels," *Journal of Transportation Engineering*, vol. 128, no. 2, pp. 182–190, 2002.

[10] J. Flinn and M. Satyanarayanan, "Managing battery lifetime with energy-aware adaptation," *ACM Transactions on Computer Systems*, vol. 22, no. 2, pp. 137–179, 2004.

[11] Rotronics, "Vehicle loads," Rotronics, Tech. Rep., 2009.

[12] "Google Map API," https://developers.google.com/maps/, [Online; accessed 21-October-2012].

[13] D. Linden and T. Reddy, *Handbook of batteries*, 3rd ed. McGraw-Hill, 2002.

[14] D. Rakhmatov and S. Vrudhula, "Energy management for battery-powered embedded systems," *ACM Transactions on Embedded Computing Systems*, vol. 2, pp. 277–324, 2003.

[15] H. Kim and K. G. Shin, "On dynamic reconfiguration of a large-scale battery system," in *Proceedings of the 15th IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS)*, 2009, pp. 87–96.

[16] ——, "Dependable, efficient, scalable architecture for management of large-scale batteries," in *Proceedings of the 1st ACM/IEEE International Conference on Cyber-Physical Systems (ICCPS)*, 2010, pp. 178–187.

[17] R. E. Chandler, R. Herman, and E. W. Montroll, "Traffic dynamics: studies in car following," *Operations Research*, vol. 6, pp. 165–184, 1958.

[18] Y. Li and D. Sun, "Microscopic car-following model for the traffic flow: the state of the art," *Journal of Control Theory and Applications*, vol. 10, pp. 133–143, 2012.

[19] K. Nagel and M. Schreckenberg, "A cellular automaton model for freeway traffic," *J. Phys. I France*, no. 12, pp. 2221–2229, 1992.

[20] M. Bando, K. Hasebe, A. Nakayama, A. Shibata, and Y. Sugiyama, "Dynamics model of traffic congestion and numerical simulation," *Physics Review E*, vol. 51, pp. 1035–1042, 1995.

[21] M. McDonald, J. Wu, and M. Brackstone, "Development of a fuzzy logic based microscopic motorway simulation model," in *IEEE Conference on Intelligent Transportation System*, 1997, pp. 82–87.

[22] "Introduction of NGSIM," http://ngsim-community.org/, [Online; accessed 21-October-2012].

[23] "Carsim simulator," http://www.carsim.com/products/carsim/index.php, [Online; accessed 21-October-2012].