State of Charge Estimation for Electric Scooters by Using Learning Mechanisms

D. T. Lee, *Fellow, IEEE,* S.-J. Shiah, C.-M. Lee and Y.-C. Wang

Abstract— Because of its nonlinear discharge characteristics, the residual electric energy of a battery remains to be an open problem. As a result, the reliability of electric scooters or electric vehicles is lacking. To alleviate this problem and enhance the capabilities of present electric scooters or vehicles, we propose a state-of-charge learning system that can provide more accurate information about the state-of-charge or residual capacity when a battery discharges under dynamic conditions. The proposed system is implemented by learning controllers, fuzzy neural networks and cerebellar model articulation controller networks, which can estimate and predict nonlinear characteristics of the energy consumption of a battery. With this learning system, not only could it give an estimate of how much residual battery power is available, but it also could provide users with more useful information such as an estimated traveling distance at a given speed, and the maximum allowable speed to guarantee safety arrival at the destination.

Index Terms— Battery, state of charge, learning controller, electric scooter, electric vehicle, fuzzy neural network, cerebellar model articulation controller.

I. INTRODUCTION

IN Taiwan, most of the commercially available electric scooters are powered by four 12-volt lead-acid batteries which are connected in series. These scooters to date have some drawbacks, including high cost, long battery recharging time, relatively short traveling distance for each re-charge, and inadequate feedback information to the user with respect to the residual battery capacity. The lack of a more reliable or accurate electric energy prediction often results in situations that the riders unwittingly run out of battery power before they reach their destinations or a facility to re-charge the battery. This uncertainty as to when the battery power will run out, could be rather troublesome and therefore hinder the sale of electric scooters.

Although lead-acid batteries have been used widely to store and supply electric energy, a lot of research is still underway, searching for a good state-of-charge (SOC) estimation method for specific applications. For an electric scooter or electric vehicle (EV), the crux in getting a good estimate of the lead-acid battery SOC is mainly due to lack of effective means to predict the dynamic nonlinear battery discharge characteristics which may vary with the discharge rate, depth of discharge, recharging times, temperature, aging etc.

At present, there are many known methods for estimating the SOC, and they can be classified as open circuit voltage measurement [1], [2], loaded voltage measurement [3], [4], ampere hour accumulation or coulometric measurement [5], [6], [7], [8], [9], [10], [11], [12], impedance measurement [13], and battery learning model [14], [15], [16]. The open circuit measurement can be used to estimate the SOC at no load condition; and the loaded voltage measurement is suitable for constant load current applications. For the electric scooter or EV, both methods cannot provide good estimations, because the load current varies with the road condition, payload, and speed. The coulometric measurement uses accumulated discharge current to estimate the SOC based on a pre-recorded data describing the relationship between battery discharge current and capacity. The pre-recorded data is not valid in every discharge condition, so methods making use of models of fixed parameters to estimate SOC will suffer from loss of precision when the discharge conditions change over time. The impedance measurement needs to measure the frequency response of the battery to determine its SOC. Because it needs extra electric circuits and function generators, it is not suitable to be implemented in applications like electric scooters or EVs.

To meet the need to provide online calculation of SOC and increase its estimation accuracy without using extra measurement hardware circuit we propose an SOC learning system based on a learning mechanism and the coulometric measurement for estimating the SOC of batteries under dynamic conditions. The learning mechanism consists of a fuzzy neural network (FNN) [17] and a cerebellar model articulation controller (CMAC) [18], [19], and is used to estimate nonlinear discharge behavior of battery under different conditions, and the information is further utilized to modify the results calculated by coulometric measurement. This design is shown to have an advantage that only a few experimental data is needed to obtain a better estimate of SOC under all possible conditions.

II. PROPOSED SOC LEARNING CONCEPT

A. Coulometric Algorithm

We will first describe the so called coulometric measurement that was generally used for the SOC estimation.

In this paper, the capacity of a battery means the available capacity released from a battery at a certain discharge rate or discharge current. And when the battery voltage drops below

a cutoff voltage V_{cutoff}, we say its capacity is practically zero. In the electric scooter application, we use the battery voltage at and below which a scooter cannot operate as the cutoff voltage. Another important factor for estimating SOC is the discharge efficiency, and it is defined as

$$\begin{aligned} \eta_I &= \frac{I \cdot T}{C_0} \\ &= \frac{C_I^{dis}}{C_0} \end{aligned} \tag{1}$$

where I stands for a constant discharge current, T for total discharge time, and C_0 for the available capacity corresponding to a specific reference discharge current I_{ref}. The discharge efficiency described by Eq.(1) represents the ratio of how much capacity a battery can offer at a given discharge rate I with respect to a certain capacity C_0.

Since the discharge rate varies drastically in the electric scooter or EV application, we need to formulate the relation between two capacities at their corresponding discharge currents. This can be derived from the constant current discharge data shown in Figure 1. In Figure 1, C_m and C_n are the total released
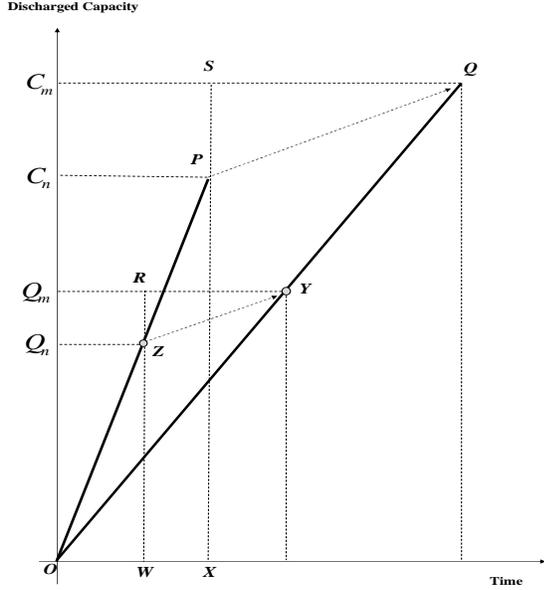

Fig. 1. The constant current discharge curves : discharged capacity versus time

capacities from a battery at discharge currents I_m and I_n respectively and $\overline{OQ}$ and $\overline{OP}$ are the corresponding time histories of discharged capacities. That is to say that the battery would release C_n(Ah) at I_n(Amp) while it would release C_m(Ah) at I_m(Amp). The following is in general true that the battery releases capacity $C_m > C_n$ at discharge current $I_m < I_n$. We can roughly estimate a released capacity Q_m at discharge rate I_m by using its value Q_n at discharge rate I_n in a certain time interval in a linear fashion. By drawing a line $\overrightarrow{ZY}$ parallel to $\overline{PQ}$, we can have an intersection point Y on the line $\overline{OQ}$, and the y-coordinate of Y is Q_m. We can use the following pairs of similar triangles to calculate Q_m.

$$\Delta ZYR \sim \Delta PQS \tag{2}$$

$$\Delta OYZ \sim \Delta OQP \tag{3}$$

$$\Delta OZW \sim \Delta OPX \tag{4}$$

According to Eqs.(2), (3), and (4), we have

$$\begin{aligned} \frac{Q_m - Q_n}{C_m - C_n} &= \overline{ZY}/\overline{PQ} = \overline{OZ}/\overline{OP} = \overline{ZW}/\overline{PX} \\ &= \frac{Q_n}{C_n} \end{aligned} \tag{5}$$

Thus, a released capacity Q_m which is relative to I_m can be calculated as

$$\begin{aligned} Q_m &= Q_n + Q_n \frac{C_m - C_n}{C_n} \tag{6} \\ &= Q_n \frac{C_m}{C_n} \tag{7} \\ &= Q_n \frac{\left(\frac{C_m}{C_0}\right)}{\left(\frac{C_n}{C_0}\right)} \tag{8} \\ &= \left(\frac{\eta_{I_m}}{\eta_{I_n}}\right) Q_n \tag{9} \end{aligned}$$

Since values of capacities released from a battery vary with discharge currents we need a reference discharge current to transfer the SOC under different discharge currents, and these values of SOC will be consistent. Thus, we use Eq.(9) to approximate the instant released capacity as follows.

$$C_{I(k)}^{dis} = \frac{\eta_{I(k)}}{\eta_{I(k-1)}} C_{I(k-1)}^{dis} + I(k)\Delta T \tag{10}$$

$$C_{I(1)}^{dis} = I(1)\Delta T \tag{11}$$

where ΔT is the sampling time interval; $C_{I(k-1)}^{dis}$ and $C_{I(k)}^{dis}$ are released capacities corresponding to discharge rates $I(k-1)$ and $I(k)$ at the $(k-1)$th and kth sampling time.

According to Eq.(10) and (11), we can calculate the SOC corresponding to I_{ref} at the kth sampling time as

$$SOC(k) = 1 - \left(\frac{1}{C_0}\right)\left(\frac{C_{I(k)}^{dis}}{\eta_{I(k)}}\right) \tag{12}$$

B. State of Health

The state of health(SOH) indicating the degree of aging of a battery is also an important factor which determines the accuracy of the estimated SOC. In general, the capacity check is the most accurate method for determining SOH of a battery, but it is very time consuming [20], [21]. So impedance measurement [13] is a good choice for checking the SOH of a battery. But the impedance measurement needs additional hardware circuits, which is a disadvantage for the electric scooter or EV application. Because the battery aging is an obvious phenomenon in the electric scooter application, we need to introduce SOH as a variable when estimating SOC. The SOH used in our proposed system is defined as

$$SOH = \frac{C_{I_{ref}}}{C_0} \tag{13}$$

where $C_{I_{ref}}$ is the total discharged capacity at I_{ref}. Presumably $C_{I_{ref}} = C_0$, i.e., SOH= 100%, when the battery
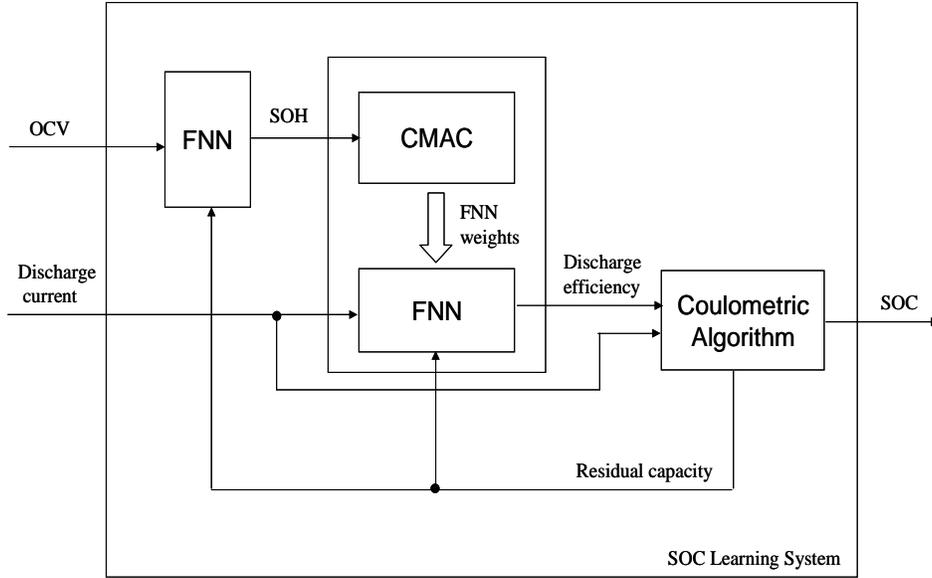
Fig. 2. The structure of the proposed SOC learning System

is brand new. In other words, SOH is an indicator of how much capacity can be released at the reference discharge rate after usage, with respect to the initial available capacity when the battery is brand new. Basically, the coulometric-like algorithms use Eqs.(10) and (11) to fix the value of discharged capacity and then calculate the SOC. But they use different methods to obtain the correction coefficient $\frac{C_m}{C_n}$ as described by Eq.(7) [8], [9], [10], [11]. Some of these methods use an average discharge current determined by a whole discharge process as the reference, and others use the reference discharge current selected from characteristics of electric scooters or vehicles. The former method cannot correct capacity error when discharge rate changes very drastically. The latter can only deal with some specific cases, because characteristics of electric vehicles are almost determined by driving behaviors of users. Moreover, as linear methods were used to fit correction coefficients and aging of battery (SOH) problem was not considered, SOC estimation is thus imprecise in the electric scooter or EV application [12].

C. Learning Controllers

There are two well-known types of learning controllers, artificial neural networks and fuzzy systems. Both are biologically inspired and intended to model human experience [22], [23], [24]. The structure of artificial neural networks is modeled after the organization of the brain, although the similarity between the two is actually slight [23], [25], [26], [27]. On the other hand, fuzzy systems are meant to encode pieces of knowledge presented by experts [24], [26], [28], [29]. However, most of them need to repeat the learning process each time a new pattern is encountered [30]. Otherwise, a neural network will require a huge number of neurons or a fuzzy system will require numerous rules because the learning space needed to handle arbitrary nonlinear dynamics is quite

large [31], [32]. For solving this problem, a learning structure consisting of FNNs and CMAC networks was proposed in [33]. It was designed to effectively generalize from the data patterns learned to predict information about a nonlinear dynamic system, which could not be obtained in advance due to its huge size of learning space.

D. Combination of the Coulometric Measurement and Learning Controllers

In general, the dynamic behavior of a battery is nonlinear and complex. Therefore, it is by no means an easy task to obtain an adequate battery model and its parameters accurately. A learning controller, however, is capable of tackling highly complex dynamics without an explicit model dependence, and therefore is an attractive alternative modeling nonlinear dynamics of a battery. Moreover, in the electric scooter application, it is necessary to downsize the hardware, software, and physical memory of data storage requirement, so we shall use the learning structure consisting of an FNN and a CMAC as the core of our SOC learning system.

To be more specific we propose a combination of the coulometric measurement described by Eqs. (10), (11) and (12) and learning controllers to estimate the SOC. The accuracy of an estimated SOC depends on how accurate the discharge efficiency can be estimated. Since the discharge efficiency is a nonlinear function of discharge rate, residual capacity, temperature, aging of battery, etc., and it is impossible to obtain all discharge efficiency data from experiments, we use a learning structure which learns to fit the available experimental data of discharge efficiency at some specific conditions, and then generalizes the discharge efficiency surface to cover the whole dynamic state space of a battery. With this learned discharge efficiency surface, we can improve the degree of accuracy of the coulometric measurement. In our electric

scooter application, the SOC strongly depends on the discharge current, SOH, and residual capacity. Temperature is also another important factor, but we only consider the condition in which electric scooters are used in an urban area with negligible temperature effects on batteries.

The conceptual organization of our proposed SOC learning system is shown in Figure 2. The proposed system includes a learning mechanism composed of an FNN and a CMAC, a single FNN, and a coulometric measurement module. Components of an input vector to the learning mechanism are discharge current, residual capacity, and SOH. The component, SOH, is used as the input to the one-dimensional CMAC in the learning mechanism, and this CMAC will send the corresponding FNN weights to the two-dimensional FNN in the same learning mechanism. According to the FNN weights and the other two components, discharge current and residual capacity, of the input vector, the FNN produces an output, the estimated discharge efficiency, of this learning mechanism. This estimated discharge efficiency along with discharge current will be sent to the coulometric measurement module to calculate the instant SOC and residual capacity. According to the estimated residual capacity and the open circuit voltage (OCV) of the battery, a single FNN is used to estimate the battery SOH. Since the OCV can be used to predict residual capacity of a battery without the influence of its temperature and discharge history [1], [2], [34], it is a good indicator for estimating SOH. The only restriction is that the battery must be stabilized for at least two hours in our electric scooter application. The present residual capacity and SOH, combined with the sensed discharge current will be used as input to the learning mechanism at the next sampling time. The above process will repeat, and thus we can get the value of the dynamic SOC at every sampling time.

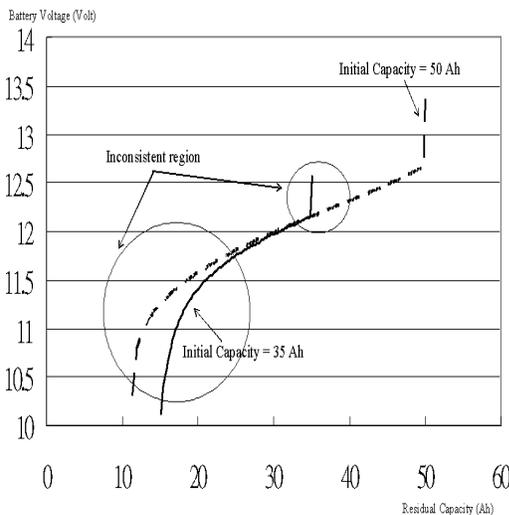In order to avoid having the learning process divergent, we

Fig. 3. Two curves of battery voltage VS residual capacity at the same discharge current(12.4Amp) and temperature, but different initial capacities(50Ah and 35Ah).

do not use the loaded battery voltage as one of input variables
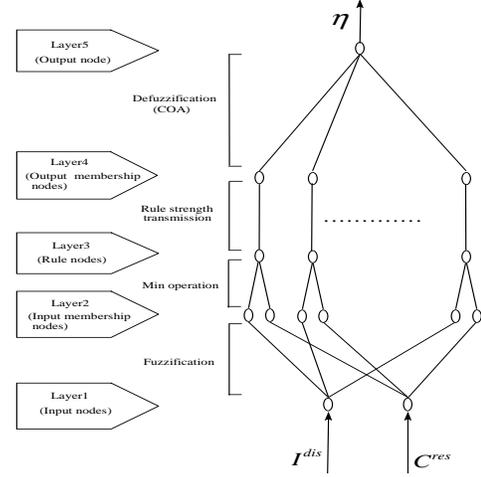

Fig. 4. The structure of the FNN.

to our learning mechanism, and only use learning controllers to estimate SOC [16]. In Figure 3, a battery discharge characteristics is shown when the battery is discharged at the same discharge current and temperature but with different initial capacities. We can see that the relationship between battery voltage and residual capacity is a not a function in the inconsistent region as marked in circles in Figure 3, and this means a value of the battery voltage will be mapped to over two different values of residual capacities. This kind of training patterns will lead to diverging of learning processes or converging to an average value of all desired outputs.

III. SOC LEARNING SYSTEM IMPLEMENTATION

A. Implementation of the FNN

An FNN shown in Figure 4 is designed to realize the process of fuzzy reasoning by using the structure of a neural network. The parameters of fuzzy reasoning are expressed by the connection weights or node functions of the neural network [17], [35]. The representation of a fuzzy system using a neural network enables us to take advantage of the learning capability of the neural network for automatic tuning of the parameters in the fuzzy system. In Figure 4, the inputs to the FNN are the sensed discharge current and residual capacity calculated by coulometric measurement module, and the output is the discharge efficiency. The structure of the FNN adopted here consists of five layers of nodes, which are of the same type within the same layer. Each of the five layers performs one stage of the fuzzy inference process, as described below.

Layer 1: This layer is the input layer, and the inputs are transmitted to the next layer directly without any computation. In this paper, x_1 is the discharge current I^{dis} and x_2 is the residual capacity C^{res}.

Layer 2: This layer is intended for the input membership functions, and used to perform the fuzzification process of the fuzzy system. The Gaussian function is used as a membership

function. Therefore, we have

$$O_{ij}^{(2)} = \exp\left\{-\left(\frac{x_i - m_{ij}^{(2)}}{\sigma_{ij}^{(2)}}\right)^2\right\}, \qquad (14)$$

where $m_{ij}^{(2)}$ and $\sigma_{ij}^{(2)}$ are the mean and the variance, respectively, of the Gaussian membership function of the jth term of the ith input variable x_i.

Layer 3: This layer is intended for the implementation of the fuzzy rules. The jth node in this layer represents the firing strength of the jth fuzzy rule, which is defined as a fuzzy conditional statement of the form

$$R^j : \text{IF } x_1 \text{ is } A_{1j} \text{ and } x_2 \text{ is } A_{2j}, \text{ then } O^{(5)} \text{ is } B_j.$$

where x_1 and x_2 are input variables, $O^{(5)}$ represents the output variable. A_{1j}, A_{2j}, and B_j represent linguistic terms, such as small, medium, and large. In this layer, each node also outputs the firing strength $O_j^{(3)}$ by performing the fuzzy AND.

$$O_j^{(3)} = \min_{\forall i}(O_{ij}^{(2)}), \qquad (15)$$

where $O_{ij}^{(2)}$ is the jth term of x_i in Layer 2, connected to the jth node in Layer 3.

Layers 4 and 5: Layer 4 is intended for the output membership functions, and the Gaussian function is also used as a node function for each node in this Layer. Layer 5 is the output layer, which has as many nodes as there are output variables. In Figure 4, only one node is needed for the discharge efficiency η. These two layers work together for performing the center of area defuzzification process. Thus, the output of the FNN can be represented as

$$\eta = O^{(5)} \qquad (16)$$

$$= \frac{\sum_j (m_j^{(4)} \sigma_j^{(4)}) O_j^{(3)}}{\sum_j \sigma_j^{(4)} O_j^{(3)}} \qquad (17)$$

where $m_j^{(4)}$ and $\sigma_j^{(4)}$ are the mean and the variance, respectively, of the Gaussian membership function of the jth term of output variable $O^{(5)}$ in layer 5. Because the number of rules in Layer 3 and weights for the input and output layers (Layer 1 and 5) are fixed, the parameters to learn in this FNN are the modifiable weights present on the input links to Layers 2 and 4, which correspond to the input and output membership functions. When the FNN learns the parameters of the input and output membership functions for generating the discharge efficiency η, an error is first specified in the last layer (Layer 5). We define this error as

$$E = \frac{1}{2}(\eta_d - \eta)^2 \qquad (18)$$

$$= \frac{1}{2}(\eta_d - O^{(5)})^2 \qquad (19)$$

where η_d represents the desired discharge efficiency obtained from experiments. This error is then backpropagated to adjust the parameters from layer to layer sequentially. With this error and some straightforward manipulations, we can derive updates of the parameters in Layers 2 and 4, and this will be developed in the following.

Layer 4: Using Eqs.(17) and (19), the amount of modification for $m_j^{(4)}$ is derived as

$$\Delta m_j^{(4)} = -\beta\left(\frac{\partial E}{\partial O^{(5)}}\right)\left(\frac{\partial O^{(5)}}{\partial m_j^{(4)}}\right)$$

$$= \beta(\eta_d - O^{(5)})\left(\frac{\sigma_j^{(4)} O_j^{(3)}}{\sum_j \sigma_j^{(4)} O_j^{(3)}}\right) \qquad (20)$$

where β is the learning rate.

Also, the amount of modification for $\sigma_j^{(4)}$ is derived as follows by using Eqs.(17) and (19).

$$\Delta \sigma_j^{(4)} = -\beta\left(\frac{\partial E}{\partial O^{(5)}}\right)\left(\frac{\partial O^{(5)}}{\partial \sigma_j^{(4)}}\right)$$

$$= \beta(\eta_d - O^{(5)})(m_j^{(4)} - O^{(5)})$$

$$\times \left(\frac{O_j^{(3)}}{\sum_j \sigma_j^{(4)} O_j^{(3)}}\right) \qquad (21)$$

Layer 2: In order to derive the updates of $m_{ij}^{(2)}$ and $\sigma_{ij}^{(2)}$, the terms, $\frac{\partial O^{(5)}}{\partial O_j^{(3)}}$, $\frac{\partial O_j^{(3)}}{\partial O_{ij}^{(2)}}$, $\frac{\partial O_{ij}^{(2)}}{\partial m_{ij}^{(2)}}$, and $\frac{\partial O_{ij}^{(2)}}{\partial \sigma_{ij}^{(2)}}$, must be first calculated. According to Eqs.(14), (15), and (17), the results of calculation are shown below:

$$\frac{\partial O^{(5)}}{\partial O_j^{(3)}} = \frac{(m_j^{(4)} - O^{(5)})\sigma_j^{(4)}}{\sum_j \sigma_j^{(4)} O_j^{(3)}} \qquad (22)$$

$$\frac{\partial O_j^{(3)}}{\partial O_{ij}^{(2)}} = \begin{cases} 1, & \text{if } O_{ij}^{(2)} \text{ is the minimum node} \\ 0, & \text{otherwise.} \end{cases} \qquad (23)$$

$$\frac{\partial O_{ij}^{(2)}}{\partial m_{ij}^{(2)}} = \frac{2O_{ij}^{(2)}(x_i - m_{ij}^{(2)})}{(\sigma_{ij}^{(2)})^2} \qquad (24)$$

$$\frac{\partial O_{ij}^{(2)}}{\partial \sigma_{ij}^{(2)}} = \frac{2O_{ij}^{(2)}(x_i - m_{ij}^{(2)})^2}{(\sigma_{ij}^{(2)})^3} \qquad (25)$$

Now, we can derive the amounts of modification for $m_{ij}^{(2)}$ and $\sigma_{ij}^{(2)}$ by using Eqs.(19), (22), and (23)-(25). If $O_{ij}^{(2)}$ is the minimum node in layer 2,

$$\Delta m_{ij}^{(2)} = -\beta\left(\frac{\partial E}{\partial O^{(5)}}\right)\left(\frac{\partial O^{(5)}}{\partial O_j^{(3)}}\right)\left(\frac{\partial O_j^{(3)}}{\partial O_{ij}^{(2)}}\right)\left(\frac{\partial O_{ij}^{(2)}}{\partial m_{ij}^{(2)}}\right)$$

$$= \beta(\eta_d - O^{(5)})\left(\frac{(m_j^{(4)} - O^{(5)})\sigma_j^{(4)}}{\sum_j \sigma_j^{(4)} O_j^{(3)}}\right)$$

$$\times \left(\frac{2O_{ij}^{(2)}(x_i - m_{ij}^{(2)})}{(\sigma_{ij}^{(2)})^2}\right) \qquad (26)$$

$$\Delta \sigma_{ij}^{(2)} = -\beta\left(\frac{\partial E}{\partial O^{(5)}}\right)\left(\frac{\partial O^{(5)}}{\partial O_j^{(3)}}\right)\left(\frac{\partial O_j^{(3)}}{\partial O_{ij}^{(2)}}\right)\left(\frac{\partial O_{ij}^{(2)}}{\partial \sigma_{ij}^{(2)}}\right)$$

$$= \beta(\eta_d - O^{(5)})\left(\frac{(m_j^{(4)} - O^{(5)})\sigma_j^{(4)}}{\sum_j \sigma_j^{(4)} O_j^{(3)}}\right)$$

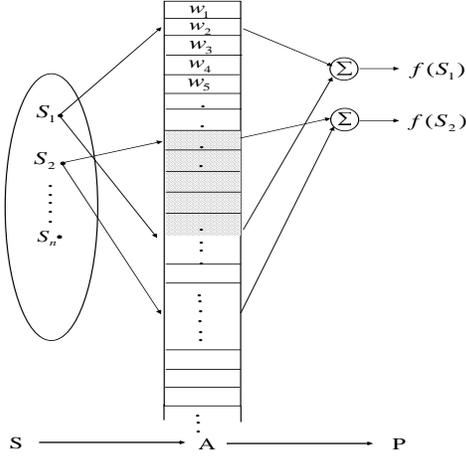$$\times \left(\frac{2O_{ij}^{(2)}(x_i - m_{ij}^{(2)})^2}{(\sigma_{ij}^{(2)})^3}\right) \qquad (27)$$

Fig. 5. Basic CMAC mappings.

B. Implementation of the CMAC

The CMAC is a trainable linear network pattern classifier that emulates the behavior of human beings in dealing with stimuli and responses [18], [19], [36]. The CMAC computes control functions by referring to a table rather than by solving analytic equations. Function values are stored in a distributed fashion such that the value of a function at any point in input space is derived by summing the contents over a number of memory locations. A unique feature of the CMAC is a mapping algorithm which converts the distance between input vectors into the degree of overlap between sets of data where the function values are stored. Thus, the CMAC serves our purposes well, because mappings can be provided to generate proper discharge efficiency surfaces for the whole battery dynamic state space based on the finite sets of experimental data used as training patterns. The basis concept behind the CMAC can be represented by a pair of mappings shown in Figure 5

$$f : \mathbf{S} \rightarrow \mathbf{A} \tag{28}$$

$$g : \mathbf{A} \rightarrow \mathbf{P} \tag{29}$$

where $\mathbf{S}$ represents the set of input vectors, $\mathbf{A}$ represents the set of association cell vectors, and $\mathbf{P}$ represents the set of response output vectors. The first mapping maps the input data onto a finite set of intermediate states called association cells. The mapping is generally a fixed relation since it is a process of indexing the input data. The number of units of $\mathbf{A}$ that become excited in response to both different inputs S_i and S_j decreases monotonically as the similarity between S_i and S_j decreases. This arrangement produces generalization between nearby input vectors and no generalization between distant input vectors. The second mapping depends on the values of weights, assigned to every association cell, which will be modified during the training stage. These weights can be adjusted by the difference between the desired output and the produced output. This mapping then sums up the weights

attached to the active association cells to produce the output $\mathbf{P}$.

We will explain how to store the weights, $\mathcal{F}$'s, of FNN which represent a battery discharge surface, corresponding to a certain value of SOH into the CMAC. The input vector s is used for indexing the FNN weight, described as

$$s = p \tag{30}$$

where p stands for a value of battery SOH.

Via the mappings of the network, the input should correspond to an output response consisting of a desired FNN weight. Thus, we need to find appropriate CMAC weights to attach to the active association cells in $\mathbf{A}$ by utilizing the FNN weight for a value of SOH as training patterns. However, when the training patterns are stored into the network, the FNN weight generated by the CMAC may be different from the desired FNN weight because of memory overlapping in the association cells in the first mapping from the input vector to the association cells. Thus, a learning process is needed to modify the CMAC weights through an updating function using the difference between the desired FNN weight and that generated by the CMAC. The updating function is as follows:

$$w_{k+1} = w_k + \beta \left(\frac{\mathcal{F}_d - \mathcal{F}}{c} \right) \tag{31}$$

where k denotes the number of learning iterations, β is learning rate, c is the number of weights contributing to the output, w_{k+1} and w_k stand for the CMAC weights before and after the kth learning iteration, respectively, and $\mathcal{F}_d$ and $\mathcal{F}$ stand for desired and actual FNN weights, respectively. The learning process will terminate when the difference between $\mathcal{F}_d$ and $\mathcal{F}$ is within a certain tolerance ε_c.

C. Learning Process for Generating Discharge Efficiency Surfaces

The process of learning for generating discharge efficiency surfaces can be divided into two stages. In the first stage, the FNN in the proposed learning structure is used to learn to generate the desired discharge efficiency η_d obtained from experiments and indexed by the corresponding discharge rate I^{dis} and residual capacity C^{res}. The weight of the FNN, $\mathcal{F}$, provided by the CMAC starts to be updated according to the error described in Eq.(19). In the second stage, the CMAC will use the updated FNN weight $\mathcal{F}_d$ obtained in the first stage as a training pattern to update its own weights. We repeat stages one and two until errors decrease under a threshold. This learning process is shown in Figure 6.

IV. Experiment

The experimental equipment is shown in Figure 7. In Figure 7, batteries under test were sealed lead-acid batteries with a rating of 12V open-circuit voltage and 50Ah capacity, manufactured by Long Battery Co., Taiwan. The battery testing system used to perform testing cycles was manufactured by DIGATRON Co., Germany, and the charge and discharge currents provided ranged between 0A and 200A. It is programmable by downloading charge or discharge patterns from
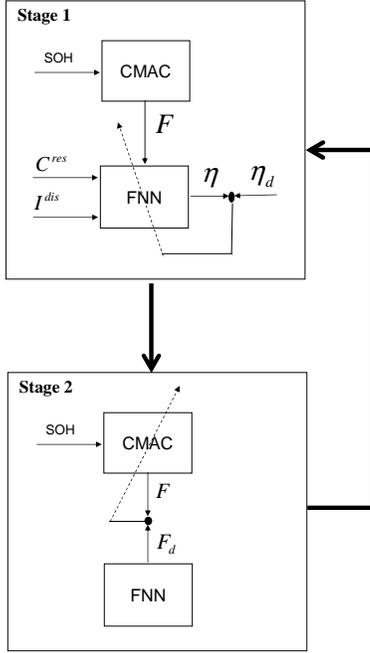
Fig. 6. Conceptual organization of discharge efficiency surfaces learning process.
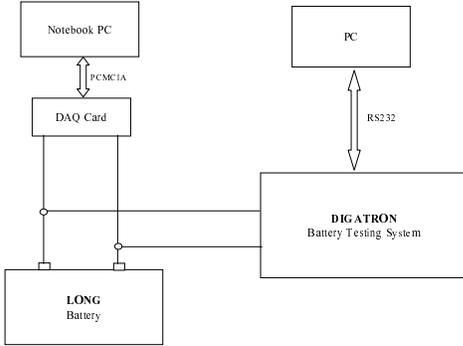

Fig. 7. Experimental equipment

a PC. A notebook PC with the EMS Windows application software was used to estimate the SOC of batteries, and the EMS software is implemented by using Borland C++ Builder. We used 98 data sets of constant current discharge to generate our training patterns. Every data set included a time history of battery voltage, discharge current, and capacity. The corresponding discharge efficiency and SOH can be calculated according to Eqs.(1) and (13). All of these data were combined into training patterns for the initial weights of learning of our proposed system. Because the initial weights were obtained from static(constant current) discharge process, we need to modify the initial weights dynamically so as to increase the precision of estimation of our system.
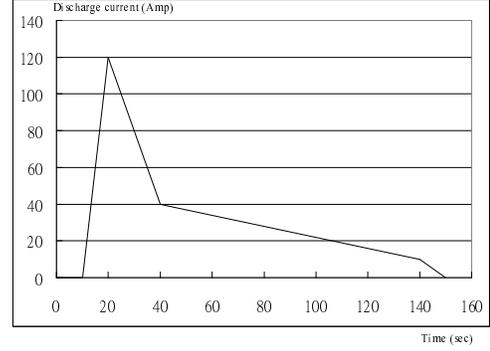

Fig. 8. Discharge pattern used to modify initial weights of the proposed learning system.

A. Dynamic Modification

We used the discharge pattern shown in Figure 8 to modify the initial weights dynamically. This is exploited in order to increase the precision of estimation of our system when it is used under real discharge conditions. This discharge pattern used to modify initial weights is determined intuitively: it seems to accelerate a scooter first, then maintain the speed of the scooter at a certain constant and stop the scooter. Although different modification discharge patterns should result in different precisions of the SOC estimation, we think that if modification discharge patterns cover the range of all possible discharge current values, the actual prediction errors should be small.

In each modification process, the battery was charged when its steady state voltage is 13.2V, and discharged according to the pattern repeatedly until the battery voltage dropped below the cutoff voltage, 10.25V, and all relative data is recorded. By using the recorded data and an iterative computation, we can obtain the time history of desired discharge efficiency for dynamic conditions. An iterative computation is used to minimize a cost function defined as

$$E_N(i) = \frac{1}{2}\left[C_0\left(\frac{C_a}{C_b}\right) - T\left(\sum_{k=1}^{N}\frac{I_k}{\eta_k(i)}\right)\right]^2 \quad (32)$$

Thus, the iterative process can be expressed as

$$
\begin{aligned}
\eta_k^d(i+1) &= \eta_k^d(i) - \gamma\frac{\partial E_N}{\partial \eta_k} \\
&= \eta_k^d(i) + \gamma\sqrt{2E_N(i)}T\left(\frac{I_k}{\eta_k^2(i)}\right) \quad (33)
\end{aligned}
$$

with

$$\eta_k^d(0) = \eta_k \quad (34)$$

where C_a is the discharged capacity when estimated SOC is zero, and C_b is the discharged capacity when battery voltage is V_{cutoff}; $\eta_k^d(i)$ stands for the desired discharge efficiency at the kth discrete time and the ith iterative step, γ is step size, and T is sampling time.

The obtained desired discharge efficiency will be used to train our EMS, and this modification process is performed when values of SOH are 40%, 70%, and 100%.

(a)   (b)

(c)   (d)
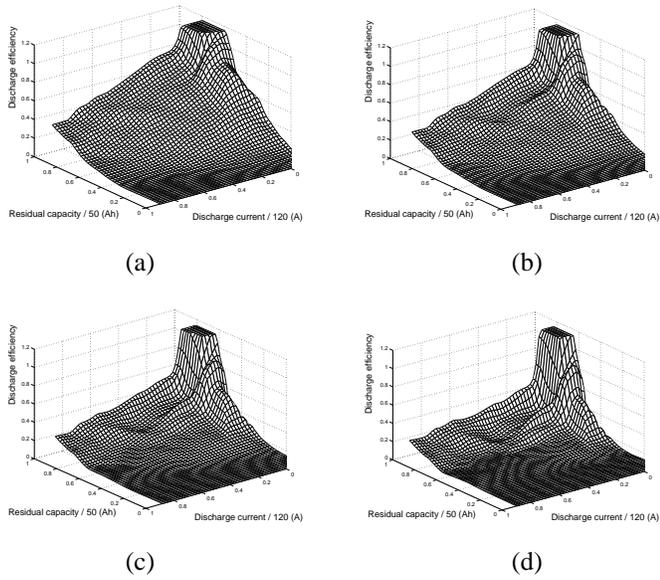
Fig. 9. Discharge efficiency surfaces generated by EMS with initial weights at (a) $SOH = 100\%$. (b) $SOH = 80\%$. (c) $SOH = 60\%$. (d) $SOH = 40\%$.
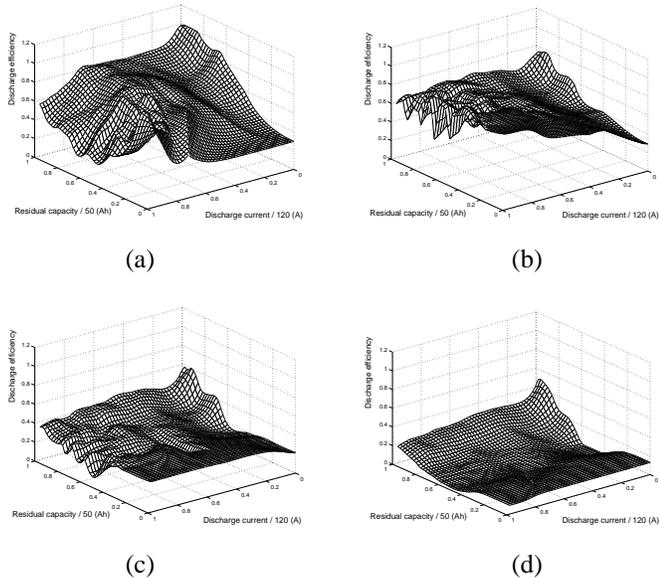

(a)   (b)

(c)   (d)

Fig. 10. Discharge efficiency surfaces generated by modified EMS at (a) $SOH = 100\%$. (b) $SOH = 80\%$. (c) $SOH = 60\%$. (d) $SOH = 40\%$.

A restriction to perform the modification process is that we need to collect relative data of a battery from the state when it is of full capacity to the state when it is empty, because we can only make sure that when battery voltage is lower than V_{cutoff}, its SOC is zero. That is why the learning process of our EMS is very difficult to implement as an on-line process, because users always recharge their batteries before they run out of the capacities of their batteries.
Figure 9 and Figure 10 show the discharge efficiency surfaces before and after the modifications dynamically.

B. Testing for EMS

We use four urban drive cycles shown in Figure 11 to test the proposed EMS, and these drive cycles are CNS-D3029 of Taiwan, EPA75 of the United States, ECE15 of the EEC nations, and M10-15 of Japan. By using these drive cycles and dynamic data of the electric scooter we used, we can transfer these drive cycles into the corresponding power consumption cycles shown in Figure 12. It is convenient to do experiments by using power consumption data in our laboratory. In each
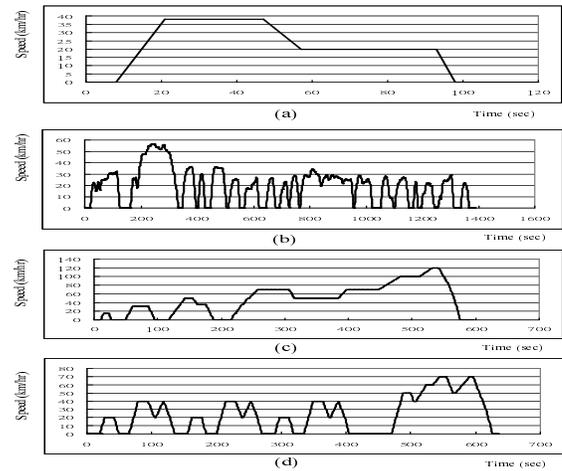

Fig. 11. Drive cycles (a) CNS-D3029 of Taiwan (b) EPA75 of the United States (c) ECE15 of the EEC nations (d) M10-15 of Japan
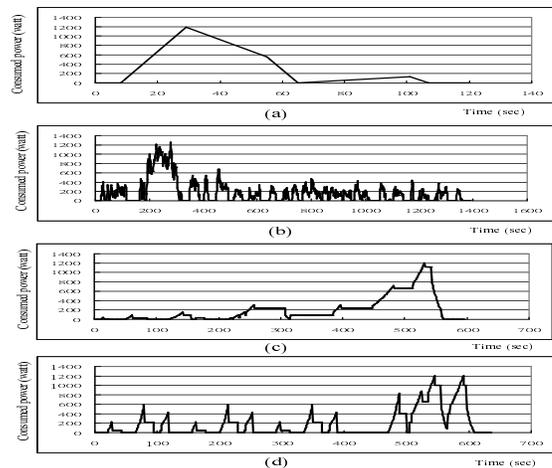

Fig. 12. Power consumption cycles for testing EMS (a) CNS-D3029 of Taiwan (b) EPA75 of the United States (c) ECE15 of the EEC nations (d) M10-15 of Japan

test, the battery is also charged when its steady state voltage is 13.2V, and discharged according to these drive cycles repeatedly until the battery voltage drops under the cutoff

voltage, 10.25V. The prediction error ϵ is defined as

$$\epsilon = \left| \frac{100(C_b - C_a)}{C_b} \right| \quad (\%) \tag{35}$$

Figure 13 shows results of drive cycle testing of CNS-D3029. In Figure 13(a), the SOC prediction error of EMS is 5.423 %, and the SOC trajectory is smoothly decreasing, so users can easily determine the timing when the batteries should be recharged. Figure 13(b) shows the discharge efficiency generated by our proposed EMS. In Figure 13(c), we can see the battery voltage varies fast in dynamic operations, so it is not suitable to be used to estimate battery SOC. Figure 13(d) shows the discharged capacity recorded by using only ampere hour accumulation, and that the prediction error can reach 50%. In addition, to show a better performance can be achieved by the proposed scheme, we test this system by performing real-time experiment of four drive cycle patterns. The test results of another three drive cycles (EPA75, ECE15, and M10-15) are shown in Figure 14, Figure 15, and Figure 16, respectively. The SOC prediction errors of four drive cycles testing at different SOH values, 82.5%, 66%, and 43.2%, are shown in Figure 17. From results of the experiment, we have a good average prediction error under 6% in the whole battery life cycle. In practice, it is recommended that a battery should be replaced when its SOH is below 70%, so our proposed learning system is quite enough to deal with the effect of decreasing of SOH.

V. CONCLUSION

We have proposed an SOC learning system for improving the performance of present SOC measurement used in electric scooters. Only static data obtained from experiments is used as training patterns for the proposed system, and this system is shown to be able to estimate the dynamic SOC accurately. The aging effect is also considered in the proposed system, because the aging problem is very obvious in the electric scooter or EV application. Moreover, making use of the dynamic SOC learned by this system, we can obtain a corresponding allowable traveling distance. A safe speed can be calculated and provided to the rider, such that the rider can control the speed of the scooter within a proper range to ensure that the scooter can arrive at the destination safely [37]. Experimental results have demonstrated the effectiveness of the proposed SOC learning system.

As future work, we will consider the effect of temperature, and introduce this factor into our learning system for developing a more precise SOC estimating system.
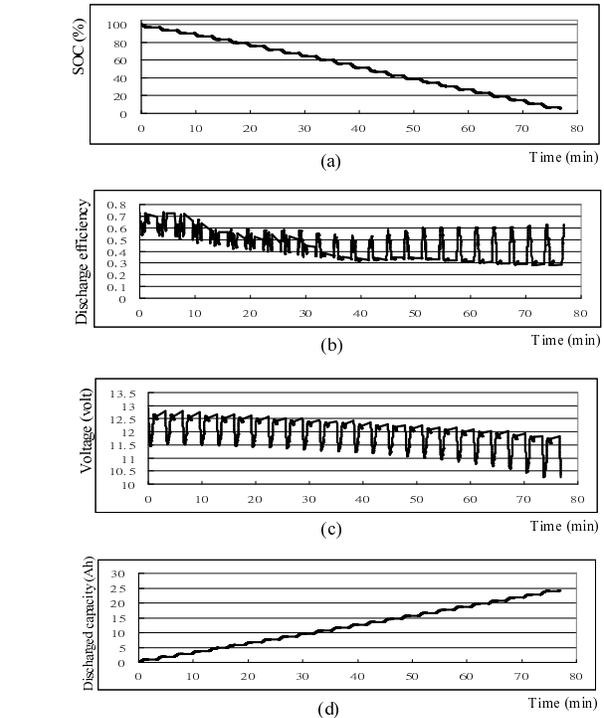

Fig. 13. Drive cycle testing of CNS-D3029 : (a)Battery SOC predicted by EMS. (b)Discharge efficiency data generated by EMS. (c)Battery voltage (d)Discharged capacity

REFERENCES

[1] P. B. Patil, "Fuel gauges for electric vehicles," in *Proc. 17th IECEC, IEEE, New York*, 1982.
[2] N. M. Banes et al., *The Sealed Lead Battery Handbook*. General Electric Co., 1982.
[3] P. F. Eugene and N. Y. Brewsler, "Battery of charge metering method and apparatus," *U.S. Patent*, no. 4560937, 1985.
[4] Simmonds et al., "Device for indicating the residual capacity of secondary cells," *U.S. Patent*, no. 5479085, 1995.
[5] Kopmann, "Method of and apparatus for monitoring the state of a rechargeable battery," *U.S. Patent*, no. 5518835, 1996.
[6] Kozaki et al., "Remaining battery capacity meter and method for computing remaining capacity," *U.S. Patent*, no. 5691078, 1997.
[7] A. S. Clegg and C. England, "Battery monitor which indicates remaining capacity by continuously monitoring instantaneous power consumption relative by expected hyperbolic discharge rates," *U.S. Patent*, no. 5394089, 1997.
[8] E. Karden, P. Mauracher, and A. Lohner, "Battery management system for energy-efficient battery operation : Strategy and practical exprience," *EVS 13*, vol. 2, p. 91, 1996.
[9] S. K. Song and K. H. Kim, "A dynamic state of charge model for electric vehicle batteries," *EVS 12*, vol. 2, p. 519, 1994.
[10] Q. Guogang, L. Jianming, and J. Hang, "A new battery state of charge indicator for electric vehicles," *EVS 13*, vol. 2, p. 631, 1996.
[11] M. Kitagawa, "Development of battery state of charge indicator for electric vehicles," *EVS 12*, vol. 1, p. 293, 1994.
[12] O. Caumont, P. L. Moigne, C. Rombaut, X. Muneret, and P. Lenain, "Energy gauge for lead-acid batteries in electric vehicles," *IEEE Trans. on Energy Conversion*, vol. 15, no. 3, pp. 354–360, 2000.
[13] F. Huet, "A review of impedance measurements for determination of the state of charge or state of health of secondary batteries," *J. Power Sources*, vol. 70, pp. 59–69, 1998.
[14] T. Yamazaki, K. Sakurai, and K. I. Muramoto, "Estimation of the residual capacity of sealed lead-acid batteries by neural network," in *Int. Telecommunications Energy Conf.*, vol. 20, 1998, pp. 210–214.
[15] A. J. Salkind, C. Fennie, P. Singh, T. Atwater, and D. E. Reisner, "Determination of state of charge and state of health of batteries by fuzzy logic methodology," *J. Power Sources*, vol. 80, no. 1–2, pp. 293–300, 1999.
[16] W. X. Shen, C. C. Chan, E. W. C. Lo, and K. T. Chau, "Adaptive neuro-fuzzy modeling of battery residual capacity for electric vehicles," *IEEE Trans. on industrial electronics*, vol. 49, no. 3, pp. 677–684, 2002.
[17] C. T. Lin and C. S. G. Lee, "Reinforcement structure/parameter learning
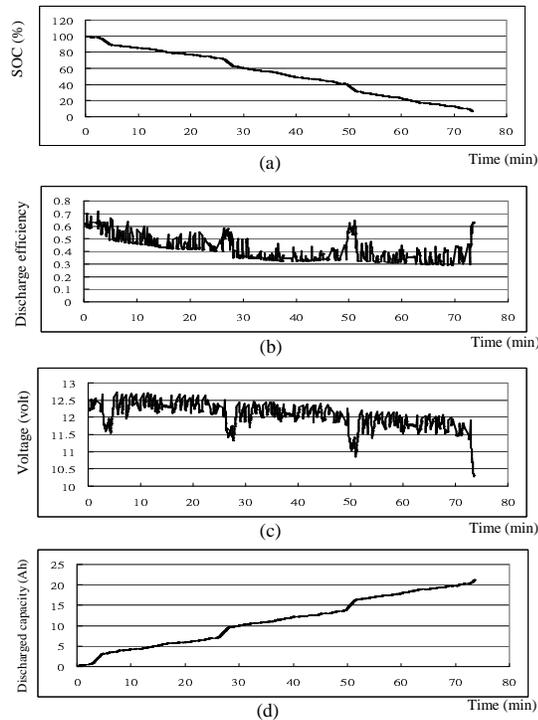
Fig. 14. Drive cycle testing of EPA75 : (a)Battery SOC predicted by EMS. (b)Discharge efficiency data generated by EMS. (c)Battery voltage (d)Discharged capacity
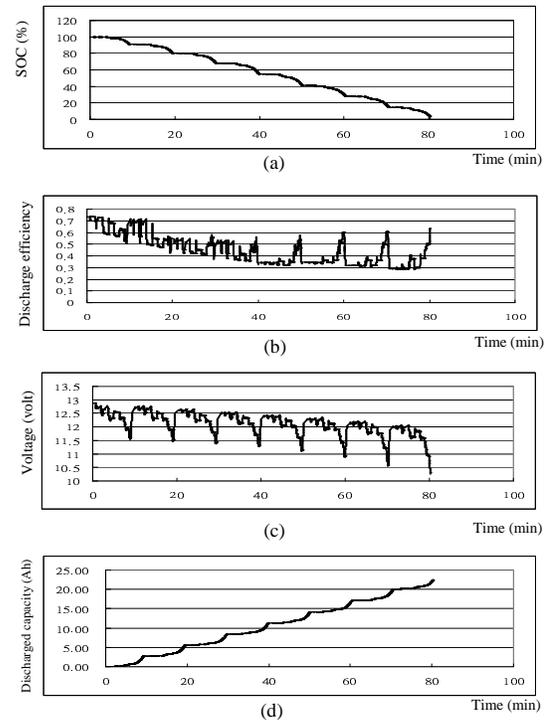

Fig. 15. Drive cycle testing of ECE15 : (a)Battery SOC predicted by EMS. (b)Discharge efficiency data generated by EMS. (c)Battery voltage (d)Discharged capacity

for neural-network-based fuzzy logic control systems," *IEEE Trans. Fuzzy Syst.*, vol. 2, no. 1, pp. 46–63, 1994.

[18] J. S. Albus, "A new approach to manipulator control: the cerebellar model articulation controller (CMAC)," *ASME J. Dynamic Syst., Measurement, Contr.*, vol. 97, no. 3, pp. 220–227, 1975.

[19] ——, "Data storage in the cerebellar model articulation controller (CMAC)," *ASME J. Dynamic Syst., Measurement, Contr.*, vol. 97, no. 3, pp. 228–233, 1975.

[20] C. S. C. Bose, D. Wilkins, S. McCluer, and M. J. Model, "Lessons learned in using ohmic techniques for battery monitoring," in *The Sixteenth Annual Battery Conference on Applications and Advances*, 2001, pp. 99–104.

[21] V. Spath, A. Jossen, H. Doring, and J. Garche, "The detection of the state of health of lead-acid batteries," in *Int. Telecommunications Energy Conf.*, Oct. 1997, pp. 681–686.

[22] C. Alippi, A. Ferrero, and V. Piuri, "Artificial intelligence for instruments and measurement applications," *IEEE Instrumentation & Measurement Magazine*, vol. 12, pp. 9–17, 1998.

[23] P. D. Wasserman, *Neural Computing–Theory and Practice*. NewYork: Van Nostrand Reinhold, 1989.

[24] H. J. Zimmermann, *Fuzzy Set Theory and Its Applications*, 3rd ed. Norwell: MA: Kluwer, 1996.

[25] J. Hertz, A. Korgh, and R. G. Palmer, *Introduction To The Theory Of Neural Computation*. Addison-Wesley, 1991.

[26] B. Kosko, *Neural Networks And Fuzzy Systems*. Prentice-Hall, 1992.

[27] J. M. Zurada, *Introduction To Artificial Neural Systems*. West Publishing Co., 1992.

[28] D. Dubois and H. Prade, *Fuzzy Sets and Systems: Theory and Application*. New York: Academic Press, 1980.

[29] C. C. Lee, "Fuzzy logic in control systems: fuzzy logic controller–Part I, II," *IEEE Trans. Syst., Man, and Cybern.*, vol. 20, no. 2, pp. 404–435, 1990.

[30] D. Gorinevsky, D. E. Torfs, and A. A. Goldenberg, "Learning approximation of feedforward control dependence on the task parameters with application to direct-drive manipulator tracking," *IEEE Trans. Rob. and Aut.*, vol. 13, no. 4, pp. 567–581, 1997.

[31] T. D. Sanger, "Neural network learning control of robot manipulators using gradually increasing task difficulty," *IEEE Trans. Robot. Automat.*, vol. 10, no. 3, pp. 323–333, 1994.

[32] T. Shibata and T. Fukuda, "Hierarchical intelligent control for robotic motion," *IEEE Trans. Neural Networks*, vol. 5, pp. 823–832, 1994.

[33] K. Y. Young and S. J. Shiah, "An approach to enlarge learning space coverage for robot learning control," *IEEE Trans. Fuzzy Syst.*, vol. 5, no. 4, pp. 511–522, 1997.

[34] J. H. Aylor, A. Thieme, and B. W. Johnson, "A battery state-of-charge indicator for electric wheelchairs," *IEEE Trans. on industrial electronics*, vol. 39, no. 5, pp. 398–409, 1992.

[35] H. R. Berenji and P. Khedkar, "Learning and tuning fuzzy logic controllers through reinforcements," *IEEE Trans. Neural Networks*, vol. 3, no. 5, pp. 724–740, 1992.

[36] W. T. Miller, F. H. Glanz, and L. G. Kraft, "Application of a general learning algorithm to the control of robotic manipulators," *Int. J. Robot. Res.*, vol. 6, no. 2, pp. 84–98, 1987.

[37] D. T. Lee, S. J. Shiah, C. M. Lee, and C. H. Wu, "Intelligent control of electric scooters," in *The 5th IASTED International Conference on Intelligent Systems and Control*, Tsukuba, Japan, Oct., 2002, pp. 63–69.
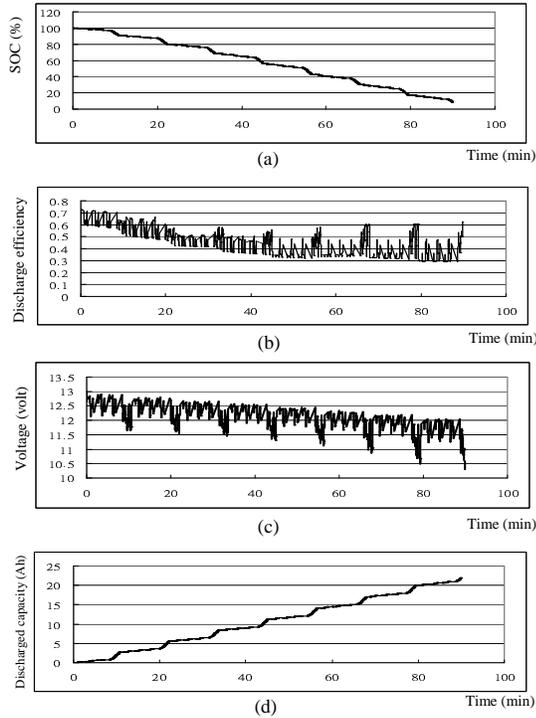
Fig. 16. Drive cycle testing of M10-15 : (a)Battery SOC predicted by EMS. (b)Discharge efficiency data generated by EMS. (c)Battery voltage (d)Discharged capacity

SOC prediction error $\diagdown$ SOH Drive cycle	82.5 %	66.0 %	43.2 %
CNS-D3029, Taiwan	5.423 %	5.418 %	4.774 %
EPA75, USA	6.445 %	0.627 %	5.357 %
ECE15, EEC nations	3.387 %	3.893 %	4.552 %
M10-15, Japan	8.462 %	7.668 %	8.133 %

Fig. 17. Errors of four drive cycles testing

Shaw-Ji Shiah was born in Taoyuan, Taiwan, 1969. He received the B.S. degree in electrical engineering from the Chung Yuan Christian University, Chung-Li, Taiwan, in 1992, and the M.S. and Ph.D. degree in Electrical and Control engineering from the National Chiao-Tung University, Hsinchu, Taiwan, in 1994 and 1999, respectively. From 1999 to 2003, he was a Postdoctoral Fellow of the Institute of Information Science, Academia Sinica. He is currently a senior engineer of the EE Department, Research and Development Division, Optical Storage BU, Quanta Storage Inc. His research interests are in robotics, neural networks, fuzzy systems, intelligent control, battery SOC estimation, and optical drive servo control.

Der-Tsai Lee received his B.S. degree in Electrical Engineering from the National Taiwan University in 1971, and the M.S. and Ph. D. degrees in Computer Science from the University of Illinois at Urbana-Champaign in 1976 and 1978 respectively.

Dr. Lee has been with the Institute of Information Science, Academia Sinica, Taiwan, where he is a Distinguished Research Fellow and Director since July 1, 1998. Prior to joining the Institute, he was a Professor of the Department of Electrical and Computer Engineering, Northwestern University, where he has worked since 1978.

His research interests include design and analysis of algorithms, computational geometry, VLSI layout, web-based computing, algorithm visualization, software security, bio-informatics, compliant controller for active suspension and vibration control, digital libraries and advanced IT for intelligent transportation systems.

He has published over 120 technical articles in scientific journals and conference proceedings, and he also holds three U.S. patents, and one R.O.C. patent. He is Editor of Algorithmica, Computational Geometry: Theory and Applications, International Journal of Computational Geometry and Applications, Journal of Information Science and Engineering, and Series Editor of Lecture Notes Series on Computing for World Scientific Publishing Co., Singapore.

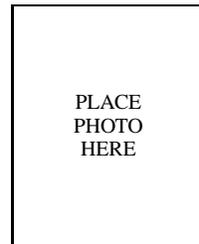He is Fellow of IEEE, Fellow of ACM, President of IICM and Academician of Academia Sinica.

Chien-Ming Lee received the B.S. degree from the College of Electrical Engineering and Information science, National Taiwan Ocean University in 1997. He received the M.S. degree in Electrical Engineering from the National Taiwan University in 1999. From 1999 to 2004, he worked as a research assistant in the Institute of Information Science, Academia Sinica, for his military service. From February 2004 on, he worked at ASUSTek Computer Inc. as a senior engineer of Handset device driver for 21 months before he joined the Media TeK Inc. as a GSM/GPRS Firmware Engineer in November 2005. His earlier research interest while working in National Taiwan University was focused on real-time and fault-tolerant enhancements to CORBA. His main contribution while working at IIS was to develop embedded system software for intelligent electrical motorcycle control system.
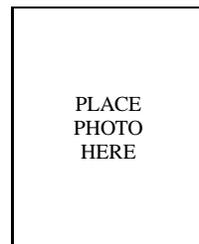
Ying-Chung Wang was born in Taipei, Taiwan, R.O.C. in 1972. He received the B.S. degree in Electronic Engineering and M.S. degree in Mechatronic Engineering from the Huafan University, Taipei, Taiwan, R.O.C. in 1996 and 1998, respectively, and the Ph.D. degree in Electrical and Control Engineering from the National Chiao-Tung University in 2003. He is currently a Postdoctoral Fellow of the Institute of Information Science, Academia Sinica. His main research interests include fuzzy systems, neural networks, fuzzy neural systems, iterative learning control, nonlinear control, adaptive control, and robot control.