Supplement of “Modeling and Real-time Scheduling of Large-Scale Batteries for Maximizing Performance”

Eugene Kim and Kang G. Shin
Department of Electrical Engineering and Computer Science
The University of Michigan – Ann Arbor, U.S.A.
{kimsun,kgshin}@umich.edu

Jinkyu Lee
Department of Computer Science and Engineering
Sungkyunkwan University (SKKU), Republic of Korea
jinkyu.lee@skku.edu

I. Algorithm for in a cycle

To minimize energy consumption of auxiliary devices, we must consider both energy consumption of batteries and UCs. Therefore, we compare the power consumption at each control interval so as to search the set of discharge current of battery for the minimum energy consumption by using the battery voltage model as shown in Algorithm 1. In a vehicle system, energy consumption occurs in a battery and auxiliary devices. This energy consumption can be estimated by integrating power consumptions ($E_{\text{total}} = \int_0^\tau P_{\text{total}}(t)dt$), and Sec. III shows how to calculate the power requirements. To maximize battery operation-time (t_{op}), we have to remove as many voltage drop periods as possible. So, we iteratively find the voltage drop period and compensate the voltage by regulating battery temperature as seen in Algorithm 2.

Algorithm 1 Algorithm for the target current calculation

procedure ICAL($n, P_{\text{req}}, I^*_{\text{bat}}, T_{\text{bat}}$)
  for $i == 0; i < n; i++$ **do**
    $I_{\text{tot}} = \frac{P_{\text{req}}}{V_{\text{o}}}$;
    $I_{\text{UC}}[i] = 0$;
    $I_{\text{bat}}[i] = I_{\text{tot}}$;
    while $I_{\text{bat}}[i] > I^*_{\text{bat}}$ & $E(I_{\text{bat}}[i], I_{\text{UC}}[i], T_{\text{bat}}[i]) > E(I_{\text{bat}}[i] - 1, I_{\text{UC}}[i] + 1, T_{\text{bat}}[i])$ **do**
      $j \leftarrow$ FindPrechargePeriod(); //
      if j == NULL **then**
        Break;
      else
        $I_{\text{UC}}[i] \leftarrow I_{\text{UC}}[i] + 1$;
        $I_{\text{bat}}[i] \leftarrow I_{\text{bat}}[i] - 1$;
        $I_{\text{UC}}[j] \leftarrow I_{\text{UC}}[j] - 1$; // Precharge UC at time j
        $I_{\text{bat}}[j] \leftarrow I_{\text{bat}}[j] + 1$; // Precharge at time j
      end if
    end while
  end for
return $[I_{\text{bat}}, I_{\text{UC}}]$;
end procedure

Algorithm 2 provides offline temperature scheduling for a given power requirement, and an insight on how to schedule temperature in real-time. According to the scheduling results, battery temperature should be increased when it cannot supply the required power, which is estimated by using the battery output voltage model. We determine the target temperature based on the output voltage model. It makes battery operate first at a low temperature because initially battery SoC is enough to supply the required power to the electric load. Then, we increase battery temperature as the SoC goes down while regulating battery temperature based on the power requirement prediction. That is, when a large power requirement is predicted, we increase temperature to decrease battery internal resistance. After determining the target discharge current, we calculate the temperature scheduling period by considering its reliability in the worst case such that maximum current is discharged from battery. That is, we have to consider the possibility of reaching an upper bound of temperature when maximum available current is discharged based on thermal transfer function. Note that temperature scheduling period would be longer than current scheduling period, since battery temperature changes more slowly than battery current.

Algorithm 2 Algorithm for the target temperature

procedure TCAL($n, T^*_{\text{bat}}, I_{\text{bat}}$)
  for $i == 0; i < n; i++$ **do**
    $T_{\text{bat}}[i] = T^*_{\text{bat}}$;
  end for
  for $i == 0; i < n; i++$ **do**
    while $V_{\text{bat}}(I_{\text{bat}}[i], T_{\text{bat}}[i]) < V_{\text{lb}}$ **do**
      $j \leftarrow$ FindCoolingPeriod();
      if j == NULL **then**
        Break;
      else
        $T_{\text{bat}}[i] \leftarrow T_{\text{bat}}[i] + 1$;
        $T_{\text{bat}}[j] \leftarrow T_{\text{bat}}[j] - 1$; // Cooling at time j
      end if
    end while
  end for
return $[T_{\text{bat}}]$;
end procedure

Algorithm 3 Algorithm for real-time temperature scheduling

procedure PROC4($I^*_{\text{bat}}, T^*_{\text{bat}}$)
  while Driving **do**
    while $V^*_{\text{o}} + V_{\text{margin}} < V_{\text{o}}(I^*_{\text{bat}}, T^*_{\text{bat}})$ **do**
      $T^*_{\text{bat}}--$;
    end while
    while $V^*_{\text{o}} + V_{\text{margin}} > V_{\text{o}}(I^*_{\text{bat}}, T^*_{\text{bat}})$ **do**
      T^*_{bat}++;
    end while
    Sleep(p_{T});
  end while
end procedure

II. Battery degradation

Fig. 1 shows battery degradation at some temperature. First, initial degradation level was 8.32 $m\Omega$. After 37 discharge/charge cycles, it became 9.02 $m\Omega$. That is, average degradation was 18.9 $\mu\Omega$ for a cycle. Therefore, we assumed its change has a negligible influence on battery capacity within a cycle in the paper.

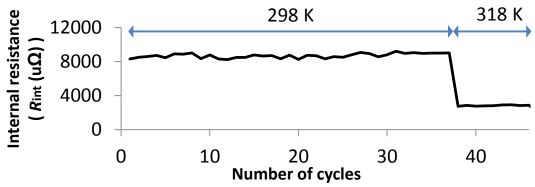

Fig. 1. An example of battery degradation: During the first 37 cycles, the battery operates at $298K$, 1 A of the average discharged current ($\frac{1}{t_o^k}\int_0^{t_o^k} I_{\text{bat}}(t)dt$ ($\frac{1}{t_o^k}Q_d^k$)), and 1.5A of the average amounf of battery discharge/charge current ($\frac{1}{t_o^k}\int_0^{t_o^k} |I_{\text{bat}}(t)|dt$), and the battery operates at $318K$ with the same condition after 38 cycle.

III. POWER REQUIREMENT FOR BMS IN EVS

To supply required power (P_{req}) for operating electric motors of vehicles, energy should be extracted from energy storages and auxiliary devices while inducing power loss (P_{total}) as follows:

$$P_{\text{total}}(t) = P_{\text{battery}}(t) + P_{\text{UC}}(t) + P_{\text{converter}}(t) + P_{\text{cooler}}(t).$$

Supplying power from batteries to electric load induces voltage drop with power loss on battery $P_{\text{battery}}(t)$ due to battery's non-idealistic properties.

$$P_{\text{bat}}(t) = (V_{\text{oc}}(t) - V_{\text{o}}(t)) \cdot I_{\text{bat}}(t),$$

where $V_{\text{oc}}(t)$ is open circuit voltage, $V_{\text{o}}(t)$ is output voltage, and $I_{\text{bat}}(t)$ is discharged current. Also, auxiliary devices for mitigating discharging stress dissipate energy in path from UC ($P_{\text{UC}}(t)$) and converter ($P_{\text{conv}}(t)$) to electric load.

$$P_{\text{UC}}(t) = I_{\text{UC}}(t)^2 \cdot R_{\text{UC}},$$
$$P_{\text{converter}}(t) = I_{\text{UC}}(t)^2 \cdot R_{\text{converter}}.$$

In addition, cooler for regulating batteries' temperature causes power dissipation ($P_{\text{cooler}}(t)$).

$$P_{\text{cooler}}(t) = f_{\text{cooler}}(\Delta T(t)),$$

where $\Delta T(t)$ is target battery temperature changes for thermal management.

IV. POWER REQUIREMENT FOR EVALUATIONS

We extracted realistic power requirements for real driving data from "The US Environmental Protection Agency (EPA)" and "California Air Resources Board" by using the vehicle simulator as shown in Fig. 2.

V. EXTRACTED GENES

We extracted the most effective genes for building battery models via the algorithm in the paper. Fig. 3 and 4 represents the extracted genes for predicting battery output voltage (V_{o}) and battery degradation rate (ΔR_{int}) in the order of occurrence, respectively. In Fig. 3, x1 is battery temperature ($T_{\text{bat}}(t)$), x2 the rate change of battery current ($\frac{dI_{\text{bat}}}{dt}(t)$), x3 is a battery discharge/charge current ($I_{\text{bat}}(t)$), x4 the amount of discharged charge ($Q_{\text{d}}(t)$), and x5 the rate change of battery voltage ($\frac{dV_{\text{bat}}}{dt}(t)$), respectively. In Fig. 4, x1 is a previous degradation rate (R_{int}), x2 accumulative temperature exposure ($\int_0^{t_o^k} T_{\text{bat}}(t)dt$), x3 the amount of discharged charge (Q_d^k), and x4 the average amount of discharge/charge current ($\frac{1}{t_o^k}\int_0^{t_o^k} |I_{\text{bat}}(t)|dt$), respectively.

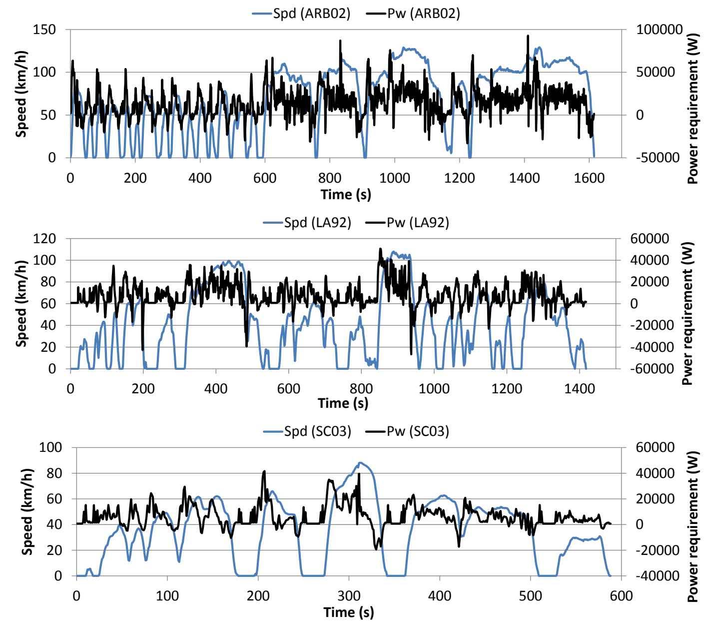

Fig. 2. Power requirements for driving patterns (ARB02, LA92, SC03)

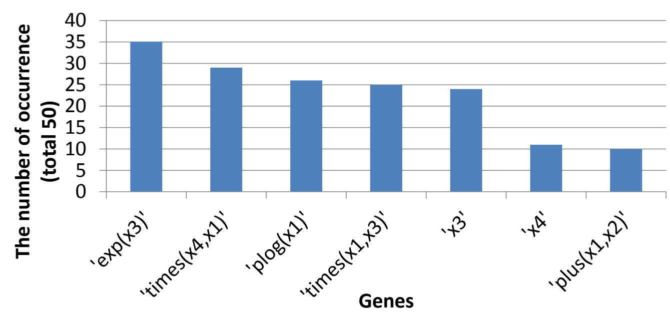

Fig. 3. The most commonly extracted genes for battery voltage model.

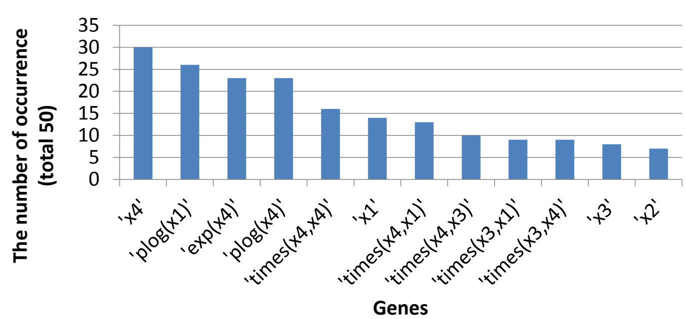

Fig. 4. The most commonly extracted genes for battery degradation model.