

온라인 전기자동차의 대용량 배터리를 위한 전압-온도를 이용한 패시브 셀 밸런싱 에이전트

오근현*, 김종우, 서동관, 김종돈**

*KAIST IT 융합연구소

**KAIST 온라인전기자동차사업단

e-mail : {ocworld, galamwhi, challenseo, kimjd921}@kaist.ac.kr

A Passive Cell Balancing Agent by using Voltage and Temperature for Large Batteries in Online Electric Vehicles

Keunhyun Oh*, Jongwoo Kim, Dongkwan Seo, Jongdon Kim**

*KAIST Institute for IT Convergence, Korea Institute of Science and Technology(KAIST)

**On-Line Electric Vehicle, Korea Institute of Science and Technology(KAIST)

요 약

친환경적 전기자동차에 대한 관심이 커짐에 따라 전력의 효율적인 저장과 관리를 위한 대용량 배터리에 대한 연구가 활발히 진행 중이다. 배터리 용량의 극대화, 안정성, 그리고 수명 등을 위한 다양한 셀 밸런싱 방법들이 소개되고 있다. 기존 패시브 셀 밸런싱 방법은 하드웨어의 기능을 수행하는 것에 초점을 맞추어 배터리 상태에 따라 다양한 반응을 정의하기 어려움으로 효과적인 운용에 한계가 있다. 본 논문에서는 소프트웨어 에이전트 모델을 적용한 배터리 셀 밸런싱 시스템을 제안하며 에이전트가 현재 전압과 온도를 고려한 셀 밸런싱을 수행함으로 고속 충전시에 발생하는 셀간 전압차를 최소화 한다. 또한 충전의 신속성과 셀 전압 안정성을 유지한다. 현재 연구 개발 중인 카이스트 온라인 전기자동차에서 사용 예정인 배터리 관리시스템 기반 시뮬레이션을 수행함으로 제안하는 방법의 유용성을 입증하였다.

1. 서론

화석 연료를 사용한 엔진기반의 자동차에서 발생하는 환경과 에너지에 관련된 문제가 대두됨에 따라 친환경적 전기 자동차에 대한 관심이 높아지고 있다[1]. 전기자동차의 동력원으로 대용량 리튬이온 또는 리튬 폴리머 배터리가 사용된다. 충전전시 물리적, 화학적인 요소에 따라 셀간 전압 차이가 발생하게 된다. 이러한 차이는 배터리 용량과 안정성, 그리고 수명에 부정적인 영향을 끼친다. 대용량 배터리의 셀 전압 균등화를 위한 다양한 연구가 진행되고 있다 [2].

기존 연구에서 패시브 셀 밸런싱은 주로 회로 설계와 하드웨어 의존적인 방법이 주를 이루고 있다 [2,3]. 이러한 접근은 셀 전압과 온도를 적절하게 반영하지 못한다. 셀 밸런싱을 신속하게 할 수 있는 상태에서나 하지 않아도 될 상황을 고려하기에 어려움이 있어 비효율적이다.

본 논문에서는 소프트웨어 에이전트 모델에 기반한 대용량 배터리의 패시브 셀 밸런싱 시스템을 제안한다. 배터리 전압과 온도를 입력으로 하여 상황에 적절한 셀 밸런싱 방법을 적용한다. 실제 온라인 전기 자동차에서 사용되는 배터리 관리 시스템(BMS, Battery Management system)에 시뮬레이션을 수행하여 제안하는 에이전트와 기존 방법을 비교함으로써 시스템

의 유용성을 입증하였다.

2. 관련연구

셀 밸런싱은 리튬 이온 또는 리튬 폴리머 배터리의 충전전시 발생하는 셀간 전압차를 감소시키는 과정을 의미한다. 셀 전압에 따라 SoC(State of charge)가 결정되므로 배터리의 충전량을 극대화하기 위한 방법이다. 셀 균등화는 크게 패시브 셀 밸런싱과 액티브 셀 밸런싱으로 나뉘어진다. 패시브 셀 밸런싱은 낮은 전압의 셀을 기준으로 높은 전압의 셀 전압을 감소시킨다. 액티브 셀 밸런싱은 높은 전압의 셀에서 낮은 전압의 셀로 충전을 수행하여 전압차를 줄이는 방법이다. 액티브 셀 밸런싱의 경우 전압 손실을, 속도면에서 장점이 있는 반면 패시브 셀 밸런싱의 경우 안정성과 비용면에서 유리하다 [2,4].

액티브 셀밸런싱의 경우 지능형 시스템을 도입하여 빠른 셀 밸런싱을 수행하는 연구가 진행되었다. Lee와 Cheng은 각 배터리 셀 전압을 입력으로 받고 퍼지규칙을 통하여 전압이 높은 셀과 낮은 셀을 구분하였다. 구분된 셀을 이용하여 전압이 낮은 셀을 전압이 높은 셀의 전류를 이용하여 충전을 수행하였다 [3]. 액티브 셀 밸런싱과는 달리 패시브 셀 밸런싱의 경우 다양한 상태에 맞는 방법을 적용하기에 어려움이 있

다. 따라서 상황에 적절한 셀 밸런싱을 수행하기 어렵다. 본 논문에서는 소프트웨어 에이전트 모델을 기반으로 패시브 셀 밸런싱 제어 방법을 제안한다.

3. 제안하는 방법

3.1 온라인 전기자동차 배터리 구조

그림 1은 KAIST 온라인 전기자동차에 사용될 예정인 대용량 배터리 모듈 설계 도형이다 [1]. 온라인 전기자동차의 중대형배터리는 다수의 모듈로 구성되어 있다. 1개의 모듈은 다수의 배터리 셀로 구성되어 있으며 온도관리를 위한 자체 냉각팬이 내장되어 있다. 전체 168개의 셀로 구성되어 있으며 셀 전압은 노말 3.8V 최대 4.2V로 배터리이다. 팩 전압은 노말 650V, 최대 705V이다.



(그림 1) 온라인 전기자동차 배터리 모듈

3.2 셀 밸런싱 에이전트

그림 2는 제안하는 셀 밸런스 에이전트의 개요를 보여준다. 반응형(Reactive) 에이전트 모델을 기반으로 하였다. 반응형 에이전트는 환경을 인식하여 설계된 모델 상에서 계획(plan)에 의해 상태를 결정하여 적합한 반응을 수행하는 것이다[5]. 본 논문에서 사용한 반응형 에이전트의 계획은 아래 식과 같은 구조이다.

$$a_i = \begin{cases} null, & \text{if a goal is achieved} \\ p(s), & \text{if otherwise} \end{cases} \quad (1)$$

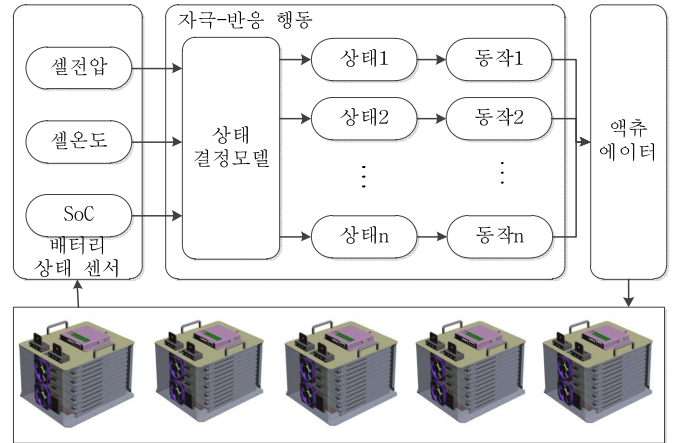
$$s' = A(s, a_i)$$

a_i 는 행동 집합의 원소를 의미한다. 만약 에이전트의 목적이 달성되었을 경우 아무 동작도 하지 않으며 그렇지 않을 경우 현재 상태(s)에 적합한 계획 p 에 따른 값을 가진다. 여기서 목적은 셀 밸런싱의 완료, 즉, 셀간 전압차가 일정 수준 이내로 들어옴을 의미한다. s' 는 다음 상태를 의미하며 $A(s, a_i)$ 는 현재상태(s)에서 행동 a_i 를 수행한 결과 다음 상태값을 반환하는 함수이다. 배터리 상태 센서는 현재 배터리에 대한 정보를 획득한다. 획득되는 정보는 셀전압, 셀온도, SoC이다. 자극-반응 행동 모듈은 현재 상황에서 에이전트가 상태를 파악하고 어떤 행동을 수행하게 될지를 결정하게 되는 부분이다. 액츄에이터(Actuator)는 배터리 셀 밸런싱을 물리적으로 수행하는 부분이다.

3.3 자극-반응 행동

자극-반응 행동 모듈은 에이전트가 상태결정모델을 통해 인식하고 그에 적절한 반응 도출 단계를 담당한다.

다. 상태 결정 모델은 환경 요소들이 배터리가 어떤 상태에 있는지를 반영한다. 그림 3은 n 개의 셀을 가진 배터리의 상태결정모델을 보여준다. 본 논문에서는 모델을 의사결정트리을 이용하여 모델을 표현된다. SoC는 배터리의 충전 상태, C_1 은 배터리 빠른 충전이 요구되는 기준 SoC, C_2 는 셀 전압 허용 편차, V_c 는 셀 전압의 표준편차, V_i 은 i 셀의 전압, V_{min} 은 배터리의 전체 셀 중 최소 전압, V_c 는 셀별 과방전을 방지하기 위한 유의수준, T_i 은 i 셀의 온도, T_t 은 현재 배터리 모듈에 장착된 팬 동작이 가능한지를 판단하는 안전 온도의 유의수준이다.

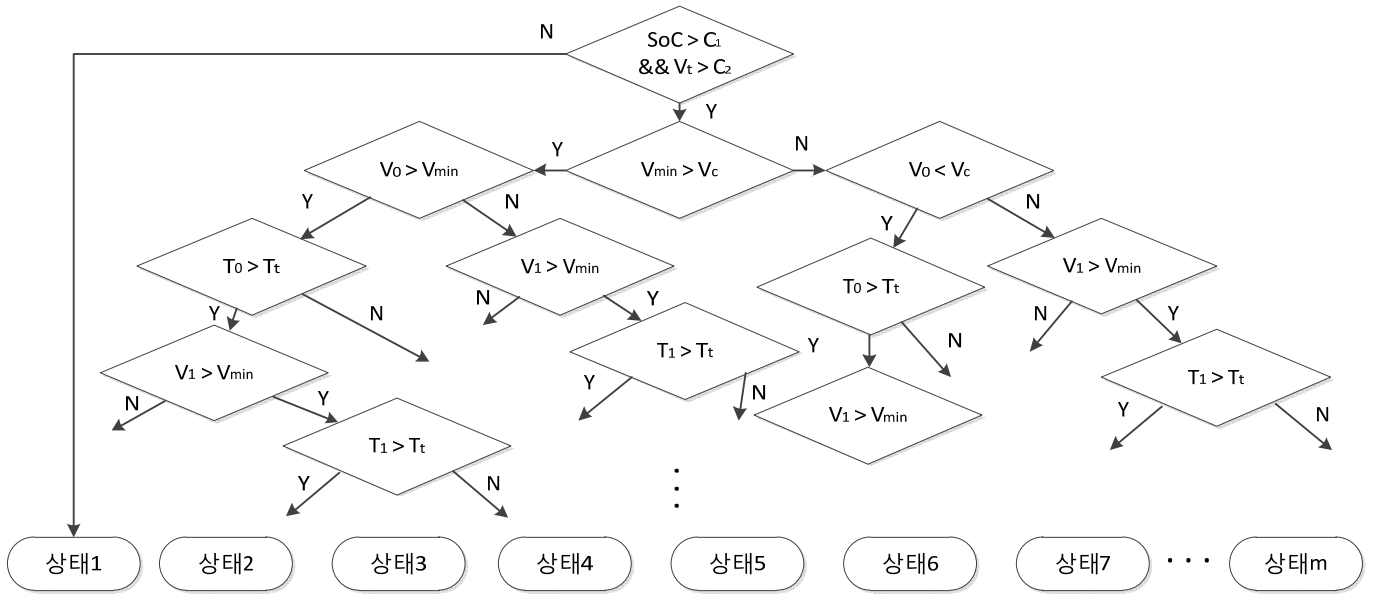


(그림 2) 셀 밸런스 에이전트

배터리의 SoC 잔존량이 시스템을 유지하기에 부족하여 빠른 충전이 요구되는 상황인지와 전체 셀간 전압 편차가 안정된 범위 이내로 들어왔는지를 통해 균등화 여부를 결정한다. 균등화가 허용되었으면 V_{min} 이 V_c 보다 큰 경우 V_{min} 보다 큰 전압을 가진 셀을, 반대의 경우 V_c 보다 큰 전압을 가진 셀의 전압을 감소시키기로 결정한다. 이때 T_i 를 측정해 현재 온도가 셀 모듈에 설치된 냉각팬을 가동시킬 수 있는 상태인지 판단한다. 밸런싱의 신속성을 위해 높은 셀의 전압을 빨리 감소시키기 위함이다. 배터리 상황과 환경을 바탕으로 상태를 인식한다.

C_3 는 추가적인 조건의 수를 나타낸다. 그림 3의 경우 C_3 는 1이다. 자극-반응행동에서 모델을 통해 인식된 각 상태는 적절한 동작과 매칭된다.

표 1은 상태별 셀밸런싱 상태별 동작 기준을 보여준다. 그림 4은 상태에 따른 셀 밸런싱을 위한 동작 결정 의사코드를 보여준다. $T_{au}(Module)$ 은 모듈 안에 모든 셀 온도의 집합을 나타낸다. $M(i)$ 은 셀 i 가 포함된 모듈을 의미한다. 각 셀별 밸런싱을 수행하기 위한 전압의 조건이 만족하면 해당 모듈의 전압을 감소시킨다. 모듈당 모든 셀 온도가 T_i 이상이고 V_i 와 셀 기준 전압간의 차가 팬이 동작할 때 소모되는 전압 V_{Fan} 보다 크다면 팬을 동작시켜 전압감소를 돕는다.



(그림 3) 의사결정트리 기반 상태결정 모델

<표 1> 상태별 셀 밸런싱 수행 자격

상태	동작
1	셀 밸런싱 수행 X
2~m/2	V_{min} 기준 셀 밸런싱
$\frac{m}{2} + 1 \sim m$	V_c 기준 셀 밸런싱

이러한 조건 하에서 표 3 과 같이 셀 밸런싱을 수행하지 않았을 경우, 패시브 셀 밸런싱 방법처럼 최저 셀 전압을 기준으로 다른 셀의 전압을 일정 간격으로 소모 시키는 방법, 제안하는 방법을 사용에 대한 시뮬레이션 결과를 비교 분석하였다.

```

V=((Vc>Vmin)?Vc:Vmin);
If Vi > V
then If Tall(M(i)) > Ti && (Vi - V) > VFan
    then
        FanOn(M(i));
        DoCellBalancing(i);
    else DoCellBalancing(i);
end
end
    
```

(그림 4) 상태별 동작 결정 의사코드

<표 2> 실험 변수

변수	값
모듈/배터리	14 개
셀/모듈	12 개(총 168 개)
전압범위	3V~4.2V(최대 705V)
단위 충전전압	1mV
저항 소비전압	0.1mV~0.8mV
단위 셀 밸런싱 전압	0.7mV
모듈 팬 소비 전압	2.4mV
최소 전압	3V
최소 SoC	10%
표준편차 유의수준	0.001
모듈 팬	온도 조건 항상 만족

4. 실험 및 평가

제안하는 방법의 유용성을 입증하기 위해 온라인 전기자동차에 내장된 중대형 대용량 배터리 환경을 기반으로 시뮬레이션을 수행하였다.

4.1 실험 환경

표 2 는 시뮬레이션 변수들을 나타낸다. 배터리는 총 14 개의 셀로 구성되어 있으며 하나의 모듈은 그림 1 과 같이 12 개의 셀로 구성이 되어 있다. 배터리가 완전방전 되었을 시 전압은 0 이고 최대 충전되었을 시 전압은 4.2V 이다. 각 단위 시간별 충전 전압은 1mV, 충전을 방해하는 저항 소비전압은 각 셀의 특성에 따라 0.0mV~0.8mV 로 임의 설정된다. 패시브 셀 밸런싱시 단위 시간당 0.07mV 가 소모된다. 배터리 관리 시스템과 안정성, 수명을 위해 배터리가 유지해야 할 최소 전압은 온라인 전기자동차에서 3V 로 정의하였다. 셀 초기 전압은 또한 최소 전압의 값을 따른다.

<표 3> 실험군

실험군	셀밸런싱	충전	무부하
A	수행 X	○	
B	기존 방법	○	○
C	제안 방법	○	○

충전 실험은 충전시 셀 밸런싱을 수행하면서 전압에 대한 로그를 남겼다. 무부하 실험은 실험군 A 처럼 셀 밸런싱 없이 충전을 하고 무부하 상태에서 셀 밸런싱을 수행하였다.

4.2 평가

표 4 는 충전 실험 시 표 3 의 실험군 A, B, C 의 결과를 보여준다. 충전 완료 후 셀 밸런싱을 수행하지

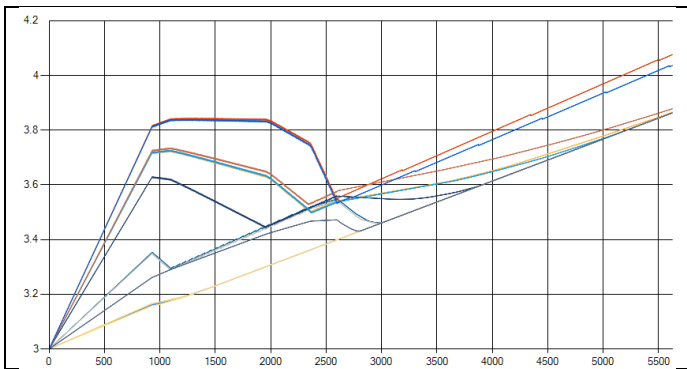
많은 A의 표준편차가 가장 컸다. B의 경우 표준편차는 A에 비해 더 많은 충전 단계가 소요됨을 확인할 수 있다. A와 B의 팩 전압을 관찰했을 시 셀 밸런싱을 수행한 B의 팩 전압이 높은 모습을 보였다. C는 충전시 필요한 단계는 B보다 작았다. SoC 10% 도달까지 소요되는 시간은 B가 가장 컸고 A와 C가 유사한 수를 보였다. 셀 밸런싱이 잘 진행된 경우 전체 팩 전압이 더 높아져 더 많은 수용량을 가질 수 있음을 확인할 수 있다.

<표 4> 충전 실험 결과

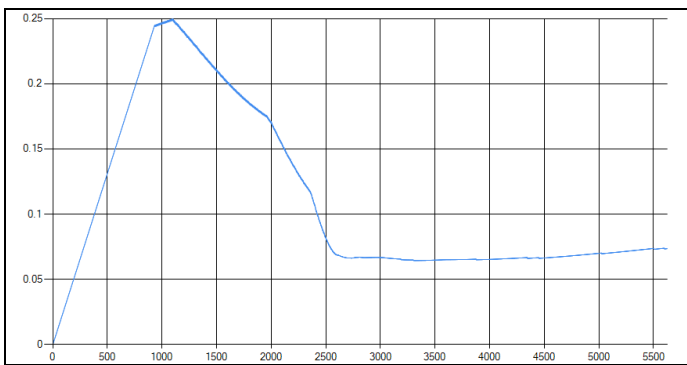
실험군	소요 단계	SoC 10%	팩전압	표준편차
A	1356	931	606.45	0.28
B	5996	4671	619.97	0.11
C	5642	931	653.82	0.07

<표 5> 무부하 실험 결과

실험군	소요 단계	SoC 10% 도달 단계	팩전압	표준편차
B	1409	423	538.06	0.00093
C	89	89	589.65	0.12169



(그림 5) B의 충전시 모듈 단위 전압변화량 그래프
(x 축: 전압, Y 축: 소요단계)



(그림 6) B의 충전시 표준편차 변화량 그래프
(X 축: 표준편차, Y 축: 소요단계)

그림 5는 제안하는 방법을 사용하여 충전하였을 경우 모듈 단위 전압 변화량을 보여준다. SoC 10%가 되는 점까지 상대적으로 빠른 충전 양상을 보였으며 그 후 냉각 팬과 자체 저항을 사용한 셀 밸런싱을 하는 모습을 보였다. 그림 6은 제안하는 방법의 충전시 표준편차 변화 양상을 보여준다. SoC 10%가 되는 지점까지는 셀 간의 편차가 컸지만 SoC 값이 최소 조건

을 만족한 후에는 셀 밸런싱을 통해 셀 간 전압차가 줄어드는 모습을 보여준다. 표 5는 무부하시 셀 밸런싱 실험 결과를 보여준다. SoC 10%에 도달하는 단계를 보면 제안하는 방법 C가 기존 하드웨어 의존적인 B보다 더 빠른 밸런싱 속도를 보여줌을 확인할 수 있다. 그리고 최저 SoC와 최저 전압을 보호할 수 있으므로 배터리 안정성을 보장한다. 표준 편차의 차이는 셀 밸런싱을 많이 진행할수록 줄어드는 부분이기 때문에 단순 비교가 불가능하다. 셀 밸런싱 완료 표준편차를 고려하면 SoC 10%일 때 B의 표준편차는 0.16으로 C보다 큰 값을 지닌다.

소프트웨어 에이전트를 사용하여 다양한 셀의 상태에 따른 다른 셀 밸런싱 방법을 적용하였을 경우 셀 밸런싱의 신속성, 시스템의 안정성을 보장함을 확인할 수 있었다.

5. 결론

본 논문에서는 반응성 소프트웨어 에이전트 모델을 기반한 리튬 이온-리튬 폴리머 대용량 배터리의 효율적인 패시브 셀 밸런싱 방법을 제안하였다. SoC가 특정 수치 이하로 낮은 경우 충전의 신속성에 최우선 순위를 두었다. 셀 밸런싱 기준점을 두어 과방전을 방지하였고 상태에 따라 전력 소모량을 크게 함으로 셀 밸런싱의 신속성을 보장하였다. 온라인 전기자동차 BMS 환경을 반영한 시뮬레이션을 통하여 소프트웨어 에이전트의 규칙을 기반으로 셀 밸런싱을 수행함으로 효율적인 셀 밸런싱이 가능하여짐을 확인하였다.

카이스트 온라인 전기자동차 환경에서의 적용과 환경의 불확실성에 반응할 수 있는 지능형 시스템을 적용하여 패시브와 액티브 셀 밸런싱 방법에 효율성을 높이는 연구와 향후에 진행할 예정이다.

감사의 글

본 연구는 지식경제부의 지원사업인 기술혁신사업의 일환으로 수행되었음(10036221, 온라인 전기버스용 다중 동력원 전력공급 및 제어장치 개발)

참고문헌

- [1] D. Seo et al., "Research on development of hardware platform system for on-line electric vehicle," *IEEK Information and Control Symposium*, pp. 411-412, 2009.
- [2] S. W. Moore and P. J. Schneider, "A review of cell equalization methods polymer battery systems," *in SAE Technical Paper Series*, pp. 1-5, 2001.
- [3] Y.-S. Lee and M.-W. Cheng, "Intelligent control battery equalization for series connected lithium-ion battery strings," *IEEE Trans. on Industrial Electronics*, vol. 52, no. 5, pp.1297-1307, 2005.
- [4] T. A. Stuart and W. Zhu, "Fast equalization for large lithium ion batteries," *IEEE A&E Systems Magazine*, pp. 27-31, 2009.
- [5] G. Boella and R. Damiano, "A replanning algorithm for a reactive agent architecture," *Lecture Notes in Computer Science*, vol. 2443, pp. 183-187, 2002.