

# Dynamic Aggregation of Energy Storage Systems Into Virtual Power Plants Using Distribution Real-Time Clustering Algorithm

## 분산 실시간 클러스터링 알고리즘을 사용하여 에너지 저장 시스템을 가상 발전소로 동적 집합

**Abstract**—에너지 저장 시스템은 간헐적인(얼마 동안의 시간 간격을 두고 되풀이하여 일어난다는 뜻) 재생 에너지를 보상하고 시스템 주파수 및 전압을 복원하는 데 널리 사용됩니다. 기존 운영 방식에서는 모든 분산형 에너지 저장 시스템이 하나의 고정된 가상 발전소로 클러스터링 되고 충전 상태는 공통된 값으로 유지합니다. 이 논문에서는 에너지 저장 시스템의 전력 수요와 용량에 따라 에너지 저장 시스템을 여러 가상 발전소로 동적으로 클러스터링하는 방법을 제안합니다. 그 결과 네트워크 전력 손실을 줄일 수 있습니다. 제안된 동적 클러스터링 알고리즘을 사용하면 미리 선택된 feature 상태(로컬 전력 수요 및 에너지 저장 용량)를 기반으로 에이전트(에너지 저장 시스템)를 클러스터링할 수 있습니다. 클러스터를 결정하기 위해 모든 클러스터에서 에이전트의 현재 feature 상태의 거리가 평균 추정치와의 거리로부터 결정됩니다. 이 알고리즘은 제어 목적으로 사용할 수 있는 지정된 보조 상태의 평균 추정치도 제공합니다. 제시된 RTDS 기반 실시간 구현 결과를 통해 에너지 저장 시스템(배터리)을 동적 가상 발전소로 클러스터링하면 네트워크 전력 손실을 줄일 수 있음을 확인할 수 있습니다.

**Index Terms**—Balancing, battery, 분산 클러스터, microgrid(마이크로그리드), multiagent, reduction of power losses(전력 손실 감소), state of charge(SoC), voltage regulation(전압 조정)

## I. INTRODUCTION

microgrid는 지역 단위의 소규모 전력망으로, 태양광(PV) 및 풍력 발전과 같은 재생 에너지원과 에너지 저장 시스템을 효과적으로 활용할 수 있게 해줍니다. 전통적으로 microgrid에는 1차, 2차, 3차 수준의 제어 시스템을 갖춘 계층적 제어 구조가 적용되어 왔습니다. 1차 레벨 제어는 비례적인 전력 공유를 달성하기 위해 droop 제어를 구현합니다. 그 후 2차 레벨 제어가 적용되어 droop 제어로 인해 발생하는 전압 및/또는 주파수 편차를 제거하고 정확한 전력 공유를 달성합니다. 필요한 경우 3차 레벨 제어에서 전력 흐름 목표를 구현할 수 있습니다.

분산형 에너지 저장 시스템(ESS)의 활용도와 microgrid의 전력 품질을 개선하기 위해 2차 레벨 제어에서의 충전 상태(SoC) 밸런싱 제어 전략이 제안되었습니다. 이 전략에서는 모든 ESS를 하나의 그룹으로 묶어 ESS 기반 가상 발전소(VPP)를 만들었습니다. 모든 ESS의 SoC를 동일한 수준으로 유지하고 ESS 간 비례적인 전력 공유를 제공했습니다. 하지만 다양한 ESS와 로컬 부하가 많은 마이크로그리드에서는 전력망 선로 전류가 흐르기 때문에 추가적인 전력 손실이 발생할 수 있습니다. 또한, 분산형 에너지 자원 관리 측면에서 P2P 에너지 거래는 유망한 수단이며, P2P 네트워크에 기반한 에너지 거래 전략에서 [9]에서 검토되었습니다. 이 문제는 물리적 계층과 가상 계층으로 나뉘었습니다. 각 계층에서 게임 이론 [10]-[14] 및 이중 경매와 같은 기술적 접근 방식이 도입되었습니다. 연구 결과, [10], [14]은 분산 에너지 자원, 에너지 저장소, 프로슈머, 난방 부하에 대한 P2P 에너지 거래에서 다양한 비협조적인 스택켈버그 게임(Stackelberg Game)을 구현하여 프로슈머에게 혜택을 주고 전력 흐름을 최적화하며 거래 가격을 결정할 수 있는 방법을 제시했습니다. [15]에서는 분산형 발전기의 가동을 줄이고 배터리 수명을 연장하며 태양광 발전의 효율을 높이기 위한 배터리 관리 전략이 제안되었습니다. 더 중요한 것은, [9]에서 VPP형성을 포함한 P2P 에너지 거래의 잠재적 주제를 확인했다는 것입니다.

클러스터링 알고리즘은 데이터를 유사한 속성을 가진 그룹으로 나누는 데 사용됩니다. 중앙 집중식 및 분산 클러스터링 알고리즘은 모두 문헌에서 제안되었습니다. 중앙 집중식 클러스터링 알고리즘 중 가장 일반적으로 사용되는 알고리즘은 다음과 같습니다. 자기 조직화 맵, 기대 최대화 클러스터, K-means 또는 계층적 클러스터링. 분산된 데이터에 대해 분산 클러스터링 알고리즘이 제안되었습니다. 그러나 이러한 분산 알고리즘은 글로벌 클러스터링이 수행되는 메인/중앙 사이트가 필요하고 개별 사이트는 중앙 네트워크를 통해 메인 사이트와 통신합니다[22]. 언급된 모든 알고리즘은 이전에 수집된 데이터를 기반으로 클러스터를 생성하여 실시간이 아니므로 분산 네트워크를 통해 통신하는 시스템에는 클러스터링에 적합하지 않습니다.

클러스터링 알고리즘은 다음 네 가지 분야에서 microgrid에 적용되었습니다. 첫 번째 카테고리, 클러스터링은 전력 시스템 경제학을 개선하기 위해 적용되었습니다. 전력 시스템은 다음과 같습니다.

과거 부하 및 재검토 가능한 에너지 생성을 기반으로 클러스터링 되었습니다.[23]-[29] 두 번째 카테고리, 확률론적 모델링을 사용하여 365일 데이터를 다음과 같이 나누었습니다. 거래에서 다양하고 확률적인 시나리오로 수익을 창출합니다. [30],[31] 세 번째 카테고리, 전력 시스템은 지리적 영역 내에서 이익을 위해 집계되었지만 클러스터링 알고리즘을 적용하지 않았

습니다 [32]-[35]. 마지막 카테고리에서는 클러스터링 방법을 사용하여 전압 소스 변환기 감도 행렬 및 전압 영향 요인[36]을 기반으로 전력 시스템 네트워크를 구성하는 데 사용되었습니다. 그러나 모든 군집화 알고리즘은 과거 데이터를 필요로 하며 오프라인에 적용해야 합니다. 즉, 시스템 작동 전위에서 언급한 논의를 바탕으로

이 글에서는 글로벌 ESS의 SoC 관리로 인한 전력 손실을 줄이기 위해 분산형 동적 VPP 및 분산 제어 시스템에 대해 다룹니다. 모든 ESS를 “글로벌” 공통 SoC 수준으로 밸런싱하는 것이 아니라 ESS는 여러 VPP로 클러스터링되며 각 VPP의 ESS는 “로컬” 공통 SoC 수준으로 밸런싱됩니다. 이러한 목적을 달성하기 위해, 첫 째, 본 논문에서는 분산 네트워크를 통해 통신하는 다중 에이전트(ESS)를 위한 새로운 완전 분산 동적 실시간 클러스터링 알고리즘을 제안합니다. 이러한 유형의 분산형 실시간 클러스터링 알고리즘은 자신과 이웃 에이전트의 현재 상태(feature 상태)를 기반으로 에이전트를 특정 그룹에 동적으로 클러스터링할 수 있으며, 기존 문헌에서는 보고된 적이 없습니다. 둘째, 전력 손실을 줄이기 위해 에너지 저장소 용량과 해당 지역의 전력수요를 특정 상태로 식별합니다. 그 이유는 용량이 비슷하고 로컬 전력 수요가 비슷한 에너지 저장장치를 vpp로 클러스터링 해야 하기 때문입니다. 셋 째, 이 알고리즘은 기존 분산형 보조 제어 시스템과 협력하여 각 VPP에 있는 ESS의 균형 잡힌 SoC를 유지합니다. 따라서 클러스터링 알고리즘은 에이전트(ESS)를 클러스터(VPP)로 동적으로 집계하는

동시에 각 클러스터(VPP) 내에서 균형 잡힌 SoC를 유지합니다. 하나의 고정된 VPP에 비해 ESS를 여러 개의 VPP로 동적으로 통합하면 네트워크 전력 손실을 줄일 수 있습니다. 이 글의 주요 내용은 다음과 같습니다.

- 1) 클러스터링 알고리즘은 분산되어 있으며 멀티 에이전트(ESS) 상태의 변화에 실시간으로 동적으로 조정할 수 있습니다. 과거 데이터가 필요하지 않으며 확장성이 뛰어납니다.
- 2) 클러스터링 알고리즘은 멀티 에이전트 상태의 빠른 변화에 강하므로 잦은 클러스터 변경을 제거 할 수 있습니다.
- 3) 이 알고리즘은 ESS의 로컬 수요와 용량에 따라 VPP를 생성하여 전력 손실을 줄이고 전압 조절을 향상 시킵니다.

## II. 분산된 실시간 클러스터링 알고리즘

제안하는 클러스터링 알고리즘은 분산 네트워크를 활용하기 때문에 먼저 다중 분산 네트워크의 개념을 소개하고, 이어서 클러스터링 알고리즘에 대해 자세히 설명합니다.

### A. 분산 통신 네트워크

멀티에이전트는 노드  $V = \{1, \dots, N\}$ 과 에지  $E$ 로 표현됩니다[37]. 각 그래프 노드는 에이전트를 나타냅니다.  $E$ 의 요소는  $(i, j)$ 로 표시되며, 노드  $i$ 에서 노드  $j$ 로 정보를 전달할 수 있는 링크가 있는 경우  $(i, j) \in E$ 입니다. 노드  $i$ 의 이웃은  $N_i$ 로 표시되며, 다음과 같은 경우  $j \in N_i$ 입니다.  $(i, j) \in E$ . 그래프 인접 행렬은 다음과 같이 주어집니다.

$$\mathcal{A} = [a_{ij}] \in \mathbb{R}^{N \times N}, a_{ij} = \begin{cases} \alpha, & (j, i) \in \mathcal{E} \\ 0, & \text{otherwise} \end{cases} \quad (1)$$

여기서  $\alpha$ 는 결합 강도입니다. 그러면 그래프 라플라시안 행렬은  $L = D - A$ , where  $D = \text{diag} \{d_i\}$  그리고  $d_i = \sum_{j=1}^N a_{ij}$ 는 각 노드의 진입차수입니다.

## B. 동적 실시간 클러스터링 알고리즘

클러스터를 결정하는 데 사용되는 에이전트의 feature 상태를  $x_i \in \mathcal{R}^n$ 으로 정의합니다. 제안된 알고리즘의 목표는 feature 상태에 따라 N개의 에이전트를 M개의 클러스터로 클러스터링하는 것입니다.  $\bar{x}_i^{cstrj}$  ( $i = 1, \dots, N$  and  $j = 1, \dots, M$ )은 j번째 클러스터에 있는 feature 상태  $x_i$ 의 평균 추정치라고 합니다. 각각의 i번째 에이전트는 통신 네트워크를 통해 모든 클러스터의 현재 평균 상태  $\bar{x}_i^{cstrj}$ 에 실시간으로 접근할 수 있습니다.

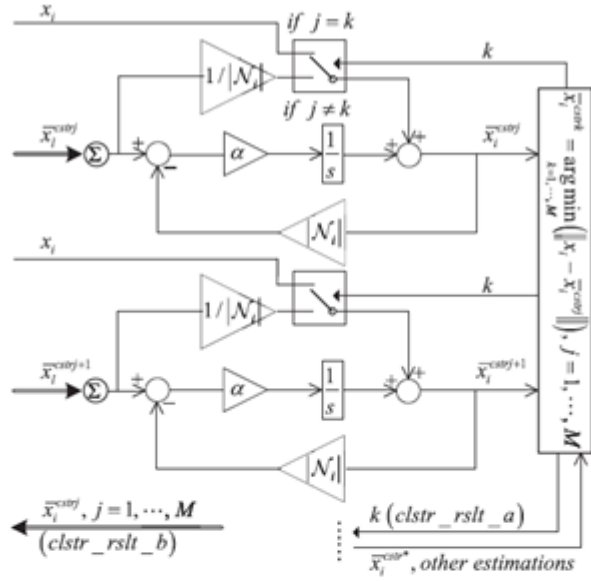
먼저, (임의의) 초기 추정치  $\bar{x}_i^{cstrj}$ ,  $j = 1, \dots, M$ 를 선택합니다. 다음으로 i번째 에이전트의 현재 상태  $x_i$ 를 모든 현재 추정치와 비교하여 i번째 에이전트가 어느 클러스터에 속할지 결정/업데이트합니다. (4)에 의해 결정된 대로 현재 상태  $x_i$ 에서 가장 작은 거리를 가진 k번째 추정치  $\bar{x}_i^{cstrk}$ 가 k번째 클러스터에 할당됩니다. 모든 클러스터가 결정되면 각 i번째 에이전트는 자신의 현재 상태  $x_i$ 와 네트워크를 통해 전파된 k번째 클러스터 에이전트의 추정치를 기반으로 (2)를 사용하여 자신의 k번째 클러스터 상태의 평균을 추정합니다. k번째 클러스터에 속하지 않는 이웃 에이전트의 추정치  $j \in \{1, \dots, k-1, k+1, \dots, M\}$ ,  $j \neq k$ 인 경우, (3)과 같이 다음과 같이 전달됩니다:

$$\dot{\bar{x}}_i^{cstrk} = \dot{x}_i + \sum_{l \in \mathcal{N}_i} a_{il} (\bar{x}_l^{cstrk} - \bar{x}_i^{cstrk}) \quad (2)$$

$$\dot{\bar{x}}_i^{cstrj} = \frac{1}{|\mathcal{N}_i|} \sum_{l \in \mathcal{N}_i} \dot{\bar{x}}_l^{cstrj} + \sum_{l \in \mathcal{N}_i} a_{il} (\bar{x}_l^{cstrj} - \bar{x}_i^{cstrj}), \quad j \neq k \quad (3)$$

(3)에서  $|\mathcal{N}_i|$ 는 i번째 에이전트의 이웃 수를 나타냅니다.

$$x_i \in \left\{ k\text{th cluster} \left| \bar{x}_i^{cstrk} = \arg \min_{k \in \{1, \dots, M\}} \left( \|x_i - \bar{x}_i^{cstrj}\| \right), \right. \right. \\ \left. \left. j = 1, \sum_{k=1}^{j=1} \dots, M \right\}. \quad (4)$$



**Algorithm 1** Distributed Dynamic Clustering

- 1: design  $\mathcal{G}(\mathcal{V}, \mathcal{E})$  and  $M$
- 2: initialize  $\bar{x}_i^{cstrj}, j = 1, \dots, M$
- 3: **each**  $i$ -th agent at time  $t$
- 4:   measure  $x_i$
- 5:   receive  $\bar{x}_l^{cstrj}, l \in \mathcal{N}_i, j = 1, \dots, M$  from neighbours
- 6:    $k \leftarrow$  label of the cluster with the smallest distance between  $x_i$  and  $\bar{x}_i^{cstrj}, j = 1, \dots, M$ , where  $k \in \{1, \dots, M\}$
- 7:   **if**  $j = k$  **then**
- 8:     Eq. (2)
- 9:   **else**
- 10:    Eq. (3)
- 11:   **end if**
- 12:   send  $\bar{x}_i^{cstrj}, j = 1, \dots, M$  to neighbours
- 13: **end each**
- 14: Each local agent  $i$  implements the same algorithm.

**Fig.1.** 클러스터링 알고리즘과  $i$ 번째 에이전트의 구현에 대한 일반적인 설명



**Fig.2.** 9개의 에이전트 노드를 3개의 그룹으로 클러스터링했을 때 클러스터링 알고리즘의 동작을 보여주는 예시입니다. 노드 사이의 검은색 실선은 에이전트 노드(ESS) 간의 실제 분산 통신 네트워크를 나타냅니다. 원, 사각형, 다이아몬드 블록은 현재 클러스터(VPP) 구성원을 나타냅니다. 빨간색, 초록색, 파란색 선은 각 클러스터의 알고리즘 정보 흐름을 나타내는 가상 통신선입니다. 별표는 에이전트가 로컬 측정 데이터(2)를 사용하여 평균값을 업데이트하고, 삼각형은 에이전트가 이웃 에이전트(3)를 사용하여 평균값을 업데이트하고 평균 추정치만 전송한다는 것을 나타냅니다. 예를 들어 가상 녹색 선을 통해 통신하는 정사각형 클러스터에서 노드 2, 3, 6은 (2)를 사용하여 클러스터 상태의 평균 추정치를 업데이트하고 다른 노드는 (3)을 사용하여 추정치를 업데이트합니다.

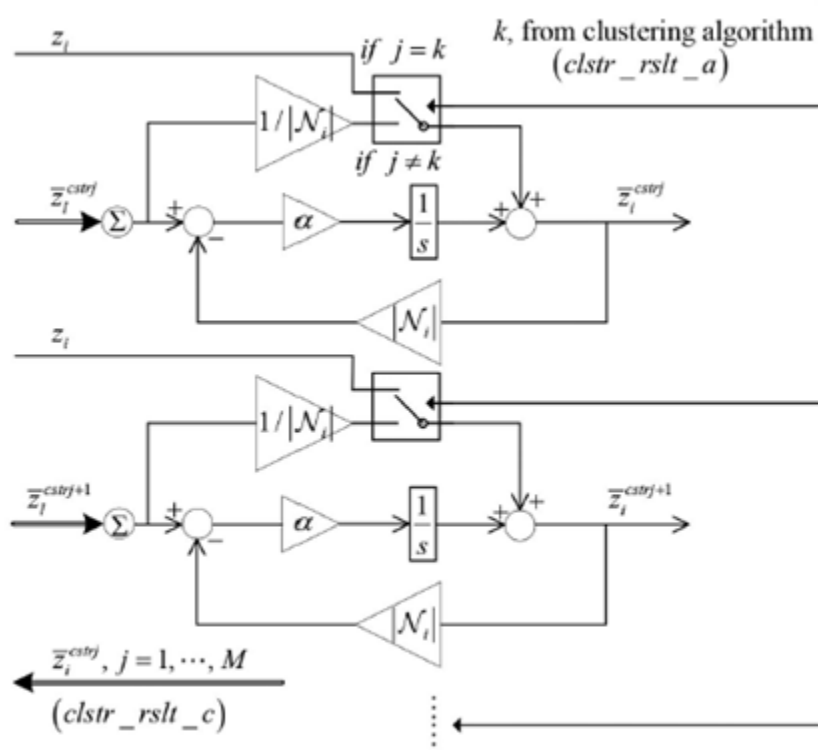
### C. 클러스터 내 제어, 최적화 및 분석

필요한 경우, 제안된 알고리즘은 클러스터 내 또는 클러스터 간 제어, 최적화 및 분석 목적으로 feature 상태가 아닌 다른 에이전트의 상태를 사용할 수 있습니다. 이러한 상태는 보조 상태( $z_i \in m$ )로 정의되며, 보조 상태  $z_i$ 의 추정치는  $\bar{x}_i^{cstrj}$ 이며, 여기서  $i = 1, \dots, N$ 이고  $j = 1, \dots, M$ . 이러한 보조 상태는 클러스터 내 상태의 평균을 계산하는 데만 사용됩니다.

예를 들어, 배터리 전력 수요 및 용량을 배터리를 클러스터링하기 위한 feature 상태로 선택하고, 배터리의 SoC를 클러스터 내의 밸런싱 제어를 위한 SoC의 보조 상태로 선택할 수 있습니다.  $j$ 번째 클러스터에 있는  $i$ 번째 에이전트의 평균 상태 추정치는  $clstr\_slt\_a$  결과를 사용하여  $\bar{x}_i^{cstrj}$ 와 동일한 절차에 따라 얻을 수 있습니다.

$$\dot{\bar{z}}_i^{cstrk} = \dot{z}_i + \sum_{l \in \mathcal{N}_i} a_{il} (\bar{z}_l^{cstrk} - \bar{z}_i^{cstrk}) \quad (5)$$

$$\dot{\bar{z}}_i^{cstrj} = \frac{1}{|\mathcal{N}_i|} \sum_{l \in \mathcal{N}_i} \dot{\bar{z}}_l^{cstrj} + \sum_{l \in \mathcal{N}_i} a_{il} (\bar{z}_l^{cstrj} - \bar{z}_i^{cstrj}), \quad j \neq k. \quad (6)$$




---

**Algorithm 2** Distributed Dynamic Clustering with Utilization of Clusters

---

- 1: design  $\mathcal{G}(\mathcal{V}, \mathcal{E})$  and  $M$
  - 2: initialize  $\bar{x}_i^{cstrj}, j = 1, \dots, M$
  - 3: **each**  $i$ -th agent at time  $t$
  - 4:   measure  $x_i$  and  $z_i$
  - 5:   receive  $\bar{x}_l^{cstrj}, l \in \mathcal{N}_i, j = 1, \dots, M$  from neighbours
  - 6:   receive  $\bar{z}_l^{cstrj}, l \in \mathcal{N}_i, j = 1, \dots, M$  from neighbours
  - 7:    $k \leftarrow$  label of the cluster with the smallest distance between  $x_i$  and  $\bar{x}_i^{cstrj}, j = 1, \dots, M$ , where  $k \in \{1, \dots, M\}$
  - 8:   **if**  $j = k$  **then**
  - 9:     Eq. (2) and (5)
  - 10:   **else**
  - 11:     Eq. (3) and (6)
  - 12:   **end if**
  - 13:   send  $\bar{x}_i^{cstrj}, j = 1, \dots, M$  to neighbours
  - 14:   send  $\bar{z}_i^{cstrj}, j = 1, \dots, M$  to neighbours
  - 15: **end each**
  - 16: Each local agent  $i$  implements the same algorithm.
-



**Fig.3.**  $i$ 번째 에이전트에 대한 보조 상태  $z_i$ 의 평균 추정치입니다. 이중 선 화살표는 분산 통신을 나타냅니다.

따라서 각  $i$ 번째 에이전트는 항상 보조 상태의 평균 추정치  $\bar{z}_i^{cstrj}$ ,  $j = 1, \dots, M$ 을 포함하는 `clstr_rslt_c` 결과에 접근할 수 있습니다. 이 알고리즘은 그림 3에 알고리즘 2로 설명 및 요약되어 있습니다. 알고리즘 2는 알고리즘 1에 보조 상태를 포함시킴으로써 얻을 수 있습니다.

참고 1: feature 상태가 크게 변경되는 경우에만 에이전트가 한 클러스터에서 다른 클러스터로 이동하게 됩니다. 이는 feature 상태가 변경되면 현재 클러스터 내의 평균 추정치와의 거리도 변경되기 때문입니다. 따라서 변동이 충분히 커서 feature 상태가 인접 클러스터의 평균 추정치에 가까워지는 경우에만 에이전트가 클러스터를 변경합니다. feature 상태의 빈번한 큰 변화로 인해 빈번한 재클러스터링을 줄이는 것이 바람직한 경우 feature 상태에 low pass 필터를 적용할 수 있습니다.

### III. VIRTUAL POWER PLANTS

이 절에서는 VPP의 개념을 소개합니다.

#### A. Problem Description

각각의 ESS는 배터리 및 양방향 dc-dc 컨버터를 포함한다. dc-dc 컨버터는 배터리와 마이크로그리드 및 로컬 부하를 연결합니다.

기존에는 전력계통의 신뢰성과 안정성을 향상시키고 배터리 수명을 증가시키기 위해 마이크로그리드에서 모든 배터리의 SoC 균형을 맞추는 것이 제안되었다. 그러나, 모든 배터리가 하나의 그룹으로 통합되거나, 임의로 여러 고정 그룹(VPP)으로 통합되는 경우, SoC가 균형을 이루더라도 추가적인 네트워크 전력 손실이 발생할 것이다. 이러한 전력 손실은 다음에 설명된 바와 같이 다른 전력 요구량 및 배터리 용량에 의해 발생합니다.

#### B. Power Demand

전력 수요는 배터리에 필요한 전력으로 정의됩니다. 구체적으로  $P_{dmnd} = P_{ld} - P_{pv}$ 이며, 여기서  $P_{dmnd}$ 는 전력 수요,  $P_{ld}$ 는 로컬 부하,  $P_{pv}$ 는 PV 소스 전력이다. 모든 배터리의 용량이 같다고 가정하자. 비슷한 전력 수요 프로파일을 가진 배터리를 클러스터링하여 전력 손실을 줄일 수 있다. 클러스터의 각 배터리가 유사한 전력 수요 프로파일을 가지면 배터리 사이에 순환하는 라인 전류가 없기 때문이다. 즉, 클러스터의 각 배터리는 로컬 PV 소스 또는 로컬 부하에 의해 각각 충전 또는 방전됩니다.

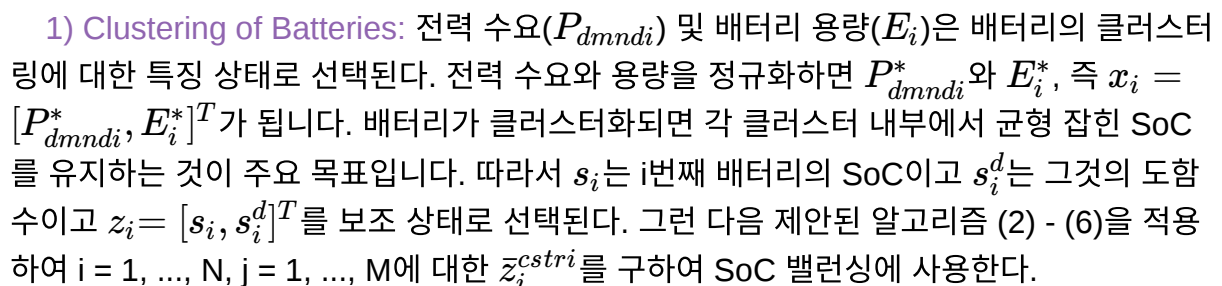
#### C. Battery Capacity

모든 배터리에 동일한 로컬 PV 소스 및/또는 부하가 있고 균형 상태의 SoC에 도달했다고 가정합니다. 그러면, 용량이 큰 전지는 용량이 작은 전지보다 더 많은 전력을 공급/흡수해야

이상의 논의를 바탕으로, 제안된 분산 클러스터링 알고리즘을 이용하여 배터리를 VPP로 클러스터링하기 위해 전력 수요와 배터리 용량을 두 가지 특징 상태로 선택한다.

이 절에서는 네트워크 전력 손실을 줄이기 위해 제안된 알고리즘을 배터리를 VPP로 클러스터링하는 데 적용한다.

제안된 클러스터링 알고리즘은 클러스터 내 SoC 밸런싱(VPP) 및 microgrid 내 전압 조절을 달성하기 위해 1차 및 2차 레벨 제어와 병행된다. Fig. 4는 클러스터링 알고리즘이 동적으로 변화하는 클러스터링 알고리즘과 제어 시스템의 구현을 설명하기 위한 구조이다.



Dynamic Aggregation of Energy Storage Systems Into Virtual Power Plants Using Distribution Real-Time Clustering Algorithm

(period)에서 신호가 켜져있는 시간의 비율을 백분율로 나타낸 수치)을 생성합니다. 이 제어 수준에서는 V-I droop 제어가 BESS를 조절하여 비례 전력 공유를 제공한다. 그런 다음, 2차 레벨 제어에서 제안된 알고리즘과 전압 복원을 기반으로 한 SoC 밸런싱 제어를 제안한다. 1차 레벨 컨트롤은 droop 컨트롤을 통해 2차 레벨 컨트롤과 상호 작용합니다. droop 제어의 전압 기준은 2차 레벨 제어  $v_i^{ref} = \int (v_i^s + v_i^v) dt$ 에 의해 설정되며, 여기서  $v_i^{ref}$ 는 드롭 제어의 전압 기준이고,  $v_i^s$  및  $v_i^v$ 는 각각 2차 SoC 밸런싱 제어 및 전압 복원 제어이며, 다음과 같이 설계된다.

**3) SoC Balancing:** VPP로 군집된 배터리의 균형 잡힌 SoC를 유지하기 위해, 제어 입력  $v_i^s$ 를 생성하기 위해 [39]의 분산 슬라이딩 모드 제어가 구현된다. SoC 밸런싱 전략에 필요한 오차  $e_i = [e_i^s, e_i^{ds}]^T$ 는  $e_i = z_i - \bar{z}_i^{cstrk}$ 과 같이 구해진다. 로컬 슬라이딩 모드 제어는 에러  $e_i$ 를 0으로 조절한다. 따라서 일정 기간이 지난 후 각 클러스터에서  $z_i = \bar{z}_i^{cstrk}$ 와 SoC 균형이 달성된다.

**4) Voltage Restoration:** 전압 복원 제어와 SoC 밸런싱의 주된 차이점은 전압 제어에 기준이 필요하다는 것이다. 본 논문에서 제안하는 전압 복원 제어는 microgrid 전체의 평균 전압을 정격 microgrid 전압 기준  $v_{mg}$ 로 조정한다. 클러스터링 알고리즘은 특징 상태  $x_i$ 를 기반으로 동적으로 변화하기 때문에 microgrid에 분포된 모든 BESS의 전압이 함께 제어된다. 이것은 모든 BESS가 전압 기준에 접근할 수 있음을 보장한다. 따라서 적어도 하나의 BESS는 가정 1에 따른 정격 microgrid 전압 기준에 의해 고정되어야 한다.

**Assumption 1:** 전압 조절 통신 그래프  $\mathcal{G}_v$ 는 무방향으로 연결되어 있다. 적어도 하나의 에이전트는 참조  $v_{mg}$ 에 접근할 수 있다.

평균 전압을 정격값으로 유지하기 위해 [40]의 observer를 이용하여, 여기서  $v_i$ 와  $\bar{v}_i$ 는 각각 출력 전압 및  $i$ 번째 BESS의 이웃을 기반으로 한 분산 네트워크의 평균 전압을 구하였다.

$$\begin{aligned} \dot{\bar{v}}_i &= \dot{v}_i + c_1 \sum_{j=1}^N a_{ij} \text{sig}(\bar{v}_j - \bar{v}_i)^{\kappa_1} \\ &+ c_2 \sum_{j=1}^N a_{ij} \text{sig}(\bar{v}_j - \bar{v}_i)^{\kappa_2} \end{aligned} \quad (8)$$

$c_1, c_2 > 0, k_1 = 2\dot{k}_1 - 1, \dot{k}_1 = 1 - (1/\mu), k_2 = 2\dot{k}_2 - 1, \dot{k}_2 = 1 + (1/\mu)$  및  $\mu \geq 2$ 는 관측자의 제어 파라미터이다.  $\text{sig}(x)^k = |x|^k \text{sign}(x)$ ,  $\text{sign}(x)$ 는 부호 함수이고  $k$ 는 실수이다.

그러면, 통신 그래프 상에 인접한 변환기들로부터의 측정치들을 이용한 전압 조절  $u_i^v$ 에 대한 제어 신호는

$$u_i^v = \alpha_v \left( \sum_{j=1}^N a_{ij} \text{sig}(\bar{v}_j - \bar{v}_i)^{p_v} - g_i \text{sig}(e_i^{\bar{v}})^{p_v} \right) + \beta_v \left( \sum_{j=1}^N a_{ij} \text{sig}(\bar{v}_j - \bar{v}_i)^{q_v} - g_i \text{sig}(e_i^{\bar{v}})^{q_v} \right) \quad (9)$$

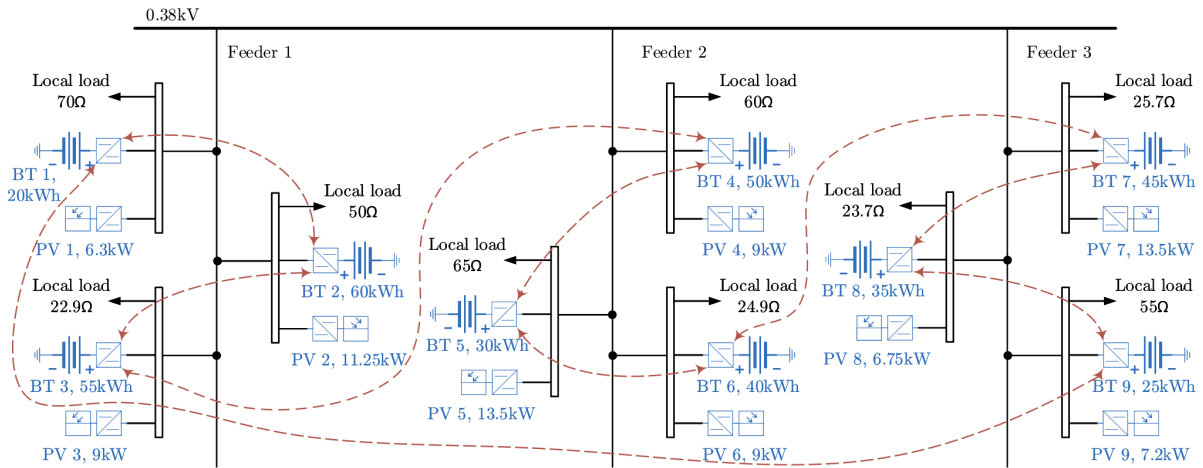
에 의해 주어지고, 여기서  $\alpha_v, \beta_v > 0$ ,  $p_v = 2\acute{p}_v - 1$ ,  $\acute{p}_v = 1 - (1/\mu_v)$ ,  $q_v = 2\acute{q}_v - 1$ ,  $\acute{q}_v = 1 + (1/\mu_v)$  및  $\mu_v \geq 2$ 는 제어 파라미터들이다.  $e_i^{\bar{v}}$ 는  $e_i^{\bar{v}} = v_{mg} - \bar{v}_i$  [40]로 정의된 오류이다.

## V. RESULTS

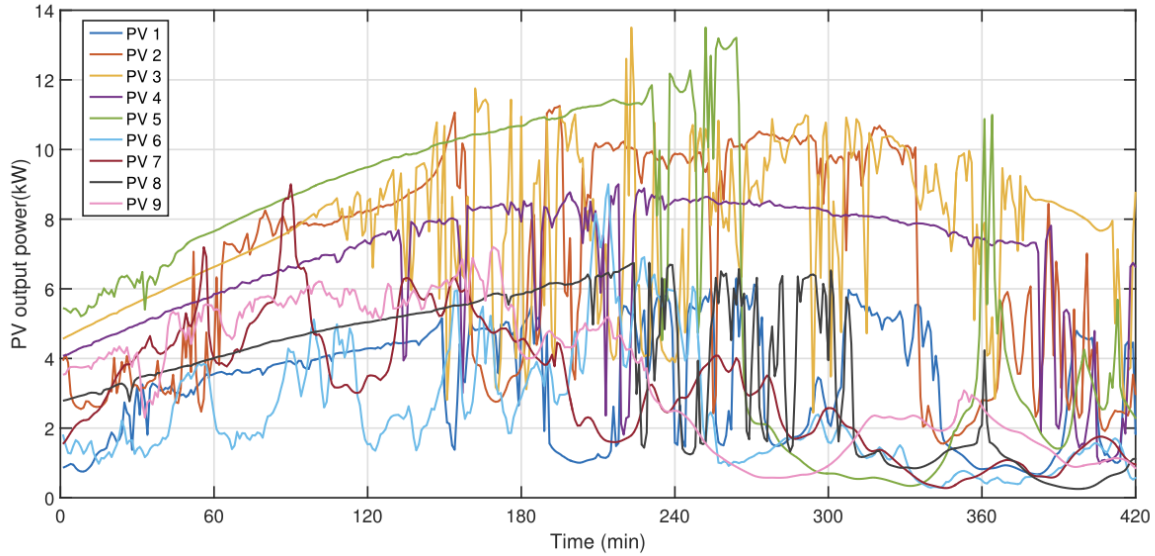
클러스터링 알고리즘은 dc microgrid에서 종합적으로 검증된다. 제안된 클러스터링 알고리즘이 일반적임을 설명하기 위해 기존의 microgrid에도 적용된다.

### A. Verification of the Algorithm

제안된 클러스터링 알고리즘을 검증하기 위해 Fig.5와 같이 3개의 feeder를 갖는 마이크로그리드를 사용하였다.



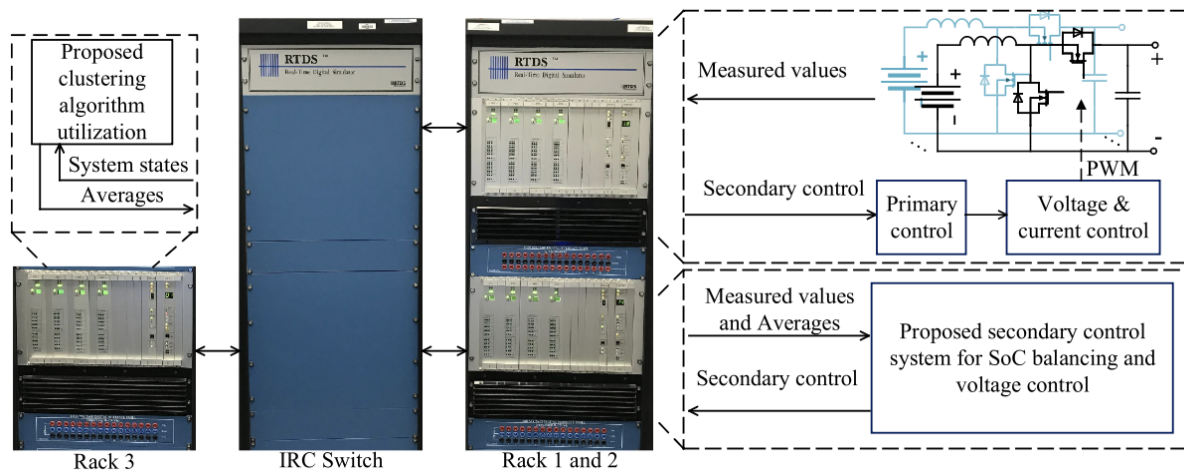
microgrid는 9개의 BESS와 9개의 PV 소스로 구성된다. 각각의 BESS는 Fig.5에 명시된 바와 같이 다른 local load와 배터리 용량을 갖는다. 정격 microgrid 전압은 0.38kV이다. PV 발전원 최대 출력점은 콜로라도의 NREL Baseline Measurement Station에서 one-minute resolution으로 일사량 및 온도 데이터를 기반으로 계산되었다. 2015년 7월 1일부터 7월 9일까지 오전 8시 5분부터 오후 15시 5분까지 자료를 수집하였다. 정격 PV 전력의 범위는 6.3 kW ~ 13.5 kW이며, Fig.11에서 보여주고 TABLE IV에 요약되어 있다. 통신 네트워크 결합 강도  $\alpha = 10$ 이다.



**TABLE IV**  
**AVERAGE PV OUTPUT POWERS (kW)**

Time	1h	2h	3h	4h	5h	6h	7h
PV1	2.2711	3.8719	4.2871	3.1786	3.3631	3.7612	1.9172
PV2	3.3348	7.6648	7.0593	8.8551	9.7023	6.5549	3.8421
PV3	5.6164	7.6481	7.9932	7.0700	8.9324	8.9685	7.4256
PV4	4.9721	6.6059	7.6773	7.5776	8.4380	7.8566	4.7360
PV5	6.3430	8.6514	10.162	10.838	5.7490	1.1066	3.3324
PV6	2.0113	2.9611	2.8904	5.1061	2.0906	1.0040	0.9025
PV7	3.7937	4.8608	5.1402	2.7750	2.9306	1.0443	1.0915
PV8	3.3820	4.5834	5.4632	5.7867	4.1859	1.6552	0.7597
PV9	4.1567	5.6020	5.9074	4.1457	1.0512	2.1455	1.4051

microgrid는 Fig. 6과 같이 RTDS 실시간 디지털 시뮬레이터에서 dc-dc converters의 스위칭 모델, 비선형 배터리 모델, 제어 시스템 및 클러스터링 알고리즘과 함께 구현된다.



dc microgrid와 로컬 1차 제어 시스템은 하나의 RTDS rack에 구현된다. 분산 2차 제어 시스템은 다른 랙에 구현되고, 제안된 분산 클러스터링 알고리즘은 마지막 rack에 구현된다. 3개의 RTDS rack은 IRC 스위치를 통해 상호 작용됩니다. 전체 시스템은 dc-dc 컨버터 및 전력 네트워크 모델의 경우  $1.4\mu s$  시간 단계, 모든 제어 시스템 및 클러스터링 알고리즘의 경우  $50\mu s$  시간 단계의 두 가지 다른 시간 단계로 모델링된다. 슬라이딩 모드 제어 파라미터, 전압 복원 제어 파라미터 및 BESS의 정격 전력은 각각 TABLE I, II 및 III에 제시되어 있다.

**TABLE I**  
**SLIDING MODE CONTROL PARAMETERS [40]**

$m_1$	5	$n_1$	3	$m_2$	7
$n_2$	5	$m_3$	7	$n_3$	3
$\alpha_1$	0.9	$\beta_1$	$3.5453 \times 10^{-6}$	$r$	0.9
$k_1$	1.0431	$k_2$	$5.1879 \times 10^{-7}$	$p_1$	3
$q_1$	5				

**TABLE II**  
**VOLTAGE RESTORATION CONTROL PARAMETERS [40]**

$p_v$	0.6	$q_v$	1.4	$\alpha_v$	25
$\beta_v$	35	$c_1$	$9.5 \times 10^{-7}$	$c_2$	$1.85 \times 10^{-6}$
$\kappa_1$	0.75	$\kappa_2$	1.25		

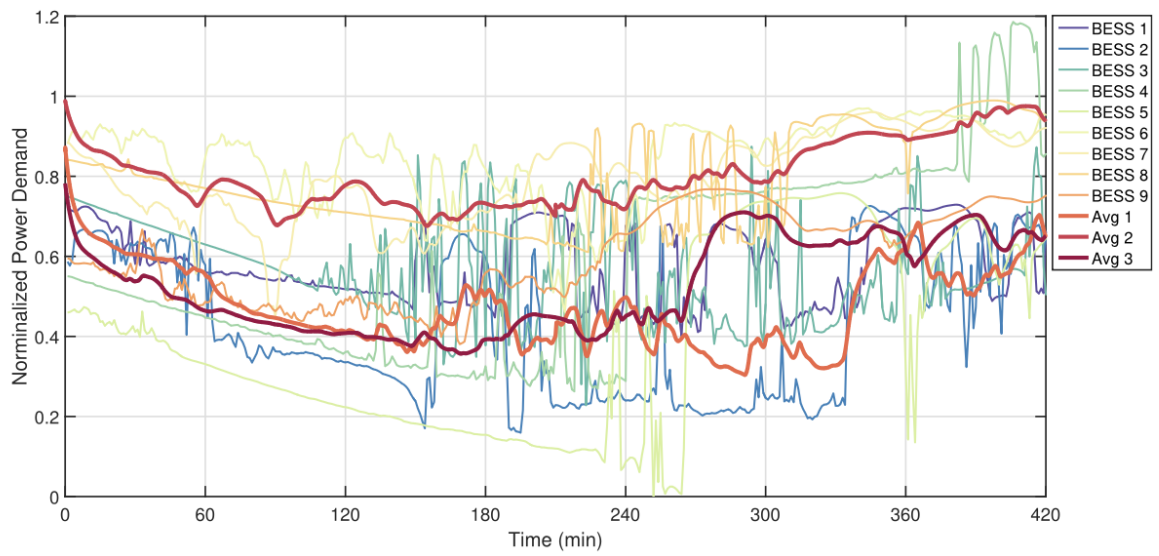
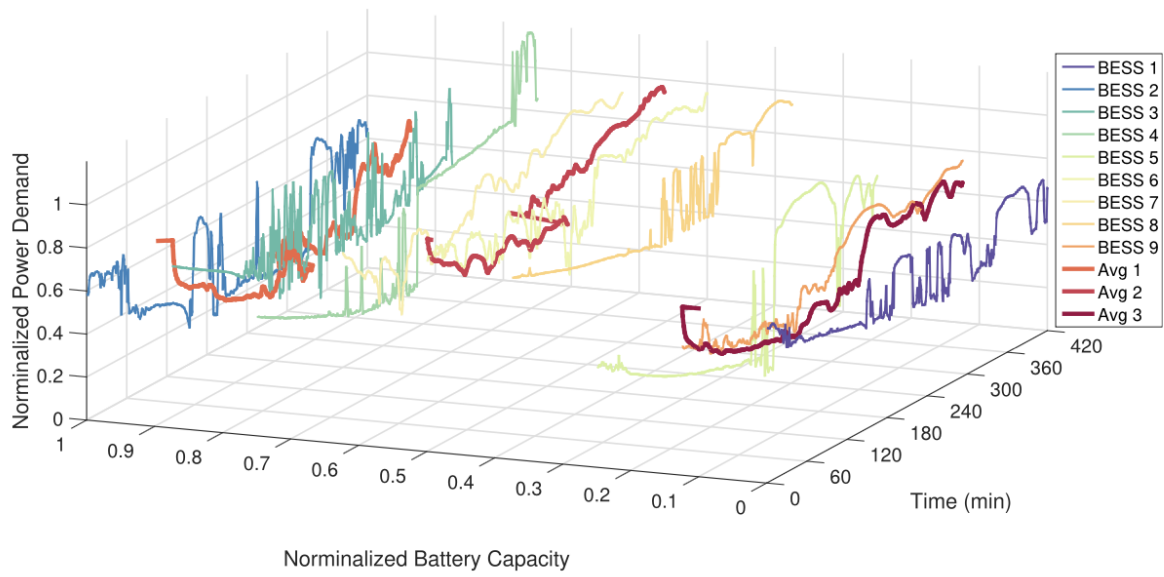
**TABLE III**  
**BESSs RATED OUTPUT POWERS (kW)**

BESS 1	7.60	BESS 2	22.8	BESS 3	17.1
BESS 4	19.0	BESS 5	11.4	BESS 6	15.2
BESS 7	20.9	BESS 8	13.3	BESS 9	9.50

BESS 4의 local load는  $t = 210$  min에서 2.41 kW에서 10.2 kW로 증가한다. PV 출력 전력  $P_{pv}$ 는 재클러스터링 빈도를 줄이기 위해 240s의 시간 상수를 갖는 1차 저역 통과 필터를 적용하였다.

제안된 알고리즘의 실시간 성능은 Fig. 7-10에 도식화되어 있다.







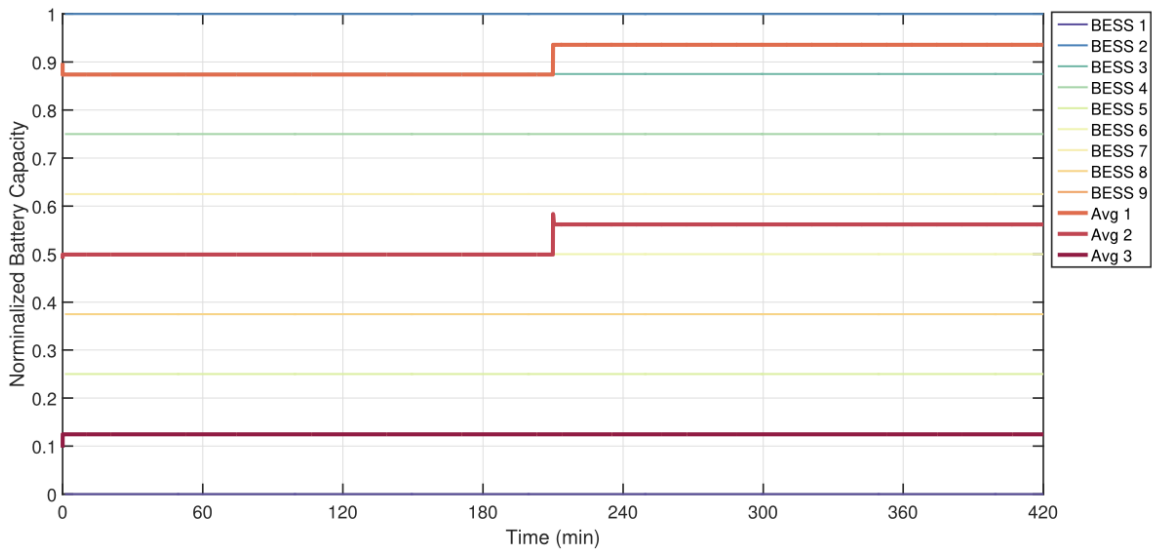
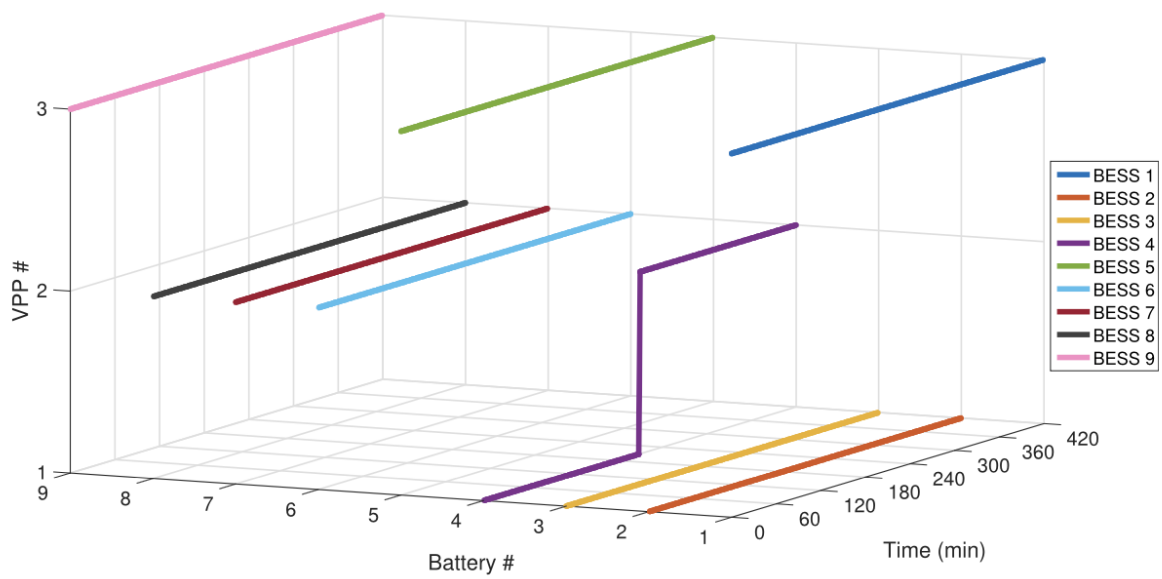


Fig. 7-9는 전력 수요 및 배터리 용량 특징 상태와 이들의 평균 추정치(두꺼운 선)의 발전을 보여준다.

운영 중에, 각 BESS(에이전트)는 분산 통신망을 통해 평균 추정치를 수신하고, 이를 자신의 로컬 특징(얇은 선)과 비교하여 VPP(클러스터)의 멤버십을 결정한다. 전력 수요의 변동에 따라 평균 추정치도 동적으로 변화하고 있습니다. 따라서 로컬 특징과 평균 추정 사이의 거리도 달라져 동적 클러스터링이 발생한다. 형상 상태의 저역 통과 필터링을 통해 재클러스터링의 빈도를 줄일 수 있다. 또한, 전지 용량이 고정되어 있기 때문에, Fig. 9의 얇은 선이 일정하다. 그러나 시간  $t = 210$  min에서는 BESS 4의 부하의 큰 변화에 의한 재클러스터링에 의해 굵은선(평균 추정치)이 변화한다. Fig. 10은 VPP의 멤버십 변화를 보여준다. 전력 수요 특징 상태는 PV 소스 전원과 로컬 부하가 모두 변화하기 때문에 다양하다. PV 소스 전원이 계속 변경되지만  $t=210$ 분까지는 재클러스터되지 않습니다.  $t=210$ 분에서 큰 부하 변동이 있은 후에야 VPP의 회원 자격이 변경된다.

Fig. 10에 도식화된 바와 같이, 제안된 분산 클러스터링 알고리즘은 초기에 BESS를 다음의 3개의 VPP(클러스터)로 클러스터링하였다.



1) VPP 1: BESS 2, BESS 3, BESS 4

2) VPP 2: BESS 6, BESS 7, BESS 8

3) VPP 3: BESS 1, BESS 5, BESS 9

t=210 min에서 BESS 4의 local load가 증가한다.

BESS 4는 클러스터1에서 클러스터2로 이동하며 새로운VPP는

1) VPP 1: BESS 2, BESS 3

2) VPP 2: BESS 4, BESS 6, BESS 7, BESS 8

3) VPP 3: BESS 1, BESS 5, BESS 9

TABLE V  
AVERAGE BATTERY OUTPUT POWERS (kW)

Time	1h	2h	3h	4h	5h	6h	7h
BESS1	-2.5681	-2.8972	-3.3830	-2.8937	0.3068	1.1557	1.3600
BESS2	-0.9495	-2.3639	-2.5281	-3.0111	-5.4101	-4.4658	-1.8229
BESS3	4.4862	-2.5829	-2.5352	-2.4718	-5.1000	-4.2439	-1.5997
BESS4	-6.1866	-1.6136	-2.5446	0.4498	2.8315	5.1018	5.0577
BESS5	-1.0349	-4.5631	-5.1349	-4.5649	1.1035	1.1862	2.4694
BESS6	2.4227	0.9737	0.9374	0.2469	2.9974	3.8951	4.0166
BESS7	4.7895	1.3711	0.8645	1.1280	2.3641	4.6050	4.5659
BESS8	-1.3592	1.0508	0.6071	1.2569	1.2800	3.5005	3.6156
BESS9	2.7454	-3.5053	-4.1974	-2.8842	-0.2832	0.9782	2.3469

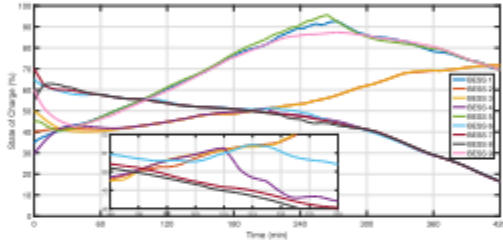


Fig. 12. Battery SoCs. BESSs maintain balanced SoCs within their clusters in accordance with the algorithm. Dynamic reclustering occurs at  $t = 210$  min when the BESS 4 leaves cluster 1 and joins cluster 2 due to its load increasing from 2.41 kW to 10.2 kW.

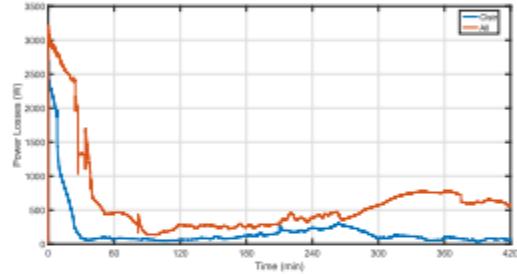


Fig. 14. Power losses due to line resistances. Comparison of the case when the batteries are clustered into three VPPs using the proposed clustering algorithm and when all batteries form one fixed VPP (cluster).

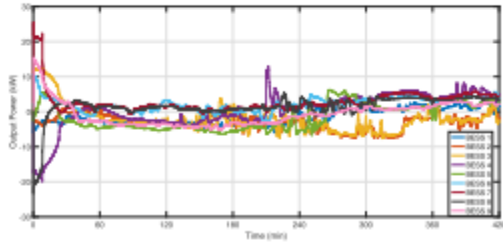


Fig. 13. Battery dc-dc converters output powers. After balanced SoCs are achieved within each cluster (VPP), the BESSs within the same cluster deliver/absorb similar powers in order to reduce the network power losses.

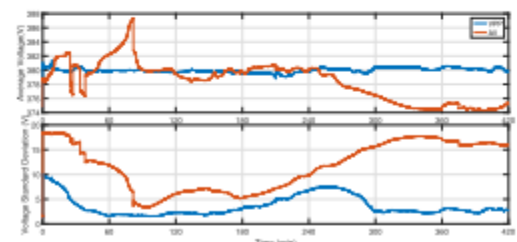


Fig. 15. Average microgrid voltage and its standard deviation. Comparison of the case when the batteries are clustered into three VPPs (clusters) using the proposed clustering algorithm and when all batteries form one fixed (cluster).

Fig. 12 - 13은 각각 배터리들의 SoC과 dc-dc converter 출력 전력들을 보여준다. TABLE V는 배터리 평균 출력 전력들을 보여준다. Fig. 12를 보면 모든 3개의 VPP(clusters)는 약 50분 후 그들 스스로 균형이 잡힌 배터리 SoC을 이룬다. 다시 클러스터링한 후, 균형이 잡힌 배터리 SoC은 역시 유지된다.

Fig. 14는 모든 배터리가 수정된 VPP로 그룹화되어졌을 때의 경우와 비교하면서 전력 손실에서 감소를 도식화한다. 전력 손실들은 현재 선에서 감소로 인해 감소되어진다.

추가적인 결과로서, 현재 라인에서 감소는 저항 선에 따른 voltage drops는 감소한다. 그리고 향상된 voltage regulation(전압 규정)을 초래한다. 결과적으로 microgrid 전압들은 표준 microgrid 전압 기준 값인 0.38 kV에 더 가까이 유지될 수 있다. Fig. 15는 제안된 클러스터링 알고리즘이 적용되고 모든 배터리들이 수정된 VPP에 그룹화 될 때 평균 microgrid 전압과 그것의 표준 편차를 비교한다.

추가적인 이점으로서, VPP에 제안된 클러스터링은 또한 배터리 수명을 향상시킬수 있다. 배터리 수명은 [41]에서 제안된 가중 AH 처리량 모델을 통해 정의될 수 있습니다. 제안된 클

러스터링 알고리즘을 사용하여 VPPs로 클러스터링된 모든 배터리와 하나의 고정된 VPP로 클러스터링된 모든 배터리의 정규화된 ESS 수명은 Fig. 16 - 17에 나타나 있습니다.

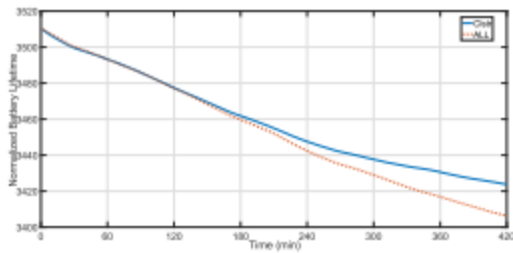


Fig. 16. Normalized ESS lifetimes of all batteries operating either clustered into three VPPs using the proposed clustering algorithm or all clustered into one fixed VPP.

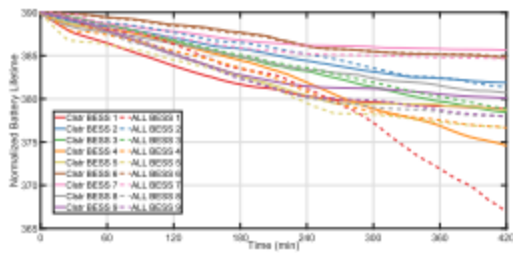


Fig. 17. Individual normalized ESS lifetimes of all batteries operating either clustered into three VPPs using the proposed clustering algorithm or all clustered into one fixed VPP.

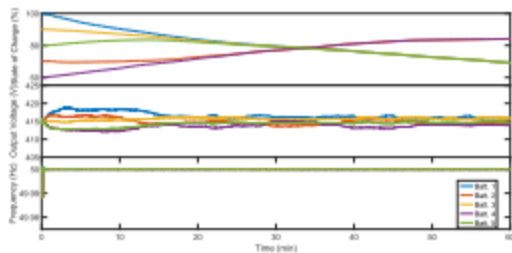


Fig. 18. Application of the proposed algorithm to an existing ac microgrid and control structure from [40]. Five BESS-PV sources are clustered into two VPPs based on the local power demands and battery capacities. Each VPP maintains its SoC balance while both the voltage and frequency regulation is maintained.

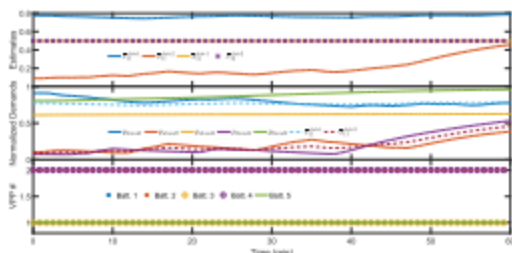


Fig. 19. Average estimations, normalized power demands and VPPs for the clustering algorithm applied to an ac microgrid.

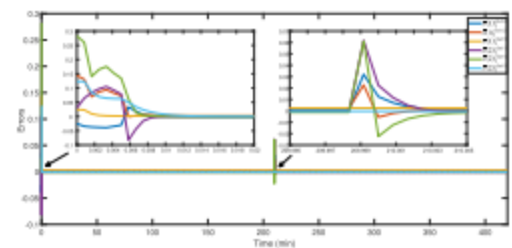


Fig. 20. Differences (errors) between the average estimations obtained by the distributed algorithm and the average states obtained by the centralized algorithm.

## B. Verification of Clustering Algorithm in AC microgrid

이 제안된 클러스터링 알고리즘이 일반적이며, microgrid 유형과 제어 전략과 독립적임을 보여주기 위해, 이 경우의 연구는 논문 [40]에서 제시된 기존 AC microgrid와 제어 시스템에 해당 클러스터링 알고리즘을 적용합니다. [40]에 나온 AC microgrid는 VSCs를 통해 연결된 다섯 개의 BESS-PV(Battery Energy Storage System - Photovoltaic) 자원을 포함하고 있습니다. VSC의 정격(규격) 출력 전압은 0.415 킬로볼트이며 정격(규격) 주파수는 50 헤르츠입니다. 이 차수 수준의 제어에는 주파수 복원, SoC 균형, 평균 전압 복원 및 무효 전력 공유가 포함됩니다. 결과는 Fig. 18 - 19에 나와 있습니다. 이 알고리즘은 로컬 전력 수요와 배터리 용량을 기반으로 다섯 개의 BESS-PV 소스를 두 개의 VPP(Virtual Power Plant)로 클러스터링합니다.

## C. Comparison With Centralized Clustering Algorithm

중앙집중식 클러스터링 알고리즘에서, 각 BESS는 중앙 프로세서와 중앙 집중식 통신 네트워크를 통해 직접 통신합니다. 각 시간 단계에서 BESS는 배터리 용량과 전력 수요와 같은 특성 상태를 중앙 프로세서로 보냅니다. 그런 다음 중앙 프로세서는 각 VPP의 평균 상태를 계산하며, 이는 분산 알고리즘의 분산 평균 관측자(Distributed Average Observer)에 의한 평균 추정과 유사합니다. BESS의 특성 상태와 평균 상태를 비교하여 VPP의 소속을 결정합니다. BESS는 자신의 평균 상태가 BESS의 특성 상태에 가장 가까운 VPP에 가입합니다. 클러스터링 결과는 BESS로 다시 보내지며, BESS는 자신의 제어 시스템을 조정합니다. 중앙집중식 클러스터링 알고리즘은 Case Study A에서 검증되었습니다. 얻은 클러스터링 결과는 제안된 분산 클러스터링 알고리즘의 Fig. 10에 표시된 것과 동일합니다. 분산 알고리즘에 의한 평균 추정과 중앙집중식 알고리즘에 의한 평균 상태 간의 차이(오류)는 Fig. 20에 표시됩니다. 초기 클러스터링 중의 큰 오류 및 클러스터 변동은 평균 상태 추정의 수렴으로 인한 것입니다.

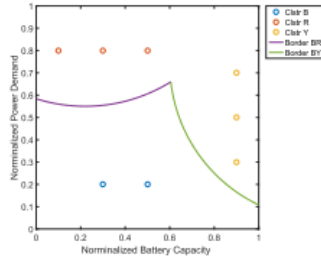


Fig. 21. Borders for a varying blue cluster agent to join the red or yellow cluster.

TABLE VI  
BIG-O NOTATION OF ALGORITHM 1

O (1)	Size	Comment
$O_{(4)}(1)$	$M$	cluster determination (4)
$O_{clstr\_rslt\_a}(1)$	1	output $clstr\_rslt\_a$
$O_{clstr\_rslt\_b}(1)$	$M \times N_{ftr}$	output $clstr\_rslt\_b$
$O_{(2,3)2}(1)$	$M \times N_{ftr} \times  \mathcal{N}_i $	second terms of (2)-(3)
$O_{(2)1}(1)$	$N_{ftr}$	first term of (2)
$O_{(3)1}(1)$	$(M - 1) \times N_{ftr}$	first term of (3)
$O_{(3)1avg}(1)$	$(M - 1) \times N_{ftr} \times  \mathcal{N}_i $	calculate the average

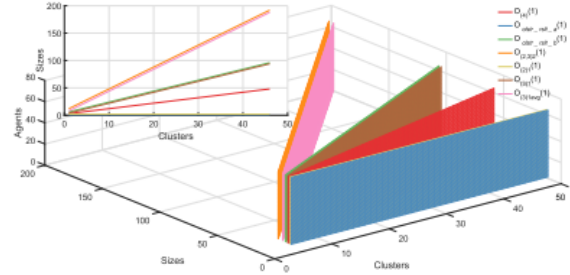


Fig. 22. Analysis of the distributed clustering algorithm for different number of agents and clusters. Big-O notation is implemented to indicate the algorithm complexity.

## VI. DISCUSSIONS

### A. Robustness to Frequent Cluster Changes

이동하는 에이전트가 다른 클러스터에 들어가는 경계는 다음과 같은 함수로 설명할 수 있습니다.

$$\begin{aligned} \|x_i^{bj} - \bar{x}^{cstrk}\| - \|x_i^{bj} - \bar{x}^{cstrj}\| &= 0 \\ 0 \leq x_i^{bj} \leq 1, j, k \in M, j \neq k \end{aligned} \quad (10)$$

여기서  $X_i^{bj}$ 는 k번째 클러스터의 i번째 에이전트가 j번째 클러스터에 가입하는 경계이며,  $\bar{x}^{cstrk}$ 와  $\bar{x}^{cstrj}$ 는 각각 k번째 클러스터와 j번째 클러스터의 중심 (평균 값)입니다. 그런 다음, 에이전트와 경계 사이의 거리는 다음과 같이 계산할 수 있습니다.

$$\rho_{ij} = \|x_i - x_i^{bj}\|, j \in M, j \neq k \quad (11)$$

여기서  $P_{ij}$ 는 i번째 이동 에이전트와 j번째 클러스터의 경계 사이의 거리입니다. 이 거리는 특성 상태의 어떤 변화가 클러스터 소속의 변경으로 이어질 수 있는지를 나타냅니다.

파란색, 빨간색, 노란색 세 개의 클러스터가 있다고 가정하고, 파란색 클러스터에 속한 하나의 에이전트의 특성 상태가 변하면서 다른 에이전트의 특성 상태는 고정된다고 가정합니다. 파란 에이전트가 빨간색 클러스터 또는 노란색 클러스터에 가입하기 위한 경계는 각각 그림 21의 보라색 실선과 녹색 실선으로 표시됩니다. 파란색 에이전트가 보라색 경계를 넘어서면 빨간색 클러스터에 가입하게 되고, 녹색 경계를 넘어서면 노란색 클러스터에 가입하게 됩니다. 에이전트가 다른 클러스터에 가입하면 경계가 변경되며, 이는 클러스터 변경의 가능성을 줄입니다. 또한, 특성 상태에 대한 필터가 도입되면 갑작스러운 클러스터 변경을 줄일 수 있습니다.

## B. Scalability(확장성)

클러스터링 알고리즘의 복잡성을 분석하기 위해 빅-O 표기법이 사용됩니다 [42]. 표 VI에는 클러스터링 알고리즘의 파티션에 대한 빅-O 표기법이 표시되어 있으며, 여기서  $N_{ftr}$ 은 특성의 수를 나타냅니다. 이 논문에서는  $N_{ftr} \equiv 2$ 입니다. 알고리즘이 분산되기 때문에 가정 2가 만족되는 한 몇 개의 이웃  $|N_i|$ 만 있는 희소 통신 네트워크를 구현할 수 있습니다. 이 논문에서는  $|N_i| \equiv 2$ 입니다. 모든 빅-O 표기법의 합계(크기와 곱해진 '1차 빅-O 표기법')은 각 에이전트에 대한 Algorithm 1의 복잡성을 나타냅니다. 표 VI에서 M과  $|N|$ 는 알고리즘 복잡성을 결정하며, 알고리즘 복잡성은 에이전트 수 N과는 독립적입니다. 따라서 클러스터 수 M만 알고리즘 복잡성을 정의합니다. 특정 클러스터 및 에이전트 수에 대한 빅-O 표기법은 Fig. 22에 표시되어 있습니다.

알고리즘 2의 빅-O 표기법은 방금 언급한 알고리즘 1과 동일하며 차이점은 이용 상태 방정식 (5)와 (6)이 추가되었다는 것뿐입니다.

## C. Advantages of Real-Time Distributed Clustering Algorithm

제안된 분산 알고리즘의 주요 장점은 중앙집중식 해결책과 비교했을 때 다음과 같습니다.

1. 총 계산 부담이 로컬 프로세서 사이에 분산되어 전체 비용을 감소시킵니다.
2. 이웃 간 통신을 기반으로 하는 방식은 단일 고장 지점을 제거합니다.
3. 분산 알고리즘은 플러그 앤 플레이 기능(별도의 물리적인 설정을 하지 않아도 설치만 하면 그대로 사용할 수 있도록 하는 기능)을 제공하며 확장 가능합니다.
4. 분산 알고리즘을 위한 통신 네트워크는 희소하며, 이로써 인프라 비용이 더욱 감소합니다.

이 알고리즘은 특성 상태의 어떤 변화든 즉시 평균 추정치(전통적인 클러스터링 알고리즘에서의 중심점)에 영향을 미치므로 실시간으로 작동합니다. 전통적인 클러스터링 알고리즘에서는 과거(역사적) 상태가 미래 클러스터링 결과를 결정하는 데 사용되므로 상태 예측 정확도에 영향을 받습니다.

---

## VII. CONCLUSION

이 논문에서는 사전 지정된 특성 상태를 기반으로 한 분산, 동적, 실시간 클러스터링 알고리즘을 제안했습니다. 이 클러스터링 알고리즘은 비슷한 전력 수요와 용량을 가진 배터리를 VPP(Virtual Power Plant)로 클러스터링하여 microgrid의 선전류 및 전력 손실을 줄이기 위해 적용되었습니다. RTDS 실시간 구현을 통해 제안된 클러스터링 알고리즘과 관련된 제어 시스템의 효과를 검증했습니다. 제안된 클러스터링 알고리즘은 범용적이며, 사전 지정된 특성 상태를 기반으로 실시간 클러스터링이 필요한 모든 시스템에 적용할 수 있으며 필요한 제어, 최적화 또는 분석을 위해 보조 상태를 활용할 수 있습니다.

이 클러스터링 알고리즘은 에너지 거래를 위한 상업용 VPP(Virtual Power Plant) 형성과 같은 다른 시나리오에도 적용할 수 있습니다. 분산 클러스터링 알고리즘은 유연한 에너지 저장 기반 VPP를 형성하여 에너지 거래, 전력 할당 및 전력 흐름 최적화를 위해 사용될 수 있습니다. 이러한 목표를 달성하기 위해 제안된 알고리즘은 협력적이거나 비협력적인 게임과 경매를 모두 활용하여 분산 네트워크를 통해 프로슈머(생산자 및 소비자) 및 마이크로그리드 운영자에게 이익을 제공할 수 있습니다. 이 클러스터링 알고리즘은 특히 유사성을 기반으로 여러 VPP로 분리할 수 있는 대규모 분산 에너지 자원을 가진 전력 네트워크에 적합합니다.