

## 무선 IoT 그리드 네트워크에서의 분산 주성분 분석 기법

허 홍준\*, 손경락\*, 최 완°

## Distributed Principal Component Analysis for Wireless IoT Grid Networks

Hongjun Heo\*, Kyungrak Son\*, Wan Choi°

## 요 약

본 논문에서는 무선채널 환경에서 양 방향 행렬 분할을 통한 분산 주성분 분석 알고리즘의 성능을 평가하였다. 데이터 행렬을 행 방향과 열 방향의 양 방향으로 분할하면, 각 장치에 할당되는 행렬의 차원을 더 작게 줄일 수 있으며, 주변 환경에 맞춰 유연하게 크기를 조절할 수 있다는 장점이 있다. 이 때, 각 장치 간 통신을 통해 전체 행렬의 주성분을 추정하게 되는데, 통신 채널이 무선채널이라면 채널 페이딩과 잡음으로 인해 본래의 분산 주성분 분석 알고리즘이 동작을 하지 않는다. 따라서, 무선채널 효과를 최소화하기 위해 시간 다이버시티 (time diversity) 기법을 이용하고, 수신 노드에서 최우도 결합기 (maximal ratio combiner)를 이용하여 신호를 취합하는 방식을 적용하였다. 본 논문은 신호 결합 기법과 다이버시티, 거듭제곱법, 평균 합의 알고리즘 개념을 활용하여, 무선 채널에서 거듭제곱법 기반의 분산 주성분 분석을 제안하고, 시뮬레이션을 통해 무선 그리드 네트워크에서 분산 주성분 분석이 수렴한다는 것을 확인한다.

**Key Words** : Wireless Distributed PCA, Grid Networks, Power Method, Average Consensus, IoT

## ABSTRACT

In this study, the performance of the distributed principal component analysis (PCA) algorithm through bidirectional matrix division in a wireless channel environment was evaluated. Dividing the data matrix in both directions into a row direction and a column direction has an advantage of reducing the dimension of the matrix assigned to each device. The assigned matrices can be flexibly adjusted according to computing capability and environments. The principal component of the entire original matrix is estimated through communications between devices, but the existing distributed PCA algorithm cannot be applied to wireless environments due to wireless channel fading and noise. Therefore, in order to address the wireless channel effect, a time diversity scheme and a maximum ratio combiner at each receiving node are exploited. Utilizing signal combining techniques, diversity in the wireless channels, power methods, average consensus algorithm, this paper develops a power method-based distributed PCA for wireless IoT network. Our simulation results show that the distributed PCA still converges to its principal components in wireless grid networks.

※ 이 논문은 정부(과학기술정보통신부)의 재원으로 한국연구재단의 지원을 받아 수행된 연구임(No. NRF-2021R1A6A3A01086690).

※ 이 논문은 정부(과학기술정보통신부)의 재원으로 정보통신기술진흥센터의 지원을 받아 수행된 연구임 (No. 2018-0-00809, 새로운 자원을 활용한 beyond 5G 이동통신 변혁기술 개발).

• First Author : Department of Electrical and Computer Engineering, Seoul National University (SNU), hhj2681@snu.ac.kr, 학생회원

° Corresponding Author : Institute of New Media and Communications and Department of Electrical and Computer Engineering, Seoul National University (SNU), wanchoi@snu.ac.kr, 중신회원

\* Institute of New Media and Communications and Department of Electrical and Computer Engineering, Seoul National University (SNU), kyungrakson@snu.ac.kr, 정회원

논문번호 : 202202-021-C-RN, Received February 7, 2022; Revised April 28, 2022; Accepted May 19, 2022

## I. 서론

빅데이터 분석, 기계학습, 신호처리 등의 어플리케이션에서 데이터는 행렬 혹은 텐서 (tensor)의 형태로 처리된다. 일반적으로 행렬의 각 행 (row)은 특징 (feature) 혹은 공간 (spatial) 차원이며, 각 열 (column)은 표본 (sample) 혹은 시간 차원이다. 한편, 실제 응용에서는 표본의 개수가 상당하며, 고해상도 이미지처리, 자연어처리 등 최근의 어플리케이션에서는 다수의 특징을 포함하는 경우도 빈번하다. 그러나 관련 없는 변수와 특징을 포함한 행렬의 모든 성분으로 연산할 경우 성능과 속도를 떨어뜨리는 원인이 된다. 이를 해결하기 위해, 행렬의 중요한 정보 손실을 최소화하며 차원을 줄이는 주성분 분석 (Principal Component Analysis, PCA)이 패턴 인식, 손실 압축 등에 활용된다<sup>[1]</sup>.

주성분 분석은 거듭제곱법 (power method)<sup>[2]</sup>, Oja's Algorithm<sup>[3]</sup> 등 다양한 알고리즘을 통해 계산된다. 위와 같은 알고리즘은 단순한 행렬 곱과 정규화 (normalization)의 반복적인 연산이지만, 단일 장치에서 이러한 고차원 행렬 연산의 수행은 시간 소모가 크며, 실시간 응용이나 빠른 데이터 전송을 요구하는 어플리케이션에서 속도 저하를 유발하기 때문에 적합하지 않다. 혹은 단일 장치의 저장 공간 제약으로 인해 단일 장치에 고차원의 행렬을 저장하며 연산을 처리하는 것이 어려울 수 있고, 컴퓨팅 파워의 한계로 연산 자체가 불가능할 수 있다. 이러한 이유로, 하나의 연산을 여러 주변 장치에 나누어 수행하는 분산 컴퓨팅 기법에 대한 연구가 활발하게 진행되고 있다<sup>[4]</sup>. 행렬을 적절한 크기로 주변 장치에 나누어 연산을 수행한 뒤, 장치 간의 통신을 통해 본래 행렬의 주성분을 분석하는 분산 주성분 분석 (Distributed Principal Component Analysis, Distributed PCA)에 대한 연구가 활발하게 진행되고 있다<sup>[5],[6]</sup>.

한편, 기존의 분산 주성분 분석에 대한 연구는 행렬을 행 방향으로만 (row-wise), 혹은 열 방향으로만 (column-wise) 나누어 네트워크 토폴로지 (network topology)에 적용하는 방식이 주를 이루었다<sup>[6]</sup>. 그러나 데이터 행렬의 차원이 매우 큰 경우, 행 혹은 열의 한 방향 행렬 분할 방식은 여전히 장치에서 연산 수행이 어려움을 야기한다. 즉, 행과 열 두 방향으로 행렬을 분할하여 네트워크 구조에 적용하는 방안이 필요하며, 최근 거듭제곱법을 기반으로 2차원 그리드 네트워크에서 양 방향 (both row-wise and column-wise) 행렬 분할 방식의 분산 주성분 분석 알고리즘이 제안

되었다<sup>[7]</sup>.

대부분의 분산 주성분 분석에 대한 선행연구는 장치 간 정보 전송이 완벽하게 이루어지는 이상적인 네트워크 위에서 진행되어왔다. 하지만 IoT와 같이 이동성과 장치의 성능제약조건이 중요시되는 환경에서도 주성분 분석은 이상 탐지<sup>[8]</sup> 등을 위해 이용되기에, 무선 채널 환경 위에서 분산 주성분 분석 알고리즘의 성능에 대한 연구가 필요하다.

이러한 배경 아래 본 연구에서는 양 방향 행렬 분할 방식의 분산 거듭제곱법을 바탕으로, 각 장치 사이 링크를 무선 채널로 모델링하여 채널 페이딩과 잡음 (noise)이 존재하는 무선 채널 환경이 연산의 정확도에 미치는 영향을 분석하고자 하였다.

2장에서는 본 연구와 관련된 주성분 분석, 거듭제곱법, 평균 합의 (average consensus) 알고리즘, 분산 거듭제곱법에 대해 다룬다. 이어 시스템 모델에 대해 설명하고, 연구결과를 제시한다. 3장 결론에서는 결과를 요약하고, 향후 과제를 다룬다.

## II. 시스템 모델 및 문제 설정

### 2.1 주성분 분석

주성분 분석<sup>[1]</sup>은 데이터 행렬  $X \in \mathbb{R}^{N \times L}$ 에 대해  $X$ 의 손실을 최소화하는 열 공간의 부분공간 (subspace), 즉 주요 부분공간 (principal subspace)의 기저 (basis)를 찾는다. 주요 부분공간이 정의되면 행렬의 각 열을 투영 (projection)하여 낮은 차원의 벡터로 손실 대응 (mapping)이 가능하다. 그림 1은 주성분 분석의 예를 보여준다. 2차원 데이터를 1차원으로 투영하여 차원을 낮추는 과정이고, 1차원의 부분공간  $y$ 는 정규 직교 기저 (orthonormal basis)  $u$ 에 의해 형성 (span)된다.

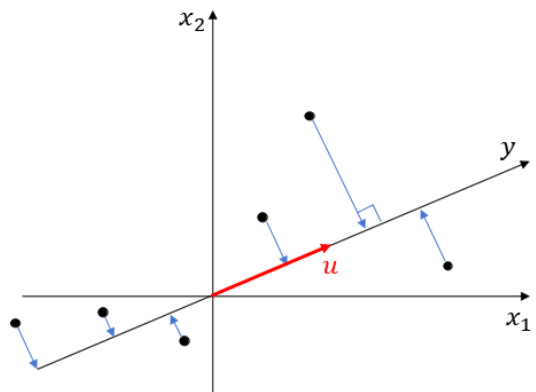


그림 1. 2차원 데이터의 주성분 분석 예시  
Fig. 1. An example of PCA of 2-dimensional data

$p$ -차원 ( $p \leq L$ ) 주요 부분공간으로의 투영은 부분 공간의 정규 직교 기저로 이루어진 직교 (orthogonal) 행렬  $U_p \in \mathbb{R}^{N \times p}$ 에 의한 변환으로 표현할 수 있다. 주 성분 분석 문제는 본래의 데이터와 투영 사이의 평균 제곱오차 (mean square error)를 최소화하는 부분공간을 찾는 것이며, 다음의 최적화 문제와 같다.

$$\begin{aligned} \mathcal{U}^* = \operatorname{argmin}_{\mathcal{U}} & \| (I - \mathcal{U}\mathcal{U}^T)\mathbf{X} \|_F^2 \\ \text{subject to} & \quad \mathcal{U}^T \mathcal{U} = I \end{aligned} \quad (1)$$

목적함수 (objective function)는  $\mathbf{X}$ 와  $\mathcal{U}\mathcal{U}^T\mathbf{X}$ 의 차이를 최소화는  $\mathcal{U}$ 를 찾는 것이며,  $\| \cdot \|_F$ 은 행렬의 Frobenius norm이고,  $I$ 는 항등행렬 (Identity matrix)이다. 제약조건은 행렬  $\mathcal{U}$ 의 열벡터가 서로 정규직교 (orthonormal)함을 의미한다.

한편, 상관행렬 (correlation matrix)  $\mathbf{R} = \frac{1}{L}\mathbf{X}\mathbf{X}^T$ 에 대해 고윳값 분해과정 (Eigenvalue Decomposition)을 수행하여,  $\mathbf{U}\mathbf{A}\mathbf{U}^T$ 로 표현할 수 있다. 여기서  $\mathbf{U}$ 는  $N \times N$  직교행렬이고,  $\mathbf{A} = \operatorname{Diag}([\lambda_1, \dots, \lambda_k, 0, \dots, 0])$ ,  $k = \operatorname{rank}(\mathbf{X})$ ,  $\lambda_1 \geq \dots \geq \lambda_k > 0$ 이다. 주성분 분석을  $p$ -차원 주요 부분공간에 대해 진행할 때,  $\lambda_1 > \dots > \lambda_p > \lambda_{p+1} > \dots > \lambda_k > 0$  조건을 가정하면,  $\mathbf{U}$ 의 왼쪽  $p$ 개 고유벡터 (top  $p$  eigenvector)로 이루어진 부분행렬 (submatrix)은 위 최적화 문제의 해가 된다. 따라서 주성분 분석은 상관 행렬의 가장 큰  $p$ 개의 고윳값에 대응되는 고유벡터를 구하는 문제와 동치이다<sup>6)</sup>.

### 2.2 거듭제곱법

거듭제곱법<sup>2)</sup>은 고유벡터들을 얻기 위한 반복 알고리즘이며, 간단한 행렬 곱으로 쉽게 구현할 수 있기 때문에 주성분 분석에 널리 활용된다. 먼저, 임의의 단위 (unit) 2-norm 벡터  $\mathbf{u}_1[0] \in \mathbb{R}^N$ 를 선택한다. 아래 첨자는 최대 고윳값에 대응되는 고유벡터를 의미하고, 대괄호 안은 알고리즘 반복횟수이다. 그리고 다음의 방식을 따라 unit 벡터를  $K$ 번 업데이트한다.

$$\mathbf{u}_1[k] = \frac{\mathbf{R}\mathbf{u}_1[k-1]}{\|\mathbf{R}\mathbf{u}_1[k-1]\|_2}, \quad 1 \leq k \leq K \quad (2)$$

$K \rightarrow \infty$ 에 따라  $\mathbf{u}_1[k]$ 가  $\mathbf{R}$ 의 최대 고윳값에 대응되는 고유벡터에 수렴한다는 것이 알려져 있다<sup>6)</sup>. 두 번째로 큰 고윳값에 대응되는 고유벡터  $\mathbf{u}_2$ 는 행렬  $\mathbf{R}$ 에서 첫 번째 주성분  $\mathbf{u}_1[k]$ 을 제하고, 위 과정을 반복하여 계산된다.

$$\mathbf{R} \leftarrow (I - \mathbf{u}_1[\mathbf{K}]\mathbf{u}_1[\mathbf{K}]^T)\mathbf{R}$$

같은 방식으로 반복하면 순차적으로  $\mathbf{R}$ 의 가장 큰  $p$ 개의 고윳값에 대응되는 고유벡터, 즉  $p$ 개의 주성분을 얻을 수 있다.

### 2.3 평균 합의 알고리즘

평균 합의 (average consensus) 알고리즘은 네트워크의 부분 상태값 (local state-value)으로부터 전체 평균 (global average)을 추정하는 알고리즘으로 분산 주성분 분석에서 핵심적인 역할을 한다. 하나의 노드가 네트워크 전체 데이터 흐름을 통제하는 마스터-슬레이브 구조에서는 데이터를 취합하고 분배하기 쉽기 때문에, 이하의 네트워크는 마스터-슬레이브 구조 대신 모든 노드가 데이터 흐름과 처리에 있어 협업하는 구조를 가정한다. 이러한 네트워크 구조에서 각 노드는 분산 주성분 분석을 위한 장치에 대응된다.

전체  $S$ 개의 노드를 갖는 네트워크를 고려할 때, 시간  $k$ 에서  $i$ 번째 노드의 상태값을  $z_i[k]$ , 모든 노드의 상태값을 벡터 형태로  $\mathbf{z}[k] = (z_1[k], z_2[k], \dots, z_S[k])^T$ 과 같이 표현할 수 있다. 특정 시점에서 모든 노드의 상태값이 시간 0일 때의 상태값의 평균과 같아졌을 때, 즉,  $\mathbf{z}[k] = (\frac{1}{S} \sum_{j=1}^S z_j[0])\mathbf{1}$ 을 만족하면 평균 합의에 도달하였다고 한다.  $\mathbf{1}$ 는 모든 성분이 1인 열벡터이다.

노드 간의 연결이 존재하는 경우에만 노드 간 상태값을 주고받을 수 있다고 가정하면, 평균 합의 알고리즘은 다음의 규칙으로 각 노드의 상태값을 업데이트한다.

$$\mathbf{z}[k+1] = \mathbf{W}\mathbf{z}[k] \Leftrightarrow z_i[k+1] = \sum_{j=1}^S \mathbf{W}_{ij}z_j[k] \quad (3)$$

$\mathbf{W}$ 는 가중치 행렬 (Weight matrix)로, 노드  $i$ 에서 노드  $j$ 의 상태값에 대해  $\mathbf{W}_{ij}$ 만큼의 가중치를 곱하여 합산한다. 노드  $i$ 와 노드  $j$ 사이의 연결이 없으면  $\mathbf{W}_{ij} = 0$ 이다. 이 때, 가중치 행렬이 다음 두 조건을 만족하는 경우 평균 합의로 수렴이 보장된다<sup>9)</sup>.

- 1)  $\mathbf{W}$ 가 이중 확률 (doubly stochastic) 행렬  
 $\Leftrightarrow \mathbf{W}\mathbf{1} = \mathbf{1}, \mathbf{1}^T\mathbf{W} = \mathbf{1}^T$
- 2)  $\rho(\mathbf{W} - \frac{\mathbf{1}\mathbf{1}^T}{S}) < 1$

$\rho(\cdot)$ 는 행렬의 스펙트럼 반지름 (spectral radius)

이며, 후술할 제안 기법에서 가중치 행렬은 위 두 조건을 만족한다고 가정한다.

2.4 거듭 제곱법 기반의 분산 주성분 분석

그림 2와 같이 행과 열의 양 방향 (both row-wise and column-wise) 행렬 분할을 이용하면 주변 장치의 수와 컴퓨팅 파워 등을 고려하여 부분행렬의 크기를 유동적으로 조정할 수 있다.

이때, 전체 데이터 행렬의 주성분을 구하기 위해서는 평균 합의 알고리즘을 이용하여 거듭제곱법을 변형해야 한다. 최근 이와 관련된 연구가 제안되었고, 몇 가지 조건하에서 최대 고품값에 대응되는 고유벡터로 수렴하는 것이 확인되었다기.

각 노드는 이웃한 노드와 평균 합의 알고리즘을 통해 통신을 한다고 가정한다. 또한 평균 합의 알고리즘은 동기화 (synchronization) 방식으로 이루어진다.

- 1) 데이터 행렬  $X \in \mathbb{R}^{N \times L}$  은  $I \times J$  개의 부분행렬  $X_{ij} \in \mathbb{R}^{N_i \times L_j}$  ( $i = 1, 2, \dots, I, j = 1, 2, \dots, J$ ,  $\sum_{i=1}^I N_i = N, \sum_{j=1}^J L_j = L$ )로 나뉘고, 부분행렬  $X_{ij}$  는 노드  $(i, j)$  에 할당된다.
- 2) 각 노드  $(i, j)$  에서 단위 2-norm 벡터  $u_{ij} \in \mathbb{R}^{N_i}$  를 선택한다.
- 3) 각 노드  $(i, j)$  에서  $p_{ij}$  를 다음과 같이 계산한다.  

$$p_{ij} \leftarrow X_{ij}^T u_{ij} \in \mathbb{R}^{L_j}$$
 ←는 ←기준 우측의 값을 좌측의 변수에 새로운 값으로 대입함을 의미한다.

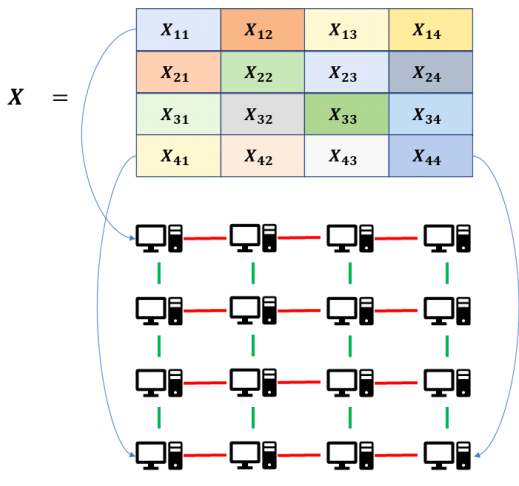


그림 2. 양방향 행렬 분할과 장치의 그리드 배치  
 Fig. 2. Bidirectional matrix segmentation and grid placement of devices

- 4) 열별로, 즉 각  $j \in \{1, 2, \dots, J\}$  에 대해  $I$  개의 노드  $(1, j), \dots, (I, j)$  간에 평균 합의 알고리즘을  $M_c$  회 수행한다. 평균 합의 알고리즘 업데이트 식 (3)에 의해  $p_{ij} \leftarrow \sum_{i'=1}^I W_{(i,j),(i',j)} p_{i'j}$  를  $M_c$  회 반복하는 것이며, 이웃한 노드 사이에만 링크가 존재하기 때문에 다음의 식으로 쓸 수 있다.

$$p_{ij} \leftarrow \begin{cases} W_{(i,j)(i,j)} p_{ij} + W_{(i,j)(i+1,j)} p_{i+1,j} & i = 1 \\ W_{(i,j)(i-1,j)} p_{i-1,j} + W_{(i,j)(i,j)} p_{ij} + W_{(i,j)(i+1,j)} p_{i+1,j} & 2 \leq i \leq I-1 \\ W_{(i,j)(i-1,j)} p_{i-1,j} + W_{(i,j)(i,j)} p_{ij} & i = I \end{cases}$$

$W$ 는 각 열의 링크에 대한 가중치 행렬로,  $W_{(i,j),(i',j)}$  는 노드  $(i, j)$  에서  $(i', j)$  에 대한 양의 가중치를 의미하며,  $\sum_{i'=1}^I W_{(i,j),(i',j)} = 1$  를 만족한다. 이웃한 노드 사이에만 링크가 존재하기 때문에

$$\begin{cases} W_{(i,j)(i,j)} + W_{(i,j)(i+1,j)} = 1 & i = 1 \\ W_{(i,j)(i-1,j)} + W_{(i,j)(i,j)} + W_{(i,j)(i+1,j)} = 1 & 2 \leq i \leq I-1 \\ W_{(i,j)(i-1,j)} + W_{(i,j)(i,j)} = 1 & i = I \end{cases} \text{이다.}$$

- 5) 각 노드  $(i, j)$  에서  $p_{ij}$  를 다음과 같이 업데이트한다.  

$$p_{ij} \leftarrow I p_{ij}$$
- 6) 각 노드  $(i, j)$  에서  $q_j$  를 다음과 같이 계산한다.  

$$q_j \leftarrow X_{ij} p_{ij} \in \mathbb{R}^{N_i}$$
- 7) 행별로, 즉 각  $i \in \{1, 2, \dots, I\}$  에 대해  $J$  개의 노드  $(i, 1), \dots, (i, J)$  간에 평균 합의 알고리즘을  $M_r$  회 수행한다. 평균 합의 알고리즘 업데이트 식 (3)에 의해  $q_j \leftarrow \sum_{j'=1}^J V_{(i,j),(i,j')} q_{j'}$  를  $M_r$  회 반복하는 것과 같으며, 이웃한 노드 사이에만 링크가 존재하기 때문에 다음의 식으로 쓸 수 있다.

$$q_j \leftarrow \begin{cases} V_{(i,j)(i,j)} q_j + V_{(i,j)(i,j+1)} q_{j+1} & j = 1 \\ V_{(i,j)(i,j-1)} q_{j-1} + V_{(i,j)(i,j)} q_j + V_{(i,j)(i,j+1)} q_{j+1} & 2 \leq j \leq J-1 \\ V_{(i,j)(i,j-1)} q_{j-1} + V_{(i,j)(i,j)} q_j & j = J \end{cases}$$

$V$ 는 각 행의 링크에 대한 가중치 행렬로,  $V_{(i,j),(i,j')}$  는 노드  $(i, j)$  에서  $(i, j')$  에 대한 양의 가중치를 의미하며,  $\sum_{j'=1}^J V_{(i,j),(i,j')} = 1$  를 만족한다. 이웃한 노드 사이에만 링크가 존재하기 때문에

$$\begin{cases} V_{(i,j)(i,j)} + V_{(i,j)(i,j+1)} = 1 & j = 1 \\ V_{(i,j)(i,j-1)} + V_{(i,j)(i,j)} + V_{(i,j)(i,j+1)} = 1 & 2 \leq j \leq J-1 \\ V_{(i,j)(i,j-1)} + V_{(i,j)(i,j)} = 1 & j = J \end{cases} \text{이다.}$$

- 8) 각 노드  $(i, j)$  에서  $r_{ij}$  를 다음과 같이 계산한다.

$$r_{i,j} \leftarrow \| \mathbf{q}_j \|^2 \in \mathbb{R}$$

- 9) 열별로, 즉 각  $j \in \{1, 2, \dots, J\}$ 에 대해  $I$ 개의 노드  $(1, j), \dots, (I, j)$  간에 평균 합의 알고리즘을  $M_C$ 회 수행한다.

$$r_{i,j} \leftarrow \begin{cases} \mathbf{W}_{(i,j)(i,j)} \mathbf{r}_{ij} + \mathbf{W}_{(i,j)(i+1,j)} \mathbf{r}_{i+1,j} & i = 1 \\ \mathbf{W}_{(i,j)(i-1,j)} \mathbf{r}_{i-1,j} + \mathbf{W}_{(i,j)(i,j)} \mathbf{r}_{ij} \\ \quad + \mathbf{W}_{(i,j)(i+1,j)} \mathbf{r}_{i+1,j} & 2 \leq i \leq I-1 \\ \mathbf{W}_{(i,j)(i-1,j)} \mathbf{r}_{i-1,j} + \mathbf{W}_{(i,j)(i,j)} \mathbf{r}_{ij} & i = I \end{cases}$$

- 10) 각 노드  $(i, j)$ 에서  $\mathbf{r}_{ij}$ 와  $\mathbf{u}_{ij}$ 를 다음과 같이 업데이트한다.

$$\begin{aligned} \mathbf{r}_{ij} &\leftarrow I \mathbf{r}_{ij} \\ \mathbf{u}_{ij} &\leftarrow \mathbf{q}_{ij} / \sqrt{r_{ij}} \end{aligned}$$

- 11) 3) ~ 9)를  $K$ 번 반복한다.

위 알고리즘을 수행 후, 노드  $(i, j)$ 에서 반환하는  $\mathbf{u}_{ij}$ 는 최대 곱셈값에 대응되는 고유벡터  $\mathbf{u}$ 의  $i$ 번째 부분 벡터에 대응된다. 즉,  $\mathbf{u}_{ij}$ 는  $\mathbf{u}$ 의  $1 + \sum_{i'=1}^{i-1} N_{i'}$ 번째 성분에서  $\sum_{i'=1}^i N_{i'}$ 번째 성분까지의 근사값이다.

각 단계의 평균 합의 알고리즘은 행렬과 열벡터의 곱이 행렬의 각 행과 열벡터의 표준내적 (standard inner product)과 같음을 이용한 것이다 (그림 3).

각 노드  $(i, j)$ 에서 부분행렬과 부분벡터의 곱을 계산하고, 행방향으로 이들의 합을 계산하면 전체 행렬과 벡터의 곱을 얻을 수 있다. 그러나 마스터 노드가 없기 때문에 한 번에 모든 부분행렬 연산 값을 합할 수 없다. 따라서 평균 합의 알고리즘을 이용하여 각 노드에서 근사값을 계산하게 된다.

4)의 평균 합의 알고리즘을 통해 계산한 열방향 평균 합의  $(1/I) \sum_{i=1}^I (\mathbf{X}_{i,j}^T \mathbf{u}_{i,j})$ 의 근사값에 5)에서 스칼라  $I$ 를 곱하여,  $\mathbf{p}_{ij}$ 는  $\mathbf{X}^T \mathbf{u}$ 의  $j$ 번째 부분벡터,  $\sum_{i=1}^I (\mathbf{X}_{i,j}^T \mathbf{u}_{i,j})$ 의 근사값이 된다.

마찬가지로, 7)에서 평균 합의 알고리즘을 통해  $(J/L) \mathbf{q}_j$ 는  $(1/L) \mathbf{X} \mathbf{X}^T \mathbf{u}$ 의  $i$ 번째 부분벡터,

$$Ab = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} \begin{bmatrix} b_1 \\ b_2 \end{bmatrix} = \begin{bmatrix} A_{11}b_1 + A_{12}b_2 \\ A_{21}b_1 + A_{22}b_2 \end{bmatrix}$$

그림 3. 부분행렬 곱의 합에 의한 전체 행렬 곱 예시  
Fig. 3. An example of overall matrix multiplication as a sum of submatrix multiplications

$(1/L) \sum_{j=1}^J \mathbf{X}_{i,j} \sum_{i'=1}^I (\mathbf{X}_{i',j}^T \mathbf{u}_{i',j})$ 의 근사값이 되고, 10)의  $\mathbf{r}_{ij}$ 은  $\| \mathbf{q}_j \|^2$ ,  $\mathbf{u}_{ij}$ 는  $\mathbf{q}_j / \| (\mathbf{q}_j^T \dots \mathbf{q}_j^T)^T \|$ 의 근사값으로 거듭제곱법의 정규화 과정을 나타낸다.

### III. 제안 기법

네트워크의 구조는 II-4. 에서 제안한 2차원 그리드 구조 (그림 2)를 이용한다. 노드에 장치를 하나씩 배치하며, 하나의 장치는 최대 4개의 이웃한 장치와 무선채널로 연결된다. 이와 같은 그리드형의 네트워크에서는 부분행렬이 네트워크의 하나의 장치에 직관적으로 대응된다. 즉, 행렬의 곱 연산과 합 연산에도 용이하고, 분산 주성분 분석에 유리하다. 분산 주성분 분석 연산이 진행되는 시간 동안 장치의 상대적인 이동은 무시할 수 있으며, 따라서 네트워크 구조가 유지된다고 가정한다.

무선 채널은 부가 백색 가우시안 잡음 (addictive white gaussian noise, awgn)의 레일리 평탄 페이딩 (Rayleigh flat fading) 채널로 모델링한다. 레일리 페이딩은 작은 산란체가 많은 환경에서 직접경로 성분 (line-of-sight, LOS)이 없을 때 적합한 모델이나, 모델의 간단함으로 인해 산란체가 적은 환경에서도 사용된다<sup>10)</sup>. 또한 수신 노드에서는 채널을 완벽하게 추정할 수 있다고 가정한다. 이러한 가정 하에, 노드  $i$ 에서 이웃한 노드로부터 수신하는 신호  $\mathbf{y}_i$ 를 수식으로 정리하면 다음과 같다.

$$\mathbf{y}_i = \sum_j [h_{ij} (\mathbf{W}_{ij} \mathbf{s}_j) + \mathbf{n}_{ij}] \quad (4)$$

$\mathbf{W}_{ij}$ 는 노드  $i$ 에서 노드  $j$ 에 대한 가중치이며,  $h_{ij}$ 는 노드  $i$ 와 노드  $j$ 사이의 채널 이득,  $\mathbf{n}_{ij}$ 는 잡음 벡터이다. 부가 백색 가우시안 잡음을 갖는 레일리 페이딩 채널을 가정했으므로, 채널 이득과 잡음은 다음의 분포를 따른다.

$$h_{ij} \sim \text{CN}(0, 1), \mathbf{n}_{ij} \sim \text{CN}(0, N_0 \mathbf{I})$$

$h_{ij}$ 는 표준편차 1로 정규화 하였으며,  $\mathbf{I}$ 는 항등행렬,  $N_0$ 는 잡음의 분산 (variance)이다.

한편, 채널 이득과 잡음으로 인해 위 모델에 분산 주성분 분석 알고리즘을 그대로 적용할 경우 알고리즘의 반복횟수를 늘려도 평균 합의에 도달하지 못하고, 결과적으로 누적된 오차로 인해 주성분에 수렴하지 못한다. 때문에 무선채널 환경에서 적합한 방식으

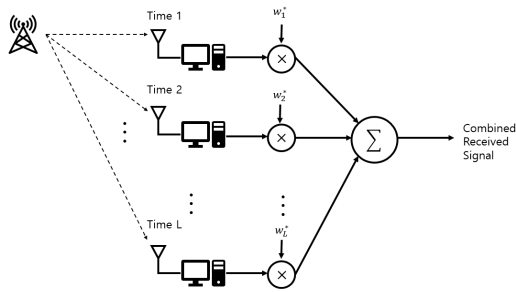


그림 4. 반복 송신에서 수신단의 최우도 결합기 구조  
Fig. 4. Maximal ratio combiner in time repeated transmissions

로 무선채널 효과를 상쇄해야 한다. 여기서는 시간 다이버시티 (time diversity) 기법 중 반복전송 (repetition)<sup>[10]</sup>을 이용해 같은 신호를 수신 노드를 향해 반복해서 전송하는 방식을 이용한다. 이 경우 같은 신호가 서로 다른 채널 경로를 통해 수신 노드에 도착하므로, 강한 상쇄 간섭으로 인해 채널 이득이 크게 감소하는 딥 페이드 (deep fade)에도 잘 대처할 수 있다는 장점이 있다.

수신 노드에서는 무선 채널의 효과를 최소화하기 위해 다양한 수신기법을 이용할 수 있는데, 그 중 최우도 결합기 (maximal ratio combiner)는 각 시간의 신호에 가중치를 곱해서 더하는 방식으로, 수신 신호 대 잡음비가 높은 시간의 신호에 더 큰 가중치를 곱하여 더한다 (그림 4).

총  $L$ 회 전송 중,  $l$ 번째 수신한 신호를  $y[l] = h[l]s + n[l]$ 이라 할 때, 최우도 결합기로 추정할 송신 신호는 다음의 수식으로 표현된다.

$$\hat{s} = \sum_{l=1}^L \frac{h[l]^* y[l]}{|h[l]|^2} \quad (5)$$

#### IV. 성능 분석

##### 4.1 데이터와 전처리

본 실험에 사용된 “Arrhythmia data set”은 452개의 표본과 279개의 특징으로 이루어진  $452 \times 279$  차원의 데이터 행렬이다. 분석에 앞서 먼저 빈칸을 포함하는 4개의 열과 서로 동일한 17개의 열을 제거한다. 한편, 각 특징(열)은 서로 다른 단위로 측정되었기 때문에 그대로 주성분 분석을 수행할 경우 상대적으로 큰 스케일의 성분이 우세하여 정확한 분석이 어렵다. 따라서 각 특징(열)별 정규화를 진행한다. 즉, 열별로 평균을 빼고 표준편차로 나누는 과정을 거치며, 처리

후 각 열은 평균 0, 표준편차 1의 벡터가 된다.

$$\tilde{X}_{ij} = \frac{X_{ij} - \frac{1}{N} \sum_{k=1}^N X_{kj}}{\sqrt{\frac{1}{N} \sum_{i=1}^N (X_{ij} - \frac{1}{N} \sum_{k=1}^N X_{kj})^2}} \quad (6)$$

여기서  $X_{ij}$ 는 데이터 행렬  $X$ 의  $i$ 번째 행,  $j$ 번째 열 성분이고,  $N$ 은 행 차원이다. 위 과정을 거쳐, 데이터 행렬은 452개의 표본 (행)과 258개의 정규화된 특징 (열)을 가지는  $452 \times 258$  행렬이 된다.

##### 4.2 시뮬레이션 환경

실험은 파이썬 (Python) 언어를 이용하여 진행하였으며, Numpy Library를 통해 행렬과 벡터 연산 등 선형대수 연산을 처리하였다.

데이터 행렬  $X$ 의 행 방향 분할 크기  $L$ 는 4, 열 방향 분할 크기  $J$ 는 6으로 행렬을 등분하여 각 노드에 할당했다. 즉, 행렬  $X$ 를 24등분하였으며, 각 부분행렬은  $113 \times 43$ 의 크기이다. 그림 5는 실험 모델의 모식도이다.

평균 합의 알고리즘의 반복횟수  $M_r = M_c = M_n = 80$ , 160인 경우를 고려하였고, 거듭제곱법의 반복횟수  $K$ 는 100으로 설정하였다. 열별 평균 합의 알고리즘(II-4. 4)와 9)에 해당의 가중치 행렬은

$$W_{(i,j)(i',j')} = \begin{cases} 0.7 & i=i' \in \{1,4\} \\ 0.4 & i=i' \in \{2,3\} \\ 0.3 & |i-i'|=1 \\ 0.0 & |i-i'| \geq 2 \end{cases}$$

이고, 행별 평균 합의 알

고리즘(II-4. 7)에 해당의 가중치 행렬은

$$V_{(i,j)(i,j')} = \begin{cases} 0.7 & j=j' \in \{1,6\} \\ 0.4 & j=j' \in \{2,3,4,5\} \\ 0.3 & |j-j'|=1 \\ 0.0 & |j-j'| \geq 2 \end{cases}$$

로 II-3.의 평균 합

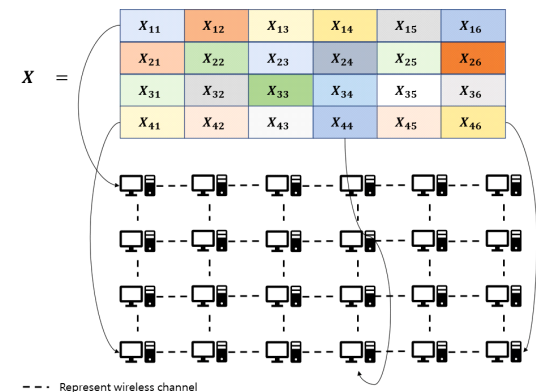


그림 5. 제안한 모델의 모식도  
Fig. 5. An illustration of the proposed model



의로의 수렴조건을 만족하는 임의의 가중치 행렬을 설정하였다.

채널 환경에 따른 알고리즘의 성능을 평가하기 위해 송신 신호 파워에 대한 잡음의 파워 비인 평균 신호 대 잡음비 (average SNR) 변화에 따른 실험을 진행하였으며, 신호 대 잡음비는 decibel (dB) 스케일로 -10에서 50까지 5 dB 간격으로 측정하였다. 또한 신호를 반복해서 전송하는 횟수는 1회에서 10회까지 1회 간격이다.

알고리즘의 성능은 각 노드  $(i, j)$ 에서 최대 곱셈에 대응되는 고유벡터  $\mathbf{u}$ 의  $i$ 번째 부분벡터  $\mathbf{u}^*$ 와 알고리즘을 통해 얻은 근사값  $\mathbf{u}_{ij}$  사이의 상대오차 (relative error)로 평가한다. 상대 오차를 계산한 수식은 다음과 같다.

$$E_{ij} = \|\mathbf{u}_{ij} - \mathbf{u}^*\|_2 / \|\mathbf{u}^*\|_2 \quad (7)$$

즉,  $E_{ij}$ 는 각 노드에서 알고리즘을 통해 업데이트하는 벡터  $\mathbf{u}_{ij}$ 와 목표 벡터 사이의 차이에 대한 비율이다. 본 실험에서는 알고리즘의 매 반복마다  $E_{ij}$ 를 측정하여 알고리즘 진행에 따른 상대오차의 변화를 분석하였다. 오차 측정의 기준이 되는 목표 벡터  $\mathbf{u}$ 는 Numpy의 "linalg.eig" 함수를 이용하여 구하였다.

### 4.3 시뮬레이션 결과

평균 신호 대 잡음비가 40 dB인 무선채널 환경에서 평균 합의 알고리즘 반복횟수  $M=80$ 으로 신호를 5회씩 전송한 결과는 그림 6과 같다.

세로축은 상대오차로 로그 스케일이며, 가로축은 거듭제곱법 반복횟수  $K$ 이다.  $K=40$ 까지 오차가 빠르

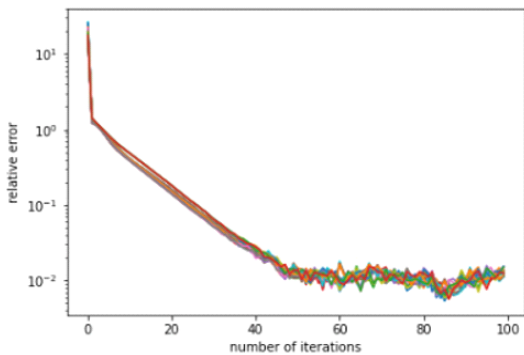


그림 6. 평균 신호 대 잡음비 40 dB 채널에서 평균 합의 알고리즘 반복횟수  $M=80$ 일 때, 거듭제곱법 반복횟수에 따른 상대오차  
Fig. 6. The relative error with respect to the number of power method iterations when SNR is 40dB and  $M=80$

게 감소하지만, 이후로는 감소폭이 작아지고, 상대오차는  $10^{-2}$ 수준까지 감소한다.

동일한 무선채널 환경에서 평균 합의 알고리즘 반복횟수  $M=160$ 으로 신호를 5회씩 전송한 실험결과는 그림 7과 같다.

마찬가지로  $K=40$ 까지 오차가 빠르게 감소하고, 이후로 감소폭이 줄어들며 상대오차  $10^{-2}$ 이하까지 내려간다.

신호 반복전송횟수  $L$ 을 높였을 때 상대오차의 변화를 확인하기 위해 평균 신호 대 잡음비가 40 dB인 환경에서 신호를 10회씩 전송했을 때 결과를 측정했고, 이에 따른 상대오차는 그림 8과 같다.

그림 8의 두 그래프의 평균 합의 알고리즘 반복횟수는 각각  $M=80$ ,  $M=160$ 이며, 두 경우 모두  $K=60$ 까지 오차가 지수적으로 빠르게 감소한다. 또한 그림 6, 7과 비교할 때 상대오차가 더 낮은 수준까지 내려갔다. 평균 합의 알고리즘의 반복횟수  $M$ 을 증가시킬수록 상대오차는 감소하는데,  $M$ 이 증가하면 거듭제곱법의 매 반복마다 평균 합으로 더 가까워지기 때문에, 최종적으로 반환하는 벡터와 최대 곱셈에 대응되는

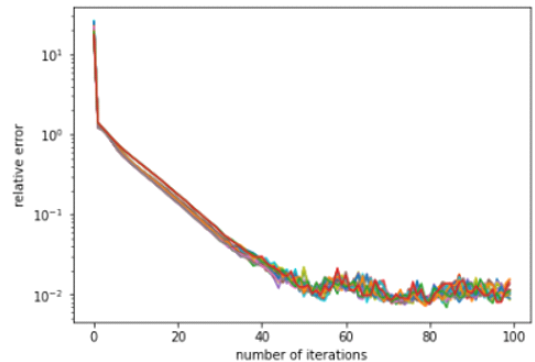


그림 7. 평균 신호 대 잡음비 40 dB 채널에서 평균 합의 알고리즘 반복횟수  $M=160$ 일 때, 거듭제곱법 반복횟수에 따른 상대오차  
Fig. 7. The relative error with respect to the number of power method iterations when SNR is 40dB and  $M=160$

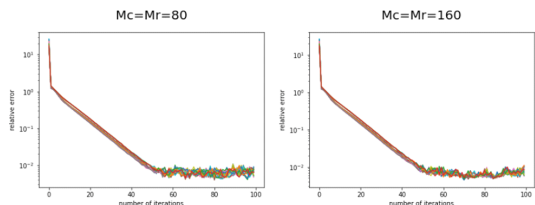


그림 8. 평균 신호 대 잡음비 40 dB 채널에서 신호 반복전송횟수  $L=10$ 일 때, 거듭제곱법 반복횟수에 따른 상대오차  
Fig. 8. The relative error with respect to the number of power method iterations when SNR is 40dB and  $L=10$

고유벡터 사이의 오차가 감소한다고 해석할 수 있다. 그러나 채널 환경이 좋지 않을 때는 잡음으로 인해 매 반복마다 평균 합의로 수렴하지 않고, 누적된 오차로 인해  $K$ 가 커짐에 따라 상대오차는 소폭 증감을 반복한다. 이 경우 오차 허용 한계를 미리 설정하고 오차가 그 이하로 떨어졌을 때 분산 주성분 분석 알고리즘을 중단하는 방법 등을 이용하여, 알고리즘의 수행시간을 절약하고 적절한 연산 성능을 얻을 수 있다.

신호 대 잡음비와 신호 반복 전송횟수가 성능에 미치는 영향을 보다 명확하게 분석하기 위해 신호 대 잡음비에 따른 상대오차와 신호 반복전송횟수에 따른 상대오차를 각각 측정했고, 결과는 그림 9와 10과 같다. 두 그래프 모두 분산 주성분 분석 알고리즘 수행이 끝났을 때, 즉,  $K=100$ 일 때 반환한 최종 상대오차의 값을 나타낸 것이다.

먼저 그림 9는 반복전송횟수  $L=5$ , 평균 합의 알고리즘 반복횟수  $M=80$ 일 때, 평균 신호 대 잡음비와 상대오차 간의 관계에 대한 그래프이다. 평균 신호 대 잡음비가 20 dB보다 작은 무선 채널 환경에서는 상대오차가  $10^{-1}$ 보다 크고 최대 고품값에 대응되는 고유벡터와 큰 차이를 보인다. 그러나 평균 신호 대 잡음비가 증가함에 따라 상대오차가 빠르게 감소한다.

그림 10은 평균 신호 대 잡음비 35 dB, 평균 합의 알고리즘 반복횟수  $M=80$ 일 때, 신호 반복전송횟수에 따른 상대오차를 나타냈다. 신호를 1회 전송한 경우, 즉 다이버시티 기법을 이용하지 않았을 때는 채널 페이딩과 잡음으로 인해 상대오차가 발산한다. 그러나 2회 반복전송만으로도 상대오차가  $10^{-1}$ 이하까지 크게 감소하고, 반복전송횟수가 증가함에 따라 상대오차가 감소하는 관계를 확인할 수 있다.

마지막으로 제안 기법의 연산 복잡도를 다음과 같

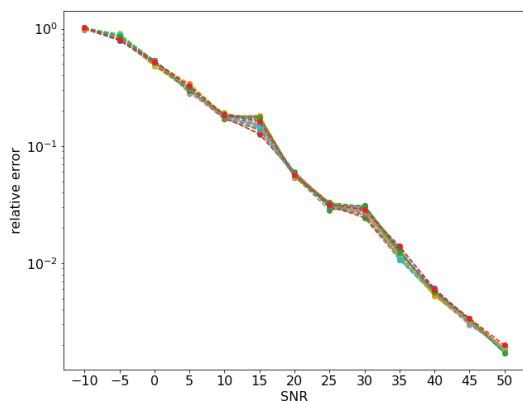


그림 9. 신호 대 잡음비에 따른 상대오차  
Fig. 9. The relative error with respect to SNR

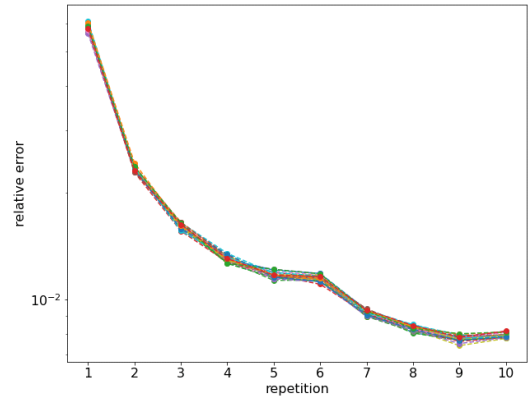


그림 10. 반복전송횟수에 따른 상대오차  
Fig. 10. The relative error with respect the number of repetition

이 분석하였다. 데이터 행렬을  $\mathbf{X} \in \mathbb{R}^{N \times L}$ , 전체 거듭제곱법 반복횟수를  $K$ , 평균 합의 알고리즘 반복횟수를  $M$ , 신호 반복전송횟수를  $D$ , 분할된 행렬의 행 차원과 열 차원을 각각  $N_i = N/I$ ,  $L_j = L/J$  라 할 때, 기존의 거듭제곱법과 무선 채널 환경에서 분산 주성분 분석 알고리즘의 연산 복잡도 분석은 다음과 같다. 기존의 거듭제곱법은  $\mathbf{X}$ 로부터 상관행렬  $\mathbf{R}$ 을 계산하고, 이를 바탕으로 (2)의 연산을  $K$ 번 반복 수행하는데  $O(N^2(K+L))$ 의 복잡도를 갖는다. 반면, 분산 주성분 분석 수행 시 각 노드는 ① 분할된 행렬과 벡터 곱 연산  $O(KN_iL_j)$  ② 최우도 결합 및 평균 합의 알고리즘  $O(KMD(N_i+L_j))$ 으로, 전체 연산 복잡도는  $O(K(MD(N_i+L_j)+N_iL_j))$ 이다.  $K, M, D$ 를 고차원 행렬의 차원  $N, L$ 보다 충분히 작다고 하면, 연산 복잡도는 행렬의 차원에 크게 의존한다. 분석을 위해 정사각

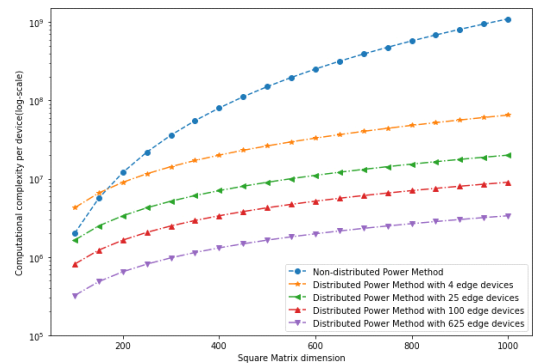


그림 11. 행렬 차원에 따른 연산 횟수.  $K=100, M=80, D=5$   
Fig. 11. The computational complexity with respect to the matrix dimension



행렬(Square matrix,  $N=L$ )에 대해 행과 열 차원을 균등하게 나누는 상황( $I=J$ )을 가정하여 연산 복잡도를 계산하면 그림 11과 같다.

분산 주성분 분석이 기존의 단일 장치에서의 거듭제곱법에 대해 연산 복잡도에서 우위를 보이며, 노드 수가 증가할수록 노드별 연산 횟수가 감소한다는 것을 알 수 있다.

### V. 결 론

본 논문에서는 무선 채널 환경에서 분산 주성분 분석이 수렴한다는 것을 확인하였다. 채널 이득과 잡음의 영향을 줄이기 위해서 시간 다이버시티 기법을 이용했고, 수신 노드에서도 최우도 결합기를 통해서 반복전송한 신호를 가중치 합하는 방식을 이용해야 한다. 또한 평균 신호 대 잡음비에 따른 상대오차, 신호 반복전송횟수에 따른 상대오차 분석을 통해 잡음의 파워가 성능에 크게 영향을 끼친다는 것을 확인하였다. 신호 대 잡음비가 높은 환경인 경우와 반복전송횟수를 높이는 경우 주성분과의 오차가 줄어들며 분산 주성분 분석 알고리즘 정확도가 향상된다. 단일 장치에서 거듭제곱법을 수행할 때에 비해 제안 기법은 장치당 연산 복잡도 측면에서도 이득이 있으며, 행렬의 차원이 증가함에 따라 연산 횟수의 차이는 더욱 커진다.

신호 대 잡음비는 환경에서 주어져 통제할 수 없다고 할 때, 주성분의 정확도 측면에서 성능을 높이기 위해서는 반복전송횟수를 높일 수 있다. 그러나 신호의 반복적인 전송 또한 시간을 소모하며, 결국 연산 속도를 높이기 위한 방법으로는 적절하지 않을 것이다. 따라서 다른 다이버시티 기법을 이용하여 잡음의 영향을 줄이면서, 연산 속도와 성능을 보장할 수 있는 방안에 대한 연구가 필요하다.

### References

[1] C. M. Bishop, *Pattern Recognition and Machine Learning*, Springer, 2006.  
 [2] G. H. Golub and C. F. Van Loan, *Matrix Computations*, 4th Ed., Johns Hopkins University Press, 2013.  
 [3] E. Oja and J. Karhunen, "On stochastic approximation of the eigenvectors and eigenvalues of the expectation of a random matrix," *J. Math. Anal. and Applicat.*, vol. 106, no. 1, pp. 69-84, 1985.

[4] K. Son, A. Ramamoorthy, and W. Choi, "Distributed matrix multiplication using group algebra for on-device edge computing," *IEEE Sig. Proc. Lett.*, vol. 28, pp. 2097-2101, Oct. 2021.  
 [5] W. Kim, K. Son, and W. Choi, "DoA Estimation Using Wireless Distributed PCA," *J. KICS*, vol. 47, no. 6, pp. 836-844, Jun. 2022.  
 [6] S. X. Wu, H.-T. Wai, L. Li, and A. Scaglione, "A review of distributed algorithms for principal component analysis," in *Proc. IEEE*, vol. 106, no. 8, pp. 1321-1340, Aug. 2018.  
 [7] T. Norikazu, M. Oura, and T. Migita. "Distributed algorithm for principal component analysis based on power method and average consensus algorithm," in *IEEE Int. Conf. PIC*, pp. 16-21, 2020.  
 [8] H. Kye and M. Kwon, "PCA-based low-complexity anomaly detection," *J. KICS*, vol. 46, no. 6, pp. 941-955, Jan. 2021.  
 [9] L. Xiao and S. Boyd, "Fast linear iterations for distributed averaging," *Systems & Control Lett.*, vol. 53, no. 1, pp. 65-78, 2004.  
 [10] D. Tse and P. Viswanath, *Fundamentals of Wireless Communication*, 1st Ed., Cambridge University Press, 2005.

허 흥 준 (Hongjun Heo)

2022년 2월 : 서울대학교 전기정보공학부 학사

2022년 3월~현재 : 서울대학교 전기정보공학부 석박통합과정



<관심분야> 무선통신, 정보이론  
 [ORCID:0000-0002-9176-9514]

손 경 락 (Kyungrak Son)



2014년 2월 : 한국과학기술원 (KAIST) 전기 및 전자공학과 학사 (수리과학과 복수전공)

2016년 2월 : 한국과학기술원 (KAIST) 전기 및 전자공학부 석사

2016년 3월~2021년 8월 : 한국과학기술원 (KAIST) 전기 및 전자공학부 박사

2021년 9월~현재 : 서울대학교 뉴미디어통신공동연구소 선임연구원

<관심분야> 무선통신, 정보이론, 부호이론

[ORCID:0000-0002-2983-985X]

최 완 (Wan Choi)



1996년 2월 : 서울대학교 전자공학 학사

1998년 2월 : 서울대학교 전자공학 석사

2006년 12월 : University of Texas at Austin, Department of Electrical and Computer Engineering 박사

2007년 1월~2020년 2월 : 한국과학기술원 (KAIST) 전기 및 전자공학부 조교수, 부교수, 정교수

2020년 3월~현재 : 서울대학교 뉴미디어통신공동연구소 및 전기정보공학부 정교수

<관심분야> 무선통신, 정보이론, 머신러닝

[ORCID:0000-0003-3930-7088]