MMSE Based MIMO Channel Estimator Via Primal-Dual Optimization Mehthod with Neural Network

Jing-Hao, Lee and Zhao-Jie, Luo

Advisor: Professor Carrson C. Fung

National Yang Ming Chiao Tung University

Jun. 12 2024

Reference

1. C. C. Fung, and D. Ivakhnenkov, ”Model-Driven Neural Network Based MIMO Channel Estimator”.
2. M. Eisen and A. Ribeiro, “Large scale wireless power allocation with graph neural networks,” *Proc. of the 2019 IEEE 20th Workshop on Signal Processing Advances in Wireless Communications (SPAWC)*, pp. 1-5, 2019.
3. M. Eisen, C. Zhang, L.F.O. Chamon, D.D. Lee and A. Ribeiro, “Learning optimal power allocations in wireless systems,” *IEEE Trans. on Signal Processing*, vol. 67(10), pp. 2775-2790, May 2019.
4. M. Eisen and A. Ribeiro, “Optimal wireless resource allocation with random edge graph neural networks,” *IEEE Trans. on Signal Processing*, vol. 68, pp. 2977-2991, 2020.
5. N. NaderiAlizadeh, M. Eisen and A. Ribeiro, “State-Augmented learnable algorithms for resource management in wireless networks,” *IEEE Trans. on Signal Processing*, vol. 70, pp. 5898-5912, Dec. 2022.

Problem Setup

The system model can be represented as:

$$\mathbf{Y} = \mathbf{H}\mathbf{X} + \mathbf{W}, \tag{1}$$

where $\mathbf{X} \in \mathbb{C}^{n_T \times T}$ represents the transmitted pilot signal, $\mathbf{W} \in \mathbb{C}^{n_R \times T}$ represents the additive white Gaussian noise with zero mean and unit variance, $\mathbf{Y} \in \mathbb{C}^{n_R \times T}$ represents the received signal, and $\mathbf{H} \in \mathbb{C}^{n_R \times n_T}$ is the channel matrix containing channel coefficients which represents $n_R \times n_T$ flat fading channel.
Vectorizing (1), it becomes

$$\begin{aligned} \mathbf{y} &= (\mathbf{X} \otimes \mathbf{I}_{n_R})\mathbf{h} + \mathbf{w} \\ &= \tilde{\mathbf{X}}\mathbf{h} + \mathbf{w}, \end{aligned} \tag{2}$$

where $\tilde{\mathbf{X}} \triangleq (\mathbf{X} \otimes \mathbf{I}_{n_R}) \in \mathbb{C}^{n_T n_R \times T n_R}$.

Problem Formulation

The MMSE channel estimator aims to minimize the Bayesian MSE of the channel estimate $\hat{\mathbf{h}}$

$$\hat{\mathbf{h}} = \arg\min_{\hat{\mathbf{h}}} \mathrm{BMSE}(\hat{\mathbf{h}}), \tag{3}$$

where

$$\mathrm{BMSE}(\hat{\mathbf{h}}) = \mathbb{E}_{\mathbf{y},\mathbf{h}}\left[\left\|\mathbf{h} - \hat{\mathbf{h}}\right\|_2^2\right]. \tag{4}$$

The closed-form solution of this MMSE problem is $\hat{\mathbf{h}} = \mathbb{E}_{\mathbf{h}|\mathbf{y}}[\mathbf{h}|\mathbf{y}]$. However, it requires knowledge about the conditional probability $p(\mathbf{h}|\mathbf{y})$, which may be unknown and/or difficult to obtain. Then (3) can be written as

$$\min_{\hat{\mathbf{h}}} \mathbb{E}_{\mathbf{y},\mathbf{h}}\left[\left\|\mathbf{h} - \hat{\mathbf{h}}\right\|_2^2\right]. \tag{5}$$

Primal-Dual Optimization Mehthod (1)

(5) can be reformulated to its epigraph form as

$$\begin{aligned} \min_{t,\hat{\mathbf{h}}} \quad & t \\ \text{s.t.} \quad & \mathbb{E}_{\mathbf{y},\mathbf{h}}\left[\left\|\mathbf{h}-\hat{\mathbf{h}}\right\|_2^2\right] \leq t. \end{aligned} \tag{6}$$

Then its Lagrangian function can be written as

$$\mathcal{L}\left(\hat{\mathbf{h}}, t, \lambda\right) = t + \lambda\left(\mathbb{E}_{\mathbf{y},\mathbf{h}}\left[\left\|\mathbf{h}-\hat{\mathbf{h}}\right\|_2^2\right] - t\right) \tag{7}$$

Primal-Dual Optimization Mehthod (2)

We use parameterize channel estimator so that $\hat{\mathbf{h}} = \phi(\mathbf{y}; \boldsymbol{\theta})$, with $\boldsymbol{\theta}$ denoting the parameters of the neural network.

Then the Lagrangian function of (15) can be written as

$$\begin{aligned}\mathcal{L}\left(\hat{\mathbf{h}}, t, \lambda\right) &= t + \lambda\left(\mathbb{E}_{\mathbf{y},\mathbf{h}}\left[\left\|\mathbf{h} - \hat{\mathbf{h}}\right\|_2^2\right] - t\right) \\ &= t + \lambda\left(\mathbb{E}_{\mathbf{y},\mathbf{h}}\left[\|\mathbf{h} - \phi(\mathbf{y}; \boldsymbol{\theta})\|_2^2\right] - t\right)\end{aligned}$$

Primal-Dual Optimization Mehthod (3)

It is uncertain whether or not the duality gap equals zero.
However, the stationary point of $\mathcal{L}\left(\hat{\mathbf{h}}, t, \lambda\right)$ can be found via the KKT conditions by solving for the primal and dual variables alternately using gradient descent and ascent, respectively:

$$
\begin{aligned}
\boldsymbol{\theta}_{k+1} &= \boldsymbol{\theta}_k - \alpha_{\boldsymbol{\theta},k}\lambda_k \nabla_{\boldsymbol{\theta}_k}\mathbb{E}\left[\|\mathbf{h} - \phi(\mathbf{y};\boldsymbol{\theta}_k)\|_2^2\right] \\
t_{k+1} &= t_k - \alpha_{t,k}(1 - \lambda_k) \\
\lambda_{k+1} &= \left[\lambda_k + \alpha_{\lambda,k}\left(\mathbb{E}\left[\|\mathbf{h} - \phi(\mathbf{y};\boldsymbol{\theta}_{k+1})\|_2^2\right] - t_{k+1}\right)\right]_+
\end{aligned}
$$

Policy Gradient (1)

The policy gradient theorem is:

$$\nabla_{\boldsymbol{\theta}} \mathbb{E}_{\tau}[G(\tau)] = \mathbb{E}_{\tau}\left[\sum_{t=0}^{T-1} G(\tau) \nabla_{\boldsymbol{\theta}} \log \pi_{\boldsymbol{\theta}}(A_t|S_t)\right],$$

where $G(\tau)$ is the overall return of the trajectory, and $\pi_{\boldsymbol{\theta}(A_t|S_t)}$ is the distribution of action that the parameterized policy makes under the observation.
At each time step, $t = 1, ..., T-1$:

$$\nabla_{\boldsymbol{\theta}} \mathbb{E}_{\tau}[G(\tau)] = \mathbb{E}_{\tau}\left[G(\tau) \nabla_{\boldsymbol{\theta}} \log \pi_{\boldsymbol{\theta}}(A_t|S_t)\right]$$

And we can estimate the policy gradient with sample mean:

$$\widehat{\nabla_{\boldsymbol{\theta}}} \mathbb{E}_{\tau}[G(\tau)] = \frac{1}{|\mathcal{D}|} \sum_{\mathcal{D}} G(\tau) \nabla_{\boldsymbol{\theta}} \log \pi_{\boldsymbol{\theta}}(A_t|S_t)$$

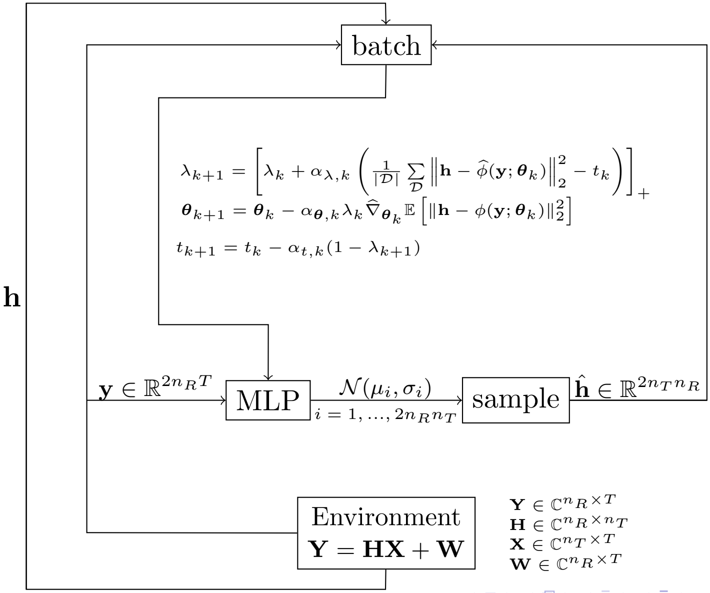
Policy Gradient (2)

Our goal is to minimize the mean square error, by substituting $\mathbb{E}_\tau[G(\tau)]$, $\pi_{\boldsymbol{\theta}}(A_t|S_t)$ with $\mathbb{E}_{\mathbf{y},\mathbf{h}}\left[\|\mathbf{h}-\phi(\mathbf{y};\boldsymbol{\theta})\|_2^2\right]$, and $\pi_{\boldsymbol{\theta}}(\hat{\mathbf{h}}|\mathbf{y})$. Thus, we obtain the estimated policy gradient for our problem:

$$\widehat{\nabla_{\boldsymbol{\theta}}}\mathbb{E}_{\mathbf{y},\mathbf{h}}\left[\|\mathbf{h}-\phi(\mathbf{y};\boldsymbol{\theta})\|_2^2\right] = \frac{1}{|\mathcal{D}|}\sum_{\mathcal{D}}\left\|\mathbf{h}-\widehat{\phi}(\mathbf{y};\boldsymbol{\theta})\right\|_2^2 \nabla_{\boldsymbol{\theta}} \log \pi_{\boldsymbol{\theta}}\left(\hat{\mathbf{h}}|\mathbf{y}\right) \quad (8)$$

where $\hat{\phi}(\mathbf{y};\boldsymbol{\theta}) = \hat{\mathbf{h}}$ is the sampled output of the policy.

Experiment Diagram



$\mathbf{Y} \in \mathbb{C}^{n_R \times T}$
$\mathbf{H} \in \mathbb{C}^{n_R \times n_T}$
$\mathbf{X} \in \mathbb{C}^{n_T \times T}$
$\mathbf{W} \in \mathbb{C}^{n_R \times T}$

GNN

- [1]How to optimally allocating resources across a set of transmitters and receivers in a a wireless network?
- This paper proposes a method using REGNN.
- The performance of using REGNN is better than using MLP
  1. number of parameter
  2. time complexity
  3. scalability
  4. transference
- Therefore, we introduce the GNN method to replace MLP for estimating channel H.

[1]M. Eisen and A. Ribeiro, “Optimal wireless resource allocation with random edge graph neural networks,” IEEE Trans. on Signal Processing, vol. 68, pp. 2977-2991, 2020.

GNN

- F_ℓ : the number of the filters at ℓ layer.
- L : total layer of GNN.
- K_ℓ : the order of the filters at ℓ layer.

$$F_\ell^{I/O}(\mathbf{W}) = \sum_{k=0}^{K_\ell - 1} a_{\ell/k}^{I/O}\mathbf{W}^k \tag{9}$$

$$= a_{\ell/0}^{I/O}\mathbf{I} + a_{\ell/1}^{I/O}\mathbf{W} + a_{\ell/2}^{I/O}\mathbf{W}^2 + \ldots + a_{\ell/K_{\ell-1}}^{I/O}\mathbf{W}^{K_\ell - 1} \tag{10}$$

$$\text{where } \ell : \text{layer, } I : \text{input index, } O : \text{output index} \tag{11}$$

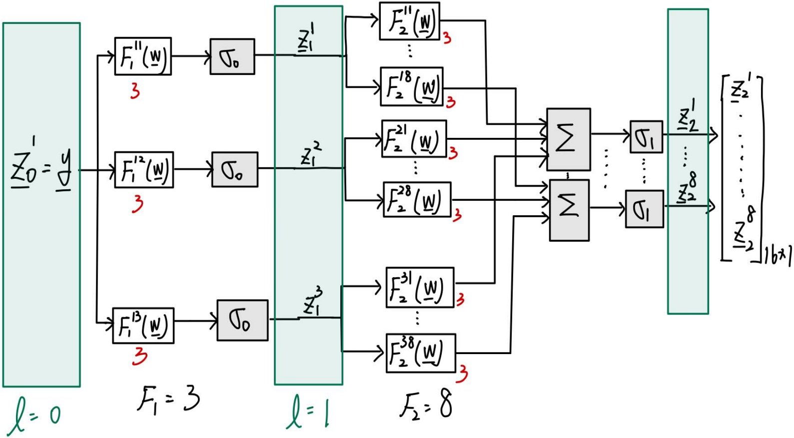
- Example :

$$F_0^{1/3}(\mathbf{W}) = \sum_{k=0}^{2} a_{0/k}^{1/3}\mathbf{W}^k \tag{12}$$

$$= a_{0/0}^{1/3}\mathbf{I} + a_{0/1}^{1/3}\mathbf{W} + a_{0/2}^{1/3}\mathbf{W}^2 \tag{13}$$

GNN

- $K_\ell = 3, L = 2, F_1 = 3, F_2 = 8, n_T = 2, n_R = 2$



- Total parameter : 3x3+3x3x8 = 81

BUT!

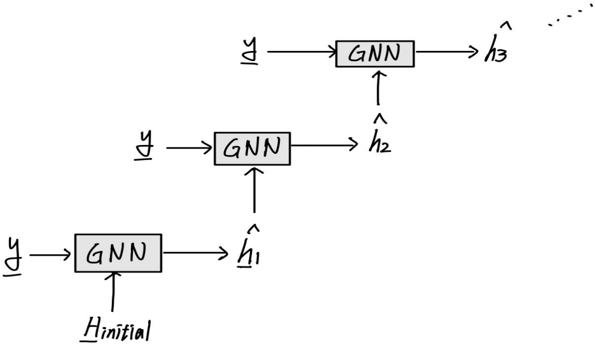
- It is not possible to use the adjacency matrix (Channel matrix, $\mathbf{H}$) as the Graph Shift Operator because the Channel matrix is the output of the REGNN, and we do not know $\mathbf{H}$ initially.
- Because in Eisen's paper, the Channel matrix $\mathbf{H}$ is known, the adjacency matrix is also known, which means the Graph Shift Operator can be used for convolution.
- Therefore, it might be impossible to use GNN methods to solve our problem.

Guess

- Since we don't have $\mathbf{H}$, we use Least-Square estimator to find $\mathbf{H}_{initial}$. With this $\mathbf{H}_{initial}$,we can determine the adjacency matrix, which gives us the GSO. Then, we can use the GNN output to obtain $\hat{\mathbf{H}}$. If the performance of this $\hat{\mathbf{H}}$ is not satisfactory, we can use $\hat{\mathbf{H}}$ to determine a new adjacency matrix as the GSO and use the GNN again to obtain another $\hat{\mathbf{H}}$. We can repeat this process until the performance approaches that of the MMSE estimator.

Guess

- As illustrated in the diagram below.



- I guess that using this method, the estimated $\hat{\mathbf{H}}$ will get increasingly better.

Simulation Configuration (1)

parameters	values
$[n_R, n_T, T]$	[4,8,8]
μ_H, σ_H	0, 1
Hidden layer size	1024 or 4×1024
Length of each trajectory	1
Batch size $= \lvert\mathcal{D}\rvert$	10,000
Number of batches	100
Trainning dataset	1,000,000
Validation dataset	2,000
Epoch	200 $\sim$ 1,000
Learning rate	1e-5 $\sim$ 1e-6

Table: simulation configuration

Simulation Configuration (2)

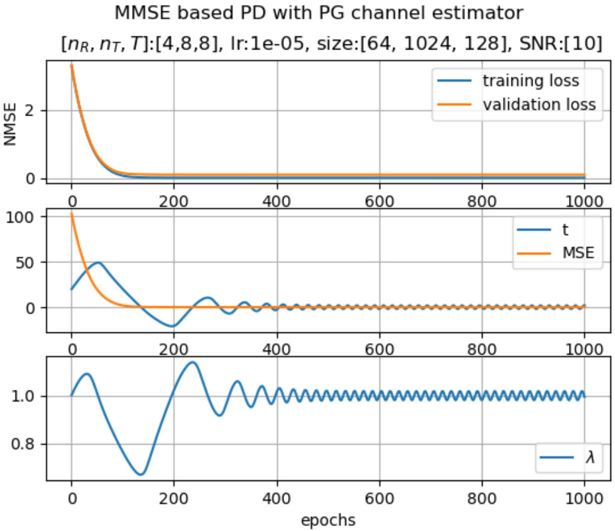
- Let the loss be Normalized Mean Squared Error (NMSE):

$$NMSE = \frac{1}{N} \sum_{n=1}^{N} \frac{\|\mathbf{h}_n - \phi(\mathbf{y}_n; \boldsymbol{\theta}_k)\|_2^2}{\|\mathbf{h}_n\|_2^2} \tag{14}$$

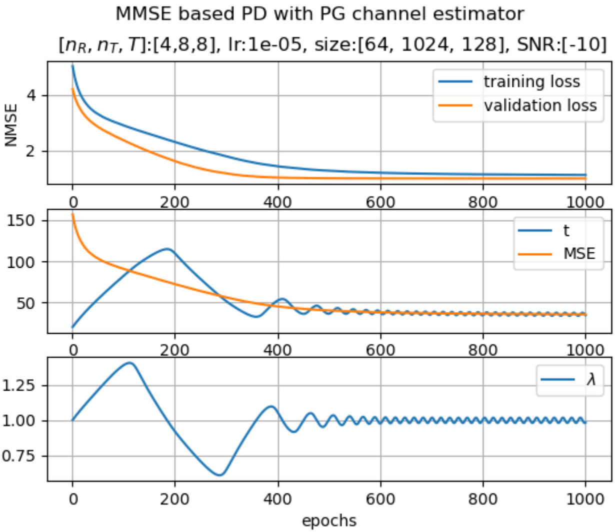
  Here, N represents the size of the training or validation dataset, implying that the Normalized Mean Squared Error (NMSE) is calculated for each epoch.

- Using Adam optimizer to update the primal and dual variables.

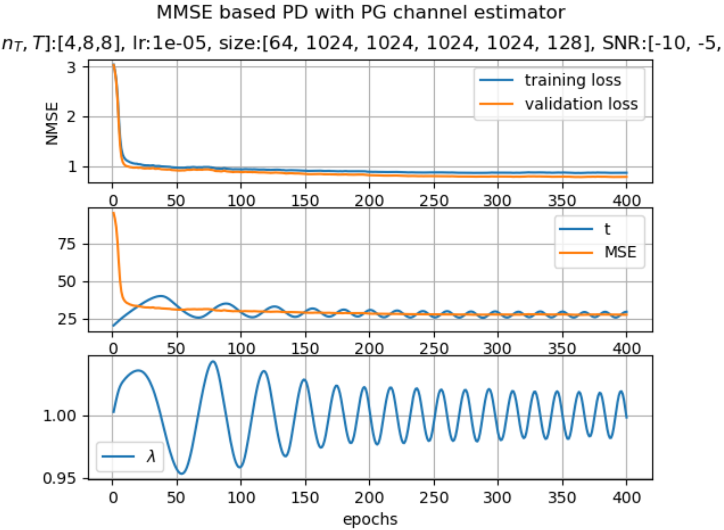
Simulation Result (1)



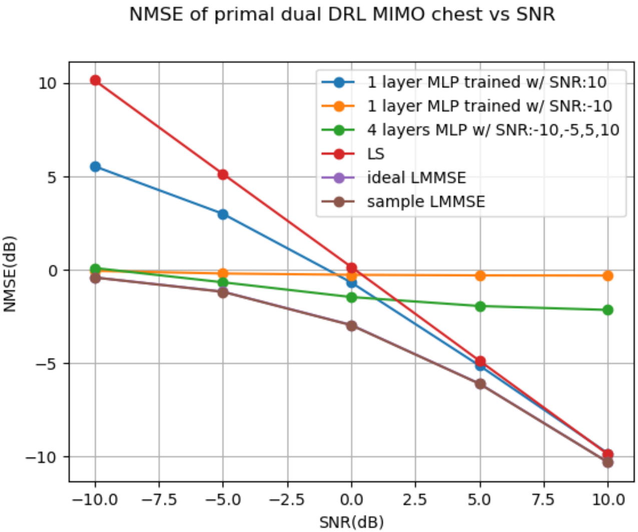
Simulation Result (2)



Simulation Result (3)



Simulation Result (4)



Conclusion

- Our simulation results demonstrate that the proposed neural network-based MMSE estimator can outperform the least squares (LS) and approach the performance of linear MMSE (LMMSE) estimators under specific signal-to-noise ratio (SNR) conditions, especially under challenging conditions with high noise levels.
- However, it may requires a larger model size or more sophisticated neural network architectures to achieve robustness in its performance.