Bi-Directional Training in Interference Network

Shao-Han Chen

March 17, 2016

1 System Model 1: Private and Common Messages

Under this model, we want to see how cooperation on transmitter(base station) side will improve the channel capacity. One way we implement the cooperation scheme is to make all transmitters for common messages, instead of working separately, a big cooperative array. That is to say, for the backward training part, we **minimize the mean square error(MSE) of total received signals together**. However, we found this method is actually **worse** than the one without cooperation. **How Doesn't this Work?** This is related the way we **measure performance**. Our objective is to maximize the **sum capacity** of individual users, which is equal to minimize $\sum_k MSE_k$. However, our cooperation scheme actually leads to the solution of maximizing total capacity, that is minimize MSE. Please note that we are not claiming cooperations imply worse performance. It's just this particular cooperation scheme doesn't work.

1.1 System Model

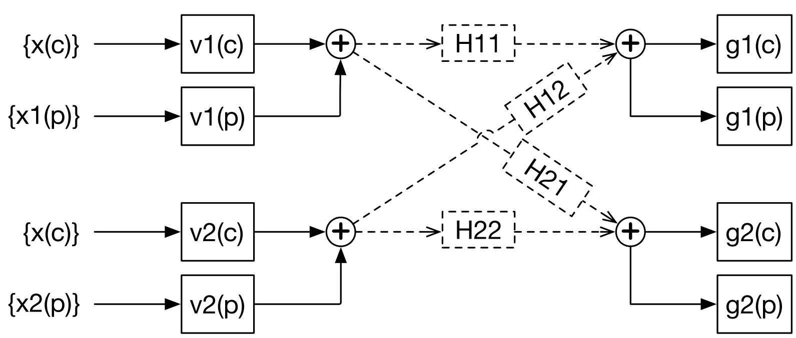

Figure 1: Forward Channel

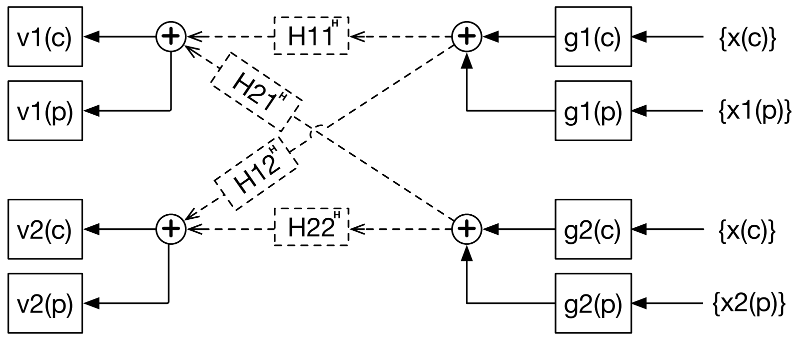

Figure 2: Backward Channel

1.2 Optimization Problem

$$\min_{\mathbf{v}_k,\mathbf{g}_k} \sum_k MSE_k^{(c)} + MSE_k^{(p)}$$

$$\text{subject to } \|\mathbf{v}_k^{(c)}\|^2 + \|\mathbf{v}_k^{(p)}\|^2 = P; \|\mathbf{g}_k\|^2 = P$$

where

$$MSE_k^{(c)} = \mathrm{E}[(x - \mathbf{g}_k^{H(c)}\mathbf{y}_k)(x - \mathbf{g}_k^{H(c)}\mathbf{y}_k)^H]$$

$$MSE_k^{(p)} = \mathrm{E}[(x_k^{(p)} - \mathbf{g}_k^{H(p)}\mathbf{y}_k)(x_k^{(p)} - \mathbf{g}_k^{H(p)}\mathbf{y}_k)^H]$$

1.3 The received signal vector at k-th receiver

$$\mathbf{y}_k = \mathbf{H}_{kk}(\mathbf{v}_k^{(c)}x + \mathbf{v}_k^{(p)}x_k^{(p)}) + \sum_{j\neq k}\mathbf{H}_{kj}(\mathbf{v}_j^{(c)}x + \mathbf{v}_j^{(p)}x_j^{(p)}) + \mathbf{n}_k$$

1.4 SINR Derivation

$$s_k^{(c)} = \mathbf{g}_k^{H(c)}(\sum_i \mathbf{H}_{ki}\mathbf{v}_i^{(c)}x)$$

$$s_k^{(p)} = \mathbf{g}_k^{H(p)}(\mathbf{H}_{kk}\mathbf{v}_k^{(p)}x_k^{(p)})$$

$$n_k^{(c)} = \mathbf{g}_k^{H(c)}(\sum_i \mathbf{H}_{ki}\mathbf{v}_i^{(p)}x_i^{(p)} + \mathbf{n}_k)$$

$$n_k^{(p)} = \mathbf{g}_k^{H(p)}(\sum_i \mathbf{H}_{ki}\mathbf{v}_i^{(c)}x + \sum_{j\neq k} \mathbf{H}_{kj}\mathbf{v}_j^{(p)}x_j^{(p)} + \mathbf{n}_k)$$

$$\frac{|s_k^{(c)}|^2}{|n_k^{(c)}|^2} = \frac{|\mathbf{g}_k^{H(c)}\sum_i \mathbf{H}_{ki}\mathbf{v}_i^{(c)}|^2}{\sum_i |\mathbf{g}_k^{H(c)}\mathbf{H}_{ki}\mathbf{v}_i^{(p)}|^2 + |\mathbf{g}_k^{H(c)}\mathbf{R}_k\mathbf{g}_k^{(c)}|}$$

$$\frac{|s_k^{(p)}|^2}{|n_k^{(p)}|^2} = \frac{|\mathbf{g}_k^{H(p)}\mathbf{H}_{kk}\mathbf{v}_k^{(p)}|^2}{|\mathbf{g}_k^{H(p)}\sum_i \mathbf{H}_{ki}\mathbf{v}_i^{(c)}|^2 + \sum_{j\neq k}|\mathbf{g}_k^{H(p)}\mathbf{H}_{kj}\mathbf{v}_j^{(p)}|^2 + |\mathbf{g}_k^{H(p)}\mathbf{R}_k\mathbf{g}_k^{(p)}|}$$

1.5 Max-SINR Algorithm[Gomadam, 2011]

For Max-SINR algorithm, we assume **all channel state information(CSI) is available** to each user. In this method, the solution for each transceivers are simply Wiener filters. In other words, solving it by Wiener-Hopf equation: $R^{-1}p$ [Adaptive Filter Theory, Simon Haykin]

1.5.1 Forward Training (fix $\mathbf{v}_k^{(c)}$, $\mathbf{v}_k^{(p)}$, $\forall k$)

$$\begin{aligned}\mathbf{g}_k^{*(c)} = \Big[&\mathbf{H}_{kk}\mathbf{v}_k^{(c)}\mathbf{v}_k^{H(c)}\mathbf{H}_{kk}^H + \mathbf{H}_{kk}\mathbf{v}_k^{(p)}\mathbf{v}_k^{H(p)}\mathbf{H}_{kk}^H \\ &+ (\sum_{j\neq k}\mathbf{H}_{kj}\mathbf{v}_j^{(c)})(\sum_{j\neq k}\mathbf{v}_j^{H(c)}\mathbf{H}_{kj}^H) + (\sum_{j\neq k}\mathbf{H}_{kj}\mathbf{v}_j^{(p)}\mathbf{v}_j^{H(p)}\mathbf{H}_{kj}^H) \\ &+ \mathbf{H}_{kk}\mathbf{v}_k^{(c)}(\sum_{j\neq k}\mathbf{v}_j^{H(c)}\mathbf{H}_{kj}^H) + (\sum_{j\neq k}\mathbf{H}_{kj}\mathbf{v}_j^{(c)})\mathbf{v}_k^{H(c)}\mathbf{H}_{kk}^H + \sigma^2\mathbf{I}\Big]^{-1}(\sum_i \mathbf{H}_{ki}\mathbf{v}_i^{(c)})\end{aligned}$$

$$\begin{aligned}\mathbf{g}_k^{*(p)} = \Big[&\mathbf{H}_{kk}\mathbf{v}_k^{(c)}\mathbf{v}_k^{H(c)}\mathbf{H}_{kk}^H + \mathbf{H}_{kk}\mathbf{v}_k^{(p)}\mathbf{v}_k^{H(p)}\mathbf{H}_{kk}^H \\ &+ (\sum_{j\neq k}\mathbf{H}_{kj}\mathbf{v}_j^{(c)})(\sum_{j\neq k}\mathbf{v}_j^{H(c)}\mathbf{H}_{kj}^H) + (\sum_{j\neq k}\mathbf{H}_{kj}\mathbf{v}_j^{(p)}\mathbf{v}_j^{H(p)}\mathbf{H}_{kj}^H) \\ &+ \mathbf{H}_{kk}\mathbf{v}_k^{(c)}(\sum_{j\neq k}\mathbf{v}_j^{H(c)}\mathbf{H}_{kj}^H) + (\sum_{j\neq k}\mathbf{H}_{kj}\mathbf{v}_j^{(c)})\mathbf{v}_k^{H(c)}\mathbf{H}_{kk}^H + \sigma^2\mathbf{I}\Big]^{-1}(\mathbf{H}_{kk}\mathbf{v}_k^{(p)})\end{aligned}$$

1.5.2 Backward Training (fix $\mathbf{g}_k^{(c)}$, $\mathbf{g}_k^{(p)}$, $\forall k$)

$$\mathbf{Z}_{ab} = \mathbf{H}_{ba}^H$$

without cooperation

$$\begin{aligned}\mathbf{v}_k^{*(c)} =& \Big[\mathbf{Z}_{kk}\mathbf{g}_k^{(c)}\mathbf{g}_k^{H(c)}\mathbf{Z}_{kk}^H + \mathbf{Z}_{kk}\mathbf{g}_k^{(p)}\mathbf{g}_k^{H(p)}\mathbf{Z}_{kk}^H \\ &+ (\sum_{j\neq k}\mathbf{Z}_{kj}\mathbf{g}_j^{(c)})(\sum_{j\neq k}\mathbf{g}_j^{H(c)}\mathbf{Z}_{kj}^H) + (\sum_{j\neq k}\mathbf{Z}_{kj}\mathbf{g}_j^{(p)}\mathbf{g}_j^{H(p)}\mathbf{Z}_{kj}^H) \\ &+ \mathbf{Z}_{kk}\mathbf{g}_k^{(c)}(\sum_{j\neq k}\mathbf{g}_j^{H(c)}\mathbf{Z}_{kj}^H) + (\sum_{j\neq k}\mathbf{Z}_{kj}\mathbf{g}_j^{(c)})\mathbf{g}_k^{H(c)}\mathbf{Z}_{kk}^H + \sigma^2\mathbf{I}\Big]^{-1}(\sum_i \mathbf{Z}_{ki}\mathbf{g}_i^{(c)})\end{aligned}$$

with cooperation

$$\mathbf{V}^{*(c)} = \left\{[\mathbf{Z}]\mathbf{g}^{(c)}\mathbf{g}^{H(c)}[\mathbf{Z}]^H + [\mathbf{Z}]\begin{bmatrix}\mathbf{g}_1^{(p)}\mathbf{g}_1^{H(p)} & & \mathbf{O} \\ & \ddots & \\ \mathbf{O} & & \mathbf{g}_k^{(p)}\mathbf{g}_k^{H(p)}\end{bmatrix}[\mathbf{Z}]^H + \sigma^2\mathbf{I}\right\}^{-1}([\mathbf{Z}]\mathbf{g}^{(c)})$$

$$\begin{aligned}\mathbf{v}_k^{*(p)} =& \Big[\mathbf{Z}_{kk}\mathbf{g}_k^{(c)}\mathbf{g}_k^{H(c)}\mathbf{Z}_{kk}^H + \mathbf{Z}_{kk}\mathbf{g}_k^{(p)}\mathbf{g}_k^{H(p)}\mathbf{Z}_{kk}^H \\ &+ (\sum_{j\neq k}\mathbf{Z}_{kj}\mathbf{g}_j^{(c)})(\sum_{j\neq k}\mathbf{g}_j^{H(c)}\mathbf{Z}_{kj}^H) + (\sum_{j\neq k}\mathbf{Z}_{kj}\mathbf{g}_j^{(p)}\mathbf{g}_j^{H(p)}\mathbf{Z}_{kj}^H) \\ &+ \mathbf{Z}_{kk}\mathbf{g}_k^{(c)}(\sum_{j\neq k}\mathbf{g}_j^{H(c)}\mathbf{Z}_{kj}^H) + (\sum_{j\neq k}\mathbf{Z}_{kj}\mathbf{g}_j^{(c)})\mathbf{g}_k^{H(c)}\mathbf{Z}_{kk}^H + \sigma^2\mathbf{I}\Big]^{-1}(\mathbf{Z}_{kk}\mathbf{g}_k^{(p)})\end{aligned}$$

1.6 Bi-Directional Training with LS Algorithm[Shi, 2014]

For Bi-Directional Training, we assume **all channel state information(CSI) is not available** to each user. In this case, we could either adopt Least Mean Square Method(LMS) or Least Square Method(LS) to carry out the solution[Adaptive Filter Theory, Simon Haykin]. **There is actually another interesting finding here**. We all have seen the monotonically decreasing diagram of MSE when designing/studying filters with both methods. We can also show that when the filter converges(achieve minimum MSE), it also maximizes Signal to Interference plus Noise Ratio(SINR). **However, how does the SINR behave before it converges?** For LS, it is monotonically decreasing just as MSE diagram, but for LMS, it is **not!** Therefore, LMS isn't applicable for our application. We need SINR increases after each iteration.

1.6.1 Forward Training (fix $\mathbf{v}_k^{(c)}$, $\mathbf{v}_k^{(p)}$, $\forall k$)

$$\mathbf{g}_k^{(c)}(n+1) = \mathbf{g}_k^{(c)}(n) + \mu\mathbf{y}_k(n)[x(n) - \mathbf{g}_k^{H(c)}(n)\mathbf{y}_k(n)]^*$$

$$\mathbf{g}_k^{(p)}(n+1) = \mathbf{g}_k^{(p)}(n) + \mu\mathbf{y}_k(n)[x_k^{(p)}(n) - \mathbf{g}_k^{H(p)}(n)\mathbf{y}_k(n)]^*$$

1.6.2 Backward Training (fix $\mathbf{g}_k^{(c)}$, $\mathbf{g}_k^{(p)}$, $\forall k$)

without cooperation

$$\mathbf{v}_k^{(c)}(n+1) = \mathbf{v}_k^{(c)}(n) + \mu\mathbf{y}_k(n)[x(n) - \mathbf{v}_k^{H(c)}(n)\mathbf{y}_k(n)]^*$$

$$\mathbf{v}_k^{(p)}(n+1) = \mathbf{v}_k^{(p)}(n) + \mu\mathbf{y}_k(n)[x_k^{(p)}(n) - \mathbf{v}_k^{H(p)}(n)\mathbf{y}_k(n)]^*$$

with cooperation

$$\mathbf{V}^{(c)}(n+1) = \mathbf{V}^{(c)}(n) + \mu\mathbf{Y}(n)[x(n) - \mathbf{V}^{H(c)}(n)\mathbf{Y}(n)]^*$$

$$\mathbf{v}_k^{(p)}(n+1) = \mathbf{v}_k^{(p)}(n) + \mu\mathbf{y}_k(n)[x_k^{(p)}(n) - \mathbf{v}_k^{H(p)}(n)\mathbf{y}_k(n)]^*$$

1.7 Special Case(2 Users, MIMO Channel, Only Common Messages)

$$\mathbf{y}_k = \sum_{i=1}^{2} \mathbf{H}_{ki}\mathbf{v}_i^{(c)}x + \mathbf{n}_k$$

$$\begin{aligned}
MSE_k^{(c)} &= \mathrm{E}[(x - \mathbf{g}_k^{H(c)}\mathbf{y}_k)(x - \mathbf{g}_k^{H(c)}\mathbf{y}_k)^H] \\
&= \mathrm{E}[x^2] - \mathrm{E}[x\mathbf{y}_k^H\mathbf{g}_k^{(c)}] - \mathrm{E}[x\mathbf{g}_k^{H(c)}\mathbf{y}_k] + \mathrm{E}[\mathbf{g}_k^{H(c)}\mathbf{y}_k\mathbf{y}_k^H\mathbf{g}_k^{(c)}] \\
&= 1 - \sum_{i=1}^{2}\mathbf{v}_i^{H(c)}\mathbf{H}_{ki}^H\mathbf{g}_k^{(c)} - \mathbf{g}_k^{H(c)}\sum_{i=1}^{2}\mathbf{H}_{ki}\mathbf{v}_i^{(c)} + \mathbf{g}_k^{H(c)}(\sum_{i=1}^{2}\mathbf{H}_{ki}\mathbf{v}_i^{(c)})(\sum_{i=1}^{2}\mathbf{v}_i^{H(c)}\mathbf{H}_{ki}^H)\mathbf{g}_k^{(c)} + \sigma^2\mathbf{g}_k^{H(c)}\mathbf{g}_k^{(c)}
\end{aligned}$$

$$\begin{aligned}
\mathbf{v}_1^{*(c)} &= \underset{\mathbf{v}_1^{(c)}}{\mathrm{argmin}}(\sum_{k=1}^{2} MSE_k^{(c)}) \\
&= \left[2\mathbf{H}_{11}^H\mathbf{g}_1^{(c)}\mathbf{g}_1^{H(c)}\mathbf{H}_{11} + 2\mathbf{H}_{21}^H\mathbf{g}_2^{(c)}\mathbf{g}_2^{H(c)}\mathbf{H}_{21})\right]^{-1}(2\mathbf{H}_{11}^H\mathbf{g}_1^{(c)} + 2\mathbf{H}_{21}^H\mathbf{g}_2^{(c)} \\
&\quad - \mathbf{g}_1^{H(c)}\mathbf{H}_{12}\mathbf{v}_2^{(c)}\mathbf{H}_{11}^H\mathbf{g}_1^{(c)} - \mathbf{H}_{11}^H\mathbf{g}_1^{(c)}\mathbf{v}_2^{H(c)}\mathbf{H}_{12}^H\mathbf{g}_1^{(c)} - \mathbf{g}_2^{H(c)}\mathbf{H}_{22}\mathbf{v}_2^{(c)}\mathbf{H}_{21}^H\mathbf{g}_2^{(c)} - \mathbf{H}_{21}^H\mathbf{g}_2^{(c)}\mathbf{v}_2^{H(c)}\mathbf{H}_{22}^H\mathbf{g}_2^{(c)})
\end{aligned}$$

$$
\begin{aligned}
\mathbf{v}_2^{*(c)} &= \underset{\mathbf{v}_2^{(c)}}{\operatorname{argmin}}\Big(\sum_{k=1}^{2} MSE_k^{(c)}\Big) \\
&= \Big[2\mathbf{H}_{12}^H\mathbf{g}_1^{(c)}\mathbf{g}_1^{H(c)}\mathbf{H}_{12} + 2\mathbf{H}_{22}^H\mathbf{g}_2^{(c)}\mathbf{g}_2^{H(c)}\mathbf{H}_{22})\Big]^{-1}(2\mathbf{H}_{12}^H\mathbf{g}_1^{(c)} + 2\mathbf{H}_{22}^H\mathbf{g}_2^{(c)} \\
&\quad - \mathbf{g}_1^{H(c)}\mathbf{H}_{11}\mathbf{v}_1^{(c)}\mathbf{H}_{12}^H\mathbf{g}_1^{(c)} - \mathbf{H}_{12}^H\mathbf{g}_1^{(c)}\mathbf{v}_1^{H(c)}\mathbf{H}_{11}^H\mathbf{g}_1^{(c)} - \mathbf{g}_2^{H(c)}\mathbf{H}_{21}\mathbf{v}_1^{(c)}\mathbf{H}_{22}^H\mathbf{g}_2^{(c)} - \mathbf{H}_{22}^H\mathbf{g}_2^{(c)}\mathbf{v}_1^{H(c)}\mathbf{H}_{21}^H\mathbf{g}_2^{(c)})
\end{aligned}
$$

2 System Model 2 : Cooperative Transmitters

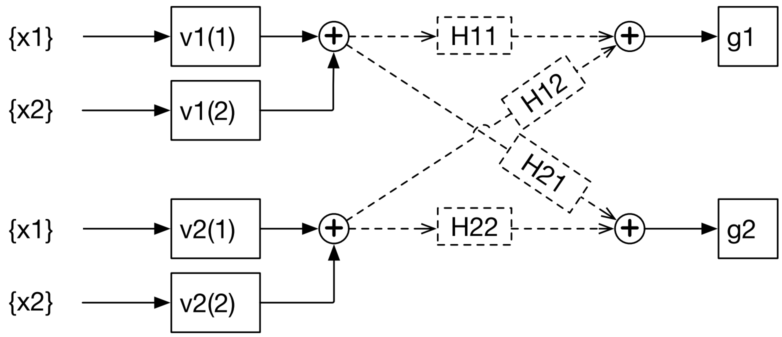

Figure 3: Forward Channel

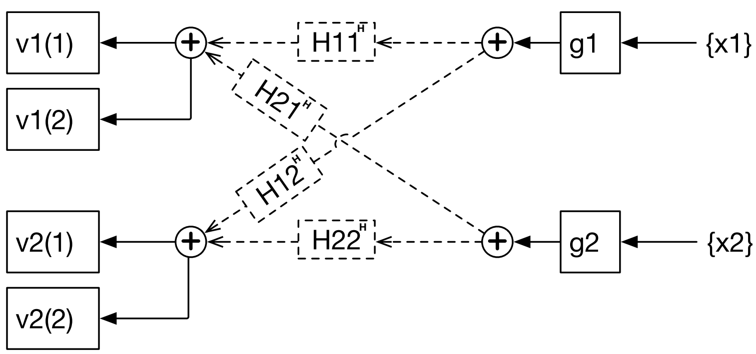

Figure 4: Backward Channel

2.1 Optimization Problem

$$\min_{\mathbf{v}_k^{(j)},\mathbf{g}_k} \sum_k w_k MSE_k$$

$$\text{subject to } \sum_j \|\mathbf{v}_k^{(j)}\|^2 \leq P; \|\mathbf{g}_k\|^2 \leq P$$

where $w_k \in R^+$ is weighting factor, and P is the power constraint of transmitters.

2.2 The received signal vector at k-th receiver

$$\mathbf{y}_k = \sum_i \left[\mathbf{H}_{ki} \sum_j (\mathbf{v}_i^{(j)} x_j) \right] + \mathbf{n}_k$$

2.3 SINR Derivation

Our actual objective is to maximize SINR of the system. Since SINR is inversely proportional to MSE, we can simply transform the problem into minimizing MSE of the system.

$$s_k = \mathbf{g}_k^H (\sum_i \mathbf{H}_{ki} \mathbf{v}_i^{(k)} x_k)$$

$$n_k = \mathbf{g}_k^H \left[\sum_i \big(\mathbf{H}_{ki} \sum_{j \neq k} \mathbf{v}_i^{(j)} x_j \big) + \mathbf{n}_k \right]$$

$$\frac{|s_k|^2}{|n_k|^2} = \frac{|\mathbf{g}_k^H \sum_i \mathbf{H}_{ki} \mathbf{v}_i^{(k)}|^2}{\sum_{j \neq k} |\mathbf{g}_k^H \sum_i \mathbf{H}_{ki} \mathbf{v}_i^{(j)}|^2 + |\mathbf{g}_k^H \mathbf{R}_k \mathbf{g}_k|}$$

2.4 Forward Direction

The solution is simply $R^{-1}p$, where R is the correlation matrix, and p is the cross-correlation vector.

$$\begin{aligned}
\mathbf{g}_k^* &= \underset{\mathbf{g}_k}{\operatorname{argmin}} (\sum_k \left[MSE_k + \lambda_i (\sum_j \|\mathbf{v}_i^{(j)}\| - P) \right]) \\
&= \left[\sum_j \left[(\sum_i \mathbf{H}_{ki} \mathbf{v}_i^{(j)}) (\sum_i \mathbf{v}_i^{H(j)} \mathbf{H}_{ki}^H) \right] + \sigma^2 \mathbf{I} \right]^{-1} (\sum_i \mathbf{H}_{ki} \mathbf{v}_i^{(k)})
\end{aligned}$$

2.5 Backward Direction: Iterative Method which Doesn't Work

In this method, we found a solution for each $\mathbf{v}_k^{(j)}$ depending on some $\mathbf{v}_x^{(j)}$, where k $\neq$ x. For example, $\mathbf{v}_1^{(1)}$ depends on $\mathbf{v}_2^{(1)}$. Then, we try to solve the optimization problem by iterating between $\mathbf{v}_k^{(j)}$ and $\mathbf{v}_x^{(j)}$, and hope it will work. However, our numerical experiment shows that it will not converge for some channels.

$$
\begin{aligned}
MSE_k &= \mathrm{E}[(x_k - \mathbf{g}_k^H \mathbf{y}_k)(x_k - \mathbf{g}_k^H \mathbf{y}_k)^H] \\
&= \mathrm{E}[x_k^2] - \mathrm{E}[x_k \mathbf{y}_k^H \mathbf{g}_k] - \mathrm{E}[x_k \mathbf{g}_k^H \mathbf{y}_k] + \mathrm{E}[\mathbf{g}_k^H \mathbf{y}_k \mathbf{y}_k^H \mathbf{g}_k] \\
&= 1 - \sum_i \mathbf{v}_i^{H(k)} \mathbf{H}_{ki}^H \mathbf{g}_k - \mathbf{g}_k^H \sum_i \mathbf{H}_{ki} \mathbf{v}_i^{(k)} + \mathbf{g}_k^H \sum_j \left[(\sum_i \mathbf{H}_{ki} \mathbf{v}_i^{(j)})(\sum_i \mathbf{v}_i^{H(j)} \mathbf{H}_{ki}^H) \right] \mathbf{g}_k + \sigma^2 \mathbf{g}_k^H \mathbf{g}_k
\end{aligned}
$$

$$
\begin{aligned}
\mathbf{v}_1^{*(1)} &= \underset{\mathbf{v}_1^{(1)}}{\mathrm{argmin}} \Big(\sum_{k=1,2} \left[MSE_k + \lambda_k \Big(\sum_{j=1,2} \|\mathbf{v}_k^{(j)}\| - P \Big) \right] \Big) \\
&= \left[2\mathbf{H}_{11}^H \mathbf{g}_1 \mathbf{g}_1^H \mathbf{H}_{11} w_1 + 2\mathbf{H}_{21}^H \mathbf{g}_2 \mathbf{g}_2 \mathbf{H}_{21} w_2 + 2\lambda_1^* I \right]^{-1} (2\mathbf{H}_{11}^H \mathbf{g}_1 w_1 - \mathbf{g}_1^H \mathbf{H}_{12} \mathbf{v}_2^{(1)} \mathbf{H}_{11}^H \mathbf{g}_1 w_1 \\
&\quad - \mathbf{H}_{11}^H \mathbf{g}_1 \mathbf{v}_2^{H(1)} \mathbf{H}_{12}^H \mathbf{g}_1 w_1 - \mathbf{g}_2^H \mathbf{H}_{22} \mathbf{v}_2^{(1)} \mathbf{H}_{21}^H \mathbf{g}_2 w_2 - \mathbf{H}_{21}^H \mathbf{g}_2 \mathbf{v}_2^{H(1)} \mathbf{H}_{22}^H \mathbf{g}_2 w_2) \\
&= \left[2\mathrm{E}[x_1^* \mathbf{y}_1] \mathrm{E}[x_1^* \mathbf{y}_1]^H w_1 + 2\mathrm{E}[x_2^* \mathbf{y}_1] \mathrm{E}[x_2^* \mathbf{y}_1]^H w_2 + 2\lambda_1^* I \right]^{-1} (2\mathrm{E}[x_1^* \mathbf{y}_1] w_1 - \mathrm{E}[x_1^* \mathbf{y}_2]^H \mathbf{v}_2^{(1)} \mathrm{E}[x_1^* \mathbf{y}_1] w_1 \\
&\quad - \mathrm{E}[x_1^* \mathbf{y}_1] \mathbf{v}_2^{H(1)} \mathrm{E}[x_1^* \mathbf{y}_2] w_1 - \mathrm{E}[x_2^* \mathbf{y}_2]^H \mathbf{v}_2^{(1)} \mathrm{E}[x_2^* \mathbf{y}_1] w_2 - \mathrm{E}[x_2^* \mathbf{y}_1] \mathbf{v}_2^{H(1)} \mathrm{E}[x_2^* \mathbf{y}_2] w_2)
\end{aligned}
$$

$$
\begin{aligned}
\mathbf{v}_2^{*(1)} &= \underset{\mathbf{v}_2^{(1)}}{\mathrm{argmin}} \Big(\sum_{k=1,2} \left[MSE_k + \lambda_k \Big(\sum_{j=1,2} \|\mathbf{v}_k^{(j)}\| - P \Big) \right] \Big) \\
&= \left[2\mathbf{H}_{12}^H \mathbf{g}_1 \mathbf{g}_1^H \mathbf{H}_{12} w_1 + 2\mathbf{H}_{22}^H \mathbf{g}_2 \mathbf{g}_2^H \mathbf{H}_{22} w_2 + 2\lambda_2^* I) \right]^{-1} (2\mathbf{H}_{12}^H \mathbf{g}_1 w_1 - \mathbf{g}_1^H \mathbf{H}_{11} \mathbf{v}_1^{(1)} \mathbf{H}_{12}^H \mathbf{g}_1 w_1 \\
&\quad - \mathbf{H}_{12}^H \mathbf{g}_1 \mathbf{v}_1^{H(1)} \mathbf{H}_{11}^H \mathbf{g}_1 w_1 - \mathbf{g}_2^H \mathbf{H}_{21} \mathbf{v}_1^{(1)} \mathbf{H}_{22}^H \mathbf{g}_2 w_2 - \mathbf{H}_{22}^H \mathbf{g}_2 \mathbf{v}_1^{H(1)} \mathbf{H}_{21}^H \mathbf{g}_2 w_2) \\
&= \left[2\mathrm{E}[x_1^* \mathbf{y}_2] \mathrm{E}[x_1^* \mathbf{y}_2]^H w_1 + 2\mathrm{E}[x_2^* \mathbf{y}_2] \mathrm{E}[x_2^* \mathbf{y}_2]^H w_2 + 2\lambda_2^* I \right]^{-1} (2\mathrm{E}[x_1^* \mathbf{y}_2] w_1 - \mathrm{E}[x_1^* \mathbf{y}_1]^H \mathbf{v}_1^{(1)} \mathrm{E}[x_1^* \mathbf{y}_2] w_1 \\
&\quad - \mathrm{E}[x_1^* \mathbf{y}_2] \mathbf{v}_1^{H(1)} \mathrm{E}[x_1^* \mathbf{y}_1] w_1 - \mathrm{E}[x_2^* \mathbf{y}_1]^H \mathbf{v}_1^{(1)} \mathrm{E}[x_2^* \mathbf{y}_2] w_2 - \mathrm{E}[x_2^* \mathbf{y}_2] \mathbf{v}_1^{H(1)} \mathrm{E}[x_2^* \mathbf{y}_2] w_2)
\end{aligned}
$$

$$
\begin{aligned}
\mathbf{v}_1^{*(2)} &= \underset{\mathbf{v}_1^{(2)}}{\operatorname{argmin}}(\sum_{k=1,2}\left[MSE_k + \lambda_k(\sum_{j=1,2}\|\mathbf{v}_k^{(j)}\| - P)\right]) \\
&= \left[2\mathbf{H}_{11}^H\mathbf{g}_1\mathbf{g}_1^H\mathbf{H}_{11}w_1 + 2\mathbf{H}_{21}^H\mathbf{g}_2\mathbf{g}_2\mathbf{H}_{21}w_2 + 2\lambda_1^* I\right]^{-1}(2\mathbf{H}_{21}^H\mathbf{g}_2 w_2 - \mathbf{g}_1^H\mathbf{H}_{12}\mathbf{v}_2^{(2)}\mathbf{H}_{11}^H\mathbf{g}_1 w_1 \\
&\quad - \mathbf{H}_{11}^H\mathbf{g}_1\mathbf{v}_2^{H(2)}\mathbf{H}_{12}^H\mathbf{g}_1 w_1 - \mathbf{g}_2^H\mathbf{H}_{22}\mathbf{v}_2^{(2)}\mathbf{H}_{21}^H\mathbf{g}_2 w_2 - \mathbf{H}_{21}^H\mathbf{g}_2\mathbf{v}_2^{H(2)}\mathbf{H}_{22}^H\mathbf{g}_2 w_2) \\
&= \left[2\mathrm{E}[x_1^*\mathbf{y}_1]\mathrm{E}[x_1^*\mathbf{y}_1]^H w_1 + 2\mathrm{E}[x_2^*\mathbf{y}_1]\mathrm{E}[x_2^*\mathbf{y}_1]^H w_2 + 2\lambda_1^* I\right]^{-1}(2\mathrm{E}[x_2^*\mathbf{y}_1]w_2 - \mathrm{E}[x_1^*\mathbf{y}_2]^H\mathbf{v}_2^{(2)}\mathrm{E}[x_1^*\mathbf{y}_1]w_1 \\
&\quad - \mathrm{E}[x_1^*\mathbf{y}_1]\mathbf{v}_2^{H(2)}\mathrm{E}[x_1^*\mathbf{y}_2]w_1 - \mathrm{E}[x_2^*\mathbf{y}_2]^H\mathbf{v}_2^{(2)}\mathrm{E}[x_2^*\mathbf{y}_1]w_2 - \mathrm{E}[x_2^*\mathbf{y}_1]\mathbf{v}_2^{H(2)}\mathrm{E}[x_2^*\mathbf{y}_2]w_2)
\end{aligned}
$$

$$
\begin{aligned}
\mathbf{v}_2^{*(2)} &= \underset{\mathbf{v}_2^{(2)}}{\operatorname{argmin}}(\sum_{k=1,2}\left[MSE_k + \lambda_k(\sum_{j=1,2}\|\mathbf{v}_k^{(j)}\| - P)\right]) \\
&= \left[2\mathbf{H}_{12}^H\mathbf{g}_1\mathbf{g}_1^H\mathbf{H}_{12}w_1 + 2\mathbf{H}_{22}^H\mathbf{g}_2\mathbf{g}_2^H\mathbf{H}_{22}w_2 + 2\lambda_2^* I)\right]^{-1}(2\mathbf{H}_{22}^H\mathbf{g}_2 w_2 - \mathbf{g}_1^H\mathbf{H}_{11}\mathbf{v}_1^{(2)}\mathbf{H}_{12}^H\mathbf{g}_1 w_1 \\
&\quad - \mathbf{H}_{12}^H\mathbf{g}_1\mathbf{v}_1^{H(2)}\mathbf{H}_{11}^H\mathbf{g}_1 w_1 - \mathbf{g}_2^H\mathbf{H}_{21}\mathbf{v}_1^{(2)}\mathbf{H}_{22}^H\mathbf{g}_2 w_2 - \mathbf{H}_{22}^H\mathbf{g}_2\mathbf{v}_1^{H(2)}\mathbf{H}_{21}^H\mathbf{g}_2 w_2) \\
&= \left[2\mathrm{E}[x_1^*\mathbf{y}_2]\mathrm{E}[x_1^*\mathbf{y}_2]^H w_1 + 2\mathrm{E}[x_2^*\mathbf{y}_2]\mathrm{E}[x_2^*\mathbf{y}_2]^H w_2 + 2\lambda_2^* I\right]^{-1}(2\mathrm{E}[x_2^*\mathbf{y}_2]w_2 - \mathrm{E}[x_1^*\mathbf{y}_1]^H\mathbf{v}_1^{(2)}\mathrm{E}[x_1^*\mathbf{y}_2]w_1 \\
&\quad - \mathrm{E}[x_1^*\mathbf{y}_2]^H\mathbf{v}_1^{H(2)}\mathrm{E}[x_1^*\mathbf{y}_1]w_1 - \mathrm{E}[x_2^*\mathbf{y}_1]^H\mathbf{v}_1^{(2)}\mathrm{E}[x_2^*\mathbf{y}_2]w_2 - \mathrm{E}[x_2^*\mathbf{y}_2]^H\mathbf{v}_1^{H(2)}\mathrm{E}[x_2^*\mathbf{y}_1]w_2)
\end{aligned}
$$

2.6 Backward Direction: Duality Method

For the duality method, if the optimization problem is convex, then the optimal solution is guaranteed. If it is not, there will exist a **duality gap** which will not give the optimal solution[Convex Optimization, Boyd]. Since our problem is non-convex, getting the optimal solution is possible but not guaranteed. The same problem with single constraint(only contains one λ) has been solved[Boyd], but not for the cases with multiple constraints.

2.6.1 Convexity Analysis

We could rewrite our cost function in

$$\sum_k w_k MSE_k = (\sum_k w_k) - \mathbf{kv} - (\mathbf{kv})^H + \mathbf{v}^H\mathbf{A}\mathbf{v} + \sigma^2\mathbf{g}^H\mathbf{g}$$

where $\mathbf{k}$ is a constant row vector with channel information, $\mathbf{A}$ is a constant matrix with channel information, and $\boldsymbol{\lambda}$ is a diagonal matrix with each $\boldsymbol{\lambda}_{ii}$ to be some λ_k. Since A is **not a positive definite** matrix, the cost function is non-convex. In addition, $\mathbf{A}$ has very similar properties to the **correlation matrix** in adaptive filter theory. Due to no additional noise on transmitter side, $\mathbf{A}$ is singular.

2.6.2 Primal Problem

It's the same as the optimization problem in 2.1.

2.6.3 Primal Algorithm

$$\min_{\mathbf{v}_k^{(j)}} \ L(\mathbf{v}, \boldsymbol{\lambda}) = \sum_k \left[w_k MSE_k + \lambda_k (\sum_j \|\mathbf{v}_k^{(j)}\| - P) \right]$$

Since the Lagrangian L is **not always convex(depends on λ_k)**, using gradient descent can only give **local optimum**. There do exist better ways to deal with non-convex problems, with potential to find global optimum, such as **line search with Wolfe condition**. However, due to the enormous complexity of Wolfe algorithm, we choose to stay with gradient descent. The gradient is

$$\frac{\partial L(\mathbf{v}, \boldsymbol{\lambda})}{\partial \mathbf{v}} = -2\mathbf{k} + 2\mathbf{v}^H [\mathbf{A} + \boldsymbol{\lambda}]$$

and the algorithm is

$$\mathbf{v}(i+1) = \mathbf{v}(i) - \mu_{\mathbf{v}} (\frac{\partial L(\mathbf{v}, \boldsymbol{\lambda})}{\partial \mathbf{v}})^H$$

where $\mu_{\mathbf{v}}$ is a fixed stepsize.

2.6.4 Dual Problem

$$\max_{\boldsymbol{\lambda} \geq \mathbf{0}} G(\mathbf{v}^*, \boldsymbol{\lambda}) \ = \max_{\boldsymbol{\lambda} \geq \mathbf{0}} [\min_{\mathbf{v}_k^{(j)}} \ L(\mathbf{v}, \boldsymbol{\lambda})]$$

The **feasible set** of λ_k is positive number set. The reason could be found in textbook.

2.6.5 Dual Algorithm

By the optimization theory, dual problems are **always** concave. Therefore, we could get the optimal $\boldsymbol{\lambda}$ simply with gradient algorithm. The gradient is shown below.

$$\frac{\partial G(\mathbf{v}^*, \boldsymbol{\lambda})}{\partial \lambda_k} = \sum_j \|\mathbf{v}_k^{(j)}\| - P$$

The algorithm is

$$\boldsymbol{\lambda}(i+1) = \boldsymbol{\lambda}(i) + \mu_{\boldsymbol{\lambda}}(\frac{\partial L(\mathbf{v}^*, \boldsymbol{\lambda})}{\partial\ \boldsymbol{\lambda}})^T, \boldsymbol{\lambda} \geq \mathbf{0}$$

where $\mu_{\boldsymbol{\lambda}}$ is a fixed stepsize.

2.7 Backward Direction: Noisy Transmitter Approach

In the duality method section, we've seen that our cost function is non-convex. However, in practical situation, there must exist some noise at transmitters. Thus, with the presence of noise, the system becomes convex. With this assumption, we could get the analytical solution for this problem. We get

$$\mathbf{v}^{*H} = \mathbf{k}[\mathbf{A} + \boldsymbol{\lambda}]^{-1}, \forall \lambda_k \geq 0$$

The corresponding dual problem is

$$\max_{\boldsymbol{\lambda} \geq \mathbf{0}} G(\mathbf{v}^*, \boldsymbol{\lambda}) \ = \max_{\boldsymbol{\lambda} \geq \mathbf{0}}[\min_{\mathbf{v}_k^{(j)}}\ L(\mathbf{v}, \boldsymbol{\lambda})] = \max_{\boldsymbol{\lambda} \geq \mathbf{0}}\ \sum_k (w1 + w2 + w3) - \mathbf{v}^{*H}\mathbf{k}^H + \sigma^2 \mathbf{g}^H \mathbf{g} - P(\lambda_1 + \lambda_2 + \lambda_3)$$

We could actually find the **analytic solution** for $\boldsymbol{\lambda}$. However, for 3 users case, the process includes finding the **inverse of an 6x6 symbolic matrix** which is **solvable** but **complicated**, let alone cases above 3 users. Thus, we choose to solve the dual problem with **numerical method**. By the optimization theory, dual problems are **always** convex. Therefore, we could get the optimal $\boldsymbol{\lambda}$ simply with gradient algorithm. The gradient is shown below.

$$\frac{\partial G(\mathbf{v}^*, \boldsymbol{\lambda})}{\partial \lambda_k} = -P + \mathrm{Tr}[(A + \boldsymbol{\lambda})^{-T}\mathbf{k}^H\mathbf{k}(A + \boldsymbol{\lambda})^{-T}]$$

2.8 Turning into Bi-Directional Training

There will be some issues if we want to transform the optimization problem into bi-directional training algorithm. First, **the algorithm we provided doesn't really solve the optimization problem.** The reason is that we deal the variables $\mathbf{v}$ and $\mathbf{g}$ separately. In order to get the optimizer of this problem, we need to find the optimizer over a single variable [$\mathbf{v}$,$\mathbf{g}$]. **The second issue is about the power constraint of g**. For $\mathbf{v}$, the magnitude is controlled by the lagrange multiplier λ. However, for $\mathbf{g}$, we don't have such mechanism. Thus, we adopt the following **scaling method**:

$$\mathbf{g}_k = \begin{cases} \mathbf{g}_k & \text{if } \|\mathbf{g}_k\|^2 \leq P \\ \frac{\sqrt{P}}{\|\mathbf{g}_k\|}\mathbf{g}_k & \text{if } \|\mathbf{g}_k\|^2 > P \end{cases}$$

2.9 Numerical Results

For numerical simulation, we compare the performance of 3 different algorithms/models. By simple transmitter, we mean system model in [Shi, 2014]. The diagram of the simple transmitter is also shown below. First thing to notice is **we set some of the power constraints to be $\leq P$ or $= P$**. The reason behind this is there really is **no noticeable difference in numerical results**(I've done both of them). The second thing is that we can only guarantee that method 1 will perform best. The reason why method 2 may not be better than method 3 is simple: Transmitters in method 2 cannot cooperate with each other, which may produce additional interference.

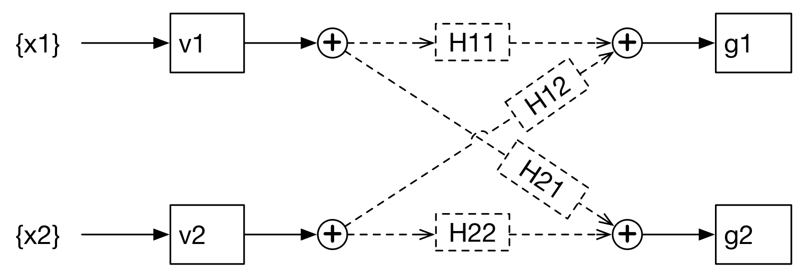

Figure 5: Simple Transmitters

	Coop.(Duality)	Coop.($R^{-1}p$)	Simple
System Model	2	2	Simple
Forward Alg.	$R^{-1}p$	$R^{-1}p$	$R^{-1}p$
Backward Alg.	Duality	$R^{-1}p$	$R^{-1}p$
g constraint	$\leq P$	$\leq P$ or $= P$	$\leq P$ or $= P$
v constraint	$\leq P$	$= P$	$= P$

Table 1: 3 different algorithms/models

2.9.1 Sum Capacity vs. Iterations

A. Iteration Count: Start with backward training. A full cycle of backward then forward training counts as 1 iteration.
B. Parameters:
1. Number of User $k = 3$
2. Power Constraint $P = 1$
3. Noise Variance n_0=$\mathbb{E}[\mathbf{n}_k^H \mathbf{n}_k]$ ranges from 10^{-3} to 10^{-1}
4. Results are averaged over 100 random complex gaussian channels.
5. Channel Capacity $C = \log_2(1 + SINR)$

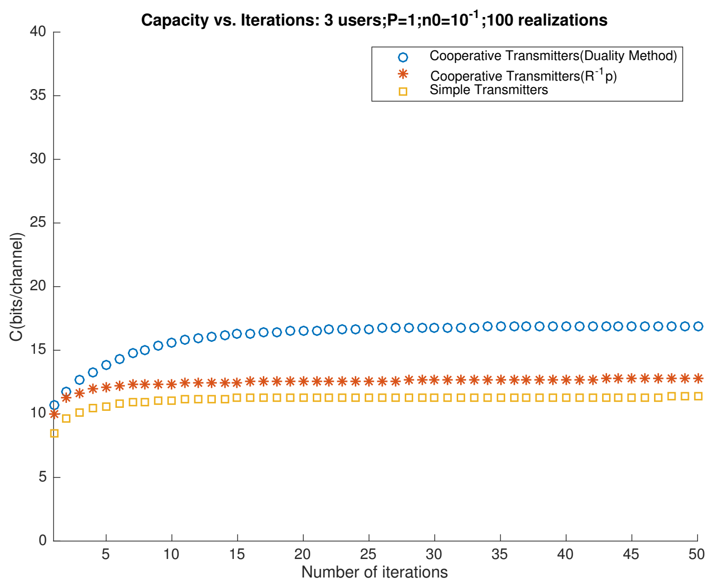

Figure 6: Noise Variance $n0 = 10^{-1}$

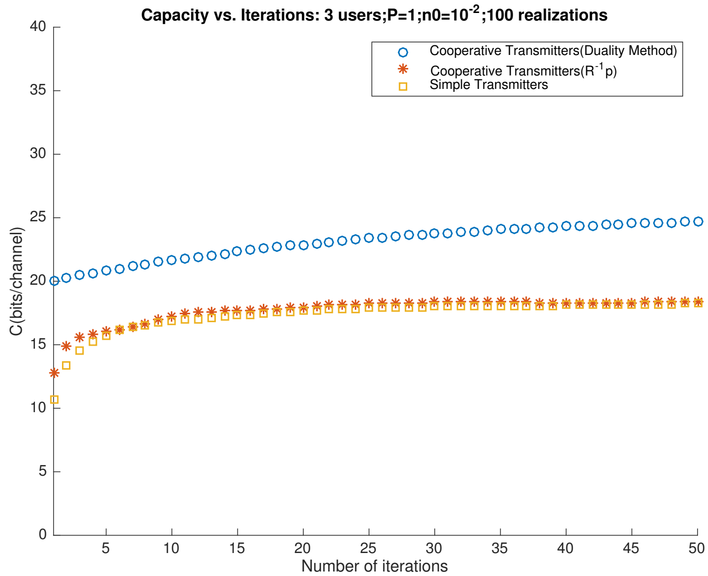

Figure 7: Noise Variance $n0 = 10^{-2}$

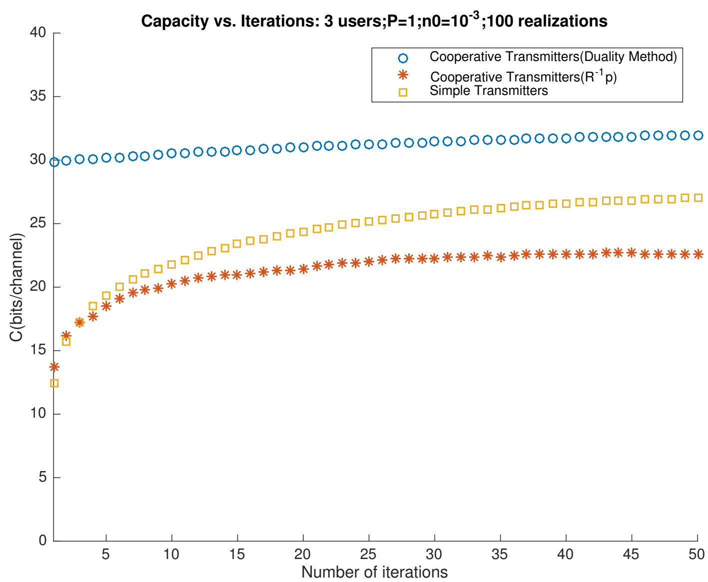

Figure 8: Noise Variance $n0 = 10^{-3}$

2.9.2 Sum Capacity vs. Noise

A. Convergence Criterion: As we can see from the previous example, filters tend to converge at **20th iteration**. Thus, we choose 20 iterations for doing this simulation.
B. Parameters:
1. Number of User $k = 3$
2. Power Constraint $P = 1$
3. Noise Variance n_0=$\mathbb{E}[\mathbf{n}_k^H \mathbf{n}_k]$ ranges from 10^{-3} to 1
4. Results are averaged over 10 random complex gaussian channels.
5. Channel Capacity $C = \log_2(1 + SINR)$

3 System Model 3 : Augmented Receivers

In this model, we make each receivers(mobile station) possess the ability to assist channel estimation in backward direction. Since it is rather easy to add a filter to receivers, if this method does improve the capacity, it will be extremely fascinating. We solve each transceivers with Wiener-Hopf equation: $R^{-1}p$.

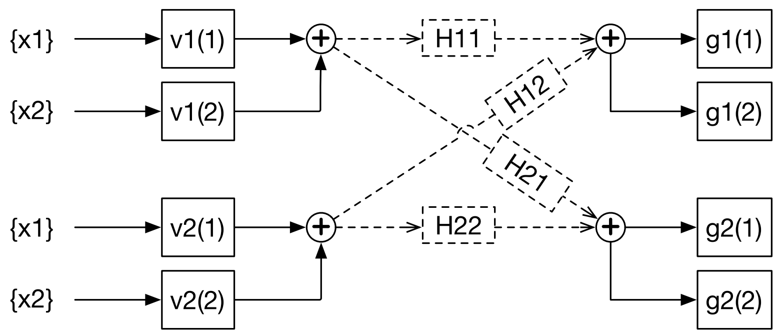

Figure 9: Forward Channel

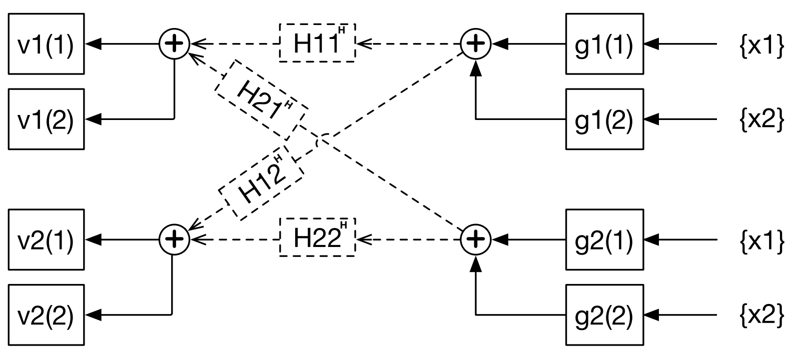

Figure 10: Backward Channel

3.1 Optimization Problem

$$\min_{\mathbf{v}_k^{(j)}, \mathbf{g}_k} \sum_k w_k MSE_k$$

$$\text{subject to } \sum_j \|\mathbf{v}_k^{(j)}\|^2 = P; \|\mathbf{g}_k\|^2 = P$$

3.2 Numerical Results

This is a 3 users, and 2X2 MIMO interference channel, with SNR equals to 20dB. The results are averaged over 1000 complex gaussian channels. X-axis represents the number of iterations: start from backward direction and end in forward direction. Y-axis represents the sum capacity of three users. The first thing we can tell is the **augmented one** is a lot worse than the **simple one**(Only one filter at each transceivers. For example, for 2 users case, only have $v_1^{(1)}$,$v_2^{(2)}$,$g_1^{(1)}$,$g_2^{(2)}$). **This is because transmitters distribute all messages to all users.** In the experiment, transmitter 1 sends 2 bits of message 1 to user 1, 2, and 3. However user 2 and 3 simply drop message 1 off(they don't need it). Thus, the sum capacity of augmented receivers only have 1/3 of capacity compared with the simple one.

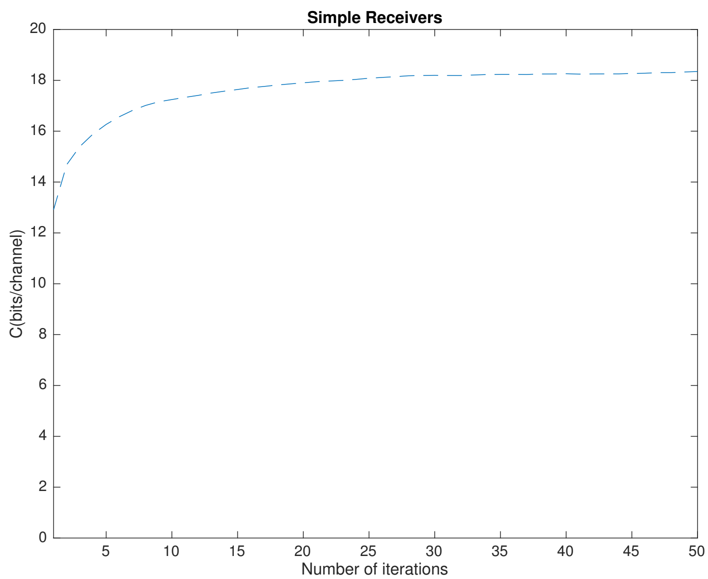

Figure 11: Simple Receivers

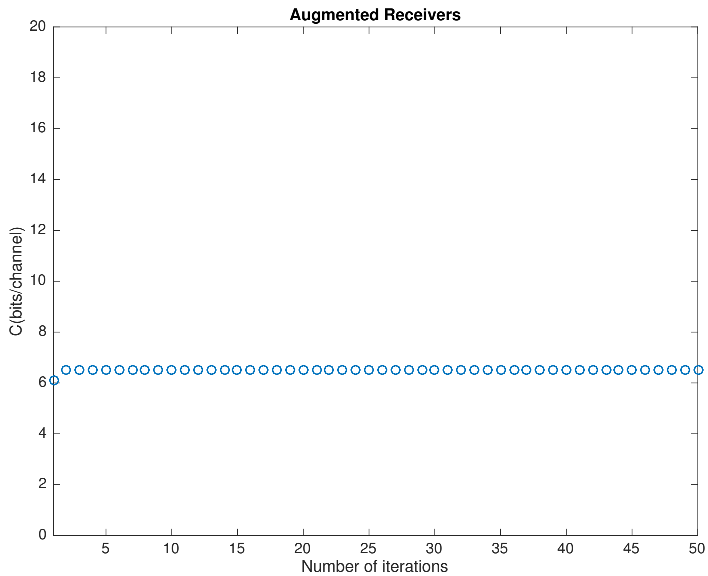

Figure 12: Augmented Receivers

There's an interesting result of λ_k. This is, if we **change our constraint from equality to inequality**, the duality method will give the same answer. Although this is not too surprising, it is justified by mathematics. 9. Numerical Simulation(2 Users, 2X2 MIMO Channel)

$$Rayleigh\ Fading\ Channel$$

$$Cross\ Channel\ Gain = 0.8 * Direct\ Channel\ Gain$$

$$SNR = \frac{1}{\sigma^2} = 10^3 = 30dB$$

$$Observation1: If\ the\ training\ length\ is\ long\ enough,\ each\ LMS\ filter\ (\mathbf{v}_k^{(c)}, \mathbf{v}_k^{(p)}, \mathbf{g}_k^{(c)}, \mathbf{g}_k^{(p)},)\ will\ converge\ to\ Wiener\ filter$$

$$Observation2: \ Use\ Wiener\ filters,\ and\ only\ send\ common\ messages.\ Sum\ rate\ C\ =\ 11.6\ bit/channel$$

$Observation3: \ Use \ Wiener \ filters, \ and \ only \ send \ private \ messages. \ Sum \ rate \ C \ = \ 2.63 \ bit/channel$

$Observation4: \ Use \ Wiener \ filters, \ and \ send \ both \ messages. \ Sum \ rate \ C \ = \ 3.35 \ bit/channel$

$Observation5: \ Under \ the \ cooperation \ scheme, \ transmitters \ don't \ converge \ to \ Wiener \ filters$

$$Observation5: \ Under \ the \ cooperation \ scheme, \ C = 3.15 \ bit/channel$$

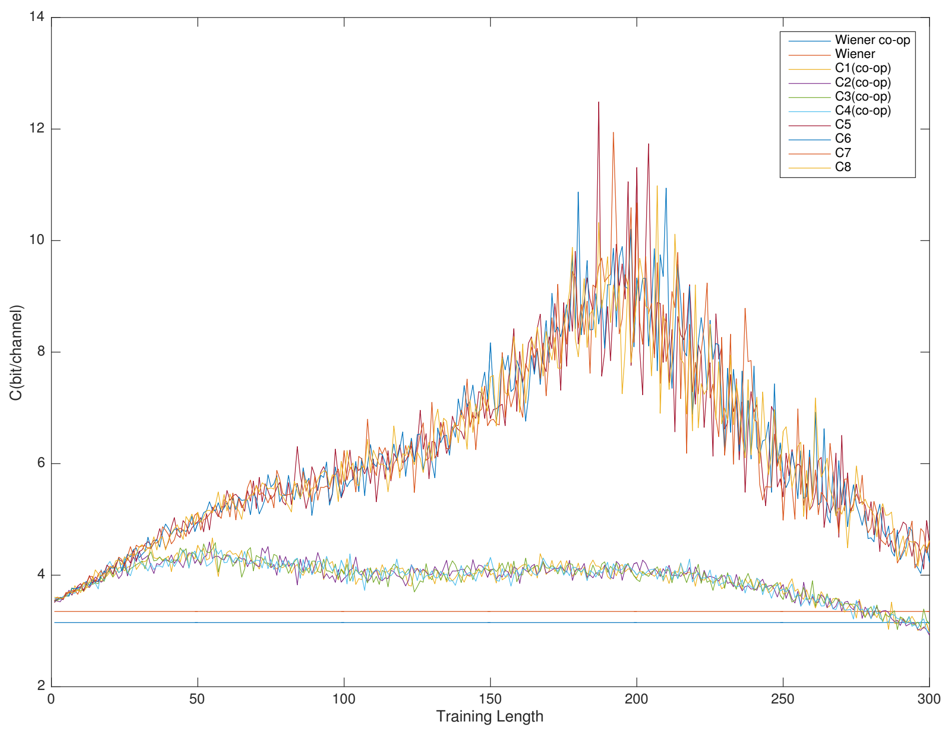

Figure 13: Insert caption

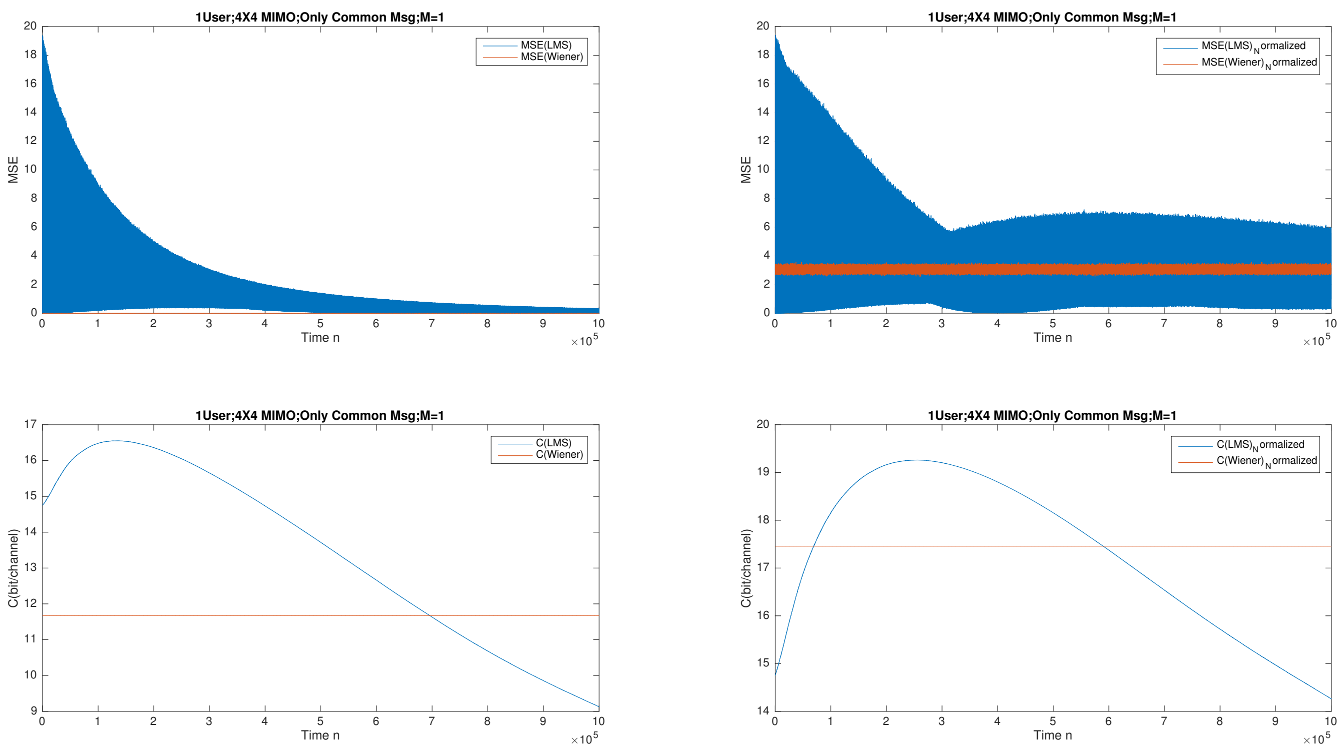

Figure 14: Insert caption

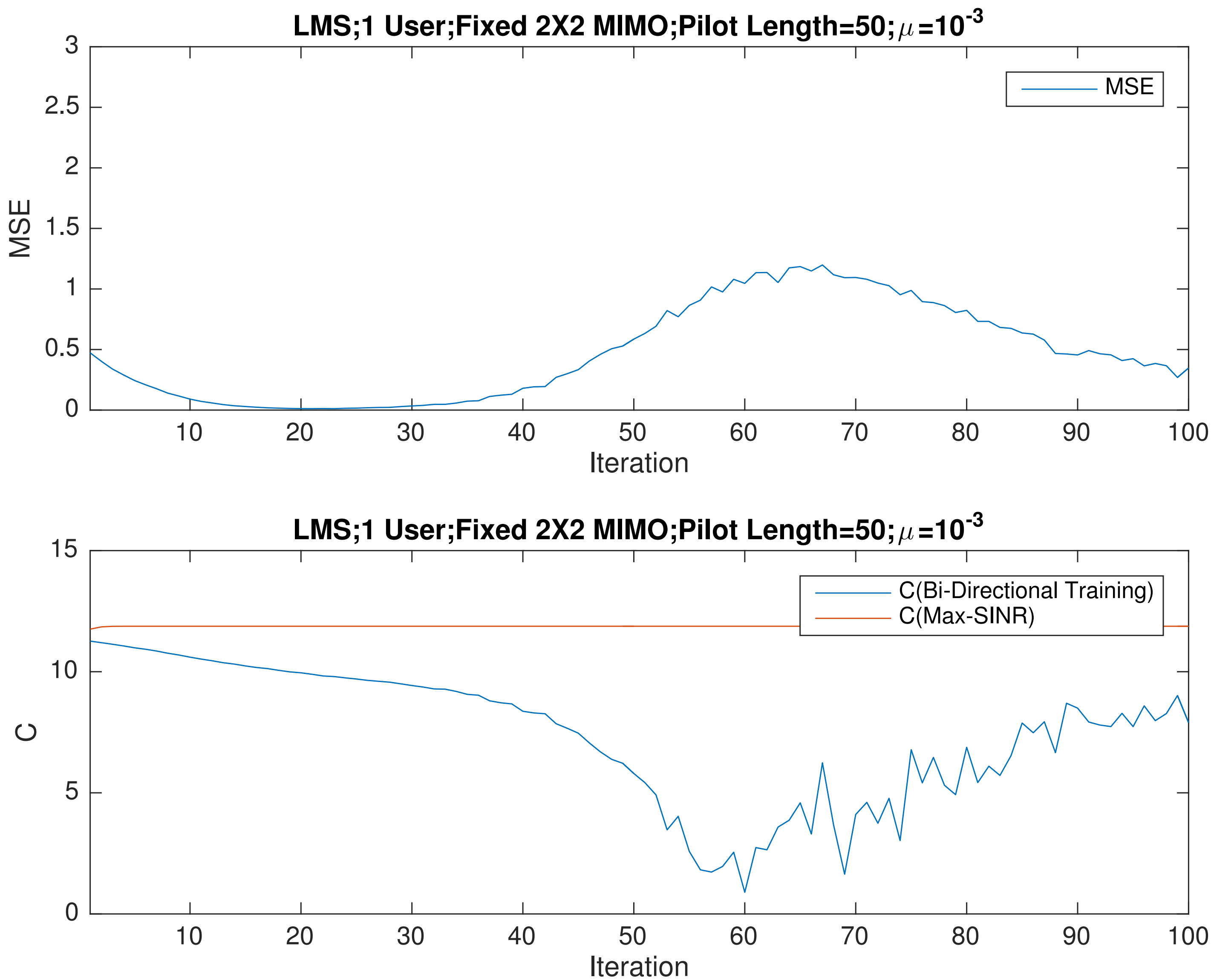

Figure 15: Insert caption

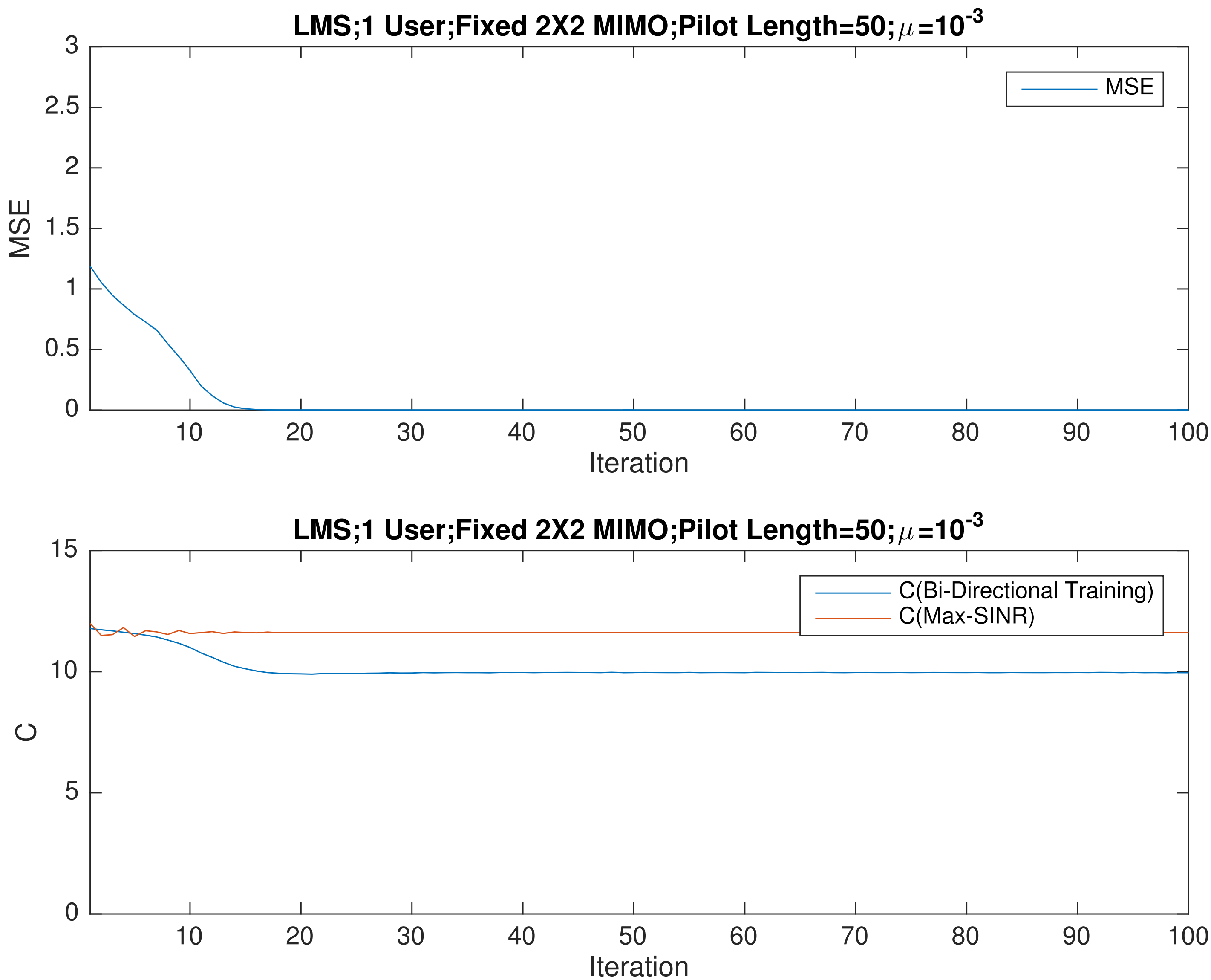

Figure 16: Insert caption

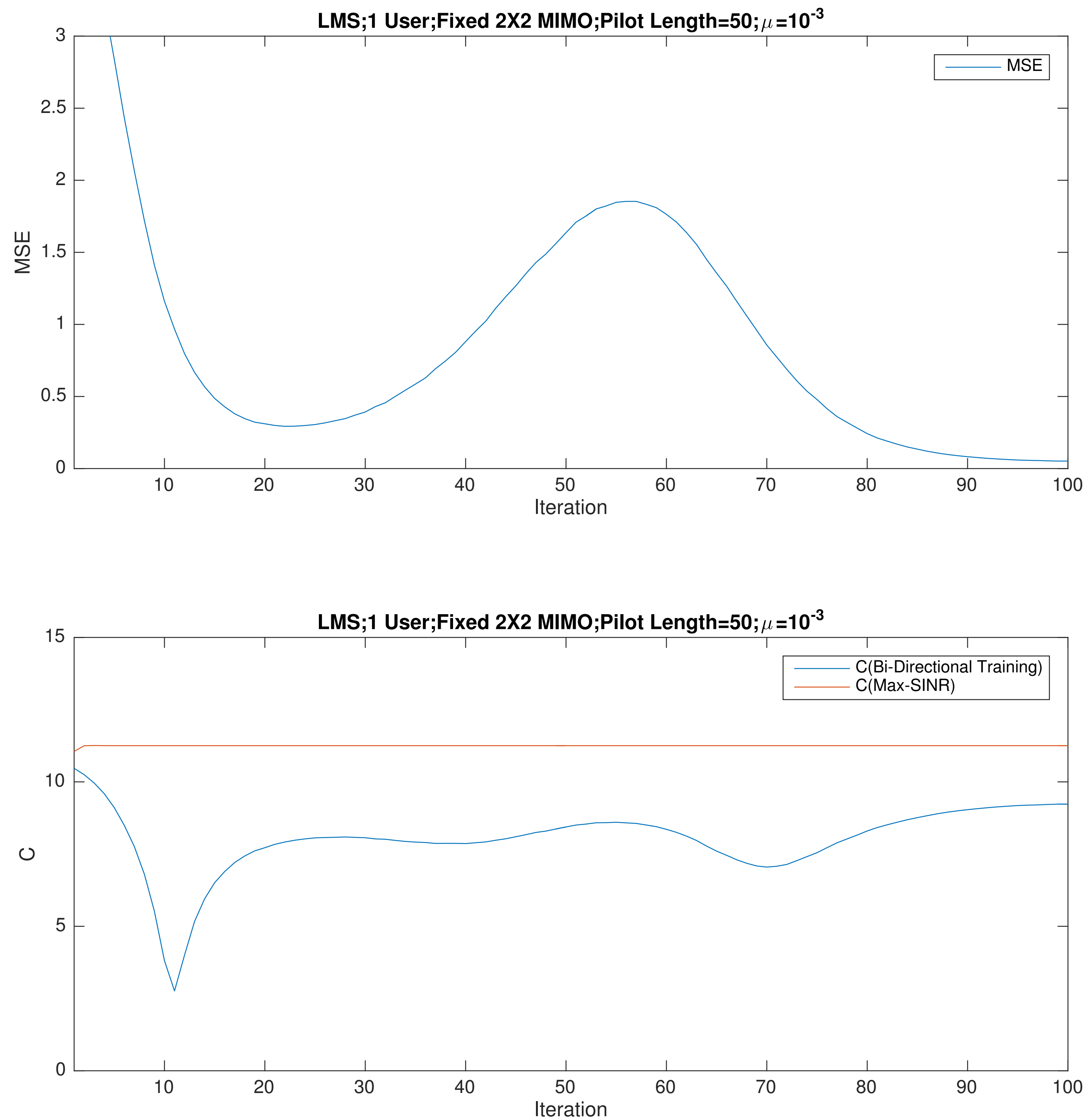

Figure 17: Insert caption

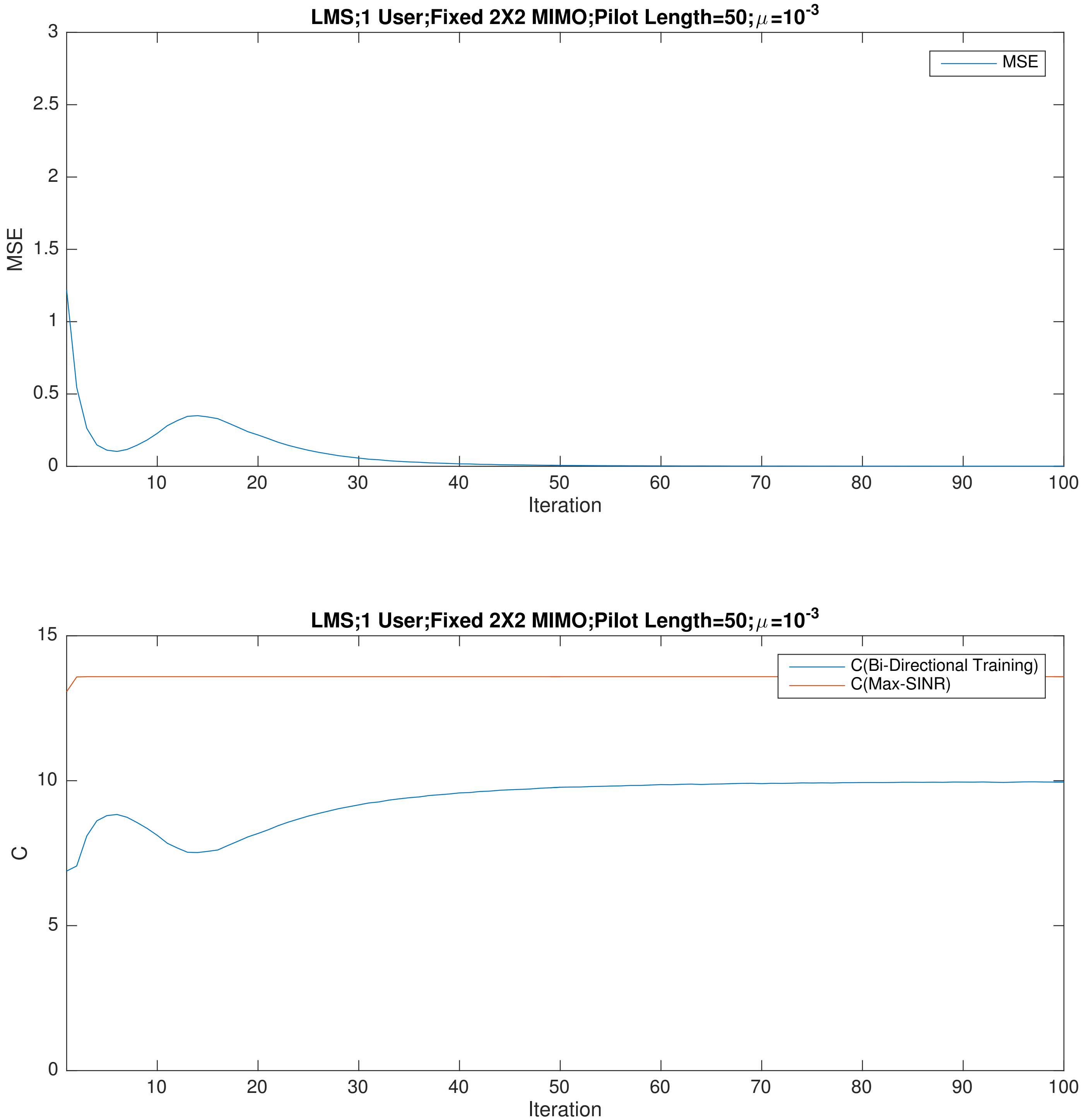

Figure 18: Insert caption

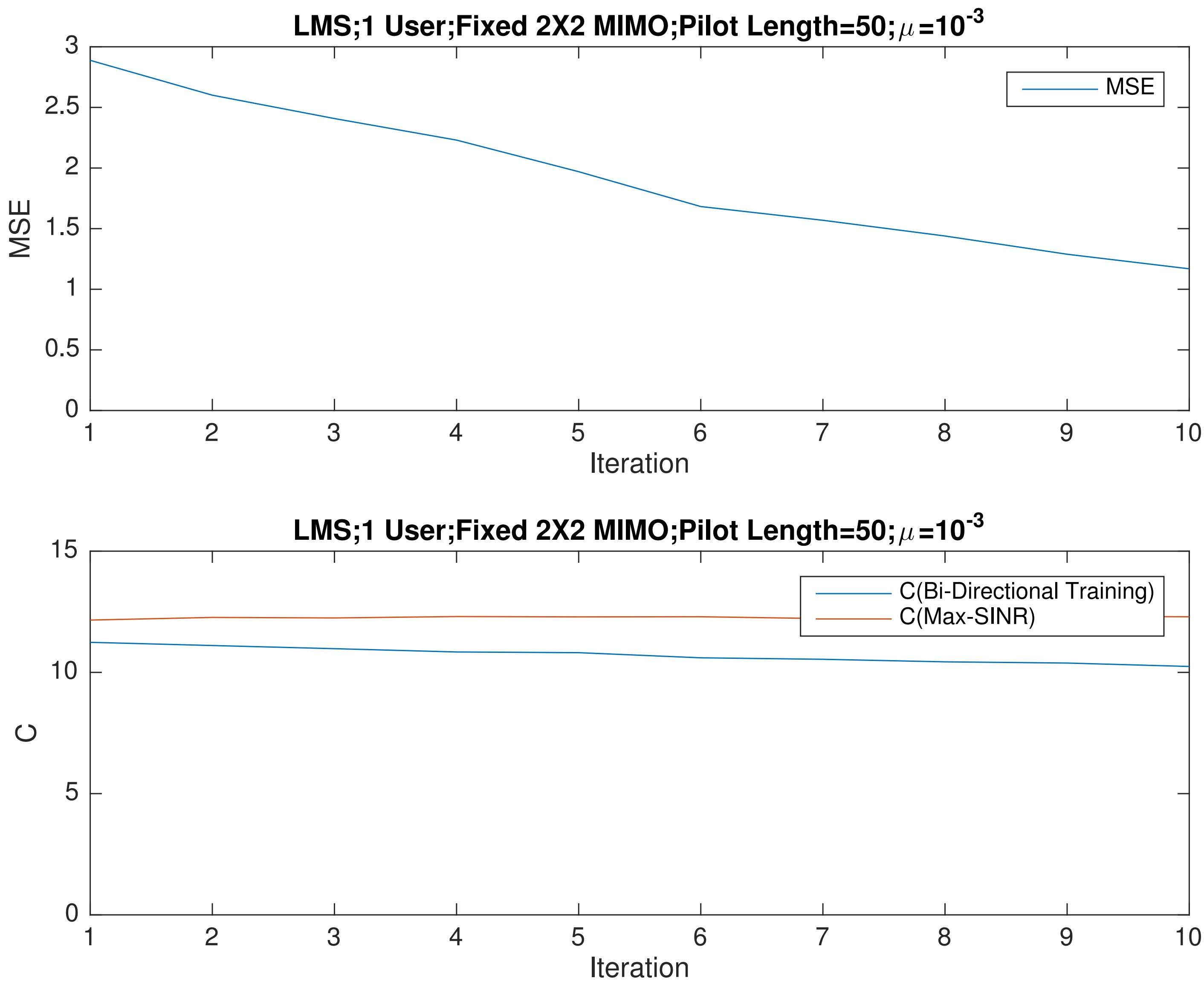

Figure 19: Insert caption

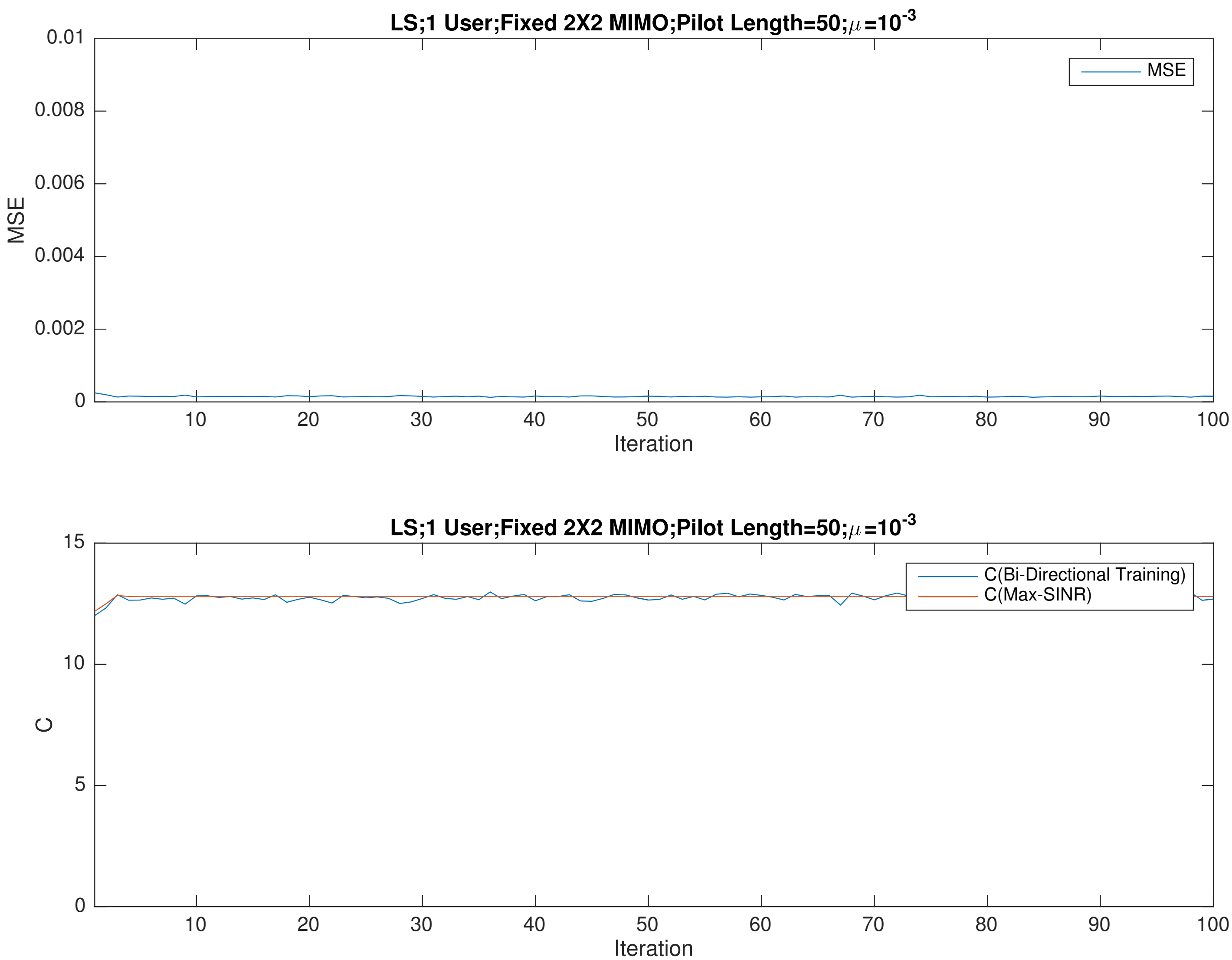

Figure 20: Insert caption

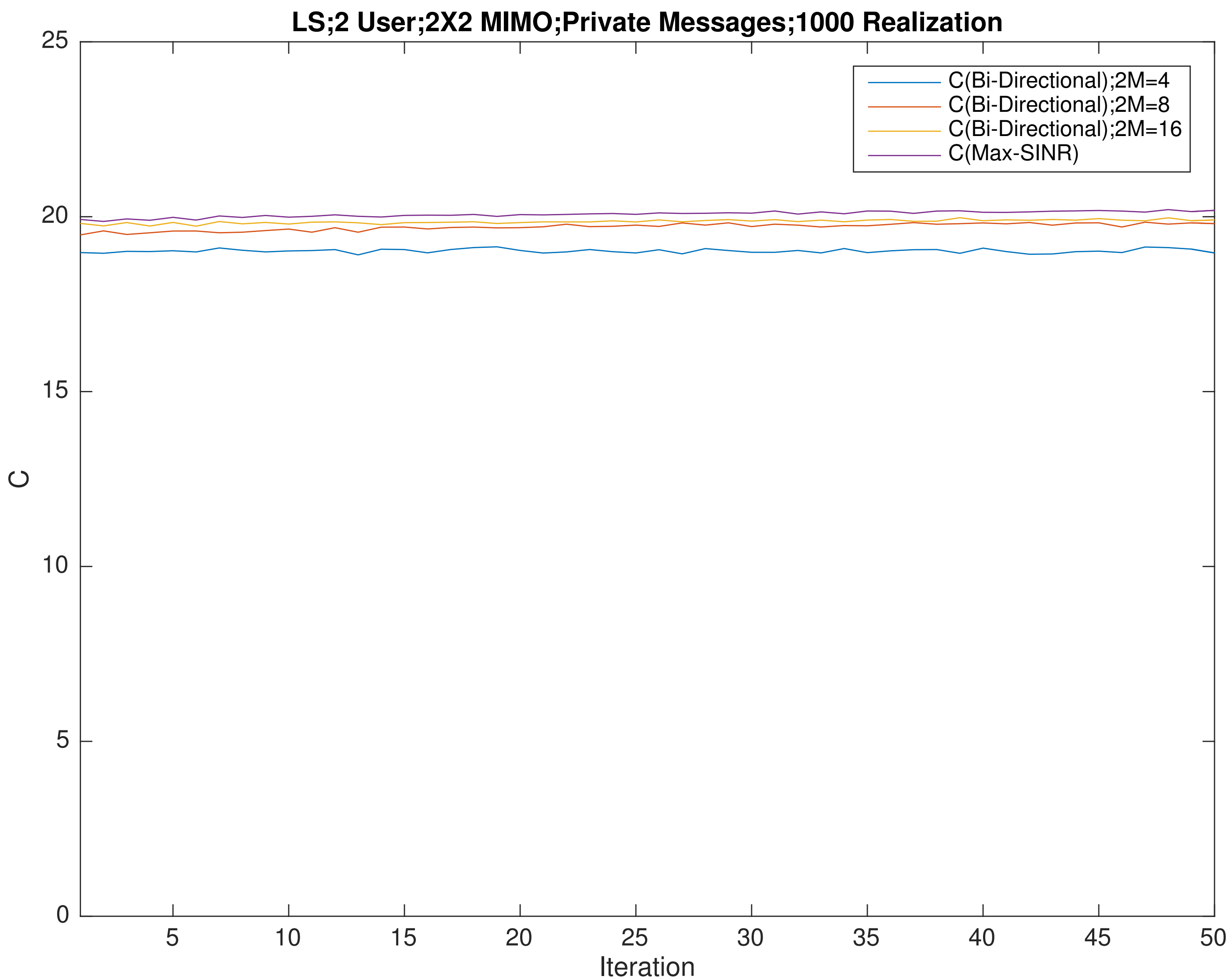

Figure 21: Insert caption

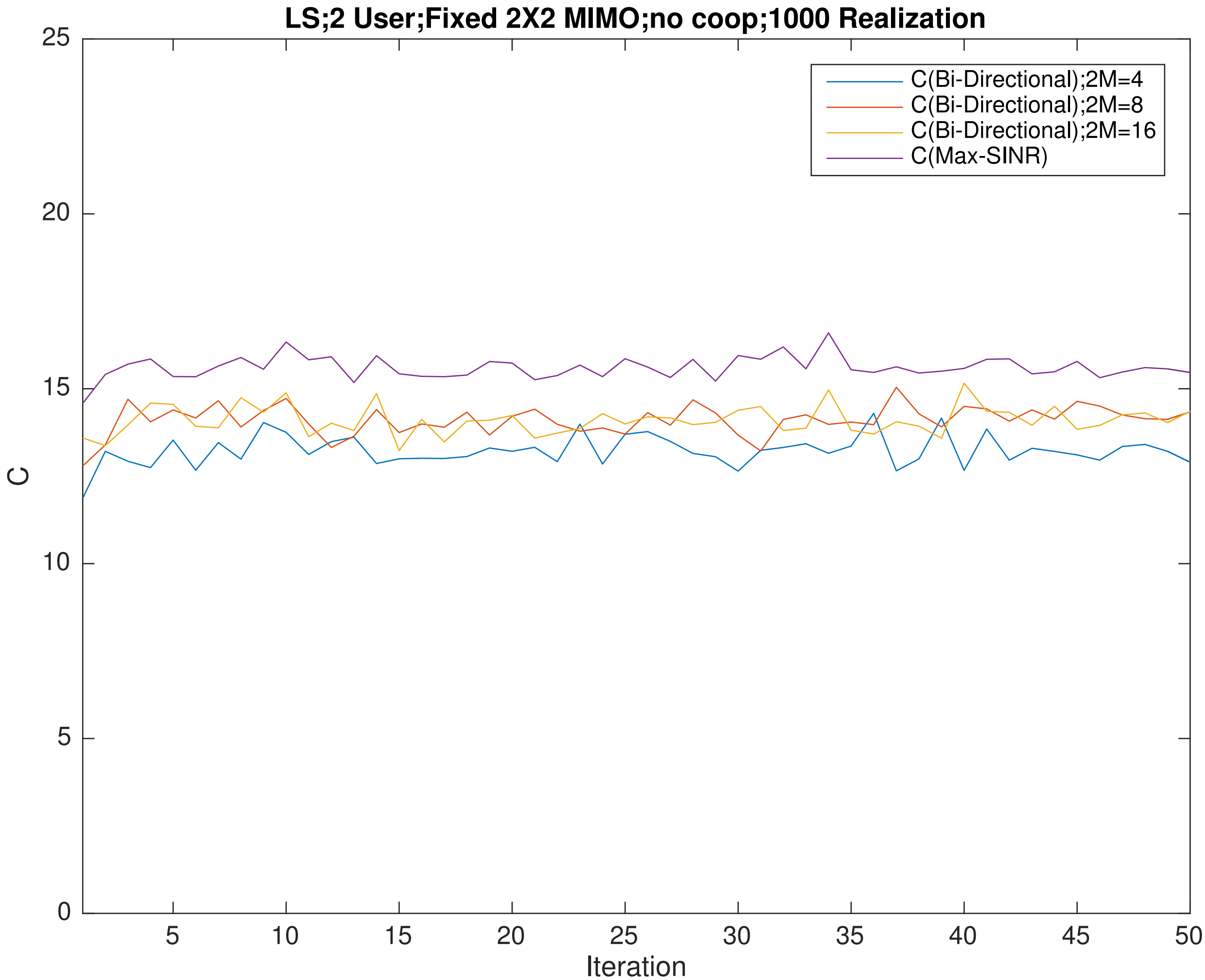

Figure 22: Insert caption

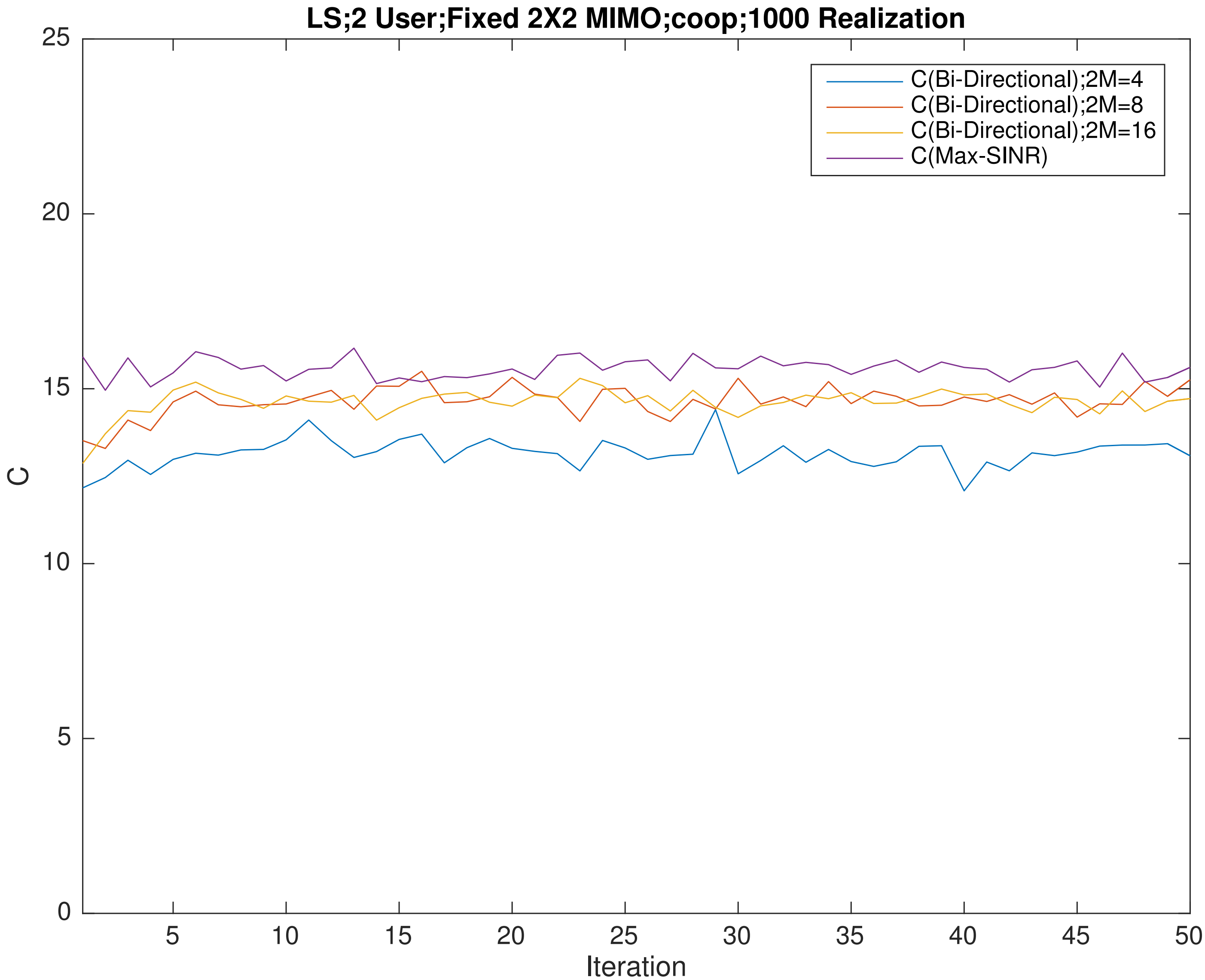

Figure 23: Insert caption