Design Trade-offs for Decentralized Baseband Processing in Massive MU-MIMO Systems

Kaipeng Li, James McNaney, Oscar Castañeda, Chance Tarver, Charles Jeon, Joseph Cavallaro, Christoph Studer

Asilomar Conference on Signals, Systems, and Computers
November 5, 2019
Massive MU-MIMO systems

- Uplink
- Downlink

BS

$B$ antennas
(\~hundreds)

$U$ users
(\~tens)
How do we handle this much data?

Possible Limitations:
- Chip I/O and interconnection bandwidth
- On-chip memory and storage
- Computing capability of modern computing fabrics

Can be on order of Tbps

BS

Channel Estimation

Detector

Decoder

uplink

B antennas
(~hundreds)

U users
(~tens)
Decentralized to resolve bottlenecks

Centralized

Decentralized

uplink transmission
downlink transmission
Decentralized to resolve bottlenecks

**Uplink Implementations**
- Consensus Methods
- Feedforward Methods:
  - MMSE:
    - Fully Decentralized
    - Partially Decentralized
  - Zero Forcing:
    - Fully Decentralized
    - Partially Decentralized

**Downlink Implementations**
- Consensus Methods
- Feedforward Methods:
  - Zero Forcing:
    - Fully Decentralized
    - Partially Decentralized
  - Weiner Filter
    - Fully Decentralized
    - Partially Decentralized

**How should we pick an implementation?**
Outline

• Overview of decentralized architectures and algorithms
• Architecture trade-offs
• Algorithm trade-offs
• Precision trade-offs
• Practical design flow
• Conclusion
Decentralized feedforward architecture

Feedforward local information *only once* instead of multiple rounds to centralized unit

- **BS unit 1**
  - *local detect*
- **BS unit 2**
  - *local detect*
- **BS unit C**
  - *local detect*

- **Centralized control unit**
  - Information gathering, computing

**Partially decentralized** (PD) architecture:
less local computation + more centralized computation

**Fully decentralized** (FD) architecture:
more local computation + less centralized computation

Data transfer we are interested in. Would be on PCIe, NVLink, InfiniBand.
Uplink linear MMSE detection

Centralized linear MMSE (C-LMMSE) detection:

\[ \hat{x} = \left( H^H H + \frac{N_0}{E_x} I \right)^{-1} H^H y \]
\[ = \left( G + \frac{N_0}{E_x} I \right)^{-1} y^{\text{MRC}} \]

*Partially* decentralization: *decentralized* matrix preprocessing + *centralized* detection

\[ \sigma^2: \text{error variance} \]
Uplink linear MMSE detection

Centralized LMMSE (C-LMMSE):

\[ G = H^H H \quad y^{MRC} = H^H y \]

\[ \hat{x} = (G + \frac{N_0}{E_x} I)^{-1} y^{MRC} \]

Partially Decentralized:

\[ G_c = H_c^H H_c \quad y_c^{MRC} = H_c^H y_c \]

\[ G = \sum_{c=1}^{C} G_c \quad y^{MRC} = \sum_{c=1}^{C} y_c^{MRC} \]

\[ \hat{x} = (G + \frac{N_0}{E_x} I)^{-1} y^{MRC} \]

PD-LMMSE obtains the same \( \hat{x} \) as C-LMMSE

Complexity: \( O(B_c U^2) + O(U^3) \) mults.

Data transfer size: \( O(U^2) \) samples / cluster
Fully decentralized (FD-) LMMSE detection

Decentralized local detection

\[ \hat{x}_c = (G_c + \frac{N_0}{E_x} I)^{-1} y_c^{\text{MRC}} \]

Fusion of local \( \hat{x}_c \) using weights, \( \lambda_c \):

\[ \hat{x} = \sum_{c=1}^{C} \lambda_c \hat{x}_c \]

Optimal \( \lambda_c \) is a function of \( \sigma_c \)

Complexity: \( O(B_c U^2) + O(U^3) \) mults.

Data transfer size: \( O(U) \) samples / cluster
**Downlink Beamforming**

- **Linear beamforming:**
  - Power constraint: \( E[\|x\|^2] \leq \rho^2 \)
  - Precoding matrix: \( P \)
  - Linear precoding: \( x = Ps \)

- **Zero-Forcing beamforming:**
  - Precoding Matrix: \( H^H(HH^H)^{-1} = H^H G^{-1} \)
  - Power constraint: \( \hat{x} = \rho \|\hat{x}\|_2 \)

- **Channel reciprocity:**
  - TDD Transmission: \( H^{dl} = (H^{ul})^H \)
Decentralized feedforward ZF beamforming

**Partially decentralized ZF beamforming:**
Set $\rho_c^2 = \rho^2 / C$

$$G_c = H_c H_c^H$$

$$G = \sum_{c=1}^C G_c$$  \hspace{1cm}  $$z = G^{-1}s$$

Broadcast $z$ to local clusters

$$\hat{x}_c = H_c^H z, \quad \hat{x}_c = \rho_c \| \hat{x}_c \|_2$$

Complexity: $O(B_c U^2) + O(U^3)$ mults.
Data transfer: $O(U^2)$ samples / cluster

**Fully decentralized ZF beamforming:**
Broadcast $s$ and set $\rho_c^2 = \rho^2 / C$

$$\hat{x}_c = H_c^H (H_c H_c^H)^{-1} s$$

$$\hat{x}_c = \rho_c \| \hat{x}_c \|_2$$

Complexity: $O(B_c U^2) + O(U^3)$ mults.
Data transfer: $O(U)$ samples / cluster
Decentralized feedforward Wiener Filter (WF) beamforming

**Partially decentralized** WF beamforming

Set $\rho_c^2 = \rho^2 / C$

$G_c = H_c H_c^H$

$G = \sum_{c=1}^{C} G_c$

$z = \frac{1}{\beta} (G + \gamma I_U)^{-1} s$

Broadcast $z$ to local BS unit

$\hat{x}_c = H_c^H z$

Complexity: $O(B_c U^2) + O(U^3) + O(\beta)$ mult.

Data transfer: $O(U^2)$ samples / cluster

**Fully decentralized** WF beamforming

Broadcast $s$ and set $\rho_c^2 = \rho^2 / C$

$P_c = \frac{1}{\beta_c} H_c^H (H_c H_c^H + \gamma I_U)^{-1} s$

$\hat{x}_c = P_c s$

Complexity: $O(B_c U^2) + O(U^3) + O(\beta)$ mult.

Data transfer: $O(U)$ samples / cluster
Architecture Trade-offs: PD vs. FD

PD-MMSE and FD-MMSE: Data transfer Depends on channel coherency

• BER: Centralized MMSE = PD-MMSE, FD-MMSE sacrifices BER

• Computation (timing) complexity: PD-MMSE = FD-MMSE

• $N_{coh}$: Period in which we update channel state information

\[
\begin{align*}
    m_{PD} &= \frac{C \times (U^2 - U + 2N_{coh}U)}{N_{coh}}, \\
    m_{FD} &= \frac{C \times 3N_{coh}U}{N_{coh}} = 3CU.
\end{align*}
\]
PD vs. FD trade-off on BER and data transfer

\[ C = 4, \ U = 16, \ B_c = 32, \ B = 128, \ 16\text{QAM} \]
Simple i.i.d. Gaussian channel and Quadriga NLOS urban campus channel

(a) BER: PD-MMSE vs. FD-MMSE
(b) Data transfer size vs. \( N_{coh} \)
Decentralized feedforward architecture

Feedforward local information *only once* instead of multiple rounds to centralized unit

<table>
<thead>
<tr>
<th></th>
<th>Partially Decentralized</th>
<th>Fully Decentralized</th>
</tr>
</thead>
<tbody>
<tr>
<td>Computation per cluster</td>
<td>Less computation</td>
<td></td>
</tr>
<tr>
<td>Internal Data Transfer to Central Node</td>
<td></td>
<td>Less data movement</td>
</tr>
<tr>
<td>BER Performance</td>
<td>Better BER</td>
<td></td>
</tr>
</tbody>
</table>
Algorithm Trade-offs: Explicit vs. Implicit method

• Example: PD-MMSE with explicit matrix inversion vs. implicit matrix inversion

• Implicit matrix inversion $A^{-1} = (G + \frac{N_0}{E_x} I)^{-1}$ for PD-MMSE
  
  • $A = LL^H$ (Cholesky decomposition, $L$ is lower triangular matrix)
  
  • Get $z$ by solving $Lz = y^{MRC}$ using forward substitution
  
  • Get $\hat{x}$ by solving $L^H \hat{x} = z$ using backward substitution

• Per-symbol complexity of explicit and implicit methods depend on $N_{coh}$

\[
 n_{ex} = \frac{(2B_c U^2 + \frac{10}{3} U^3 - \frac{1}{3} U)/N_{coh} + 4B_c U + 4U^2}{N_{coh} + 4B_c U + 4U^2}
\]

\[
 n_{im} = \frac{(2B_c U^2 + \frac{2}{3} U^3 + \frac{1}{3} U)/N_{coh} + 4B_c U + 4U^2}{N_{coh} + 4B_c U + 4U^2}
\]
Complexity of explicit vs. implicit PD-MMSE

$C=4, U=16, B_c=32, B=128$

Implicit always has lower complexity
Reusing Uplink (UL) Results for Downlink (DL)

• Channel reciprocity in TDD system: \( H^{UL} = (H^{DL})^H \)
• Gram matrix: \( G^{DL} = H^{DL}(H^{DL})^H = (H^{UL})^H H^{UL} = G^{UL} \)
• Store and reuse computed uplink results for downlink to reduce complexity
• UL MMSE detection + DL WF beamforming can only reuse \( G^{UL} \)
• UL ZF detection + DL ZF beamforming can even reuse \( (G^{UL})^{-1} \)
UL and DL integration trade-offs on BER and complexity

Example: UL PD-MMSE + DL PD-WF integration vs. UL ZF + DL ZF integration

$C=4$, $U=16$, $B_c=32$, $B=128$, 16QAM, LOS channel

MMSE and WF offer better performance

(a) BER: PD-MMSE detection vs. PD-ZF detection  (b) BER: PD-WF precoding vs. PD-ZF precoding
UL and DL integration trade-offs on BER and complexity

Example: UL $PD-MMSE + DL PD-WF$ integration vs. $UL ZF + DL ZF$ integration

$C=4$, $U=16$, $B_c=32$, $B=128$, 16QAM, LOS channel

By integrating, ZF only requires 65% of the multiplies
Precision Trade-offs: 32-bit vs. 8bit floating point

Example: PD-MMSE and FD-MMSE
C=4, U=16, B_{c}=32, B=128, 16QAM, Quadriga NLOS urban campus channel

8-bit floating point reduces 4x data transfer size compared to 32-bit at only little loss of BER
Summary of Tradeoffs
Conclusion

• Decentralized baseband processing resolves complexity and interconnection bandwidth bottlenecks for massive MU-MIMO

• Practical massive MU-MIMO should leverage design trade-offs at different aspects:
  • Architecture trade-offs of PD and FD on BER vs. data transfer size
    • *Unless you expect very low coherence time, choose partially decentralized.*
  • Algorithms trade-offs of explicit and implicit methods on BER vs. complexity
    • *Use implicit matrix inversions whenever possible. Reuse results from uplink to downlink.*
  • Precision trade-offs of various data precision options on BER vs. efficiency
    • *Use fp16 or even fp8 unless BER is serious concern.*