**Finite Alphabet Beamforming Codebooks in MIMO-OFDM Systems**

*Melissa Duarte and Ashutosh Sabharwal*  
(mdarte, ashu)@rice.edu

---

**Goal**

Efficient implementation of a real-time beamforming MIMO-OFDM system.

**Motivation**

- OFDM
  - Convert wideband channel into parallel narrowband channels
  - Simplified receiver complexity
  - Allows use of narrowband techniques per subcarrier
- MIMO
  - More throughput
  - Better reliability
  - In every standard (WiMax, WiFi, 3GPP)

- Few feedback bits are useful
  - Example: 3x3 MIMO, 2 MIMO OFDM
- Beamforming
  - Better reliability
  - More throughput
  - Convert wideband channel into parallel narrowband
  - Power

**Contributions**

- Design of a low complexity Quantizer using Finite Alphabet Codebooks
- Construction of Finite Alphabet Codebooks
  - For low complexity Quantizer
  - With good performance
- Mixed Codebook scheme: A WiMax compliant scheme using Finite Alphabet Codebooks to simplify the implementation of the Quantizer

**Future work**

Integrate beamforming into Rice Wireless Open Access Research Platform (WARP) and perform experiments over real wireless channels.

---

**Beamforming MIMO OFDM**

- Beamform each subcarrier
- Example: 3x3 MIMO, 2 subcarriers

**Quantizer Complexity**

- Pipeline processing \( b_k = \arg \max |H_k w_k|^2 \)
- Complexity of Q and Qn blocks
  - WiMax example: T=R=4 and N=64
    - Q block: 4096 Multipliers and 3584 Adders
    - Qn block: 512 Multipliers and 448 Adders
- Quantizer not appropriate for mobile station
  - High complexity
  - Tight timing constraints in FDD mode
  - Cost
  - Power

**New approach**

- Use a Finite Alphabet Codebook (FAC)
- Each beamforming vector in the FAC can be decomposed as
  \( w_j = c_j / ||c_j|| \)
- Entries of \( c_j \) belong to finite set \( \mu \)
- Set \( \mu \) is the finite codebook alphabet
- The codebook alphabet \( \mu \) determines the performance of the codebook and the complexity of the Quantizer

**Quantizer simplification using Finite Alphabet Codebooks (FAC)**

- Consider \( \mu = \{1, -1, j, -j\} \)
- Quantizer processing simplifies
  - Pipeline processing \( b_k = \arg \max |H_k w_k|^2 \)
  - Multiplications at Q block can be implemented without any multiplier because entries of \( c_j \) belong to \( \mu \)

**Vector Mapping to Construct Finite Alphabet Codebooks (FAC)**

- Map any codebook \( W \) into a FAC \( W_{FAC} \) with codebook alphabet \( \mu \)
  - Map each beamforming vector \( w_j \) into \( W_{FAC j} \)
  \( c_j = \arg \max < m, w_j > \)
  \( m \) : Set of T-dim vectors with entries \( \in \mu \)

**BER Results**

- The FAC was obtained by mapping the USTC codebook
- The alphabet of the FAC is \( \mu = \{1, -1, j, -j\} \)
- Results are for a single subcarrier (per subcarrier)

**Mixed Codebook (MxC) Scheme for WiMax**

- Mixed Codebook (MxC) Scheme: Use WiMax codebook at Beamforming and MRC blocks and a FAC version of the WiMax codebook at the Quantizer
- WiMax codebook at Beamforming and MRC blocks
  - Makes MxC scheme WiMax compliant
- FAC version of the WiMax codebook at the Quantizer
  - Significantly simplifies the implementation of the Quantizer
- FAC version of the WiMax codebook is obtained using the proposed vector mapping

**Resource Estimate for WiMax**

- \( T=\mu=4, N=64, 1536 \) data subcarriers (2048FFT), \( T_{OFDM}=100.8 \mu s \)
- \( \mu \) : Duration of an OFDM symbol

<table>
<thead>
<tr>
<th>Quantizer</th>
<th>Multipliers</th>
<th>Adders</th>
</tr>
</thead>
<tbody>
<tr>
<td>WiMax</td>
<td>549</td>
<td>0</td>
</tr>
<tr>
<td>FAC</td>
<td>61</td>
<td>244</td>
</tr>
</tbody>
</table>

- Time constraint: Quantize all channels (1536) in a time equal to \( T_{OFDM} \)
- \( f_{FPGA}=128 \) MHz

- FAC1 has alphabet \( \mu = \{1, -1, j, -j\} \)
- FAC2 has alphabet \( \mu = \{1, -1, j, -j, 1j, -1j, -1j, 1j\} \)
- Results are for a single subcarrier (per subcarrier)