Precoding Matrix Design In Linear Video Coding

Shuo Zheng\textsuperscript{1}, Marco Cagnazzo\textsuperscript{1}, Michel Kieffer\textsuperscript{2}

\textsuperscript{1}LTCI, Télécom ParisTech, Univ Paris-Saclay
\textsuperscript{2}L2S, CNRS–CentraleSupelec–Univ Paris-Sud, Univ Paris-Saclay

2018, April, 17.
IEEE International Conference on Acoustics, Speech and Signal Processing, Calgary, Alberta, Canada.
Outline

1. Linear Video Coding Scheme: SoftCast
2. Optimal precoding matrix design under per-subchannel power constraint
3. Lower complexity design approaches
4. Simulations
5. Conclusions
6. Appendix
SoftCast

- Joint source-channel video coder
- Linear source coder and channel precoder
  - 3D-DCT on Group of Pictures (GoP) of video
  - Linear scaling for each DCT-coefficients for error protection
  - No motion compensation, no entropy coding, no quantization
- Analog-like modulation
- LMMSE decoder

---

SoftCast

- Joint source-channel video coder
- Linear source coder and channel precoder
  - 3D-DCT on Group of Pictures (GoP) of video
  - Linear scaling for each DCT-coefficients for error protection
  - No motion compensation, no entropy coding, no quantization

- Analog-like modulation
- LMMSE decoder

SoftCast ¹

- Joint source-channel video coder
- Linear source coder and channel precoder
  - 3D-DCT on Group of Pictures (GoP) of video
  - Linear scaling for each DCT-coefficients for error protection
  - No motion compensation, no entropy coding, no quantization
- Analog-like modulation
- LMMSE decoder

SoftCast

- Joint source-channel video coder
- Linear source coder and channel precoder
  - 3D-DCT on Group of Pictures (GoP) of video
  - Linear scaling for each DCT-coefficients for error protection
  - No motion compensation, no entropy coding, no quantization
- Analog-like modulation
- LMMSE decoder

---

Performance of SoftCast

Figure: Performance of SoftCast vs. single-layer MPEG4

Video quality commensurates with channel quality

---

SoftCast Transmitter\(^3\)

**Figure:** Softcast coding and power allocation

- **Chunk:** \(n_r \times n_c\) spatial DCT coefficients assumed uncorrelated.
- Low-variance chunks discarded when bandwidth limited.
- Power allocation: each chunk is scaled to meet transmission power constraint.

SoftCast Transmitter

Figure: Softcast coding and power allocation

- Chunk: $n_r \times n_c$ spatial DCT coefficients assumed uncorrelated.
- Low-variance chunks discarded when bandwidth limited.
- Power allocation: each chunk is scaled to meet transmission power constraint.

---

## SoftCast Transmitter

- **Chunk**: \( n_r \times n_c \) spatial DCT coefficients assumed uncorrelated.
- **Low-variance chunks discarded when bandwidth limited.**
- **Power allocation**: each chunk is scaled to meet transmission power constraint.

---

Total power constraint

Power allocation assumes in most of the cases under total power constraint $P_T$.

**Figure:** AWGN channel and total power constraint

$G$: Precoding matrix

$H$: LMMSE decoder
Per-subchannel power constraints

Per-subchannel power constraints in

- DSL or powerline telecommunication (PLT) channels\(^4\)
- Multi-antenna transmission

Per-subchannel power constraints

Here: power allocation with per-subchannel power constraints.

Figure: AWGN channel and per-subchannel constraint
Chunk vector

**Figure:** First components in each chunk form first chunk vector.
Chunk vector

Figure: Last components in each chunk form last chunk vector.
Optimal precoding matrix design

Transmitted vector

- Sequence of \( n_r \times n_c \) chunk vectors \( t \), with

\[
E(tt^T) = \Lambda = \text{diag}(\lambda_1 \ldots \lambda_{n_{c_k}}),
\]

\( \lambda_i \) in decreasing order.

- Transmitted vector: \( x = Gt \). Precoding matrix: \( G \in \mathbb{R}^{n_{SC} \times n_{C_k}} \).

- Per-subchannel power constraint:

\[
E(xx^T)_{i,i} = (G\Lambda G^T)_{i,i} \leq p_i, \ i = 1, \ldots, n_{SC}
\]

---

\(^5\)After chunk selection, we assume \( n_{C_k} = n_{SC} \) and over each subchannel one chunk is transmitted during one GoP.
Transmission Model

Received vector

\[ y = Gt + v \]  

where \( v \sim \mathcal{N}(0, N) \).

Linear decoding: \( \hat{t} = Hy \).

Mean square reconstruction error (MSE)

\[ \varepsilon = E \left[ (t - \hat{t})^T (t - \hat{t}) \right] \]
Compute the optimal precoding matrix $G$

- Decoding matrix minimizing $\varepsilon$

$$\bar{H} = \Lambda G^T \left( G\Lambda G^T + N \right)^{-1}.$$  \hfill (3)

- With $\bar{H}$, $\varepsilon$ becomes

$$\varepsilon = \text{tr} \left( \left( I + \left( G\Lambda^{\frac{1}{2}} \right)^T N^{-1} \left( G\Lambda^{\frac{1}{2}} \right) \right)^{-1} \Lambda \right),$$  \hfill (4)

and to be minimized under

$$\left( G\Lambda G^T \right)_{i,i} = p_i, \ i = 1, \ldots, n_{SC}$$  \hfill (5)
First, compute an optimal precoding matrix $\tilde{G}$ under total power constraint.

Figure: Total power constraint

---

Optimal precoding matrix $G$

Then compute an orthogonal transform matrix $Z$ to match per-subchannel power constraints.

$Z$

$\tilde{G}$

$P_T$

$G = Z \tilde{G}$

Therefore $G = Z \tilde{G}$. 
How to compute $Z$

Figure: $G = Z \tilde{G}$
Optimal precoding matrix design

Conditions for the existence of $Z$

- Let $m = E \left( \text{diag} \left( \tilde{G} t t^T \tilde{G}^T \right) \right) = \text{diag} \left( \tilde{G} \Lambda \tilde{G}^T \right)$ with $m_1 \geq \cdots \geq m_{n_{\text{SC}}}$, and $p = (p_1, \ldots, p_{n_{\text{SC}}})$ with $p_1 \geq \cdots \geq p_{n_{\text{SC}}}$.
- If
  \[ \sum_{i=1}^{k} p_i \leq \sum_{i=1}^{k} m_i \]
  for all $k = 1, 2, \ldots, n_{\text{SC}}$, then an orthonormal matrix $Z$ such that
  \[ \text{diag} \left( Z M Z^T \right) = p \]
  can be found\(^7\).
- What if the conditions do not hold?

Conditions for the existence of $Z$

- Let $m = E \left( \text{diag} \left( \tilde{G} t t^T \tilde{G}^T \right) \right) = \text{diag} \left( \tilde{G} \Lambda \tilde{G}^T \right)$ with $m_1 \geq \cdots \geq m_{n_{SC}}$, and $p = (p_1, \ldots, p_{n_{SC}})$ with $p_1 \geq \cdots \geq p_{n_{SC}}$.
- If

$$\sum_{i=1}^{k} p_i \leq \sum_{i=1}^{k} m_i$$

for all $k = 1, 2, \ldots, n_{SC}$, then an orthonormal matrix $Z$ such that

$$\text{diag} \left( Z M Z^T \right) = p$$

can be found\(^7\).

- What if the conditions do not hold?

---

Conditions for the existence of $Z$

Let $m = E \left( \text{diag} \left( \tilde{G}t t^T \tilde{G}^T \right) \right) = \text{diag} \left( \tilde{G} \Lambda \tilde{G}^T \right)$ with $m_1 \geq \cdots \geq m_{n_{SC}}$, and $p = (p_1, \ldots, p_{n_{SC}})$ with $p_1 \geq \cdots \geq p_{n_{SC}}$.

If

$$\sum_{i=1}^{k} p_i \leq \sum_{i=1}^{k} m_i \quad (6)$$

for all $k = 1, 2, \ldots, n_{SC}$, then an orthonormal matrix $Z$ such that

$$\text{diag} \left( ZMZ^T \right) = p$$

can be found$^7$.

What if the conditions do not hold?

---

Multi-Level Water-Filling

At first, compute a $\tilde{G}$ under total power constraint $\sum_{i=1}^{n_{SC}} p_i$.

\[
t: \begin{bmatrix} \lambda_1 & \lambda_2 & \lambda_3 & \cdots & \lambda_{n_{SC}} \\
1 & 1 & 1 & \cdots & 1 \\
p_1 & p_2 & p_3 & \cdots & p_{n_{SC}} \end{bmatrix}
\]

\[
m = \text{diag}(\tilde{G} \Lambda \tilde{G}^T) \sum_{i=1}^{n_{SC}} m_i = \sum_{i=1}^{n_{SC}} p_i
\]
Then, check the conditions of existence of $Z$.

\[
m = \text{diag}(\hat{G} \Lambda \hat{G}^T) \sum_{i=1}^{n_{SC}} m_i = \sum_{i=1}^{n_{SC}} p_i
\]
Check the conditions of existence of $Z$.

$$\sum_{i=1}^{n_{SC}} m_i \geq \sum_{i=1}^{n_{SC}} p_i$$
Multi-Level Water-Filling

- Check the conditions of existence of $Z$.

$$t: \begin{bmatrix} \lambda_1 & \lambda_2 & \lambda_3 & \cdots & \lambda_{n_{SC}} \end{bmatrix}$$

$$m = \text{diag}(\tilde{G} \Lambda \tilde{G}^T) \sum_{i=1}^{n_{SC}} m_i = \sum_{i=1}^{n_{SC}} p_i$$

$$\begin{bmatrix} m_1 & m_2 & m_3 & \cdots & m_{\tau_0} & m_{n_{SC}} \end{bmatrix}$$

$$\begin{bmatrix} p_1 & p_2 & p_3 & \cdots & p_{\tau_0} & p_{n_{SC}} \end{bmatrix}$$

$$\sum_{i=1}^{\tau_0} m_i < \sum_{i=1}^{\tau_0} p_i$$

- When conditions are not satisfied, recompute $\tilde{G}$ for components from 1 to $\tau_0$ of $t$ under total power constraint $\sum_{i=1}^{\tau_0} p_i$. 

S.Zheng, M.Cagnazzo, M.Kieffer
Multi-Level Water-Filling

- Check the conditions of existence of $Z$.

\[
m = \text{diag}(\widetilde{G}\Lambda\widetilde{G}^T) \quad \sum_{i=1}^{n_{\text{SC}}} m_i = \sum_{i=1}^{n_{\text{SC}}} p_i
\]

- When conditions are not satisfied, recompute $\widetilde{G}$ for components from 1 to $\tau_0$ of $t$ under total power constraint $\sum_{i=1}^{\tau_0} p_i$. 
Multi-Level Water-Filling

Compute a $\tilde{G}$ for components from 1 to $\tau_0$ of $t$.

$t$: $\lambda_1 \lambda_2 \lambda_3 \cdots \lambda_{\tau_0}$

$m = \text{diag}(\tilde{G} \Lambda \tilde{G}^T) \sum_{i=1}^{\tau_0} m_i = \sum_{i=1}^{\tau_0} p_i$

$m_{\tau_0}$

$p_{\tau_0}$

$\sum_{i=1}^{\tau_0} m_i = \sum_{i=1}^{\tau_0} p_i$
Multi-Level Water-Filling

- Check the conditions of existence of $Z$.

$$m = \text{diag}(\tilde{G} \Lambda \tilde{G}^T) \sum_{i=1}^{\tau} m_i = \sum_{i=1}^{\tau} p_i$$

- When conditions are not satisfied, recompute $\tilde{G}$ for components from 1 to $\tau_1$ of $t$ under total power constraint $\sum_{i=1}^{\tau_1} p_i$. 
Multi-Level Water-Filling

- Check the conditions of existence of $Z$.

$$t: \begin{bmatrix} \lambda_1 & \lambda_2 & \lambda_3 & \cdots & \cdots & \cdots & \lambda_{\tau_0} \end{bmatrix}$$

$$m = \text{diag}\left(\tilde{G} \Lambda \tilde{G}^T\right) \sum_{i=1}^{\tau} m_i = \sum_{i=1}^{\tau} p_i$$

$$\begin{bmatrix} m_1 & m_2 & m_3 & \cdots & \cdots & m_{\tau_1} & m_{\tau_0} \end{bmatrix}$$

$$\begin{bmatrix} p_1 & p_2 & p_3 & \cdots & \cdots & p_{\tau_1} & p_{\tau_0} \end{bmatrix}$$

$$\sum_{i=1}^{\tau_1} m_i < \sum_{i=1}^{\tau_1} p_i$$

- When conditions are not satisfied, recompute $\tilde{G}$ for components from 1 to $\tau_1$ of $t$ under total power constraint $\sum_{i=1}^{\tau_1} p_i$. 
Compute a $\tilde{G}$ for components from 1 to $\tau_1$ of $t$ and check the conditions of existence of $Z$.

$$m = \text{diag}(\tilde{G} \Lambda \tilde{G}^T)$$

$$\sum_{i=1}^{\tau_1} m_i = \sum_{i=1}^{\tau_1} p_i$$

When conditions are not satisfied, recompute $\tilde{G}$ for components from 1 to $\tau_2$ of $t$ under total power constraint $\sum_{i=1}^{\tau_2} p_i$ and continue.
Multi-Level Water-Filling

- Compute a $\tilde{G}$ for components from 1 to $\tau_1$ of $t$ and check the conditions of existence of $Z$.

$$m = \text{diag}(\tilde{G} \Lambda \tilde{G}^T) \sum_{i=1}^{\tau_1} m_i = \sum_{i=1}^{\tau_1} p_i$$

$$\sum_{i=1}^{\tau_2} m_i < \sum_{i=1}^{\tau_2} p_i$$

- When conditions are not satisfied, recompute $\tilde{G}$ for components from 1 to $\tau_2$ of $t$ under total power constraint $\sum_{i=1}^{\tau_2} p_i$ and continue.
At the end, conditions of existence of $Z$ are satisfied for $\lambda_1, \lambda_2, \lambda_3$ and $p_1, p_2, p_3$. 

$$m = \text{diag}(\tilde{G} \Lambda \tilde{G}^T) \quad \sum_{i=1}^{3} m_i = \sum_{i=1}^{3} p_i$$

$$\sum_{i=1}^{j} m_i \geq \sum_{i=1}^{j} p_i, \ 1 \leq j \leq 3$$
Multi-Level Water-Filling

First subblock of $Z \tilde{G}$ has been found

$$(Z_1 \tilde{G}_1) \begin{bmatrix} \lambda_1 & \lambda_2 & \lambda_3 \end{bmatrix} (Z_1 \tilde{G}_1)^T = \begin{bmatrix} p_1 & * & * \\ * & p_2 & * \\ * & * & p_3 \end{bmatrix}$$

Figure: First subblock of $Z \tilde{G}$ and associated water-level under total power constraint $\sum_{i=1}^{3} p_i$. 
Multi-Level Water-Filling

Second subblock of $Z \tilde{G}$

$$
(Z_2 \tilde{G}_2) \begin{bmatrix} 
\lambda_4 \\
\vdots \\
\lambda_k
\end{bmatrix} (Z_2 \tilde{G}_2)^T = \begin{bmatrix} 
p_4 & * & * \\
* & \ddots & * \\
* & * & p_k
\end{bmatrix}
$$

Figure: Second subblock of $Z \tilde{G}$ and associated water-level under total power constraint $\sum_{i=4}^{k} p_i$. 
Multi-Level Water-Filling compute $Z \tilde{G}$

Last subblock of $Z \tilde{G}$

\[
\begin{bmatrix}
\lambda_{k+1} \\
\vdots \\
\lambda_{n_{SC}}
\end{bmatrix}
\begin{bmatrix}
(Z_3 \tilde{G}_3) \\
(Z_2 \tilde{G}_2) \\
(Z_1 \tilde{G}_1)
\end{bmatrix}^T =
\begin{bmatrix}
p_{k+1} \\
\vdots \\
p_{n_{SC}}
\end{bmatrix}
\]

Figure: Last subblock of $Z \tilde{G}$ and associated water-level under total power constraint $\sum_{i=k+1}^{n_{SC}} p_i$. 

S. Zheng, M. Cagnazzo, M. Kieffer

Precoding Matrix Design In Linear Video Coding
Result

- Structure of optimal precoding matrix

\[
\begin{align*}
G &= \begin{pmatrix}
G_1 & 0 & \cdots & 0 & 0 \\
0 & G_2 & 0 & \cdots & 0 \\
0 & 0 & \ddots & \ddots & \vdots \\
\vdots & \vdots & \ddots & G_{s-1} & 0 \\
0 & 0 & \cdots & 0 & G_s
\end{pmatrix} = \begin{pmatrix}
Z_1 \tilde{G}_1 & 0 & \cdots & 0 & 0 \\
0 & Z_2 \tilde{G}_2 & 0 & \cdots & 0 \\
0 & 0 & \ddots & \ddots & \vdots \\
\vdots & \vdots & \ddots & Z_{s-1} \tilde{G}_{s-1} & 0 \\
0 & 0 & \cdots & 0 & Z_s \tilde{G}_s
\end{pmatrix}
\end{align*}
\]

- Worst-case complexity to find the first subblock: \( \sum_{k=n_{SC}}^1 k = O\left(n_{SC}^2\right) \).
- Worst-case total complexity: \( \sum_{i=n_{SC}}^1 n_i^2 = O\left(n_{SC}^3\right) \).
Result

- Structure of optimal precoding matrix

\[
G = \begin{pmatrix}
G_1 & 0 & \cdots & 0 & 0 \\
0 & G_2 & 0 & \cdots & 0 \\
0 & 0 & \ddots & \ddots & \vdots \\
\vdots & \vdots & \ddots & G_{s-1} & 0 \\
0 & 0 & \cdots & 0 & G_s
\end{pmatrix} = \begin{pmatrix}
Z_1 \tilde{G}_1 & 0 & \cdots & 0 & 0 \\
0 & Z_2 \tilde{G}_2 & 0 & \cdots & 0 \\
0 & 0 & \ddots & \ddots & \vdots \\
\vdots & \vdots & \ddots & Z_{s-1} \tilde{G}_{s-1} & 0 \\
0 & 0 & \cdots & 0 & Z_s \tilde{G}_s
\end{pmatrix}
\]

- Worst-case complexity to find the first subblock: \( \sum_{k=n_{SC}}^{1} k = O\left(n_{SC}^2\right) \).
- Worst-case total complexity: \( \sum_{i=n_{SC}}^{1} n_{i}^2 = O\left(n_{SC}^3\right) \).
Result

- Structure of optimal precoding matrix

\[
G = \begin{pmatrix}
G_1 & 0 & \cdots & 0 & 0 \\
0 & G_2 & 0 & \cdots & 0 \\
0 & 0 & \ddots & \ddots & \vdots \\
\vdots & \vdots & \ddots & G_{s-1} & 0 \\
0 & 0 & \cdots & 0 & G_s
\end{pmatrix} = \begin{pmatrix}
Z_1 \hat{G}_1 & 0 & \cdots & 0 & 0 \\
0 & Z_2 \hat{G}_2 & 0 & \cdots & 0 \\
0 & 0 & \ddots & \ddots & \vdots \\
\vdots & \vdots & \ddots & Z_{s-1} \hat{G}_{s-1} & 0 \\
0 & 0 & \cdots & 0 & Z_s \hat{G}_s
\end{pmatrix}
\]

- Worst-case complexity to find the first subblock: \(\sum_{k=n_{SC}}^{1} k = O(n_{SC}^2)\).
- Worst-case total complexity: \(\sum_{i=n_{SC}}^{1} n_i^2 = O(n_{SC}^3)\).
Lower complexity design approaches

- Power Allocation with Inferred Split Position (PAISP)
  - Infer split position by dichotomy
  - Less iterations required to recompute $\tilde{G}$ compared to optimal approach

- Power Allocation with Local Power Adjustement (PALPA)
  - Try to adjust allocated power when conditions for $Z$ are not satisfied
  - Again, less iterations required
Lower complexity designs

Lower complexity design approaches

- Power Allocation with Inferred Split Position (PAISP)
  - Infer split position by dichotomy
  - Less iterations required to recompute $\tilde{G}$ compared to optimal approach

- Power Allocation with Local Power Adjustment (PALPA)
  - Try to adjust allocated power when conditions for $Z$ are not satisfied
  - Again, less iterations required
Power line channel

- Power line channel considered for transmission\(^8\). Unit noise variance assumed on each channel.

**Figure**: SNR as a function of the subchannel index for the considered PLT channel

\(^8\)ETSI, “Powerline telecommunications (PLT); powerline HDMI analysis for very short range link HD and UHD applications,” ETSI, Technical Report 103 343 V1.1.1, December 2015.
Reference: Simple Chunk Scaling (SCS)

Figure: Simple chunk scaling
Simulation results

(a) Kimonol (Class B)

(b) RaceHorses (Class C)

Figure: PSNRs for optimal, PAISP, and SCS precoding matrix design techniques and the associated decoding matrix.
## Simulation results

<table>
<thead>
<tr>
<th>Cl.</th>
<th>Name</th>
<th>PSNR (dB)</th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>SCS</td>
<td>Opt. Alloc/PAISP</td>
<td>PALPA</td>
<td>Gain</td>
</tr>
<tr>
<td>B</td>
<td>Kimonol</td>
<td>42.79</td>
<td>47.57</td>
<td>47.56</td>
<td>4.78</td>
</tr>
<tr>
<td></td>
<td>BasketballDrive</td>
<td>38.83</td>
<td>39.54</td>
<td>39.53</td>
<td>0.71</td>
</tr>
<tr>
<td></td>
<td>BQ Terrace</td>
<td>34.83</td>
<td>35.86</td>
<td>35.85</td>
<td>1.03</td>
</tr>
<tr>
<td></td>
<td>Cactus</td>
<td>36.53</td>
<td>38.47</td>
<td>38.44</td>
<td>1.94</td>
</tr>
<tr>
<td></td>
<td>ParkScene</td>
<td>41.83</td>
<td>44.06</td>
<td>44.03</td>
<td>2.23</td>
</tr>
<tr>
<td>C</td>
<td>PartyScene</td>
<td>40.89</td>
<td>42.94</td>
<td>42.94</td>
<td>2.05</td>
</tr>
<tr>
<td></td>
<td>BQMall</td>
<td>41.24</td>
<td>44.91</td>
<td>44.91</td>
<td>3.67</td>
</tr>
<tr>
<td></td>
<td>BasketballDrill</td>
<td>44.96</td>
<td>47.32</td>
<td>47.31</td>
<td>2.36</td>
</tr>
<tr>
<td></td>
<td>RaceHorses</td>
<td>42.81</td>
<td>46.21</td>
<td>46.21</td>
<td>3.4</td>
</tr>
</tbody>
</table>
## Simulation results

<table>
<thead>
<tr>
<th>Cl.</th>
<th>Name</th>
<th>PSNR (dB)</th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>SCS</td>
<td>Opt. Alloc/PAISP</td>
<td>PALPA</td>
<td>Gain</td>
</tr>
<tr>
<td>D</td>
<td>BQSquare</td>
<td>39.38</td>
<td>44.55</td>
<td>44.55</td>
<td>5.17</td>
</tr>
<tr>
<td></td>
<td>RaceHorses</td>
<td>43.89</td>
<td>49.03</td>
<td>49.03</td>
<td>5.14</td>
</tr>
<tr>
<td></td>
<td>BlowingBubbles</td>
<td>42.26</td>
<td>47.90</td>
<td>47.90</td>
<td>5.64</td>
</tr>
<tr>
<td></td>
<td>BasketballPass</td>
<td>45.03</td>
<td>49.55</td>
<td>49.55</td>
<td>4.52</td>
</tr>
<tr>
<td>E</td>
<td>FourPeople</td>
<td>40.74</td>
<td>47.13</td>
<td>47.13</td>
<td>6.39</td>
</tr>
<tr>
<td></td>
<td>Jonny</td>
<td>40.56</td>
<td>48.43</td>
<td>48.43</td>
<td>7.87</td>
</tr>
<tr>
<td></td>
<td>KristenAndSara</td>
<td>39.77</td>
<td>46.95</td>
<td>46.95</td>
<td>7.18</td>
</tr>
<tr>
<td>F</td>
<td>SlideShow</td>
<td>35.28</td>
<td>46.83</td>
<td>46.80</td>
<td>11.55</td>
</tr>
</tbody>
</table>
# Complexity comparison

<table>
<thead>
<tr>
<th>Class</th>
<th>Name</th>
<th>Speed-Up</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>PAISP</td>
</tr>
<tr>
<td>B</td>
<td>Kimonol</td>
<td>8.6</td>
</tr>
<tr>
<td></td>
<td>BasketballDrive</td>
<td>6</td>
</tr>
<tr>
<td></td>
<td>BQ Terrace</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td>Cactus</td>
<td>3.7</td>
</tr>
<tr>
<td></td>
<td>ParkScene</td>
<td>6</td>
</tr>
<tr>
<td>C</td>
<td>PartyScene</td>
<td>3.3</td>
</tr>
<tr>
<td></td>
<td>BasketballDrill</td>
<td>1.0</td>
</tr>
<tr>
<td></td>
<td>BQMall</td>
<td>4.5</td>
</tr>
<tr>
<td></td>
<td>RaceHorses</td>
<td>1.0</td>
</tr>
<tr>
<td>E</td>
<td>FourPeople</td>
<td>3.5</td>
</tr>
<tr>
<td></td>
<td>Jonny</td>
<td>5</td>
</tr>
<tr>
<td></td>
<td>KristenAndSara</td>
<td>5</td>
</tr>
<tr>
<td>F</td>
<td>SlideShow</td>
<td>1.0</td>
</tr>
</tbody>
</table>
Conclusions

- Lower complexity precoding matrix design for SoftCast Scheme under per-channel power constraint
- Average speed up factor of PALPA and PAISP are respectively 13 and 6
- Lower complexity approaches have negligible performance loss compared to optimal precoding matrix design
Dichotomy approach

As in optimal method, at first, we compute \( \tilde{G} \) under total power constraint and check the condition of existence of \( Z \).

\[
\begin{align*}
    t: & \quad \lambda_1 \quad \lambda_2 \quad \lambda_3 \\
    m & = \text{diag} \left( \tilde{G} \Lambda \tilde{G}^T \right) \sum_{i=1}^{n_{SC}} m_i = \sum_{i=1}^{n_{SC}} p_i
\end{align*}
\]

\[
\begin{align*}
    m_1 & \quad m_2 \quad m_3 \\
    p_1 & \quad p_2 \quad p_3
\end{align*}
\]

\[
\sum_{i=1}^{\tau} m_i < \sum_{i=1}^{\tau} p_i
\]
Dichotomy approach

In the next step, we compute a $\tilde{G}$ for components from 1 to $\frac{\tau+1}{2}$ of $t$ under total power constraint $\sum_{i=1}^{\frac{\tau+1}{2}} p_i$.

$$t: \begin{bmatrix} \lambda_1 & \lambda_2 & \lambda_3 & \cdots & \lambda_{\frac{\tau+1}{2}} & \lambda_{\frac{\tau+1}{2}+1} & \cdots & \lambda_{n_{SC}} \end{bmatrix}$$

$$m = \text{diag}(\tilde{G}\Lambda\tilde{G}^T) \sum_{i=1}^{n_{SC}} m_i = \sum_{i=1}^{n_{SC}} p_i$$

$$m_1 m_2 m_3 \cdots m_{\frac{\tau+1}{2}} m_{\frac{\tau+1}{2}+1} \cdots m_{n_{SC}}$$

$$p_1 p_2 p_3 \cdots p_{\frac{\tau+1}{2}} p_{\frac{\tau+1}{2}+1} \cdots p_{n_{SC}}$$

$$\sum_{i=1}^{\tau} m_i < \sum_{i=1}^{\tau} p_i$$
Dichotomy approach

We compute a $\tilde{G}$ for components from 1 to $\frac{\tau+1}{2}$ of $t$ under total power constraint $\sum_{i=1}^{\frac{\tau+1}{2}} p_i$ and check the condition of existence of $Z$.

$$t: \begin{bmatrix} \lambda_1 & \lambda_2 & \lambda_3 & \cdots & \lambda_{\frac{\tau+1}{2}} \end{bmatrix}$$

$$m = \text{diag}(\tilde{G} \Lambda \tilde{G}^T) \sum_{i=1}^{\frac{\tau+1}{2}} m_i = \sum_{i=1}^{\frac{\tau+1}{2}} P_i$$

$$\begin{bmatrix} m_1 & m_2 & m_3 & \cdots & m_{\frac{\tau+1}{2}} \end{bmatrix}$$

$$\begin{bmatrix} p_1 & p_2 & p_3 & \cdots & p_{\frac{\tau+1}{2}} \end{bmatrix}$$

$$\sum_{i=1}^{\frac{\tau-1}{2}} m_i < \sum_{i=1}^{\frac{\tau-1}{2}} P_i$$
Dichotomy approach

In the next step, compute a $\tilde{G}$ for components from 1 to 3 of $t$ under total power constraint $\sum_{i=1}^{3} p_i$.

$$t: \lambda_1 \lambda_2 \lambda_3 \lambda_{\frac{\tau+1}{2}}$$

$$m = \text{diag}\left(\tilde{G} \Lambda \tilde{G}^T\right) \sum_{i=1}^{\frac{\tau+1}{2}} m_i = \sum_{i=1}^{\frac{\tau+1}{2}} P_i$$

$$P_1 P_2 P_3 P_{\frac{\tau+1}{2}}$$

$$\sum_{i=1}^{\frac{\tau-1}{2}} m_i < \sum_{i=1}^{\frac{\tau-1}{2}} P_i$$
The conditions of existence of $Z$ are satisfied for $\lambda_1, \lambda_2, \lambda_3$ and $p_1, p_2, p_3$.

\[
\begin{align*}
t: & \begin{cases} 
\lambda_1 \\
\lambda_2 \\
\lambda_3 
\end{cases} \\
\begin{array}{ccc}
m_1 \\
m_2 \\
m_3 
\end{array} & \xleftarrow{\text{diag}(\tilde{G}\Lambda\tilde{G}^T)} & \begin{array}{c}
\sum_{i=1}^{3} m_i = \sum_{i=1}^{3} p_i
\end{array} \\
p_1 & p_2 & p_3 \\
\end{align*}
\]

\[\sum_{i=1}^{j} m_i \geq \sum_{i=1}^{j} p_i, 1 \leq j \leq 3\]
Complexity of PAISP

- The complexity to find the first subblock is $O(n_{SC})$
- The complexity to find all subblocks in worst case ($n_{SC}$ subblocks) is $\sum_{i=1}^{n_{SC}} n_i = O(n_{SC}^2)$
Complexity of PAISP

- The complexity to find the first subblock is $O(n_{SC})$
- The complexity to find all subblocks in worst case ($n_{SC}$ subblocks) is $\sum_{i=n_{SC}}^{1} n_i = O(n_{SC}^2)$
Computation cost comparison

(a) Optimal
(b) PAISP

Figure: One chunk vector in a GoP of video Kimonol

$\tau$ is the position where condition of existence of $Z$ is violated.
Local Power Adjustment

- The allocated power vector under total power constraint $\sum_{i=1}^{n_{SC}} p_i$ are $m_i$, $i = 1, \ldots, n_{SC}$.
- If at position $\tau$ condition is violated:
  \[ \Delta = \sum_{i=1}^{\tau} p_i - \sum_{i=1}^{\tau} m_i > 0 \]
- Since total power constraint implies that
  \[ \sum_{i=1}^{n_{SC}} p_i = \sum_{i=1}^{n_{SC}} m_i, \]
- therefore too much power has been allocated to the last $n_{SC} - \tau$ subchannels.
  \[ \sum_{i=\tau+1}^{n_{SC}} m_i - \sum_{i=\tau+1}^{n_{SC}} p_i = \Delta > 0 \]
Local Power Adjustment

- The allocated power vector under total power constraint $\sum_{i=1}^{n_{SC}} p_i$ are $m_i$, $i = 1, \ldots, n_{SC}$.
- If at position $\tau$ condition is violated:

$$\Delta = \sum_{i=1}^{\tau} p_i - \sum_{i=1}^{\tau} m_i > 0$$

- Since total power constraint implies that

$$\sum_{i=1}^{n_{SC}} p_i = \sum_{i=1}^{n_{SC}} m_i,$$

- therefore too much power has been allocated to the last $n_{SC} - \tau$ subchannels.

$$\sum_{i=\tau+1}^{n_{SC}} m_i - \sum_{i=\tau+1}^{n_{SC}} p_i = \Delta > 0$$
Local Power Adjustment

- The allocated power vector under total power constraint $\sum_{i=1}^{n_{SC}} p_i$ are $m_i$, $i = 1, \ldots, n_{SC}$.
- If at position $\tau$ condition is violated:

$$\Delta = \sum_{i=1}^{\tau} p_i - \sum_{i=1}^{\tau} m_i > 0$$

- Since total power constraint implies that

$$\sum_{i=1}^{n_{SC}} p_i = \sum_{i=1}^{n_{SC}} m_i,$$

therefore too much power has been allocated to the last $n_{SC} - \tau$ subchannels.

$$\sum_{i=\tau+1}^{n_{SC}} m_i - \sum_{i=\tau+1}^{n_{SC}} p_i = \Delta > 0$$
Local Power Adjustment

- The allocated power vector under total power constraint \( \sum_{i=1}^{n_{SC}} p_i \) are \( m_i \), \( i = 1, \ldots, n_{SC} \).
- If at position \( \tau \) condition is violated:

\[
\Delta = \sum_{i=1}^{\tau} p_i - \sum_{i=1}^{\tau} m_i > 0
\]

- Since total power constraint implies that

\[
\sum_{i=1}^{n_{SC}} p_i = \sum_{i=1}^{n_{SC}} m_i,
\]

therefore too much power has been allocated to the last \( n_{SC} - \tau \) subchannels.

\[
\sum_{i=\tau+1}^{n_{SC}} m_i - \sum_{i=\tau+1}^{n_{SC}} p_i = \Delta > 0
\]
Figure: Power Allocation under Total Power Constraint and Check the condition of

\[ m_i < \sum_{i=1}^{\tau} p_i \]
SubOptimal precoding matrix design

PALPA

\[
\begin{align*}
\lambda_1 & \quad \lambda_2 & \quad \lambda_3 & \quad \cdots & \quad \lambda_{n_{SC}} \\
\downarrow & & & & \\
m &= \text{diag}(\tilde{G}\Lambda\tilde{G}^T) \\
\sum_{i=1}^{n_{SC}} m_i &= \sum_{i=1}^{n_{SC}} p_i \\
m_1 & \quad m_2 & \quad m_3 & \quad \cdots & \quad m_{T} & \quad \cdots & \quad m_{n_{SC}} \\
\downarrow & & & & \\
p_1 & \quad p_2 & \quad p_3 & \quad \cdots & \quad p_{T} & \quad \cdots & \quad p_{n_{SC}} \\
\downarrow & & & & \\
\sum_{i=1}^{T} m_i < \sum_{i=1}^{T} p_i \\
\end{align*}
\]

\[m_i\]

\[\tau + 1 \quad n_{SC}\]

\[\Delta^{(0)}\]

Figure: compute $\Delta^{(0)}$
PALPA

\[ m = \text{diag}\left(\tilde{G}\Lambda\tilde{G}^T\right) \sum_{i=1}^{n_{SC}} m_i = \sum_{i=1}^{n_{SC}} p_i \]

\[ \sum_{i=1}^{\tau} m_i < \sum_{i=1}^{\tau} p_i \]

Figure: Local Power Adjustment

S. Zheng, M. Cagnazzo, M. Kieffer
Figure: The conditions of existence of $Z$ satisfied for $\tilde{m}_1, \tilde{m}_2, \tilde{m}_3$ and $p_1, p_2, p_3$, the first subblock is found.

- Compute $\tilde{G} = \begin{bmatrix} \sqrt{\frac{\tilde{m}_\tau}{\lambda_{SC-2}}} & \sqrt{\frac{\tilde{m}_{\tau+1}}{\lambda_{\tau+1}}} & \sqrt{\frac{\tilde{m}_{nSC}}{\lambda_{SC}}} \end{bmatrix}$.
- Compute $Z \left( Z \tilde{G} \right) \begin{bmatrix} \lambda_{\tau} & \lambda_{\tau+1} & \lambda_{nSC} \end{bmatrix} \left( Z \tilde{G} \right)^T = \begin{bmatrix} p_{\tau} & * & * \\ * & p_{\tau+1} & * \\ * & * & p_{nSC} \end{bmatrix}$. 
Figure: The conditions of existence of $Z$ satisfied for $\tilde{m}_1, \tilde{m}_2, \tilde{m}_3$ and $p_1, p_2, p_3$, the first subblock is found.

- Compute $\tilde{G} = \begin{bmatrix} \sqrt{\frac{\tilde{m}_\tau}{\lambda_{SC} - 1}} \\ \sqrt{\frac{\tilde{m}_{\tau+1}}{\lambda_{\tau+1}}} \\ \sqrt{\frac{\tilde{m}_{nSC}}{\lambda_{SC}}} \end{bmatrix}$.

- Compute $Z (Z\tilde{G}) \begin{bmatrix} \lambda_\tau \\ \lambda_{\tau+1} \\ \lambda_{nSC} \end{bmatrix} (Z\tilde{G})^T = \begin{bmatrix} p_\tau & * & * \\ * & p_{\tau+1} & * \\ * & * & p_{nSC} \end{bmatrix}$. 
Figure: The conditions of existence of $Z$ satisfied for $\tilde{m}_1, \tilde{m}_2, \tilde{m}_3$ and $p_1, p_2, p_3$, the first subblock is found.

- Compute $\tilde{G} = \begin{bmatrix} \sqrt{\tilde{m}_\tau / \lambda_{SC}} \\sqrt{\tilde{m}_{\tau+1} / \lambda_{\tau+1}} \\sqrt{\tilde{m}_{n_{SC}} / \lambda_{SC}} \end{bmatrix}$.

- Compute $Z (Z \tilde{G}) \begin{bmatrix} \lambda_{\tau} & \lambda_{\tau+1} & \lambda_{n_{SC}} \end{bmatrix} (Z \tilde{G})^T = \begin{bmatrix} p_\tau & * & * & * \* & p_{\tau+1} & * & * \* & * & * & p_{n_{SC}} \end{bmatrix}$.
We continue for the remaining parts.

\[ t: \begin{bmatrix} \lambda_1 & \lambda_2 & \lambda_3 & \text{ } & \text{ } & \text{ } & \text{ } & \lambda_{\tau_0} \end{bmatrix} \]

\[ m = \text{diag} \left( \tilde{G} \Lambda \tilde{G}^T \right) \sum_{i=1}^{\tau_0} m_i = \sum_{i=1}^{\tau_0} p_i \]

\[ \begin{bmatrix} m_1 & m_2 & m_3 & \text{ } & \text{ } & \text{ } & \text{ } & m_{\tau_0} \end{bmatrix} \]

\[ \begin{bmatrix} p_1 & p_2 & p_3 & \text{ } & \text{ } & \text{ } & \text{ } & p_{\tau_0} \end{bmatrix} \]

\[ \sum_{i=1}^{\tau_0} m_i = \sum_{i=1}^{\tau_0} p_i \]

**Figure**: Power Allocation under Total Power Constraint and Check the condition of existence of \( Z \)
Complexity to find the first subblock in worst case (adjustment the power for $m_2$ to $m_{SC}$) $O(n_{SC})$.

The complexity to find all split positions in worst case (at end $s = n_{SC}$) is $\sum_{i=n_{SC}}^{1} n_i = O(n_{SC}^2)$.
• Complexity to find the first subblock in worst case (ajustement the power for $m_2$ to $m_{SC}$) $O(n_{SC})$.

• The complexity to find all split positions in worst case (at end $s = n_{SC}$) is $\sum_{i=n_{SC}}^{1} n_i = O(n_{SC}^2)$. 
SubOptimal precoding matrix design

Computation cost comparison\(^9\)

\[ \tau \]

\(\tau\) is the position where condition of existence of \(Z\) is violated.

\(^9\)The complexity of PALPA to find all subblocks in worst case is as same as PAISP \(O(n_{SC}^2)\).