# Joint Matrix Decompositions for Gaussian Communication Networks

Anatoly Khina

EE–Systems Department
Tel Aviv University

*"A friend to all is a friend to none."* -Aristotle

# Talk Outline

1. Framework and motivation
2. Background:
    - MIMO point-to-point scheme
    - Overview of orthogonal matrix decompositions: SVD, QR, GMD, GTD, . . .
3. (New) MIMO multicast scheme
    - Two-user: via new joint decomposition of two matrices
    - Multi-user: via algebraic space–time coding structure
4. Various applications
5. New information-theoretic results

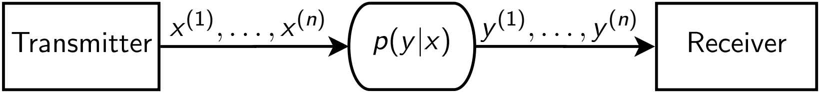
# Part I

# Framework and Motivation

# MIMO Multicast (Closed Loop): State of the Art

| | Unicast | Multicast |
|---|---|---|
| Theory | | |
| SISO | | |
| MIMO | | |

# Unicast: Point-to-Point Communication




## Memoryless channel

$$p\left(y^{(1)}, \ldots, y^{(n)} \middle| x^{(1)}, \ldots, x^{(n)}\right) = \prod_{t=1}^{n} p\left(y^{(t)} \middle| x^{(t)}\right)$$

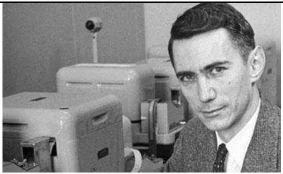
## Channel capacity [Shannon '48]

Best achievable rate over memoryless channel $p(y|x)$:

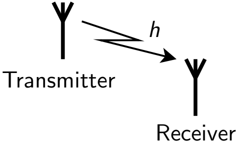
$$C = \max_{p(x)} I(x;y)$$

- Maximization over all admissible input distributions $p(x)$

# MIMO Multicast (Closed Loop): State of the Art

| | Unicast | Multicast |
|---|---|---|
| Theory |  ✓ | |
| SISO | | |
| MIMO | | |

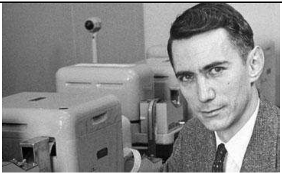
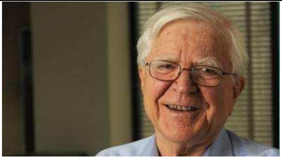
# Single-Input Single-Output (SISO) Unicast



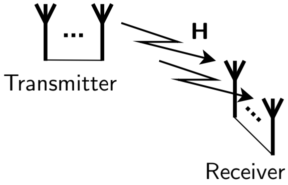

$$y^{(t)} = hx^{(t)} + z^{(t)}$$

- $x$ – Input of power 1
- $y$ – Output
- $h$ – Channel gain
- $z$ – White Gaussian noise $\sim \mathcal{CN}(0,1)$
- Optimal communication rate (capacity): $C = \log(1 + |h|^2)$
- **Good practical codes that approach capacity are known!**

# MIMO Multicast (Closed Loop): State of the Art

| | Unicast | Multicast |
|---|---|---|
| Theory |  ✓ | |
| SISO |  ✓ | |
| MIMO | | |

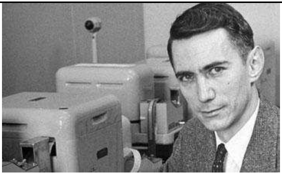
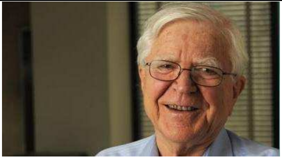
# Multiple-Input Multiple-Output (MIMO) Unicast



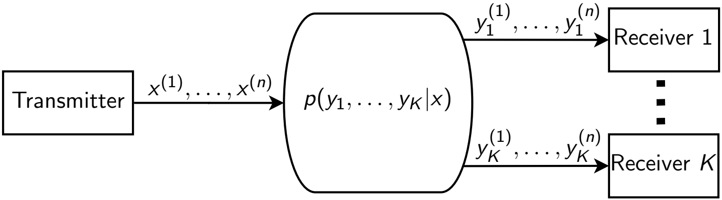

$$\boldsymbol{y}^{(t)} = \mathbf{H}\boldsymbol{x}^{(t)} + \boldsymbol{z}^{(t)}$$

- $\boldsymbol{x}$ – Input vector of power $1 \cdot N$
- $\boldsymbol{y}$ – Output vector
- $\mathbf{H}$ – Channel matrix
- $\mathbf{H}_{k\ell}$ – Gain from transmit-antenna $\ell$ to receive-antenna $k$
- $\boldsymbol{z}$ – White Gaussian noise $\sim \mathcal{CN}(\mathbf{0}, \mathbf{I})$
- Capacity: $C = \max_{\mathbf{C}_{\boldsymbol{x}}} \log \left| \mathbf{I} + \mathbf{H}\mathbf{C}_{\boldsymbol{x}}\mathbf{H}^\dagger \right|$

# MIMO Multicast (Closed Loop): State of the Art

| | Unicast | Multicast |
|---|---|---|
| Theory |  ✓ | |
| SISO |  ✓ | |
| MIMO |  ✓ | |

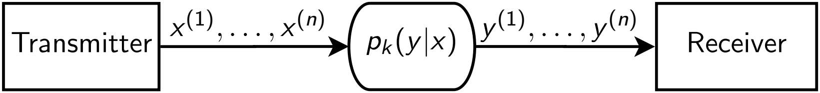
# Multicast: Communication over a Compound Channel




## Physical-layer multicast

- Transmit same message to $K$ receivers: $y_1, \ldots, y_K$
- All receivers recover message with negligible error probability

# Multicast: Communication over a Compound Channel




## Compound channel

- $K$ possible channel realizations: $\{p_k(y|x) | k = 1, \ldots K\}$
- Transmitter does not know $k$
- Error probability is negligible for all $p_k(y|x)$ simultaneously

# Multicast: Communication over a Compound Channel

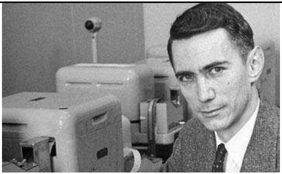
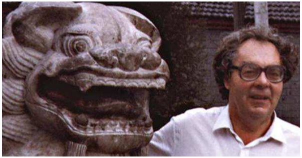
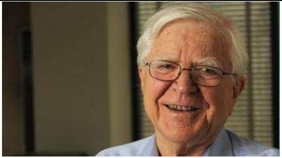
**Compound channel / multicast capacity**
[Dobrushin '59][Blackwell-Breiman-Thmoasian '59][Wolfowitz '60]

Best achievable rate over $K$-user memoryless channel $\{p(y_i|x)\}$:

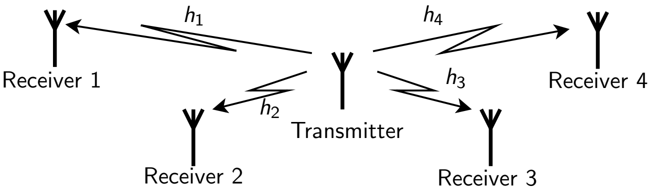
$$C = \max_{p(x)} \min_{i=1,\ldots,K} I(x; y_i)$$

- Maximization over all admissible input distribution $p(x)$
- Minimization over all users

# MIMO Multicast (Closed Loop): State of the Art

| | Unicast | Multicast |
|---|---|---|
| Theory |  ✓ |  ✓ |
| SISO |  ✓ | |
| MIMO |  ✓ | |

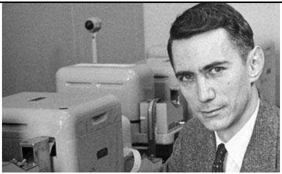
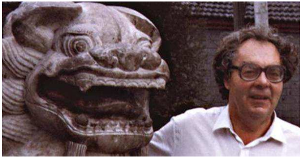
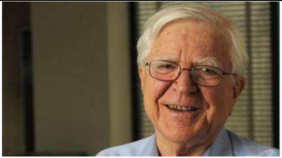
# SISO Multicast



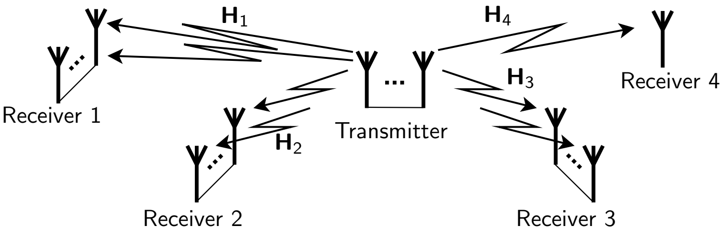

$$y_i = h_i x + z_i \quad i = 1, \ldots, K$$

- $x$ – Input of power 1
- $y_i$ – Output of user $i$
- $h_i$ – Channel gain to user $i$
- $z_i$ – White Gaussian noise $\sim \mathcal{CN}(0,1)$
- Capacity: $C = \min_i \log(1 + |h_i|^2)$

# MIMO Multicast (Closed Loop): State of the Art

| | Unicast | Multicast |
|---|---|---|
| Theory |  ✓ |  ✓ |
| SISO |  ✓ |  ✓ |
| MIMO |  ✓ | |

# Gaussian MIMO Multicast




$$\boldsymbol{y}_i = \mathbf{H}_i \boldsymbol{x} + \boldsymbol{z}_i \quad i = 1, \ldots, K$$

- $\boldsymbol{x}$ – $N \times 1$ input vector of power $N \cdot 1$
- $\boldsymbol{y}_i$ – Output vector of user $i$
- $\mathbf{H}_i$ – Channel matrix to user $i$
- $\boldsymbol{z}_i$ – White Gaussian noise vector $\sim \mathcal{CN}(\mathbf{0}, \mathbf{I})$
- "Closed loop" (Full channel knowledge everywhere)

# Optimal Achievable Rate (Capacity)

## Multicast capacity

$$C = \max_{\mathbf{C_x}} \min_{i=1,\ldots,K} \log \left| \mathbf{I} + \mathbf{H}_i \mathbf{C_x} \mathbf{H}_i^\dagger \right|$$

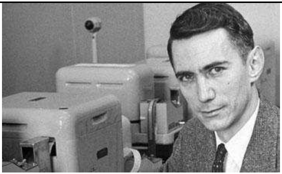
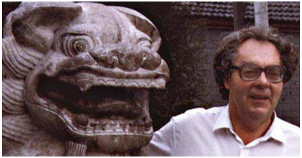
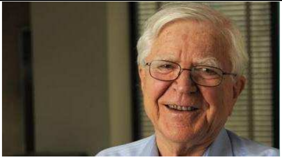
- Optimization over covariance matrices $\mathbf{C_x}$ satisfying the power constraint

## High SNR (and Square Matrices)

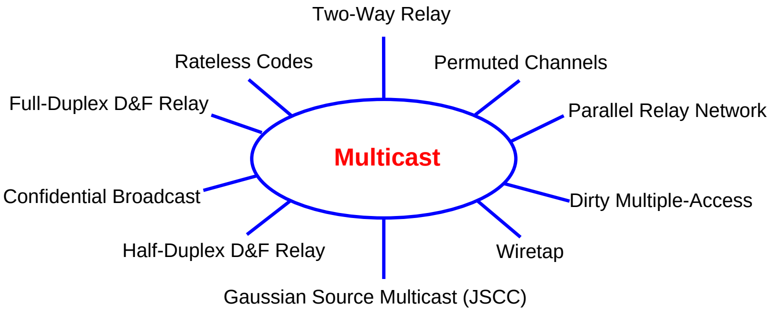
- Optimal covariance is (approximately) white: $\mathbf{C_x} \approx \mathbf{I}$

$$C_{\mathrm{WI}} \approx 2 \min_{i=1,\ldots,K} \log \left| \mathbf{H}_i \right|$$

# MIMO Multicast (Closed Loop): State of the Art

| | Unicast | Multicast |
|---|---|---|
| **Theory** |  ✓ |  ✓ |
| **SISO** |  ✓ |  ✓ |
| **MIMO** |  ✓ |  ? |

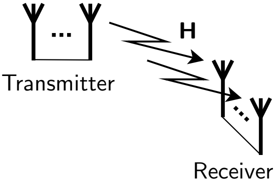
# Summary: Multicast is (Almost) Everywhere...



# Part II

# MIMO Point-to-Point Schemes

# MIMO Unicast




$$\boldsymbol{y} = \mathbf{H}\boldsymbol{x} + \boldsymbol{z}$$

## Capacity

$$C = \log \left| \mathbf{I} + \mathbf{H}\mathbf{C}_{\boldsymbol{x}}\mathbf{H}^{\dagger} \right|$$

- But how is this rate achieved in practice?

# What Do We Mean by Practical?

**Capacity is achieved**

**Black box approach: Reduce MIMO to SISO**

- "Off-the-shelf" standard encoders and decoders
- Any fixed-rate SISO AWGN codes
- Simple signal processing:
  - linear operations (+modulo)
  - Successive interference cancellation (SIC)
  - Or modulo arithmetic instead of SIC
- Gap-to-capacity dictated by gap-to-capacity of SISO codes

# Singular-Value Decomposition (SVD) Scheme [Telatar '99]

- $\mathbf{H} = \mathbf{Q}\mathbf{D}\boldsymbol{V}^\dagger$
- $\mathbf{Q}$ and $\boldsymbol{V}$ – unitary
- Tx applies $\boldsymbol{V}$ and Rx applies $\mathbf{Q}^\dagger$
- $\mathbf{D} = \begin{pmatrix} d_1 & 0 & 0 & \cdots & 0 \\ 0 & d_2 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & d_{N-1} & 0 \\ 0 & \cdots & 0 & 0 & d_N \end{pmatrix} \Rightarrow \begin{aligned} y_1 &= d_1 x_1 + z_1 \\ y_2 &= d_2 x_2 + z_2 \\ &\vdots \\ y_N &= d_N x_N + z_N \end{aligned}$
- Results in parallel scalar sub-channels (each sub-channel has a different SNR)
- Apply water-filling on $\{x_1, \ldots, x_N\}$: $\boldsymbol{x} = \boldsymbol{V}\boldsymbol{W}\boldsymbol{c}$

# SVD-based scheme for a given input covariance $\mathbf{C}_x$

- $\mathbf{H}\mathbf{C}_{\boldsymbol{x}}^{1/2} = \mathbf{Q}\mathbf{D}\boldsymbol{V}^\dagger$
- $\mathbf{Q}$ and $\boldsymbol{V}$ – unitary; $\mathbf{C}_{\boldsymbol{x}}^{1/2}$ – any matrix $\mathbf{B}$ s.t. $\mathbf{B}\mathbf{B}^\dagger = \mathbf{C}_{\boldsymbol{x}}$
- Tx applies $\mathbf{C}_{\boldsymbol{x}}^{1/2}\boldsymbol{V}$ and Rx applies $\mathbf{Q}$
- $$\mathbf{D} = \begin{pmatrix} d_1 & 0 & 0 & \cdots & 0 \\ 0 & d_2 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & d_{N-1} & 0 \\ 0 & \cdots & 0 & 0 & d_N \end{pmatrix} \Rightarrow \begin{aligned} y_1 &= d_1 x_1 + z_1 \\ y_2 &= d_2 x_2 + z_2 \\ &\vdots \\ y_N &= d_N x_N + z_N \end{aligned}$$
- Results in parallel scalar sub-channels (each sub-channel has a different SNR)
- ~~Apply water-filling on $\{x_1, \dots, x_n\}$: $\boldsymbol{x} = \boldsymbol{V}\boldsymbol{W}\boldsymbol{c}$~~ $\boldsymbol{x} = \mathbf{C}_{\boldsymbol{x}}^{1/2}\boldsymbol{V}\boldsymbol{c}$

# Singular-Value Decomposition (SVD) Scheme [Telatar '99]

- SVD scheme with given $\mathbf{C}_{\boldsymbol{x}}$ achieves : $R = \log \left| \mathbf{I}_N + \mathbf{H}\mathbf{C}_{\boldsymbol{x}}\mathbf{H}^\dagger \right|$
- Attains capacity for optimal choice of $\mathbf{C}_{\boldsymbol{x}}$
- Can be used to attain capacity for other covariance constraint scenarios (e.g., individual power constraints)

# QRD-based: Zero-forcing VBLAST / GDFE [Foschini '96]

- Based on QR decomposition (QRD)
- $\mathbf{H} = \mathbf{Q}\boldsymbol{T}$
- $\mathbf{Q}$ – unitary; $\boldsymbol{T}$ – triangular
- Rx applies $\mathbf{Q}^\dagger$ (no SP is required by Tx)
- $$\boldsymbol{T} = \begin{pmatrix} t_1 & * & * & \cdots & * \\ 0 & t_2 & * & \cdots & * \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ 0 & 0 & \cdots & t_{N-1} & * \\ 0 & 0 & \cdots & 0 & t_N \end{pmatrix} \quad \Rightarrow \quad \begin{aligned} y_1^{\text{eff}} &= t_1 x_1 + z_1 \\ y_2^{\text{eff}} &= t_2 x_2 + z_2 \\ &\vdots \\ y_N^{\text{eff}} &= t_N x_N + z_N \end{aligned}$$
- Off-diagonal elements are canceled via successive interference cancellation (SIC)

# QRD-based: Zero-forcing VBLAST / GDFE [Foschini '96]

- Based on QR decomposition (QRD)
- $\mathbf{H} = \mathbf{Q}\boldsymbol{T}$
- $\mathbf{Q}$ – unitary; $\boldsymbol{T}$ – triangular
- Rx applies $\mathbf{Q}^\dagger$ (no SP is required by Tx)
- $$\boldsymbol{T} = \begin{pmatrix} t_1 & * & * & \cdots & * \\ 0 & t_2 & * & \cdots & * \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ 0 & 0 & \cdots & t_{N-1} & * \\ 0 & 0 & \cdots & 0 & t_N \end{pmatrix} \Rightarrow \begin{aligned} y_1^{\text{eff}} &= t_1 x_1 + z_1 \\ y_2^{\text{eff}} &= t_2 x_2 + z_2 \\ &\vdots \\ y_N^{\text{eff}} &= t_N x_N + z_N \end{aligned}$$
- Off-diagonal elements are canceled via successive interference cancellation (SIC)

# QRD-based: Zero-forcing VBLAST / GDFE [Foschini '96]

- Based on QR decomposition (QRD)
- $\mathbf{H} = \mathbf{Q}\boldsymbol{T}$
- $\mathbf{Q}$ – unitary; $\boldsymbol{T}$ – triangular
- Rx applies $\mathbf{Q}^\dagger$ (no SP is required by Tx)
- $$\boldsymbol{T} = \begin{pmatrix} t_1 & * & * & \cdots & 0 \\ 0 & t_2 & * & \cdots & 0 \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ 0 & 0 & \cdots & t_{N-1} & 0 \\ 0 & 0 & \cdots & 0 & t_N \end{pmatrix} \Rightarrow \begin{aligned} y_1^{\text{eff}} &= t_1 x_1 + z_1 \\ y_2^{\text{eff}} &= t_2 x_2 + z_2 \\ &\vdots \\ y_N^{\text{eff}} &= t_N x_N + z_N \end{aligned}$$
- Off-diagonal elements are canceled via successive interference cancellation (SIC)

# QRD-based: Zero-forcing VBLAST / GDFE [Foschini '96]

- Based on QR decomposition (QRD)
- $\mathbf{H} = \mathbf{Q}\boldsymbol{T}$
- $\mathbf{Q}$ – unitary; $\boldsymbol{T}$ – triangular
- Rx applies $\mathbf{Q}^\dagger$ (no SP is required by Tx)
- $$\boldsymbol{T} = \begin{pmatrix} t_1 & * & * & \cdots & 0 \\ 0 & t_2 & * & \cdots & 0 \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ 0 & 0 & \cdots & t_{N-1} & 0 \\ 0 & 0 & \cdots & 0 & t_N \end{pmatrix} \Rightarrow \begin{matrix} y_1^{\text{eff}} = t_1 x_1 + z_1 \\ y_2^{\text{eff}} = t_2 x_2 + z_2 \\ \vdots \\ y_N^{\text{eff}} = t_N x_N + z_N \end{matrix}$$
- Off-diagonal elements are canceled via successive interference cancellation (SIC)

# MMSE-VBLAST for a given covariance $\mathbf{C}_{\boldsymbol{x}}$ [Hassibi '00]

- $\begin{bmatrix} \mathbf{H}\mathbf{C}_{\boldsymbol{x}}^{1/2} \\ \mathbf{I}_N \end{bmatrix} = \mathbf{Q}\boldsymbol{T}$
- $\mathbf{Q}$ – unitary; $\tilde{\mathbf{Q}}$ – $N \times N$ submatrix of $\mathbf{Q}$
- Rx applies $\tilde{\mathbf{Q}}^\dagger$ (no SP is required by Tx)
- $\tilde{\mathbf{Q}}^\dagger$ contains Wiener-filtering ("FFE")
- Effective noise has channel noise and "ISI" components
- Effective SNRs satisfy: $t_i^2 = 1 + \mathrm{SNR}_i$

$$\log(t_i^2) = \log(1 + \mathrm{SNR}_i) = I(c_i; \boldsymbol{y} | c_{i+1}^N)$$

- Off-diagonal elements above diagonal canceled via SIC

# MMSE-VBLAST for a given covariance $\mathbf{C_x}$

- For square invertible $\mathbf{H}$, ZF-VBLAST achieves: $R = \left|\mathbf{H}\mathbf{H}^\dagger\right|$
  $\left( \text{Using } \mathbf{C_x} \text{ at the transmitter achieves: } R = \left|\mathbf{H}\mathbf{C_x}\mathbf{H}^\dagger\right| \right)$
- MMSE-VBLAST achieves: $R = \left|\mathbf{I}_N + \mathbf{H}\mathbf{C_x}\mathbf{H}^\dagger\right|$

## Canonical channel matrix

$$\begin{bmatrix} \mathbf{H}\mathbf{C}_{\mathbf{x}}^{1/2} \\ \mathbf{I}_N \end{bmatrix} = \mathbf{Q} \begin{bmatrix} \mathbf{H}^{\mathrm{eff}} \\ \mathbf{0} \end{bmatrix}$$

- Canonical channel matrix $\mathbf{H}^{\mathrm{eff}}$ is square and invertible
- Analagous to the canonical system response of [Cioffi-Dudevoir-Eyuboglu-Forney '95]
- Treating square invertible matrices suffices!

# MMSE-VBLAST with precoding for a given covariance $\mathbf{C_x}$

- $\mathbf{H}^{\text{eff}} = \mathbf{Q}\boldsymbol{T}\boldsymbol{V}^{\dagger}$
- $\boldsymbol{V}$ can be used to design diagonal values $\Leftrightarrow$ design SNRs

# MMSE-VBLAST with precoding for a given covariance $\mathbf{C_x}$

- $\mathbf{H}^{\text{eff}} = \mathbf{Q}\boldsymbol{T}\boldsymbol{V}^{\dagger}$
- $\boldsymbol{V}$ can be used to design diagonal values $\Leftrightarrow$ design SNRs

**SVD-scheme as MMSE-VBLAST**

Choosing $\boldsymbol{V}$ of the SVD of $\mathbf{H}^{\text{eff}}$ $\Rightarrow$ SVD scheme (no SIC needed)

# MMSE-VBLAST with precoding for a given covariance $\mathbf{C_x}$

- $\mathbf{H}^{\text{eff}} = \boldsymbol{Q}\boldsymbol{T}\boldsymbol{V}^{\dagger}$
- $\boldsymbol{V}$ can be used to design diagonal values $\Leftrightarrow$ design SNRs

**SVD-scheme as MMSE-VBLAST**

Choosing $\boldsymbol{V}$ of the SVD of $\mathbf{H}^{\text{eff}}$ $\Rightarrow$ SVD scheme (no SIC needed)

- What about other choices of $\boldsymbol{V}$?

# Generalized Triangular Decomposition (GTD)

## [Jiang-Hager-Li '08][Zhang-Wong]

- $\boldsymbol{T}$ is upper-triangular

$$\mathbf{H}^{\mathrm{eff}} = \mathbf{Q}\boldsymbol{T}\boldsymbol{V}^\dagger = \mathbf{Q}\begin{pmatrix} t_1 & * & \cdots & * \\ 0 & t_2 & \cdots & * \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & t_N \end{pmatrix}\boldsymbol{V}^\dagger$$

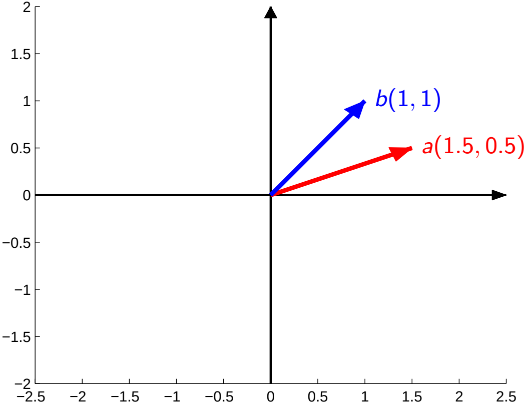
- Desired diagonal: $\boldsymbol{t} = (t_1, t_2, \ldots, t_N) \to$ Ordered vector: $\tilde{\boldsymbol{t}}$
- Ordered singular-value vector: $\boldsymbol{\sigma} = (\sigma_1, \sigma_2, \ldots, \sigma_N)$
- Weyl's condition: $\boldsymbol{\sigma} \succeq \boldsymbol{t}$

$$\prod_{i=1}^{\ell} \sigma_i \geq \prod_{i=1}^{\ell} |\tilde{t}_i| \qquad \ell = 1, \ldots, N$$

$$\prod_{i=1}^{N} \sigma_i = \prod_{i=1}^{N} |\tilde{t}_i| \qquad (\ell = N)$$

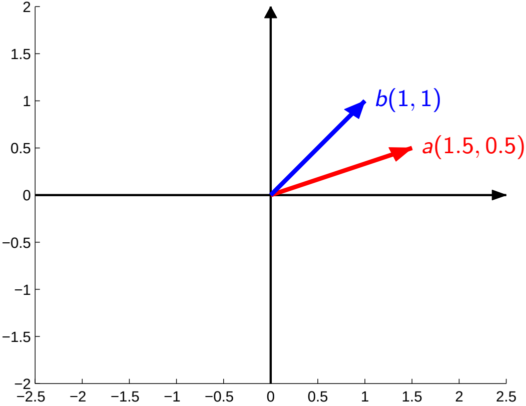
# QR Interpretation

$$\mathbf{H}^{\text{eff}} = \begin{bmatrix} a_x & b_x \\ a_y & b_y \end{bmatrix} = \mathbf{Q}\boldsymbol{T} \Leftrightarrow \boldsymbol{T} = \mathbf{Q}^\dagger \mathbf{H}^{\text{eff}} = \begin{bmatrix} \cos\theta_\ell & \sin\theta_\ell \\ -\sin\theta_\ell & \cos\theta_\ell \end{bmatrix} \begin{bmatrix} a_x & b_x \\ a_y & b_y \end{bmatrix}$$



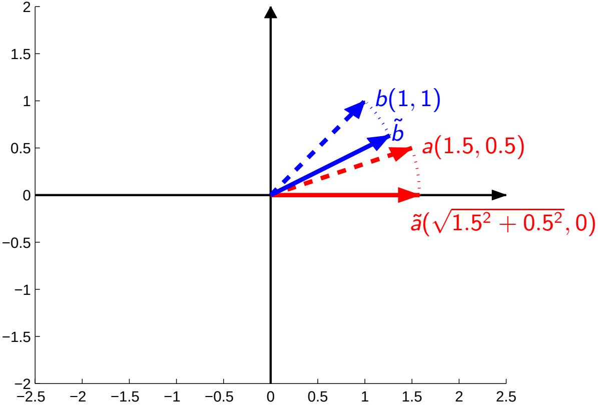
# QR Interpretation

$$\mathbf{H}^{\text{eff}} = \begin{bmatrix} 1.5 & 1 \\ 0.5 & 1 \end{bmatrix} = \mathbf{Q}\,\boldsymbol{T} \;\Leftrightarrow\; \boldsymbol{T} = \mathbf{Q}^{\dagger}\mathbf{H}^{\text{eff}} = \begin{bmatrix} \cos\theta_\ell & \sin\theta_\ell \\ -\sin\theta_\ell & \cos\theta_\ell \end{bmatrix} \begin{bmatrix} 1.5 & 1 \\ 0.5 & 1 \end{bmatrix}$$



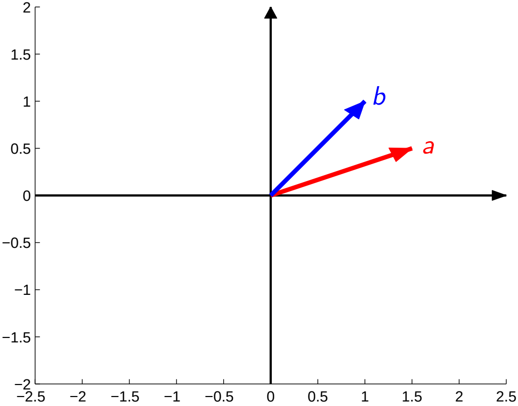
# QR Interpretation

$$\mathbf{H}^{\text{eff}} = \begin{bmatrix} 1.5 & 1 \\ 0.5 & 1 \end{bmatrix} = \mathbf{Q}\boldsymbol{T} \Leftrightarrow \boldsymbol{T} = \mathbf{Q}^{\dagger}\mathbf{H}^{\text{eff}} = \begin{bmatrix} \cos\theta_\ell & \sin\theta_\ell \\ -\sin\theta_\ell & \cos\theta_\ell \end{bmatrix} \begin{bmatrix} 1.5 & 1 \\ 0.5 & 1 \end{bmatrix}$$



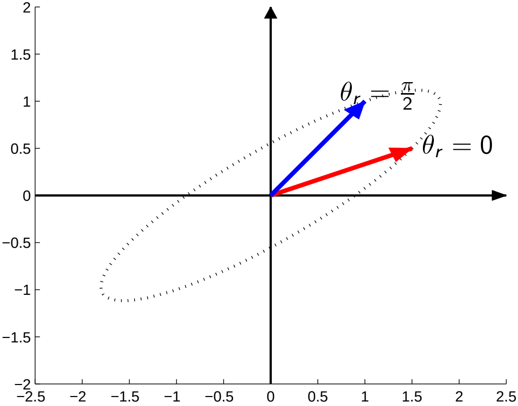
# GTD Interpretation

$$\mathbf{Q}^{\dagger}\mathbf{H}^{\text{eff}}\boldsymbol{V} = \begin{bmatrix} \cos\theta_\ell & \sin\theta_\ell \\ -\sin\theta_\ell & \cos\theta_\ell \end{bmatrix} \begin{bmatrix} a_x & b_x \\ a_y & b_y \end{bmatrix} \begin{bmatrix} \cos\theta_r & -\sin\theta_r \\ \sin\theta_r & \cos\theta_r \end{bmatrix}$$

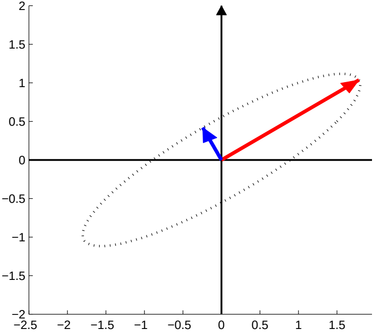


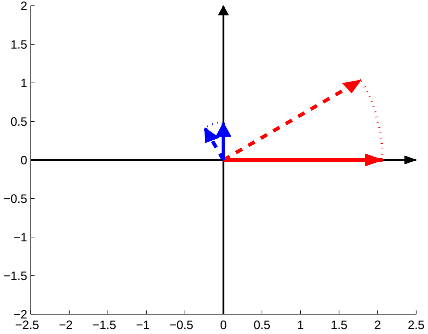
# GTD Interpretation

$$\mathbf{Q}^{\dagger}\mathbf{H}^{\text{eff}}\boldsymbol{V} = \begin{bmatrix} \cos\theta_\ell & \sin\theta_\ell \\ -\sin\theta_\ell & \cos\theta_\ell \end{bmatrix} \begin{bmatrix} a_x\cos\theta_r + b_x\sin\theta_r & a_x\cos(\theta_r+\frac{\pi}{2}) + b_x\sin(\theta_r+\frac{\pi}{2}) \\ a_y\cos\theta_r + b_y\sin\theta_r & a_y\cos(\theta_r+\frac{\pi}{2}) + b_y\sin(\theta_r+\frac{\pi}{2}) \end{bmatrix}$$



# SVD Interpretation




- The SVD corresponds to the longest and shortest vectors/diagonal elements
- These vectors are necessarily orthogonal

# Geometric Mean Decomposition (GMD)

[Kosowski-Smoktunowicz '99][Zhang-Kavčić-Wong '05][Jiang-Hager-Li '05]

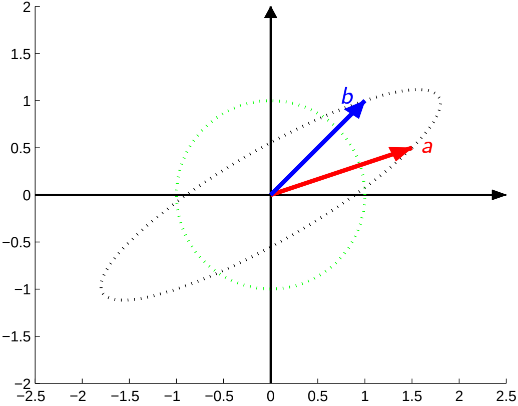
$$\mathbf{H}^{\text{eff}} = \mathbf{Q}\boldsymbol{T}\boldsymbol{V}^{\dagger} = \mathbf{Q}\begin{pmatrix} t & * & \cdots & * \\ 0 & t & \cdots & * \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & t \end{pmatrix}\boldsymbol{V}^{\dagger}$$

- Constant diagonal: $t = \sqrt[N]{\prod_{i=1}^{N} \sigma_i}$
- Geometric mean of singular values
- Always possible!
- AM-GM inequality $\Rightarrow$ Weyl's condition is always satisfied

$$\prod_{i=1}^{\ell} \sigma_i \geq |t|^{\ell} \qquad \ell = 1, \ldots, N$$

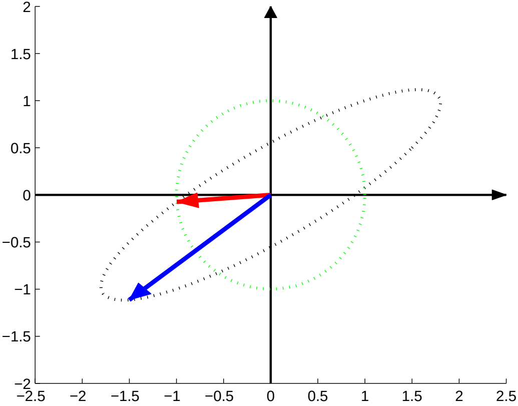
# Geometric Mean Decomposition (GMD)

[Kosowski-Smoktunowicz '99][Zhang-Kavčić-Wong '05][Jiang-Hager-Li '05]



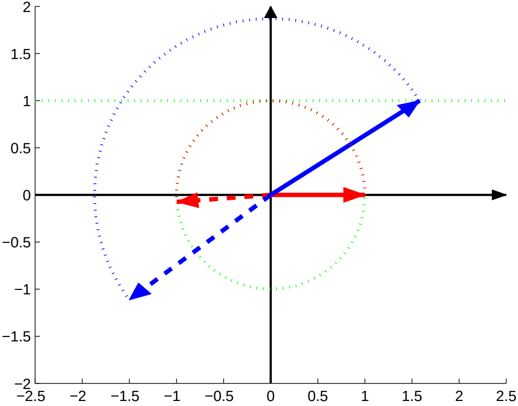
# Geometric Mean Decomposition (GMD)

[Kosowski-Smoktunowicz '99][Zhang-Kavčić-Wong '05][Jiang-Hager-Li '05]



# Geometric Mean Decomposition (GMD)

[Kosowski-Smoktunowicz '99][Zhang-Kavčić-Wong '05][Jiang-Hager-Li '05]



# GMD-based Scheme

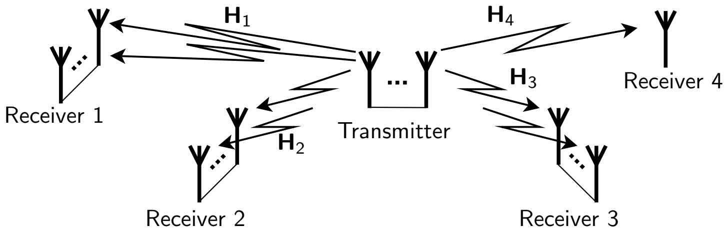
## [Zhang-Kavčić-Wong IT'05][Jiang-Hager-Li SP'05]

- All sub-channels have the same SNR
- No need for bit-loading
- The same codebook can be used over all sub-channels
- Again, a DPC variant can be constructed

# Part III

## MIMO Multicast via Joint Matrix Decompositions

# Gaussian MIMO Multicast




$$\boldsymbol{y}_i = \mathbf{H}_i \boldsymbol{x} + \boldsymbol{z}_i \quad i = 1, \ldots, K$$

- $\boldsymbol{x}$ – $N \times 1$ input vector of power $N \cdot 1$
- $\boldsymbol{y}_i$ – Output vector of user $i$
- $\mathbf{H}_i$ – Channel matrix to user $i$
- $\boldsymbol{z}_i$ – White Gaussian noise vector $\sim \mathcal{CN}(\mathbf{0}, \mathbf{I})$
- "Closed loop" (Full channel knowledge everywhere)

# Goal: As in the Point-to-Point Setting...

## Capacity is achieved

## Black box approach: Reduce MIMO to SISO

- "Off-the-shelf" standard encoders and decoders
- Any fixed-rate SISO AWGN codes
- Simple signal processing:
  - linear operations (+modulo)
  - Successive interference cancellation (SIC)
  - Or modulo arithmetic instead of SIC
- Gap-to-capacity dictated by gap-to-capacity of SISO codes

# Generalization of SVD-based Scheme?

$$\mathbf{H}_1^{\text{eff}} = \mathbf{Q}_1 \mathbf{D}_1 \boldsymbol{V}_1^\dagger$$
$$\mathbf{H}_2^{\text{eff}} = \mathbf{Q}_2 \mathbf{D}_2 \boldsymbol{V}_2^\dagger$$

- Precoding matrix $\boldsymbol{V}_i$ depends on the channel matrix $\mathbf{H}_i^{\text{eff}}$
- But $\boldsymbol{V}$ is shared by all users!
- Cannot be used for multi-user case ☹

**Diagonal Matrices**

Even if all matrices are diagonal ⇒ **Bottleneck problem!**

# Generalization of QR-based Scheme?

$$\mathbf{H}_1^{\text{eff}} = \mathbf{Q}_1 \boldsymbol{T}_1$$
$$\mathbf{H}_2^{\text{eff}} = \mathbf{Q}_2 \boldsymbol{T}_2$$

- $\boldsymbol{T}_i$ depends on $\mathbf{H}_i$
- $\text{diag}(\boldsymbol{T}_1) \neq \text{diag}(\boldsymbol{T}_2) \Rightarrow$ **different sub-channel gains!**

## Bottleneck problem

- **Info. Theory:** $\sum_{j=1}^{N} \log |T_{1;jj}|^2 = \sum_{j=1}^{N} \log |T_{2;jj}|^2$ ✓
- **Comm.:** $R_j = \log |\min \{ T_{1;jj}, T_{2;jj} \}|^2$ ✗
- Can we have equal diagonals?

# Bottleneck Problem

**P2P:**

$$\mathbf{H}_1^{\text{eff}} = \begin{pmatrix} 2 & * \\ 0 & 6 \end{pmatrix} \qquad \mathbf{H}_2^{\text{eff}} = \begin{pmatrix} 3 & * \\ 0 & 4 \end{pmatrix}$$

$$R_{1;1} = \log(2^2), R_{1;2} = \log(6^2) \qquad R_{2;1} = \log(3^2), R_{2;2} = \log(4^2)$$

$$C_1 = R_{1;1} + R_{1;2} = \log(12^2) \qquad C_2 = R_{2;1} + R_{2;2} = \log(12^2)$$

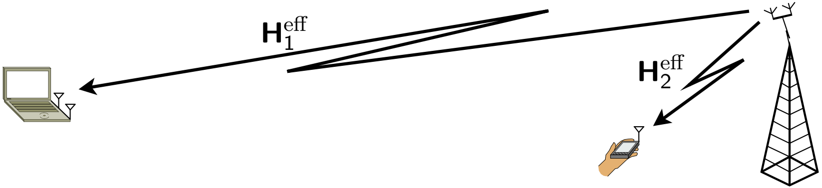
**Multicast:**

$$R_1 = \log(\min\{2^2, 3^2\}) = \log\left(2^2\right)$$

$$R_2 = \log(\min\{6^2, 4^2\}) = \log\left(4^2\right)$$

$$R^{\text{multicast}} = R_1 + R_2 = \log\left(64\right) < \log\left(144\right) = \log(12^2) = C^{\text{multicast}}$$

# Example: Degrees-of-Freedom Mismatch




$$\mathbf{H}_1 = \begin{bmatrix} 3 & 0 \\ 0 & 3 \end{bmatrix}, \qquad \mathbf{H}_2 = \begin{bmatrix} \sqrt{99} & 0 \end{bmatrix}$$

$$\mathbf{H}_1^{\text{eff}} = \begin{bmatrix} \sqrt{10} & 0 \\ 0 & \sqrt{10} \end{bmatrix}, \qquad \mathbf{H}_2^{\text{eff}} = \begin{bmatrix} 10 & 0 \\ 0 & 1 \end{bmatrix}$$

- $C_1^{\text{WI}} = 2\log\left(1+3^2\right) = \log\left(1+\left(\sqrt{99}\right)^2\right) = C_2^{\text{WI}}$

# Example: Degrees-of-Freedom Mismatch

Best practical existing schemes for the example at **high SNR**:

- Time-sharing: 50% of capacity $\left(= \frac{1}{\text{No. of users}}\right)$
- Single-stream beamforming: 50% of capacity $\left(= \frac{\text{used DoF}}{\text{total DoF}}\right)$
- Alamouti coding: 50% of capacity $\left(= \frac{\text{used DoF}}{\text{total DoF}}\right)$

None of these schemes approaches capacity!

- For more users/antennas $\rightarrow$ achievable rate goes down

# Idea

- SVD uses both $\mathbf{Q}$ and $\boldsymbol{V}$ but tries to diagonalize
- But triangularization suffices
- QR uses only $\mathbf{Q}$...

# Idea

- SVD uses both $\mathbf{Q}$ and $\boldsymbol{V}$ but tries to diagonalize
- But triangularization suffices
- QR uses only $\mathbf{Q}$...

**What else can the $\boldsymbol{V}$ serve for?**

# Idea

- SVD uses both $\mathbf{Q}$ and $\boldsymbol{V}$ but tries to diagonalize
- But triangularization suffices
- QR uses only $\mathbf{Q}$...

**What else can the $\boldsymbol{V}$ serve for?**

- Can $\boldsymbol{V}$ help in QR case to achieve equal diagonals?

# Idea

- SVD uses both $\mathbf{Q}$ and $\boldsymbol{V}$ but tries to diagonalize
- But triangularization suffices
- QR uses only $\mathbf{Q}$...

### What else can the $\boldsymbol{V}$ serve for?

- Can $\boldsymbol{V}$ help in QR case to achieve equal diagonals?
- **YES!**

# Idea

- SVD uses both $\mathbf{Q}$ and $\boldsymbol{V}$ but tries to diagonalize
- But triangularization suffices
- QR uses only $\mathbf{Q}$...

**What else can the $\boldsymbol{V}$ serve for?**

- Can $\boldsymbol{V}$ help in QR case to achieve equal diagonals?
- **YES!**

# Idea

- SVD uses both $\mathbf{Q}$ and $\boldsymbol{V}$ but tries to diagonalize
- But triangularization suffices
- QR uses only $\mathbf{Q}$...

**What else can the $\boldsymbol{V}$ serve for?**

- Can $\boldsymbol{V}$ help in QR case to achieve equal diagonals?
- **YES!**

*"The worst form of inequality is to try to make unequal things equal."* -Aristotle

# Joint Triangularization

## Theorem [Kh.-Kochman-Erez SP'12]

- $\mathbf{H}_1^{\mathrm{eff}}$ and $\mathbf{H}_2^{\mathrm{eff}}$ — $N \times N$ matrices
- $\mathbf{H}_1^{\mathrm{eff}}$ and $\mathbf{H}_2^{\mathrm{eff}}$ can be jointly decomposed as:

$$\mathbf{H}_1^{\mathrm{eff}} = \mathbf{Q}_1 \boldsymbol{T}_1 \boldsymbol{V}^\dagger$$
$$\mathbf{H}_2^{\mathrm{eff}} = \mathbf{Q}_2 \boldsymbol{T}_2 \boldsymbol{V}^\dagger$$

where

- $\mathbf{Q}_1$, $\mathbf{Q}_2$, $\boldsymbol{V}$ — Unitary
- $\boldsymbol{T}_1$, $\boldsymbol{T}_2$ — Upper-triangular
- $\mu(\mathbf{H}_1^{\mathrm{eff}}, \mathbf{H}_2^{\mathrm{eff}})$ — Generalized singular values vector
- If and only if $\mathrm{diag}(\boldsymbol{T}_1)/\mathrm{diag}(\boldsymbol{T}_2) \preceq \mu(\mathbf{H}_1^{\mathrm{eff}}, \mathbf{H}_2^{\mathrm{eff}})$

# Joint Triangularization

## Special case: Joint Equi-Diagonal Triangularization (JET)

- $\mathbf{H}_1^{\text{eff}}$ and $\mathbf{H}_2^{\text{eff}}$ — $N \times N$ matrices
- $\det(\mathbf{H}_1^{\text{eff}}) = \det(\mathbf{H}_2^{\text{eff}})$
- $\mathbf{H}_1^{\text{eff}}$ and $\mathbf{H}_2^{\text{eff}}$ can be jointly decomposed as:

$$\mathbf{H}_1^{\text{eff}} = \mathbf{Q}_1 \boldsymbol{T}_1 \boldsymbol{V}^\dagger$$
$$\mathbf{H}_2^{\text{eff}} = \mathbf{Q}_2 \boldsymbol{T}_2 \boldsymbol{V}^\dagger$$

where

- $\mathbf{Q}_1$, $\mathbf{Q}_2$, $\boldsymbol{V}$ — Unitary
- $\boldsymbol{T}_1$, $\boldsymbol{T}_2$ — Upper-triangular
- $\text{diag}(\boldsymbol{T}_1) = \text{diag}(\boldsymbol{T}_2)$

# Joint Triangularization

## Proof idea

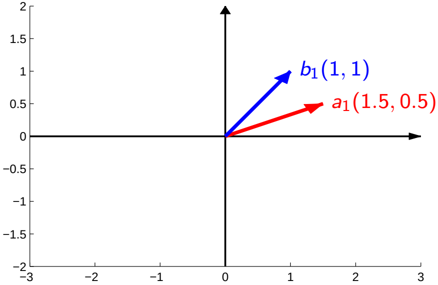
GTD condtion on diagonal $\rightarrow$ condition on ratio of 2 diagonals

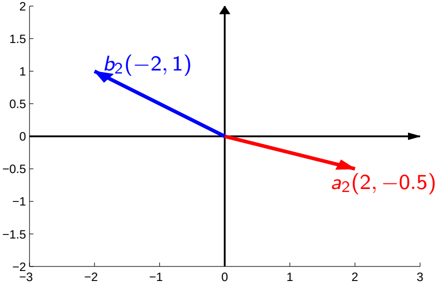
## Block-triangular version

- Generalizes to a block-triangular variant:
  Desired ratios between the block determinants
- Necessary and sufficient conditions

# JET Interpretation

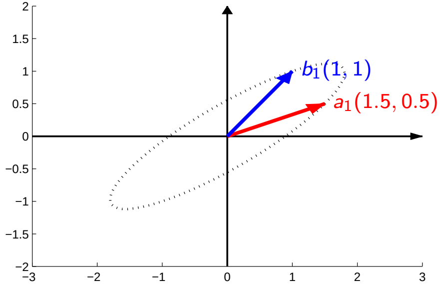
$$H_1 = \begin{bmatrix} 1.5 & 1 \\ 0.5 & 1 \end{bmatrix} \qquad H_2 = \begin{bmatrix} 2 & -2 \\ -0.5 & 1 \end{bmatrix}$$

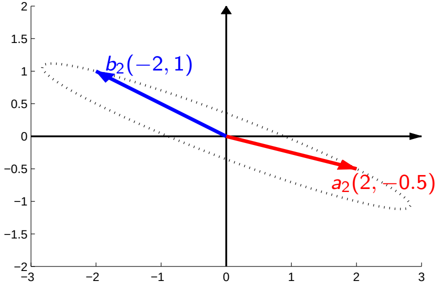


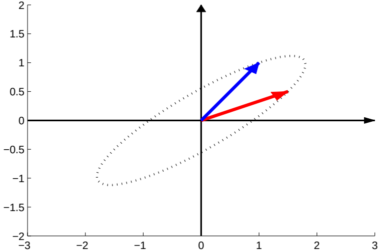
# JET Interpretation

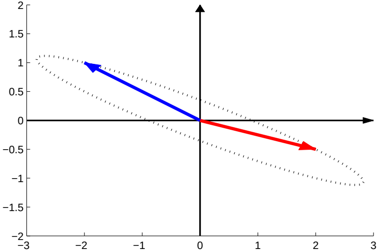
$$H_1 = \begin{bmatrix} 1.5 & 1 \\ 0.5 & 1 \end{bmatrix} \qquad H_2 = \begin{bmatrix} 2 & -2 \\ -0.5 & 1 \end{bmatrix}$$

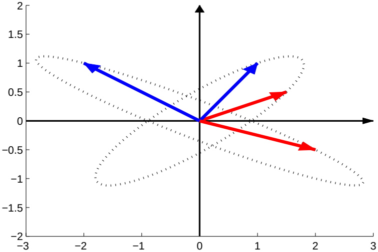


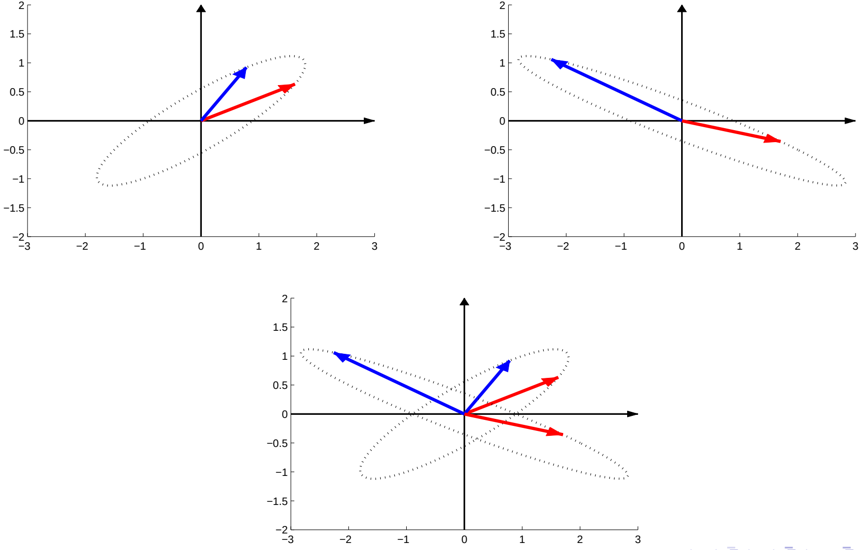
# JET Interpretation



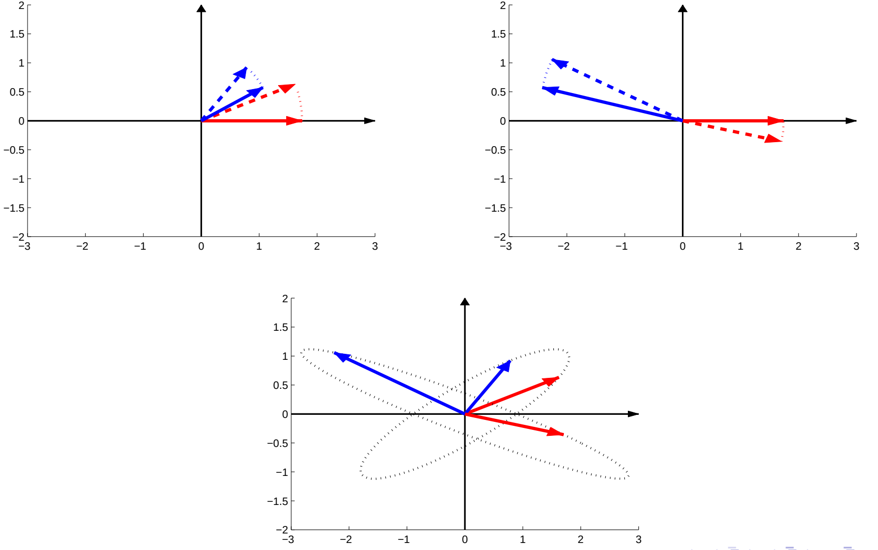



# JET Interpretation



# JET Interpretation



# Bottleneck Problem

$$\mathbf{H}_1^{\text{eff}} = \begin{bmatrix} 2 & 3 \\ 0 & 6 \end{bmatrix} = \overbrace{\begin{bmatrix} 0.238 & 0.971 \\ -0.971 & 0.238 \end{bmatrix}}^{\mathbf{Q}_1} \overbrace{\begin{bmatrix} \color{red}{2.522} & -4.472 \\ 0 & \color{blue}{4.758} \end{bmatrix}}^{\boldsymbol{T}_1} \overbrace{\begin{bmatrix} 0.913 & -0.408 \\ 0.408 & 0.913 \end{bmatrix}}^{\boldsymbol{V}^\dagger}$$

$$\mathbf{H}_2^{\text{eff}} = \begin{bmatrix} 3 & 2 \\ 0 & 4 \end{bmatrix} = \overbrace{\begin{bmatrix} 0.762 & 0.647 \\ -0.647 & 0.762 \end{bmatrix}}^{\mathbf{Q}_1} \overbrace{\begin{bmatrix} \color{red}{2.522} & -0.039 \\ 0 & \color{blue}{4.758} \end{bmatrix}}^{\boldsymbol{T}_2} \overbrace{\begin{bmatrix} 0.913 & -0.408 \\ 0.408 & 0.913 \end{bmatrix}}^{\boldsymbol{V}^\dagger}$$

- $\text{diag}(\boldsymbol{T}_1) = \text{diag}(\boldsymbol{T}_2) = \begin{bmatrix} \color{red}{2.522} & \color{blue}{4.758} \end{bmatrix}$
- $\mathbf{Q}_1^\dagger \mathbf{Q}_1 = \mathbf{Q}_2^\dagger \mathbf{Q}_2 = \boldsymbol{V}^\dagger \boldsymbol{V} = \mathbf{I}_2$

# Degrees-of-Freedom Mismatch Example

Matrix $\boldsymbol{V}$ is applied to $\begin{bmatrix} \mathbf{H}_i \\ \mathbf{I}_N \end{bmatrix}$ (MMSE variant):

$$\begin{bmatrix} \mathbf{H}_1 \\ \mathbf{I}_2 \end{bmatrix} = \begin{bmatrix} 3 & 0 \\ 0 & 3 \\ 1 & 0 \\ 0 & 1 \end{bmatrix} = \overbrace{\begin{bmatrix} 0.286 & -0.905 \\ 0.905 & 0.286 \\ 0.095 & -0.301 \\ 0.301 & 0.095 \end{bmatrix}}^{\mathbf{Q}_1} \overbrace{\begin{bmatrix} \sqrt{10} & 0 \\ 0 & \sqrt{10} \end{bmatrix}}^{\boldsymbol{T}_1} \overbrace{\begin{bmatrix} 0.302 & 0.954 \\ -0.954 & 0.302 \end{bmatrix}}^{\boldsymbol{V}^\dagger}$$

$$\begin{bmatrix} \mathbf{H}_2 \\ \mathbf{I}_2 \end{bmatrix} = \begin{bmatrix} \sqrt{99} & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix} = \overbrace{\begin{bmatrix} 0.949 & -0.300 \\ 0.905 & -0.030 \\ 0.302 & 0.954 \end{bmatrix}}^{\mathbf{Q}_2} \overbrace{\begin{bmatrix} \sqrt{10} & -9 \\ 0 & \sqrt{10} \end{bmatrix}}^{\boldsymbol{T}_2} \overbrace{\begin{bmatrix} 0.302 & 0.954 \\ -0.954 & 0.302 \end{bmatrix}}^{\boldsymbol{V}^\dagger}$$

- $\mathbf{Q}_1^\dagger \mathbf{Q}_1 = \mathbf{Q}_2^\dagger \mathbf{Q}_2 = \boldsymbol{V}^\dagger \boldsymbol{V} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$
- $\mathrm{diag}(\boldsymbol{T}_1) = \mathrm{diag}(\boldsymbol{T}_2) = \begin{bmatrix} \sqrt{10} & \sqrt{10} \end{bmatrix}^T$

$$\Downarrow$$

**Parallel SISO channels with equal gains for both users!**

# Part IV

# Multiple Users

# Multiple Users

## Problem

- We have used $\boldsymbol{V}$ to triangularize two matrices
- **What to do for more??**

**Is 2 just a bit more than 1?**
**Or... Is 2 a simplified $\infty$?**

- How one buys more **degrees of freedom**?
- And at what price?

# Multiple Users

## Problem

- We have used $\boldsymbol{V}$ to triangularize two matrices
- **What to do for more??**

**Is 2 just a bit more than 1?**
**Or... Is 2 a simplified $\infty$?**

- How one buys more **degrees of freedom**?
- And at what price?

**Space–Time Coding to the Rescue!**

# $K$-user JET/GMD via Space–Time Coding

## [Kh.-Hitron-Livni-Erez IT '15]

**Main Idea**

Create more degrees of freedom using **space–time modulation**

- Original channel: $\boldsymbol{y}_i = \mathbf{H}_i \boldsymbol{x} + \boldsymbol{z}_i$

$$\underbrace{\boldsymbol{x}_i^{(1)} | \boldsymbol{x}_i^{(2)} | \cdots | \boldsymbol{x}_i^{(L)}}_{\mathcal{X}} \rightarrow \boxed{\mathbf{H}_i} \rightarrow \overset{\boldsymbol{z}_i^{(j)}}{\underset{\downarrow}{\oplus}} \rightarrow \underbrace{\boldsymbol{y}_i^{(1)} | \boldsymbol{y}_i^{(2)} | \cdots | \boldsymbol{y}_i^{(L)}}_{\mathcal{Y}_i}$$

- Time extended channel: $\mathcal{Y}_i = \mathcal{H}_i \mathcal{X} + \mathcal{Z}_i$
  - $\mathcal{X}, \mathcal{Y}_i, \mathcal{Z}_i$: vectors of length $N \cdot L$
  - $\mathcal{H}_i$: matrix of size $NL \times NL$

# $K$-user JET/GMD via Space–Time Coding

## [Kh.-Hitron-Livni-Erez IT '15]

$$\mathbf{H}_i^{\text{eff}} = \mathbf{Q}_i \boldsymbol{T}_i \boldsymbol{V}^\dagger \qquad \text{X}$$

- **Bunch two channel uses together:**

$$\overbrace{\begin{pmatrix} \mathbf{H}_i^{\text{eff}} & \mathbf{0} \\ \mathbf{0} & \mathbf{H}_i^{\text{eff}} \end{pmatrix}}^{\mathcal{H}_i} = \overbrace{\begin{pmatrix} \mathbf{Q}_i & \mathbf{0} \\ \mathbf{0} & \mathbf{Q}_i \end{pmatrix}}^{\mathcal{Q}_i} \overbrace{\begin{pmatrix} \boldsymbol{T}_i & \mathbf{0} \\ \mathbf{0} & \boldsymbol{T}_i \end{pmatrix}}^{\mathcal{T}_i} \overbrace{\begin{pmatrix} \boldsymbol{V}^\dagger & \mathbf{0} \\ \mathbf{0} & \boldsymbol{V}^\dagger \end{pmatrix}}^{\mathcal{V}} \qquad \text{X}$$

- $\mathcal{H}_i$ have a block-diagonal structure
- Use general $\mathcal{Q}_i$, $\mathcal{V}$ (*not* block-diagonal):

$$\overbrace{\begin{pmatrix} \mathbf{H}_i^{\text{eff}} & \mathbf{0} \\ \mathbf{0} & \mathbf{H}_i^{\text{eff}} \end{pmatrix}}^{\mathcal{H}_i} = \Big(\mathcal{Q}_i\Big) \Big(\mathcal{T}_i\Big) \Big(\mathcal{V}\Big)^\dagger \qquad \checkmark$$

- Exploiting off-diagonal **0**s enables JET/GMD of **more users!**

# Multiple Users: $K$-User JET

- $K$-GMD $\Leftrightarrow$ $(K+1)$-JET
- But $K$-GMD for $K > 1$ is not possible in general ☹

**2-GMD for $2 \times 2$ matrices [Kh.-Hitron-Livni-Erez IT '15]**

2-GMD of the $2 \times 2$ matrices $H_1$ and $H_2$ is possible if and only if

$$F\left(\mathbf{H}_1^{\mathrm{eff}\,\dagger}\mathbf{H}_1^{\mathrm{eff}} - \mathbf{I}, \mathbf{H}_2^{\mathrm{eff}\,\dagger}\mathbf{H}_2^{\mathrm{eff}} - \mathbf{I}\right) \geq 0$$

$$F(\mathbf{A}_1, \mathbf{A}_2) \triangleq |\mathbf{A}_1 \mathrm{adj}(\mathbf{A}_2) - \mathbf{A}_2 \mathrm{adj}(\mathbf{A}_1)|$$

**Another special case**

Diagonal permuted matrices [Presented in the sequel]

# Space–Time Coding Structure

## Theorem: $K$-GMD [Kh.-Hitron-Livni-Erez IT '15]

- Any number of users $K$
- Any number of antennas at each node
- Joint **constant-diagonal** triangularization of $K$ matrices
- Process jointly #symbols $\geq N^{K-1}$
- Prefix–suffix loss of $(N^{K-1} - 1)$ scalar code entries total
- **Numerical evidence**: Can be improved!

## $K$-JET

For joint **equal-diagonal** (~~**constant**~~) triangularization:

- Process jointly #symbols $\geq N^{K-2}$
- Prefix–suffix loss of $(N^{K-2} - 1)$ symbols total

# Demonstration of 3-JET for $2 \times 2$ Matrices

**Step 1:** Construct time-extended matrices

$$\mathcal{H}_1 = \begin{pmatrix} \mathbf{H}_1^{\text{eff}} & 0 & 0 & 0 \\ 0 & \mathbf{H}_1^{\text{eff}} & 0 & 0 \\ 0 & 0 & \mathbf{H}_1^{\text{eff}} & 0 \\ 0 & 0 & 0 & \mathbf{H}_1^{\text{eff}} \end{pmatrix} \qquad \mathcal{H}_2 = \begin{pmatrix} \mathbf{H}_2^{\text{eff}} & 0 & 0 & 0 \\ 0 & \mathbf{H}_2^{\text{eff}} & 0 & 0 \\ 0 & 0 & \mathbf{H}_2^{\text{eff}} & 0 \\ 0 & 0 & 0 & \mathbf{H}_2^{\text{eff}} \end{pmatrix}$$

$$\mathcal{H}_3 = \begin{pmatrix} \mathbf{H}_3^{\text{eff}} & 0 & 0 & 0 \\ 0 & \mathbf{H}_3^{\text{eff}} & 0 & 0 \\ 0 & 0 & \mathbf{H}_3^{\text{eff}} & 0 \\ 0 & 0 & 0 & \mathbf{H}_3^{\text{eff}} \end{pmatrix}$$

$$\begin{pmatrix} * & * & * & * & * & * & * & * \\ 0 & t_1 & * & * & * & * & * & * \\ 0 & 0 & t_2 & * & * & * & * & * \\ 0 & 0 & 0 & t_1 & * & * & * & * \\ 0 & 0 & 0 & 0 & t_2 & * & * & * \\ 0 & 0 & 0 & 0 & 0 & t_1 & * & * \\ 0 & 0 & 0 & 0 & 0 & 0 & t_2 & * \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & * \end{pmatrix}$$

# Demonstration of 3-JET for $2 \times 2$ Matrices

Step 2: blockwise JET for $H_1$ and $H_2$

$$\begin{pmatrix} r_1 & * & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & {\color{red}r_2} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & {\color{red}r_1} & * & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & {\color{red}r_2} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & {\color{red}r_1} & * & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & {\color{red}r_2} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & {\color{red}r_1} & * \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & r_2 \end{pmatrix} \quad \begin{pmatrix} r_1 & * & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & {\color{red}r_2} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & {\color{red}r_1} & * & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & {\color{red}r_2} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & {\color{red}0} & {\color{red}r_1} & * & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & {\color{red}r_2} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & {\color{red}r_1} & * \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & r_2 \end{pmatrix}$$

$$\begin{pmatrix} s_1 & * & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & {\color{red}s_2} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & {\color{red}s_1} & * & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & {\color{red}s_2} & {\color{red}0} & 0 & 0 & 0 \\ 0 & 0 & 0 & {\color{red}0} & {\color{red}s_1} & * & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & {\color{red}s_2} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & {\color{red}s_1} & * \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & s_2 \end{pmatrix}$$

# Demonstration of 3-JET for 2 × 2 Matrices

Step 2: “off-by-one” blockwise JET

$$\begin{pmatrix} r_1 & * & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & r_2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & r_1 & * & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & r_2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & r_1 & * & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & r_2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & r_1 & * \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & r_2 \end{pmatrix} \quad \begin{pmatrix} r_1 & * & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & r_2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & r_1 & * & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & r_2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & r_1 & * & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & r_2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & r_1 & * \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & r_2 \end{pmatrix}$$

$$\begin{pmatrix} s_1 & * & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & s_2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & s_1 & * & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & s_2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & s_1 & * & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & s_2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & s_1 & * \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & s_2 \end{pmatrix}$$

# Demonstration of 3-JET for $2 \times 2$ Matrices

Step 2: "off-by-one" blockwise JET

$$\begin{pmatrix} r_1 & * & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & t_2 & * & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & t_1 & * & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & t_2 & * & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & t_1 & * & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & t_2 & * & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & t_1 & * \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & r_2 \end{pmatrix} \quad \begin{pmatrix} r_1 & * & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & t_2 & * & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & t_1 & * & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & t_2 & * & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & t_1 & * & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & t_2 & * & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & t_1 & * \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & r_2 \end{pmatrix}$$

$$\begin{pmatrix} s_1 & * & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & t_2 & * & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & t_1 & * & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & t_2 & * & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & t_1 & * & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & t_2 & * & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & t_1 & * \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & s_2 \end{pmatrix}$$

# Demonstration of 3-JET for 2 × 2 Matrices

Step 4: Extract middle matrices using $\mathcal{O}$

$$\begin{pmatrix} \cancel{r_1} & * & * & * & * & * & * & * \\ 0 & t_1 & * & * & * & * & * & * \\ 0 & 0 & t_2 & * & * & * & * & * \\ 0 & 0 & 0 & t_1 & * & * & * & * \\ 0 & 0 & 0 & 0 & t_2 & * & * & * \\ 0 & 0 & 0 & 0 & 0 & t_1 & * & * \\ 0 & 0 & 0 & 0 & 0 & 0 & t_2 & * \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \cancel{r_2} \end{pmatrix} \qquad \begin{pmatrix} \cancel{r_1} & * & * & * & * & * & * & * \\ 0 & t_1 & * & * & * & * & * & * \\ 0 & 0 & t_2 & * & * & * & * & * \\ 0 & 0 & 0 & t_1 & * & * & * & * \\ 0 & 0 & 0 & 0 & t_2 & * & * & * \\ 0 & 0 & 0 & 0 & 0 & t_1 & * & * \\ 0 & 0 & 0 & 0 & 0 & 0 & t_2 & * \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \cancel{r_2} \end{pmatrix}$$

$$\begin{pmatrix} \cancel{s_1} & * & * & * & * & * & * & * \\ 0 & t_1 & * & * & * & * & * & * \\ 0 & 0 & t_2 & * & * & * & * & * \\ 0 & 0 & 0 & t_1 & * & * & * & * \\ 0 & 0 & 0 & 0 & t_2 & * & * & * \\ 0 & 0 & 0 & 0 & 0 & t_1 & * & * \\ 0 & 0 & 0 & 0 & 0 & 0 & t_2 & * \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \cancel{s_2} \end{pmatrix}$$

$$\mathcal{O}^\dagger = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix}$$

Extract: $\mathcal{O}^\dagger T_i \mathcal{O}$

# Demonstration of 3-GMD for 2 × 2 Matrices

**Step 1:** Construct time-extended matrices

$$\mathcal{H}_1 = \begin{pmatrix} \mathbf{H}_1^{\text{eff}} & 0 & 0 & 0 \\ 0 & \mathbf{H}_1^{\text{eff}} & 0 & 0 \\ 0 & 0 & \mathbf{H}_1^{\text{eff}} & 0 \\ 0 & 0 & 0 & \mathbf{H}_1^{\text{eff}} \end{pmatrix} \qquad \mathcal{H}_2 = \begin{pmatrix} \mathbf{H}_2^{\text{eff}} & 0 & 0 & 0 \\ 0 & \mathbf{H}_2^{\text{eff}} & 0 & 0 \\ 0 & 0 & \mathbf{H}_2^{\text{eff}} & 0 \\ 0 & 0 & 0 & \mathbf{H}_2^{\text{eff}} \end{pmatrix}$$

$$\mathcal{H}_3 = \begin{pmatrix} \mathbf{H}_3^{\text{eff}} & 0 & 0 & 0 \\ 0 & \mathbf{H}_3^{\text{eff}} & 0 & 0 \\ 0 & 0 & \mathbf{H}_3^{\text{eff}} & 0 \\ 0 & 0 & 0 & \mathbf{H}_3^{\text{eff}} \end{pmatrix}$$

$$\begin{pmatrix} * & * & * & * & * & * & * & * \\ 0 & * & * & * & * & * & * & * \\ 0 & 0 & * & * & * & * & * & * \\ 0 & 0 & 0 & \boxed{1} & * & * & * & * \\ 0 & 0 & 0 & 0 & 1 & * & * & * \\ 0 & 0 & 0 & 0 & 0 & * & * & * \\ 0 & 0 & 0 & 0 & 0 & 0 & * & * \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & * \end{pmatrix}$$

# Demonstration of 3-GMD for $2 \times 2$ Matrices

Step 2: blockwise GMD for $H_1$

$$
\begin{pmatrix}
1 & * & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & * & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & * & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & * \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1
\end{pmatrix}
\quad
\begin{pmatrix}
r_1^2 & * & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & r_2^2 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & r_1^2 & * & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & r_2^2 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & r_1^2 & * & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & r_2^2 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & r_1^2 & * \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & r_2^2
\end{pmatrix}
$$

$$
\begin{pmatrix}
r_1^3 & * & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & r_2^3 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & r_1^3 & * & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & r_2^3 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & r_1^3 & * & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & r_2^3 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & r_1^3 & * \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & r_2^3
\end{pmatrix}
$$

# Demonstration of 3-GMD for $2 \times 2$ Matrices

Step 3: Perform GMD on ■ in $\mathcal{H}_2$

$$\begin{pmatrix} 1 & * & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & * & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & * & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & * \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \quad \begin{pmatrix} r_1^2 & * & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & r_2^2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & r_1^2 & * & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & r_2^2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & r_1^2 & * & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & r_2^2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & r_1^2 & * \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & r_2^2 \end{pmatrix}$$

$$\begin{pmatrix} r_1^3 & * & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & r_2^3 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & r_1^3 & * & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & r_2^3 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & r_1^3 & * & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & r_2^3 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & r_1^3 & * \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & r_2^3 \end{pmatrix} \qquad \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} = V^\dagger \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} V$$

# Demonstration of 3-GMD for $2 \times 2$ Matrices

Step 4: Perform the same GMD on [yellow] in $\mathcal{H}_2$

$$\begin{pmatrix} 1 & * & 0 & 0 & * & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & * & 0 & 0 \\ 0 & 0 & 1 & * & 0 & 0 & * & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & * \\ 0 & 0 & 0 & 0 & 1 & * & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & * \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \quad \begin{pmatrix} r_1^2 & * & 0 & 0 & * & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & * & * & 0 & 0 \\ 0 & 0 & r_1^2 & * & 0 & 0 & * & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & * & * \\ 0 & 0 & 0 & 0 & 1 & * & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & r_2^2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & * \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & r_2^2 \end{pmatrix}$$

$$\begin{pmatrix} r_1^3 & * & 0 & 0 & * & 0 & 0 & 0 \\ 0 & d_2 & 0 & 0 & * & * & 0 & 0 \\ 0 & 0 & r_1^3 & * & 0 & 0 & * & 0 \\ 0 & 0 & 0 & d_2 & 0 & 0 & * & * \\ 0 & 0 & 0 & 0 & d_1 & * & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & r_2^3 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & d_1 & * \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & r_2^3 \end{pmatrix}$$

# Demonstration of 3-GMD for $2 \times 2$ Matrices

Step 5: Perform GMD on ▇ in $\mathcal{H}_3$

$$\begin{pmatrix} 1 & * & 0 & 0 & * & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & * & 0 & 0 \\ 0 & 0 & 1 & * & 0 & 0 & * & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & * \\ 0 & 0 & 0 & 0 & 1 & * & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & * \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \quad \begin{pmatrix} r_1^2 & * & 0 & 0 & * & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & * & * & 0 & 0 \\ 0 & 0 & r_1^2 & * & 0 & 0 & * & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & * & * \\ 0 & 0 & 0 & 0 & 1 & * & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & r_2^2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & * \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & r_2^2 \end{pmatrix}$$

$$\begin{pmatrix} r_1^3 & * & 0 & 0 & * & 0 & 0 & 0 \\ 0 & d_2 & 0 & 0 & * & * & 0 & 0 \\ 0 & 0 & r_1^3 & * & 0 & 0 & * & 0 \\ 0 & 0 & 0 & d_2 & 0 & 0 & * & * \\ 0 & 0 & 0 & 0 & d_1 & * & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & r_2^3 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & d_1 & * \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & r_2^3 \end{pmatrix}$$

# Demonstration of 3-GMD for $2 \times 2$ Matrices

Step 6: Extract middle matrices using $\mathcal{O}$

$$\begin{pmatrix} \cancel{1} & * & 0 & * & * & 0 & 0 & 0 \\ 0 & \cancel{1} & 0 & 0 & 0 & * & 0 & 0 \\ 0 & 0 & \cancel{1} & * & * & 0 & * & 0 \\ 0 & 0 & 0 & 1 & 0 & * & 0 & * \\ 0 & 0 & 0 & 0 & 1 & * & 0 & * \\ 0 & 0 & 0 & 0 & 0 & \cancel{1} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \cancel{1} & * \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \cancel{1} \end{pmatrix}$$

$$\begin{pmatrix} \cancel{r_1^2} & * & 0 & * & * & 0 & 0 & 0 \\ 0 & \cancel{1} & 0 & * & * & * & 0 & 0 \\ 0 & 0 & \cancel{r_1^2} & * & * & 0 & * & 0 \\ 0 & 0 & 0 & 1 & 0 & * & * & * \\ 0 & 0 & 0 & 0 & 1 & * & * & * \\ 0 & 0 & 0 & 0 & 0 & \cancel{r_2^2} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \cancel{1} & * \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \cancel{r_2^2} \end{pmatrix}$$

$$\begin{pmatrix} \cancel{r_1^3} & * & 0 & * & * & 0 & 0 & 0 \\ 0 & \cancel{d_2} & 0 & * & * & * & 0 & 0 \\ 0 & 0 & \cancel{r_1^3} & * & * & 0 & * & 0 \\ 0 & 0 & 0 & 1 & * & * & * & * \\ 0 & 0 & 0 & 0 & 1 & * & * & * \\ 0 & 0 & 0 & 0 & 0 & \cancel{r_2^3} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \cancel{d_1} & * \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \cancel{r_2^3} \end{pmatrix}$$

$$\mathcal{O}^\dagger = \begin{pmatrix} 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \end{pmatrix}$$

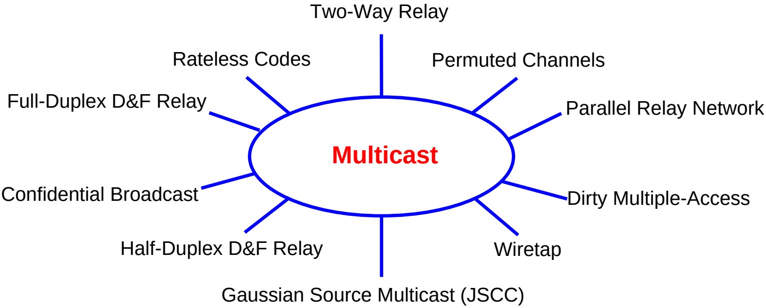
$$\mathcal{O}^\dagger T_i^{(3)} \mathcal{O} = \begin{pmatrix} 1 & * \\ 0 & 1 \end{pmatrix}$$

approach capacity when $L \to \infty$

# Part V

# Applications

# Summary: Multicast is (Almost) Everywhere...



# Gaussian Rateless (Incremental Redundancy) Coding

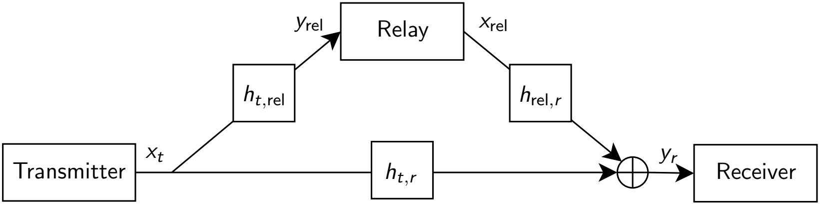
$$y = \alpha x + z\,,$$

- $\alpha$ is **unknown at Tx** but is **known at Rx**
- Rx sends NACKs/ACKs until it is able to recover the message
- Assume $\alpha$ can take only a finite number of values: $\alpha_1, \alpha_2, \dots$
- Can be represented as a MIMO multicast problem [Kh.-Kochman-Erez-Wornell ITW'11]

### Example $\alpha \in \{\alpha_1, \alpha_2\}$, $\alpha_1 > \alpha_2$

- $C_1 = 2C_2$
- Effective matrices: $\mathbf{H}_1 = \begin{pmatrix} \alpha_1 & 0 \end{pmatrix}, \mathbf{H}_2 = \begin{pmatrix} \alpha_2 & 0 \\ 0 & \alpha_2 \end{pmatrix}$
- Coincides with the solution of [Erez-Trott-Wornell IT'12]
- Works for MIMO channels $\mathbf{H}_1, \mathbf{H}_2$ (replacing $\alpha_1, \alpha_2$)

# Half-Duplex Relay

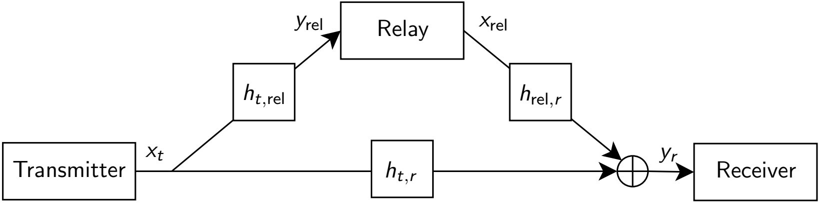



- **Half-duplex:** Relay can receive or transmit but **not both**
- Decode-and-forward implementation: "**rateless relay**"

**Effective Matrices [Kh.-Kochman-Erez-Wornell ITW'11]**

$$\mathcal{H}_{\text{rel}} = \begin{bmatrix} \sqrt{P_1} h_{t,\text{rel}} & 0 \end{bmatrix}, \mathcal{H}_r = \begin{bmatrix} \sqrt{P_1} h_{t,r} & 0 \\ 0 & \sqrt{P_2} h_{t,r} + \sqrt{P_{\text{rel}}} h_{\text{rel},r} \end{bmatrix}$$

# Full-Duplex Relay

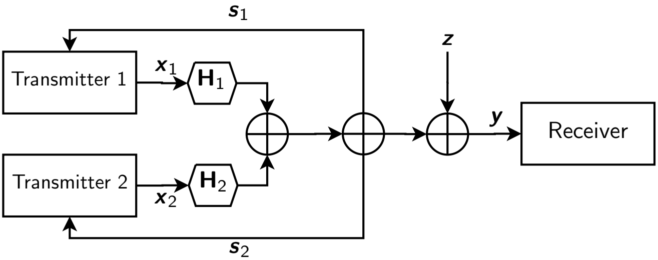



- **Full-duplex:** Relay can receive and transmit *simultaneously*
- Decode-and-forward implementation (previous works): *Special* code constructions.
- But... "Off-the-shelf" codes suffice!

## Effective Matrices [Kh.-Ordentlich-Erez-Kochman-Wornell ITW'12]

$$\mathcal{H}_{\text{rel}} = \sqrt{2} \begin{pmatrix} \sqrt{1-\rho^2} h_{t,\text{rel}} & 0 \end{pmatrix} , \qquad \mathcal{H}_r = \sqrt{2} \begin{pmatrix} \sqrt{1-\rho^2} h_{t,r} & 0 \\ 0 & \frac{\rho h_{t,r} + h_{\text{rel},r}}{\sqrt{\left((1-\rho^2\right) h_{t,r}^2 P + 1}} \end{pmatrix}$$

# Dirty MIMO Multiple-Access Channel (New Achievable)




## SISO capacity region at high SNR [Philosof-Erez-Zamir-Khisti IT'11]

$$R_1 + R_2 \leq \log \min \left\{ |h_1|^2 , |h_2|^2 \right\}$$

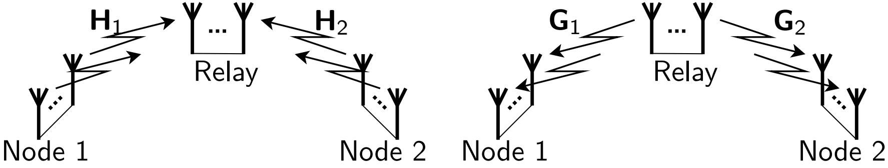
- Sum capacity limited by minimum of individual capacity
- Best for balanced powers!

## MIMO capacity region at high SNR

$$R_1 + R_2 \leq \log \min \left\{ |H_1|^2 , |H_2|^2 \right\}$$

# MIMO Two-Way Relay (New Achievable) [Kh.-Kochman-Erez ISIT'11]

- Two nodes want to exchange messages via a relay




(a) MAC Phase (b) Broadcast Phase

## MAC Phase

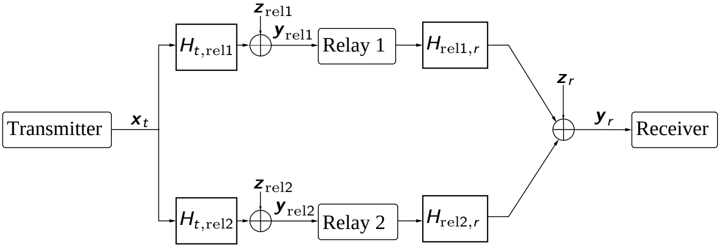
- Apply JET to $\mathbf{H}_1$ and $\mathbf{H}_2$ (roles of $\boldsymbol{V}$ and $\mathbf{Q}$ switched)
- Use dirty-paper coding to pre-cancel off-diagonal elements (Replaces successive interference cancellation of broadcast)

## Broadcast (Multicast!) Phase

- Use proposed multicast scheme

# Parallel MIMO Relay Network

- Tx conveys message to Rx via parallel relays




## Decode-and-Forward

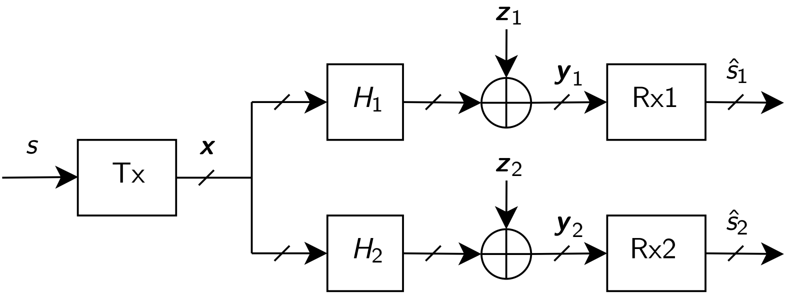
- BC (multicast!) phase: Use proposed multicast scheme
- MAC phase: Equivalent to MIMO-P2P with individual power constraints

## Decode-and-Forward + Amplify-and-Forward

Can be constructed for specific cases (under generalized Weyl's condition)

# MIMO Multicast of a Gaussian Source (New Achievable)

[Kh.-Kochman-Erez SP'12]




- $s$ – Scalar white Gaussian source of power $P_S$.
- Separation does not hold!
- Different triangularization is needed
- Combine with hybrid digital–analog scheme

# MIMO Multicast of a Gaussian Source (New Achievable)

[Kh.-Kochman-Erez SP'12]

## Hybrid digital–analog scheme

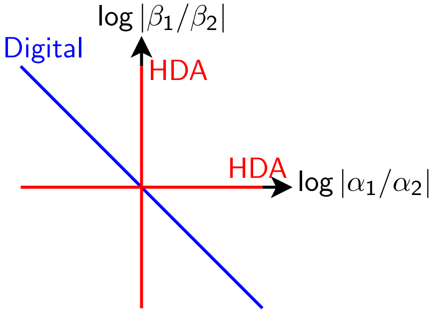
- $(N_t - 1)$ sub-channels with **equal** diagonal values: Transmit digital message = quantized source
- Last gain differs: Transmit analog quantization error

- Decomposition possible under a "generalized Weyl condition"
- When decomposition is possible: **New achievable distortion!**
- For 2 transmit-antennas: **Optimum performance!**

# MIMO Multicast of a Gaussian Source (New Achievable) [Kh.-Kochman-Erez SP'12]

**Example: $2 \times 2$ diagonal channels**

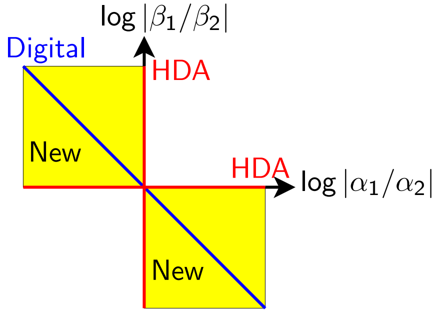
$$\mathbf{H}_1^{\text{eff}} = \begin{bmatrix} \alpha_1 & 0 \\ 0 & \beta_1 \end{bmatrix}, \qquad \mathbf{H}_2^{\text{eff}} = \begin{bmatrix} \alpha_2 & 0 \\ 0 & \beta_2 \end{bmatrix}$$



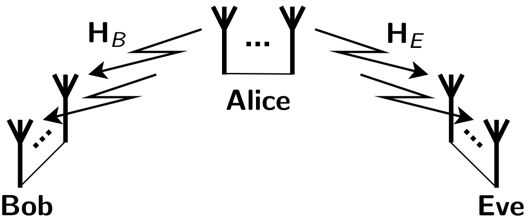
# MIMO Multicast of a Gaussian Source (New Achievable) [Kh.-Kochman-Erez SP'12]

## Example: $2 \times 2$ diagonal channels

$$\mathbf{H}_1^{\text{eff}} = \begin{bmatrix} \alpha_1 & 0 \\ 0 & \beta_1 \end{bmatrix}, \qquad \mathbf{H}_2^{\text{eff}} = \begin{bmatrix} \alpha_2 & 0 \\ 0 & \beta_2 \end{bmatrix}$$



# Channel Model: Gaussian MIMO Wiretap Channel




$$\boldsymbol{y}_B = \mathbf{H}_B \boldsymbol{x} + \boldsymbol{z}_B \qquad \boldsymbol{y}_E = \mathbf{H}_E \boldsymbol{x} + \boldsymbol{z}_E$$

- $\boldsymbol{x}$ – $N \times 1$ input vector of power $P$
- $\boldsymbol{y}_B$, $\boldsymbol{y}_E$ – $N_B \times 1$, $N_E \times 1$ received vectors
- $\mathbf{H}_B$, $\mathbf{H}_E$ – $N_B \times N$, $N_E \times N$ channel matrices
- $\boldsymbol{z}_B \sim \mathcal{CN}(\mathbf{0}, \mathbf{I}_{N_B})$, $\boldsymbol{z}_E \sim \mathcal{CN}(\mathbf{0}, \mathbf{I}_{N_E})$ – noise vectors
- “Closed loop” (full channel knowledge everywhere)

# Capacity

## Gaussian SISO channel capacity [Leung-Yan-Cheong, Hellman '78]

$$C_S\left(h_B, h_E\right) = \Big[\overbrace{\log\left(1 + \left|h_B\right|^2 P\right)}^{I(X;Y_B)} - \overbrace{\log\left(1 + \left|h_E\right|^2 P\right)}^{I(X;Y_E)}\Big]_+$$

## Gaussian MIMO channel capacity [Khisti,Wornell '10][Oggier,Hassibi '11]

$$C_S\left(\mathbf{H}_B, \mathbf{H}_E\right) = \max_{C_{\boldsymbol{x}}:} \Big[\overbrace{\log\left|\mathbf{I} + \mathbf{H}_B C_{\boldsymbol{x}} \mathbf{H}_B^\dagger\right|}^{I(\boldsymbol{x};\boldsymbol{y}_B)} - \overbrace{\log\left|\mathbf{I} + \mathbf{H}_E C_{\boldsymbol{x}} \mathbf{H}_E^\dagger\right|}^{I(\boldsymbol{x};\boldsymbol{y}_E)}\Big]$$

- Maximization over $C_{\boldsymbol{x}}$ satisfying power constraint: $\mathrm{tr}\left\{C_{\boldsymbol{x}}\right\} \leq P$
- Power constraint can be replaced with covariance constraint [Liu-Shamai '09]

# Scheme for General SNR [Kh.-Kochman-Khisti ISIT'14]

$$\begin{bmatrix} \mathbf{H}_B C_{\mathbf{x}}^{1/2} \boldsymbol{V}_A \\ \mathbf{I}_N \end{bmatrix} = \mathbf{Q}_B \overbrace{\begin{pmatrix} b_1 & * & * \\ 0 & \ddots & * \\ 0 & 0 & b_N \end{pmatrix}}^{\boldsymbol{T}_B}, \quad b_i^2 = 1 + \mathrm{SNR}_i^B$$

$$\begin{bmatrix} \mathbf{H}_C C_{\mathbf{x}}^{1/2} \boldsymbol{V}_A \\ \mathbf{I}_N \end{bmatrix} = \mathbf{Q}_C \overbrace{\begin{pmatrix} e_1 & * & * \\ 0 & \ddots & * \\ 0 & 0 & e_N \end{pmatrix}}^{\boldsymbol{T}_C}, \quad e_i^2 = 1 + \mathrm{SNR}_i^E$$

- Use good SISO wiretap codes for SNR-pairs $(b_i^2 - 1, e_i^2 - 1)$
- $\boldsymbol{V}_A$ of Charlie's SVD $\Rightarrow$ Easy secrecy analysis + strong secrecy
- $\boldsymbol{V}_A$ of Bob's SVD $\Rightarrow$ No need for V-BLAST
- $\mathrm{diag}\{\boldsymbol{T}_B\}, \mathrm{diag}\{\boldsymbol{T}_E\}$ are const. $\Rightarrow$ Same code over all channels

# Scheme for General SNR [Kh.-Kochman-Khisti ISIT'14]

$$\begin{bmatrix} \mathbf{H}_B C_{\mathbf{x}}^{1/2} \boldsymbol{V}_A \\ \mathbf{I}_N \end{bmatrix} = \mathbf{Q}_B \overbrace{\begin{pmatrix} b_1 & * & * \\ 0 & \ddots & * \\ 0 & 0 & b_N \end{pmatrix}}^{\boldsymbol{T}_B}, \qquad b_i^2 = 1 + \mathrm{SNR}_i^B$$

$$\begin{bmatrix} \mathbf{H}_C C_{\mathbf{x}}^{1/2} \boldsymbol{V}_A \\ \mathbf{I}_N \end{bmatrix} = \mathbf{Q}_C \overbrace{\begin{pmatrix} e_1 & * & * \\ 0 & \ddots & * \\ 0 & 0 & e_N \end{pmatrix}}^{\boldsymbol{T}_C}, \qquad e_i^2 = 1 + \mathrm{SNR}_i^E$$

- Use good SISO wiretap codes for SNR-pairs $(b_i^2 - 1, e_i^2 - 1)$

## Genie-aided secrecy-proof

- Charlie tries to recover messages sequentially (from last to first)
- For the recovery of message $i$ all previous messages are revealed

# Wiretap Capacity under an Input Covariance Constraint

- $\mathbf{C_x} \preceq \bar{\mathbf{C}}_{\mathbf{x}}$

**Theorem [Bustin-Liu-Poor-Shamai '09]**

Let $\mu_i(\mathbf{H}_B, \mathbf{H}_C, \bar{\mathbf{C}}_{\mathbf{x}})$ be the GSVs of $G(\mathbf{H}_B, \bar{\mathbf{C}}_{\mathbf{x}})$, $G(\mathbf{H}_C, \bar{\mathbf{C}}_{\mathbf{x}})$. Then,

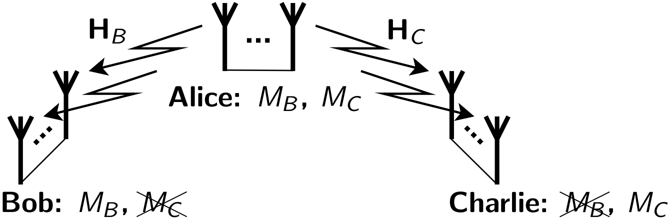
$$C(\mathbf{H}_B, \mathbf{H}_C, \bar{\mathbf{C}}_{\mathbf{x}}) = \sum_{i=1}^{N_A} [\log \mu_i^2(\mathbf{H}_B, \mathbf{H}_C, \bar{\mathbf{C}}_{\mathbf{x}})]^+$$

- Proof in [Bustin *et al.* '09] uses heavy tools such as channel enhancement and I-MMSE connection

**Alternative simple proof [Kh.-Kochman-Khisti, submitted ISIT'15]**

1. The GSVD majorizes all other joint triangularizations
2. Apply GSVD and take all GSVs $> 1$

# Model: Confidential Gaussian MIMO Broadcast




$$\boldsymbol{y}_B = \mathbf{H}_B \boldsymbol{x}_A + \boldsymbol{z}_B \qquad \boldsymbol{y}_C = \mathbf{H}_C \boldsymbol{x}_A + \boldsymbol{z}_C$$

- $M_B$ – message intended for Bob
  kept secret from Charlie
- $M_C$ – message intended for Charlie
  kept secret from Bob

# Capacity-Achieving Confidential MIMO Broadcast

**Covariance constraint [Liu-Liu-Poor-Shamai IT'10]**

- No tension between users
- Both users achieve optimal wiretap capacities simultaneously!

- Again, proof uses heavy machinery...

$$\begin{bmatrix} \mathbf{H}_B \bar{\mathbf{C}}_{\boldsymbol{x}}^{1/2} \\ \mathbf{I}_N \end{bmatrix} = \mathbf{Q}_B \begin{pmatrix} b_1 & * & * \\ 0 & \ddots & * \\ 0 & 0 & b_N \end{pmatrix} \boldsymbol{V}_A^\dagger , \quad \begin{bmatrix} \mathbf{H}_C \bar{\mathbf{C}}_{\boldsymbol{x}}^{1/2} \\ \mathbf{I}_N \end{bmatrix} = \mathbf{Q}_C \begin{pmatrix} c_1 & * & * \\ 0 & \ddots & * \\ 0 & 0 & c_N \end{pmatrix} \boldsymbol{V}_A^\dagger$$

- Choosing directions of $b_i > c_i$ is **optimal for Bob**
- **But...** Choosing directions of $b_i < c_i$ is **optimal for Charlie!**

$\Rightarrow$ **Allocate $b_i > c_i$ to Bob**
**Allocate $b_i < c_i$ to Charlie**

# Capacity-Achieving Confidential MIMO Broadcast

**Covariance constraint [Liu-Liu-Poor-Shamai IT'10]**

- No tension between users
- Both users achieve optimal wiretap capacities simultaneously!

- Again, proof uses heavy machinery...

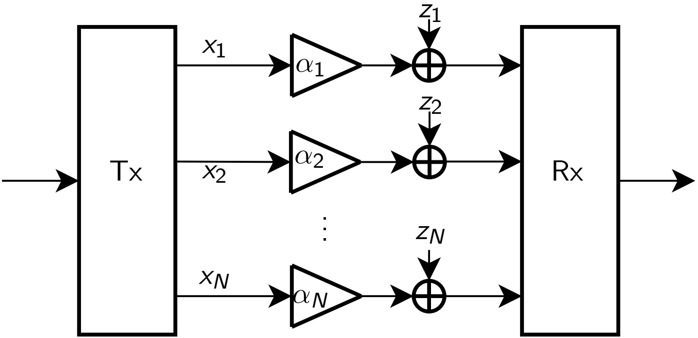
$$\begin{bmatrix} \mathbf{H}_B \bar{\mathbf{C}}_{\boldsymbol{x}}^{1/2} \\ \mathbf{I}_N \end{bmatrix} = \mathbf{Q}_B \begin{pmatrix} b_1 & * & * \\ 0 & \ddots & * \\ 0 & 0 & b_N \end{pmatrix} \boldsymbol{V}_A^{\dagger}, \quad \begin{bmatrix} \mathbf{H}_C \bar{\mathbf{C}}_{\boldsymbol{x}}^{1/2} \\ \mathbf{I}_N \end{bmatrix} = \mathbf{Q}_C \begin{pmatrix} c_1 & * & * \\ 0 & \ddots & * \\ 0 & 0 & c_N \end{pmatrix} \boldsymbol{V}_A^{\dagger}$$

**Alternative simple proof [Kh.-Kochman-Khisti, submitted ISIT'15]**

- Apply GSVD
- Send information to Bob over sub-channels with $b_i > c_i$
- Send information to Charlie over sub-channels with $c_i > b_i$

# Gaussian Permuted Parallel Channels

- General channels: [Willems, Gorokhov IT'08][Hof, Sason, Shamai ITW'10]




- Gains $\{\alpha_i\}$ are known
- **Order** of gains is **not known at Tx**, but **known at Rx**

**Equivalent Problem**

Be optimal for all permutation-orders simultaneously.

# Gaussian Permuted Parallel Channels

**Special case of MIMO multicast problem!**

**$N$! effective channel matrices:**

$$\mathbf{H}_i \triangleq \begin{pmatrix} \alpha_{\pi_i(1)} & 0 & \cdots & 0 \\ 0 & \alpha_{\pi_i(2)} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \alpha_{\pi_i(N)} \end{pmatrix}, \qquad \begin{aligned} &\boldsymbol{\pi}_i \in S_N \\ &i = 1, \ldots, N! \end{aligned}$$

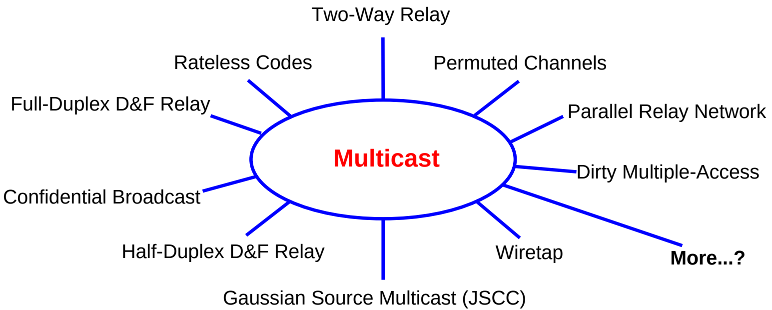
## Optimal precoding matrices [Hitron-Kh.-Erez ISIT'12]

- **2 gains:** Hadamard/DFT; *1 real channel use*
- **3 gains:** DFT; *1 complex channel use* $\Rightarrow$ *2 real uses*
- **4 gains:** Quaternion-based matrix; *1 quater.* $\Rightarrow$ *2 complex uses*
- $N > 4$ **gain:** • • • **?**

# Part VI

# Summary

# Summary: Multicast is (Almost) Everywhere...




**Even now, me talking to you...**

# Part VII

# Supplementary

# Wiretap under Cov. Constraint: Alternative Proof Outline

- W.l.o.g., $\mathbf{C}_{\boldsymbol{x}} \preceq \bar{\mathbf{C}}_{\boldsymbol{x}}$ can be written as

$$\mathbf{C}_{\boldsymbol{x}} = \bar{\mathbf{C}}_{\boldsymbol{x}}^{1/2} \boldsymbol{V}_A \mathbf{D} \boldsymbol{V}_A^\dagger \bar{\mathbf{C}}_{\boldsymbol{x}}^{\dagger/2}$$

  where $\mathbf{D}$ is non-negative diagonal with all elements $\leq 1$
- For any $\boldsymbol{V}_A$,

$$I(\mathbf{H}_B, \mathbf{C}_{\boldsymbol{x}}) - I(\mathbf{H}_C, \mathbf{C}_{\boldsymbol{x}}) = \sum_{i=1}^{N} \log \frac{b_i^2}{c_i^2}$$

- Optimal $\mathbf{D}$ for a given $\boldsymbol{V}_A$: truncation

$$C_B(\mathbf{H}_B, \mathbf{H}_C, \bar{\mathbf{C}}_{\boldsymbol{x}}) = \max_{\boldsymbol{V}_A} \sum_{i=1}^{N} \left[ \log \frac{b_i^2}{c_i^2} \right]^+$$

- By multiplicative majorization of joint triangularization [Khina, Kochman, Erez SP'12], $\boldsymbol{V}_A$ of the GSVD is optimal