Compute-and-Forward:

An Explicit Link between Finite Field and Gaussian Interference Networks

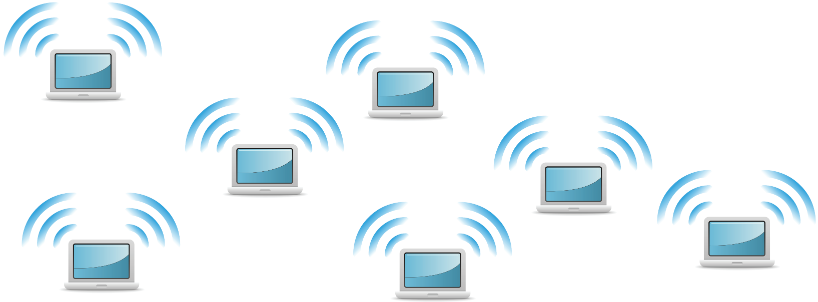
Bobak Nazer
Boston University

European Information Theory School
April 14, 2014

Multi-User Wireless Networks

- Must cope with interference, fading, and noise.

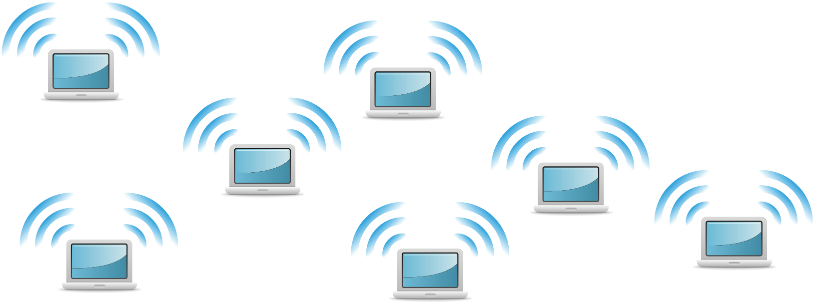
Multi-User Wireless Networks



- Must cope with interference, fading, and noise.
- Receivers observe noisy linear combinations of transmitted signals:

$$\mathbf{y} = \sum_{\ell} h_{\ell} \mathbf{x}_{\ell} + \mathbf{z}$$

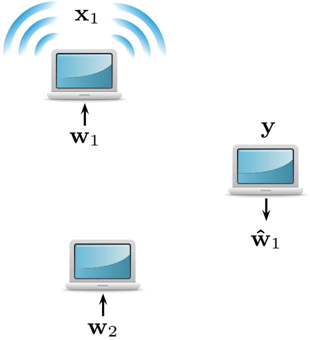
Multi-User Wireless Networks



- Must cope with interference, fading, and noise.
- Receivers observe noisy linear combinations of transmitted signals:

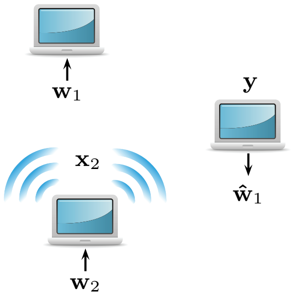
$$\mathbf{y} = \sum_{\ell} h_{\ell} \mathbf{x}_{\ell} + \mathbf{z}$$

- How should we deal with interference?



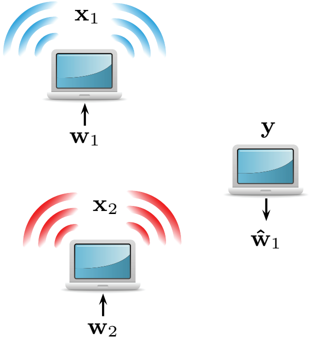
Possible Coding Strategies:

- Avoid interference / orthogonalize.



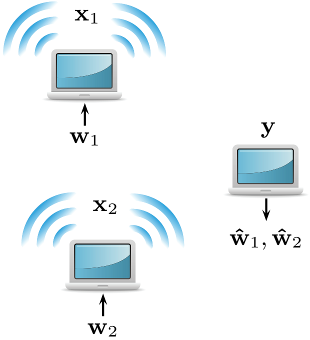
Possible Coding Strategies:

- Avoid interference / orthogonalize.



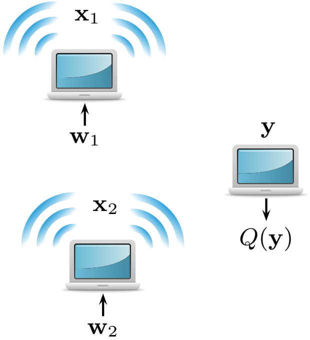
Possible Coding Strategies:

- Avoid interference / orthogonalize.
- Treat interference as noise.



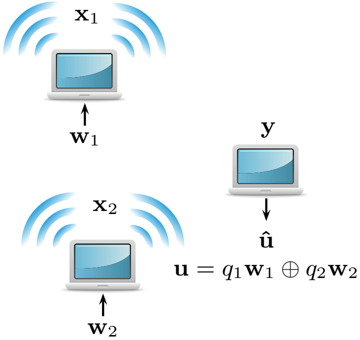
Possible Coding Strategies:

- Avoid interference / orthogonalize.
- Treat interference as noise.
- Decode interfering codewords.



Possible Coding Strategies:

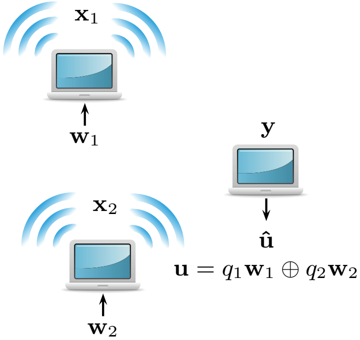
- Avoid interference / orthogonalize.
- Treat interference as noise.
- Decode interfering codewords.
- Work with the analog channel output (or a quantized version).



Possible Coding Strategies:

- Avoid interference / orthogonalize.
- Treat interference as noise.
- Decode interfering codewords.
- Work with the analog channel output (or a quantized version).
- Decode linear combinations of codewords.

- **Conventional Approach:** First, eliminate interference and then remove noise.



Possible Coding Strategies:

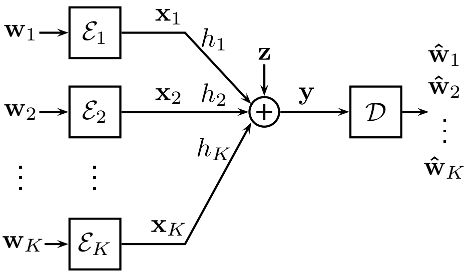
- Avoid interference / orthogonalize.
- Treat interference as noise.
- Decode interfering codewords.
- Work with the analog channel output (or a quantized version).
- Decode linear combinations of codewords.

- **Conventional Approach:** First, eliminate interference and then remove noise.
- **This Talk:** First, remove noise and then eliminate interference.

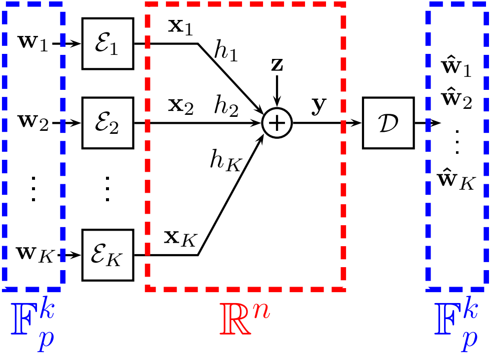
Compute-and-Forward

Goal: Convert noisy Gaussian networks into noiseless finite field ones.

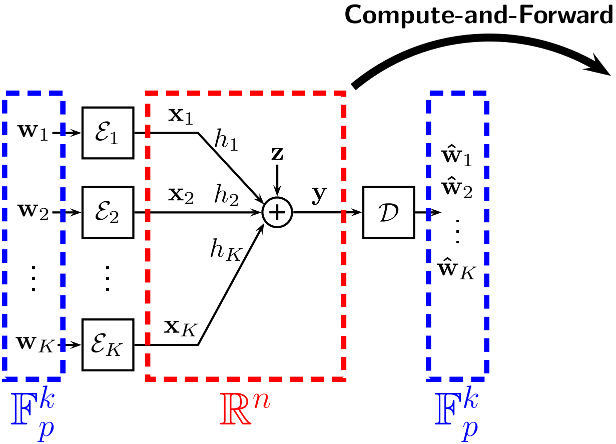
Goal: Convert noisy Gaussian networks into noiseless finite field ones.



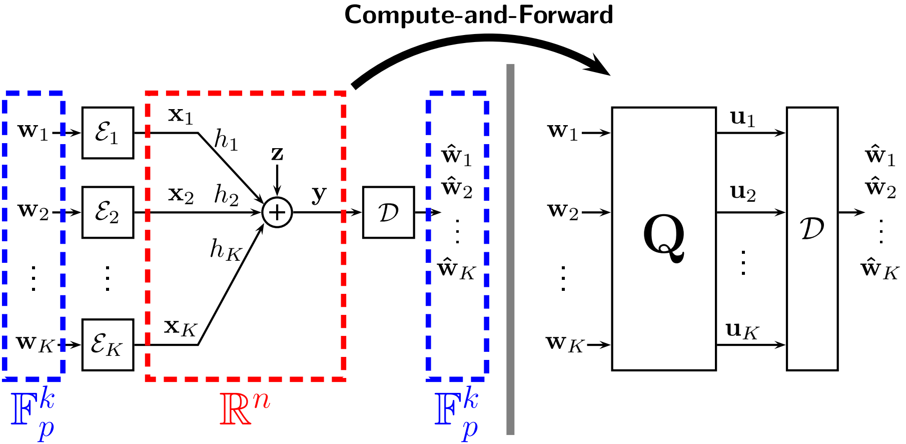
Goal: Convert noisy Gaussian networks into noiseless finite field ones.



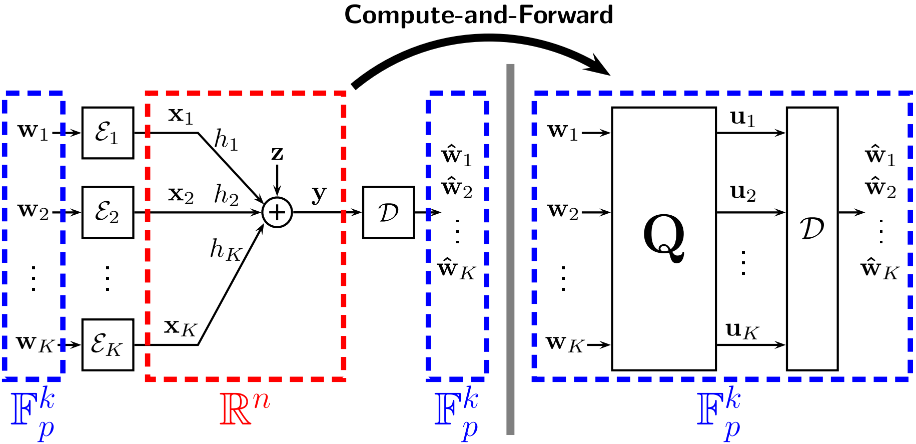
Goal: Convert noisy Gaussian networks into noiseless finite field ones.



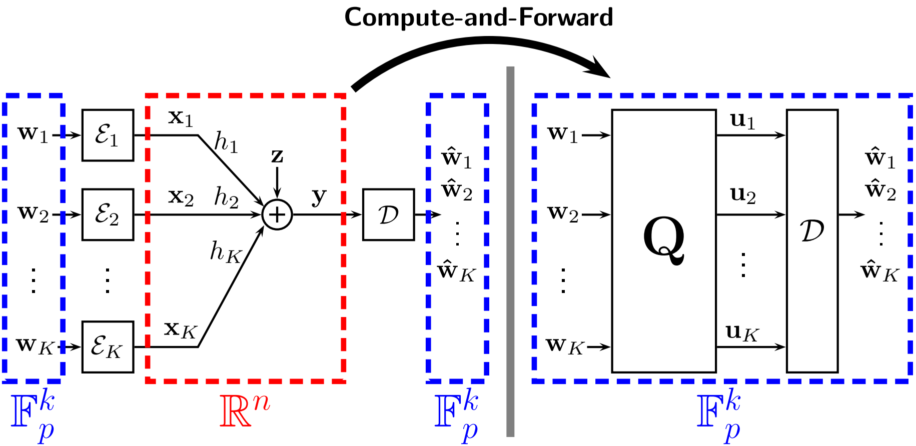
Goal: Convert noisy Gaussian networks into noiseless finite field ones.



Goal: Convert noisy Gaussian networks into noiseless finite field ones.

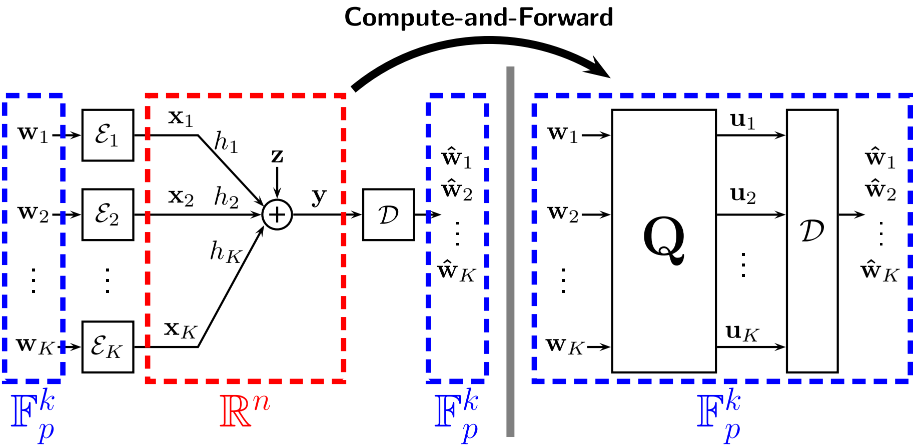


Goal: Convert noisy Gaussian networks into noiseless finite field ones.



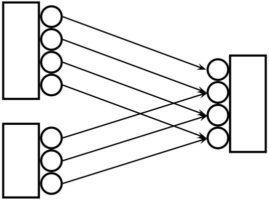
- Which linear combinations can be sent over a given channel?

Goal: Convert noisy Gaussian networks into noiseless finite field ones.



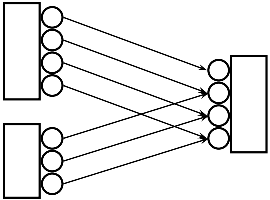
- Which linear combinations can be sent over a given channel?
- Where can this help us?

Deterministic Model from **Avestimehr-Diggavi-Tse '11:**

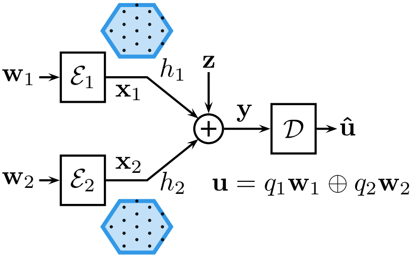


Top-Down vs. Bottom-Up

Deterministic Model from **Avestimehr-Diggavi-Tse '11:**



Compute-and-Forward from **Nazer-Gastpar '11:**



Top-Down vs. Bottom-Up

Deterministic Model from **Avestimehr-Diggavi-Tse '11:**

- Top-down approach.

Compute-and-Forward from **Nazer-Gastpar '11:**

- Bottom-up approach.

Deterministic Model from **Avestimehr-Diggavi-Tse '11:**

- Top-down approach.
- Proofs guided by finite-field models but no explicit mapping.

Compute-and-Forward from **Nazer-Gastpar '11:**

- Bottom-up approach.
- Explicit mapping from Gaussian to finite-field models.

Top-Down vs. Bottom-Up

Deterministic Model from **Avestimehr-Diggavi-Tse '11:**

- Top-down approach.
- Proofs guided by finite-field models but no explicit mapping.
- Clear interpretation of achievable rate region as a constant gap from cut-set upper bounds for many interesting topologies.

Compute-and-Forward from **Nazer-Gastpar '11:**

- Bottom-up approach.
- Explicit mapping from Gaussian to finite-field models.
- More challenging to connect achievable rate region to upper bounds.

Deterministic Model from **Avestimehr-Diggavi-Tse '11:**

- Top-down approach.
- Proofs guided by finite-field models but no explicit mapping.
- Clear interpretation of achievable rate region as a constant gap from cut-set upper bounds for many interesting topologies.
- More challenging to capture physical-layer phenomena such as MIMO and interference alignment.

Compute-and-Forward from **Nazer-Gastpar '11:**

- Bottom-up approach.
- Explicit mapping from Gaussian to finite-field models.
- More challenging to connect achievable rate region to upper bounds.
- Clear interpretation of physical-layer phenomena such as MIMO and interference alignment.

Road Map

- Warm-up: Compute-and-forward over finite field channels.

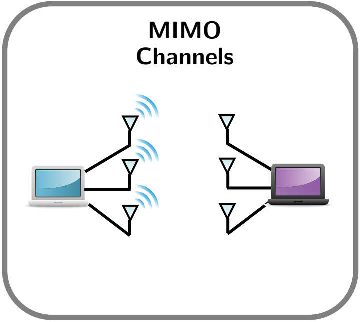
Road Map

- Warm-up: Compute-and-forward over finite field channels.
- Compute-and-forward over Gaussian channels.

Road Map

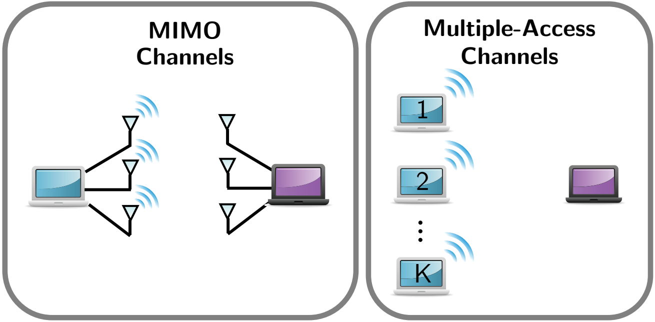
- Warm-up: Compute-and-forward over finite field channels.
- Compute-and-forward over Gaussian channels.
- Applications to communication across single-hop Gaussian networks.

- Warm-up: Compute-and-forward over finite field channels.
- Compute-and-forward over Gaussian channels.
- Applications to communication across single-hop Gaussian networks.



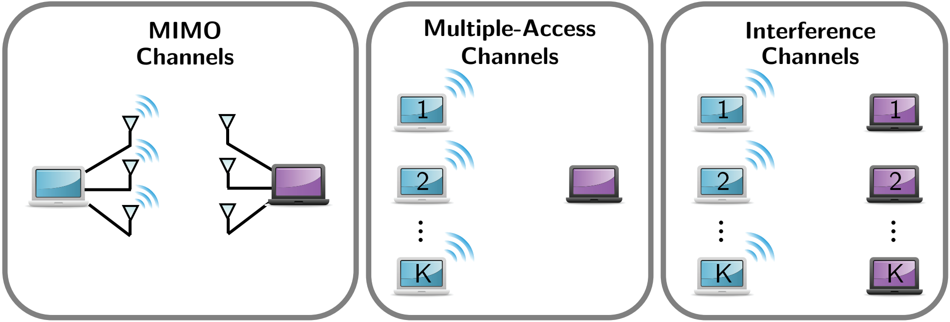
Road Map

- Warm-up: Compute-and-forward over finite field channels.
- Compute-and-forward over Gaussian channels.
- Applications to communication across single-hop Gaussian networks.



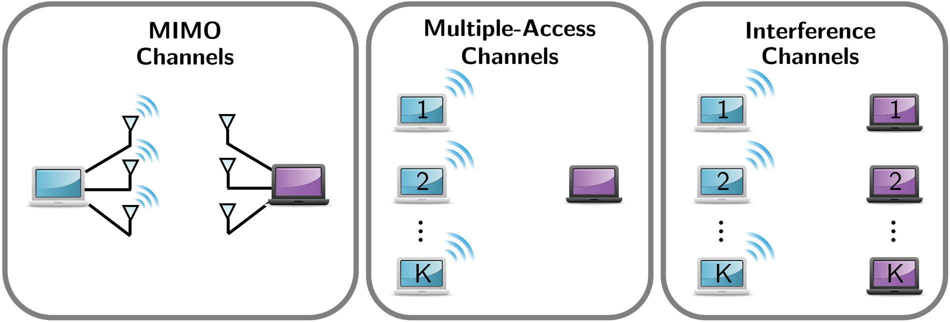
Road Map

- Warm-up: Compute-and-forward over finite field channels.
- Compute-and-forward over Gaussian channels.
- Applications to communication across single-hop Gaussian networks.

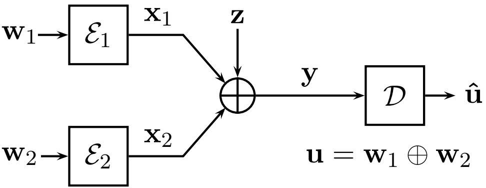


Road Map

- Warm-up: Compute-and-forward over finite field channels.
- Compute-and-forward over Gaussian channels.
- Applications to communication across single-hop Gaussian networks.

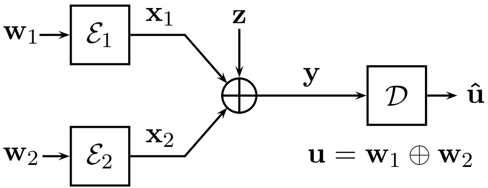


- Independent msgs $\mathbf{w}_1, \mathbf{w}_2 \in \mathbb{F}_p^k$.
- Want the sum $\mathbf{u} = \mathbf{w}_1 \oplus \mathbf{w}_2$ with vanishing prob. of error $\mathbb{P}(\hat{\mathbf{u}} \neq \mathbf{u}) \to 0$



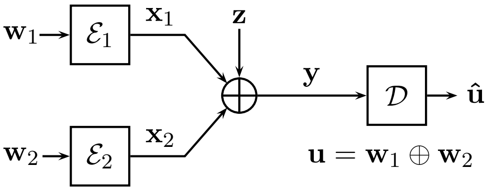
Computation over Finite Field Multiple-Access Channels

- Independent msgs $\mathbf{w}_1, \mathbf{w}_2 \in \mathbb{F}_p^k$.
- Want the sum $\mathbf{u} = \mathbf{w}_1 \oplus \mathbf{w}_2$ with vanishing prob. of error $\mathbb{P}(\hat{\mathbf{u}} \neq \mathbf{u}) \to 0$



I.I.D. Random Coding

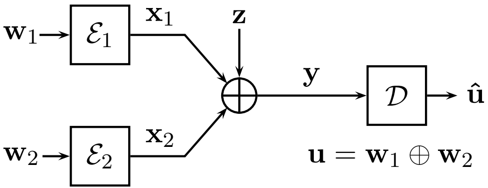
- Independent msgs $\mathbf{w}_1, \mathbf{w}_2 \in \mathbb{F}_p^k$.
- Want the sum $\mathbf{u} = \mathbf{w}_1 \oplus \mathbf{w}_2$ with vanishing prob. of error $\mathbb{P}(\hat{\mathbf{u}} \neq \mathbf{u}) \to 0$



I.I.D. Random Coding

- Generate 2^{nR_1} i.i.d. uniform codewords for user 1.

- Independent msgs $\mathbf{w}_1, \mathbf{w}_2 \in \mathbb{F}_p^k$.
- Want the sum $\mathbf{u} = \mathbf{w}_1 \oplus \mathbf{w}_2$ with vanishing prob. of error $\mathbb{P}(\hat{\mathbf{u}} \neq \mathbf{u}) \to 0$

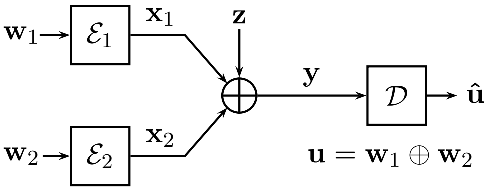


I.I.D. Random Coding

- Generate 2^{nR_1} i.i.d. uniform codewords for user 1.
- Generate 2^{nR_2} i.i.d. uniform codewords for user 2.

Computation over Finite Field Multiple-Access Channels

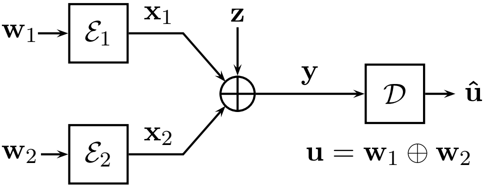
- Independent msgs $\mathbf{w}_1, \mathbf{w}_2 \in \mathbb{F}_p^k$.
- Want the sum $\mathbf{u} = \mathbf{w}_1 \oplus \mathbf{w}_2$ with vanishing prob. of error $\mathbb{P}(\hat{\mathbf{u}} \neq \mathbf{u}) \to 0$



I.I.D. Random Coding

- Generate 2^{nR_1} i.i.d. uniform codewords for user 1.
- Generate 2^{nR_2} i.i.d. uniform codewords for user 2.
- With high probability, (nearly) all sums of codewords are distinct.

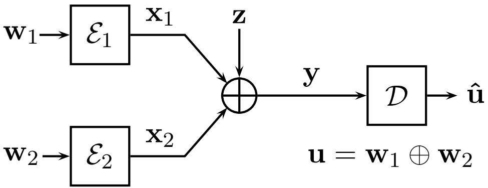
- Independent msgs $\mathbf{w}_1, \mathbf{w}_2 \in \mathbb{F}_p^k$.
- Want the sum $\mathbf{u} = \mathbf{w}_1 \oplus \mathbf{w}_2$ with vanishing prob. of error $\mathbb{P}(\hat{\mathbf{u}} \neq \mathbf{u}) \to 0$



I.I.D. Random Coding

- Generate 2^{nR_1} i.i.d. uniform codewords for user 1.
- Generate 2^{nR_2} i.i.d. uniform codewords for user 2.
- With high probability, (nearly) all sums of codewords are distinct.
- This is ideal for multiple-access but not for computation.

- Independent msgs $\mathbf{w}_1, \mathbf{w}_2 \in \mathbb{F}_p^k$.
- Want the sum $\mathbf{u} = \mathbf{w}_1 \oplus \mathbf{w}_2$ with vanishing prob. of error $\mathbb{P}(\hat{\mathbf{u}} \neq \mathbf{u}) \to 0$

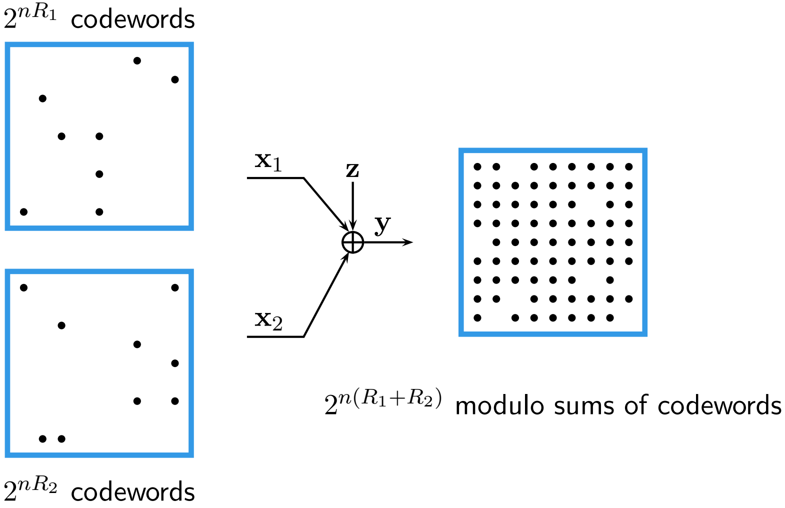


I.I.D. Random Coding

- Generate 2^{nR_1} i.i.d. uniform codewords for user 1.
- Generate 2^{nR_2} i.i.d. uniform codewords for user 2.
- With high probability, (nearly) all sums of codewords are distinct.
- This is ideal for multiple-access but not for computation.

$$\begin{aligned} R_1 + R_2 &\leq I(X_1, X_2; Y) \\ &= H(Y) - H(Y|X_1, X_2) \\ &= \log p - H(Z) \end{aligned}$$

Random i.i.d. codes are not good for computation



- Linear Codebook: A linear map between messages and codewords (instead of a lookup table).

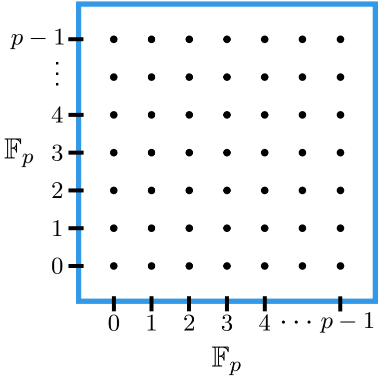
<u>p-**ary Linear Codes**</u>

- Message $\mathbf{w}$ is a length-k vector over $\mathbb{F}_p$.
- Codeword $\mathbf{x}$ is a length-n vector over $\mathbb{F}_p$.
- Encoding process is just a matrix multiplication, $\mathbf{x} = \mathbf{G}\mathbf{w}$.

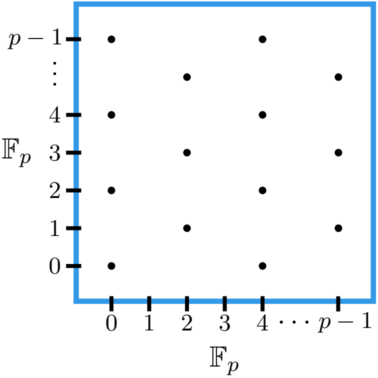
$$\begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} g_{11} & g_{12} & \cdots & g_{1k} \\ g_{21} & g_{22} & \cdots & g_{2k} \\ \vdots & \vdots & \ddots & \vdots \\ g_{n1} & g_{n2} & \cdots & g_{nk} \end{bmatrix} \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_k \end{bmatrix}$$

- Recall that, for prime p, operations over $\mathbb{F}_p$ are just $\mod p$ operations over the reals.
- Rate $R = \dfrac{k}{n} \log p$ (in bits)

- Linear code looks like a regular subsampling of the elements of $\mathbb{F}_p^n$.
- **Random linear code:** Generate each element g_{ij} of the generator matrix $\mathbf{G}$ elementwise i.i.d. according to a uniform distribution over $\mathbb{F}_p$.
- How are the codewords distributed?



- Linear code looks like a regular subsampling of the elements of $\mathbb{F}_p^n$.
- **Random linear code:** Generate each element g_{ij} of the generator matrix $\mathbf{G}$ elementwise i.i.d. according to a uniform distribution over $\mathbb{F}_p$.
- How are the codewords distributed?



It is more to instead analyze the shifted ensemble $\mathbf{x} = \mathbf{G}\mathbf{w} \oplus \mathbf{v}$ where $\mathbf{v}$ is an i.i.d. uniform sequence. (See **Gallager.**)

Shifted Codeword Properties

It is more to instead analyze the shifted ensemble $\mathbf{x} = \mathbf{G}\mathbf{w} \oplus \mathbf{v}$ where $\mathbf{v}$ is an i.i.d. uniform sequence. (See **Gallager.**)

Shifted Codeword Properties

1. Marginally uniform over $\mathbb{F}_q^n$. For a given message $\mathbf{w}$, the codeword $\mathbf{x}$ looks like an i.i.d. uniform sequence.

$$\mathbb{P}(\mathbf{x} = \mathsf{x}) = \frac{1}{p^n} \quad \text{for all } \mathsf{x} \in \mathbb{F}_p^n$$

It is more to instead analyze the shifted ensemble $\mathbf{x} = \mathbf{G}\mathbf{w} \oplus \mathbf{v}$ where $\mathbf{v}$ is an i.i.d. uniform sequence. (See **Gallager.**)

Shifted Codeword Properties

1. Marginally uniform over $\mathbb{F}_q^n$. For a given message $\mathbf{w}$, the codeword $\mathbf{x}$ looks like an i.i.d. uniform sequence.

$$\mathbb{P}(\mathbf{x} = \mathsf{x}) = \frac{1}{p^n} \quad \text{for all } \mathsf{x} \in \mathbb{F}_p^n$$

2. Pairwise independent. For $\mathbf{w}_1 \neq \mathbf{w}_2$, the associated codewords $\mathbf{x}_1 = \mathbf{G}\mathbf{w}_1 \oplus \mathbf{v}$ and $\mathbf{x}_2 = \mathbf{G}\mathbf{w}_2 \oplus \mathbf{v}$ are independent.

$$\mathbb{P}(\mathbf{x_1} = \mathsf{x}_1, \mathbf{x_2} = \mathsf{x}_2) = \frac{1}{p^{2n}} = \mathbb{P}(\mathbf{x}_1 = \mathsf{x}_1)\mathbb{P}(\mathbf{x}_2 = \mathsf{x}_2)$$

Achievable Rates

- Transmitter sends: $\mathbf{x} = \mathbf{G}\mathbf{w} \oplus \mathbf{v}$.
- Point-to-point channel: $\mathbf{y} = \mathbf{x} \oplus \mathbf{z}$. Noise is i.i.d.
- Receiver decodes via joint typicality decoding.

Achievable Rates

- Transmitter sends: $\mathbf{x} = \mathbf{G}\mathbf{w} \oplus \mathbf{v}$.
- Point-to-point channel: $\mathbf{y} = \mathbf{x} \oplus \mathbf{z}$. Noise is i.i.d.
- Receiver decodes via joint typicality decoding.
- Error occurs if the true codeword $\mathbf{x}$ is not jointly typical with $\mathbf{y}$ or any other codeword $\tilde{\mathbf{x}}$ is jointly typical with $\mathbf{y}$.

Achievable Rates

- Transmitter sends: $\mathbf{x} = \mathbf{G}\mathbf{w} \oplus \mathbf{v}$.
- Point-to-point channel: $\mathbf{y} = \mathbf{x} \oplus \mathbf{z}$. Noise is i.i.d.
- Receiver decodes via joint typicality decoding.
- Error occurs if the true codeword $\mathbf{x}$ is not jointly typical with $\mathbf{y}$ or any other codeword $\tilde{\mathbf{x}}$ is jointly typical with $\mathbf{y}$.
- Using the union bound,

$$\mathbb{P}(\hat{\mathbf{w}} \neq \mathbf{w}) \leq \mathbb{P}\Big(\big(\mathbf{x}, \mathbf{y}\big) \notin \mathcal{T}_\epsilon^{(n)}\Big) + \sum_{\tilde{\mathbf{w}} \neq \mathbf{w}} \mathbb{P}\Big(\big(\tilde{\mathbf{x}}, \mathbf{y}\big) \in \mathcal{T}_\epsilon^{(n)}\Big)$$
$$\leq \epsilon + 2^{nR}\, 2^{-n(I(X;Y)-3\epsilon)}$$

Achievable Rates

- Transmitter sends: $\mathbf{x} = \mathbf{G}\mathbf{w} \oplus \mathbf{v}$.
- Point-to-point channel: $\mathbf{y} = \mathbf{x} \oplus \mathbf{z}$. Noise is i.i.d.
- Receiver decodes via joint typicality decoding.
- Error occurs if the true codeword $\mathbf{x}$ is not jointly typical with $\mathbf{y}$ or any other codeword $\tilde{\mathbf{x}}$ is jointly typical with $\mathbf{y}$.
- Using the union bound,

$$\begin{aligned}\mathbb{P}(\hat{\mathbf{w}} \neq \mathbf{w}) &\leq \mathbb{P}\Big(\big(\mathbf{x}, \mathbf{y}\big) \notin \mathcal{T}_\epsilon^{(n)}\Big) + \sum_{\tilde{\mathbf{w}} \neq \mathbf{w}} \mathbb{P}\Big(\big(\tilde{\mathbf{x}}, \mathbf{y}\big) \in \mathcal{T}_\epsilon^{(n)}\Big) \\ &\leq \epsilon + 2^{nR}\, 2^{-n(I(X;Y) - 3\epsilon)}\end{aligned}$$

- It follows that there exists a good fixed generator matrix $\mathbf{G}$ and shift $\mathbf{v}$ for any rate $R < I(X;Y)$ where X is uniform.

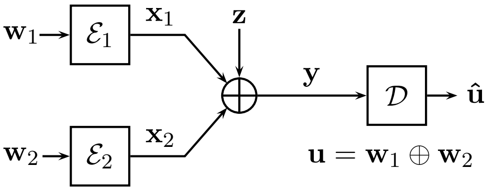
Achievable Rates

- Transmitter sends: $\mathbf{x} = \mathbf{G}\mathbf{w} \oplus \mathbf{v}$.
- Point-to-point channel: $\mathbf{y} = \mathbf{x} \oplus \mathbf{z}$. Noise is i.i.d.
- Receiver decodes via joint typicality decoding.
- Error occurs if the true codeword $\mathbf{x}$ is not jointly typical with $\mathbf{y}$ or any other codeword $\tilde{\mathbf{x}}$ is jointly typical with $\mathbf{y}$.
- Using the union bound,

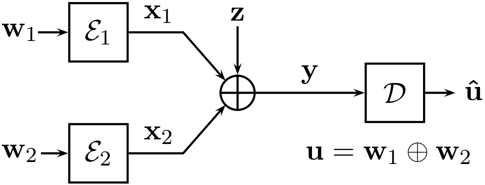
$$\begin{aligned}\mathbb{P}(\hat{\mathbf{w}} \neq \mathbf{w}) &\leq \mathbb{P}\Big((\mathbf{x}, \mathbf{y}) \notin \mathcal{T}_\epsilon^{(n)} \Big) + \sum_{\tilde{\mathbf{w}} \neq \mathbf{w}} \mathbb{P}\Big((\tilde{\mathbf{x}}, \mathbf{y}) \in \mathcal{T}_\epsilon^{(n)} \Big) \\ &\leq \epsilon + 2^{nR} \, 2^{-n(I(X;Y) - 3\epsilon)}\end{aligned}$$

- It follows that there exists a good fixed generator matrix $\mathbf{G}$ and shift $\mathbf{v}$ for any rate $R < I(X;Y)$ where X is uniform.
- Shift $\mathbf{v}$ is unnecessary for additive noise channels.

- Independent msgs $\mathbf{w}_1, \mathbf{w}_2 \in \mathbb{F}_p^k$.
- Want the sum $\mathbf{u} = \mathbf{w}_1 \oplus \mathbf{w}_2$ with vanishing prob. of error $\mathbb{P}(\hat{\mathbf{u}} \neq \mathbf{u}) \to 0$

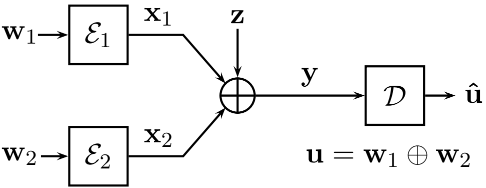


- Independent msgs $\mathbf{w}_1, \mathbf{w}_2 \in \mathbb{F}_p^k$.
- Want the sum $\mathbf{u} = \mathbf{w}_1 \oplus \mathbf{w}_2$ with vanishing prob. of error $\mathbb{P}(\hat{\mathbf{u}} \neq \mathbf{u}) \to 0$



Random Linear Coding

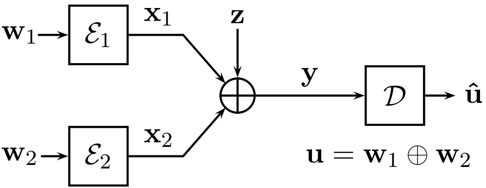
- Independent msgs $\mathbf{w}_1, \mathbf{w}_2 \in \mathbb{F}_p^k$.
- Want the sum $\mathbf{u} = \mathbf{w}_1 \oplus \mathbf{w}_2$ with vanishing prob. of error $\mathbb{P}(\hat{\mathbf{u}} \neq \mathbf{u}) \to 0$



Random Linear Coding

- User 1 transmits $\mathbf{x}_1 = \mathbf{G}\mathbf{w}_1$.

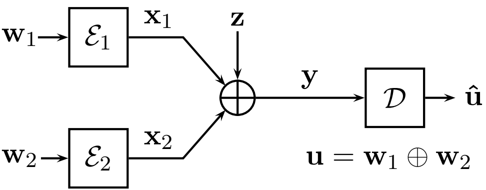
- Independent msgs $\mathbf{w}_1, \mathbf{w}_2 \in \mathbb{F}_p^k$.
- Want the sum $\mathbf{u} = \mathbf{w}_1 \oplus \mathbf{w}_2$ with vanishing prob. of error $\mathbb{P}(\hat{\mathbf{u}} \neq \mathbf{u}) \to 0$



Random Linear Coding

- User 1 transmits $\mathbf{x}_1 = \mathbf{G}\mathbf{w}_1$.
- User 2 transmits $\mathbf{x}_2 = \mathbf{G}\mathbf{w}_2$. Idea of using the same linear code at multiple terminals stems from **Körner-Marton '79**.

- Independent msgs $\mathbf{w}_1, \mathbf{w}_2 \in \mathbb{F}_p^k$.
- Want the sum $\mathbf{u} = \mathbf{w}_1 \oplus \mathbf{w}_2$ with vanishing prob. of error $\mathbb{P}(\hat{\mathbf{u}} \neq \mathbf{u}) \to 0$



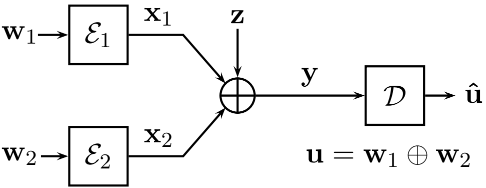
Random Linear Coding

- User 1 transmits $\mathbf{x}_1 = \mathbf{G}\mathbf{w}_1$.
- User 2 transmits $\mathbf{x}_2 = \mathbf{G}\mathbf{w}_2$. Idea of using the same linear code at multiple terminals stems from **Körner-Marton '79**.
- Receiver observes

$$\begin{aligned} \mathbf{y} &= \mathbf{G}\mathbf{w}_1 \oplus \mathbf{G}\mathbf{w}_2 \oplus \mathbf{z} \\ &= \mathbf{G}(\mathbf{w}_1 \oplus \mathbf{w}_2) \oplus \mathbf{z} \\ &= \mathbf{G}\mathbf{u} \oplus \mathbf{z} \ . \end{aligned}$$

Computation over Finite Field Multiple-Access Channels

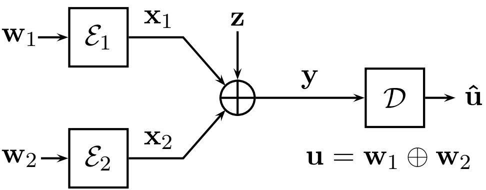
- Independent msgs $\mathbf{w}_1, \mathbf{w}_2 \in \mathbb{F}_p^k$.
- Want the sum $\mathbf{u} = \mathbf{w}_1 \oplus \mathbf{w}_2$ with vanishing prob. of error $\mathbb{P}(\hat{\mathbf{u}} \neq \mathbf{u}) \to 0$



Random Linear Coding

- User 1 transmits $\mathbf{x}_1 = \mathbf{G}\mathbf{w}_1$.
- User 2 transmits $\mathbf{x}_2 = \mathbf{G}\mathbf{w}_2$. Idea of using the same linear code at multiple terminals stems from **Körner-Marton '79**.
- Receiver observes
$$\begin{aligned} \mathbf{y} &= \mathbf{G}\mathbf{w}_1 \oplus \mathbf{G}\mathbf{w}_2 \oplus \mathbf{z} \\ &= \mathbf{G}(\mathbf{w}_1 \oplus \mathbf{w}_2) \oplus \mathbf{z} \\ &= \mathbf{G}\mathbf{u} \oplus \mathbf{z}\ . \end{aligned}$$
- **Nazer-Gastpar '07:** Decoding succeeds w.h.p. if

- Independent msgs $\mathbf{w}_1, \mathbf{w}_2 \in \mathbb{F}_p^k$.
- Want the sum $\mathbf{u} = \mathbf{w}_1 \oplus \mathbf{w}_2$ with vanishing prob. of error $\mathbb{P}(\hat{\mathbf{u}} \neq \mathbf{u}) \to 0$



Random Linear Coding

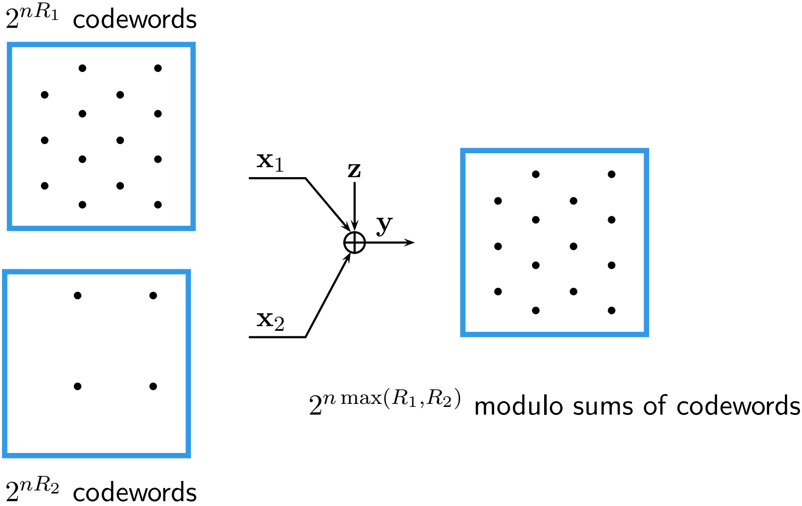
- User 1 transmits $\mathbf{x}_1 = \mathbf{G}\mathbf{w}_1$.
- User 2 transmits $\mathbf{x}_2 = \mathbf{G}\mathbf{w}_2$. Idea of using the same linear code at multiple terminals stems from **Körner-Marton '79**.
- Receiver observes

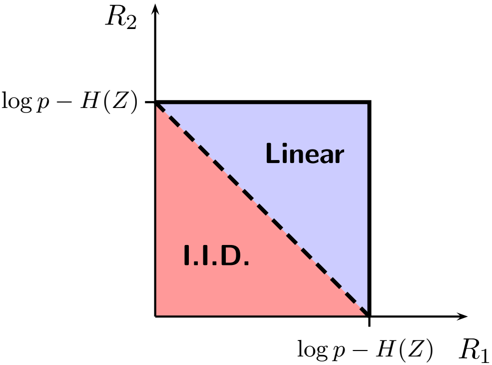
$$\begin{aligned}\mathbf{y} &= \mathbf{G}\mathbf{w}_1 \oplus \mathbf{G}\mathbf{w}_2 \oplus \mathbf{z} \\ &= \mathbf{G}(\mathbf{w}_1 \oplus \mathbf{w}_2) \oplus \mathbf{z} \\ &= \mathbf{G}\mathbf{u} \oplus \mathbf{z}\ .\end{aligned}$$

- **Nazer-Gastpar '07:** Decoding succeeds w.h.p. if

$$\begin{aligned}\max(R_1, R_2) \leq I(X_1 \oplus X_2; Y) &= H(Y) - H(Y|X_1 \oplus X_2) \\ &= \log p - H(Z)\ .\end{aligned}$$

Random linear codes are good for computation

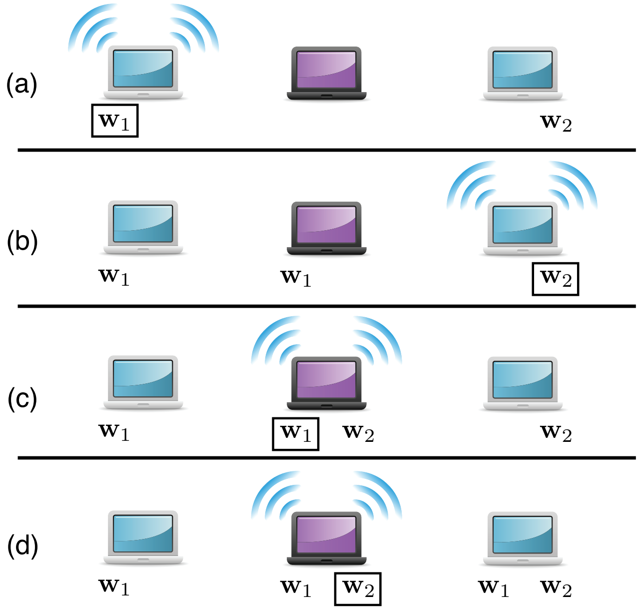




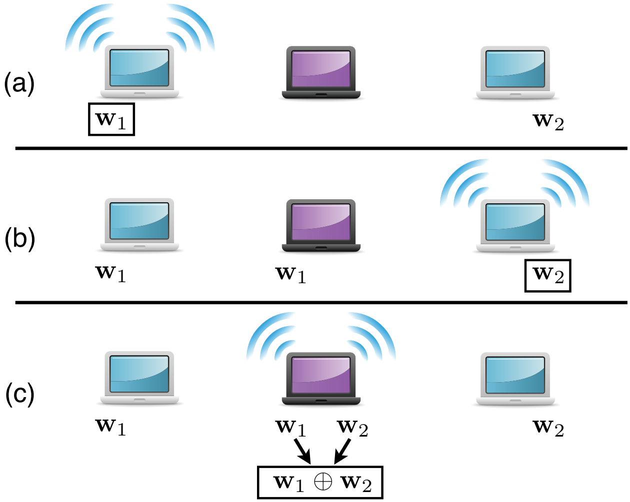
- **I.I.D. Random Coding**: $R_1 + R_2 \leq \log p - H(Z)$
- **Random Linear Coding**: $\max(R_1, R_2) \leq \log p - H(Z)$
- Linear codes *double the sum rate.*
- Are they also useful for *sending messages* (rather than functions thereof)?

- Elegant example proposed by **Wu-Chou-Kung '04.**
- Closely related to butterfly network from **Ahlswede-Cai-Li-Yeung '00.**

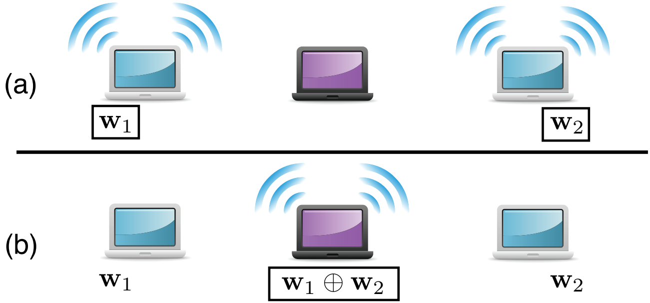
Two-Way Relay Channel – Time-Division

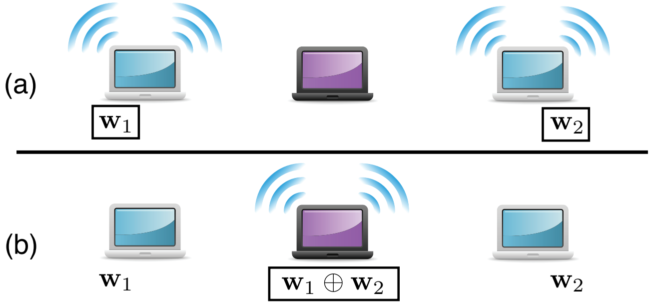


Two-Way Relay Channel – Network Coding



Two-Way Relay Channel – Physical-Layer Network Coding

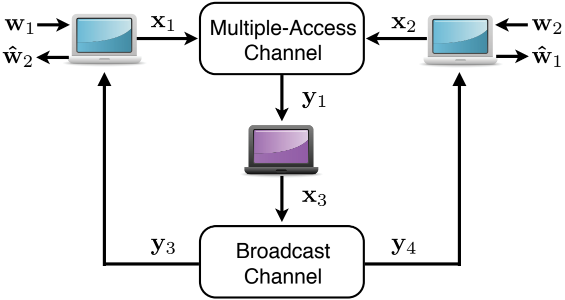




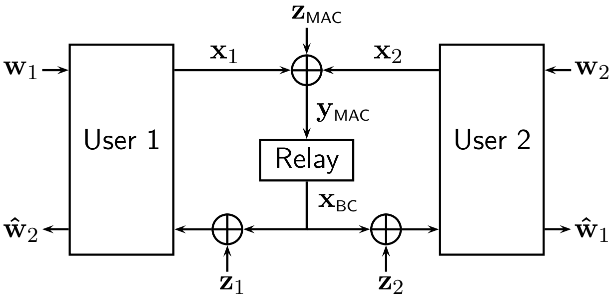
- Physical-layer network coding: exploiting the wireless medium for network coding. Independently and concurrently proposed by **Zhang-Liew-Lam '06, Popovski-Yomo '06, Nazer-Gastpar '06.**
- Sometimes referred to as Analog Network Coding **Katti-Gollakota-Katabi '07.**
- Some recent surveys **Liew-Zhang-Lu '11, Nazer-Gastpar '11.**

(Finite Field) Two-Way Relay Channel

(Finite Field) Two-Way Relay Channel



(Finite Field) Two-Way Relay Channel



- i.i.d. noise sequences with entropy $H(Z)$.
- Rates R_1 and R_2.

- Cut-Set Upper Bound:

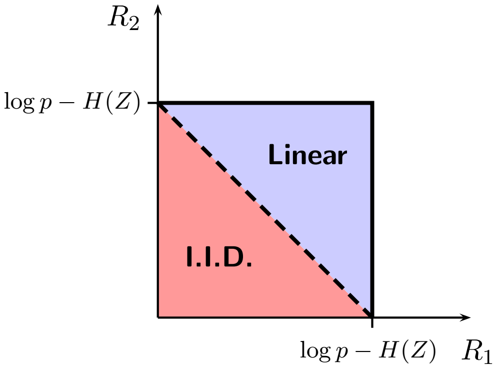
$$\max(R_1, R_2) \leq \log p - H(Z)$$

- I.I.D. Random Coding: Relay decodes $\mathbf{w}_1, \mathbf{w}_2$, transmits $\mathbf{w}_1 \oplus \mathbf{w}_2$.

$$R_1 + R_2 \leq \log p - H(Z)$$

- Random Linear Coding: Relay decodes and retransmits $\mathbf{w}_1 \oplus \mathbf{w}_2$.

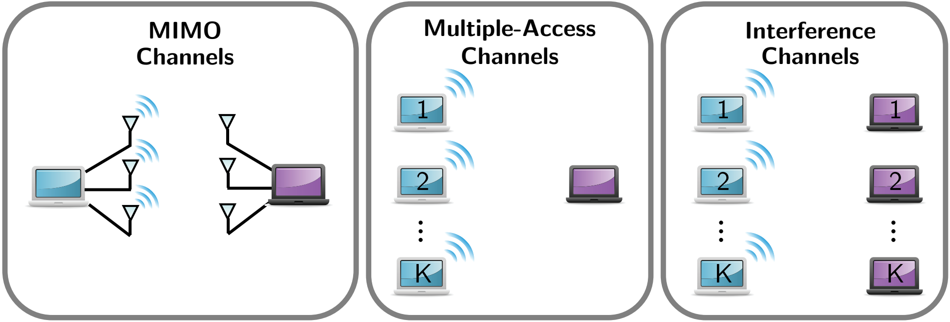
$$\max(R_1, R_2) \leq \log p - H(Z)$$



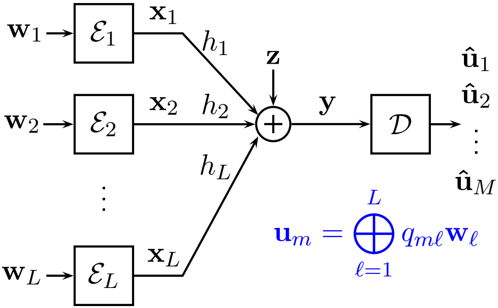
- Linear codes can double the sum rate *for exchanging messages.*

Road Map

- Warm-up: Compute-and-forward over finite field channels.
- Compute-and-forward over Gaussian channels.
- Applications to communication across single-hop Gaussian networks.

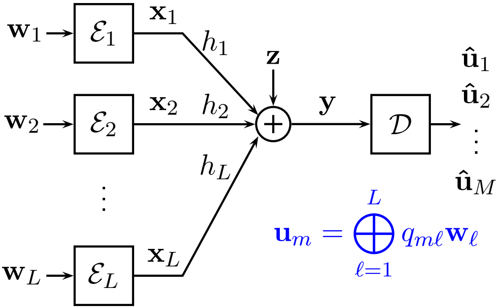


Compute-and-Forward: Problem Statement



- Messages are finite field vectors, $\mathbf{w}_\ell \in \mathbb{F}_p^k$.
- Real-valued inputs and outputs, $\mathbf{x}_\ell, \mathbf{y} \in \mathbb{R}^n$.
- Power constraint, $\frac{1}{n}\mathbb{E}\|\mathbf{x}_\ell\|^2 \leq P$.
- Gaussian noise, $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$.
- Equal rates: $R = \dfrac{k}{n}\log_2 p$

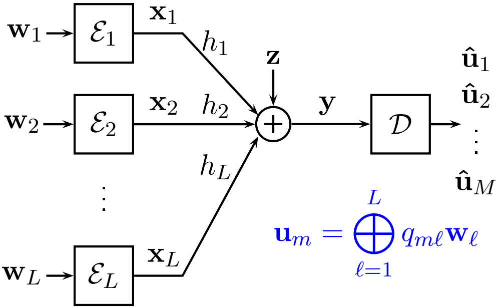
Compute-and-Forward: Problem Statement



- Messages are finite field vectors, $\mathbf{w}_\ell \in \mathbb{F}_p^k$.
- Real-valued inputs and outputs, $\mathbf{x}_\ell, \mathbf{y} \in \mathbb{R}^n$.
- Power constraint, $\frac{1}{n}\mathbb{E}\|\mathbf{x}_\ell\|^2 \leq P$.
- Gaussian noise, $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$.
- Equal rates: $R = \dfrac{k}{n}\log_2 p$

- Decoder wants M linear combinations of the messages with vanishing probability of error $\lim_{n\to\infty} \mathbb{P}\Big(\bigcup_m \{\hat{\mathbf{u}}_m \neq \mathbf{u}_m\}\Big) = 0$.

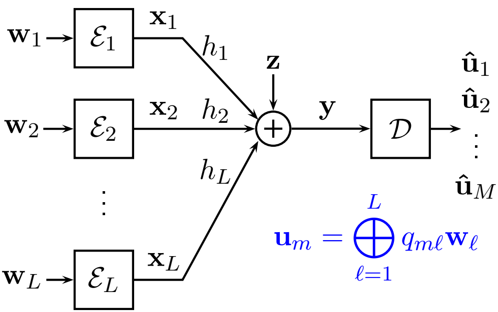
Compute-and-Forward: Problem Statement



- Messages are finite field vectors, $\mathbf{w}_\ell \in \mathbb{F}_p^k$.
- Real-valued inputs and outputs, $\mathbf{x}_\ell, \mathbf{y} \in \mathbb{R}^n$.
- Power constraint, $\frac{1}{n}\mathbb{E}\|\mathbf{x}_\ell\|^2 \leq P$.
- Gaussian noise, $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$.
- Equal rates: $R = \dfrac{k}{n}\log_2 p$

- Decoder wants M linear combinations of the messages with vanishing probability of error $\lim_{n\to\infty} \mathbb{P}\left(\bigcup_m \{\hat{\mathbf{u}}_m \neq \mathbf{u}_m\}\right) = 0$.
- Receiver can use its channel state information (CSI) to match the linear combination coefficients $q_{m\ell} \in \mathbb{F}_p$ to the channel coefficients $h_\ell \in \mathbb{R}$. Transmitters do not require CSI.

Compute-and-Forward: Problem Statement



- Messages are finite field vectors, $\mathbf{w}_\ell \in \mathbb{F}_p^k$.
- Real-valued inputs and outputs, $\mathbf{x}_\ell, \mathbf{y} \in \mathbb{R}^n$.
- Power constraint, $\frac{1}{n}\mathbb{E}\|\mathbf{x}_\ell\|^2 \leq P$.
- Gaussian noise, $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$.
- Equal rates: $R = \dfrac{k}{n}\log_2 p$

- Decoder wants M linear combinations of the messages with vanishing probability of error $\lim_{n\to\infty} \mathbb{P}\Big(\bigcup_m \{\hat{\mathbf{u}}_m \neq \mathbf{u}_m\}\Big) = 0$.
- Receiver can use its channel state information (CSI) to match the linear combination coefficients $q_{m\ell} \in \mathbb{F}_p$ to the channel coefficients $h_\ell \in \mathbb{R}$. Transmitters do not require CSI.
- What rates are achievable as a function of h_ℓ and $q_{m\ell}$?

Computation Rate

- Want to characterize achievable rates as a function of h_ℓ and $q_{m\ell}$.

Computation Rate

- Want to characterize achievable rates as a function of h_ℓ and $q_{m\ell}$.
- Easier to think about integer rather than finite field coefficients.

- Want to characterize achievable rates as a function of h_ℓ and $q_{m\ell}$.
- Easier to think about integer rather than finite field coefficients.
- The linear combination with integer coefficient vector $\mathbf{a}_m = [a_{m1}\ a_{m2}\ \cdots\ a_{mL}]^\mathsf{T} \in \mathbb{Z}^L$ corresponds to

$$\mathbf{u}_m = \bigoplus_{\ell=1}^{L} q_{m\ell}\mathbf{w}_\ell \qquad \text{where } q_{m\ell} = [a_{m\ell}] \bmod p$$

(where we assume an implicit mapping between $\mathbb{F}_p$ and $\mathbb{Z}_p$).

- Want to characterize achievable rates as a function of h_ℓ and $q_{m\ell}$.
- Easier to think about integer rather than finite field coefficients.
- The linear combination with integer coefficient vector $\mathbf{a}_m = [a_{m1}\ a_{m2}\ \cdots\ a_{mL}]^\mathsf{T} \in \mathbb{Z}^L$ corresponds to

$$\mathbf{u}_m = \bigoplus_{\ell=1}^{L} q_{m\ell}\mathbf{w}_\ell \qquad \text{where } q_{m\ell} = [a_{m\ell}] \bmod p$$

  (where we assume an implicit mapping between $\mathbb{F}_p$ and $\mathbb{Z}_p$).
- **Key Definition:** The computation rate region described by $R_{\mathsf{comp}}(\mathbf{h}, \mathbf{a})$ is *achievable* if, for any $\epsilon > 0$ and n, p large enough, a receiver can decode any linear combinations with integer coefficient vectors $\mathbf{a}_1, \dots, \mathbf{a}_M \in \mathbb{Z}^L$ for which the message rate R satisfies

$$R < \min_m R_{\mathsf{comp}}(\mathbf{h}, \mathbf{a}_m)$$

$$
\begin{aligned}
\mathbf{y} &= \sum_{\ell=1}^{L} h_\ell \mathbf{x}_\ell + \mathbf{z} \\
&= \sum_{\ell=1}^{L} a_\ell \mathbf{x}_\ell + \sum_{\ell=1}^{L} (h_\ell - a_\ell)\mathbf{x}_\ell + \mathbf{z}
\end{aligned}
$$

Desired Codebook:

- Closed under integer linear combinations $\implies$ lattice codebook.

$$\mathbf{y} = \sum_{\ell=1}^{L} h_\ell \mathbf{x}_\ell + \mathbf{z}$$
$$= \sum_{\ell=1}^{L} a_\ell \mathbf{x}_\ell + \underbrace{\sum_{\ell=1}^{L} (h_\ell - a_\ell)\mathbf{x}_\ell + \mathbf{z}}_{\text{Effective Noise}}$$

Desired Codebook:

- Closed under integer linear combinations $\Longrightarrow$ lattice codebook.
- Independent effective noise $\Longrightarrow$ dithering.

Compute-and-Forward: Effective Noise

$$\mathbf{y} = \sum_{\ell=1}^{L} h_\ell \mathbf{x}_\ell + \mathbf{z}$$

$$= \sum_{\ell=1}^{L} a_\ell \mathbf{x}_\ell + \underbrace{\sum_{\ell=1}^{L} (h_\ell - a_\ell)\mathbf{x}_\ell + \mathbf{z}}_{\text{Effective Noise}} \xrightarrow{\text{Decode}} \bigoplus_{\ell=1}^{L} q_\ell \mathbf{w}_\ell$$

Desired Codebook:

- Closed under integer linear combinations $\Longrightarrow$ lattice codebook.
- Independent effective noise $\Longrightarrow$ dithering.
- Isomorphic to $\mathbb{F}_p^k$ $\Longrightarrow$ nested lattice codebook.

Lattices

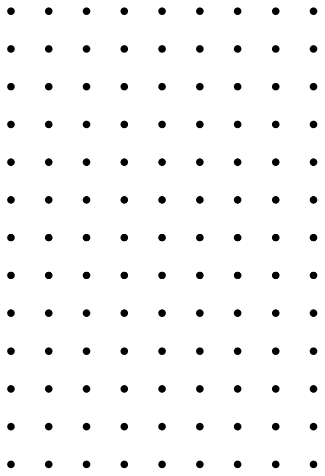
- A lattice Λ is a discrete subgroup of $\mathbb{R}^n$.
- Can express as a linear transformation of the integer vectors,

$$\Lambda = \mathbf{B}\mathbb{Z}^n ,$$

for some (non-unique) $\mathbf{B} \in \mathbb{R}^{n \times n}$.

Lattice Properties

- Closed under addition: $\lambda_1, \lambda_2 \in \Lambda \implies \lambda_1 + \lambda_2 \in \Lambda$.
- Symmetric: $\lambda \in \Lambda \implies -\lambda \in \Lambda$

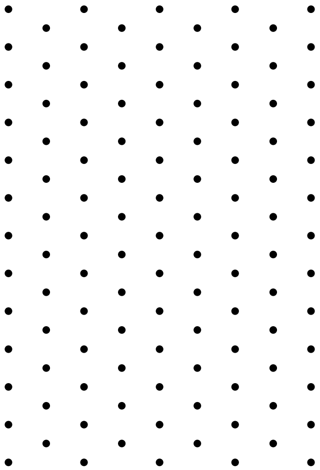


$\mathbb{Z}^n$ is a simple lattice.

- A lattice Λ is a discrete subgroup of $\mathbb{R}^n$.
- Can express as a linear transformation of the integer vectors,

$$\Lambda = \mathbf{B}\mathbb{Z}^n ,$$

for some (non-unique) $\mathbf{B} \in \mathbb{R}^{n \times n}$.



Lattice Properties

- Closed under addition: $\lambda_1, \lambda_2 \in \Lambda \implies \lambda_1 + \lambda_2 \in \Lambda$.
- Symmetric: $\lambda \in \Lambda \implies -\lambda \in \Lambda$

Voronoi Regions

- Nearest neighbor quantizer:

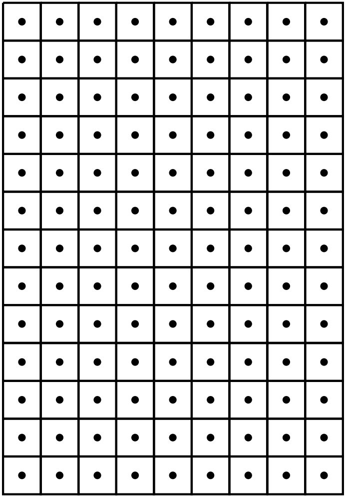
$$Q_\Lambda(\mathbf{x}) = \arg\min_{\lambda \in \Lambda} \|\mathbf{x} - \lambda\|_2$$

- The Voronoi region of a lattice point is the set of all points that quantize to that lattice point.

- Fundamental Voronoi region $\mathcal{V}$: points that quantize to the origin,

$$\mathcal{V} = \left\{\mathbf{x} \in \mathbb{R}^n : Q_\Lambda(\mathbf{x}) = \mathbf{0}\right\}$$

- Each Voronoi region is just a shift of the fundamental Voronoi region $\mathcal{V}$



Voronoi Regions

- Nearest neighbor quantizer:

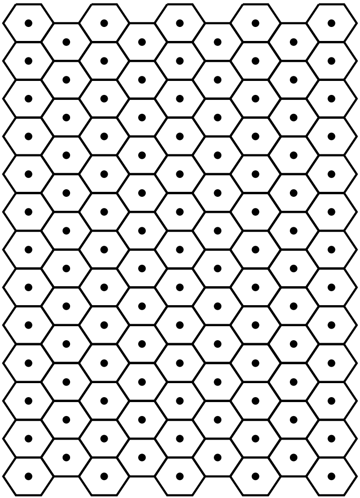
$$Q_\Lambda(\mathbf{x}) = \arg\min_{\lambda \in \Lambda} \|\mathbf{x} - \lambda\|_2$$

- The Voronoi region of a lattice point is the set of all points that quantize to that lattice point.

- Fundamental Voronoi region $\mathcal{V}$: points that quantize to the origin,

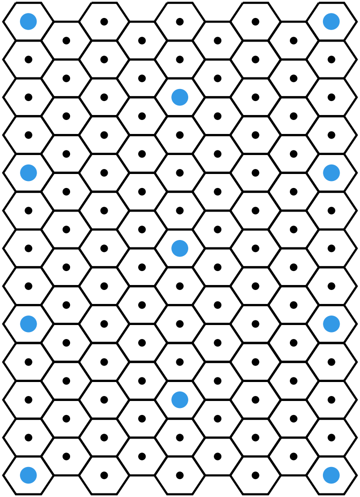
$$\mathcal{V} = \left\{\mathbf{x} \in \mathbb{R}^n : Q_\Lambda(\mathbf{x}) = \mathbf{0}\right\}$$

- Each Voronoi region is just a shift of the fundamental Voronoi region $\mathcal{V}$

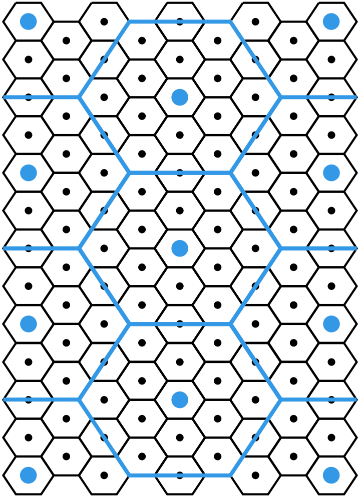


Nested Lattices

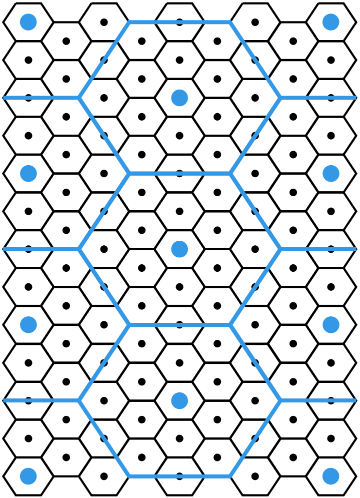
- Two lattices Λ_C and Λ_F are nested if $\Lambda_C \subset \Lambda_F$



- Two lattices Λ_C and Λ_F are nested if $\Lambda_C \subset \Lambda_F$



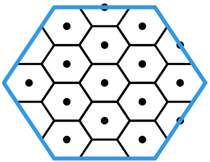
- Two lattices Λ_C and Λ_F are nested if $\Lambda_C \subset \Lambda_F$



Nested Lattices

- Two lattices Λ_{C} and Λ_{F} are nested if $\Lambda_{\mathsf{C}} \subset \Lambda_{\mathsf{F}}$

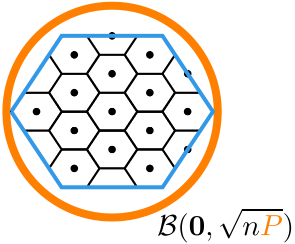
- Nested Lattice Code: All lattice points from Λ_{F} that fall in the fundamental Voronoi region $\mathcal{V}_{\mathsf{C}}$ of Λ_{C}.



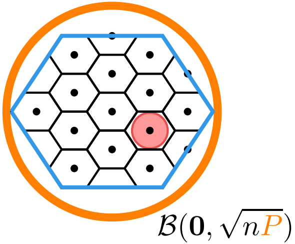
- Two lattices Λ_{C} and Λ_{F} are nested if $\Lambda_{\mathsf{C}} \subset \Lambda_{\mathsf{F}}$

- Nested Lattice Code: All lattice points from Λ_{F} that fall in the fundamental Voronoi region $\mathcal{V}_{\mathsf{C}}$ of Λ_{C}.

- Coarse lattice Λ_{C} enforces the power constraint.



- Two lattices Λ_{C} and Λ_{F} are nested if $\Lambda_{\mathsf{C}} \subset \Lambda_{\mathsf{F}}$
- Nested Lattice Code: All lattice points from Λ_{F} that fall in the fundamental Voronoi region $\mathcal{V}_{\mathsf{C}}$ of Λ_{C}.
- Coarse lattice Λ_{C} enforces the power constraint.
- Fine lattice Λ_{F} protects against noise.



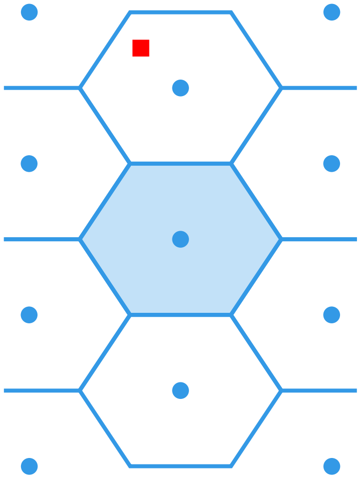
Modulo Operation

- Modulo operation with respect to lattice Λ is just the residual quantization error,

$$[\mathbf{x}] \bmod \Lambda = \mathbf{x} - Q_\Lambda(\mathbf{x}) \ .$$

- Modulo operation with respect to lattice Λ is just the residual quantization error,

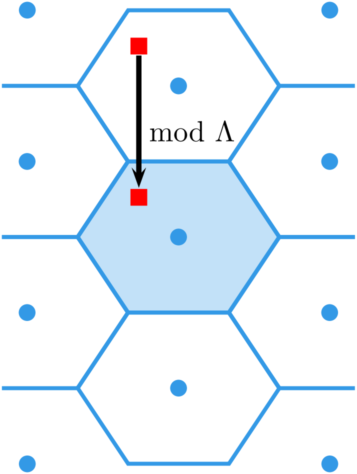
$$[\mathbf{x}] \bmod \Lambda = \mathbf{x} - Q_{\Lambda}(\mathbf{x}) \ .$$



Modulo Operation

- Modulo operation with respect to lattice Λ is just the residual quantization error,

$$[\mathbf{x}] \bmod \Lambda = \mathbf{x} - Q_\Lambda(\mathbf{x}) \ .$$



Modulo Operation

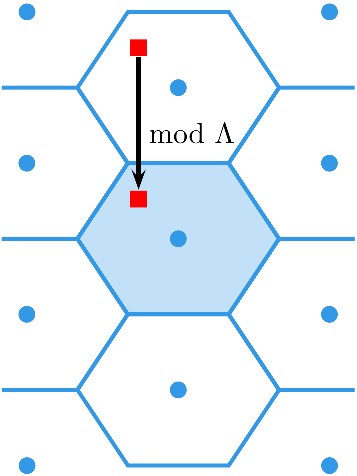
- Modulo operation with respect to lattice Λ is just the residual quantization error,

$$[\mathbf{x}] \bmod \Lambda = \mathbf{x} - Q_\Lambda(\mathbf{x}) \ .$$

- Distributive Law:

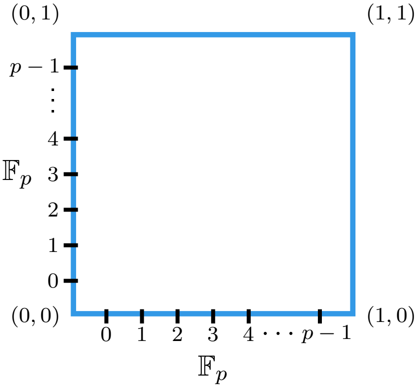
$$\Big[a_1[\mathbf{x}_1] \bmod \Lambda + a_2[\mathbf{x}_2] \bmod \Lambda\Big] \bmod \Lambda$$
$$= [a_1\mathbf{x}_1 + a_2\mathbf{x}_2] \bmod \Lambda$$

for any $a_1, a_2 \in \mathbb{Z}$ and $\mathbf{x}_1, \mathbf{x}_2 \in \mathbb{R}^n$.

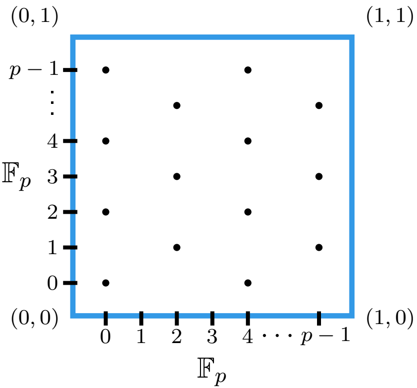


Construction A: Lattice Codes from Linear Codes

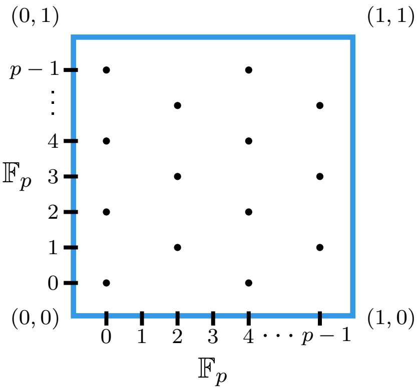
- Map elements $\{0, 1, 2, \ldots, p-1\}$ to equally spaced points on $[0, 1)$.



- Map elements $\{0, 1, 2, \ldots, p-1\}$ to equally spaced points on $[0, 1)$.
- Choose generator matrix $\mathbf{G} \in \mathbb{F}_p^{n \times k}$ and place its codewords into the unit cube $[0, 1)^n$. Tile over $\mathbb{Z}^n$ and scale by γ to get fine lattice Λ_{F}.

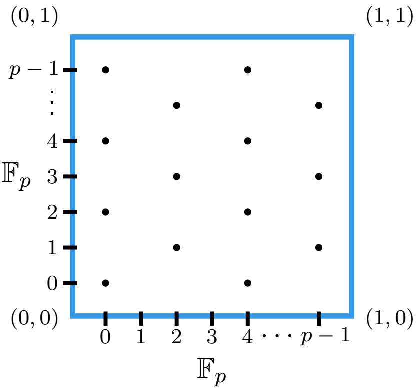


- Map elements $\{0, 1, 2, \ldots, p-1\}$ to equally spaced points on $[0, 1)$.
- Choose generator matrix $\mathbf{G} \in \mathbb{F}_p^{n \times k}$ and place its codewords into the unit cube $[0, 1)^n$. Tile over $\mathbb{Z}^n$ and scale by γ to get fine lattice Λ_{F}.
- Generator matrix usually elementwise i.i.d. uniform over $\mathbb{F}_p$.



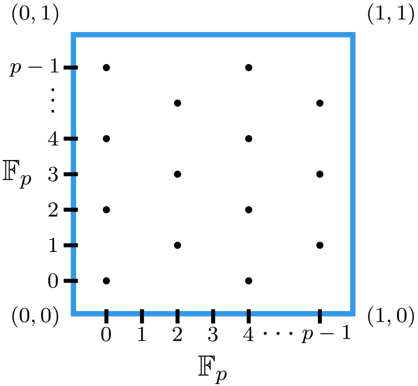
Construction A: Lattice Codes from Linear Codes

- Map elements $\{0, 1, 2, \ldots, p-1\}$ to equally spaced points on $[0, 1)$.
- Choose generator matrix $\mathbf{G} \in \mathbb{F}_p^{n \times k}$ and place its codewords into the unit cube $[0, 1)^n$. Tile over $\mathbb{Z}^n$ and scale by γ to get fine lattice Λ_{F}.
- Generator matrix usually elementwise i.i.d. uniform over $\mathbb{F}_p$.
- Scaled integers act as coarse lattice, $\Lambda_{\mathsf{C}} = \gamma \mathbb{Z}^n$.



Construction A: Lattice Codes from Linear Codes

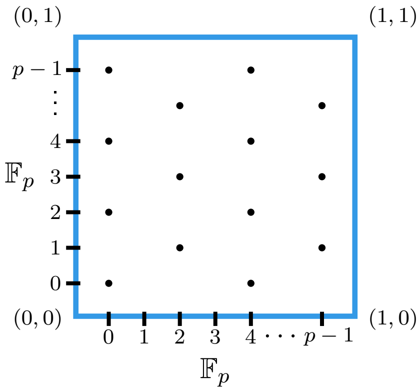
- Map elements $\{0, 1, 2, \ldots, p-1\}$ to equally spaced points on $[0, 1)$.
- Choose generator matrix $\mathbf{G} \in \mathbb{F}_p^{n \times k}$ and place its codewords into the unit cube $[0, 1)^n$. Tile over $\mathbb{Z}^n$ and scale by γ to get fine lattice Λ_{F}.
- Generator matrix usually elementwise i.i.d. uniform over $\mathbb{F}_p$.
- Scaled integers act as coarse lattice, $\Lambda_{\mathsf{C}} = \gamma \mathbb{Z}^n$.



- Can design good coarse lattice via a *two-stage approach*.

Construction A: Lattice Codes from Linear Codes

- Map elements $\{0, 1, 2, \ldots, p-1\}$ to equally spaced points on $[0, 1)$.
- Choose generator matrix $\mathbf{G} \in \mathbb{F}_p^{n \times k}$ and place its codewords into the unit cube $[0, 1)^n$. Tile over $\mathbb{Z}^n$ and scale by γ to get fine lattice Λ_{F}.
- Generator matrix usually elementwise i.i.d. uniform over $\mathbb{F}_p$.
- Scaled integers act as coarse lattice, $\Lambda_{\mathsf{C}} = \gamma \mathbb{Z}^n$.



- Can design good coarse lattice via a *two-stage approach.*
- Existence of good nested lattice codes via Construction A:
  Loeliger '97, Forney-Trott-Chung '00, Erez-Zamir '04, Erez-Litsyn-Zamir '05.

Nested Construction A

- **Ordentlich-Erez '12:** Use nested linear code in Construction A to directly obtain a good nested lattice code.

Nested Construction A

- **Ordentlich-Erez '12:** Use nested linear code in Construction A to directly obtain a good nested lattice code.
- Let $\mathbf{G}_{\mathsf{F}}$ be the $n \times k_{\mathsf{F}}$ generator matrix for the fine lattice Λ_{F}. Choose entries elementwise i.i.d. uniform over $\mathbb{F}_p$.

Nested Construction A

- **Ordentlich-Erez '12:** Use nested linear code in Construction A to directly obtain a good nested lattice code.
- Let $\mathbf{G}_{\mathsf{F}}$ be the $n \times k_{\mathsf{F}}$ generator matrix for the fine lattice Λ_{F}. Choose entries elementwise i.i.d. uniform over $\mathbb{F}_p$.
- Let $\mathbf{G}_{\mathsf{C}}$ be the $n \times k_{\mathsf{C}}$ generator matrix for the coarse lattice Λ_{C}. Set it to be equal to the first k_{C} columns of $\mathbf{G}_{\mathsf{F}}$.

Nested Construction A

- **Ordentlich-Erez '12:** Use nested linear code in Construction A to directly obtain a good nested lattice code.
- Let $\mathbf{G}_{\mathsf{F}}$ be the $n \times k_{\mathsf{F}}$ generator matrix for the fine lattice Λ_{F}. Choose entries elementwise i.i.d. uniform over $\mathbb{F}_p$.
- Let $\mathbf{G}_{\mathsf{C}}$ be the $n \times k_{\mathsf{C}}$ generator matrix for the coarse lattice Λ_{C}. Set it to be equal to the first k_{C} columns of $\mathbf{G}_{\mathsf{F}}$.
- Generate nested lattices $\Lambda_{\mathsf{C}} \subset \Lambda_{\mathsf{F}}$ by using $\mathbf{G}_{\mathsf{F}}$ in Construction A.

Nested Construction A

- **Ordentlich-Erez '12:** Use nested linear code in Construction A to directly obtain a good nested lattice code.
- Let $\mathbf{G}_{\mathsf{F}}$ be the $n \times k_{\mathsf{F}}$ generator matrix for the fine lattice Λ_{F}. Choose entries elementwise i.i.d. uniform over $\mathbb{F}_p$.
- Let $\mathbf{G}_{\mathsf{C}}$ be the $n \times k_{\mathsf{C}}$ generator matrix for the coarse lattice Λ_{C}. Set it to be equal to the first k_{C} columns of $\mathbf{G}_{\mathsf{F}}$.
- Generate nested lattices $\Lambda_{\mathsf{C}} \subset \Lambda_{\mathsf{F}}$ by using $\mathbf{G}_{\mathsf{F}}$ in Construction A.
- Ideally, the resulting code meets the power constraint and tolerates effective noise, while maintaining a high rate. We would also like an isomorphism to $\mathbb{F}_p^k$.

Nested Construction A

Fix P and σ^2_{eff}. It can be shown that, for any $\epsilon > 0$ and n large enough, there are choices of k_{F}, k_{C}, and p such that

Nested Construction A

Fix P and σ^2_{eff}. It can be shown that, for any $\epsilon > 0$ and n large enough, there are choices of k_{F}, k_{C}, and p such that

- Power constraint satisfied: If $\mathbf{x} \sim \mathsf{Unif}(\mathcal{V}_{\mathsf{C}})$, then $\frac{1}{n}\mathbb{E}\|\mathbf{x}\|^2 = P$.

Nested Construction A

Fix P and σ^2_{eff}. It can be shown that, for any $\epsilon > 0$ and n large enough, there are choices of k_{F}, k_{C}, and p such that

- Power constraint satisfied: If $\mathbf{x} \sim \mathsf{Unif}(\mathcal{V}_{\mathsf{C}})$, then $\frac{1}{n}\mathbb{E}\|\mathbf{x}\|^2 = P$.
- Noise tolerance satisfied: If $\mathbf{z}_{\mathsf{eff}}$ is effective noise satisfying $\mathbb{P}(\frac{1}{n}\|\mathbf{z}_{\mathsf{eff}}\|^2 > \sigma^2_{\mathsf{eff}}) < \epsilon$, then $\mathbb{P}\big(Q_{\Lambda_{\mathsf{F}}}(\mathbf{t} + \mathbf{z}_{\mathsf{eff}}) = \mathbf{t}\big) > 1 - \epsilon$ for any fine lattice point $\mathbf{t} \in \Lambda_{\mathsf{F}}$.

Fix P and σ^2_{eff}. It can be shown that, for any $\epsilon > 0$ and n large enough, there are choices of k_{F}, k_{C}, and p such that

- Power constraint satisfied: If $\mathbf{x} \sim \text{Unif}(\mathcal{V}_{\mathsf{C}})$, then $\frac{1}{n}\mathbb{E}\|\mathbf{x}\|^2 = P$.
- Noise tolerance satisfied: If $\mathbf{z}_{\text{eff}}$ is effective noise satisfying $\mathbb{P}(\frac{1}{n}\|\mathbf{z}_{\text{eff}}\|^2 > \sigma^2_{\text{eff}}) < \epsilon$, then $\mathbb{P}\big(Q_{\Lambda_{\mathsf{F}}}(\mathbf{t} + \mathbf{z}_{\text{eff}}) = \mathbf{t}\big) > 1 - \epsilon$ for any fine lattice point $\mathbf{t} \in \Lambda_{\mathsf{F}}$.
- Rate target satisfied: Number of useable symbols is $k = k_{\mathsf{F}} - k_{\mathsf{C}}$.

$$R = \frac{k}{n}\log p \; > \; \frac{1}{2}\log\left(\frac{P}{\sigma^2_{\text{eff}}}\right) - \epsilon \ .$$

Nested Construction A

Fix P and σ^2_{eff}. It can be shown that, for any $\epsilon > 0$ and n large enough, there are choices of k_{F}, k_{C}, and p such that

- Power constraint satisfied: If $\mathbf{x} \sim \text{Unif}(\mathcal{V}_{\mathsf{C}})$, then $\frac{1}{n}\mathbb{E}\|\mathbf{x}\|^2 = P$.
- Noise tolerance satisfied: If $\mathbf{z}_{\text{eff}}$ is effective noise satisfying $\mathbb{P}(\frac{1}{n}\|\mathbf{z}_{\text{eff}}\|^2 > \sigma^2_{\text{eff}}) < \epsilon$, then $\mathbb{P}\big(Q_{\Lambda_{\mathsf{F}}}(\mathbf{t} + \mathbf{z}_{\text{eff}}) = \mathbf{t}\big) > 1 - \epsilon$ for any fine lattice point $\mathbf{t} \in \Lambda_{\mathsf{F}}$.
- Rate target satisfied: Number of useable symbols is $k = k_{\mathsf{F}} - k_{\mathsf{C}}$.

$$R = \frac{k}{n} \log p \ > \ \frac{1}{2} \log\left(\frac{P}{\sigma^2_{\text{eff}}}\right) - \epsilon \ .$$

- Isomorphism exists: There is a function $\phi : \mathbb{F}_p^k \rightarrow \Lambda_{\mathsf{F}}/\Lambda_{\mathsf{C}}$ such that if $\mathbf{t}_\ell = \phi(\mathbf{w}_\ell)$, then

$$\phi^{-1}\left(\left[\sum_{\ell=1}^{L} a_\ell \mathbf{t}_\ell\right] \bmod \Lambda_{\mathsf{C}}\right) = \bigoplus_{\ell=1}^{L} q_\ell \mathbf{w}_\ell$$

for any $a_\ell \in \mathbb{Z}$ and $q_\ell = [a_\ell] \bmod p$.

Point-to-Point AWGN Channel: Lattice Encoding and Decoding

Before returning to the compute-and-forward problem, let's revisit the results of **Erez-Zamir '04** for point-to-point AWGN channels.

Point-to-Point AWGN Channel: Lattice Encoding and Decoding

Before returning to the compute-and-forward problem, let's revisit the results of **Erez-Zamir '04** for point-to-point AWGN channels.

- **Map message to lattice point:** $\mathbf{t} = \phi(\mathbf{w})$.

Before returning to the compute-and-forward problem, let's revisit the results of **Erez-Zamir '04** for point-to-point AWGN channels.

- **Map message to lattice point:** $\mathbf{t} = \phi(\mathbf{w})$.
- **Transmit:** $\mathbf{x} = \mathbf{t}$.

Before returning to the compute-and-forward problem, let's revisit the results of **Erez-Zamir '04** for point-to-point AWGN channels.

- **Map message to lattice point:** $\mathbf{t} = \phi(\mathbf{w})$.
- **Transmit:** $\mathbf{x} = \mathbf{t}$.
- **Receive:** $\mathbf{y} = \mathbf{x} + \mathbf{z}$.

Point-to-Point AWGN Channel: Lattice Encoding and Decoding

Before returning to the compute-and-forward problem, let's revisit the results of **Erez-Zamir '04** for point-to-point AWGN channels.

- **Map message to lattice point:** $\mathbf{t} = \phi(\mathbf{w})$.
- **Transmit:** $\mathbf{x} = \mathbf{t}$.
- **Receive:** $\mathbf{y} = \mathbf{x} + \mathbf{z}$.
- **Decode:** $\hat{\mathbf{t}} = \big[Q_{\Lambda_{\mathsf{F}}}(\mathbf{y})\big] \bmod \Lambda_{\mathsf{C}}$.

Point-to-Point AWGN Channel: Lattice Encoding and Decoding

Before returning to the compute-and-forward problem, let's revisit the results of **Erez-Zamir '04** for point-to-point AWGN channels.

- **Map message to lattice point:** $\mathbf{t} = \phi(\mathbf{w})$.
- **Transmit:** $\mathbf{x} = \mathbf{t}$.
- **Receive:** $\mathbf{y} = \mathbf{x} + \mathbf{z}$.
- **Decode:** $\hat{\mathbf{t}} = \big[Q_{\Lambda_{\mathsf{F}}}(\mathbf{y})\big] \bmod \Lambda_{\mathsf{C}}$.
- **Map back to finite field:** $\hat{\mathbf{w}} = \phi^{-1}(\hat{\mathbf{t}})$.

Point-to-Point AWGN Channel: Lattice Encoding and Decoding

Before returning to the compute-and-forward problem, let's revisit the results of **Erez-Zamir '04** for point-to-point AWGN channels.

- **Map message to lattice point:** $\mathbf{t} = \phi(\mathbf{w})$.
- **Transmit:** $\mathbf{x} = \mathbf{t}$.
- **Receive:** $\mathbf{y} = \mathbf{x} + \mathbf{z}$.
- **Decode:** $\hat{\mathbf{t}} = \big[Q_{\Lambda_{\mathsf{F}}}(\mathbf{y})\big] \bmod \Lambda_{\mathsf{C}}$.
- **Map back to finite field:** $\hat{\mathbf{w}} = \phi^{-1}(\hat{\mathbf{t}})$.
- Decoding is successful with high probability if we set $\sigma^2_{\text{eff}} > 1$. This means that the rate $R = \frac{1}{2}\log(P)$ is achievable.

Point-to-Point AWGN Channel: Lattice Encoding and Decoding

Before returning to the compute-and-forward problem, let's revisit the results of **Erez-Zamir '04** for point-to-point AWGN channels.

- **Map message to lattice point:** $\mathbf{t} = \phi(\mathbf{w})$.
- **Transmit:** $\mathbf{x} = \mathbf{t}$.
- **Receive:** $\mathbf{y} = \mathbf{x} + \mathbf{z}$.
- **Decode:** $\hat{\mathbf{t}} = \big[Q_{\Lambda_{\mathsf{F}}}(\mathbf{y})\big] \bmod \Lambda_{\mathsf{C}}$.
- **Map back to finite field:** $\hat{\mathbf{w}} = \phi^{-1}(\hat{\mathbf{t}})$.
- Decoding is successful with high probability if we set $\sigma^2_{\text{eff}} > 1$. This means that the rate $R = \frac{1}{2}\log(P)$ is achievable.
- What happened to the "$1\,+$"?

MMSE Scaling

- It turns out that we can do better by *scaling* the channel output prior to decoding.

$$\begin{aligned} \alpha \mathbf{y} &= \alpha \mathbf{x} + \alpha \mathbf{z} \\ &= \mathbf{x} + \underbrace{(\alpha - 1)\mathbf{x} + \alpha \mathbf{z}}_{\mathbf{z}_{\text{eff}}} \end{aligned}$$

MMSE Scaling

- It turns out that we can do better by *scaling* the channel output prior to decoding.

$$\begin{aligned} \alpha\mathbf{y} &= \alpha\mathbf{x} + \alpha\mathbf{z} \\ &= \mathbf{x} + \underbrace{(\alpha - 1)\mathbf{x} + \alpha\mathbf{z}}_{\mathbf{z}_{\text{eff}}} \end{aligned}$$

- The effective noise variance $\frac{1}{n}\mathbb{E}\|\mathbf{z}_{\text{eff}}\|^2 = (\alpha - 1)^2 P + \alpha^2$ is uniquely minimized by the MMSE coefficient $\alpha_{\text{MMSE}} = P/(1 + P)$,

$$\begin{aligned} \min_{\alpha \in \mathbb{R}} \frac{1}{n}\mathbb{E}\|\mathbf{z}_{\text{eff}}\|^2 &= (\alpha_{\text{MMSE}} - 1)^2 P + \alpha_{\text{MMSE}}^2 \\ &= \frac{P}{1 + P} \end{aligned}$$

MMSE Scaling

- It turns out that we can do better by *scaling* the channel output prior to decoding.

$$\begin{aligned} \alpha \mathbf{y} &= \alpha \mathbf{x} + \alpha \mathbf{z} \\ &= \mathbf{x} + \underbrace{(\alpha - 1)\mathbf{x} + \alpha \mathbf{z}}_{\mathbf{z}_{\text{eff}}} \end{aligned}$$

- The effective noise variance $\frac{1}{n}\mathbb{E}\|\mathbf{z}_{\text{eff}}\|^2 = (\alpha - 1)^2 P + \alpha^2$ is uniquely minimized by the MMSE coefficient $\alpha_{\text{MMSE}} = P/(1+P)$,

$$\begin{aligned} \min_{\alpha \in \mathbb{R}} \frac{1}{n}\mathbb{E}\|\mathbf{z}_{\text{eff}}\|^2 &= (\alpha_{\text{MMSE}} - 1)^2 P + \alpha_{\text{MMSE}}^2 \\ &= \frac{P}{1+P} \end{aligned}$$

- Plugging this in as σ_{eff}^2, we find that $\frac{1}{2}\log\left(\frac{P}{\sigma_{\text{eff}}^2}\right) = \frac{1}{2}\log(1+P)$.

MMSE Scaling

- It turns out that we can do better by *scaling* the channel output prior to decoding.

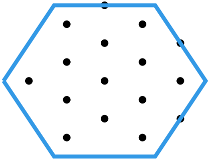
$$\begin{aligned}\alpha\mathbf{y} &= \alpha\mathbf{x} + \alpha\mathbf{z} \\ &= \mathbf{x} + \underbrace{(\alpha - 1)\mathbf{x} + \alpha\mathbf{z}}_{\mathbf{z}_{\text{eff}}}\end{aligned}$$

- The effective noise variance $\frac{1}{n}\mathbb{E}\|\mathbf{z}_{\text{eff}}\|^2 = (\alpha - 1)^2 P + \alpha^2$ is uniquely minimized by the MMSE coefficient $\alpha_{\text{MMSE}} = P/(1+P)$,

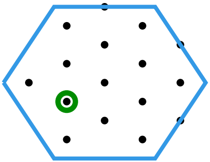
$$\begin{aligned}\min_{\alpha\in\mathbb{R}} \frac{1}{n}\mathbb{E}\|\mathbf{z}_{\text{eff}}\|^2 &= (\alpha_{\text{MMSE}} - 1)^2 P + \alpha_{\text{MMSE}}^2 \\ &= \frac{P}{1+P}\end{aligned}$$

- Plugging this in as σ_{eff}^2, we find that $\frac{1}{2}\log\left(\frac{P}{\sigma_{\text{eff}}^2}\right) = \frac{1}{2}\log(1+P)$.
- But what about the dependency between the codeword and the effective noise?

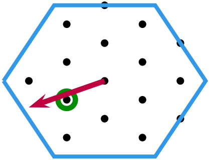
- Dithering can make the effective noise look independent from the desired lattice codeword.



- Dithering can make the effective noise look independent from the desired lattice codeword.
- Map message $\mathbf{w}$ to a lattice codeword $\mathbf{t} \in \Lambda_{\mathsf{F}}$.

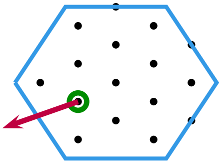


- Dithering can make the effective noise look independent from the desired lattice codeword.
- Map message $\mathbf{w}$ to a lattice codeword $\mathbf{t} \in \Lambda_{\mathsf{F}}$.
- Generate a random dither vector $\mathbf{d}$ uniformly over $\mathcal{V}_{\mathsf{C}}$.



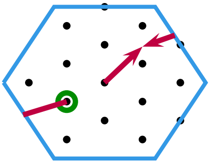
- Dithering can make the effective noise look independent from the desired lattice codeword.
- Map message $\mathbf{w}$ to a lattice codeword $\mathbf{t} \in \Lambda_{\mathsf{F}}$.
- Generate a random dither vector $\mathbf{d}$ uniformly over $\mathcal{V}_{\mathsf{C}}$.
- Transmitter sends a dithered codeword:

$$\mathbf{x}_\ell = [\mathbf{t}_\ell + \mathbf{d}_\ell] \bmod \Lambda_{\mathsf{C}}$$



- Dithering can make the effective noise look independent from the desired lattice codeword.
- Map message $\mathbf{w}$ to a lattice codeword $\mathbf{t} \in \Lambda_{\mathsf{F}}$.
- Generate a random dither vector $\mathbf{d}$ uniformly over $\mathcal{V}_{\mathsf{C}}$.
- Transmitter sends a dithered codeword:

$$\mathbf{x}_\ell = [\mathbf{t}_\ell + \mathbf{d}_\ell] \bmod \Lambda_{\mathsf{C}}$$

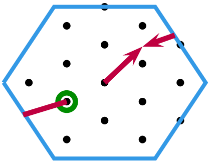


Dithering

- Dithering can make the effective noise look independent from the desired lattice codeword.
- Map message $\mathbf{w}$ to a lattice codeword $\mathbf{t} \in \Lambda_{\mathsf{F}}$.
- Generate a random dither vector $\mathbf{d}$ uniformly over $\mathcal{V}_{\mathsf{C}}$.
- Transmitter sends a dithered codeword:

$$\mathbf{x}_\ell = [\mathbf{t}_\ell + \mathbf{d}_\ell] \bmod \Lambda_{\mathsf{C}}$$

- $\mathbf{x}$ is now independent of the codeword $\mathbf{t}$.



Point-to-Point AWGN Channel: Lattice Encoding and Decoding

With MMSE scaling and dithering, we can reach the AWGN capacity.

Point-to-Point AWGN Channel: Lattice Encoding and Decoding

With MMSE scaling and dithering, we can reach the AWGN capacity.

- **Map message to lattice point:** $\mathbf{t} = \phi(\mathbf{w})$.

Point-to-Point AWGN Channel: Lattice Encoding and Decoding

With MMSE scaling and dithering, we can reach the AWGN capacity.

- **Map message to lattice point:** $\mathbf{t} = \phi(\mathbf{w})$.
- **Dither + Transmit:** $\mathbf{x} = [\mathbf{t} + \mathbf{d}] \bmod \Lambda_{\mathsf{C}}$.

Point-to-Point AWGN Channel: Lattice Encoding and Decoding

With MMSE scaling and dithering, we can reach the AWGN capacity.

- **Map message to lattice point:** $\mathbf{t} = \phi(\mathbf{w})$.
- **Dither + Transmit:** $\mathbf{x} = [\mathbf{t} + \mathbf{d}] \bmod \Lambda_{\mathsf{C}}$.
- **Receive:** $\mathbf{y} = \mathbf{x} + \mathbf{z}$.

Point-to-Point AWGN Channel: Lattice Encoding and Decoding

With MMSE scaling and dithering, we can reach the AWGN capacity.

- **Map message to lattice point:** $\mathbf{t} = \phi(\mathbf{w})$.
- **Dither + Transmit:** $\mathbf{x} = [\mathbf{t} + \mathbf{d}] \bmod \Lambda_{\mathsf{C}}$.
- **Receive:** $\mathbf{y} = \mathbf{x} + \mathbf{z}$.
- **Scale + Remove dithers:**

$$\begin{aligned}
& & \tilde{\mathbf{y}} &= [\alpha \mathbf{y} - \mathbf{d}] \bmod \Lambda_{\mathsf{C}} \\
&(\mathbf{z}_{\text{eff}} = (\alpha - 1)\mathbf{x} + \alpha \mathbf{z}) & &= [\mathbf{x} + \mathbf{z}_{\text{eff}} - \mathbf{d}] \bmod \Lambda_{\mathsf{C}} \\
& & &= \big[[\mathbf{t} + \mathbf{d}] \bmod \Lambda_{\mathsf{C}} - \mathbf{d} + \mathbf{z}_{\text{eff}}\big] \bmod \Lambda_{\mathsf{C}} \\
&(\text{Distributive Law}) & &= [\mathbf{t} + \mathbf{z}_{\text{eff}}] \bmod \Lambda_{\mathsf{C}}
\end{aligned}$$

Point-to-Point AWGN Channel: Lattice Encoding and Decoding

With MMSE scaling and dithering, we can reach the AWGN capacity.

- **Map message to lattice point:** $\mathbf{t} = \phi(\mathbf{w})$.
- **Dither + Transmit:** $\mathbf{x} = [\mathbf{t} + \mathbf{d}] \bmod \Lambda_{\mathsf{C}}$.
- **Receive:** $\mathbf{y} = \mathbf{x} + \mathbf{z}$.
- **Scale + Remove dithers:**

$$
\begin{aligned}
\tilde{\mathbf{y}} &= [\alpha \mathbf{y} - \mathbf{d}] \bmod \Lambda_{\mathsf{C}} \\
(\mathbf{z}_{\mathsf{eff}} = (\alpha - 1)\mathbf{x} + \alpha \mathbf{z}) \qquad &= [\mathbf{x} + \mathbf{z}_{\mathsf{eff}} - \mathbf{d}] \bmod \Lambda_{\mathsf{C}} \\
&= \big[[\mathbf{t} + \mathbf{d}] \bmod \Lambda_{\mathsf{C}} - \mathbf{d} + \mathbf{z}_{\mathsf{eff}}\big] \bmod \Lambda_{\mathsf{C}} \\
(\text{Distributive Law}) \qquad &= [\mathbf{t} + \mathbf{z}_{\mathsf{eff}}] \bmod \Lambda_{\mathsf{C}}
\end{aligned}
$$

- **Decode:** $\hat{\mathbf{t}} = \big[Q_{\Lambda_{\mathsf{F}}}(\tilde{\mathbf{y}})\big] \bmod \Lambda_{\mathsf{C}}$.

Point-to-Point AWGN Channel: Lattice Encoding and Decoding

With MMSE scaling and dithering, we can reach the AWGN capacity.

- **Map message to lattice point:** $\mathbf{t} = \phi(\mathbf{w})$.
- **Dither + Transmit:** $\mathbf{x} = [\mathbf{t} + \mathbf{d}] \bmod \Lambda_{\mathsf{C}}$.
- **Receive:** $\mathbf{y} = \mathbf{x} + \mathbf{z}$.
- **Scale + Remove dithers:**

$$\begin{aligned}
& \tilde{\mathbf{y}} = [\alpha \mathbf{y} - \mathbf{d}] \bmod \Lambda_{\mathsf{C}} \\
(\mathbf{z}_{\mathsf{eff}} = (\alpha - 1)\mathbf{x} + \alpha \mathbf{z}) \quad & = [\mathbf{x} + \mathbf{z}_{\mathsf{eff}} - \mathbf{d}] \bmod \Lambda_{\mathsf{C}} \\
& = \big[[\mathbf{t} + \mathbf{d}] \bmod \Lambda_{\mathsf{C}} - \mathbf{d} + \mathbf{z}_{\mathsf{eff}}\big] \bmod \Lambda_{\mathsf{C}} \\
(\text{Distributive Law}) \quad & = [\mathbf{t} + \mathbf{z}_{\mathsf{eff}}] \bmod \Lambda_{\mathsf{C}}
\end{aligned}$$

- **Decode:** $\hat{\mathbf{t}} = \big[Q_{\Lambda_{\mathsf{F}}}(\tilde{\mathbf{y}})\big] \bmod \Lambda_{\mathsf{C}}$.
- **Map back to finite field:** $\hat{\mathbf{w}} = \phi^{-1}(\hat{\mathbf{t}})$.

Point-to-Point AWGN Channel: Lattice Encoding and Decoding

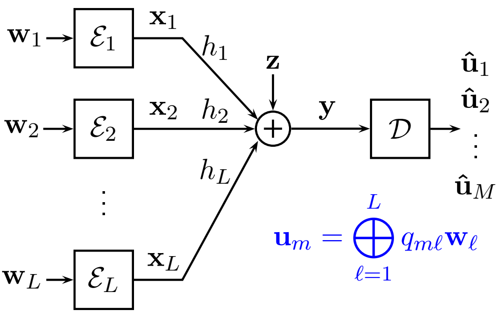
With MMSE scaling and dithering, we can reach the AWGN capacity.

- **Map message to lattice point:** $\mathbf{t} = \phi(\mathbf{w})$.
- **Dither + Transmit:** $\mathbf{x} = [\mathbf{t} + \mathbf{d}] \bmod \Lambda_{\mathsf{C}}$.
- **Receive:** $\mathbf{y} = \mathbf{x} + \mathbf{z}$.
- **Scale + Remove dithers:**

$$\begin{aligned}
& \tilde{\mathbf{y}} = [\alpha\mathbf{y} - \mathbf{d}] \bmod \Lambda_{\mathsf{C}} \\
(\mathbf{z}_{\mathsf{eff}} = (\alpha - 1)\mathbf{x} + \alpha\mathbf{z}) \quad & = [\mathbf{x} + \mathbf{z}_{\mathsf{eff}} - \mathbf{d}] \bmod \Lambda_{\mathsf{C}} \\
& = \big[[\mathbf{t} + \mathbf{d}] \bmod \Lambda_{\mathsf{C}} - \mathbf{d} + \mathbf{z}_{\mathsf{eff}}\big] \bmod \Lambda_{\mathsf{C}} \\
(\text{Distributive Law}) \quad & = [\mathbf{t} + \mathbf{z}_{\mathsf{eff}}] \bmod \Lambda_{\mathsf{C}}
\end{aligned}$$

- **Decode:** $\hat{\mathbf{t}} = \big[Q_{\Lambda_{\mathsf{F}}}(\tilde{\mathbf{y}})\big] \bmod \Lambda_{\mathsf{C}}$.
- **Map back to finite field:** $\hat{\mathbf{w}} = \phi^{-1}(\hat{\mathbf{t}})$.
- Decoding is successful with high probability if we set $\sigma^2_{\mathsf{eff}} > \frac{P}{1+P}$. This means that the rate $R = \frac{1}{2}\log(1+P)$ is achievable.

Refresher: Compute-and-Forward Problem Statement



$$\mathbf{u}_m = \bigoplus_{\ell=1}^{L} q_{m\ell} \mathbf{w}_\ell$$

- Messages are finite field vectors, $\mathbf{w}_\ell \in \mathbb{F}_p^k$.
- Real-valued inputs and outputs, $\mathbf{x}_\ell, \mathbf{y} \in \mathbb{R}^n$.
- Power constraint, $\frac{1}{n}\mathbb{E}\|\mathbf{x}_\ell\|^2 \leq P$.
- Gaussian noise, $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$.
- Equal rates: $R = \dfrac{k}{n} \log_2 p$

- Decoder wants M linear combinations of the messages with vanishing probability of error $\lim_{n \to \infty} \mathbb{P}\Big(\bigcup_m \{\hat{\mathbf{u}}_m \neq \mathbf{u}_m\} \Big) = 0$.
- The linear combination with integer coefficient vector $\mathbf{a}_m = [a_{m1} \ a_{m2} \ \cdots \ a_{mL}]^\mathsf{T} \in \mathbb{Z}^L$ corresponds to

$$\mathbf{u}_m = \bigoplus_{\ell=1}^{L} q_{m\ell} \mathbf{w}_\ell \qquad \text{where } q_{m\ell} = [a_{m\ell}] \bmod p \ .$$

Encoding operations at the ℓ^{th} transmitter:

Encoding operations at the ℓ^{th} transmitter:

- **Map message to lattice point:** $\mathbf{t}_\ell = \phi(\mathbf{w}_\ell)$.

Compute-and-Forward: Lattice Encoding

Encoding operations at the ℓ^{th} transmitter:

- **Map message to lattice point:** $\mathbf{t}_\ell = \phi(\mathbf{w}_\ell)$.
- **Dither + Transmit:** $\mathbf{x}_\ell = [\mathbf{t}_\ell + \mathbf{d}_\ell] \bmod \Lambda_{\mathsf{C}}$ where the dithers $\mathbf{d}_\ell$ are chosen independently and uniformly over $\mathcal{V}_{\mathsf{C}}$,

Encoding operations at the ℓ^{th} transmitter:

- **Map message to lattice point:** $\mathbf{t}_\ell = \phi(\mathbf{w}_\ell)$.
- **Dither + Transmit:** $\mathbf{x}_\ell = [\mathbf{t}_\ell + \mathbf{d}_\ell] \bmod \Lambda_{\mathsf{C}}$ where the dithers $\mathbf{d}_\ell$ are chosen independently and uniformly over $\mathcal{V}_{\mathsf{C}}$,
- Notice that these operations do not depend on the channel gains.

Decoding operations at the receiver to recover the linear combination with integer coefficient vector $\mathbf{a}_m = [a_{m1} \ \cdots \ a_{mL}]^{\mathsf{T}}$:

Compute-and-Forward: Lattice Decoding

Decoding operations at the receiver to recover the linear combination with integer coefficient vector $\mathbf{a}_m = [a_{m1} \ \cdots \ a_{mL}]^{\mathsf{T}}$:

- **Receive:** $\mathbf{y} = \sum_{\ell=1}^{L} h_\ell \mathbf{x}_\ell + \mathbf{z}$.

Compute-and-Forward: Lattice Decoding

Decoding operations at the receiver to recover the linear combination with integer coefficient vector $\mathbf{a}_m = [a_{m1} \ \cdots \ a_{mL}]^{\mathsf{T}}$:

- **Receive:** $\mathbf{y} = \sum_{\ell=1}^{L} h_\ell \mathbf{x}_\ell + \mathbf{z}$.
- **Scale + Remove dithers:**

$$\begin{aligned} \tilde{\mathbf{y}} &= \left[\alpha \mathbf{y} - \sum_{\ell=1}^{L} a_{m\ell} \mathbf{d}_\ell \right] \bmod \Lambda_{\mathsf{C}} \\ &= \left[\sum_{\ell=1}^{L} a_{m\ell}(\mathbf{x}_\ell - \mathbf{d}_\ell) + \sum_{\ell=1}^{L} (\alpha h_\ell - a_{m\ell}) \mathbf{x}_\ell + \alpha \mathbf{z} \right] \bmod \Lambda_{\mathsf{C}} \\ &= [\mathbf{v} + \mathbf{z}_{\mathsf{eff}}] \bmod \Lambda_{\mathsf{C}} \quad (\text{Distributive Law}) \end{aligned}$$

$$\mathbf{v} = \left[\sum_{\ell=1}^{L} a_{m\ell} \mathbf{t}_\ell \right] \bmod \Lambda_{\mathsf{C}} \qquad \mathbf{z}_{\mathsf{eff}} = \sum_{\ell=1}^{L} (\alpha h_\ell - a_{m\ell}) \mathbf{x}_\ell + \alpha \mathbf{z}$$

Compute-and-Forward: Lattice Decoding

Decoding operations at the receiver to recover the linear combination with integer coefficient vector $\mathbf{a}_m = [a_{m1} \ \cdots \ a_{mL}]^{\mathsf{T}}$:

- **Receive:** $\mathbf{y} = \sum_{\ell=1}^{L} h_\ell \mathbf{x}_\ell + \mathbf{z}$.
- **Scale + Remove dithers:**

$$\begin{aligned}
\tilde{\mathbf{y}} &= \left[\alpha \mathbf{y} - \sum_{\ell=1}^{L} a_{m\ell} \mathbf{d}_\ell \right] \bmod \Lambda_{\mathsf{C}} \\
&= \left[\sum_{\ell=1}^{L} a_{m\ell} (\mathbf{x}_\ell - \mathbf{d}_\ell) + \sum_{\ell=1}^{L} (\alpha h_\ell - a_{m\ell}) \mathbf{x}_\ell + \alpha \mathbf{z} \right] \bmod \Lambda_{\mathsf{C}} \\
&= [\mathbf{v} + \mathbf{z}_{\mathsf{eff}}] \bmod \Lambda_{\mathsf{C}} \quad \text{(Distributive Law)}
\end{aligned}$$

$$\mathbf{v} = \left[\sum_{\ell=1}^{L} a_{m\ell} \mathbf{t}_\ell \right] \bmod \Lambda_{\mathsf{C}} \qquad \mathbf{z}_{\mathsf{eff}} = \sum_{\ell=1}^{L} (\alpha h_\ell - a_{m\ell}) \mathbf{x}_\ell + \alpha \mathbf{z}$$

- **Decode:** $\hat{\mathbf{v}}_m = \left[Q_{\Lambda_{\mathsf{F}}}(\tilde{\mathbf{y}}) \right] \bmod \Lambda_{\mathsf{C}}$.

Compute-and-Forward: Lattice Decoding

Decoding operations at the receiver to recover the linear combination with integer coefficient vector $\mathbf{a}_m = [a_{m1} \ \cdots \ a_{mL}]^{\mathsf{T}}$:

- **Receive:** $\mathbf{y} = \sum_{\ell=1}^{L} h_\ell \mathbf{x}_\ell + \mathbf{z}$.
- **Scale + Remove dithers:**

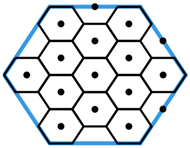
$$\begin{aligned} \tilde{\mathbf{y}} &= \left[\alpha \mathbf{y} - \sum_{\ell=1}^{L} a_{m\ell} \mathbf{d}_\ell \right] \bmod \Lambda_{\mathsf{C}} \\ &= \left[\sum_{\ell=1}^{L} a_{m\ell} (\mathbf{x}_\ell - \mathbf{d}_\ell) + \sum_{\ell=1}^{L} (\alpha h_\ell - a_{m\ell}) \mathbf{x}_\ell + \alpha \mathbf{z} \right] \bmod \Lambda_{\mathsf{C}} \\ &= [\mathbf{v} + \mathbf{z}_{\mathsf{eff}}] \bmod \Lambda_{\mathsf{C}} \quad (\text{Distributive Law}) \end{aligned}$$

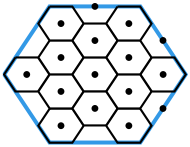
$$\mathbf{v} = \left[\sum_{\ell=1}^{L} a_{m\ell} \mathbf{t}_\ell \right] \bmod \Lambda_{\mathsf{C}} \qquad \mathbf{z}_{\mathsf{eff}} = \sum_{\ell=1}^{L} (\alpha h_\ell - a_{m\ell}) \mathbf{x}_\ell + \alpha \mathbf{z}$$

- **Decode:** $\hat{\mathbf{v}}_m = \left[Q_{\Lambda_{\mathsf{F}}}(\tilde{\mathbf{y}}) \right] \bmod \Lambda_{\mathsf{C}}$.
- **Map back to finite field:** $\hat{\mathbf{u}}_m = \phi^{-1}(\hat{\mathbf{v}})$.

Compute-and-Forward: Illustration

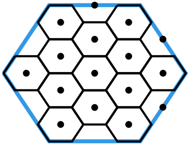
All users employ the same nested lattice code:

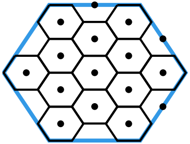




Compute-and-Forward: Illustration

Choose message vectors over finite field $\mathbf{w}_\ell \in \mathbb{F}_p^k$:

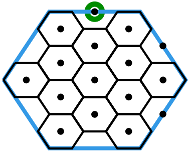
$\mathbf{w}_1 \longrightarrow$ 

$\mathbf{w}_2 \longrightarrow$ 

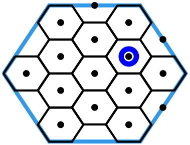
Compute-and-Forward: Illustration

Map $\mathbf{w}_\ell$ to lattice point $\mathbf{t}_\ell = \phi(\mathbf{w}_\ell)$:

$\mathbf{w}_1 \longrightarrow$

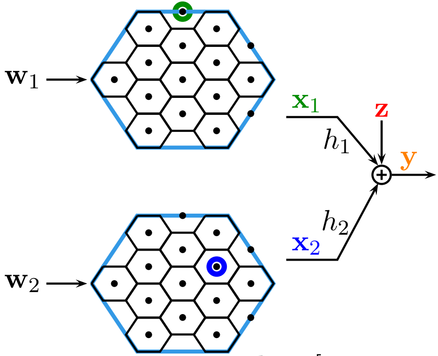


$\mathbf{w}_2 \longrightarrow$



Compute-and-Forward: Illustration

Transmit lattice points over the channel:

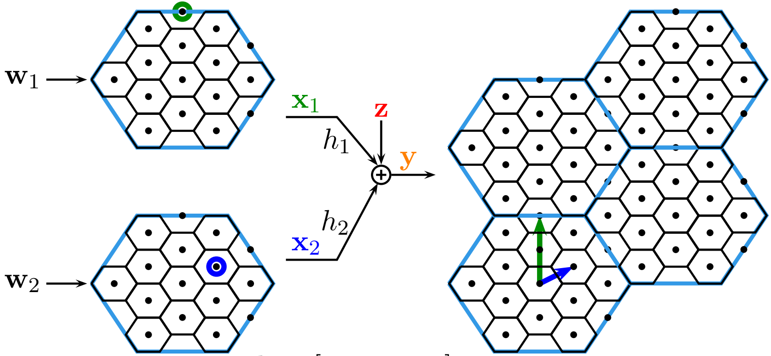


$$\mathbf{h} = [\ 1.4 \quad 2.1\ ]$$

$$\mathbf{a}_m = [\ 2 \quad 3\ ]$$

Compute-and-Forward: Illustration

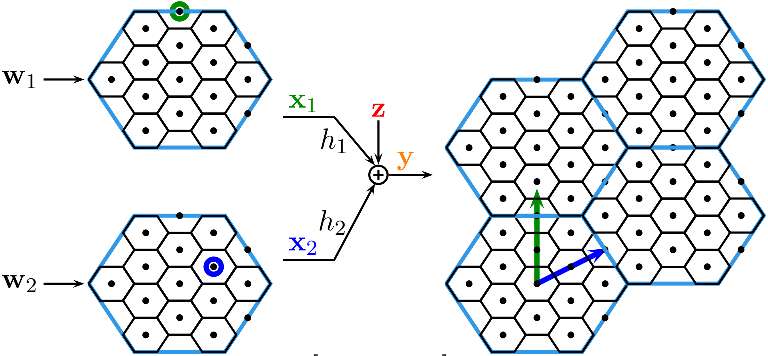
Transmit lattice points over the channel:



$$\mathbf{h} = [\ 1.4 \quad 2.1\ ]$$

$$\mathbf{a}_m = [\ 2 \quad 3\ ]$$

Lattice codewords are scaled by channel coefficients:

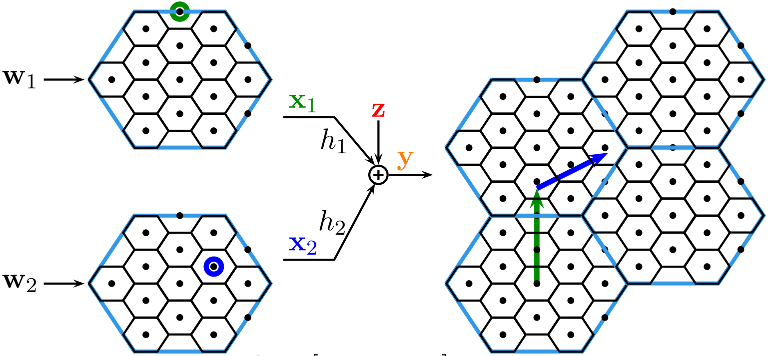


$$\mathbf{h} = [\ 1.4 \quad 2.1\ ]$$

$$\mathbf{a}_m = [\ 2 \quad 3\ ]$$

Compute-and-Forward: Illustration

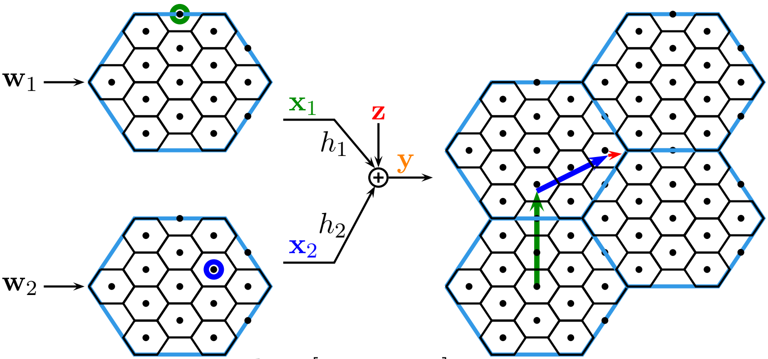
Scaled codewords added together plus noise:



$$\mathbf{h} = [\ 1.4 \quad 2.1\ ]$$

$$\mathbf{a}_m = [\ 2 \quad 3\ ]$$

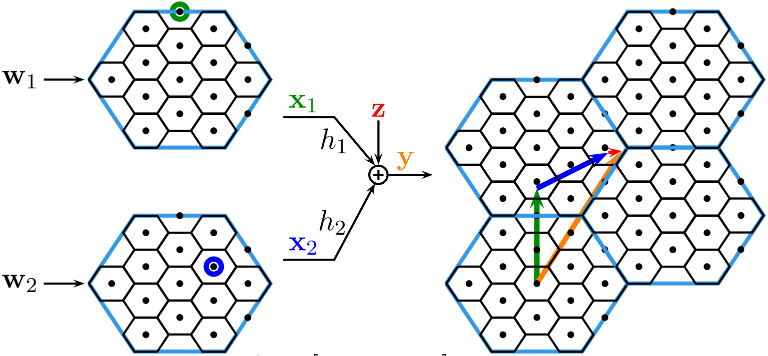
Scaled codewords added together plus noise:



$$\mathbf{h} = [\ 1.4 \quad 2.1\ ]$$

$$\mathbf{a}_m = [\ 2 \quad 3\ ]$$

Extra noise penalty for non-integer channel coefficients:



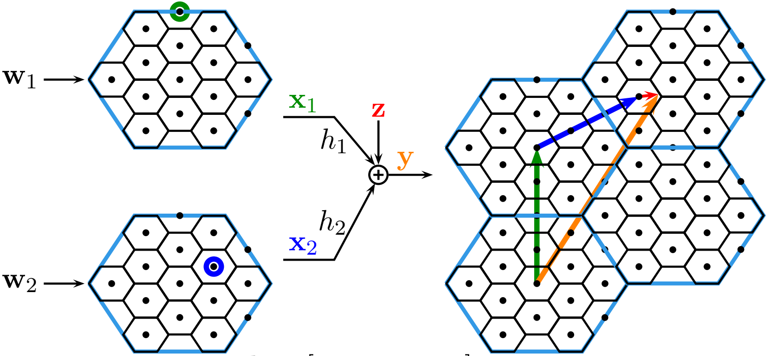
$$\mathbf{h} = [\ 1.4 \quad 2.1\ ]$$

$$\mathbf{a}_m = [\ 2 \quad 3\ ]$$

Effective noise: $1 + P\|\mathbf{h} - \mathbf{a}_m\|^2$

Compute-and-Forward: Illustration

Scale output by α to reduce non-integer noise penalty:



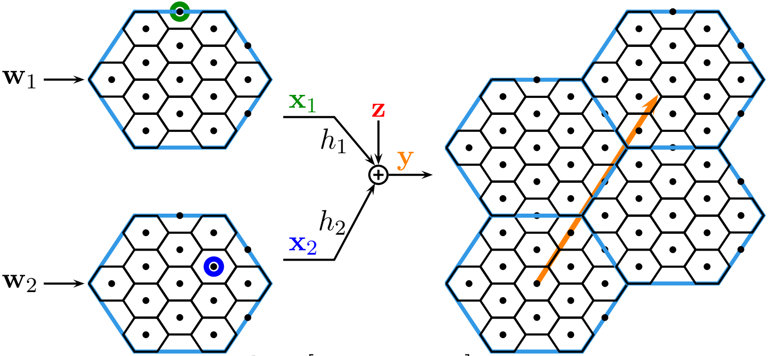
$$\alpha\mathbf{h} = [\ \alpha 1.4 \quad \alpha 2.1\ ]$$

$$\mathbf{a}_m = [\ \ 2 \qquad 3\ \ ]$$

Effective noise: $\alpha^2 + P\|\alpha\mathbf{h} - \mathbf{a}_m\|^2$

Compute-and-Forward: Illustration

Scale output by α to reduce non-integer noise penalty:



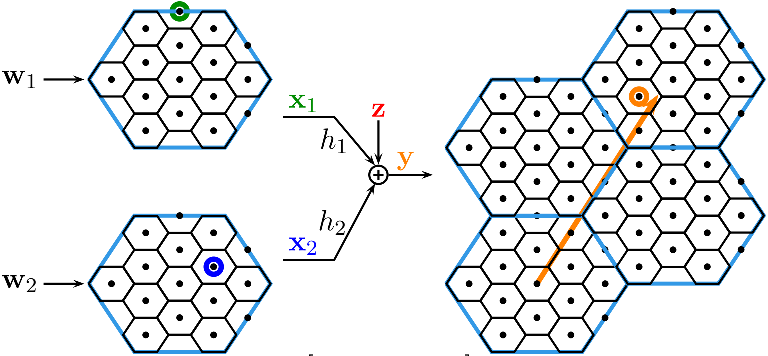
$$\alpha\mathbf{h} = [\ \alpha 1.4 \quad \alpha 2.1\ ]$$

$$\mathbf{a}_m = [\ 2 \quad 3\ ]$$

Effective noise: $\alpha^2 + P\|\alpha\mathbf{h} - \mathbf{a}_m\|^2$

Compute-and-Forward: Illustration

Decode to the closest lattice point:

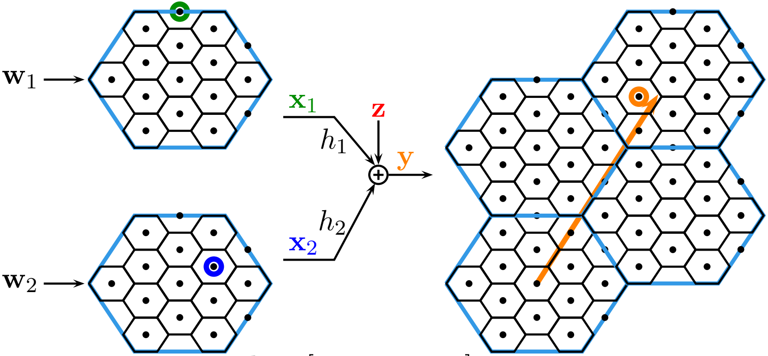


$$\alpha\mathbf{h} = [\ \alpha 1.4 \quad \alpha 2.1\ ]$$

$$\mathbf{a}_m = [\ \ 2 \qquad 3\ \ ]$$

Effective noise: $\alpha^2 + P\|\alpha\mathbf{h} - \mathbf{a}_m\|^2$

Recover integer linear combination $\mathrm{mod}\ \Lambda_{\mathsf{C}}$:



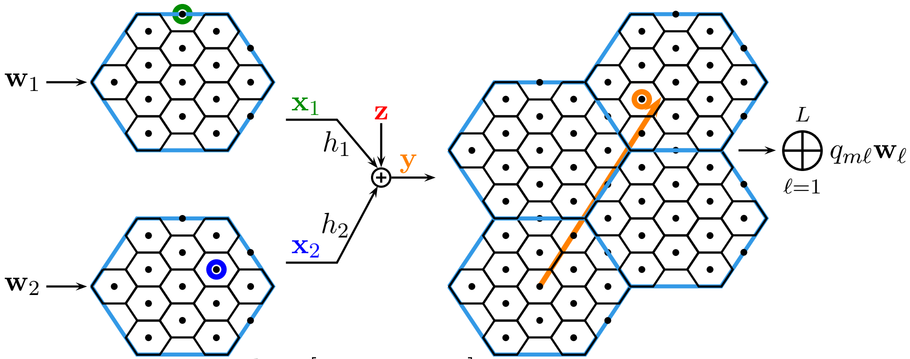
$$\alpha \mathbf{h} = [\ \alpha 1.4 \quad \alpha 2.1\ ]$$

$$\mathbf{a}_m = [\ 2 \quad 3\ ]$$

Effective noise: $\alpha^2 + P\|\alpha \mathbf{h} - \mathbf{a}_m\|^2$

Compute-and-Forward: Illustration

Map back to linear combination of the messages:



$$\alpha\mathbf{h} = [\ \alpha 1.4 \quad \alpha 2.1\ ]$$

$$\mathbf{a}_m = [\ 2 \quad 3\ ]$$

Effective noise: $\alpha^2 + P\|\alpha\mathbf{h} - \mathbf{a}_m\|^2$

Compute-and-Forward: Effective Noise

- Overall, the linear combination with integer coefficient vector $\mathbf{a}_m$ can be successfully decoded if

$$\sigma^2_{\text{eff}} > \alpha^2 + P\|\alpha\mathbf{h} - \mathbf{a}_m\|^2 \ .$$

- Overall, the linear combination with integer coefficient vector $\mathbf{a}_m$ can be successfully decoded if

$$\sigma_{\text{eff}}^2 > \alpha^2 + P\|\alpha\mathbf{h} - \mathbf{a}_m\|^2 \ .$$

- Optimal scaling α given by MMSE coefficient for estimating $\sum_\ell a_{m\ell}\mathbf{x}_\ell$ from $\sum_\ell h_\ell\mathbf{x}_\ell + \mathbf{z}$,

$$\alpha_{\text{MMSE}} = \frac{P\mathbf{a}_m^{\mathsf{T}}\mathbf{h}}{1 + P\|\mathbf{h}\|^2} \ .$$

- Overall, the linear combination with integer coefficient vector $\mathbf{a}_m$ can be successfully decoded if

$$\sigma^2_{\text{eff}} > \alpha^2 + P\|\alpha\mathbf{h} - \mathbf{a}_m\|^2 \ .$$

- Optimal scaling α given by MMSE coefficient for estimating $\sum_\ell a_{m\ell}\mathbf{x}_\ell$ from $\sum_\ell h_\ell \mathbf{x}_\ell + \mathbf{z}$,

$$\alpha_{\text{MMSE}} = \frac{P\mathbf{a}_m^{\mathsf{T}}\mathbf{h}}{1 + P\|\mathbf{h}\|^2} \ .$$

- Plugging this in and applying the Matrix Inversion Lemma, we get

$$\sigma^2_{\text{eff}} > \mathbf{a}_m^{\mathsf{T}}\big(P^{-1}\mathbf{I} + \ \mathbf{h}\mathbf{h}^{\mathsf{T}}\big)^{-1}\mathbf{a}_m \ .$$

- Overall, the linear combination with integer coefficient vector $\mathbf{a}_m$ can be successfully decoded if

$$\sigma^2_{\textsf{eff}} > \alpha^2 + P\|\alpha\mathbf{h} - \mathbf{a}_m\|^2 \ .$$

- Optimal scaling α given by MMSE coefficient for estimating $\sum_\ell a_{m\ell}\mathbf{x}_\ell$ from $\sum_\ell h_\ell \mathbf{x}_\ell + \mathbf{z}$,

$$\alpha_{\textsf{MMSE}} = \frac{P\mathbf{a}_m^{\mathsf{T}}\mathbf{h}}{1 + P\|\mathbf{h}\|^2} \ .$$

- Plugging this in and applying the Matrix Inversion Lemma, we get

$$\sigma^2_{\textsf{eff}} > \mathbf{a}_m^{\mathsf{T}}\big(P^{-1}\mathbf{I} + \ \mathbf{h}\mathbf{h}^{\mathsf{T}}\big)^{-1}\mathbf{a}_m \ .$$

- Overall, we find that if the rate satisfies

$$R < \min_m \frac{1}{2}\log\left(\frac{P}{\mathbf{a}_m^{\mathsf{T}}\big(P^{-1}\mathbf{I} + \ \mathbf{h}\mathbf{h}^{\mathsf{T}}\big)^{-1}\mathbf{a}_m}\right)$$

we can successfully decode all M linear combinations.

Compute-and-Forward: Achievable Rates

Theorem (Nazer-Gastpar '11)

The computation rate region described by

$$R_{comp}(\mathbf{h}, \mathbf{a}) = \max_{\alpha \in \mathbb{R}} \frac{1}{2} \log^{+} \left(\frac{P}{\alpha^2 + P \|\alpha \mathbf{h} - \mathbf{a}\|^2} \right)$$

is achievable.

Compute-and-Forward: Achievable Rates

Theorem (Nazer-Gastpar '11)

The computation rate region described by

$$R_{comp}(\mathbf{h}, \mathbf{a}) = \frac{1}{2} \log^+ \left(\frac{P}{\mathbf{a}^\mathsf{T} \left(P^{-1}\mathbf{I} + \ \mathbf{h}\mathbf{h}^\mathsf{T} \right)^{-1} \mathbf{a}} \right)$$

is achievable.

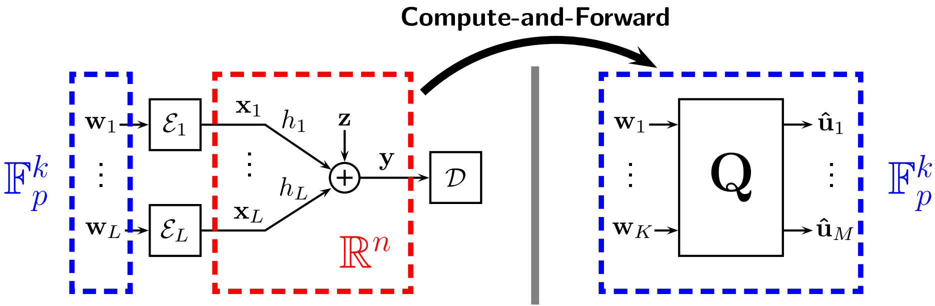
Compute-and-Forward: Achievable Rates

Theorem (Nazer-Gastpar '11)

The computation rate region described by

$$R_{comp}(\mathbf{h},\mathbf{a}) = \frac{1}{2}\log^+\left(\frac{P}{\mathbf{a}^\mathsf{T}\left(P^{-1}\mathbf{I} + \mathbf{h}\mathbf{h}^\mathsf{T}\right)^{-1}\mathbf{a}}\right)$$

is achievable.



if $R < \min_m R_{\text{comp}}(\mathbf{h},\mathbf{a}_m)$ for some $\mathbf{a}_1,\ldots,\mathbf{a}_M \in \mathbb{Z}^L$ satisfying $[\mathbf{a}_m] \bmod p = \mathbf{q}_m$.

Theorem (Nazer-Gastpar '11)

The computation rate region described by

$$R_{comp}(\mathbf{h},\mathbf{a}) = \frac{1}{2}\log^{+}\left(\frac{P}{\mathbf{a}^{\mathsf{T}}\left(P^{-1}\mathbf{I} + \ \mathbf{h}\mathbf{h}^{\mathsf{T}}\right)^{-1}\mathbf{a}}\right)$$

is achievable.

Special Cases:

- Perfect Match: $R_{\mathsf{comp}}(\mathbf{a},\mathbf{a}) = \frac{1}{2}\log^{+}\left(\frac{1}{\|\mathbf{a}\|^2} + P\right)$

Compute-and-Forward: Achievable Rates

Theorem (Nazer-Gastpar '11)

The computation rate region described by

$$R_{comp}(\mathbf{h}, \mathbf{a}) = \frac{1}{2} \log^+ \left(\frac{P}{\mathbf{a}^\mathsf{T} \left(P^{-1} \mathbf{I} + \ \mathbf{h}\mathbf{h}^\mathsf{T} \right)^{-1} \mathbf{a}} \right)$$

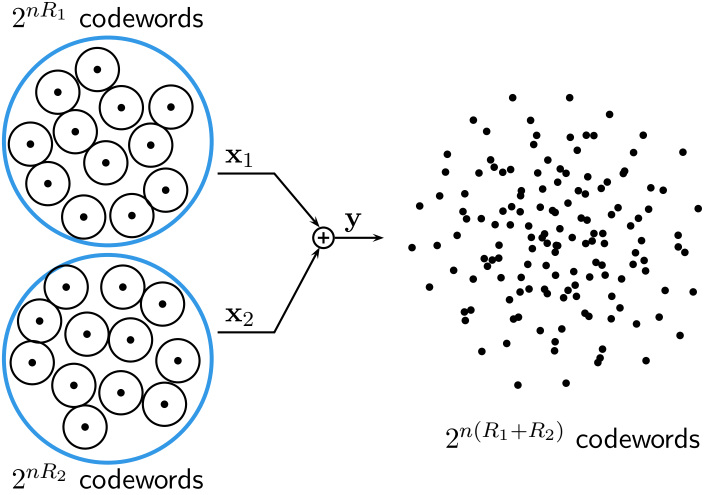
is achievable.

Special Cases:

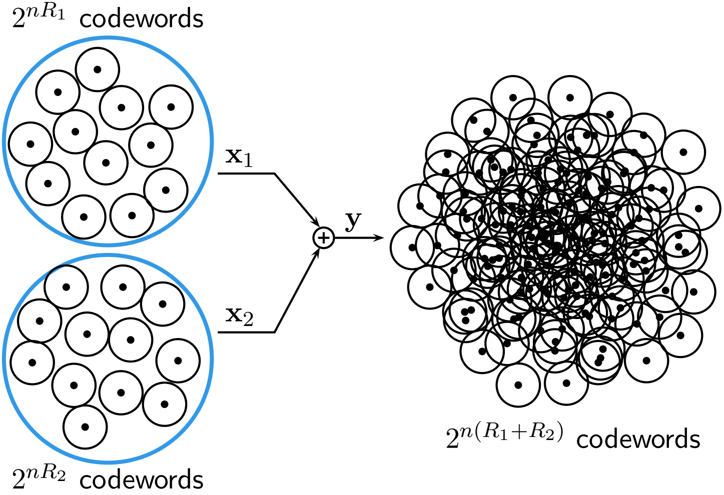
- Perfect Match: $R_{\text{comp}}(\mathbf{a}, \mathbf{a}) = \frac{1}{2} \log^+ \left(\frac{1}{\|\mathbf{a}\|^2} + P \right)$
- Decode a Message:

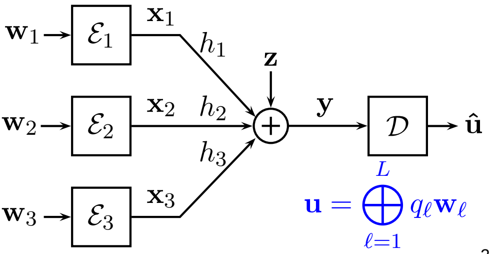
$$R_{\text{comp}}\Big(\mathbf{h}, [\underbrace{0 \ \cdots \ 0}_{m-1 \text{ zeros}} \ 1 \ 0 \ \cdots \ 0]^\mathsf{T}\Big) = \frac{1}{2} \log \left(1 + \frac{h_m^2 P}{1 + P \sum_{\ell \neq m} h_\ell^2} \right)$$

Random i.i.d. codes are not good for computation



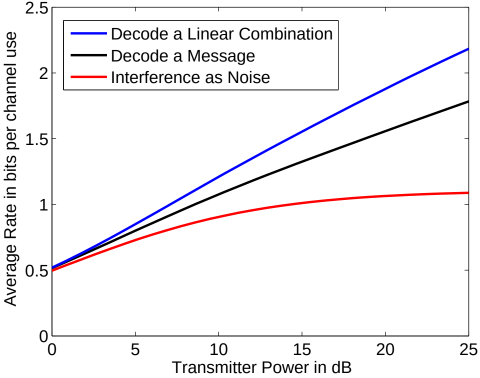
Random i.i.d. codes are not good for computation



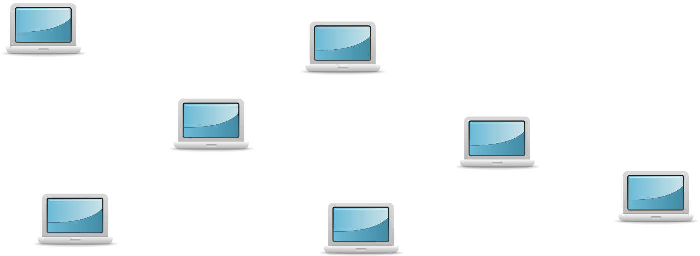


Relay either decodes some linear combination of messages or an individual message.

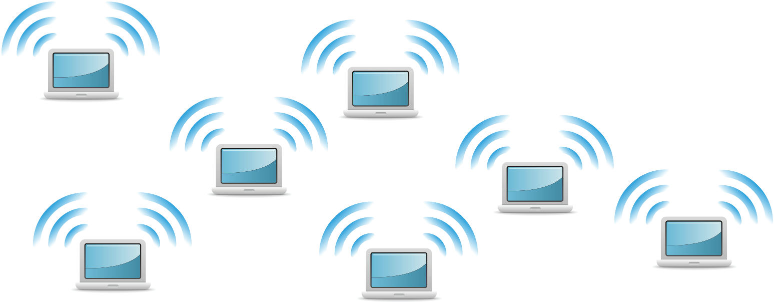
- Three transmitters that do not know the fading coefficients.
- Average rate plotted for i.i.d. Gaussian fading.



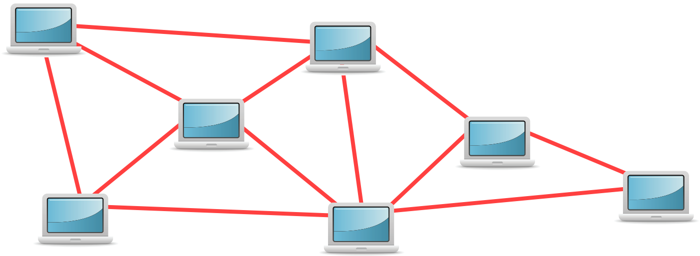
Physical-Layer Network Coding



- Usually fight interference and convert to network of bit pipes.

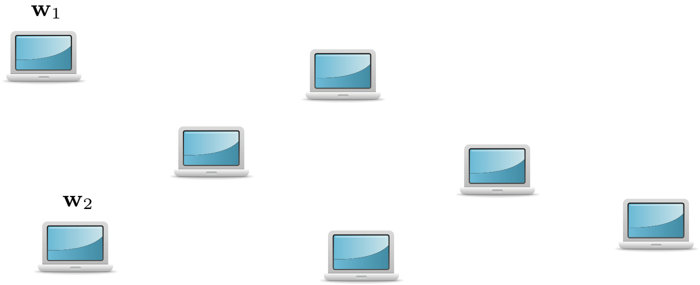


- Usually fight interference and convert to network of bit pipes.



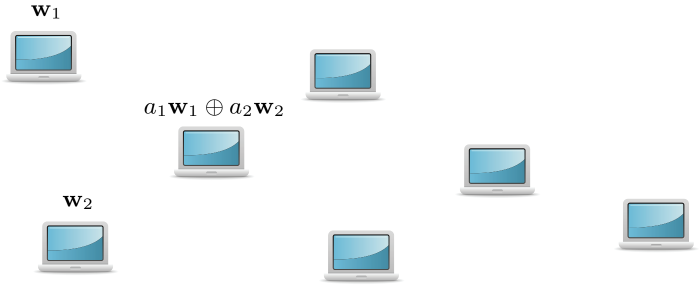
- Usually fight interference and convert to network of bit pipes.

Physical-Layer Network Coding



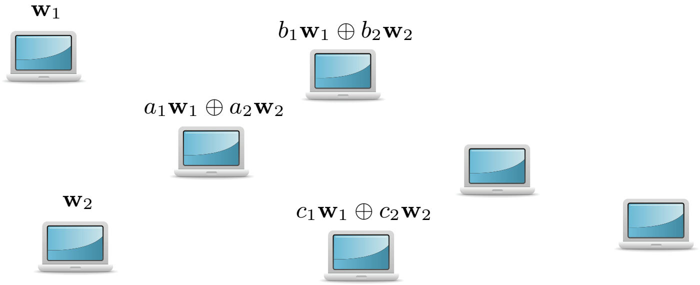
- Usually fight interference and convert to network of bit pipes.
- Physical-layer network coding: exploiting the wireless medium for network coding. Independently and concurrently proposed by **Zhang-Liew-Lam '06, Popovski-Yomo '06, Nazer-Gastpar '06.**
- Compute-and-forward is an information-theoretic approach.
- Recent surveys: **Liew-Zhang-Lu '11, Nazer-Gastpar Proc. IEEE '11.**

Physical-Layer Network Coding



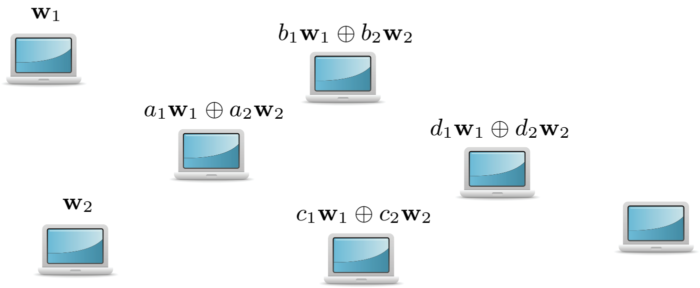
- Usually fight interference and convert to network of bit pipes.
- Physical-layer network coding: exploiting the wireless medium for network coding. Independently and concurrently proposed by **Zhang-Liew-Lam '06, Popovski-Yomo '06, Nazer-Gastpar '06.**
- Compute-and-forward is an information-theoretic approach.
- Recent surveys: **Liew-Zhang-Lu '11, Nazer-Gastpar Proc. IEEE '11.**

Physical-Layer Network Coding



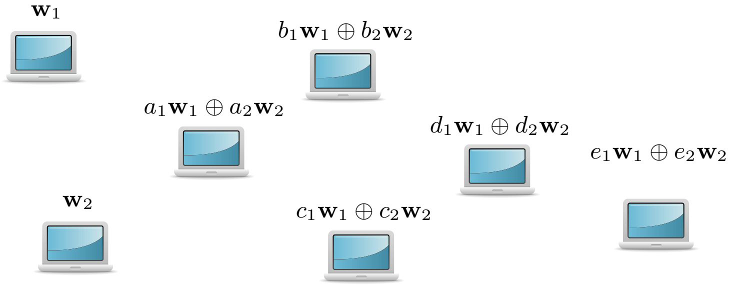
- Usually fight interference and convert to network of bit pipes.
- Physical-layer network coding: exploiting the wireless medium for network coding. Independently and concurrently proposed by **Zhang-Liew-Lam '06, Popovski-Yomo '06, Nazer-Gastpar '06.**
- Compute-and-forward is an information-theoretic approach.
- Recent surveys: **Liew-Zhang-Lu '11, Nazer-Gastpar Proc. IEEE '11.**

Physical-Layer Network Coding



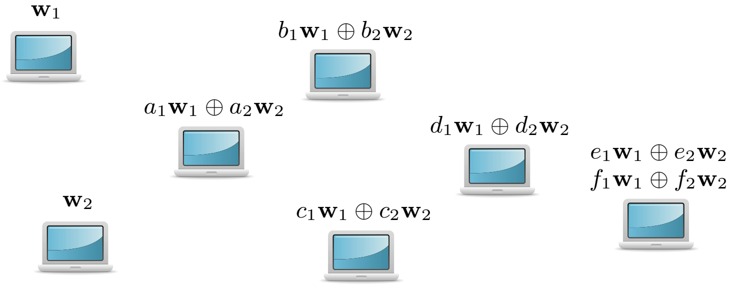
- Usually fight interference and convert to network of bit pipes.
- Physical-layer network coding: exploiting the wireless medium for network coding. Independently and concurrently proposed by **Zhang-Liew-Lam '06, Popovski-Yomo '06, Nazer-Gastpar '06.**
- Compute-and-forward is an information-theoretic approach.
- Recent surveys: **Liew-Zhang-Lu '11, Nazer-Gastpar Proc. IEEE '11.**

Physical-Layer Network Coding



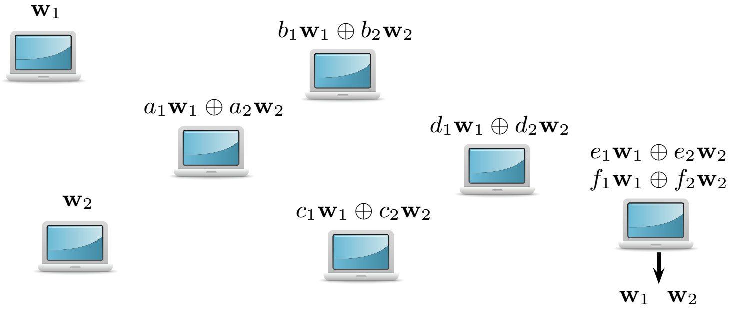
- Usually fight interference and convert to network of bit pipes.
- Physical-layer network coding: exploiting the wireless medium for network coding. Independently and concurrently proposed by **Zhang-Liew-Lam '06, Popovski-Yomo '06, Nazer-Gastpar '06.**
- Compute-and-forward is an information-theoretic approach.
- Recent surveys: **Liew-Zhang-Lu '11, Nazer-Gastpar Proc. IEEE '11.**

Physical-Layer Network Coding



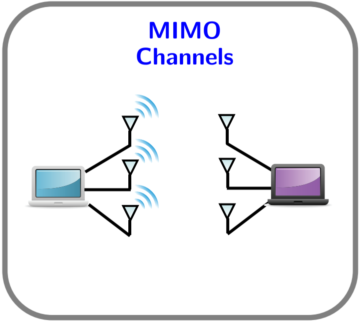
- Usually fight interference and convert to network of bit pipes.
- Physical-layer network coding: exploiting the wireless medium for network coding. Independently and concurrently proposed by **Zhang-Liew-Lam '06, Popovski-Yomo '06, Nazer-Gastpar '06.**
- Compute-and-forward is an information-theoretic approach.
- Recent surveys: **Liew-Zhang-Lu '11, Nazer-Gastpar Proc. IEEE '11.**

Physical-Layer Network Coding



- Usually fight interference and convert to network of bit pipes.
- Physical-layer network coding: exploiting the wireless medium for network coding. Independently and concurrently proposed by **Zhang-Liew-Lam '06, Popovski-Yomo '06, Nazer-Gastpar '06.**
- Compute-and-forward is an information-theoretic approach.
- Recent surveys: **Liew-Zhang-Lu '11, Nazer-Gastpar Proc. IEEE '11.**

- Warm-up: Compute-and-forward over finite field channels.
- Compute-and-forward over Gaussian channels.
- Applications to communication across single-hop Gaussian networks.



Joint work with Jiening Zhan, Uri Erez, and Michael Gastpar.

- Increasing the number of antennas in a wireless system can significantly increase its capacity.
  Foschini '96, Foschini and Gans '98, Telatar '99.
- Enormous body of work has strived to develop receiver architectures that can approach these capacity gains with manageable complexity.

Vast majority of receiver architectures fall into these two categories (references at the end of the talk):

- Joint Maximum Likelihood Receivers:

Vast majority of receiver architectures fall into these two categories (references at the end of the talk):

- Joint Maximum Likelihood Receivers:
  - Optimal but prohibitively complex for capacity-approaching codes.

MIMO Receiver Architectures

Vast majority of receiver architectures fall into these two categories (references at the end of the talk):

- Joint Maximum Likelihood Receivers:
  - Optimal but prohibitively complex for capacity-approaching codes.
  - Often ML detection is performed at the symbol level and coupled with an outer channel code. This includes the vast literature on space-time codes, sphere decoding, and lattice-aided reduction.

MIMO Receiver Architectures

Vast majority of receiver architectures fall into these two categories (references at the end of the talk):

- Joint Maximum Likelihood Receivers:
  - Optimal but prohibitively complex for capacity-approaching codes.
  - Often ML detection is performed at the symbol level and coupled with an outer channel code. This includes the vast literature on space-time codes, sphere decoding, and lattice-aided reduction.
  - Well-suited to the high SNR regime as well as scenarios where diversity is more important than rate.

MIMO Receiver Architectures

Vast majority of receiver architectures fall into these two categories (references at the end of the talk):

- Joint Maximum Likelihood Receivers:
  - Optimal but prohibitively complex for capacity-approaching codes.
  - Often ML detection is performed at the symbol level and coupled with an outer channel code. This includes the vast literature on space-time codes, sphere decoding, and lattice-aided reduction.
  - Well-suited to the high SNR regime as well as scenarios where diversity is more important than rate.

- Zero-Forcing and Linear MMSE Receivers:

MIMO Receiver Architectures

Vast majority of receiver architectures fall into these two categories (references at the end of the talk):

- Joint Maximum Likelihood Receivers:
  - Optimal but prohibitively complex for capacity-approaching codes.
  - Often ML detection is performed at the symbol level and coupled with an outer channel code. This includes the vast literature on space-time codes, sphere decoding, and lattice-aided reduction.
  - Well-suited to the high SNR regime as well as scenarios where diversity is more important than rate.

- Zero-Forcing and Linear MMSE Receivers:
  - First, decouple transmitted data streams via linear equalization at the cost of noise amplification. Then apply SISO decoding.

MIMO Receiver Architectures

Vast majority of receiver architectures fall into these two categories (references at the end of the talk):

- Joint Maximum Likelihood Receivers:
  - Optimal but prohibitively complex for capacity-approaching codes.
  - Often ML detection is performed at the symbol level and coupled with an outer channel code. This includes the vast literature on space-time codes, sphere decoding, and lattice-aided reduction.
  - Well-suited to the high SNR regime as well as scenarios where diversity is more important than rate.
- Zero-Forcing and Linear MMSE Receivers:
  - First, decouple transmitted data streams via linear equalization at the cost of noise amplification. Then apply SISO decoding.
  - Simple interface with powerful channel coding techniques.

MIMO Receiver Architectures

Vast majority of receiver architectures fall into these two categories (references at the end of the talk):

- Joint Maximum Likelihood Receivers:
  - Optimal but prohibitively complex for capacity-approaching codes.
  - Often ML detection is performed at the symbol level and coupled with an outer channel code. This includes the vast literature on space-time codes, sphere decoding, and lattice-aided reduction.
  - Well-suited to the high SNR regime as well as scenarios where diversity is more important than rate.

- Zero-Forcing and Linear MMSE Receivers:
  - First, decouple transmitted data streams via linear equalization at the cost of noise amplification. Then apply SISO decoding.
  - Simple interface with powerful channel coding techniques.
  - Performance can be enhanced via successive interference cancellation.

MIMO Receiver Architectures

Vast majority of receiver architectures fall into these two categories (references at the end of the talk):

- Joint Maximum Likelihood Receivers:
  - Optimal but prohibitively complex for capacity-approaching codes.
  - Often ML detection is performed at the symbol level and coupled with an outer channel code. This includes the vast literature on space-time codes, sphere decoding, and lattice-aided reduction.
  - Well-suited to the high SNR regime as well as scenarios where diversity is more important than rate.

- Zero-Forcing and Linear MMSE Receivers:
  - First, decouple transmitted data streams via linear equalization at the cost of noise amplification. Then apply SISO decoding.
  - Simple interface with powerful channel coding techniques.
  - Performance can be enhanced via successive interference cancellation.
  - Well-suited to scenarios where rate is more important than diversity.

MIMO Receiver Architectures

Vast majority of receiver architectures fall into these two categories (references at the end of the talk):

- Joint Maximum Likelihood Receivers:
  - Optimal but prohibitively complex for capacity-approaching codes.
  - Often ML detection is performed at the symbol level and coupled with an outer channel code. This includes the vast literature on space-time codes, sphere decoding, and lattice-aided reduction.
  - Well-suited to the high SNR regime as well as scenarios where diversity is more important than rate.

- Zero-Forcing and Linear MMSE Receivers:
  - First, decouple transmitted data streams via linear equalization at the cost of noise amplification. Then apply SISO decoding.
  - Simple interface with powerful channel coding techniques.
  - Performance can be enhanced via successive interference cancellation.
  - Well-suited to scenarios where rate is more important than diversity.

We propose a new class of Integer-Forcing Linear Receivers.

A Simple Example

- $\mathbf{Y} = \begin{bmatrix} 1 & 2 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \end{bmatrix} + \mathbf{Z}$

A Simple Example

- $\mathbf{Y} = \begin{bmatrix} 1 & 2 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \end{bmatrix} + \mathbf{Z}$

- Zero-Forcing: $\begin{bmatrix} -1 & 2 \\ 1 & -1 \end{bmatrix} \mathbf{Y} = \begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \end{bmatrix} + \begin{bmatrix} -1 & 2 \\ 1 & -1 \end{bmatrix} \mathbf{Z}$

- Effective noise variances: $\sigma_1^2 = 5$ and $\sigma_2^2 = 2$.

A Simple Example

- $\mathbf{Y} = \begin{bmatrix} 1 & 2 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \end{bmatrix} + \mathbf{Z}$

- Zero-Forcing: $\begin{bmatrix} -1 & 2 \\ 1 & -1 \end{bmatrix} \mathbf{Y} = \begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \end{bmatrix} + \begin{bmatrix} -1 & 2 \\ 1 & -1 \end{bmatrix} \mathbf{Z}$

- Effective noise variances: $\sigma_1^2 = 5$ and $\sigma_2^2 = 2$.

- Integer-Forcing: $\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \mathbf{Y} = \begin{bmatrix} \mathbf{x}_1 + 2\mathbf{x}_2 \\ \mathbf{x}_1 + \mathbf{x}_2 \end{bmatrix} + \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \mathbf{Z}$

- Effective noise variances: $\sigma_1^2 = 1$ and $\sigma_2^2 = 1$.

A Simple Example

- $\mathbf{Y} = \begin{bmatrix} 1 & 2 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \end{bmatrix} + \mathbf{Z}$

- Zero-Forcing: $\begin{bmatrix} -1 & 2 \\ 1 & -1 \end{bmatrix} \mathbf{Y} = \begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \end{bmatrix} + \begin{bmatrix} -1 & 2 \\ 1 & -1 \end{bmatrix} \mathbf{Z}$

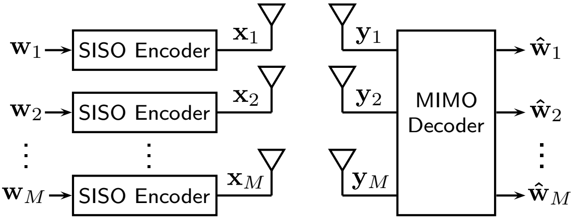
- Effective noise variances: $\sigma_1^2 = 5$ and $\sigma_2^2 = 2$.

- Integer-Forcing: $\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \mathbf{Y} = \begin{bmatrix} \mathbf{x}_1 + 2\mathbf{x}_2 \\ \mathbf{x}_1 + \mathbf{x}_2 \end{bmatrix} + \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \mathbf{Z}$

- Effective noise variances: $\sigma_1^2 = 1$ and $\sigma_2^2 = 1$.

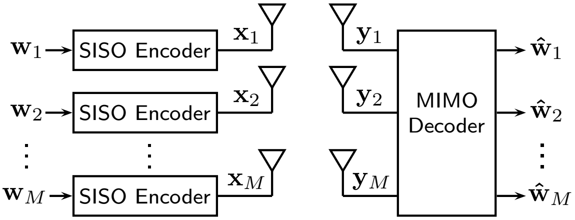
- Does this help beyond integer-valued channel matrices?

MIMO Problem Statement



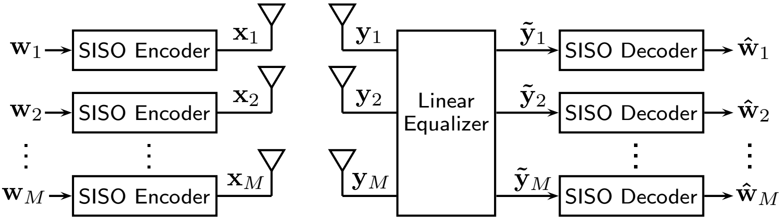
- Each antenna has an independent data stream $\mathbf{x}_\ell \in \mathbb{R}^n$ of rate R (e.g., V-BLAST setting, cellular uplink). $\mathbf{X} = [\mathbf{x}_1 \ \cdots \ \mathbf{x}_M]^\mathsf{T}$.
- Channel model: $\mathbf{Y} = \mathbf{H}\mathbf{X} + \mathbf{Z}$ where $\mathbf{Z}$ is elementwise i.i.d. $\mathcal{N}(0,1)$.
- CSIR: Only receiver knows channel realization $\mathbf{H} \in \mathbb{R}^{M \times M}$.
- Probability of error: $\mathbb{P}\big(\{\hat{\mathbf{w}}_1 \neq \mathbf{w}_1\} \cup \cdots \cup \{\hat{\mathbf{w}}_M \neq \mathbf{w}_M\}\big) < \epsilon$

MIMO Problem Statement



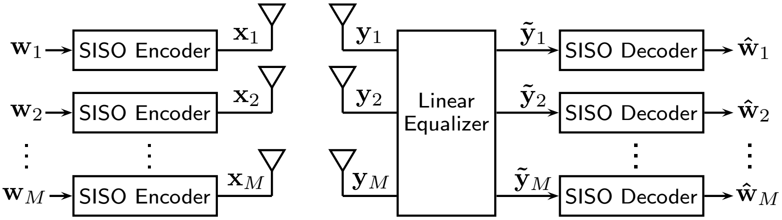
- Each antenna has an independent data stream $\mathbf{x}_\ell \in \mathbb{R}^n$ of rate R (e.g., V-BLAST setting, cellular uplink). $\mathbf{X} = [\mathbf{x}_1 \ \cdots \ \mathbf{x}_M]^\mathsf{T}$.
- Channel model: $\mathbf{Y} = \mathbf{H}\mathbf{X} + \mathbf{Z}$ where $\mathbf{Z}$ is elementwise i.i.d. $\mathcal{N}(0,1)$.
- CSIR: Only receiver knows channel realization $\mathbf{H} \in \mathbb{R}^{M \times M}$.
- Probability of error: $\mathbb{P}\big(\{\hat{\mathbf{w}}_1 \neq \mathbf{w}_1\} \cup \cdots \cup \{\hat{\mathbf{w}}_M \neq \mathbf{w}_M\}\big) < \epsilon$
- Joint maximum likelihood decoding is optimal but has high implementation complexity.

Zero-Forcing Linear Receivers

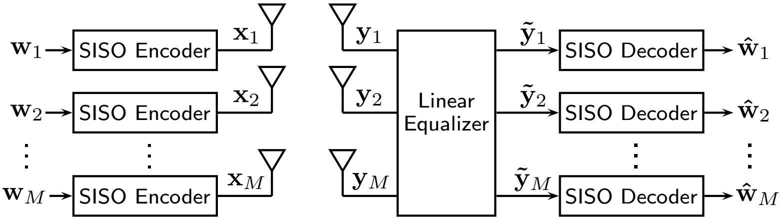


- Zero-Forcing: Project the received signal, $\tilde{\mathbf{Y}} = \mathbf{B}\mathbf{Y}$, to eliminate interference between data streams.

Zero-Forcing Linear Receivers

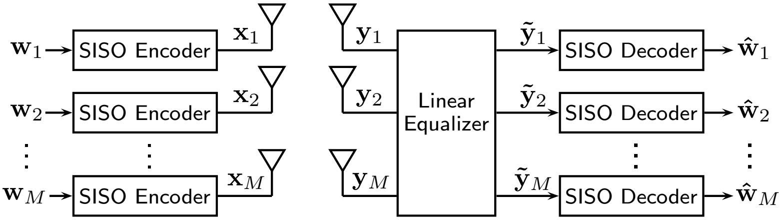


- Zero-Forcing: Project the received signal, $\tilde{\mathbf{Y}} = \mathbf{B}\mathbf{Y}$, to eliminate interference between data streams.
- Significantly reduces complexity at the expense of performance.



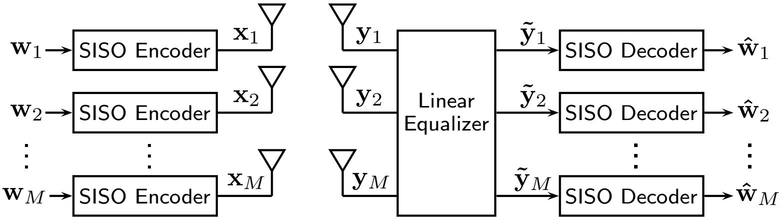
- Zero-Forcing: Project the received signal, $\tilde{\mathbf{Y}} = \mathbf{B}\mathbf{Y}$, to eliminate interference between data streams.
- Significantly reduces complexity at the expense of performance.
- Ex: If $\mathbf{H}$ is full rank, set $\mathbf{B} = \mathbf{H}^{-1}$ to get $\tilde{\mathbf{Y}} = \mathbf{X} + \mathbf{H}^{-1}\mathbf{Z}$.

Zero-Forcing Linear Receivers



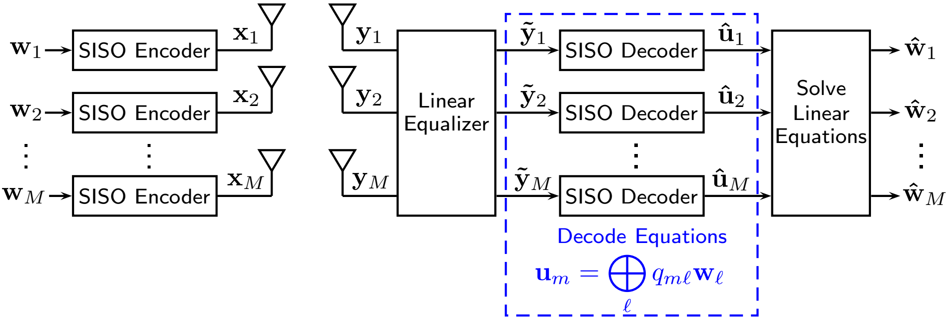
- Zero-Forcing: Project the received signal, $\tilde{\mathbf{Y}} = \mathbf{B}\mathbf{Y}$, to eliminate interference between data streams.
- Significantly reduces complexity at the expense of performance.
- Ex: If $\mathbf{H}$ is full rank, set $\mathbf{B} = \mathbf{H}^{-1}$ to get $\tilde{\mathbf{Y}} = \mathbf{X} + \mathbf{H}^{-1}\mathbf{Z}$.
- Optimal $\mathbf{B}$ is the MMSE projection: $\mathbf{B} = \mathbf{H}^{\mathsf{T}}\left(P^{-1}\mathbf{I} + \mathbf{H}\mathbf{H}^{\mathsf{T}}\right)^{-1}$.

Zero-Forcing Linear Receivers

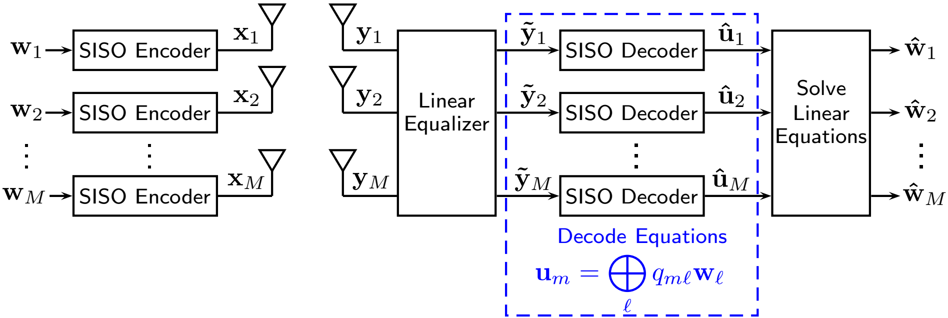


- Zero-Forcing: Project the received signal, $\tilde{\mathbf{Y}} = \mathbf{B}\mathbf{Y}$, to eliminate interference between data streams.
- Significantly reduces complexity at the expense of performance.
- Ex: If $\mathbf{H}$ is full rank, set $\mathbf{B} = \mathbf{H}^{-1}$ to get $\tilde{\mathbf{Y}} = \mathbf{X} + \mathbf{H}^{-1}\mathbf{Z}$.
- Optimal $\mathbf{B}$ is the MMSE projection: $\mathbf{B} = \mathbf{H}^{\mathsf{T}}\left(P^{-1}\mathbf{I} + \mathbf{H}\mathbf{H}^{\mathsf{T}}\right)^{-1}$.
  (Often called the linear MMSE receiver.)

Integer-Forcing Linear Receivers

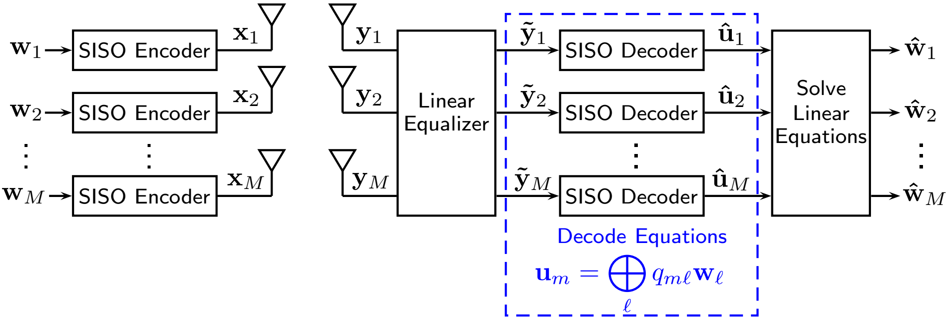


- Integer-Forcing: Project the received signal, $\tilde{\mathbf{Y}} = \mathbf{B}\mathbf{Y}$, to create an integer-valued effective channel matrix.



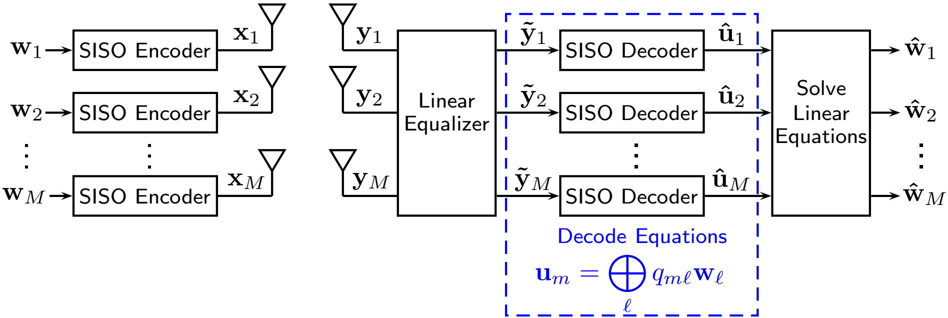
- Integer-Forcing: Project the received signal, $\tilde{\mathbf{Y}} = \mathbf{B}\mathbf{Y}$, to create an integer-valued effective channel matrix.
- Ex: If $\mathbf{H}$ is full rank, set $\mathbf{B} = \mathbf{A}\mathbf{H}^{-1}$ to get $\tilde{\mathbf{Y}} = \mathbf{A}\mathbf{X} + \mathbf{A}\mathbf{H}^{-1}\mathbf{Z}$.

Integer-Forcing Linear Receivers



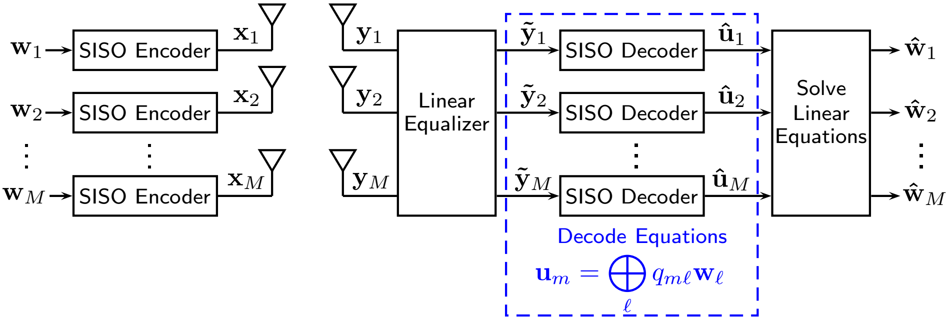
- Integer-Forcing: Project the received signal, $\tilde{\mathbf{Y}} = \mathbf{B}\mathbf{Y}$, to create an integer-valued effective channel matrix.
- Ex: If $\mathbf{H}$ is full rank, set $\mathbf{B} = \mathbf{A}\mathbf{H}^{-1}$ to get $\tilde{\mathbf{Y}} = \mathbf{A}\mathbf{X} + \mathbf{A}\mathbf{H}^{-1}\mathbf{Z}$.
- Optimize over $\mathbf{A} \in \mathbb{Z}^{M \times M}$ to minimize effective noise.

Integer-Forcing Linear Receivers



- Integer-Forcing: Project the received signal, $\tilde{\mathbf{Y}} = \mathbf{B}\mathbf{Y}$, to create an integer-valued effective channel matrix.
- Ex: If $\mathbf{H}$ is full rank, set $\mathbf{B} = \mathbf{A}\mathbf{H}^{-1}$ to get $\tilde{\mathbf{Y}} = \mathbf{A}\mathbf{X} + \mathbf{A}\mathbf{H}^{-1}\mathbf{Z}$.
- Optimize over $\mathbf{A} \in \mathbb{Z}^{M\times M}$ to minimize effective noise.
- Optimal $\mathbf{B}$ is the MMSE projection: $\mathbf{B} = \mathbf{A}\mathbf{H}^{\mathsf{T}}\left(P^{-1}\mathbf{I} + \mathbf{H}\mathbf{H}^{\mathsf{T}}\right)^{-1}$.

Integer-Forcing Linear Receivers



- Integer-Forcing: Project the received signal, $\tilde{\mathbf{Y}} = \mathbf{B}\mathbf{Y}$, to create an integer-valued effective channel matrix.
- Ex: If $\mathbf{H}$ is full rank, set $\mathbf{B} = \mathbf{A}\mathbf{H}^{-1}$ to get $\tilde{\mathbf{Y}} = \mathbf{A}\mathbf{X} + \mathbf{A}\mathbf{H}^{-1}\mathbf{Z}$.
- Optimize over $\mathbf{A} \in \mathbb{Z}^{M\times M}$ to minimize effective noise.
- Optimal $\mathbf{B}$ is the MMSE projection: $\mathbf{B} = \mathbf{A}\mathbf{H}^{\mathsf{T}}\left(P^{-1}\mathbf{I} + \mathbf{H}\mathbf{H}^{\mathsf{T}}\right)^{-1}$.
- Includes zero-forcing by setting $\mathbf{A} = \mathbf{I}$.

- Receiver observes: $$\mathbf{Y} = \sum_{m=1}^{M} \mathbf{h}_m \mathbf{x}_m^{\mathsf{T}} + \mathbf{Z} = \mathbf{H}\mathbf{X} + \mathbf{Z}.$$

- Receiver observes: $\mathbf{Y} = \sum_{m=1}^{M} \mathbf{h}_m \mathbf{x}_m^{\mathsf{T}} + \mathbf{Z} = \mathbf{H}\mathbf{X} + \mathbf{Z}.$

- To recover the linear combination with integer coefficient vector $\mathbf{a} \in \mathbb{Z}^L$, the receiver projects its observation:

$$\mathbf{b}^{\mathsf{T}}\mathbf{Y} = \mathbf{a}^{\mathsf{T}}\mathbf{X} + \underbrace{(\mathbf{b}^{\mathsf{T}}\mathbf{H} - \mathbf{a}^{\mathsf{T}})\mathbf{X} + \mathbf{b}^{\mathsf{T}}\mathbf{Z}}_{\mathbf{z}_{\text{eff}}}$$

MIMO Compute-and-Forward

- Receiver observes: $\mathbf{Y} = \sum_{m=1}^{M} \mathbf{h}_m \mathbf{x}_m^{\mathsf{T}} + \mathbf{Z} = \mathbf{H}\mathbf{X} + \mathbf{Z}.$

- To recover the linear combination with integer coefficient vector $\mathbf{a} \in \mathbb{Z}^L$, the receiver projects its observation:

$$\mathbf{b}^{\mathsf{T}}\mathbf{Y} = \mathbf{a}^{\mathsf{T}}\mathbf{X} + \underbrace{(\mathbf{b}^{\mathsf{T}}\mathbf{H} - \mathbf{a}^{\mathsf{T}})\mathbf{X} + \mathbf{b}^{\mathsf{T}}\mathbf{Z}}_{\mathbf{z}_{\mathsf{eff}}}$$

Theorem (Zhan-Nazer-Erez-Gastpar '14)

The computation rate region described by

$$R_{\mathsf{comp}}(\mathbf{H}, \mathbf{a}) = \max_{\mathbf{b} \in \mathbb{R}^M} \frac{1}{2} \log^+ \left(\frac{P}{\|\mathbf{b}\|^2 + P\|\mathbf{H}^{\mathsf{T}}\mathbf{b} - \mathbf{a}\|^2} \right)$$

is achievable.

MIMO Compute-and-Forward

- Receiver observes: $\mathbf{Y} = \sum_{m=1}^{M} \mathbf{h}_m \mathbf{x}_m^{\mathsf{T}} + \mathbf{Z} = \mathbf{H}\mathbf{X} + \mathbf{Z}.$

- To recover the linear combination with integer coefficient vector $\mathbf{a} \in \mathbb{Z}^L$, the receiver projects its observation:

$$\mathbf{b}^{\mathsf{T}}\mathbf{Y} = \mathbf{a}^{\mathsf{T}}\mathbf{X} + \underbrace{(\mathbf{b}^{\mathsf{T}}\mathbf{H} - \mathbf{a}^{\mathsf{T}})\mathbf{X} + \mathbf{b}^{\mathsf{T}}\mathbf{Z}}_{\mathbf{z}_{\mathsf{eff}}}$$

Theorem (Zhan-Nazer-Erez-Gastpar '14)

The computation rate region described by

$$R_{\mathsf{comp}}(\mathbf{H}, \mathbf{a}) = \frac{1}{2}\log^+\left(\frac{P}{\mathbf{a}^{\mathsf{T}}\left(P^{-1}\mathbf{I} + \ \mathbf{H}^{\mathsf{T}}\mathbf{H}\right)^{-1}\mathbf{a}}\right)$$

is achievable.

Integer-Forcing:

$$\mathbf{Y} = \mathbf{H}\mathbf{X} + \mathbf{Z} \xrightarrow{\text{Project}} \mathbf{B}\mathbf{Y} = \mathbf{A}\mathbf{X} + \underbrace{(\mathbf{B}\mathbf{H} - \mathbf{A})\mathbf{X} + \mathbf{B}\mathbf{Z}}_{\mathbf{Z}_{\text{eff}}} \xrightarrow{\text{Decode}} \mathbf{A}\mathbf{X}$$

Integer-Forcing:

$$\mathbf{Y} = \mathbf{H}\mathbf{X} + \mathbf{Z} \xrightarrow{\text{Project}} \mathbf{B}\mathbf{Y} = \mathbf{A}\mathbf{X} + \underbrace{(\mathbf{B}\mathbf{H} - \mathbf{A})\mathbf{X} + \mathbf{B}\mathbf{Z}}_{\mathbf{Z}_{\text{eff}}} \xrightarrow{\text{Decode}} \mathbf{A}\mathbf{X}$$

- Achievable rate: $R_{\text{IF}}(\mathbf{H}) = M \max_{\substack{\mathbf{A} \in \mathbb{Z}^{M \times M} \\ \text{rank}(\mathbf{A}) = \text{M}}} \min_{m} R_{\text{comp}}(\mathbf{H}, \mathbf{a}_m)$

Integer-Forcing:

$$\mathbf{Y} = \mathbf{H}\mathbf{X} + \mathbf{Z} \xrightarrow{\text{Project}} \mathbf{B}\mathbf{Y} = \mathbf{A}\mathbf{X} + \underbrace{(\mathbf{B}\mathbf{H} - \mathbf{A})\mathbf{X} + \mathbf{B}\mathbf{Z}}_{\mathbf{Z}_{\text{eff}}} \xrightarrow{\text{Decode}} \mathbf{A}\mathbf{X}$$

- Achievable rate: $R_{\text{IF}}(\mathbf{H}) = M \max_{\substack{\mathbf{A} \in \mathbb{Z}^{M \times M} \\ \text{rank}(\mathbf{A}) = \text{M}}} \min_{m} R_{\text{comp}}(\mathbf{H}, \mathbf{a}_m)$
- Only need to search over vectors satisfying $\|\mathbf{a}_m\|^2 \leq 1 + P\lambda^2_{\max}(\mathbf{H})$.

Integer-Forcing:

$$\mathbf{Y} = \mathbf{H}\mathbf{X} + \mathbf{Z} \xrightarrow{\text{Project}} \mathbf{B}\mathbf{Y} = \mathbf{A}\mathbf{X} + \underbrace{(\mathbf{B}\mathbf{H} - \mathbf{A})\mathbf{X} + \mathbf{B}\mathbf{Z}}_{\mathbf{Z}_{\text{eff}}} \xrightarrow{\text{Decode}} \mathbf{A}\mathbf{X}$$

- Achievable rate: $R_{\text{IF}}(\mathbf{H}) = M \max_{\substack{\mathbf{A} \in \mathbb{Z}^{M \times M} \\ \text{rank}(\mathbf{A}) = \text{M}}} \min_{m} R_{\text{comp}}(\mathbf{H}, \mathbf{a}_m)$
- Only need to search over vectors satisfying $\|\mathbf{a}_m\|^2 \leq 1 + P\lambda_{\max}^2(\mathbf{H})$.
- Faster search: Apply LLL algorithm to $\mathbf{F} = \left(P^{-1}\mathbf{I} + \mathbf{H}^{\mathsf{T}}\mathbf{H}\right)^{-1/2}$.

Comparison

Integer-Forcing:

$$\mathbf{Y} = \mathbf{H}\mathbf{X} + \mathbf{Z} \xrightarrow{\text{Project}} \mathbf{B}\mathbf{Y} = \mathbf{A}\mathbf{X} + \underbrace{(\mathbf{B}\mathbf{H} - \mathbf{A})\mathbf{X} + \mathbf{B}\mathbf{Z}}_{\mathbf{Z}_{\text{eff}}} \xrightarrow{\text{Decode}} \mathbf{A}\mathbf{X}$$

- Achievable rate: $R_{\text{IF}}(\mathbf{H}) = M \max_{\substack{\mathbf{A} \in \mathbb{Z}^{M \times M} \\ \text{rank}(\mathbf{A}) = \text{M}}} \min_{m} R_{\text{comp}}(\mathbf{H}, \mathbf{a}_m)$
- Only need to search over vectors satisfying $\|\mathbf{a}_m\|^2 \leq 1 + P\lambda_{\max}^2(\mathbf{H})$.
- Faster search: Apply LLL algorithm to $\mathbf{F} = \left(P^{-1}\mathbf{I} + \mathbf{H}^{\mathsf{T}}\mathbf{H}\right)^{-1/2}$.

Zero-Forcing:

$$\mathbf{Y} = \mathbf{H}\mathbf{X} + \mathbf{Z} \xrightarrow{\text{Project}} \mathbf{B}\mathbf{Y} = \mathbf{X} + \underbrace{(\mathbf{B}\mathbf{H} - \mathbf{I})\mathbf{X} + \mathbf{B}\mathbf{Z}}_{\mathbf{Z}_{\text{eff}}} \xrightarrow{\text{Decode}} \mathbf{X}$$

Integer-Forcing:

$$\mathbf{Y} = \mathbf{HX} + \mathbf{Z} \xrightarrow{\text{Project}} \mathbf{BY} = \mathbf{AX} + \underbrace{(\mathbf{BH} - \mathbf{A})\mathbf{X} + \mathbf{BZ}}_{\mathbf{Z}_{\text{eff}}} \xrightarrow{\text{Decode}} \mathbf{AX}$$

- Achievable rate: $R_{\text{IF}}(\mathbf{H}) = M \max_{\substack{\mathbf{A} \in \mathbb{Z}^{M \times M} \\ \text{rank}(\mathbf{A}) = \text{M}}} \min_{m} R_{\text{comp}}(\mathbf{H}, \mathbf{a}_m)$
- Only need to search over vectors satisfying $\|\mathbf{a}_m\|^2 \leq 1 + P\lambda_{\max}^2(\mathbf{H})$.
- Faster search: Apply LLL algorithm to $\mathbf{F} = \left(P^{-1}\mathbf{I} + \mathbf{H}^{\mathsf{T}}\mathbf{H}\right)^{-1/2}$.

Zero-Forcing:

$$\mathbf{Y} = \mathbf{HX} + \mathbf{Z} \xrightarrow{\text{Project}} \mathbf{BY} = \mathbf{X} + \underbrace{(\mathbf{BH} - \mathbf{I})\mathbf{X} + \mathbf{BZ}}_{\mathbf{Z}_{\text{eff}}} \xrightarrow{\text{Decode}} \mathbf{X}$$

- Achievable rate: $R_{\text{ZF}}(\mathbf{H}) = M \min_{m} R_{\text{comp}}\Big(\mathbf{H}, [\underbrace{0 \ \cdots \ 0}_{m-1 \text{ zeros}} \ 1 \ 0 \ \cdots \ 0]^{\mathsf{T}}\Big)$

Successive Interference Cancellation

- Linear receiver architectures are often augmented using successive interference cancellation (SIC).
- Basic idea: After decoding codeword $\mathbf{x}_\ell$, remove its effect from channel output to reduce the interference between data streams.

Successive Interference Cancellation

- Linear receiver architectures are often augmented using successive interference cancellation (SIC).
- Basic idea: After decoding codeword $\mathbf{x}_\ell$, remove its effect from channel output to reduce the interference between data streams.
- V-BLAST I: Decodes and cancel the data streams in a predetermined order, irrespective of the channel realization.

Successive Interference Cancellation

- Linear receiver architectures are often augmented using successive interference cancellation (SIC).
- Basic idea: After decoding codeword $\mathbf{x}_\ell$, remove its effect from channel output to reduce the interference between data streams.
- V-BLAST I: Decodes and cancel the data streams in a predetermined order, irrespective of the channel realization.
- V-BLAST II: Select the decoding order for each channel realization to maximize the effective SNR for the data stream that sees the worst channel.

Successive Interference Cancellation

- Linear receiver architectures are often augmented using successive interference cancellation (SIC).
- Basic idea: After decoding codeword $\mathbf{x}_\ell$, remove its effect from channel output to reduce the interference between data streams.
- V-BLAST I: Decodes and cancel the data streams in a predetermined order, irrespective of the channel realization.
- V-BLAST II: Select the decoding order for each channel realization to maximize the effective SNR for the data stream that sees the worst channel.
- V-BLAST III: Decodes and cancel the data streams in a predetermined order. The rate of each data stream is selected to maximize the sum rate. (Outside problem statement.)

Simulation: Outage Rates

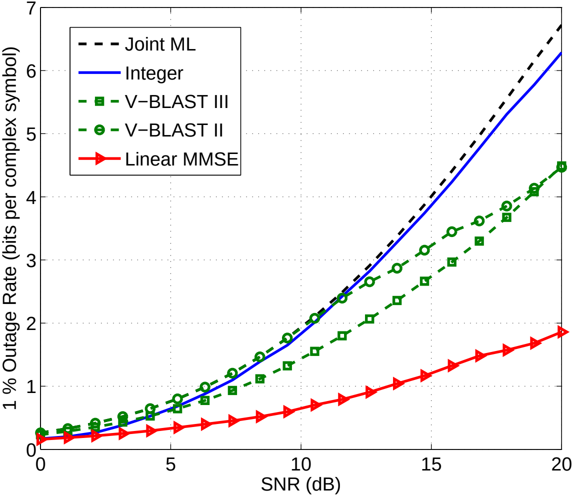
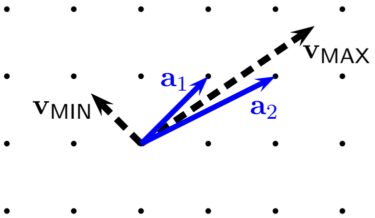


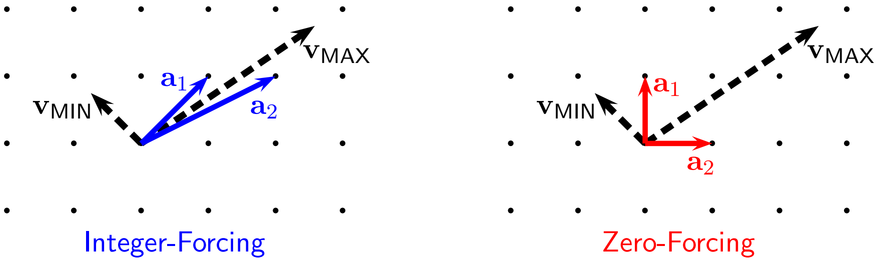
Figure: 1 percent outage rates for the 2×2 complex-valued MIMO channel with Rayleigh fading.

- Integer-forcing can adapt to the channel by choosing a basis (of integer vectors) close to the maximum singular vector.



Integer-Forcing

- Integer-forcing can adapt to the channel by choosing a basis (of integer vectors) close to the maximum singular vector.
- Zero-forcing implicitly decodes using the standard basis.



- **Zheng-Tse '03:** A family of codes is said to achieve spatial multiplexing gain r and diversity gain d if the total data rate and the average probability of error satisfy

$$\lim_{\mathsf{SNR}\to\infty} \frac{R(\mathsf{SNR})}{\log \mathsf{SNR}} \geq r$$

$$\lim_{\mathsf{SNR}\to\infty} \frac{\log p_{\mathsf{error}}(\mathsf{SNR})}{\log \mathsf{SNR}} \leq -d.$$

Diversity-Multiplexing Tradeoff

- **Zheng-Tse '03:** A family of codes is said to achieve spatial multiplexing gain r and diversity gain d if the total data rate and the average probability of error satisfy

$$\lim_{\mathsf{SNR}\to\infty} \frac{R(\mathsf{SNR})}{\log \mathsf{SNR}} \geq r$$
$$\lim_{\mathsf{SNR}\to\infty} \frac{\log p_{\mathsf{error}}(\mathsf{SNR})}{\log \mathsf{SNR}} \leq -d.$$

- **Zhan-Nazer-Erez-Gastpar '14:** Integer-forcing can attain the optimal DMT while conventional linear receivers cannot.

- **Zheng-Tse '03:** The DMTs achieved by the zero-forcing, linear MMSE, and successive interference cancellation architectures are

$$d_{\text{ZF}}(r) = d_{\text{LMMSE}}(r) = d_{\text{V-BLAST I}}(r) = \left(1 - \frac{r}{M}\right)$$

$$d_{\text{V-BLAST II}}(r) \leq (N-1)\left(1 - \frac{r}{M}\right)$$

$$d_{\text{V-BLAST III}}(r) = \text{piecewise linear curve connecting points } (r_k, n-k)$$

$$\text{where } r_0 = 0, r_k = \sum_{i=0}^{k-1} \frac{k-i}{n-i} \quad 1 \leq k \leq n$$

Diversity-Multiplexing Tradeoff

- **Zheng-Tse '03:** The DMTs achieved by the zero-forcing, linear MMSE, and successive interference cancellation architectures are

$$d_{\text{ZF}}(r) = d_{\text{LMMSE}}(r) = d_{\text{V-BLAST I}}(r) = \left(1 - \frac{r}{M}\right)$$

$$d_{\text{V-BLAST II}}(r) \leq (N-1)\left(1 - \frac{r}{M}\right)$$

$$d_{\text{V-BLAST III}}(r) = \text{piecewise linear curve connecting points } (r_k, n-k)$$

$$\text{where } r_0 = 0, r_k = \sum_{i=0}^{k-1} \frac{k-i}{n-i} \quad 1 \leq k \leq n$$

- **Zhan-Nazer-Erez-Gastpar '14:** Integer-forcing recovers the optimal DMT for $N \geq M$ receive antennas:

$$d_{\text{IF}}(r) = N\left(1 - \frac{r}{M}\right)$$

- What about space-time coding at the transmitter?

Extensions and Generalizations

- What about space-time coding at the transmitter?
- **Ordentlich-Erez '13:** Linear dispersion codes + integer-forcing achieves the MIMO capacity universally to within a constant gap. Includes optimal DMT as a special case.

- What about space-time coding at the transmitter?
- **Ordentlich-Erez '13:** Linear dispersion codes + integer-forcing achieves the MIMO capacity universally to within a constant gap. Includes optimal DMT as a special case.
- What about downlink scenarios?

Extensions and Generalizations

- What about space-time coding at the transmitter?
- **Ordentlich-Erez '13:** Linear dispersion codes + integer-forcing achieves the MIMO capacity universally to within a constant gap. Includes optimal DMT as a special case.
- What about downlink scenarios?
- **Hong-Caire '13:** Proposed integer-forcing beamforming. Each user decodes the linear combination with the least effective noise. The transmitter "pre-inverts" linear combinations using the inverse of $[A] \bmod p$ over $\mathbb{Z}_p$ so that each user obtains its desired message.

- What about space-time coding at the transmitter?
- **Ordentlich-Erez '13:** Linear dispersion codes + integer-forcing achieves the MIMO capacity universally to within a constant gap. Includes optimal DMT as a special case.
- What about downlink scenarios?
- **Hong-Caire '13:** Proposed integer-forcing beamforming. Each user decodes the linear combination with the least effective noise. The transmitter "pre-inverts" linear combinations using the inverse of $[A] \bmod p$ over $\mathbb{Z}_p$ so that each user obtains its desired message.
- **He-Nazer-Shamai '14:** Established uplink-downlink duality.

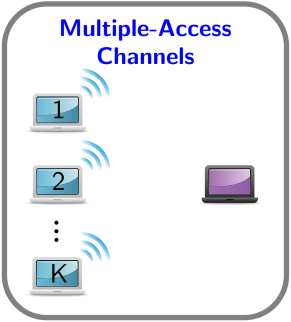
Extensions and Generalizations

- What about space-time coding at the transmitter?
- **Ordentlich-Erez '13:** Linear dispersion codes + integer-forcing achieves the MIMO capacity universally to within a constant gap. Includes optimal DMT as a special case.
- What about downlink scenarios?
- **Hong-Caire '13:** Proposed integer-forcing beamforming. Each user decodes the linear combination with the least effective noise. The transmitter "pre-inverts" linear combinations using the inverse of $[A] \bmod p$ over $\mathbb{Z}_p$ so that each user obtains its desired message.
- **He-Nazer-Shamai '14:** Established uplink-downlink duality.
- What about successive cancellation for integer-forcing?

Extensions and Generalizations

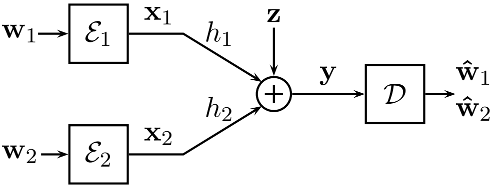
- What about space-time coding at the transmitter?
- **Ordentlich-Erez '13:** Linear dispersion codes + integer-forcing achieves the MIMO capacity universally to within a constant gap. Includes optimal DMT as a special case.
- What about downlink scenarios?
- **Hong-Caire '13:** Proposed integer-forcing beamforming. Each user decodes the linear combination with the least effective noise. The transmitter "pre-inverts" linear combinations using the inverse of $[A] \bmod p$ over $\mathbb{Z}_p$ so that each user obtains its desired message.
- **He-Nazer-Shamai '14:** Established uplink-downlink duality.
- What about successive cancellation for integer-forcing?
- **Ordentlich-Erez-Nazer '13:** Framework for IF-SIC and exact optimality proof.

- Warm-up: Compute-and-forward over finite field channels.
- Compute-and-forward over Gaussian channels.
- Applications to communication across single-hop Gaussian networks.



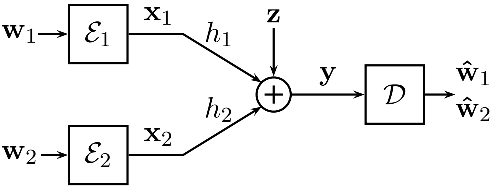
Joint work with Or Ordentlich and Uri Erez.

Gaussian Multiple-Access Channel

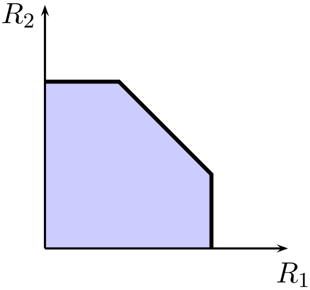


$$\|\mathbf{x}_\ell\|^2 \leq nP, \quad \mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}).$$

Gaussian Multiple-Access Channel



$$\|\mathbf{x}_\ell\|^2 \le nP, \quad \mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}).$$



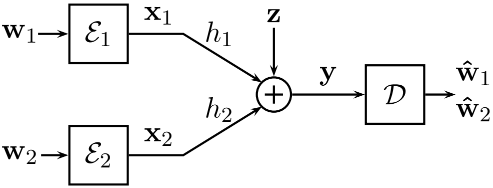
Theorem (Ahlswede '71, Liao '72, Wyner '74, Cover '75)

The capacity region is the set of all rate pairs (R_1, R_2) *satisfying:*

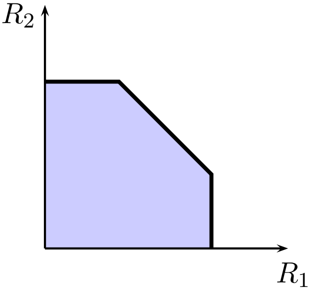
$$R_1 < \frac{1}{2}\log(1 + h_1^2 P) \qquad R_2 < \frac{1}{2}\log(1 + h_2^2 P)$$

$$R_1 + R_2 < \frac{1}{2}\log(1 + \|\mathbf{h}\|^2 P)$$

Gaussian Multiple-Access Channel



$$\|\mathbf{x}_\ell\|^2 \leq nP, \quad \mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}).$$



Theorem (Ahlswede '71, Liao '72, Wyner '74, Cover '75)

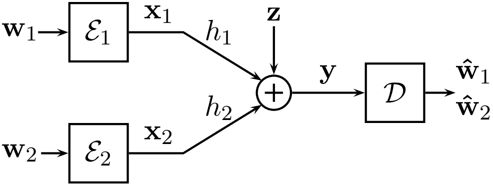
The capacity region is the set of all rate pairs (R_1, R_2) *satisfying:*

$$R_1 < \frac{1}{2}\log(1 + h_1^2 P) \qquad R_2 < \frac{1}{2}\log(1 + h_2^2 P)$$

$$R_1 + R_2 < \frac{1}{2}\log(1 + \|\mathbf{h}\|^2 P)$$

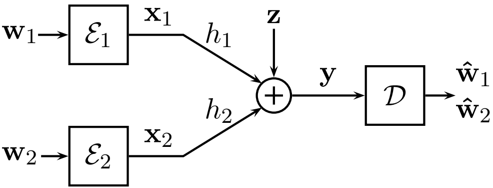
Achievable via joint decoding.

Successive Cancellation

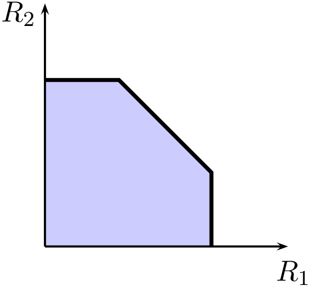


$$\|\mathbf{x}_\ell\|^2 \le nP, \quad \mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}).$$

Successive Cancellation

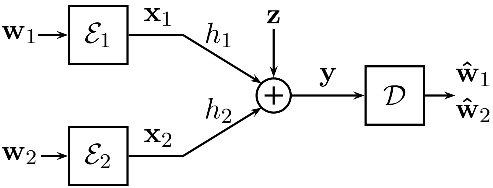


$$\|\mathbf{x}_\ell\|^2 \leq nP, \quad \mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}).$$

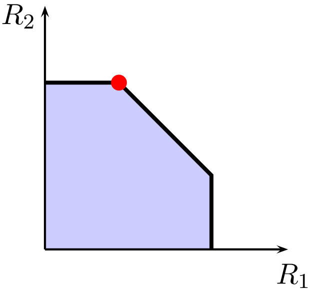


- Treat $\mathbf{x}_2$ as noise and decode $\mathbf{x}_1$, $R_1 < \frac{1}{2}\log\left(1 + \frac{h_1^2 P}{1 + h_2^2 P}\right)$.

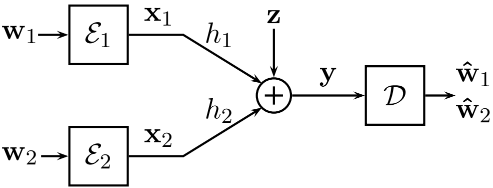
Successive Cancellation



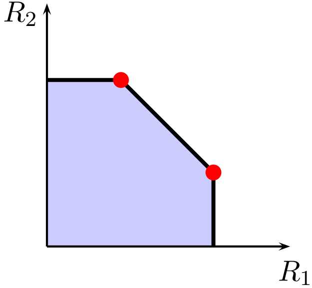
$$\|\mathbf{x}_\ell\|^2 \leq nP, \quad \mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}).$$



- Treat $\mathbf{x}_2$ as noise and decode $\mathbf{x}_1$, $R_1 < \frac{1}{2}\log\left(1 + \frac{h_1^2 P}{1 + h_2^2 P}\right)$.
- Cancel $\mathbf{x}_1$ and decode $\mathbf{x}_2$, $R_2 < \frac{1}{2}\log\left(1 + h_2^2 P\right)$.

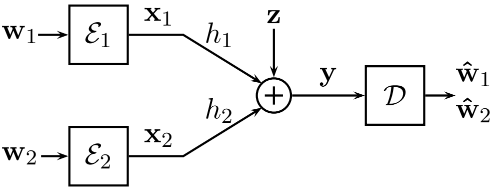


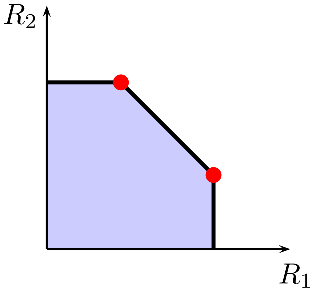
$$\|\mathbf{x}_\ell\|^2 \leq nP, \quad \mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}).$$



- Treat $\mathbf{x}_2$ as noise and decode $\mathbf{x}_1$, $R_1 < \frac{1}{2}\log\left(1 + \frac{h_1^2 P}{1 + h_2^2 P}\right)$.
- Cancel $\mathbf{x}_1$ and decode $\mathbf{x}_2$, $R_2 < \frac{1}{2}\log\left(1 + h_2^2 P\right)$.
- Switch decoding order for the other corner point.

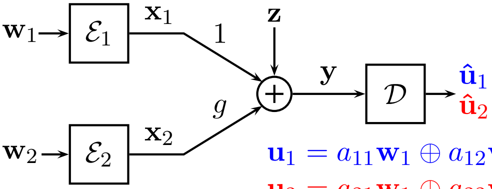
Successive Cancellation





$$\|\mathbf{x}_\ell\|^2 \leq nP, \quad \mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}).$$

- Treat $\mathbf{x}_2$ as noise and decode $\mathbf{x}_1$, $R_1 < \frac{1}{2}\log\left(1 + \frac{h_1^2 P}{1 + h_2^2 P}\right)$.
- Cancel $\mathbf{x}_1$ and decode $\mathbf{x}_2$, $R_2 < \frac{1}{2}\log\left(1 + h_2^2 P\right)$.
- Switch decoding order for the other corner point.
- Achieves capacity when combined with time-sharing or rate-splitting (**Rimoldi-Urbanke '96**).

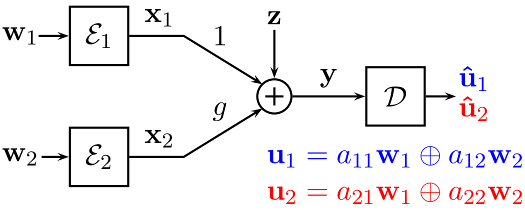


$$\mathbf{u}_1 = a_{11}\mathbf{w}_1 \oplus a_{12}\mathbf{w}_2$$

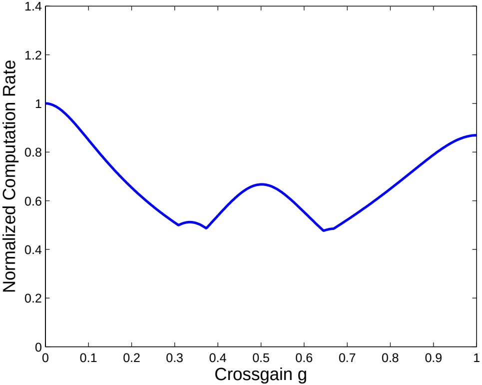
$$\mathbf{u}_2 = a_{21}\mathbf{w}_1 \oplus a_{22}\mathbf{w}_2$$

- Decode one linear combination.
- Plot rate normalized by the MAC sum rate $\frac{1}{2}\log(1 + (1 + g^2)P)$.

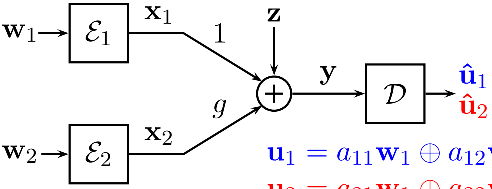
Two Linear Combinations



- Decode one linear combination.
- Plot rate normalized by the MAC sum rate $\frac{1}{2}\log(1+(1+g^2)P)$.

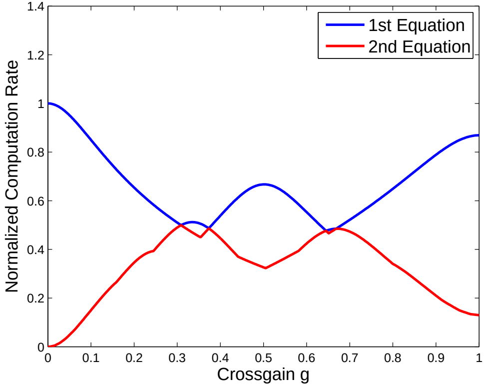


Two Linear Combinations

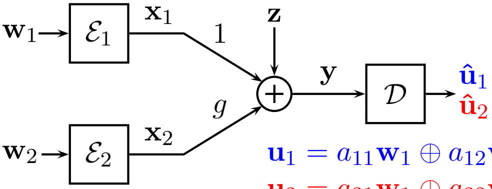


$$\mathbf{u}_1 = a_{11}\mathbf{w}_1 \oplus a_{12}\mathbf{w}_2$$
$$\mathbf{u}_2 = a_{21}\mathbf{w}_1 \oplus a_{22}\mathbf{w}_2$$

- Decode *two linearly independent* linear combinations.
- Plot rate normalized by the MAC sum rate $\frac{1}{2}\log(1 + (1 + g^2)P)$.

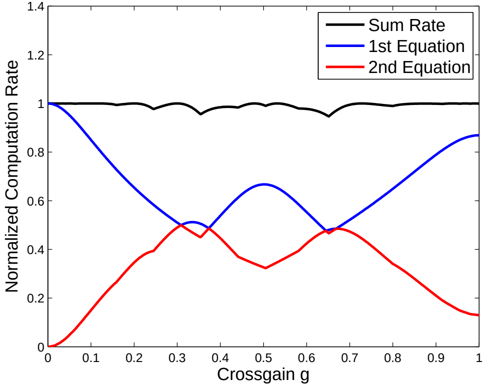


Two Linear Combinations

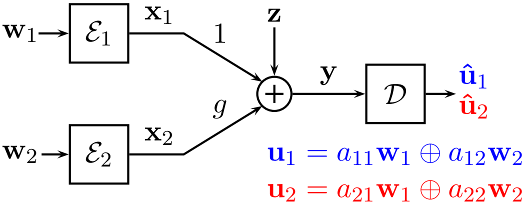


$$\mathbf{u}_1 = a_{11}\mathbf{w}_1 \oplus a_{12}\mathbf{w}_2$$
$$\mathbf{u}_2 = a_{21}\mathbf{w}_1 \oplus a_{22}\mathbf{w}_2$$

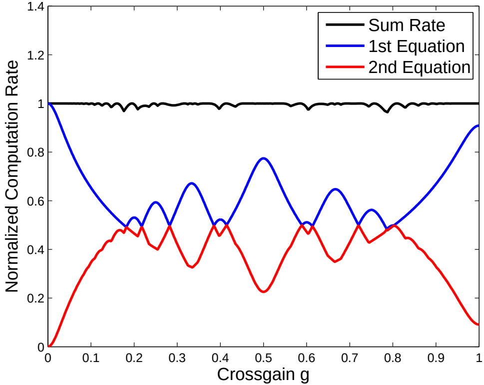
- Decode *two linearly independent* linear combinations.
- Plot rate normalized by the MAC sum rate $\frac{1}{2}\log(1+(1+g^2)P)$.



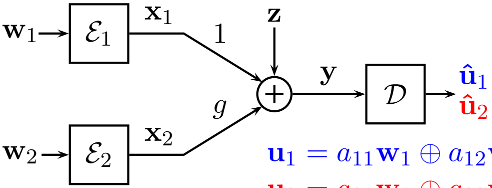
Two Linear Combinations



- Decode *two linearly independent* linear combinations.
- Plot rate normalized by the MAC sum rate $\frac{1}{2}\log(1+(1+g^2)P)$.

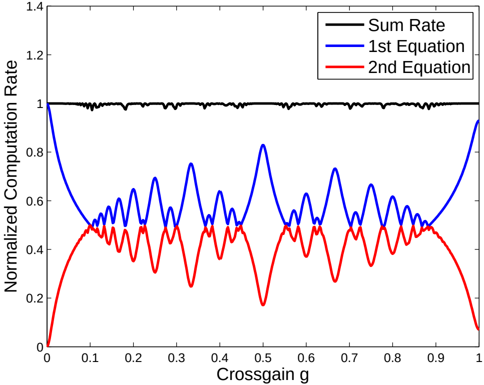


Two Linear Combinations



$$\mathbf{u}_1 = a_{11}\mathbf{w}_1 \oplus a_{12}\mathbf{w}_2$$
$$\mathbf{u}_2 = a_{21}\mathbf{w}_1 \oplus a_{22}\mathbf{w}_2$$

- Decode *two linearly independent* linear combinations.
- Plot rate normalized by the MAC sum rate $\frac{1}{2}\log(1 + (1 + g^2)P)$.



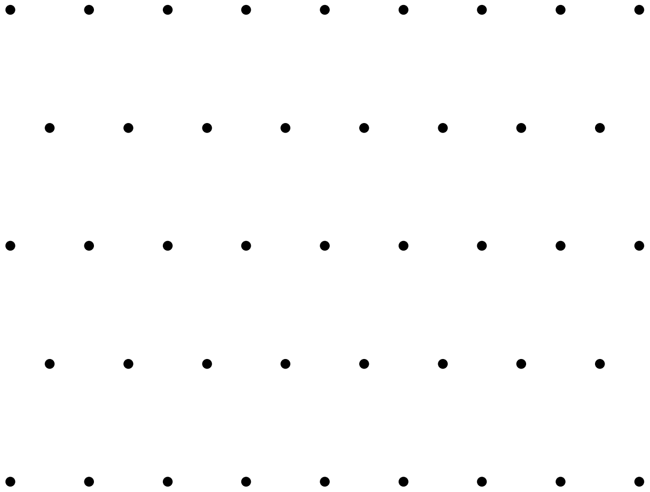
- Looks as if the sum of computation rates is nearly equal to the MAC sum capacity. Why is this happening?

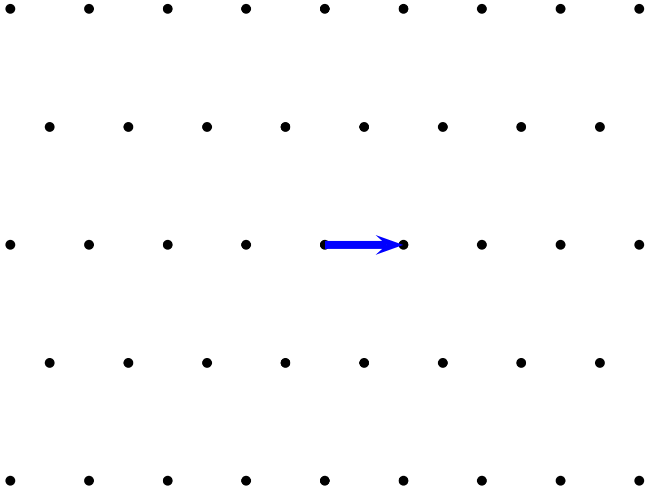
- Let $\mathbf{F} = \left(P^{-1/2}\mathbf{I} + \ \mathbf{h}\mathbf{h}^{\mathsf{T}}\right)^{-1/2}$. Then, each computation rate can be written as
$$R_{\mathsf{comp}}(\mathbf{h}, \mathbf{a}_k) = \frac{1}{2}\log^{+}\left(\frac{P}{\|\mathbf{F}\ \mathbf{a}_k\|^2}\right) .$$

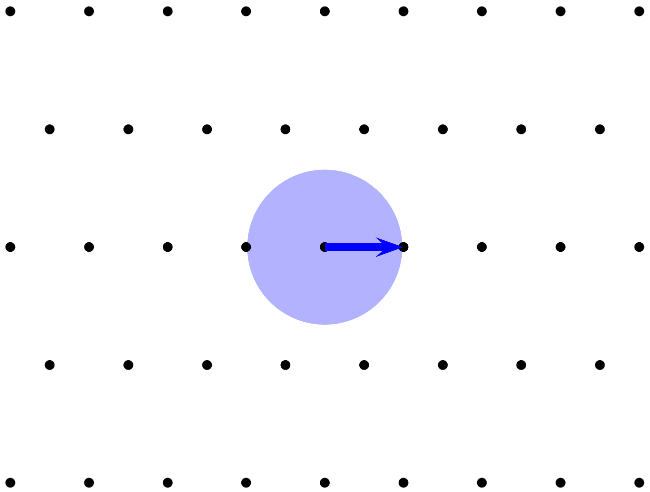
- Thus, decoding the best linear combinations is the same as finding the successive minima $\lambda_k(\mathbf{F})$ for the lattice $\Lambda(\mathbf{F}) = \mathbf{F}\mathbb{Z}^K$:

$$\lambda_k(\mathbf{F}) \triangleq \inf\left\{r : \dim\left(\operatorname{span}\left(\Lambda(\mathbf{F}) \ \cap \ \mathcal{B}(\mathbf{0}, r)\right)\right) \geq k\right\} .$$

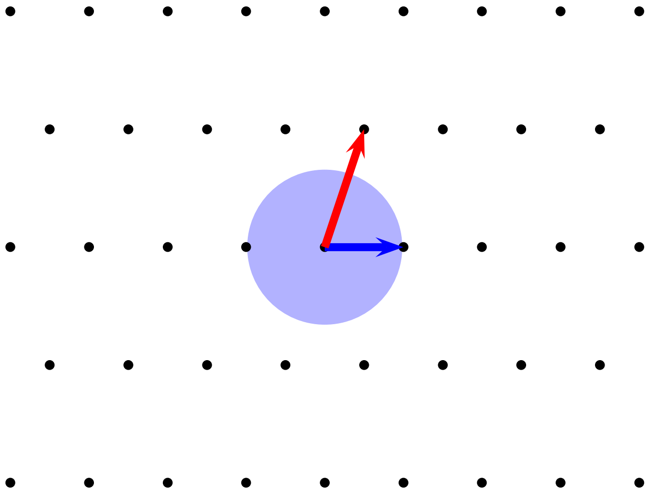
Successive Minima



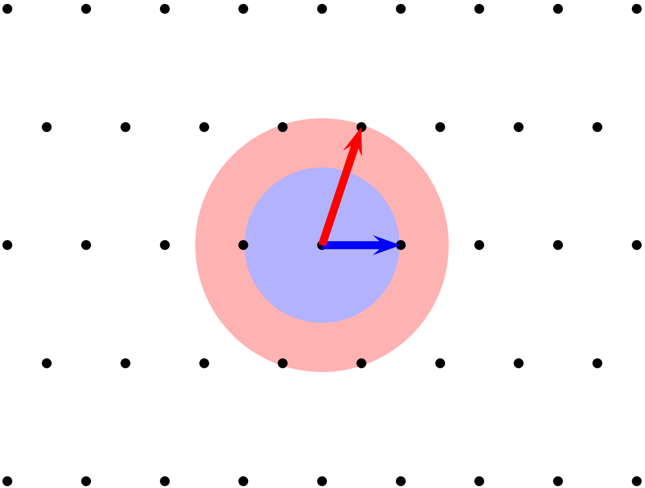




Successive Minima



Successive Minima



Minkowski's Theorem on Successive Minima

Theorem (Minkowski)

Let $\Lambda(\mathbf{F})$ be a lattice spanned by a full-rank $K \times K$ matrix $\mathbf{F}$. Its successive minima $\lambda_k(\mathbf{F})$ satisfy

$$\prod_{k=1}^{K} \lambda_k^2(\mathbf{F}) \leq K^K \big| \det(\mathbf{F}) \big|^2 \ .$$

Minkowski's Theorem on Successive Minima

Theorem (Minkowski)

Let $\Lambda(\mathbf{F})$ *be a lattice spanned by a full-rank* $K \times K$ *matrix* $\mathbf{F}$*. Its successive minima* $\lambda_k(\mathbf{F})$ *satisfy*

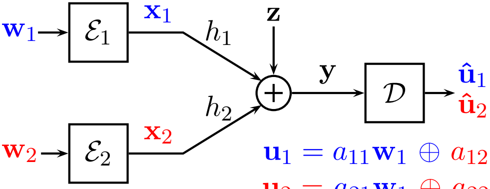
$$\prod_{k=1}^{K} \lambda_k^2(\mathbf{F}) \leq K^K \big|\det(\mathbf{F})\big|^2 \ .$$

Theorem (Ordentlich-Erez-Nazer '14)

For any channel vector $\mathbf{h} \in \mathbb{R}^K$*, there exist linearly independent integer vectors* $\mathbf{a}_1, \ldots, \mathbf{a}_K \in \mathbb{Z}^K$ *satisfying*

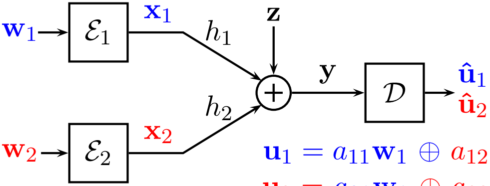
$$\sum_{k=1}^{K} R_{\textit{comp}}(\mathbf{h}, \mathbf{a}_k) \geq \frac{1}{2} \log(1 + \|\mathbf{h}\|^2 \mathsf{SNR}) - \frac{K}{2} \log K \ .$$

Operational Interpretation: Multiple-Access



$$\mathbf{u}_1 = a_{11}\mathbf{w}_1 \oplus a_{12}\mathbf{w}_2$$
$$\mathbf{u}_2 = a_{21}\mathbf{w}_1 \oplus a_{22}\mathbf{w}_2$$

- Order linear combinations by descending computation rate.
- Associate each computation rate to a message.

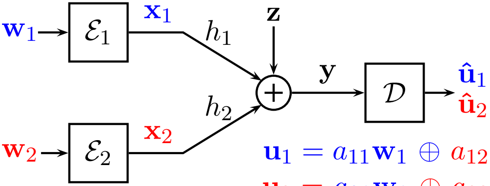


$$\mathbf{u}_1 = a_{11}\mathbf{w}_1 \oplus a_{12}\mathbf{w}_2$$
$$\mathbf{u}_2 = a_{21}\mathbf{w}_1 \oplus a_{22}\mathbf{w}_2$$

- Order linear combinations by descending computation rate.
- Associate each computation rate to a message.

- Decode $\mathbf{u}_1$ first.

Operational Interpretation: Multiple-Access

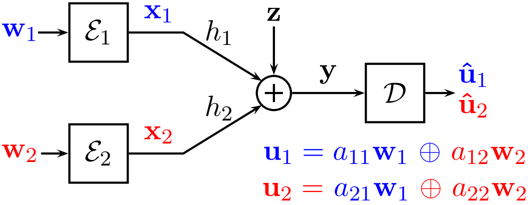


$$\mathbf{u}_1 = a_{11}\mathbf{w}_1 \oplus a_{12}\mathbf{w}_2$$
$$\mathbf{u}_2 = a_{21}\mathbf{w}_1 \oplus a_{22}\mathbf{w}_2$$

- Order linear combinations by descending computation rate.
- Associate each computation rate to a message.

- Decode $\mathbf{u}_1$ first.
- Use $\mathbf{u}_1$ to help decode $\mathbf{u}_2$ by canceling out the contribution of $\mathbf{w}_1$, in order to lower the effective rate.

Operational Interpretation: Multiple-Access



- Order linear combinations by descending computation rate.
- Associate each computation rate to a message.

- Decode $\mathbf{u}_1$ first.
- Use $\mathbf{u}_1$ to help decode $\mathbf{u}_2$ by canceling out the contribution of $\mathbf{w}_1$, in order to lower the effective rate.

Theorem (Ordentlich-Erez-Nazer '14)

For any linearly independent integer vectors $\mathbf{a}_1, \ldots, \mathbf{a}_K \in \mathbb{Z}^K$, there exists a permutation π such that the following rates are achievable:

$$R_\ell = R_{\textit{comp},\pi(\ell)} \ .$$

(Algebraic) Successive Cancellation

- After decoding the first linear combination, the receiver knows

$$\mathbf{v}_1 = \left[a_{11}\mathbf{t}_1 + a_{12}\mathbf{t}_2\right] \bmod \Lambda_{\mathsf{C}} \ .$$

(Algebraic) Successive Cancellation

- After decoding the first linear combination, the receiver knows

$$\mathbf{v}_1 = \left[a_{11}\mathbf{t}_1 + a_{12}\mathbf{t}_2\right] \bmod \Lambda_{\mathsf{C}} \ .$$

- The effective channel for the second linear combination is

$$\tilde{\mathbf{y}}_2 = \left[a_{21}\mathbf{t}_1 + a_{22}\mathbf{t}_2 + \mathbf{z}_{\mathsf{effec}}(\mathbf{h}, \mathbf{a}_2)\right] \bmod \Lambda_{\mathsf{C}} \ .$$

(Algebraic) Successive Cancellation

- After decoding the first linear combination, the receiver knows

$$\mathbf{v}_1 = \big[a_{11}\mathbf{t}_1 + a_{12}\mathbf{t}_2\big] \bmod \Lambda_{\mathsf{C}} \ .$$

- The effective channel for the second linear combination is

$$\tilde{\mathbf{y}}_2 = \big[a_{21}\mathbf{t}_1 + a_{22}\mathbf{t}_2 + \mathbf{z}_{\text{effec}}(\mathbf{h}, \mathbf{a}_2)\big] \bmod \Lambda_{\mathsf{C}} \ .$$

- Using $\mathbf{v}_1$, we can cancel out $\mathbf{t}_1$ from $\tilde{\mathbf{y}}_2$ without changing the effective noise.

$$\begin{aligned} \tilde{\mathbf{y}}_2^{\mathsf{SI}} &= \Big[\mathbf{s}_2 - b_1\mathbf{v}_1\Big] \bmod \Lambda_{\mathsf{C}} \\ &= \Big[\big(a_{22} - b_1 a_{12}\big)\mathbf{t}_2 + \mathbf{z}_{\text{effec}}(\mathbf{h}, \mathbf{a}_2)\Big] \bmod \Lambda_{\mathsf{C}} \ . \end{aligned}$$

(Algebraic) Successive Cancellation

- After decoding the first linear combination, the receiver knows

$$\mathbf{v}_1 = \left[a_{11}\mathbf{t}_1 + a_{12}\mathbf{t}_2\right] \bmod \Lambda_{\mathsf{C}} \ .$$

- The effective channel for the second linear combination is

$$\tilde{\mathbf{y}}_2 = \left[a_{21}\mathbf{t}_1 + a_{22}\mathbf{t}_2 + \mathbf{z}_{\mathsf{effec}}(\mathbf{h}, \mathbf{a}_2)\right] \bmod \Lambda_{\mathsf{C}} \ .$$

- Using $\mathbf{v}_1$, we can cancel out $\mathbf{t}_1$ from $\tilde{\mathbf{y}}_2$ without changing the effective noise.

$$\begin{aligned}\tilde{\mathbf{y}}_2^{\mathsf{SI}} &= \Big[\mathbf{s}_2 - b_1\mathbf{v}_1\Big] \bmod \Lambda_{\mathsf{C}} \\ &= \Big[\big(a_{22} - b_1 a_{12}\big)\mathbf{t}_2 + \mathbf{z}_{\mathsf{effec}}(\mathbf{h}, \mathbf{a}_2)\Big] \bmod \Lambda_{\mathsf{C}} \ .\end{aligned}$$

- Now, the receiver can decode since $R_2 < R_{\mathsf{comp}}(\mathbf{h}, \mathbf{a}_2)$.

Using One Linear Combination to Get Another

- Basic Idea: After decoding the first linear combination with coefficients $\mathbf{a}$, we should create a new effective channel with coefficients $\mathbf{h} + \beta\mathbf{a}$ to make it easier to decode the second linear combination.

- We need the real sum of codewords $\sum_{\ell} a_\ell \mathbf{x}_\ell$.

- Issue: Our decoding scheme recovers the modulo sum of lattice points $\left[\sum_\ell a_\ell \mathbf{t}_\ell\right] \bmod \Lambda_{\mathsf{C}}$ on the way to the linear combination of messages, not the real sum.

- So far, we have only decoded a modulo sum of the lattice points:

$$\Big[\sum_{\ell} a_{\ell}\mathbf{t}_{\ell}\Big] \bmod \Lambda_{\mathsf{C}} \ .$$

Successive Computation

- So far, we have only decoded a modulo sum of the lattice points:

$$\Big[\sum_{\ell} a_\ell \mathbf{t}_\ell\Big] \bmod \Lambda_{\mathsf{C}} \ .$$

- First, add back in the dithers to get the modulo sum of codewords:

$$\Big[\Big[\sum_{\ell} a_\ell \mathbf{t}_\ell\Big] \bmod \Lambda_{\mathsf{C}} + \Big[\sum_{\ell} a_\ell \mathbf{d}_\ell\Big] \bmod \Lambda_{\mathsf{C}}\Big] \bmod \Lambda_{\mathsf{C}} = \Big[\sum_{\ell} a_\ell \mathbf{x}_\ell\Big] \bmod \Lambda_{\mathsf{C}}$$

Successive Computation

- So far, we have only decoded a modulo sum of the lattice points:

$$\Big[\sum_{\ell} a_\ell \mathbf{t}_\ell\Big] \bmod \Lambda_{\mathsf{C}} \ .$$

- First, add back in the dithers to get the modulo sum of codewords:

$$\Bigg[\Big[\sum_{\ell} a_\ell \mathbf{t}_\ell\Big] \bmod \Lambda_{\mathsf{C}} + \Big[\sum_{\ell} a_\ell \mathbf{d}_\ell\Big] \bmod \Lambda_{\mathsf{C}}\Bigg] \bmod \Lambda_{\mathsf{C}} = \Big[\sum_{\ell} a_\ell \mathbf{x}_\ell\Big] \bmod \Lambda_{\mathsf{C}}$$

- Subtract this from $\mathbf{y}$ to expose the coarse lattice point nearest to the real sum:

$$\mathbf{y} - \Big[\sum_{\ell} a_\ell \mathbf{x}_\ell\Big] \bmod \Lambda_{\mathsf{C}} = Q_{\Lambda_{\mathsf{C}}}\Big(\sum_{\ell} a_\ell \mathbf{x}_\ell\Big) + \sum_{\ell} (h_\ell - a_\ell)\mathbf{x}_\ell + \mathbf{z}$$

Successive Computation

- So far, we have only decoded a modulo sum of the lattice points:

$$\Big[\sum_{\ell} a_\ell \mathbf{t}_\ell\Big] \bmod \Lambda_{\mathsf{C}} \ .$$

- First, add back in the dithers to get the modulo sum of codewords:

$$\bigg[\Big[\sum_{\ell} a_\ell \mathbf{t}_\ell\Big] \bmod \Lambda_{\mathsf{C}} + \Big[\sum_{\ell} a_\ell \mathbf{d}_\ell\Big] \bmod \Lambda_{\mathsf{C}}\bigg] \bmod \Lambda_{\mathsf{C}} = \Big[\sum_{\ell} a_\ell \mathbf{x}_\ell\Big] \bmod \Lambda_{\mathsf{C}}$$

- Subtract this from $\mathbf{y}$ to expose the coarse lattice point nearest to the real sum:

$$\mathbf{y} - \Big[\sum_{\ell} a_\ell \mathbf{x}_\ell\Big] \bmod \Lambda_{\mathsf{C}} = Q_{\Lambda_{\mathsf{C}}}\Big(\sum_{\ell} a_\ell \mathbf{x}_\ell\Big) + \sum_{\ell} (h_\ell - a_\ell)\mathbf{x}_\ell + \mathbf{z}$$

- Coarse lattice point easier to decode than fine lattice point:

$$Q_{\Lambda_{\mathsf{C}}}\left(Q_{\Lambda_{\mathsf{C}}}\Big(\sum_{\ell} a_\ell \mathbf{x}_\ell\Big) + \sum_{\ell} (h_\ell - a_\ell)\mathbf{x}_\ell + \mathbf{z}\right) = Q_{\Lambda_{\mathsf{C}}}\Big(\sum_{\ell} a_\ell \mathbf{x}_\ell\Big) \quad \text{w.h.p.}$$

Successive Computation

- Modulo sum is just the quantization error of the real sum with respect to the coarse lattice.

- Combine the modulo sum with the quantized sum to get back the real sum:

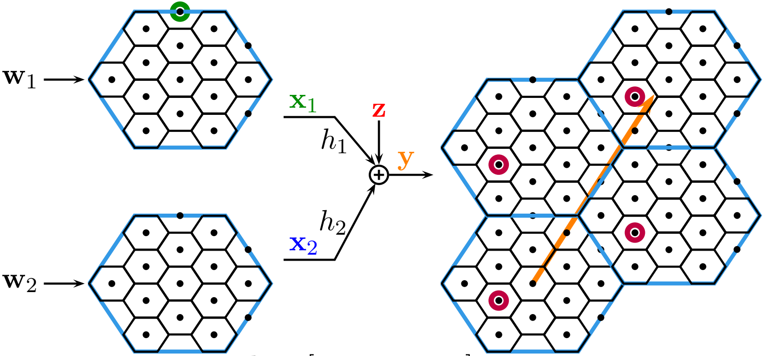
$$\Big[\sum_{\ell} a_\ell \mathbf{x}_\ell\Big] \bmod \Lambda_{\mathsf{C}} + Q_{\Lambda_{\mathsf{C}}}\Big(\sum_{\ell} a_\ell \mathbf{x}_\ell\Big) = \sum_{\ell} a_\ell \mathbf{x}_\ell$$

Lemma

In the compute-and-forward framework, if you can recover the modulo sum, you can also recover the real sum (with high probability).

Successive Computation Illustration

We have the modulo sum.



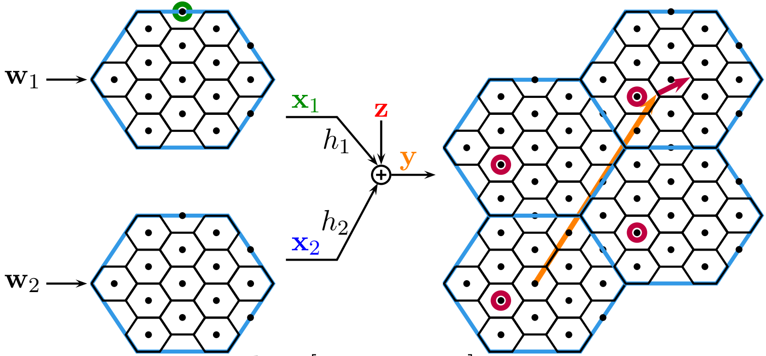
$$\alpha\mathbf{h} = [\ \alpha 1.4 \quad \alpha 2.1\ ]$$

$$\mathbf{a} = [\ 2 \quad 3\ ]$$

Effective noise: $\alpha^2 + \mathsf{SNR}\|\alpha\mathbf{h} - \mathbf{a}\|^2$

Successive Computation Illustration

Subtract modulo sum from the received signal.



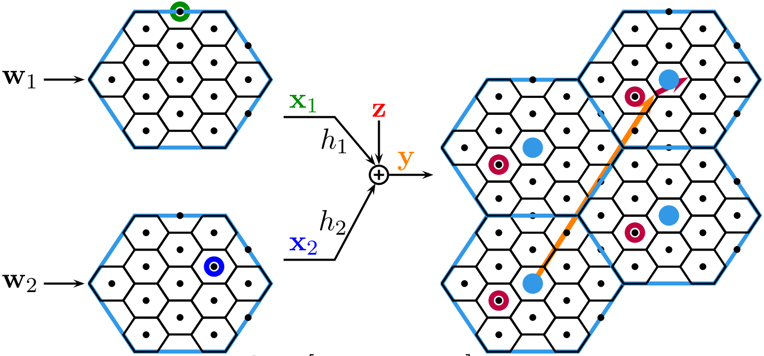
$$\alpha\mathbf{h} = [\ \alpha 1.4 \quad \alpha 2.1\ ]$$

$$\mathbf{a} = [\ 2 \quad 3\ ]$$

Effective noise: $\alpha^2 + \mathsf{SNR}\|\alpha\mathbf{h} - \mathbf{a}\|^2$

Successive Computation Illustration

Decode to the closest coarse lattice point.



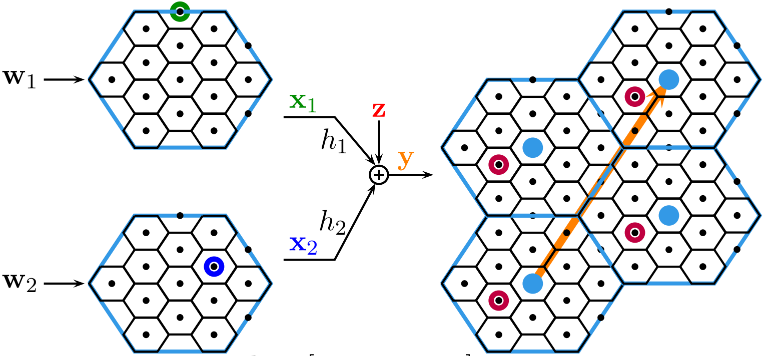
$$\alpha \mathbf{h} = [\ \alpha 1.4 \quad \alpha 2.1\ ]$$

$$\mathbf{a} = [\ 2 \quad 3\ ]$$

Effective noise: $\alpha^2 + \mathsf{SNR}\|\alpha \mathbf{h} - \mathbf{a}\|^2$

Successive Computation Illustration

Decode to the closest coarse lattice point.



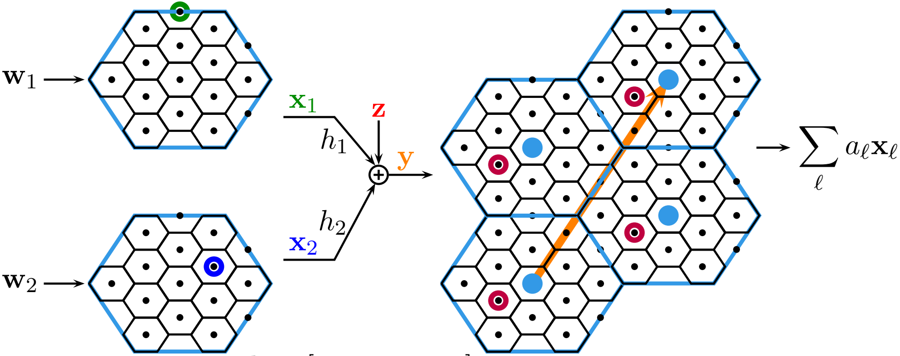
$$\alpha \mathbf{h} = [\ \alpha 1.4 \quad \alpha 2.1\ ]$$

$$\mathbf{a} = [\ 2 \quad 3\ ]$$

Effective noise: $\alpha^2 + \mathsf{SNR}\|\alpha \mathbf{h} - \mathbf{a}\|^2$

Successive Computation Illustration

Now we can infer the real sum.



$$\alpha\mathbf{h} = [\ \alpha 1.4 \quad \alpha 2.1\ ]$$

$$\mathbf{a} = [\ \ 2 \qquad 3\ \ ]$$

Effective noise: $\alpha^2 + \mathsf{SNR}\|\alpha\mathbf{h} - \mathbf{a}\|^2$

Successive Computation

- Receiver observes $\mathbf{y} = \sum_{\ell=1}^{L} h_\ell \mathbf{x}_\ell + \mathbf{z}$

Successive cancellation:

- Decode $\mathbf{x}_i$.
- Calculate $\mathbf{y} - h_i \mathbf{x}_i$.
- Receiver now has

$$\sum_{\ell \neq i} h_\ell \mathbf{x}_\ell + \mathbf{z}$$

Successive Computation

- Receiver observes $\mathbf{y} = \sum_{\ell=1}^{L} h_\ell \mathbf{x}_\ell + \mathbf{z}$

Successive cancellation:

- Decode $\mathbf{x}_i$.
- Calculate $\mathbf{y} - h_i \mathbf{x}_i$.
- Receiver now has

$$\sum_{\ell \neq i} h_\ell \mathbf{x}_\ell + \mathbf{z}$$

Successive computation:

- Decode $\sum_{\ell=1}^{L} a_\ell \mathbf{x}_\ell$.
- Calculate $\mathbf{y} + \beta \sum_{\ell=1}^{L} a_\ell \mathbf{x}_\ell$.
- Receiver now has

$$\sum_{\ell=1}^{L} (h_\ell + \beta a_\ell) \mathbf{x}_\ell + \mathbf{z}$$

Exact Sum-Rate Optimality

- **Key Idea:** Use recovered linear combinations to form a better effective channel for decoding subsequent linear combinations (as well as successive cancellation).

- **Key Idea:** Use recovered linear combinations to form a better effective channel for decoding subsequent linear combinations (as well as successive cancellation).
- Define the *successive effective noise variance*

$$\sigma^2_{\text{eff}}(\mathbf{h}, \mathbf{a}_m | \mathbf{a}_1, \ldots, \mathbf{a}_{m-1}) = \|\mathbf{C}_m^{\perp} \mathbf{F} \mathbf{a}_m\|^2$$

where $\mathbf{F} = \left(P^{-1}\mathbf{I} + \mathbf{h}\mathbf{h}^{\mathsf{T}}\right)^{-1/2}$ and $\mathbf{C}_m^{\perp}$ is the projection matrix for the nullspace of $\mathbf{F}[\mathbf{a}_1 \ \cdots \ \mathbf{a}_{m-1}]$.

Exact Sum-Rate Optimality

- **Key Idea:** Use recovered linear combinations to form a better effective channel for decoding subsequent linear combinations (as well as successive cancellation).
- Define the *successive effective noise variance*

$$\sigma^2_{\text{eff}}(\mathbf{h}, \mathbf{a}_m | \mathbf{a}_1, \ldots, \mathbf{a}_{m-1}) = \|\mathbf{C}_m^{\perp} \mathbf{F} \mathbf{a}_m\|^2$$

where $\mathbf{F} = \left(P^{-1}\mathbf{I} + \mathbf{h}\mathbf{h}^{\mathsf{T}}\right)^{-1/2}$ and $\mathbf{C}_m^{\perp}$ is the projection matrix for the nullspace of $\mathbf{F}[\mathbf{a}_1 \ \cdots \ \mathbf{a}_{m-1}]$.

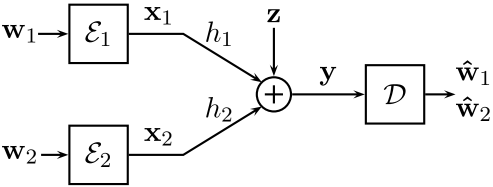
Theorem (Ordentlich-Erez-Nazer Allerton '13)

For any unimodular integer matrix $\mathbf{A} = [\mathbf{a}_1 \ \ldots \ \mathbf{a}_K]^{\mathsf{T}} \in \mathbb{Z}^{K \times K}$ *with descending successive effective noise variances, we have that*

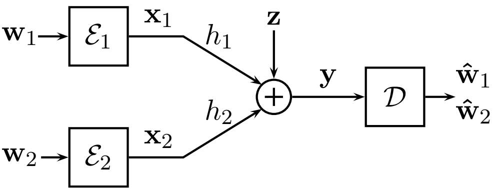
$$\sum_{m=1}^{K} \frac{1}{2} \log^+ \left(\frac{P}{\sigma^2_{\mathit{eff}}(\mathbf{h}, \mathbf{a}_m | \mathbf{a}_1, \ldots, \mathbf{a}_{m-1})} \right) = \frac{1}{2} \log \left(1 + \|\mathbf{h}\|^2 P\right) .$$

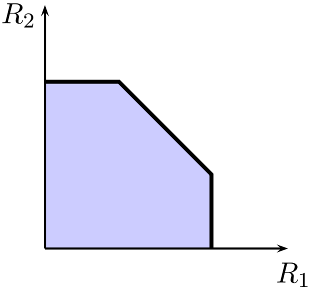
Moreover, there exists at least one permutation π *that associates each user's rate to a computation rate.*

Multiple-Access via Computation

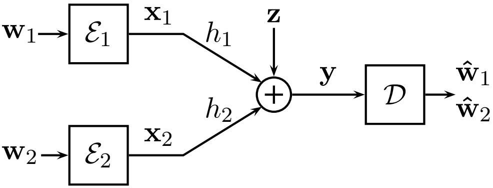


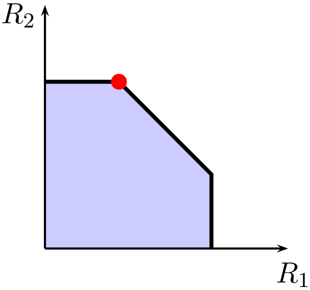
Multiple-Access via Computation





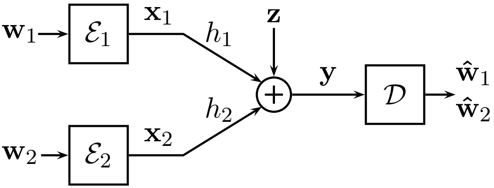
- Successive cancellation (without time-sharing or rate-splitting) achieves corner points.

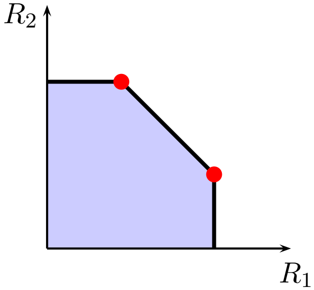




- Successive cancellation (without time-sharing or rate-splitting) achieves corner points.

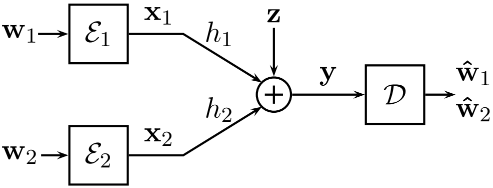
Multiple-Access via Computation

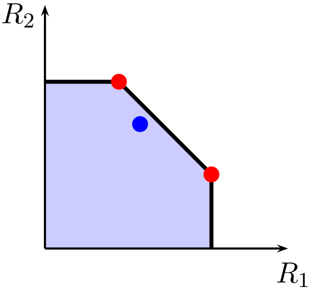




- Successive cancellation (without time-sharing or rate-splitting) achieves corner points.

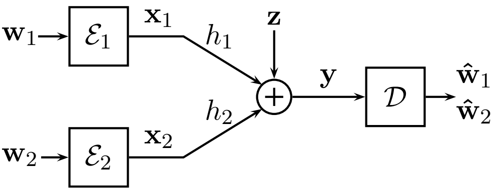
Multiple-Access via Computation

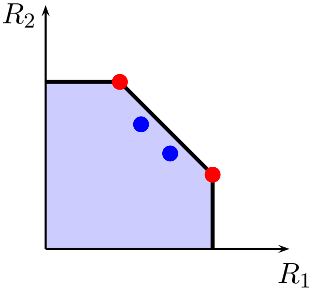




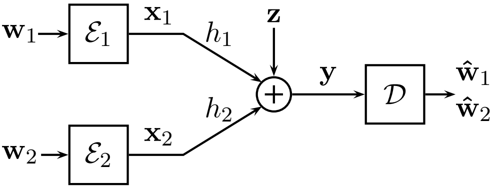
- Successive cancellation (without time-sharing or rate-splitting) achieves corner points.
- Compute-and-forward achieves another set of points near the sum rate boundary. Often closer to the symmetric capacity.

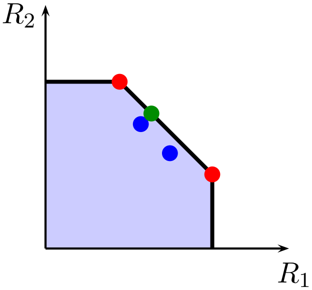
Multiple-Access via Computation



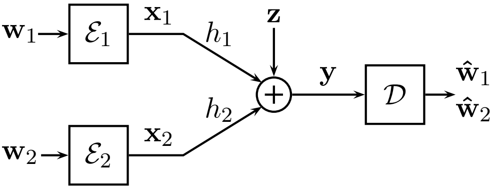


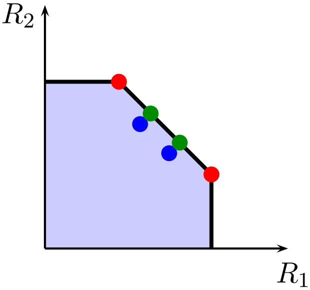
- Successive cancellation (without time-sharing or rate-splitting) achieves corner points.
- Compute-and-forward achieves another set of points near the sum rate boundary. Often closer to the symmetric capacity.





- Successive cancellation (without time-sharing or rate-splitting) achieves corner points.
- Compute-and-forward achieves another set of points near the sum rate boundary. Often closer to the symmetric capacity.
- Successive compute-and-forward can attain the exact sum rate.

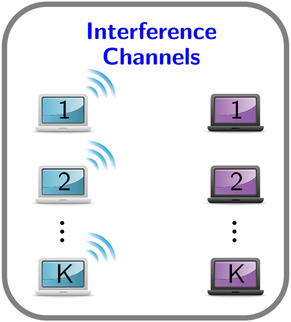




- Successive cancellation (without time-sharing or rate-splitting) achieves corner points.
- Compute-and-forward achieves another set of points near the sum rate boundary. Often closer to the symmetric capacity.
- Successive compute-and-forward can attain the exact sum rate.

Road Map

- Warm-up: Compute-and-forward over finite field channels.
- Compute-and-forward over Gaussian channels.
- Applications to communication across single-hop Gaussian networks.



Joint work with:

Symmetric case: Or Ordentlich and Uri Erez.

Stream-by-stream case: Vasilis Ntranos, Viveck Cadambe, and Giuseppe Caire.

Interference-Free Capacity

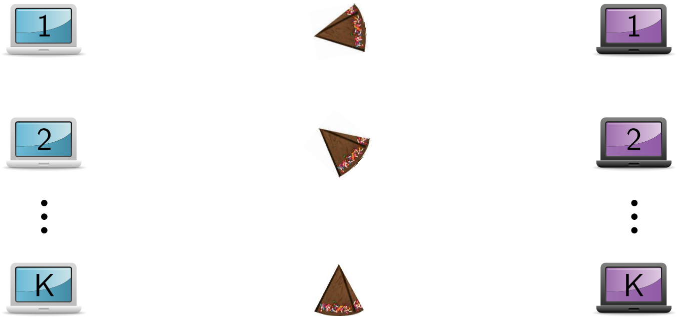
Interference-Free Capacity

Time Division

Time Division

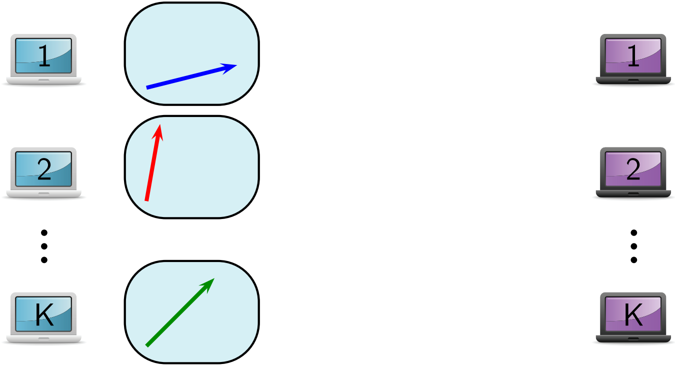
Time Division

Time Division



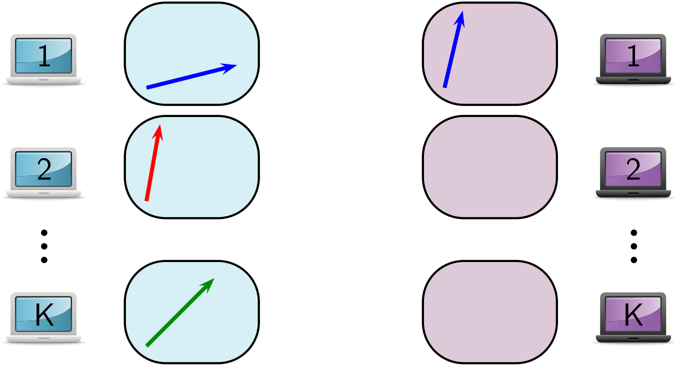
Interference Alignment

- **Cadambe-Jafar '08:** Alignment can achieve $K/2$ degrees-of-freedom for the K-user interference channel.
- **Birk-Kol '98:** Alignment for index coding. **Maddah-Ali - Motahari - Khandani '08:** Alignment for the MIMO X channel. See **Jafar '11** monograph (or recent e-book) for a richer history.

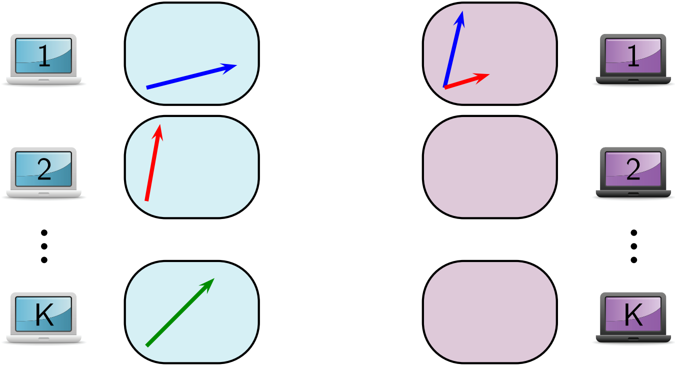


- **Cadambe-Jafar '08:** Alignment can achieve $K/2$ degrees-of-freedom for the K-user interference channel.
- **Birk-Kol '98:** Alignment for index coding. **Maddah-Ali - Motahari - Khandani '08:** Alignment for the MIMO X channel. See **Jafar '11** monograph (or recent e-book) for a richer history.

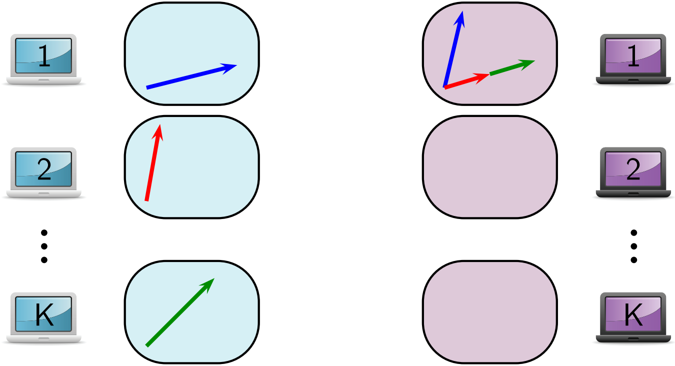
Interference Alignment



- **Cadambe-Jafar '08:** Alignment can achieve $K/2$ degrees-of-freedom for the K-user interference channel.
- **Birk-Kol '98:** Alignment for index coding. **Maddah-Ali - Motahari - Khandani '08:** Alignment for the MIMO X channel. See **Jafar '11** monograph (or recent e-book) for a richer history.

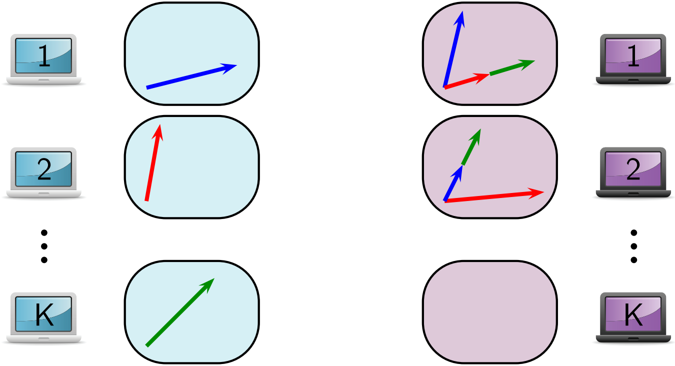


- **Cadambe-Jafar '08:** Alignment can achieve $K/2$ degrees-of-freedom for the K-user interference channel.
- **Birk-Kol '98:** Alignment for index coding. **Maddah-Ali - Motahari - Khandani '08:** Alignment for the MIMO X channel. See **Jafar '11** monograph (or recent e-book) for a richer history.

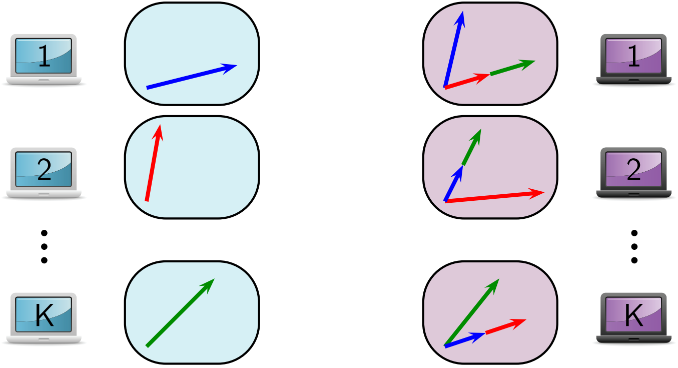


- **Cadambe-Jafar '08:** Alignment can achieve $K/2$ degrees-of-freedom for the K-user interference channel.
- **Birk-Kol '98:** Alignment for index coding. **Maddah-Ali - Motahari - Khandani '08:** Alignment for the MIMO X channel. See **Jafar '11** monograph (or recent e-book) for a richer history.

Interference Alignment



- **Cadambe-Jafar '08:** Alignment can achieve $K/2$ degrees-of-freedom for the K-user interference channel.
- **Birk-Kol '98:** Alignment for index coding. **Maddah-Ali - Motahari - Khandani '08:** Alignment for the MIMO X channel. See **Jafar '11** monograph (or recent e-book) for a richer history.



- **Cadambe-Jafar '08:** Alignment can achieve $K/2$ degrees-of-freedom for the K-user interference channel.
- **Birk-Kol '98:** Alignment for index coding. **Maddah-Ali - Motahari - Khandani '08:** Alignment for the MIMO X channel. See **Jafar '11** monograph (or recent e-book) for a richer history.

Interference Alignment

- **Cadambe-Jafar '08:** Alignment can achieve $K/2$ degrees-of-freedom for the K-user interference channel.
- **Birk-Kol '98:** Alignment for index coding. **Maddah-Ali - Motahari - Khandani '08:** Alignment for the MIMO X channel. See **Jafar '11** monograph (or recent e-book) for a richer history.

What do we need to get half the cake? (so far)

- Very large channel diversity (e.g., time extensions, frequency bands):

What do we need to get half the cake? (so far)

- Very large channel diversity (e.g., time extensions, frequency bands):
- Very high SNR:

What do we need to get half the cake? (so far)

- Very large channel diversity (e.g., time extensions, frequency bands):
  - **Cadambe, Jafar '08:** Asymptotic alignment. Achieves $\frac{K}{2}$ DoF across roughly 2^{K^2} channel realizations. Signal space alignment.
- Very high SNR:

What do we need to get half the cake? (so far)

- Very large channel diversity (e.g., time extensions, frequency bands):
  - **Cadambe, Jafar '08:** Asymptotic alignment. Achieves $\frac{K}{2}$ DoF across roughly 2^{K^2} channel realizations. Signal space alignment.
  - **Nazer, Gastpar, Jafar, Vishwanath '12:** Ergodic alignment. Achieves $\frac{K}{2}\log(1 + 2\mathsf{SNR})$ across roughly $(K\mathsf{SNR})^{K^2/2}$ channel realizations. Signal space alignment.
- Very high SNR:

What do we need to get half the cake? (so far)

- Very large channel diversity (e.g., time extensions, frequency bands):
  - **Cadambe, Jafar '08:** Asymptotic alignment. Achieves $\frac{K}{2}$ DoF across roughly 2^{K^2} channel realizations. Signal space alignment.
  - **Nazer, Gastpar, Jafar, Vishwanath '12:** Ergodic alignment. Achieves $\frac{K}{2}\log(1+2\mathsf{SNR})$ across roughly $(K\mathsf{SNR})^{K^2/2}$ channel realizations. Signal space alignment.

- Very high SNR:
  - **Motahari, Gharan, Maddah-Ali, Khandani '09:** Real alignment. Achieves $\frac{K}{2}$ DoF over one channel realization using roughly 2^{K^2} codeword layers. Signal scale alignment.

Basic Coding Framework:

- Each transmitter has one or more data streams, each of which is drawn from an i.i.d. random codebook.
- Data streams are sent using beamforming vectors, which are selected to align interference at the receivers.
- Each receiver nulls out the interfering data streams (e.g., zero-forcing) and decodes its desired data streams.

Basic Coding Framework:

- Each transmitter has one or more data streams, each of which is drawn from an i.i.d. random codebook.
- Data streams are sent using beamforming vectors, which are selected to align interference at the receivers.
- Each receiver nulls out the interfering data streams (e.g., zero-forcing) and decodes its desired data streams.

Advantages:

- Powerful optimization algorithms for power allocation and beamforming vectors.
- Some robustness to imperfect channel state information.

Basic Coding Framework:

- Each transmitter has one or more data streams, each of which is drawn from an i.i.d. random codebook.
- Data streams are sent using beamforming vectors, which are selected to align interference at the receivers.
- Each receiver nulls out the interfering data streams (e.g., zero-forcing) and decodes its desired data streams.

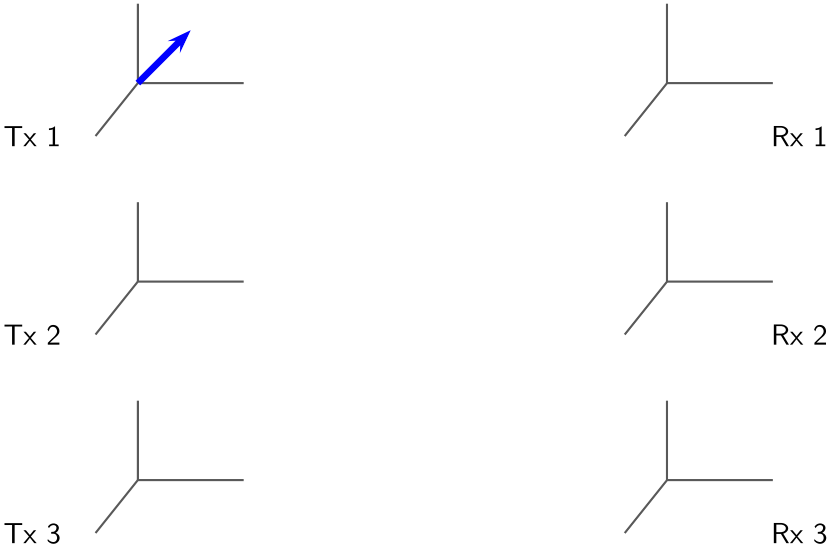
Advantages:

- Powerful optimization algorithms for power allocation and beamforming vectors.
- Some robustness to imperfect channel state information.

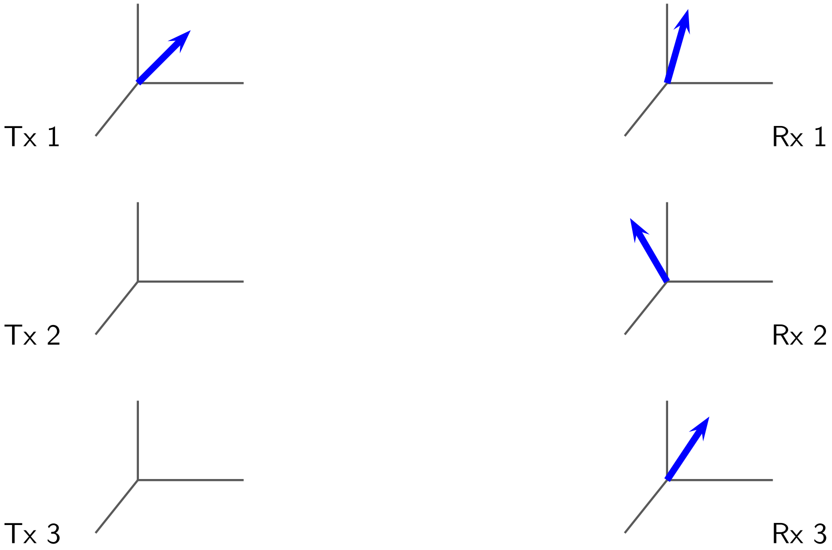
Disadvantages:

- May require enormous channel diversity.
- May require high SNR.

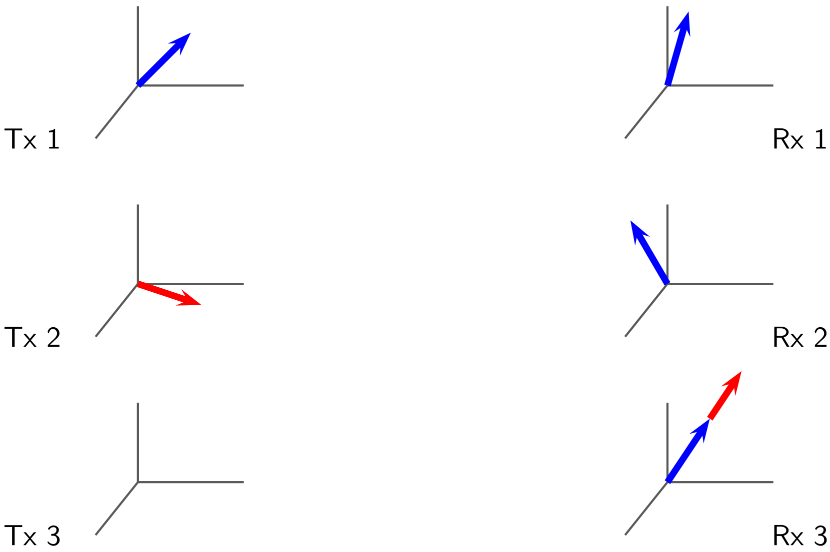
Example: Cadambe-Jafar '08 over 3 Channel Realizations



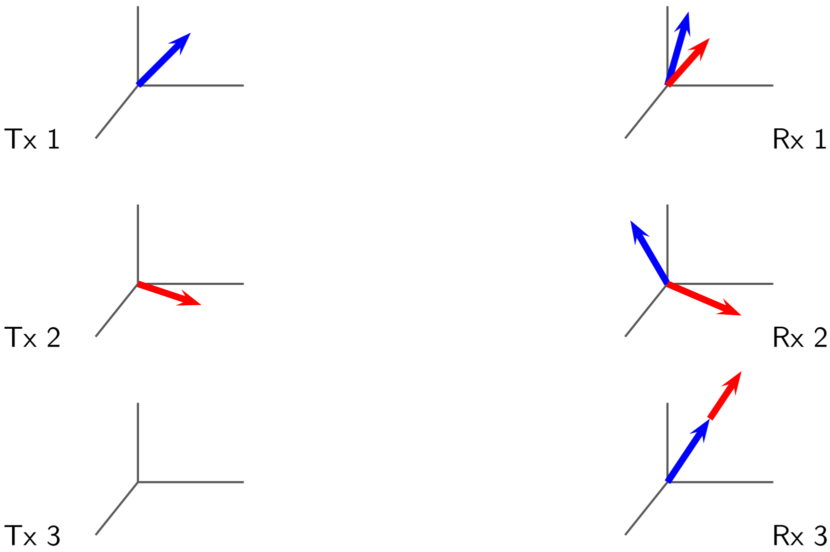
Example: Cadambe-Jafar '08 over 3 Channel Realizations



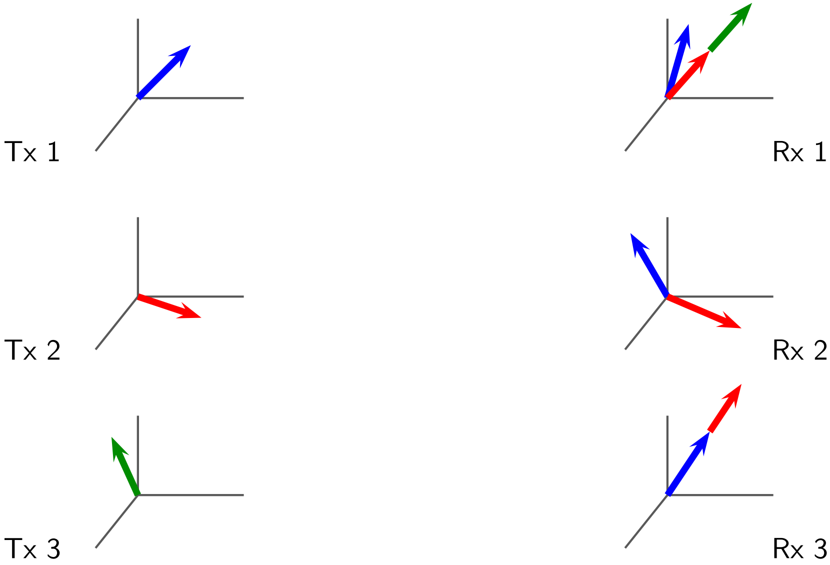
Example: Cadambe-Jafar '08 over 3 Channel Realizations



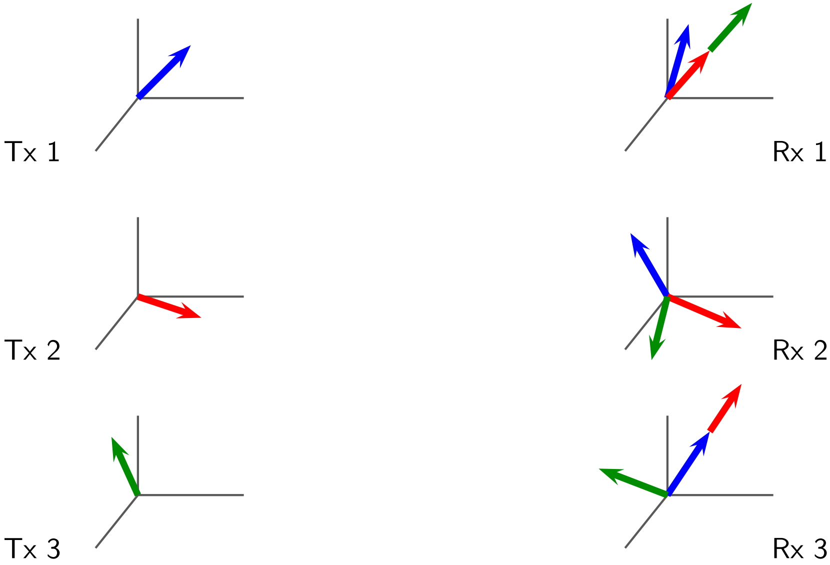
Example: Cadambe-Jafar '08 over 3 Channel Realizations



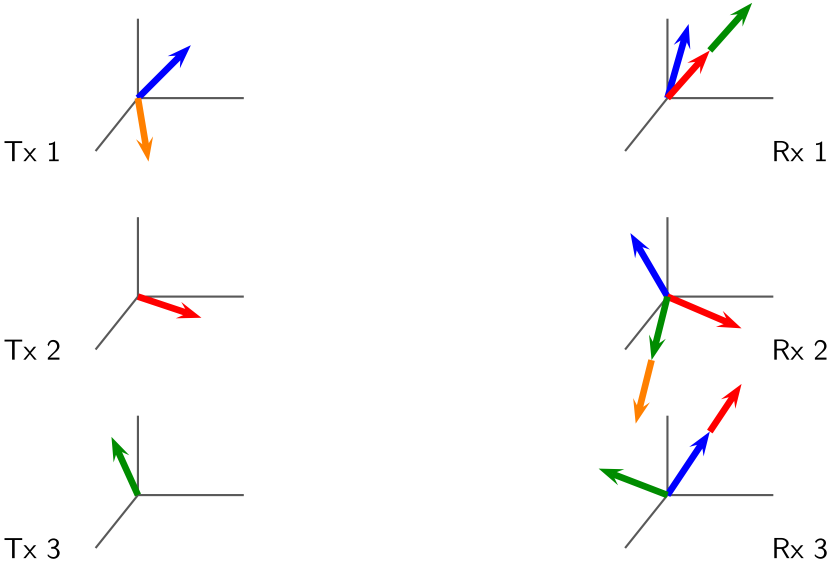
Example: Cadambe-Jafar '08 over 3 Channel Realizations



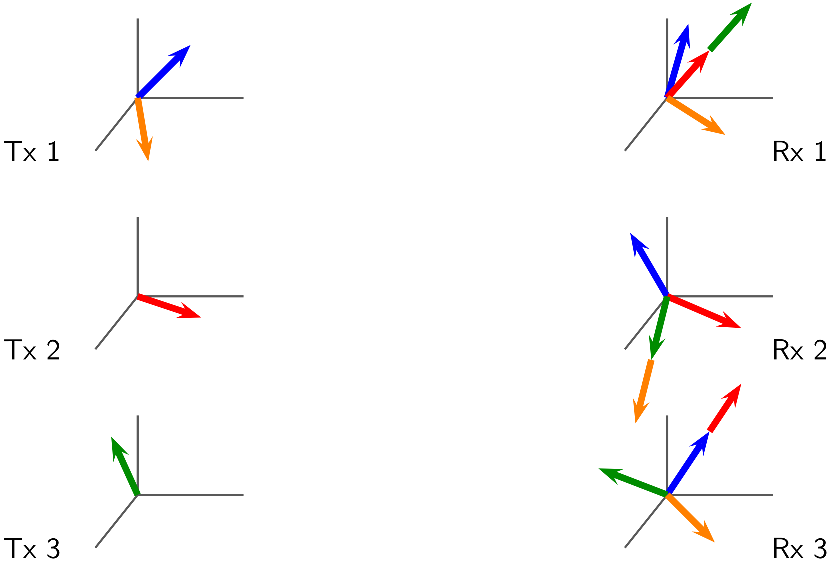
Example: Cadambe-Jafar '08 over 3 Channel Realizations



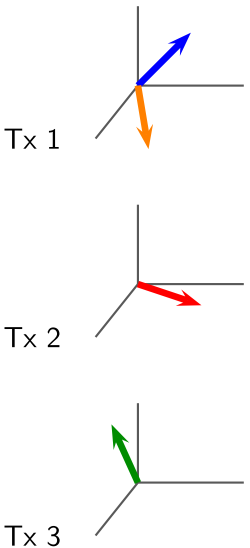
Example: Cadambe-Jafar '08 over 3 Channel Realizations



Example: Cadambe-Jafar '08 over 3 Channel Realizations

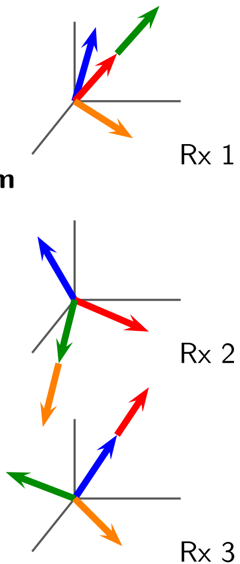


Example: Cadambe-Jafar '08 over 3 Channel Realizations



Total Degrees of Freedom

$$\text{DoF} = \frac{4 \text{ vectors}}{3 \text{ channel uses}} = \frac{4}{3}$$



Basic Coding Framework:

- Each transmitter has one or more codewords, each of which is drawn from a lattice codebook.
- Transmitter sends a linear combination of the codewords, with coefficients carefully chosen to align interference at the receivers.
- Each receiver must discern its desired lattice codewords from the sums of interfering ones.

Basic Coding Framework:

- Each transmitter has one or more codewords, each of which is drawn from a lattice codebook.
- Transmitter sends a linear combination of the codewords, with coefficients carefully chosen to align interference at the receivers.
- Each receiver must discern its desired lattice codewords from the sums of interfering ones.

Advantages:

- Only requires one channel realization.

Basic Coding Framework:

- Each transmitter has one or more codewords, each of which is drawn from a lattice codebook.
- Transmitter sends a linear combination of the codewords, with coefficients carefully chosen to align interference at the receivers.
- Each receiver must discern its desired lattice codewords from the sums of interfering ones.

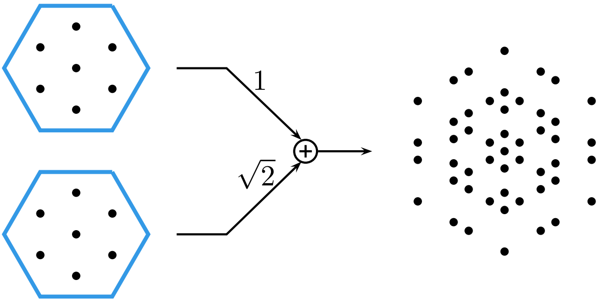
Advantages:

- Only requires one channel realization.

Disadvantages:

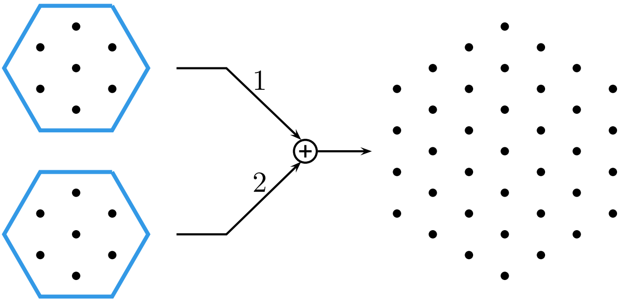
- Seems extremely sensitive to channel gains. DoF changes based on rationality/irrationality.
- Seems to require extremely high SNR.

Example: Two-User Lattice Alignment



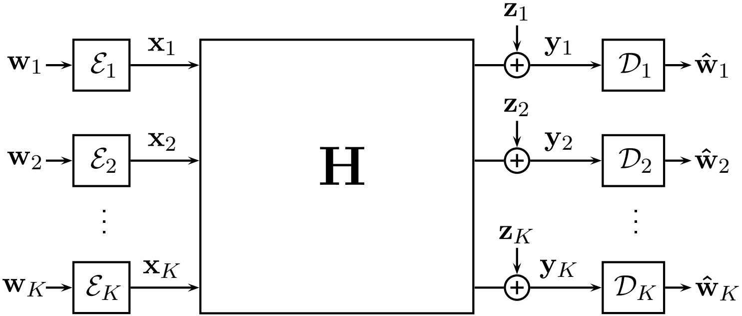
- Two lattice codewords can be recovered from their linear combination if the ratio of the coefficients is irrational.

Example: Two-User Lattice Alignment



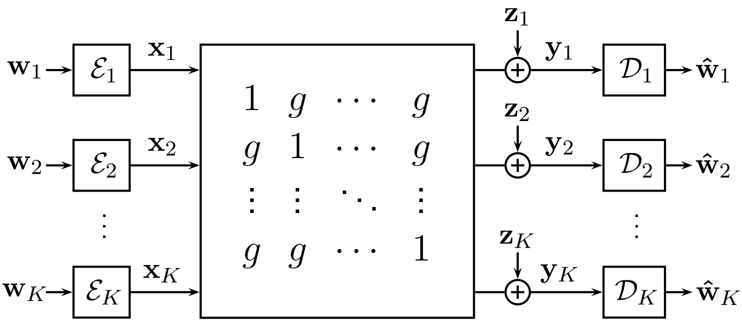
- Two lattice codewords can be recovered from their linear combination if the ratio of the coefficients is irrational.
- If the ratio is rational, it is not always possible to uniquely identify the pair of codewords.

Symmetric K-User Gaussian Interference Channel

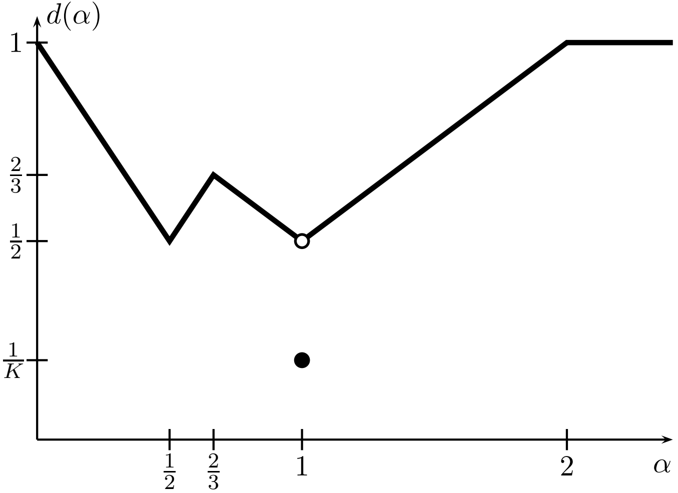


- Signal space alignment (e.g., beamforming) is infeasible.
- Signal scale alignment attains $K/2$ degrees-of-freedom for almost all channel gains, **Motahari et al. '09, Wu-Shamai-Verdu '11.**
- At finite SNR, the approximate capacity known in some special cases: two-user **Etkin-Tse-Wang '08**, many-to-one and one-to-many **Bresler-Parekh-Tse '10**, cyclic **Zhou-Yu '13**.

Symmetric K-User Gaussian Interference Channel



- Signal space alignment (e.g., beamforming) is infeasible.
- Signal scale alignment attains $K/2$ degrees-of-freedom for almost all channel gains, **Motahari et al. '09, Wu-Shamai-Verdu '11.**
- At finite SNR, the approximate capacity known in some special cases: two-user **Etkin-Tse-Wang '08**, many-to-one and one-to-many **Bresler-Parekh-Tse '10**, cyclic **Zhou-Yu '13**.
- Let's look at the symmetric case.



- Capacity understood in the high SNR regime. **Jafar-Vishwanath '10.**

$$\alpha = \frac{\log g^2 \mathsf{SNR}}{\log \mathsf{SNR}} \qquad d(\alpha) = \lim_{\mathsf{SNR}\to\infty} \frac{R(\mathsf{SNR})}{\frac{1}{2}\log \mathsf{SNR}}$$

Effective Multiple-Access Channel

- Each receiver sees an effective two-user multiple-access channel,

$$\mathbf{y}_k = \mathbf{x}_k + g \sum_{\ell \neq k} \mathbf{x}_\ell + \mathbf{z}_k \ .$$

Effective Multiple-Access Channel

- Each receiver sees an effective two-user multiple-access channel,

$$\mathbf{y}_k = \mathbf{x}_k + g \sum_{\ell \neq k} \mathbf{x}_\ell + \mathbf{z}_k \ .$$

Successive Cancellation Decoding:

- Decode and subtract interference $\sum_{\ell \neq k} \mathbf{x}_\ell$, then decode desired message.
- Only optimal when the interference is very strong, **Sridharan et al. '08**.

Effective Multiple-Access Channel

- Each receiver sees an effective two-user multiple-access channel,

$$\mathbf{y}_k = \mathbf{x}_k + g \sum_{\ell \neq k} \mathbf{x}_\ell + \mathbf{z}_k \ .$$

Successive Cancellation Decoding:

- Decode and subtract interference $\sum_{\ell \neq k} \mathbf{x}_\ell$, then decode desired message.
- Only optimal when the interference is very strong, **Sridharan et al. '08**.

Joint Decoding:

- Direct analysis is hindered by dependencies between codeword pairs.
- Existing work only applies at very high SNR, **Ordentlich-Erez '13.**

Alignment via Two Equations

- **Ordentlich-Erez-Nazer '14:** Decode two linear combinations:

$$a_1 \mathbf{x}_k + a_2 \sum_{\ell \neq k} \mathbf{x}_\ell \qquad\qquad b_1 \mathbf{x}_k + b_2 \sum_{\ell \neq k} \mathbf{x}_\ell$$

  using the compute-and-forward framework from **Nazer-Gastpar '11.** If the coefficients are linearly independent, we can solve for the desired message.

Alignment via Two Equations

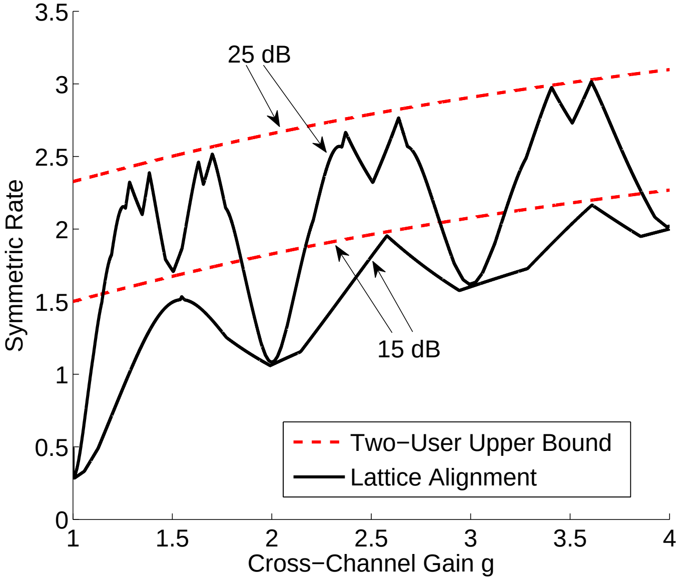
- **Ordentlich-Erez-Nazer '14:** Decode two linear combinations:

$$a_1\mathbf{x}_k + a_2\sum_{\ell \neq k}\mathbf{x}_\ell \qquad b_1\mathbf{x}_k + b_2\sum_{\ell \neq k}\mathbf{x}_\ell$$

  using the compute-and-forward framework from **Nazer-Gastpar '11.** If the coefficients are linearly independent, we can solve for the desired message.

- Set of "bad rationals" depends on the SNR. Only rationals with denominator $\sqrt{\mathsf{SNR}}$ or smaller cause issues.

Symmetric K-User Gaussian Interference Channel



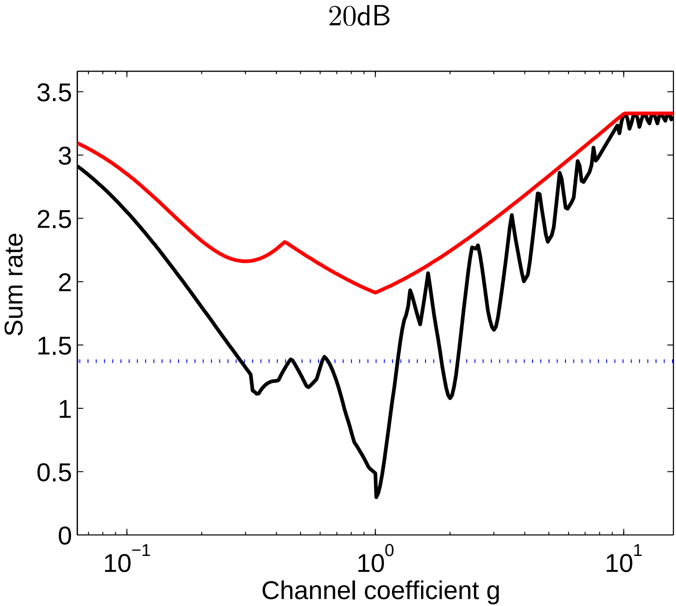
Alignment via Two Equations

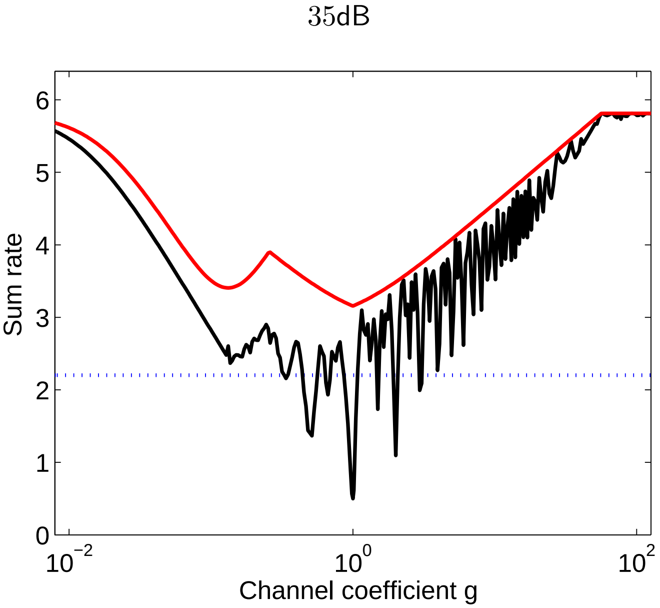
- Each receiver sees an effective two-user multiple-access channel,

$$\mathbf{y}_k = \mathbf{x}_k + g \sum_{\ell \neq k} \mathbf{x}_\ell + \mathbf{z}_k \ .$$

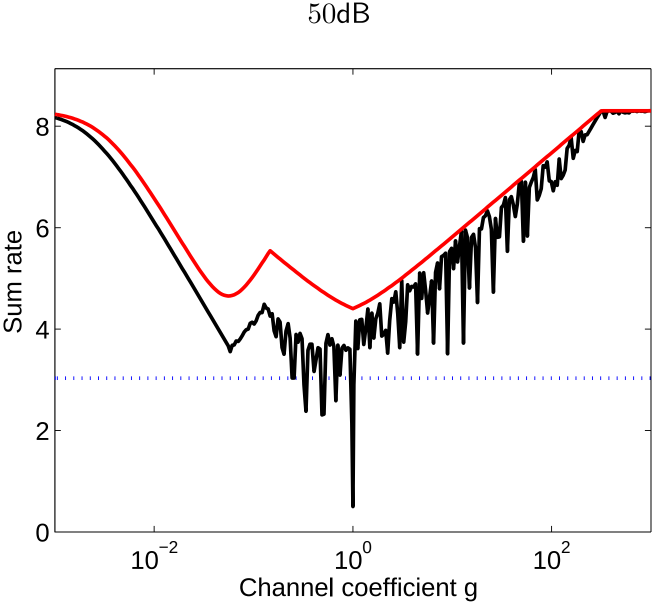
- **Ordentlich-Erez-Nazer '14:**
  - Noisy Regime: Decode one linear combination.
  - Moderately Weak and Weak Regimes: Send public and private lattice codewords. Decode three linear combinations.
  - Strong and Very Strong Regime: Decode two linear combinations.

Symmetric K-User Gaussian Interference Channel

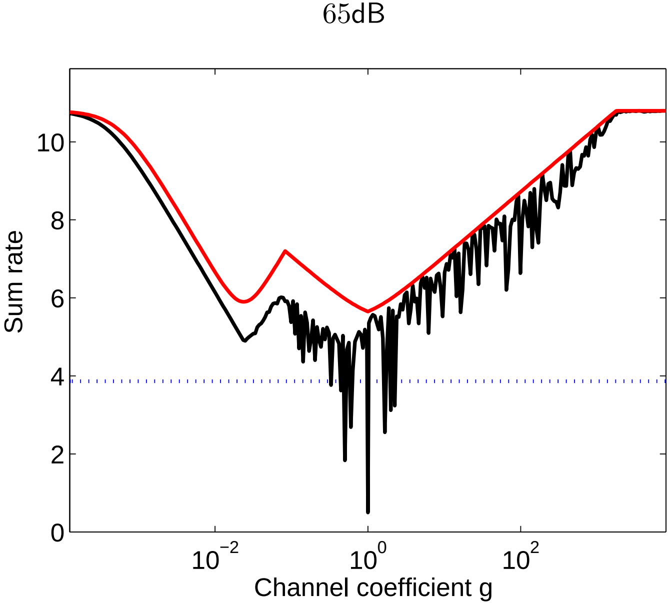



35dB
Sum rate
Channel coefficient g
0
1
2
3
4
5
6
10^{-2}
10^{0}
10^{2}

Symmetric K-User Gaussian Interference Channel



Symmetric K-User Gaussian Interference Channel



Approximate Capacity Results: Strong Regime

- Using the fact that the sum of the computation rates is nearly equal to the multiple-access sum capacity, we can approximate the sum capacity of the symmetric K-user Gaussian interference channel in all regimes.

$$R_{\mathsf{sym}} > \frac{1}{2}\log\left(1 + (1+2g^2)\mathsf{SNR}\right) - \max_{\mathbf{a}\in\mathbb{Z}^2} R_{\mathsf{comp}}\left([1 \ g]^{\mathsf{T}}, \mathbf{a}\right) - 1$$

- Via basic results from Diophantine approximation, we can approximate the sum capacity up to an outage set.

Approximate Capacity Results: Strong Regime

- Using the fact that the sum of the computation rates is nearly equal to the multiple-access sum capacity, we can approximate the sum capacity of the symmetric K-user Gaussian interference channel in all regimes.

$$R_{\mathsf{sym}} > \frac{1}{2}\log\left(1 + (1+2g^2)\mathsf{SNR}\right) - \max_{\mathbf{a}\in\mathbb{Z}^2} R_{\mathsf{comp}}\left([1\ g]^{\mathsf{T}}, \mathbf{a}\right) - 1$$

- Via basic results from Diophantine approximation, we can approximate the sum capacity up to an outage set.

- **Sample Result:** In the strong interference regime,

$$\frac{1}{4}\log^{+}(g^2\mathsf{SNR}) - \frac{c}{2} - 3 \le C_{\mathsf{sym}} \le \frac{1}{4}\log^{+}(g^2\mathsf{SNR}) + 1$$

  for all channel gains except for an outage set whose measure is a fraction of 2^{-c} of the interval $1 < |g| < \sqrt{\mathsf{SNR}}$, for any $c > 0$.

Integer-Forcing Interference Alignment

- Ideally, we should combine signal scale (e.g., lattice codes) and signal space alignment (e.g., beamforming vectors).

- Ideally, we should combine signal scale (e.g., lattice codes) and signal space alignment (e.g., beamforming vectors).

- **Ntranos-Cadambe-Nazer-Caire '13:** Proposed a new framework, integer-forcing interference alignment, that can simultaneously exploit signal space and signal scale alignment.

- Ideally, we should combine signal scale (e.g., lattice codes) and signal space alignment (e.g., beamforming vectors).

- **Ntranos-Cadambe-Nazer-Caire '13:** Proposed a new framework, integer-forcing interference alignment, that can simultaneously exploit signal space and signal scale alignment.

- Aimed at scenarios with finite channel diversity (e.g., a few independent fading realizations) and finite SNR.

- Ideally, we should combine signal scale (e.g., lattice codes) and signal space alignment (e.g., beamforming vectors).

- **Ntranos-Cadambe-Nazer-Caire '13:** Proposed a new framework, integer-forcing interference alignment, that can simultaneously exploit signal space and signal scale alignment.

- Aimed at scenarios with finite channel diversity (e.g., a few independent fading realizations) and finite SNR.

- Yields a new achievable rate region for any scenario which employs "stream-by-stream" alignment.

Problem Setting:

- Multiple data streams (i.e., codewords) $\mathbf{s}^{[\ell]} \in \mathbb{C}^T$, each assigned to its own beamforming vector $\mathbf{v}^{[\ell]} \in \mathbb{C}^M$.

Problem Setting:

- Multiple data streams (i.e., codewords) $\mathbf{s}^{[\ell]} \in \mathbb{C}^T$, each assigned to its own beamforming vector $\mathbf{v}^{[\ell]} \in \mathbb{C}^M$.

- Each receiver sees a noisy linear combination of its desired and interfering streams:

$$\mathbf{Y} = \sum_{\ell=1}^{L} \mathbf{H}_{\mathrm{D}}^{[\ell]} \mathbf{v}_{\mathrm{D}}^{[\ell]} \big(\mathbf{s}_{\mathrm{D}}^{[\ell]}\big)^{\mathsf{T}} + \sum_{j=1}^{J} \sum_{\ell=1}^{L} \mathbf{H}_{\mathrm{I}}^{[j,\ell]} \mathbf{v}_{\mathrm{I}}^{[j,\ell]} \big(\mathbf{s}_{\mathrm{I}}^{[j,\ell]}\big)^{\mathsf{T}} + \mathbf{Z} \ .$$

Problem Setting:

- Multiple data streams (i.e., codewords) $\mathbf{s}^{[\ell]} \in \mathbb{C}^T$, each assigned to its own beamforming vector $\mathbf{v}^{[\ell]} \in \mathbb{C}^M$.

- Each receiver sees a noisy linear combination of its desired and interfering streams:

$$\mathbf{Y} = \sum_{\ell=1}^{L} \mathbf{H}_{\mathrm{D}}^{[\ell]} \mathbf{v}_{\mathrm{D}}^{[\ell]} \big(\mathbf{s}_{\mathrm{D}}^{[\ell]}\big)^{\mathsf{T}} + \sum_{j=1}^{J} \sum_{\ell=1}^{L} \mathbf{H}_{\mathrm{I}}^{[j,\ell]} \mathbf{v}_{\mathrm{I}}^{[j,\ell]} \big(\mathbf{s}_{\mathrm{I}}^{[j,\ell]}\big)^{\mathsf{T}} + \mathbf{Z} \ .$$

- Signal space alignment occurs if, within each group j, all interferers have the same effective channel:

$$\mathbf{H}_{\mathrm{I}}^{[j,\ell]} \mathbf{v}_{\mathrm{I}}^{[j,\ell]} = \mathbf{H}_{\mathrm{I}}^{[j,1]} \mathbf{v}_{\mathrm{I}}^{[j,1]} \qquad \ell = 2, \ldots, L \ .$$

Problem Setting:

- Multiple data streams (i.e., codewords) $\mathbf{s}^{[\ell]} \in \mathbb{C}^T$, each assigned to its own beamforming vector $\mathbf{v}^{[\ell]} \in \mathbb{C}^M$.

- Each receiver sees a noisy linear combination of its desired and interfering streams:

$$\mathbf{Y} = \sum_{\ell=1}^{L} \mathbf{H}_{\mathrm{D}}^{[\ell]} \mathbf{v}_{\mathrm{D}}^{[\ell]} \big(\mathbf{s}_{\mathrm{D}}^{[\ell]}\big)^{\mathsf{T}} + \sum_{j=1}^{J} \mathbf{H}_{\mathrm{I}}^{[j,1]} \mathbf{v}_{\mathrm{I}}^{[j,1]} \sum_{\ell=1}^{L} \big(\mathbf{s}_{\mathrm{I}}^{[j,\ell]}\big)^{\mathsf{T}} + \mathbf{Z} \ .$$

- Signal space alignment occurs if, within each group j, all interferers have the same effective channel:

$$\mathbf{H}_{\mathrm{I}}^{[j,\ell]} \mathbf{v}_{\mathrm{I}}^{[j,\ell]} = \mathbf{H}_{\mathrm{I}}^{[j,1]} \mathbf{v}_{\mathrm{I}}^{[j,1]} \qquad \ell = 2, \ldots, L \ .$$

Problem Setting:

- Multiple data streams (i.e., codewords) $\mathbf{s}^{[\ell]} \in \mathbb{C}^T$, each assigned to its own beamforming vector $\mathbf{v}^{[\ell]} \in \mathbb{C}^M$.

- Each receiver sees a noisy linear combination of its desired and interfering streams:

$$\mathbf{Y} = \sum_{\ell=1}^{L} \mathbf{H}_{\mathrm{D}}^{[\ell]} \mathbf{v}_{\mathrm{D}}^{[\ell]} \big(\mathbf{s}_{\mathrm{D}}^{[\ell]}\big)^{\mathsf{T}} + \sum_{j=1}^{J} \mathbf{H}_{\mathrm{I}}^{[j,1]} \mathbf{v}_{\mathrm{I}}^{[j,1]} \sum_{\ell=1}^{L} \big(\mathbf{s}_{\mathrm{I}}^{[j,\ell]}\big)^{\mathsf{T}} + \mathbf{Z} \ .$$

- Signal space alignment occurs if, within each group j, all interferers have the same effective channel:

$$\mathbf{H}_{\mathrm{I}}^{[j,\ell]} \mathbf{v}_{\mathrm{I}}^{[j,\ell]} = \mathbf{H}_{\mathrm{I}}^{[j,1]} \mathbf{v}_{\mathrm{I}}^{[j,1]} \qquad \ell = 2, \ldots, L \ .$$

- If a group is aligned in signal space, we can also induce signal scale alignment using a generalization of compute-and-forward that permits unequal powers.

Problem Setting:

- Multiple data streams (i.e., codewords) $\mathbf{s}^{[\ell]} \in \mathbb{C}^T$, each assigned to its own beamforming vector $\mathbf{v}^{[\ell]} \in \mathbb{C}^M$.

- Each receiver sees a noisy linear combination of its desired and interfering streams:

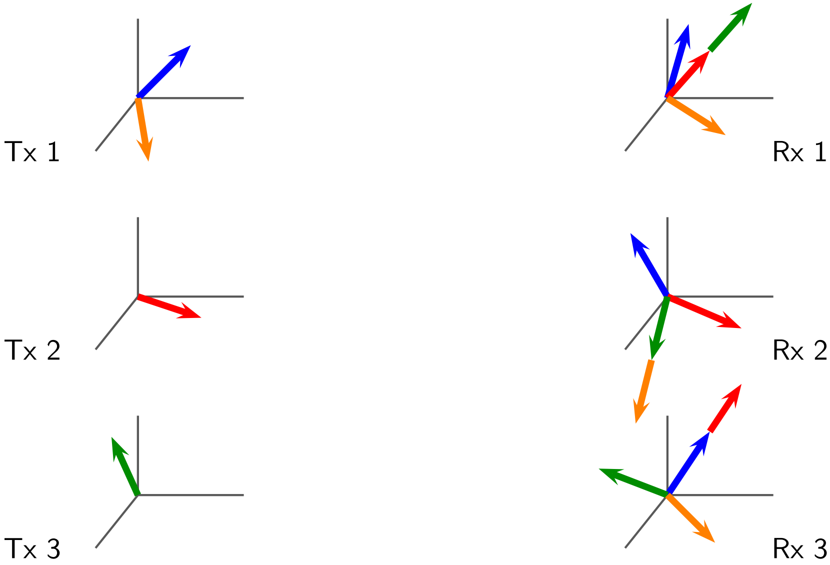
$$\mathbf{Y} = \sum_{\ell=1}^{L} \mathbf{H}_{\mathrm{D}}^{[\ell]} \mathbf{v}_{\mathrm{D}}^{[\ell]} \left(\mathbf{s}_{\mathrm{D}}^{[\ell]}\right)^{\mathsf{T}} + \sum_{j=1}^{J} \mathbf{H}_{\mathrm{I}}^{[j,1]} \mathbf{v}_{\mathrm{I}}^{[j,1]} \sum_{\ell=1}^{L} \left(\mathbf{s}_{\mathrm{I}}^{[j,\ell]}\right)^{\mathsf{T}} + \mathbf{Z} \ .$$

- Signal space alignment occurs if, within each group j, all interferers have the same effective channel:

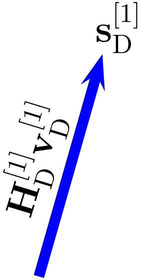
$$\mathbf{H}_{\mathrm{I}}^{[j,\ell]} \mathbf{v}_{\mathrm{I}}^{[j,\ell]} = \mathbf{H}_{\mathrm{I}}^{[j,1]} \mathbf{v}_{\mathrm{I}}^{[j,1]} \qquad \ell = 2, \ldots, L \ .$$

- If a group is aligned in signal space, we can also induce signal scale alignment using a generalization of compute-and-forward that permits unequal powers.

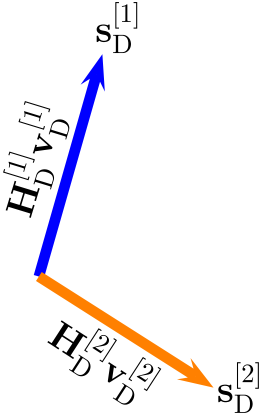
Example: Cadambe-Jafar '08 over 3 Channel Realizations



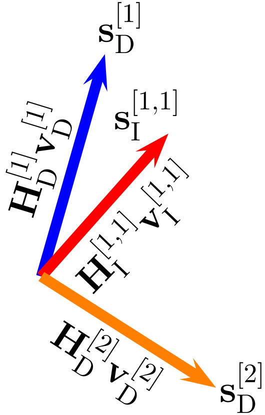
Received Signal



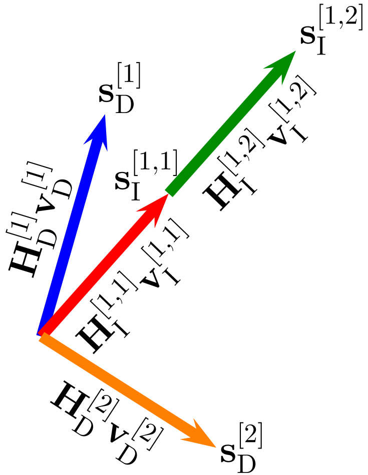
Received Signal



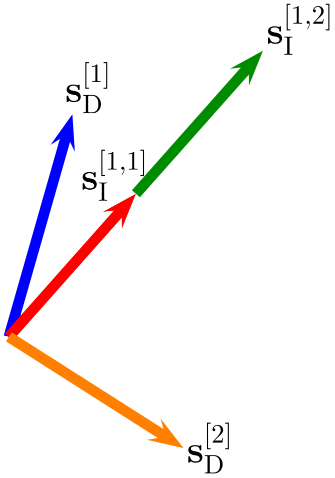
Received Signal



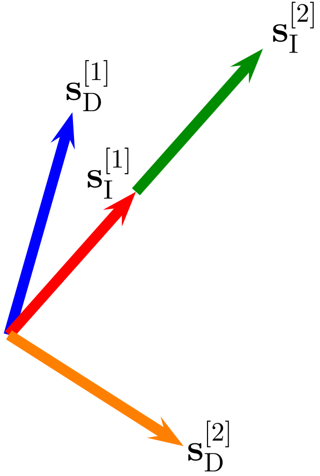
Received Signal

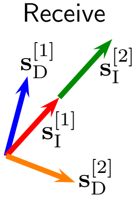


Received Signal



Received Signal

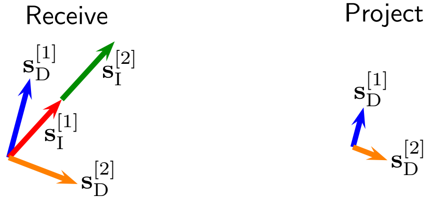




How should each receiver decoder its desired data streams?

Zero-Forcing Interference Alignment:

- Generate the data streams using i.i.d. random coding.

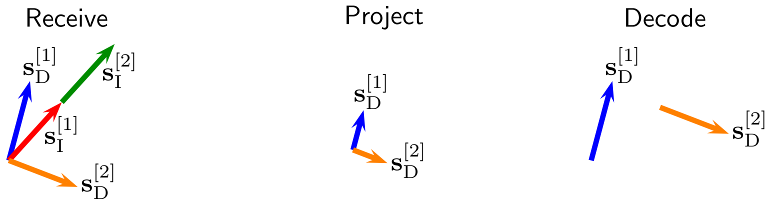


How should each receiver decoder its desired data streams?

Zero-Forcing Interference Alignment:

- Generate the data streams using i.i.d. random coding.
- First, project the received signal into the nullspace of the interference.

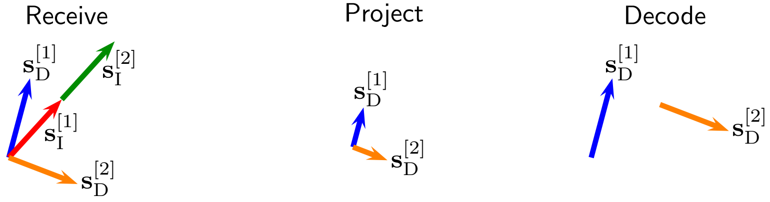
Zero-Forcing Decoding



How should each receiver decoder its desired data streams?

Zero-Forcing Interference Alignment:

- Generate the data streams using i.i.d. random coding.
- First, project the received signal into the nullspace of the interference.
- Then, jointly decode desired data streams.

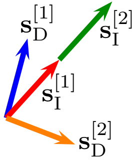


How should each receiver decoder its desired data streams?

Zero-Forcing Interference Alignment:

- Generate the data streams using i.i.d. random coding.
- First, project the received signal into the nullspace of the interference.
- Then, jointly decode desired data streams.
- Suffices from a degrees-of-freedom perspective.

Receive

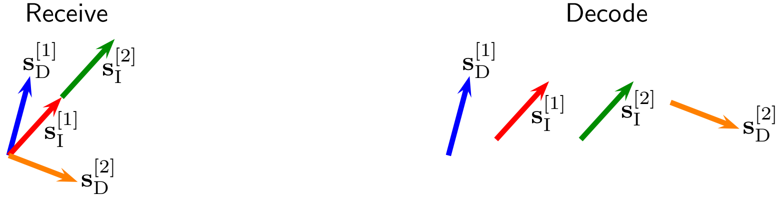


How should each receiver decoder its desired data streams?

Joint Typicality Decoding:

- Generate the data streams using i.i.d. random coding.

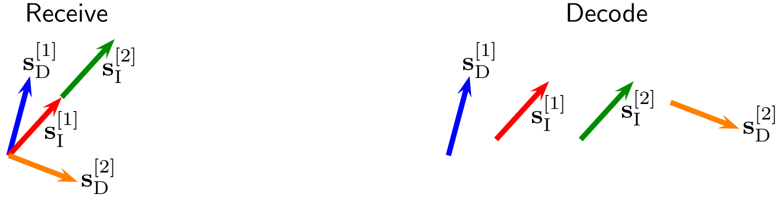
Joint Decoding (with i.i.d Random Codes)



How should each receiver decoder its desired data streams?

Joint Typicality Decoding:

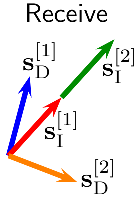
- Generate the data streams using i.i.d. random coding.
- If we attempt to decode the aligned interference, we will end up decoding each interferer separately.
- This significantly reduces the achievable rate per data stream.



How should each receiver decoder its desired data streams?

Joint Typicality Decoding:

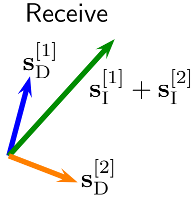
- Generate the data streams using i.i.d. random coding.
- If we attempt to decode the aligned interference, we will end up decoding each interferer separately.
- This significantly reduces the achievable rate per data stream.
- Analyzing lattice-coded data streams is beyond the reach of current techniques owing to dependencies.



How should each receiver decoder its desired data streams?

Integer-Forcing Interference Alignment:

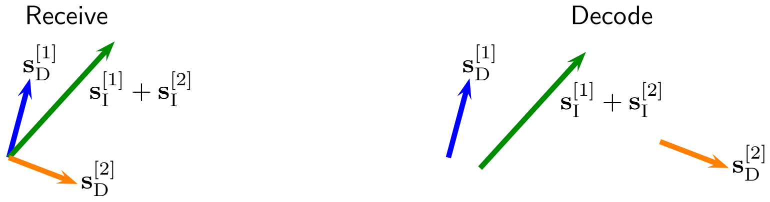
- Beamforming directions chosen to induce signal space alignment.



How should each receiver decoder its desired data streams?

Integer-Forcing Interference Alignment:

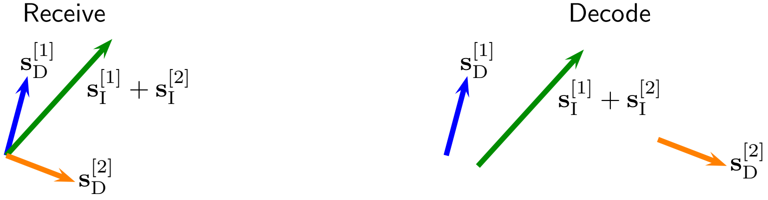
- Beamforming directions chosen to induce signal space alignment.
- Data streams are encoded using nested lattice codes according to some power allocation. This induces signal scale alignment.



How should each receiver decoder its desired data streams?

Integer-Forcing Interference Alignment:

- Beamforming directions chosen to induce signal space alignment.
- Data streams are encoded using nested lattice codes according to some power allocation. This induces signal scale alignment.
- Receiver decodes linear combinations and solves for its desired data streams.



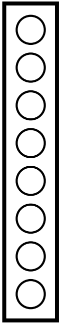
How should each receiver decoder its desired data streams?

Integer-Forcing Interference Alignment:

- Beamforming directions chosen to induce signal space alignment.
- Data streams are encoded using nested lattice codes according to some power allocation. This induces signal scale alignment.
- Receiver decodes linear combinations and solves for its desired data streams.
- Requires extension of compute-and-forward to unequal powers.

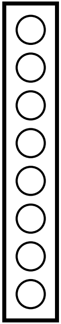
Asymmetric Compute-and-Forward: Transmitter View

- Each codeword is assigned an effective noise tolerance $\sigma^2_{\mathsf{eff},\ell}$ and power level P_ℓ. Rate is $\frac{1}{2}\log(P_\ell/\sigma^2_{\mathsf{eff},\ell})$.



Asymmetric Compute-and-Forward: Transmitter View

- Each codeword is assigned an effective noise tolerance $\sigma^2_{\text{eff},\ell}$ and power level P_ℓ. Rate is $\frac{1}{2}\log(P_\ell/\sigma^2_{\text{eff},\ell})$.
- Information symbols take values over the finite field $\mathbb{Z}_p$ (where p is prime). Collected into a vector $\mathbf{w}_{\text{info},\ell}$.

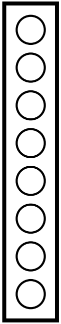


Asymmetric Compute-and-Forward: Transmitter View

- Each codeword is assigned an effective noise tolerance $\sigma^2_{\text{eff},\ell}$ and power level P_ℓ. Rate is $\frac{1}{2}\log(P_\ell/\sigma^2_{\text{eff},\ell})$.

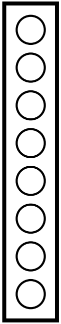
- Information symbols take values over the finite field $\mathbb{Z}_p$ (where p is prime). Collected into a vector $\mathbf{w}_{\text{info},\ell}$.

- **Lattice View:** The fine lattice is determined by the effective noise tolerance. The coarse lattice is determined by the power level.



Asymmetric Compute-and-Forward: Transmitter View

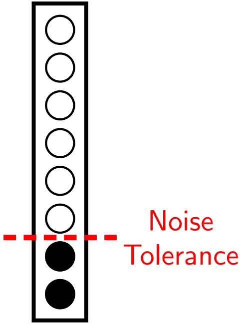
- Each codeword is assigned an effective noise tolerance $\sigma^2_{\text{eff},\ell}$ and power level P_ℓ. Rate is $\frac{1}{2}\log(P_\ell/\sigma^2_{\text{eff},\ell})$.
- Information symbols take values over the finite field $\mathbb{Z}_p$ (where p is prime). Collected into a vector $\mathbf{w}_{\text{info},\ell}$.
- **Lattice View:** The fine lattice is determined by the effective noise tolerance. The coarse lattice is determined by the power level.



- **Finite Field View:** Each transmitter has a vector $\mathbf{w}_\ell$ over $\mathbb{Z}_p^k$. Effective noise tolerance determines how many zeros to place at the bottom of the vector. Power level determines how many "don't care" entries to place at the top of the vector. Information symbols can fill the remaining spaces.

Asymmetric Compute-and-Forward: Transmitter View

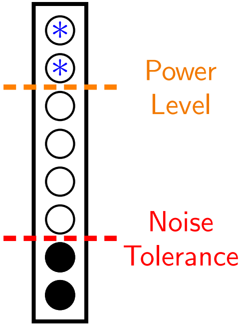
- Each codeword is assigned an effective noise tolerance $\sigma^2_{\text{eff},\ell}$ and power level P_ℓ. Rate is $\frac{1}{2}\log(P_\ell/\sigma^2_{\text{eff},\ell})$.
- Information symbols take values over the finite field $\mathbb{Z}_p$ (where p is prime). Collected into a vector $\mathbf{w}_{\text{info},\ell}$.
- **Lattice View:** The fine lattice is determined by the effective noise tolerance. The coarse lattice is determined by the power level.



- **Finite Field View:** Each transmitter has a vector $\mathbf{w}_\ell$ over $\mathbb{Z}_p^k$. Effective noise tolerance determines how many zeros to place at the bottom of the vector. Power level determines how many "don't care" entries to place at the top of the vector. Information symbols can fill the remaining spaces.

Asymmetric Compute-and-Forward: Transmitter View

- Each codeword is assigned an effective noise tolerance $\sigma^2_{\text{eff},\ell}$ and power level P_ℓ. Rate is $\frac{1}{2}\log(P_\ell/\sigma^2_{\text{eff},\ell})$.
- Information symbols take values over the finite field $\mathbb{Z}_p$ (where p is prime). Collected into a vector $\mathbf{w}_{\text{info},\ell}$.
- **Lattice View:** The fine lattice is determined by the effective noise tolerance. The coarse lattice is determined by the power level.



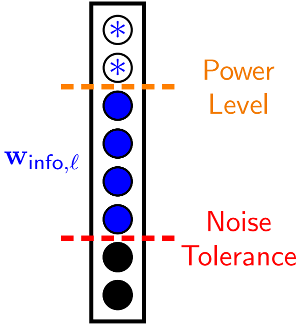
- **Finite Field View:** Each transmitter has a vector $\mathbf{w}_\ell$ over $\mathbb{Z}_p^k$. Effective noise tolerance determines how many zeros to place at the bottom of the vector. Power level determines how many "don't care" entries to place at the top of the vector. Information symbols can fill the remaining spaces.

Asymmetric Compute-and-Forward: Transmitter View

- Each codeword is assigned an effective noise tolerance $\sigma^2_{\text{eff},\ell}$ and power level P_ℓ. Rate is $\frac{1}{2}\log(P_\ell/\sigma^2_{\text{eff},\ell})$.

- Information symbols take values over the finite field $\mathbb{Z}_p$ (where p is prime). Collected into a vector $\mathbf{w}_{\text{info},\ell}$.

- **Lattice View:** The fine lattice is determined by the effective noise tolerance. The coarse lattice is determined by the power level.



- **Finite Field View:** Each transmitter has a vector $\mathbf{w}_\ell$ over $\mathbb{Z}_p^k$. Effective noise tolerance determines how many zeros to place at the bottom of the vector. Power level determines how many "don't care" entries to place at the top of the vector. Information symbols can fill the remaining spaces.

Asymmetric Compute-and-Forward: Receiver View

- Codewords observed through channel matrix $\mathbf{H}$.

Asymmetric Compute-and-Forward: Receiver View

- Codewords observed through channel matrix $\mathbf{H}$.
- **Lattice View:** Decode integer-linear combination of the codewords modulo the coarsest lattice:

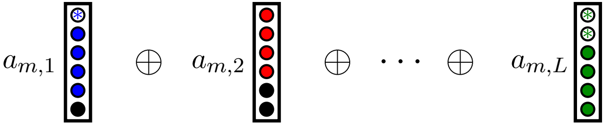
$$\left[\sum_{\ell} a_{m\ell} \, \mathbf{s}_{\ell} \right] \mod \Lambda_{\mathsf{c},1}$$

Asymmetric Compute-and-Forward: Receiver View

- Codewords observed through channel matrix $\mathbf{H}$.
- **Lattice View:** Decode integer-linear combination of the codewords modulo the coarsest lattice:

$$\left[\sum_{\ell} a_{m\ell}\,\mathbf{s}_{\ell}\right] \mod \Lambda_{\mathsf{c},1}$$

- **Finite Field View:** Decode linear combination over $\mathbb{Z}_p^k$:

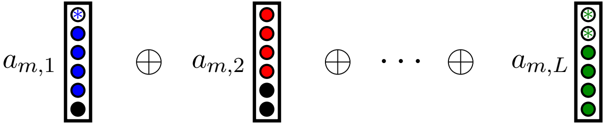


Asymmetric Compute-and-Forward: Receiver View

- Codewords observed through channel matrix $\mathbf{H}$.
- **Lattice View:** Decode integer-linear combination of the codewords modulo the coarsest lattice:

$$\Big[\sum_{\ell} a_{m\ell} \, \mathbf{s}_\ell \Big] \mod \Lambda_{\mathsf{c},1}$$

- **Finite Field View:** Decode linear combination over $\mathbb{Z}_p^k$:



- In both cases, the linear combination with coefficient vector $\mathbf{a}_m^{\mathsf{T}} = [a_{m1} \ a_{m2} \ \cdots \ a_{mL}]$ can be decoded reliably if

$$\sigma^2_{\mathsf{eff},\ell} > \mathbf{a}_m^{\mathsf{T}} \big(\mathbf{P}^{-1} + \mathbf{H}\mathbf{H}^{\mathsf{T}} \big)^{-1} \mathbf{a}_m$$

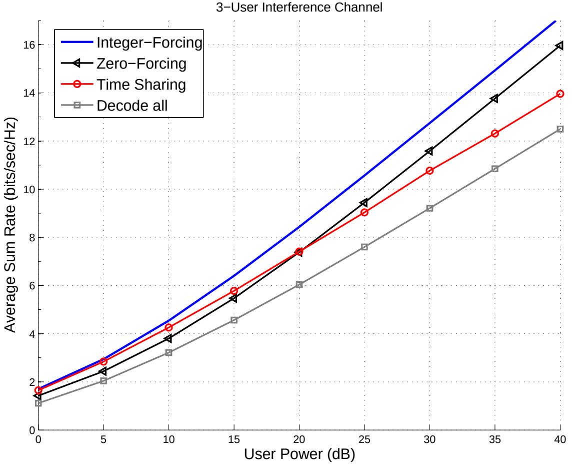
for all ℓ such that $a_{m\ell} \neq 0$.

Performance Comparison

- 3-user Gaussian interference channel.
- Can code over 3 independent fading realizations from an i.i.d. Rayleigh distribution.

Strategies:

- CJ '08 Beamforming $+$ Zero-Forcing Decoding.
- CJ '08 Beamforming $+$ Integer-Forcing Decoding.



- Currently, we take fixed beamforming directions, such as from **Cadambe-Jafar '08**, and optimize the power allocation and the integer coefficients.

Challenges

- Currently, we take fixed beamforming directions, such as from **Cadambe-Jafar '08**, and optimize the power allocation and the integer coefficients.

- Note that choosing the optimal integers is equivalent to finding the shortest basis in a certain lattice so we use LLL-type approximation algorithms.

- Currently, we take fixed beamforming directions, such as from **Cadambe-Jafar '08**, and optimize the power allocation and the integer coefficients.

- Note that choosing the optimal integers is equivalent to finding the shortest basis in a certain lattice so we use LLL-type approximation algorithms.

- Ideally, we would like to jointly optimize the beamforming directions, power allocation, and the integer coefficients. That is, we need Max-SINR type algorithms for integer-forcing interference alignment.

Challenges

- Currently, we take fixed beamforming directions, such as from **Cadambe-Jafar '08**, and optimize the power allocation and the integer coefficients.
- Note that choosing the optimal integers is equivalent to finding the shortest basis in a certain lattice so we use LLL-type approximation algorithms.
- Ideally, we would like to jointly optimize the beamforming directions, power allocation, and the integer coefficients. That is, we need Max-SINR type algorithms for integer-forcing interference alignment.
- Our current results only apply to stream-by-stream alignment, not subspace alignment. This will likely require more sophisticated lattice constructions.

Challenges

- Currently, we take fixed beamforming directions, such as from **Cadambe-Jafar '08**, and optimize the power allocation and the integer coefficients.

- Note that choosing the optimal integers is equivalent to finding the shortest basis in a certain lattice so we use LLL-type approximation algorithms.

- Ideally, we would like to jointly optimize the beamforming directions, power allocation, and the integer coefficients. That is, we need Max-SINR type algorithms for integer-forcing interference alignment.

- Our current results only apply to stream-by-stream alignment, not subspace alignment. This will likely require more sophisticated lattice constructions.

- and many more (such as joint decoding, non-unique decoding)...

Recent coding perspectives on compute-and-forward:

- **Feng-Silva-Kschischang '13:** General algebraic framework in terms of lattice partitions and R-modules.
- **Hern-Narayanan '13, Huang-Narayanan-Tunali '14:** Multilevel codes.
- **Ordentlich-Erez '12, Yang et al. '12:** Binary convolutional codes.
- **Hong and Caire '11, Ordentlich et al. '11:** Binary and p-ary LDPC codes.
- **Belfiore-Ling '12:** Code design criteria.
- **Tunali-Narayanan-Pfister '13:** Spatially-coupled LDPC codes.

Some topics we did not have a chance to cover:

- Distributed Source Coding: **Körner-Marton '79**, **Krithivasan-Pradhan '09,'11**, **Wagner '11**, **Tse-Maddah-Ali '10**
- Relaying: **Wilson-Narayanan-Pfister-Sprintson '10**, **Nam-Chung-Lee '10, '11**, **Goseling-Gastpar-Weber '11**, **Song-Devroye '13**, **Nokleby-Aazhang '12**
- Cellular Networks: **Sanderovich-Peleg-Shamai '11**, **Nazer-Sanderovich-Gastpar-Shamai '09**, **Hong-Caire '13**
- Distributed Dirty-Paper Coding: **Philosof-Zamir '09**, **Philosof-Zamir-Erez-Khisti '11**, **Wang '12**
- Joint Source-Channel Coding: **Kochman-Zamir '09**, **Nazer-Gastpar '07, '08**, **Soundararajan-Vishwanath '12**
- Physical-Layer Secrecy: **He-Yener '11, '14**, **Kashyap-Shashank-Thangaraj '12**

Concluding Remarks

- Even if you only want to recover messages, it can help to first decode linear combinations.
- Compute-and-forward creates a direct link between Gaussian interference networks and finite field ones.
- Enables more efficient encoding/decoding for networks where the capacity is already known.
- Yields new achievable rates for interference channels.
- Broader story: Algebraic Structure in Network Information Theory. ISIT '11 Tutorial. Survey on physical-layer network coding in Proceedings of the IEEE, March 2011.
- Upcoming textbook by Ram Zamir.

References - Intro

S. Avestimehr, S. Diggavi, and D. Tse, "Wireless network information flow: A deterministic approach," *IEEE Transactions on Information Theory*, vol. 57, no. 4, pp. 1872–1905, Apr. 2011.

B. Nazer and M. Gastpar, "Compute-and-forward: Harnessing interference through structured codes," *IEEE Transactions on Information Theory*, vol. 57, no. 10, pp. 6463–6486, Oct. 2011.

R. Gallager, *Information Theory and Reliable Communication.* New York: John Wiley and Sons, Inc., 1968.

J. Körner and K. Marton, "How to encode the modulo-two sum of binary sources," *IEEE Transactions on Information Theory*, vol. 25, no. 2, pp. 219–221, Mar. 1979.

B. Nazer and M. Gastpar, "Computation over multiple-access channels," *IEEE Transactions on Information Theory*, vol. 53, no. 10, pp. 3498–3516, Oct. 2007.

R. Ahlswede, N. Cai, S.-Y. R. Li, and R. W. Yeung, "Network information flow," *IEEE Transactions on Information Theory*, vol. 46, no. 4, pp. 1204–1216, Jul. 2000.

T. Cover and J. Thomas, *Elements of Information Theory*, 2nd ed. Hoboken, NJ: Wiley-Interscience, 2006.

A. El Gamal and Y.-H. Kim, *Network Information Theory*, Cambridge, UK: Cambridge University Press, 2011.

References – Physical-Layer Network Coding

- Y. Wu, P. A. Chou, and S.-Y. Kung, “Information exchange in wireless networks with network coding and physical-layer broadcast,” Microsoft Research, Redmond, WA, Tech. Rep. MSR-TR-2004-78, Aug. 2004.
- S. Zhang, S. Liew, and P. Lam, “Hot topic: Physical-layer network coding,” in *Proc. ACM Int. Conf. Mobile Comp. Netw.*, Los Angeles, CA, Sep. 2006.
- B. Nazer and M. Gastpar, “Computing over multiple-access channels with connections to wireless network coding,” in *Proc. IEEE Int. Symp. Inf. Theory*, Seattle, WA, Jul. 2006.
- P. Popovski and H. Yomo, “Bi-directional amplification of throughput in a wireless multi-hop network,” in *Proc. IEEE Veh. Tech. Conf.*, Melbourne, Australia, May 2006.
- S. Katti, S. Gollakota, and D. Katabi, “Embracing wireless interference: Analog network coding,” *ACM SIGCOMM*, Kyoto, Japan, Aug. 2007.
- M. P. Wilson, K. Narayanan, H. Pfister, and A. Sprintson, “Joint physical layer coding and network coding for bidirectional relaying,” *IEEE Transactions on Information Theory*, vol. 11, no. 56, pp. 5641–5654, Nov. 2010.
- W. Nam, S.-Y. Chung, and Y. H. Lee, “Capacity of the Gaussian two-way relay channel to within $1/2$ bit,” *IEEE Transactions on Information Theory*, vol. 56, no. 11, pp. 5488–5494, Nov. 2010.
- W. Nam, S.-Y. Chung, and Y. H. Lee, “Nested Lattice Codes for Gaussian Relay Networks with Interference,” in *IEEE Transactions on Information Theory*, vol. 57, no. 12, pp. 7733-7745, Dec. 2011.
- I. Maric, A. Goldsmith, and M. Médard, “Analog network coding in the high-SNR regime,” in *Proceedings of the IEEE Wireless Network Coding Conference (WiNC 2010)*, (Boston, MA), Jun. 2010.
- B. Nazer and M. Gastpar, “Reliable physical layer network coding,” *Proceedings of the IEEE*, vol. 99, no. 3, pp. 438–460, Mar. 2011.
- S.-C. Liew, S. Zhang, and L. Lu, “Physical-layer network coding: Tutorial, survey, and beyond,” in *Physical Communication*, vol. 6, pp. 4-42, Mar. 2013.

References – Lattice Codes

J. H. Conway and N. J. A. Sloane, *Sphere Packings, Lattices and Groups. New York: Springer, 1992.*

R. Zamir, Lattice Coding for Signals and Networks. Cambridge University Press, 2014.

R. de Buda, "Some optimal codes have structure," *IEEE Journal on Sel. Areas Comm.*, vol. 7, no. 6, pp. 893–899, Aug. 1989.

T. Linder, C. Schlegel, and K. Zeger, "Corrected proof of de Buda's theorem," *IEEE Transactions on Information Theory*, vol. 39, no. 5, pp. 1735–1737, Sep. 1993.

G. Poltyrev, "On coding without restrictions for the AWGN channel," *IEEE Transactions on Information Theory*, vol. 40, no. 2, pp. 409–417, Mar. 1994.

H.-A. Loeliger, "Averaging bounds for lattices and linear codes," *IEEE Transactions on Information Theory*, vol. 43, no. 6, pp. 1767–1773, Nov. 1997.

R. Urbanke and B. Rimoldi, "Lattice codes can achieve capacity on the AWGN channel," *IEEE Transactions on Information Theory*, vol. 44, no. 1, pp. 273–278, Jan. 1998.

G. Forney, M. Trott, and S.-Y. Chung, "Sphere-bound-achieving coset codes and multilevel coset codes," *IEEE Transactions on Information Theory*, vol. 46, no. 3, pp. 820–850, May 2000.

R. Zamir, S. Shamai (Shitz), and U. Erez, "Nested linear/lattice codes for structured multiterminal binning," *IEEE Transactions on Information Theory*, vol. 48, no. 6, pp. 1250–1276, Jun. 2002.

U. Erez and R. Zamir, "Achieving $\frac{1}{2}\log(1+\text{SNR})$ on the AWGN channel with lattice encoding and decoding," *IEEE Transactions on Information Theory*, vol. 50, no. 10, pp. 2293–2314, Oct. 2004.

U. Erez, S. Litsyn, and R. Zamir, "Lattices which are good for (almost) everything," *IEEE Transactions on Information Theory*, vol. 51, no. 10, pp. 3401–3416, Oct. 2005.

O. Ordentlich and U. Erez, "A simple proof for the existence of 'good' pairs of nested lattices," Proceedings of the 27th IEEEI, Eilat, Israel, Nov. 2012, available online: http://arxiv.org/abs/1209.5083

References – MIMO Channels I

G. J. Foschini, “Layered space-time architecture for wireless communication in a fading environment when using multi-element antennas,” *Bell Labs Technical Journal*, vol. 1, no. 2, pp. 41–59, Summer 1996.

G. Foschini and M. Gans, “On limits of wireless communications in a fading environment when using multiple antennas,” *Wireless Personal Commun.*, Mar. 1998.

E. Telatar, “Capacity of multi-antenna Gaussian channels,” *European Transactions on Telecommunications*, vol. 10, no. 6, pp. 585–595, Nov.-Dec. 1999.

A. Goldsmith, S. A. Jafar, N. Jindal, and S. Vishwanath, “Capacity limits of MIMO channels,” *IEEE Journal on Selected Areas in Communications*, vol. 21, no. 5, pp. 684–702, Jun. 2000.

L. Zheng and D. Tse, “Diversity and multiplexing: A fundamental tradeoff in multiple antenna channels,” *IEEE Transactions on Information Theory*, vol. 49, no. 5, pp. 1073–1096, May 2003.

S. M. Alamouti, “A simple transmit diversity technique for wireless communications,” *IEEE Journal on Selected Areas in Communications*, vol. 16, no. 8, pp. 1451–1458, Oct. 1998.

V. Tarokh, N. Seshadri, and A. Calderbank, “Space-time codes for high data rate wireless communication: Performance criterion and code construction,” *IEEE Transactions on Information Theory*, vol. 44, no. 2, pp. 744–765, Feb. 1998.

B. Hassibi and B. M. Hochwald, “High-rate codes that are linear in space and time,” *IEEE Transactions on Information Theory*, vol. 48, no. 7, pp. 1804–1824, Jul. 2002.

H. El Gamal, G. Caire, and M. O. Damen, “Lattice coding and decoding achieve the optimal diversity-multiplexing tradeoff of MIMO channels,” *IEEE Transactions on Information Theory*, vol. 50, no. 6, pp. 968–985, Jun. 2004.

H. Jafarkhani, *Space-time coding: theory and practice*. Cambridge University Press, 2005.

B. Hassibi and H. Vikalo, “On the sphere-decoding algorithm I. Expected complexity,” *IEEE Transactions on Signal Processing*, vol. 53, no. 8, pp. 2806 – 2818, Aug. 2005.

References – MIMO Channels II

J. Jalden and B. Ottersten, "On the complexity of sphere decoding in digital communications," *IEEE Transactions on Signal Processing*, vol. 53, no. 4, pp. 1474–1484, Apr. 2005.

A. Burg, M. Borgmann, M. Wenk, M. Zellweger, W. Fichtner, and H. Bolcskei, "VLSI implementation of MIMO detection using the sphere decoding algorithm," *IEEE Journal of Solid-State Circuits*, vol. 40, no. 7, pp. 1566–1577, Jul. 2005.

J. Jalden and P. Elia, "Sphere decoding complexity exponent for decoding full-rate codes over the quasi-static MIMO channel," *IEEE Transactions on Information Theory*, vol. 58, no. 9, pp. 5785–5803, Sep. 2012.

H. Yao and G. W. Wornell, "Lattice-reduction-aided detectors for MIMO communication systems," in *GLOBECOM*, Taipei, Taiwan, Nov. 2002.

M. Taherzadeh, A. Mobasher, and A. Khandani, "LLL reduction achieves the receive diversity in MIMO decoding," *IEEE Transactions on Information Theory*, vol. 53, no. 12, pp. 4801–4805, Dec. 2007.

R. Lupas and S. Verdú, "Linear multiuser detectors for synchronous code-division multiple-access channels," *IEEE Transactions on Information Theory*, vol. 35, no. 1, pp. 123–136, Dec. 1989.

U. Madhow and M. L. Honig, "MMSE interference suppression for direct-sequence spread-spectrum CDMA," *IEEE Transactions on Communications*, vol. 42, no. 12, pp. 3178–3188, Dec. 1994.

A. Hedayat and A. Nosratinia, "Outage and diversity of linear receivers in flat-fading MIMO channels," *IEEE Transactions on Signal Processing*, vol. 55, no. 12, pp. 5868–5873, Dec. 2007.

K. Kumar, G. Caire, and A. Moustakas, "Asymptotic performance of linear receivers in MIMO fading channels," *IEEE Transactions on Information Theory*, vol. 55, no. 10, pp. 4398–4418, Oct. 2009.

M. Varanasi and T. Guess, "Optimum decision feedback multiuser equalization with successive decoding achieves the total capacity of the Gaussian multiple-access channel," in *Asilomar Conf.* , Pacific Grove, CA, Nov. 1997.

P. W. Wolniansky, G. J. Foschini, G. D. Golden, and R. A. Valenzuela, "V-BLAST: an architecture for realizing very high data rates over the rich-scattering wireless channel," in *ISSSE*, Pisa, Italy, Sept.-Oct. 1998.

References – Integer-Forcing

J. Zhan, B. Nazer, U. Erez, M. Gastpar, "Integer-forcing linear receivers," *IEEE Transactions on Information Theory*, to appear, 2014. Available online: http://arxiv.org/abs/1003.5966

O. Ordentlich and U. Erez, "Precoded integer-forcing equalization universally achieves the MIMO capacity up to a constant gap," submitted, 2013. Available online: http://arxiv.org/abs/1301.6393

S.-N. Hong and G. Caire, "Compute-and-Forward Strategies for Cooperative Distributed Antenna Systems," *IEEE Transactions on Information Theory*, vol. 59, no. 9, pp. 5227–5243, Sep. 2013.

A. Sakzad, J. Harshan, and E. Viterbo, "Integer-Forcing MIMO Linear Receivers Based on Lattice Reduction," submitted, 2012. Available online: http://arxiv.org/abs/1209.6412

W. He, B. Nazer, S. Shamai (Shitz), "Uplink-Downlink Duality for Integer-Forcing," *Proc. IEEE Int. Symp. Inf. Theory*, Honolulu, HI, Jul. 2014.

O. Ordentlich, U. Erez, B. Nazer, "Successive Integer-Forcing and its Sum-Rate Optimality," *Proc. Allerton Conf. Commun. Control Comput.*, Monticello, IL, Oct. 2013. Available online: http://arxiv.org/abs/1307.2105

References – Multiple-Access Channels

R. Ahlswede, “Multi-way communication channels,” in *Proc. Int. Symp. Inf. Theory*, Thakadsor, Armenian SSR, SSR, 1971, pp. 23–52.

H. Liao, “Multiple access channels,” Ph.D. dissertation, Univ. Hawaii, Honolulu, HI, 1972.

T. Cover, “Some advances in broadcast channels,” Advances in Communication Systems, vol. 4. New York: Academic, 1975, pp. 229–260.

A. D. Wyner, Recent results in the Shannon theory, *IEEE Transactions on Information Theory*, vol. 20, no. 1, Jan. 1974.

B. Rimoldi and R. Urbanke, “A rate-splitting approach to the Gaussian multiple-access channel,” *IEEE Transactions on Information Theory*, vol. 42, no. 2, pp. 364–375, Mar. 1996.

O. Ordentlich, U. Erez, and B. Nazer, “The approximate sum capacity of the symmetric K-user Gaussian interference channel,” *IEEE Transactions on Information Theory*, to appear. Available online: http://dx.doi.org/10.1109/TIT.2014.2316136.

B. Nazer, “Successive compute-and-forward,” in *Proc. Int. Zurich Seminar on Comm.*, Zurich, Switzerland, March 2012.

V. Ntranos, V. Cadambe, B. Nazer, and G. Caire, “Asymmetric compute-and-forward,” in *Proc. 51st Allerton Conference*, Monticello, IL, Oct. 2013.

J. Zhu and M. Gastpar, “Asymmetric compute-and-forward with CSIT,” in *Proc. Int. Zurich Seminar on Comm.*, Zurich, Switzerland, March 2014.

References – Interference Channels

A. B. Carleial, "Interference channels," *IEEE Transactions on Information Theory*, vol. 21, pp. 569–570, Sep. 1975.

H. Sato, "The capacity of the Gaussian interference channel under strong interference," *IEEE Transactions on Information Theory*, vol. 27, pp. 786–788, Nov. 1981.

T. S. Han and K. Kobayashi, "A new achievable rate region for the interference channel," *IEEE Transactions on Information Theory*, vol. 27, no. 1, pp. 49–60, Jan. 1981.

R. H. Etkin, D. N. C. Tse, and H. Wang, "Gaussian interference channel capacity to within one bit," *IEEE Transactions on Information Theory*, vol. 54, no. 12, pp. 5534–5562, Dec. 2008.

A. S. Motahari and A. K. Khandani, "Capacity bounds for the Gaussian interference channel," *IEEE Transactions on Information Theory*, vol. 55, no. 2, pp. 620–643, Feb. 2009.

X. Shang, G. Kramer, and B. Chen, "A new outer bound and the noisy-interference sum-rate capacity for Gaussian interference channels," *IEEE Transactions on Information Theory*, vol. 55, no. 2, pp. 689–699, Feb. 2009.

V. S. Annapureddy and V. V. Veeravalli, "Gaussian interference networks: Sum capacity in the low-interference regime and new outer bounds on the capacity region," *IEEE Transactions on Information Theory*, vol. 55, no. 7, pp. 3032–3050, Jul. 2009.

V. R. Cadambe and S. A. Jafar, "Parallel Gaussian channels are not always separable," *IEEE Transactions on Information Theory*, vol. 55, no. 9, pp. 3983–3990, Sep. 2009.

L. Sankar, X. Shang, E. Erkip, and H. V. Poor, "Ergodic fading interference channels: Sum-capacity and separability," *IEEE Transactions on Information Theory*, vol. 57, pp. 2605–2626, May 2011.

L. Zhou and W. Yu, "On the capacity of the K-user cyclic Gaussian interference channel," *IEEE Transactions on Information Theory*, vol. 59, no. 1, pp. 154–165, Jan. 2013.

References – Interference Alignment I

M. A. Maddah-Ali, A. S. Motahari, and A. K. Khandani, "Communication over MIMO X channels: Interference alignment, decomposition, and performance analysis," *IEEE Transactions on Information Theory*, vol. 54, no. 8, pp. 3457–3470, Aug. 2008.

S. A. Jafar and S. Shamai (Shitz), "Degrees of freedom region for the MIMO X channel," *IEEE Transactions on Information Theory*, vol. 54, no. 1, pp. 151–170, Jan. 2008.

V. R. Cadambe and S. A. Jafar, "Interference alignment and the degrees of freedom for the K user interference channel," *IEEE Transactions on Information Theory*, vol. 54, no. 8, pp. 3425–3441, Aug. 2008.

Y. Birk and T. Kol, "Informed-Source Coding-On-Demand (ISCOD) over Broadcast Channels," *Proc. INFOCOM*, San Francisco, CA, Mar. 1998.

S. A. Jafar, "Interference alignment: A new look at signal dimensions in a communication network," *Foundations and Trends in Communications and Information Theory*, vol. 7, no. 1, pp. 1–136, 2011.

B. Nazer, M. Gastpar, S. A. Jafar, and S. Vishwanath, "Ergodic interference alignment," in *IEEE Transactions on Information Theory*, vol. 58, no. 10, pp. 6355–6371, Oct. 2012.

A. S. Motahari, S. O. Gharan, M.-A. Maddah-Ali, and A. K. Khandani, "Real interference alignment: Exploiting the potential of single antenna systems," *IEEE Transactions on Information Theory*, accepted. Available online http://arxiv.org/abs/0908.2282

References – Interference Alignment II

R. Etkin and E. Ordentlich, “The degrees-of-freedom of the K-user Gaussian interference channel is discontinuous at rational channel co- efficients,” *IEEE Transactions on Information Theory*, vol. 55, no. 11, pp. 4932–4946, Nov. 2009.

Y. Wu, S. Shamai (Shitz), and S Verdú, “Degrees of freedom of the interference channel: a general formula,” *Proc. IEEE Int. Symp. Inf. Theory*, St. Petersburg, Russia, Aug. 2011.

G. Bresler, A. Parekh, and D. Tse, “The approximate capacity of the many-to-one and one-to-many Gaussian interference channels, *IEEE Transactions on Information Theory*, vol. 56, no. 9, pp. 4566–4592, Sep. 2010.

S. A. Jafar and S. Vishwanath, “Generalized degrees of freedom of the symmetric Gaussian K-user interference channel,” *IEEE Transactions on Information Theory*, vol. 56, no. 7, pp. 3297–3303, Jul. 2010.

S. Sridharan, A. Jafarian, S. Vishwanath, and S. A. Jafar, “Capacity of symmetric K-user Gaussian very strong interference channels, in *Proc. GLOBECOM*, New Orleans, LA, Dec. 2008.

O. Ordentlich and U. Erez, On the robustness of lattice interference alignment, IEEE Transactions on Information Theory, vol. 59, no. 5, pp. 2735–2759, May 2013.

O. Ordentlich, U. Erez, and B. Nazer, “The approximate sum capacity of the symmetric K-user Gaussian interference channel,” *IEEE Transactions on Information Theory*, to appear. Available online: http://dx.doi.org/10.1109/TIT.2014.2316136.

V. Ntranos, V. Cadambe, B. Nazer, and G. Caire, “Integer-forcing interference alignment,” in *Proc. IEEE Int. Symp. Inf. Theory*, Istanbul, Turkey, Jul. 2013.

K. Gomadam, V. Cadambe, and S. A. Jafar, “A distributed numerical approach to interference alignment and applications to wireless interference networks,” *IEEE Transactions on Information Theory*, vol. 57, no. 6, pp. 3309–3322, Jun. 2011.

References – Codes and Constellations for Compute-and-Forward

C. Feng, D. Silva, and F. Kschischang, "An algebraic approach to physical-layer network coding," *IEEE Transactions on Information Theory*, vol. 59, no. 11, pp. 7576–7596, Nov. 2013.

B. Hern and K. Narayanan, "Multilevel coding schemes for compute- and-forward with flexible decoding," *IEEE Transactions on Information Theory*, vol. 59, no. 11, pp. 7613–7631, Nov. 2013.

H. J. Yang, Y. Choi, and J. Chun, "Modified high-order PAMs for binary coded physical-layer network coding," *IEEE Communications Letters*,

O. Ordentlich and U. Erez, "Cyclic-coded integer-forcing equalization," *IEEE Transactions on Information Theory*, vol. 58, no. 9, pp. 5804–5815, Sep. 2012.

T. Yang, I. Land, T. Huang, J. Yuan, and Z. Chen, "Distance spectrum and performance of channel-coded physical-layer network coding for binary-input Gaussian two-way relay channels," vol. 60, no. 6, pp. 1499–1510, Jun. 2012.

S.-N. Hong and G. Caire, "Quantized compute and forward: A low-complexity architecture for distributed antenna systems," *Proc. IEEE Inf. Theory Workshop*, Paraty, Brazil, Oct. 2011.

O. Ordentlich, J. Zhan, U. Erez, B. Nazer, and M. Gastpar. "Practical code design for compute-and-forward", *Proc. IEEE Int. Symp. Inf. Theory*, St. Petersburg, Russia, Jun. 2011.

J. C. Belfiore and C. Ling, "The flatness factor in lattice network coding: Design criterion and decoding algorithm," in *Proc. Int. Zurich Seminar on Comm.*, Zurich, Switzerland, Mar. 2012.

N. E. Tunali, K. R. Narayanan, and H. D. Pfister, "Spatially-coupled low density lattices based on construction A with applications to compute-and-forward," *Proc. IEEE Inf. Theory Workshop*, Seville, Spain, Sep. 2013.

Y.-C. Huang, K. R. Narayanan, and N. E. Tunali, "Multistage compute-and-forward with multilevel lattice codes based on product constructions," *IEEE Transactions on Information Theory*, submitted, Jan. 2014. Available online: http://arxiv.org/abs/1401.2228

References – Algebraic Structure in Network Information Theory I

- J. Körner and K. Marton, "How to encode the modulo-two sum of binary sources," *IEEE Transactions on Information Theory*, vol. 25, no. 2, pp. 219–221, Mar. 1979.
- D. Krithivasan and S. S. Pradhan, "Lattices for distributed source coding: Jointly Gaussian sources and reconstruction of a linear function," *IEEE Transactions on Information Theory*, vol. 55, pp. 5268–5651, December 2009.
- D. Krithivasan and S. S. Pradhan, "Distributed source coding using Abelian group codes: A new achievable rate-distortion region," *IEEE Transactions on Information Theory*, vol. 57, no.3, pp. 1495–1519, March 2011.
- A. B. Wagner, On distributed compression of linear functions, *IEEE Transactions on Information Theory.*, Vol. 57, No. 1, pp. 79-94, 2011.
- M. A. Maddah-Ali, and D. N. C. Tse, "Interference neutralization in distributed lossy source coding," in *Proc. IEEE Int. Symp. Inf. Theory*, Austin, TX, Jun. 2010.
- M. P. Wilson, K. Narayanan, H. Pfister, and A. Sprintson, "Joint physical layer coding and network coding for bidirectional relaying," *IEEE Trans. Inf. Theory*, vol. 11, no. 56, pp. 5641–5654, Nov. 2010.
- W. Nam, S.-Y. Chung, and Y. H. Lee, "Capacity of the Gaussian two-way relay channel to within $1/2$ bit," *IEEE Transactions on Information Theory*, vol. 56, no. 11, pp. 5488–5494, Nov. 2010.
- W. Nam, S.-Y. Chung, and Y. H. Lee, "Nested lattice codes for Gaussian relay networks with interference," *IEEE Transactions on Information Theory*, vol. 57, no. 12, pp. 7733–7745, Dec. 2011.

References – Algebraic Structure in Network Information Theory II

- J. Goseling, M. Gastpar, and J. Weber, "Line and lattice networks under deterministic interference models," *IEEE Transactions on Information Theory*, vol. 57, no. 5, pp. 3080–3099, May 2011.
- Y. Song and N. Devroye, "Lattice codes for the Gaussian relay channel: Decode-and-forward and compress-and-forward," *IEEE Transactions on Information Theory*, vol. 59, no. 8, pp. 4927– 4948, Aug. 2013.
- M. Nokleby and B. Aazhang, "Cooperative compute-and-forward," Mar. 2012. Available online: http://arxiv.org/abs/1203.0695
- A. Sanderovich, M. Peleg and S. Shamai (Shitz), "Scaling laws and techniques in decentralized processing of interfered Gaussian channels," *European Trans. on Telecommunications*, vol. 22, pp. 240–253, 2011.
- B. Nazer, A. Sanderovich, M. Gastpar, and S. Shamai, "Structured Superposition for Backhaul Constrained Cellular Uplink," in *Proc. IEEE Int. Symp. Inf. Theory*, Seoul, South Korea, Jun. 2009.
- S.-N. Hong and G. Caire, "Compute-and-Forward Strategies for Cooperative Distributed Antenna Systems," *IEEE Transactions on Information Theory*, vol. 59, no. 9, pp. 5227–5243, Sep. 2013.
- T. Philosof and R. Zamir, "The rate loss of single-letter characterization: The "dirty" multiple access channel," *IEEE Transactions on Information Theory*, vol. 55, no. 6, pp. 2442–2454, Jun. 2009.
- T. Philosof, R. Zamir, U. Erez, and A. J. Khisti, "Lattice strategies for the dirty multiple access channel," *IEEE Transactions on Information Theory*, vol. 57, no. 8, pp. 5006–5035, Aug. 2011.
- I.-H. Wang, "Approximate capacity of the dirty multiple-access channel with partial state information at the encoders," *IEEE Transactions on Information Theory*, vol. 58, no. 5, pp. 2781–2787, May 2012.

References – Algebraic Structure in Network Information Theory III

Y. Kochman and R. Zamir, “Analog matching of colored sources to colored channels,” *IEEE Transactions on Information Theory*, vol. 57, no. 6, pp. 3180–3195, Jun. 2011.

B. Nazer and M. Gastpar, “Computation over multiple-access channels,” *IEEE Transactions on Information Theory*, vol. 53, no. 10, pp. 3498–3516, Oct. 2007.

B. Nazer and M. Gastpar, “Structured random codes and sensor network coding theorems,” in *Proc. Int. Zurich Seminar on Comm.*, Zurich, Switzerland, Mar. 2008.

R. Soundararajan and S. Vishwanath, “Communicating linear functions of correlated Gaussian sources over a MAC,” *IEEE Transactions on Information Theory*, vol. 58, no. 3, pp. 1853-1860, Mar. 2012.

X. He and A. Yener, “The Gaussian many-to-one interference channel with confidential messages,” *IEEE Transactions on Information Theory*, vol. 57, no. 5, pp. 2730-2745, May 2011.

X. He and A. Yener, “Providing secrecy with structured codes: Tools and applications to two-user Gaussian channels,” *IEEE Transactions on Information Theory*, accepted. Available online: http://arxiv.org/abs/0907.5388

N. Kashyap, V. Shashank, and A. Thangaraj, “Secure Compute-and-Forward in a Bidirectional Relay,” *IEEE Transactions on Information Theory*, submitted, 2012. Available online: http://arxiv.org/abs/1206.3392