Integer-Forcing: An Algebraic Approach to Interference Management

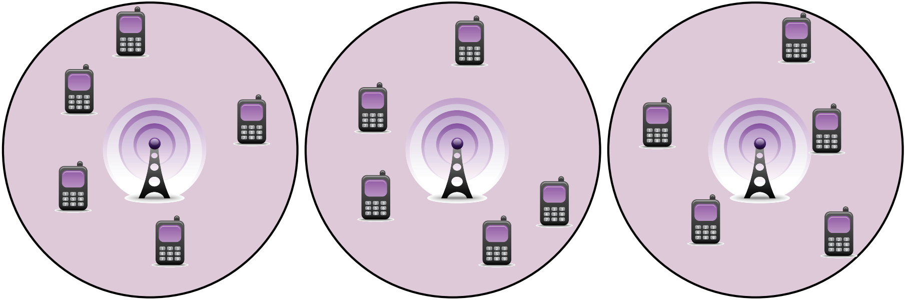
Bobak Nazer
Boston University

Thanks to Wenbo He, Engin Tunali, and Krishna Narayanan
for help with the plots.

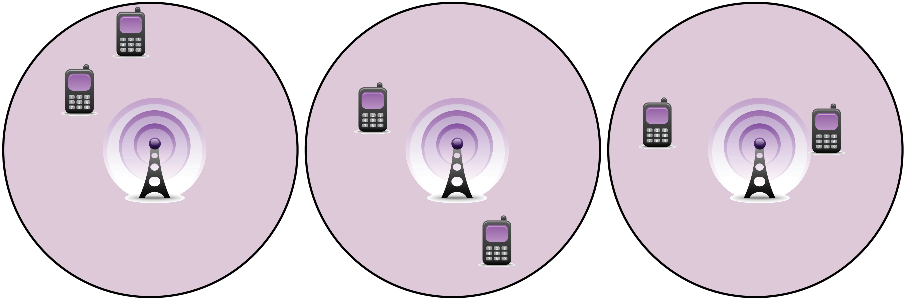
Communications Theory Workshop
Interference Management Session
May 27, 2014

Supported by NSF grants CCF-1253918 and CCF-1302600.

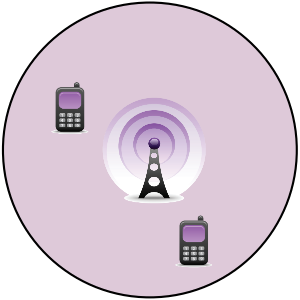
Motivation



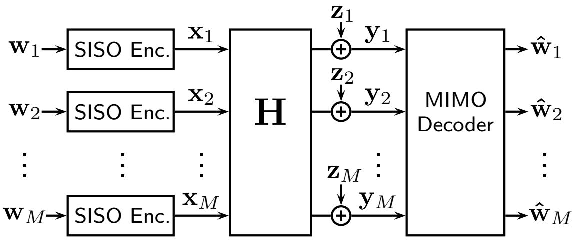
Motivation



Motivation



MIMO Uplink Channel



Usual Assumptions:

- Each antenna carries an independent data stream $\mathbf{x}_\ell \in \mathbb{C}^n$ of rate R (e.g., V-BLAST setting, cellular uplink). $\mathbf{X} = [\mathbf{x}_1 \ \cdots \ \mathbf{x}_M]^\mathsf{T}$.
- Usual power constraint: $\|\mathbf{x}_\ell\|^2 \leq n\mathsf{SNR}$.
- Channel model: $\mathbf{Y} = \mathbf{H}\mathbf{X} + \mathbf{Z}$
- $\mathbf{Z}$ is elementwise i.i.d. $\mathcal{CN}(0,1)$.
- CSIR: Only the receiver knows channel realization $\mathbf{H} \in \mathbb{C}^{M \times M}$.

Outage Rates and Probabilities

- Throughout the talk, we will assume that $\mathbf{H}$ is elementwise i.i.d. Rayleigh, remains fixed throughout the block, and is only known at the receiver.

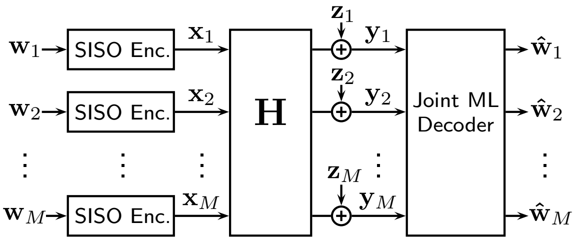
- Say that we have a scheme that achieves rate $R_{\text{scheme}}(\mathbf{H})$ under channel realization $\mathbf{H}$. For a target rate R, the outage probability is

$$p_{\text{out}}(R) = \mathbb{P}\big(R_{\text{scheme}}(\mathbf{H}) < R\big)$$

  and the outage rate is

$$R_{\text{out}}(\rho) = \sup\big\{R : p_{\text{out}}(R) \leq \rho\big\}.$$

MIMO Uplink Channel: Joint ML Decoding

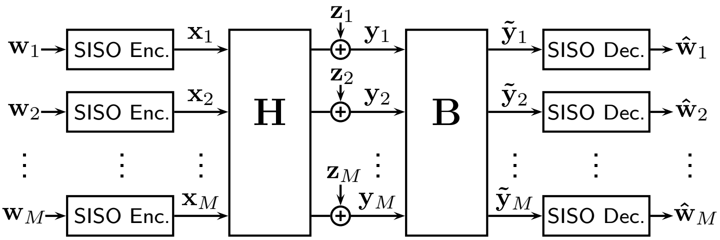


Joint Maximum Likelihood Decoding:

$$R_{\mathsf{joint}}(\mathbf{H}) = \min_{\mathcal{S} \subseteq \{1,\dots,M\}} \frac{1}{|\mathcal{S}|} \log\det\left(\mathbf{I} + \mathsf{SNR}\ \mathbf{H}_{\mathcal{S}}\mathbf{H}_{\mathcal{S}}^*\right)$$

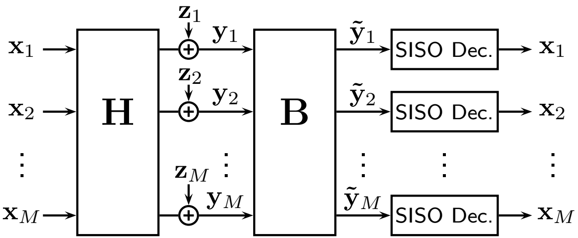
- Corresponds to the (symmetric) outage capacity.
- Naive implementation has prohibitively high complexity.
- Of course, there are many clever ways to reduce the complexity!

MIMO Uplink Channel: Zero-Forcing and Linear MMSE



Zero-Forcing and Linear MMSE Receivers:

- Project the received signal, $\tilde{\mathbf{Y}} = \mathbf{B}\mathbf{Y}$ to eliminate interference between data streams.
- After projection, single-user decoders attempt to recover the individual data streams.
- Optimal $\mathbf{B}$ is the MMSE projection.



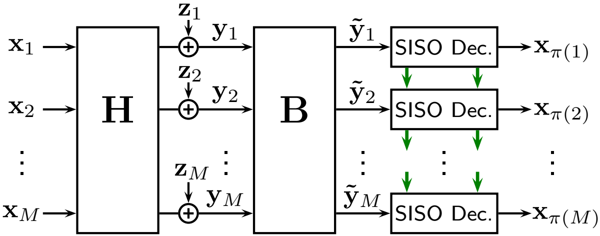
Zero-Forcing and Linear MMSE Receivers:

- The m^{th} SISO decoder tries to recover $\mathbf{x}_m$ from $\mathbf{b}_m^\mathsf{T}\mathbf{Y}$:

$$\mathsf{SINR}_{\mathsf{LMMSE},m}(\mathbf{H}) = \max_{\mathbf{b}_m} \frac{\mathsf{SNR}\ \|\mathbf{b}_m^\mathsf{T}\mathbf{h}_m\|^2}{1 + \mathsf{SNR}\ \sum_{\ell \neq m} \|\mathbf{b}_m^\mathsf{T}\mathbf{h}_\ell\|^2}$$

- Rate per user:

$$R_{\mathsf{LMMSE}}(\mathbf{H}) = \min_{m=1,\ldots,M} \log\left(1 + \mathsf{SINR}_{\mathsf{LMMSE},m}(\mathbf{H})\right)$$



Successive Interference Cancellation Receivers:

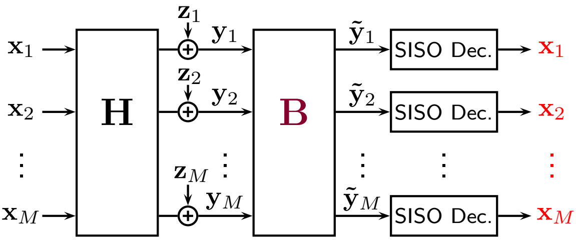
- Decode in order π. Cancel $\mathbf{x}_{\pi(1)}, \ldots, \mathbf{x}_{\pi(m-1)}$ from $\tilde{\mathbf{y}}_m$:

$$\mathsf{SINR}_{\mathsf{SIC},\pi(m)}(\mathbf{H}) = \max_{\mathbf{b}_m} \frac{\mathsf{SNR}\ \|\mathbf{b}_m^{\mathsf{T}} \mathbf{h}_{\pi(m)}\|^2}{1 + \mathsf{SNR}\ \sum_{\ell=m+1}^{M} \|\mathbf{b}_m^{\mathsf{T}} \mathbf{h}_{\pi(\ell)}\|^2}$$

- Rate per user:

$$R_{\mathsf{V\text{-}BLAST\ II}}(\mathbf{H}) = \max_{\pi} \min_{m=1,\ldots,M} \log\left(1 + \mathsf{SINR}_{\mathsf{SIC},\pi(m)}(\mathbf{H})\right)$$

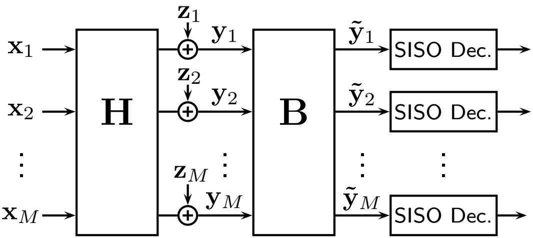
MIMO Uplink Channel: Integer-Forcing



What if we could decode something else?

- Zero-Forcing / LMMSE: First, eliminate interference.
  Then, decode individual data streams.

MIMO Uplink Channel: Integer-Forcing

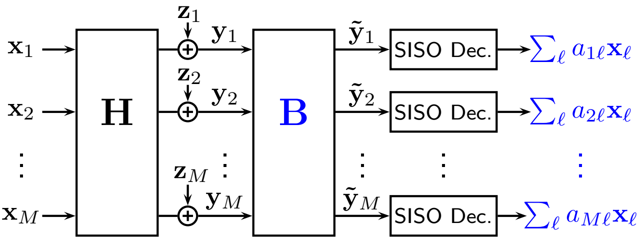


What if we could decode something else?

- Zero-Forcing / LMMSE: First, eliminate interference.
  Then, decode individual data streams.

First, decode

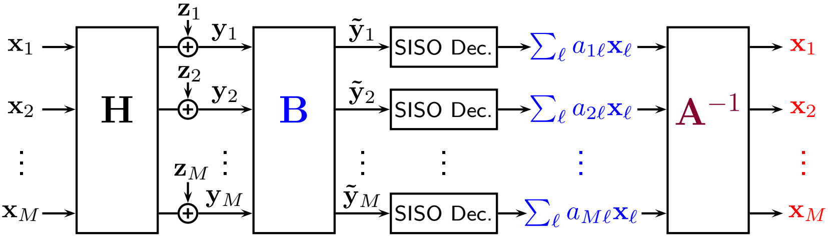
MIMO Uplink Channel: Integer-Forcing



What if we could decode something else?

- Zero-Forcing / LMMSE: First, eliminate interference.
  Then, decode individual data streams.

- Integer-Forcing: First, decode integer-linear combinations.

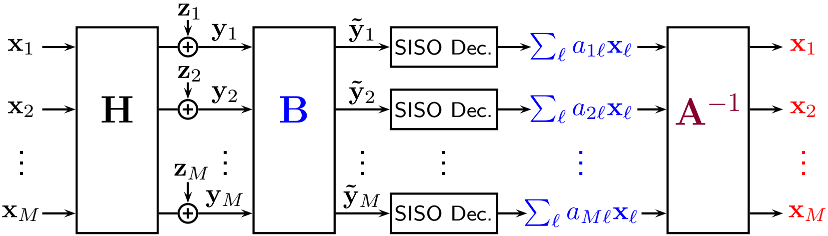


What if we could decode something else?

- Zero-Forcing / LMMSE: First, eliminate interference.
  Then, decode individual data streams.

- Integer-Forcing: First, decode integer-linear combinations.
  Then, eliminate interference.

MIMO Uplink Channel: Integer-Forcing



What if we could decode something else?

- Zero-Forcing / LMMSE: First, eliminate interference.
  Then, decode individual data streams.

- Integer-Forcing: First, decode integer-linear combinations.
  Then, eliminate interference.

- If the integer matrix $\mathbf{A}$ is full rank, we can successfully recover the individual data streams.

Integer-Forcing Linear Receivers:

- The m^{th} effective channel after projection is

$$\mathbf{b}_m^{\mathsf{T}}\mathbf{Y} = \mathbf{b}_m^{\mathsf{T}}\mathbf{H}\mathbf{X} + \mathbf{b}_m^{\mathsf{T}}\mathbf{Z}$$

Integer-Forcing Linear Receivers:

- The m^{th} effective channel after projection is

$$\begin{aligned}\mathbf{b}_m^{\mathsf{T}}\mathbf{Y} &= \mathbf{b}_m^{\mathsf{T}}\mathbf{H}\mathbf{X} + \mathbf{b}_m^{\mathsf{T}}\mathbf{Z} \\ &= \mathbf{a}_m^{\mathsf{T}}\mathbf{X} + (\mathbf{b}_m^{\mathsf{T}}\mathbf{H} - \mathbf{a}_m^{\mathsf{T}})\mathbf{X} + \mathbf{b}_m^{\mathsf{T}}\mathbf{Z}\end{aligned}$$

Integer-Forcing Linear Receivers:

- The m^{th} effective channel after projection is

$$\begin{aligned}\mathbf{b}_m^{\mathsf{T}}\mathbf{Y} &= \mathbf{b}_m^{\mathsf{T}}\mathbf{H}\mathbf{X} + \mathbf{b}_m^{\mathsf{T}}\mathbf{Z} \\ &= \mathbf{a}_m^{\mathsf{T}}\mathbf{X} + (\mathbf{b}_m^{\mathsf{T}}\mathbf{H} - \mathbf{a}_m^{\mathsf{T}})\mathbf{X} + \mathbf{b}_m^{\mathsf{T}}\mathbf{Z} \\ &= \underbrace{\sum_{\ell=1}^{M} a_{m\ell}\mathbf{x}_\ell^{\mathsf{T}}}_{\text{Codeword}} + \underbrace{(\mathbf{b}_m^{\mathsf{T}}\mathbf{H} - \mathbf{a}_m^{\mathsf{T}})\mathbf{X} + \mathbf{b}_m^{\mathsf{T}}\mathbf{Z}}_{\text{Effective Noise}}\end{aligned}$$

Integer-Forcing Linear Receivers:

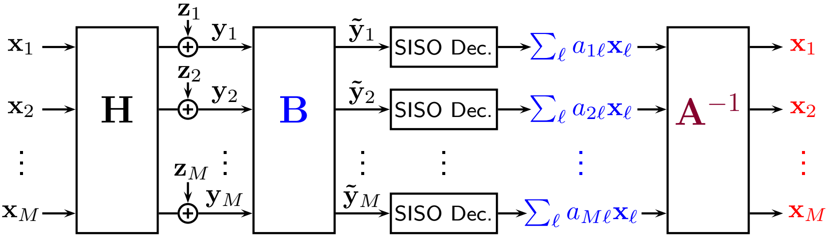
- The m^{th} effective channel after projection is

$$\begin{aligned}\mathbf{b}_m^{\mathsf{T}}\mathbf{Y} &= \mathbf{b}_m^{\mathsf{T}}\mathbf{H}\mathbf{X} + \mathbf{b}_m^{\mathsf{T}}\mathbf{Z} \\ &= \mathbf{a}_m^{\mathsf{T}}\mathbf{X} + (\mathbf{b}_m^{\mathsf{T}}\mathbf{H} - \mathbf{a}_m^{\mathsf{T}})\mathbf{X} + \mathbf{b}_m^{\mathsf{T}}\mathbf{Z} \\ &= \underbrace{\sum_{\ell=1}^{M} a_{m\ell}\mathbf{x}_\ell^{\mathsf{T}}}_{\text{Codeword}} + \underbrace{(\mathbf{b}_m^{\mathsf{T}}\mathbf{H} - \mathbf{a}_m^{\mathsf{T}})\mathbf{X} + \mathbf{b}_m^{\mathsf{T}}\mathbf{Z}}_{\text{Effective Noise}}\end{aligned}$$

- The $a_{m\ell} \in \mathbb{Z}[j]$ are Gaussian integers and the codebook should be closed under integer-linear combinations.

- We are free to choose any full-rank integer-valued matrix $\mathbf{A}$.

MIMO Uplink Channel: Integer-Forcing



Integer-Forcing Linear Receivers: (**Zhan-Nazer-Erez-Gastpar '12**)

- The m^{th} SISO decoder tries to recover $\sum_\ell a_{m\ell}\mathbf{x}_\ell$ from $\mathbf{b}_m^\mathsf{T}\mathbf{Y}$:

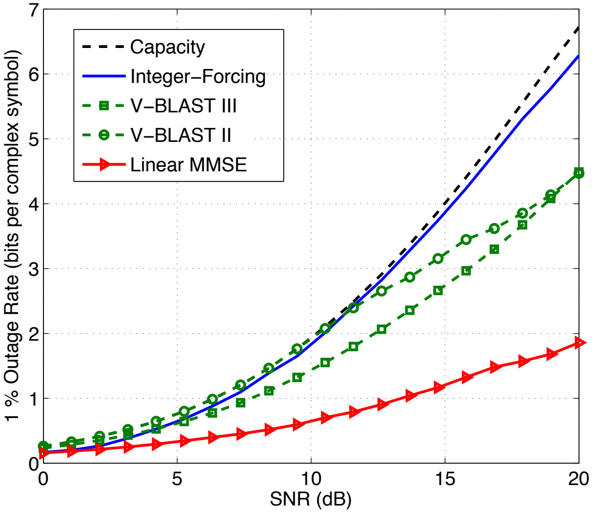
$$\mathsf{SINR}_{\mathsf{IF},m}(\mathbf{H},\mathbf{A}) = \max_{\mathbf{b}_m} \frac{\mathsf{SNR}}{\|\mathbf{b}_m\|^2 + \mathsf{SNR}\|\mathbf{b}_m^\mathsf{T}\mathbf{H} - \mathbf{a}_m^\mathsf{T}\|^2}$$

- Rate per user:

$$R_{\mathsf{IF}}(\mathbf{H}) = \max_{\mathbf{A}} \min_{m=1,\dots,M} \log^+\left(\mathsf{SINR}_{\mathsf{IF},m}(\mathbf{H},\mathbf{A})\right)$$

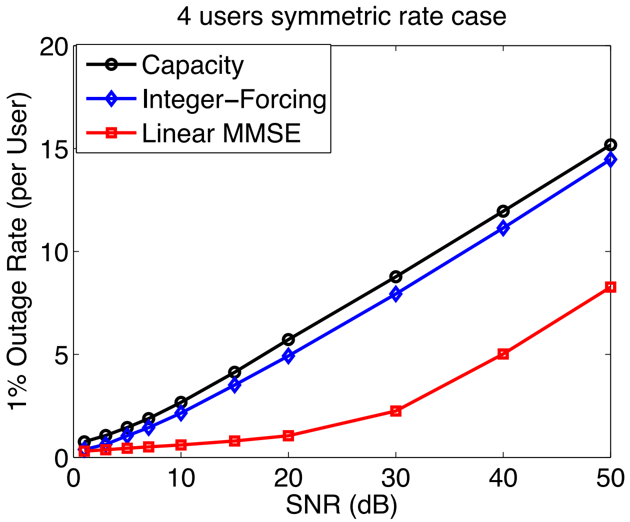
- Includes linear MMSE as a special case by setting $\mathbf{A} = \mathbf{I}$.

Comparison: Outage Rates



2 users, 2 receive antennas, Rayleigh fading, 1% outage.

Comparison: Outage Rates



4 users, 4 receive antennas, Rayleigh fading, 1% outage.

Questions

- How can we efficiently select a good integer matrix $\mathbf{A}$?
- How does the performance scale with the number of users?
- How sensitive is the performance to imperfect CSIR?
- What types of SISO encoders and decoders can we use?
- What about the downlink?
- Can we move beyond this idealized problem setting?

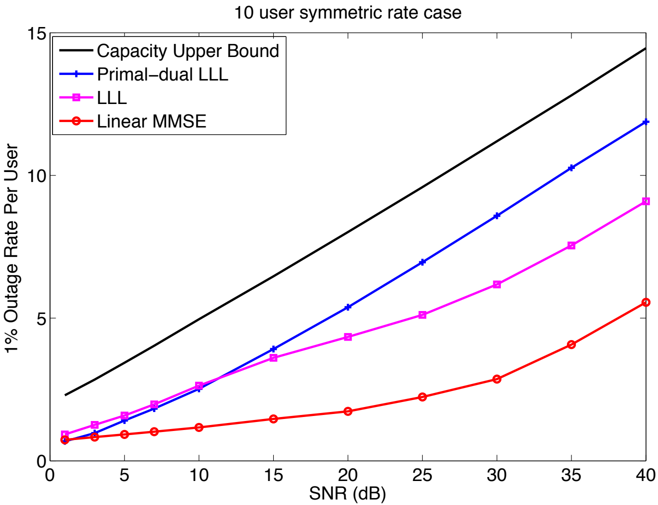
Finding a Good Integer Matrix

$$\mathsf{SINR}_{\mathsf{IF},m}(\mathbf{H},\mathbf{A}) = \max_{\mathbf{b}_m} \frac{\mathsf{SNR}}{\|\mathbf{b}_m\|^2 + \mathsf{SNR}\|\mathbf{H}^\mathsf{T}\mathbf{b}_m - \mathbf{a}_m\|^2}$$

- Optimal $\mathbf{b}_m$ is the MMSE projection.
- Plugging in and applying the Matrix Inversion Lemma, we get that

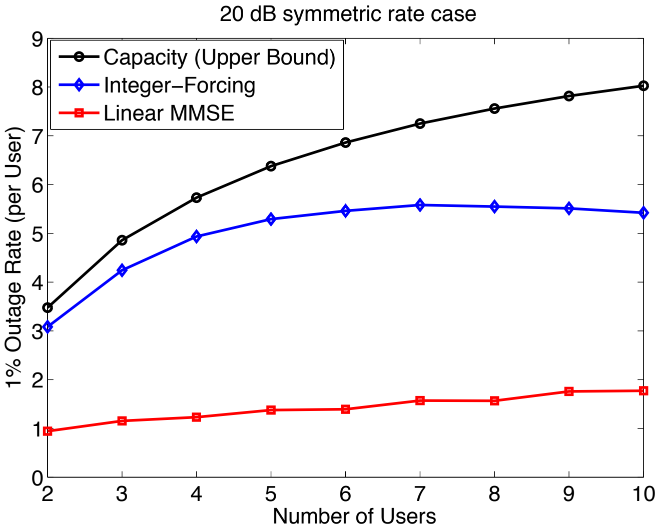
$$\mathsf{SINR}_{\mathsf{IF},m}(\mathbf{H},\mathbf{A}) = \frac{1}{\left\|\left(\mathbf{I} + \mathsf{SNR}\ \mathbf{H}^*\mathbf{H}\right)^{-1/2}\mathbf{a}_m\right\|^2}$$

- Finding the optimal $\mathbf{A}$ corresponds to finding a good lattice basis.
- This is a hard problem in general but good approximation algorithms are known, such as the LLL algorithm.
- We are currently using a slight twist: We run LLL to get a lattice basis. Then, we turn to the dual lattice and run LLL again, initializing with the first basis.



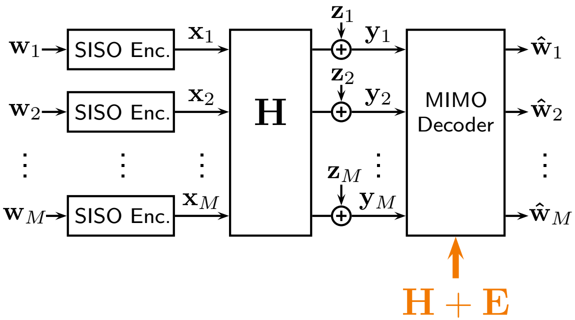
Rayleigh fading, 1% outage.

How does the performance scale with the number of users?



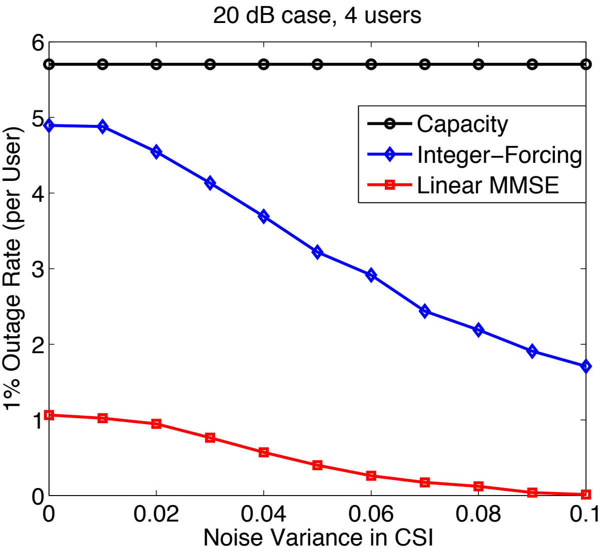
Rayleigh fading, 1% outage.

What is the impact of imperfect CSIR?



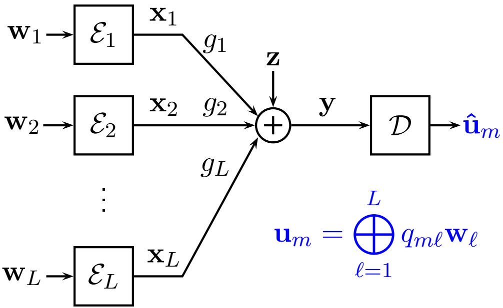
- Receiver only sees $\mathbf{H} + \mathbf{E}$ where $\mathbf{E}$ is elementwise i.i.d. $\mathcal{CN}(0, \sigma^2)$.
- May result in selecting both a suboptimal integer matrix $\mathbf{A}$ and a suboptimal projection matrix $\mathbf{B}$.

What is the impact of imperfect CSIR?



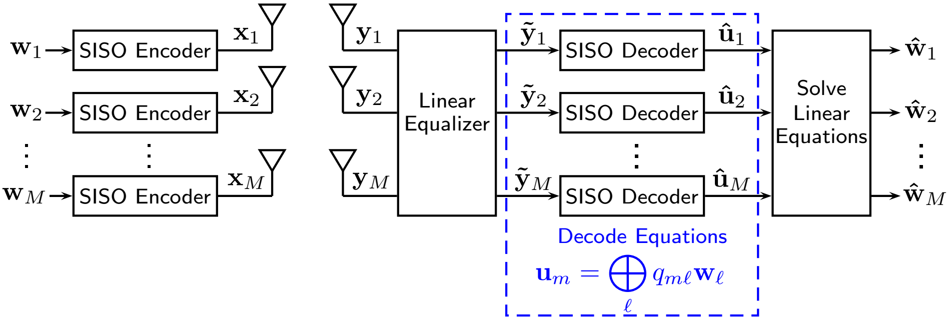
4 users, 20dB, Rayleigh fading, 1% outage.

What kinds of SISO coding schemes can be used?



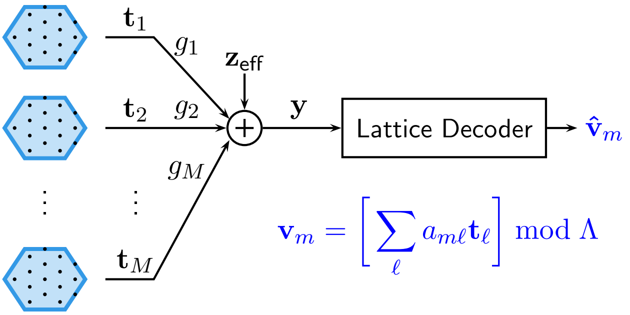
- Underlying integer-forcing is the compute-and-forward framework, which is used as a black box to recover linear combinations of the messages over some finite field $\mathbb{F}_p$.
- Messages are vectors over a prime-sized finite field, $\mathbf{w}_\ell \in \mathbb{F}_p^k$.

Integer-Forcing Linear Receiver



- Architecture is completely digital after SISO decoders.

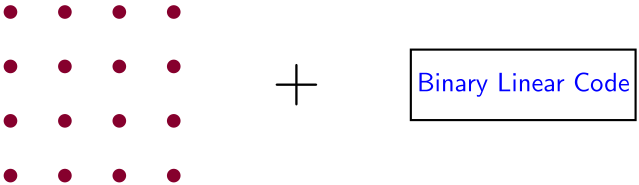
What kinds of SISO coding schemes can be used?



$$\mathbf{v}_m = \Big[\sum_{\ell} a_{m\ell} \mathbf{t}_\ell \Big] \bmod \Lambda$$

- **Nazer-Gastpar '11:** Compute-and-forward achievability proofs via nested lattice codes.
- High-dimensional nested lattice codes lead to nice $\log(\text{SINR})$ expressions but have high implementation complexity.
- Remember, all we actually need is that the codebook is closed under integer-linear combinations.

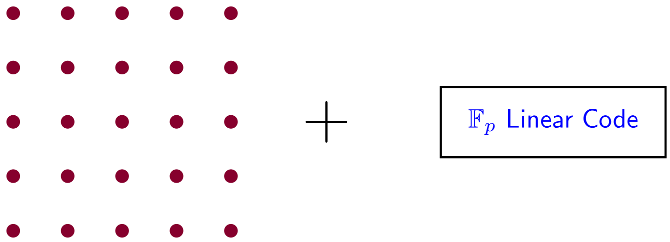
What kinds of SISO coding schemes can be used?



- What about QAM combined with a binary linear code?
- Issue: Real addition does not map well to addition over $\mathbb{F}_{2^M}$.

$$[x_1 + x_2] \bmod 2^M \neq x_1 \oplus x_2$$

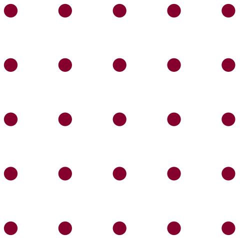
What kinds of SISO coding schemes can be used?



- What about p-ary QAM where p is prime combined with a linear code over $\mathbb{F}_p$?
- Real addition maps well to addition over $\mathbb{F}_p$.

$$[x_1 + x_2] \bmod p = x_1 \oplus x_2$$

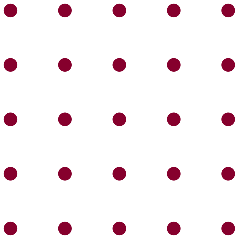
Uncoded Integer-Forcing



Uncoded Integer-Forcing:

- Project by $\mathbf{b}_m$, take $\bmod\ p$, apply slicer.
- Correct if we recover $[a_{m1}x_1 + a_{m2}x_2 + \cdots + a_{mM}x_M] \bmod p$ for all m.

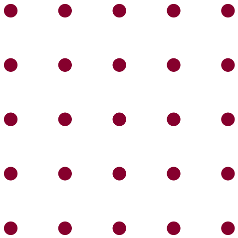
Uncoded Integer-Forcing



Uncoded Integer-Forcing:

- Project by $\mathbf{b}_m$, take $\bmod\ p$, apply slicer.
- Correct if we recover $[a_{m1}x_1 + a_{m2}x_2 + \cdots + a_{mM}x_M] \bmod p$ for all m.
- Is this lattice-aided reduction?

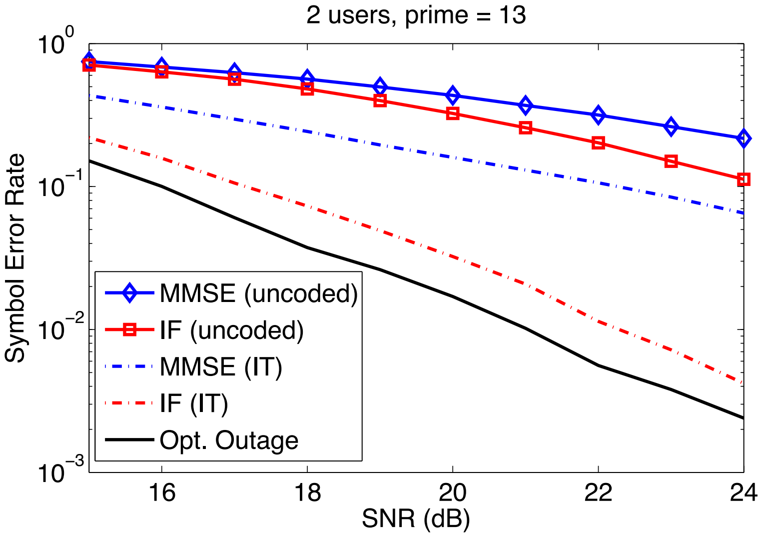
Uncoded Integer-Forcing



Uncoded Integer-Forcing:

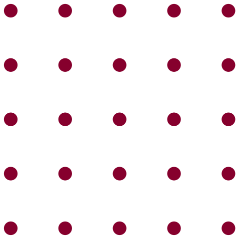
- Project by $\mathbf{b}_m$, take $\bmod\ p$, apply slicer.
- Correct if we recover $[a_{m1}x_1 + a_{m2}x_2 + \cdots + a_{mM}x_M] \bmod p$ for all m.
- Is this lattice-aided reduction? Nearly. We add the $\bmod\ p$.

Uncoded Integer-Forcing



2 users, 2 receive antennas, Rayleigh fading

$p = 13$, fixed rate $1/2 \log(13)$.

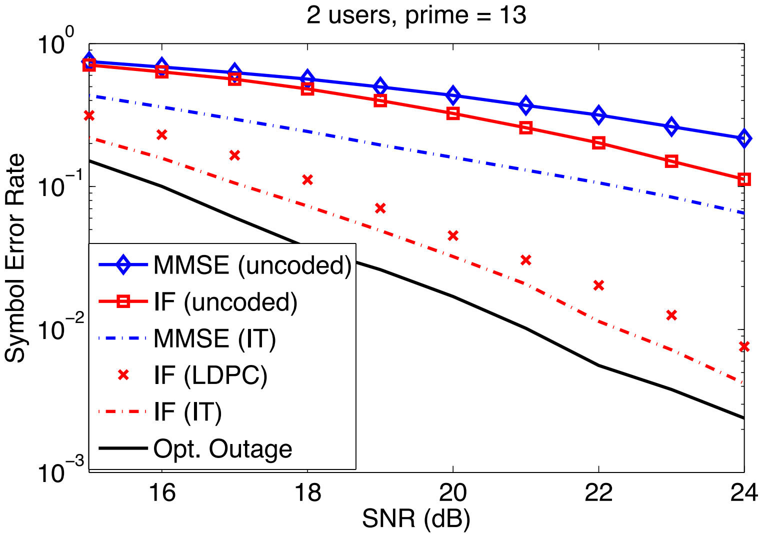


$+$

LDPC Code over $\mathbb{F}_p$

Coded Integer-Forcing:

- Project by $\mathbf{b}_m$, take $\mod p$, apply LDPC decoding algorithm.
- Correct if we recover $[a_{m1}\mathbf{x}_1 + a_{m2}\mathbf{x}_2 + \cdots + a_{mM}\mathbf{x}_M] \mod p$ for all m.



2 users, 2 receive antennas, Rayleigh fading

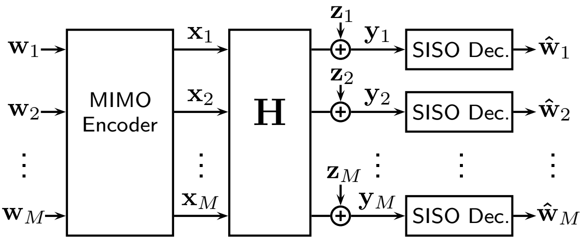
$p = 13$, fixed rate $1/2 \log(13)$, regular $(3, 6)$ LDPC code.

- Lots of interesting questions on how to design low-complexity constellations and linear codes that work well for compute-and-forward.
- Several recent papers and...

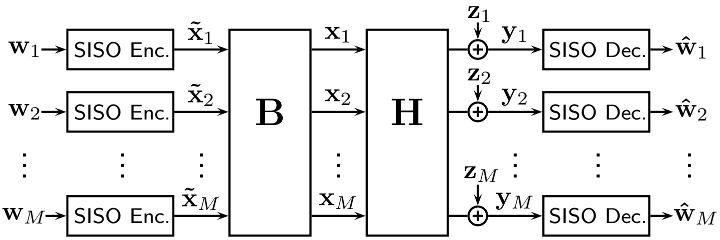
Codes for Compute-and-Forward

- Lots of interesting questions on how to design low-complexity constellations and linear codes that work well for compute-and-forward.
- Several recent papers and...
- Krishna's talk coming up next!

MIMO Downlink Channel



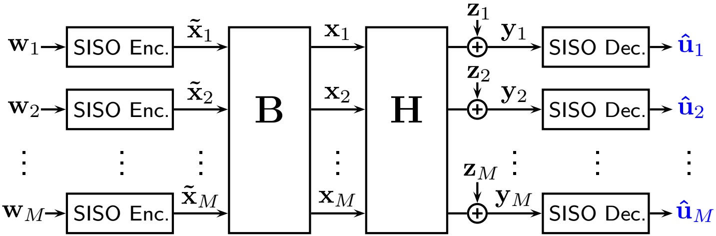
- Capacity region is known. Requires dirty-paper coding.
  Caire-Shamai '03, Vishwanath-Jindal-Goldsmith '04,
  Viswanath-Tse '03, Yu-Cioffi '04, Weingarten-Steinberg-Shamai '06.



Zero-Forcing Beamforming:

- Use beamforming matrix $\mathbf{B}$ to eliminate interference between data streams.

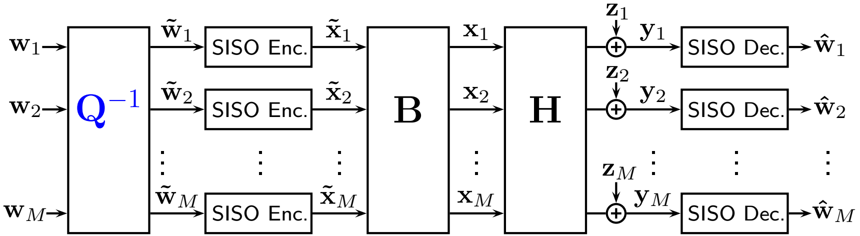
MIMO Downlink Channel: Integer-Forcing



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

Integer-Forcing Beamforming: (**Hong-Caire '12,'13**)

- Use beamforming matrix $\mathbf{B}$ to create an integer-valued effective channel $\mathbf{A}$.
- Decode linear combinations with $q_{m\ell} = [a_{m\ell}] \bmod p$.

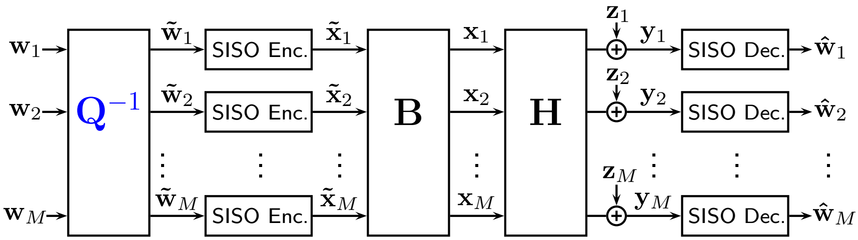


Integer-Forcing Beamforming: (**Hong-Caire '12,'13**)

- Use beamforming matrix $\mathbf{B}$ to create an integer-valued effective channel $\mathbf{A}$.

- ~~Decode linear combinations with $q_{m\ell} = [a_{m\ell}] \bmod p$.~~
  Pre-invert $\mathbf{Q} = [\mathbf{A}] \bmod p$ and decode messages.

MIMO Downlink Channel: Integer-Forcing



Integer-Forcing Beamforming: (**Hong-Caire '12,'13**)

- Use beamforming matrix $\mathbf{B}$ to create an integer-valued effective channel $\mathbf{A}$.
- ~~Decode linear combinations with $q_{m\ell} = [a_{m\ell}] \bmod p$.~~ Pre-invert $\mathbf{Q} = [\mathbf{A}] \bmod p$ and decode messages.
- In very recent work, we have shown that uplink-downlink duality holds for integer-forcing. **He-Nazer-Shamai '14**

Extensions

- What can we prove about the optimality of integer-forcing?

Extensions

- What can we prove about the optimality of integer-forcing?
- **Zhan-Nazer-Erez-Gastpar '12:** Attains the optimal DMT (for point-to-point MIMO with no space-time coding.)

Extensions

- What can we prove about the optimality of integer-forcing?
- **Zhan-Nazer-Erez-Gastpar '12:** Attains the optimal DMT (for point-to-point MIMO with no space-time coding.)
- What about space-time coding at the transmitter?

Extensions

- What can we prove about the optimality of integer-forcing?
- **Zhan-Nazer-Erez-Gastpar '12:** Attains the optimal DMT (for point-to-point MIMO with no space-time coding.)
- 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 the optimal DMT as a special case.

- What can we prove about the optimality of integer-forcing?
- **Zhan-Nazer-Erez-Gastpar '12:** Attains the optimal DMT (for point-to-point MIMO with no space-time coding.)
- 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 the optimal DMT as a special case.
- What about successive cancellation for integer-forcing?

- What can we prove about the optimality of integer-forcing?
- **Zhan-Nazer-Erez-Gastpar '12:** Attains the optimal DMT (for point-to-point MIMO with no space-time coding.)
- 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 the optimal DMT as a special case.
- What about successive cancellation for integer-forcing?
- **Ordentlich-Erez-Nazer '13:** Framework for IF-SIC. Exact optimality if CSIT is available. Rate points tend to lie very close to the symmetric capacity.

Key Issues Going Forward

- Low-complexity constellations and codes.
- New algorithms for finding integer matrix $\mathbf{A}$.
- Synchronization.
- What if the channel realization changes over the coding blocklength? (e.g., OFDM)
- How should we include rate adaptation?
- What does this mean for user selection?
- With Behnaam, Krishna, and students, we are working towards a WARP implementation.
- Any others?