Combinatorial Channel Signature Modulation for Wireless Networks

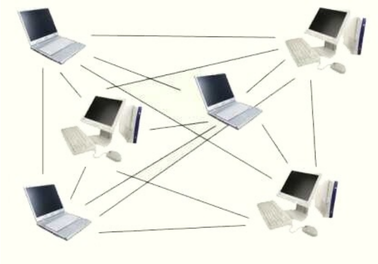
Dino Sejdinovic (Gatsby Unit, UCL)
joint work with Rob Piechocki (CCR, Bristol)

Signal Processing and Communications Laboratory, University of Cambridge

February 8, 2012

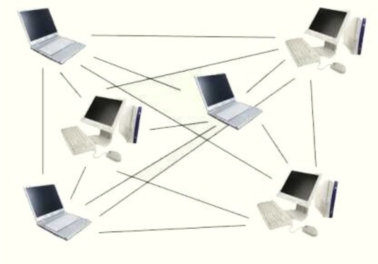
Problem outline

- wireless mutual broadcast with $N+1$ half-duplex nodes
- each node has a k-bit message to transmit to all others
- Can all nodes transmit at the same frequency without elaborate time scheduling?



Problem outline

- wireless mutual broadcast with $N+1$ half-duplex nodes
- each node has a k-bit message to transmit to all others
- Can all nodes transmit at the same frequency without elaborate time scheduling?



Introduction

CCSM (Combinatorial Channel Signature Modulation):

- sparse, combinatorial representation of messages
- compressed sensing based decoding
- minimal MAC Layer coordination: access to a shared channel
  - no need for collision detection/avoidance scheme (CSMA/CA)
- robust to time dispersion
  - no need for guard intervals to eliminate ISI

Introduction

CCSM (Combinatorial Channel Signature Modulation):

- sparse, combinatorial representation of messages
- compressed sensing based decoding
- minimal MAC Layer coordination: access to a shared channel
  - no need for collision detection/avoidance scheme (CSMA/CA)
- robust to time dispersion
  - no need for guard intervals to eliminate ISI

Introduction

CCSM (Combinatorial Channel Signature Modulation):

- sparse, combinatorial representation of messages
- compressed sensing based decoding
- minimal MAC Layer coordination: access to a shared channel
  - no need for collision detection/avoidance scheme (CSMA/CA)
- robust to time dispersion
  - no need for guard intervals to eliminate ISI

Overview

CS for multiterminal communications

Combinatorial encoder

Sparse recovery solver

Simulation results

Outline

CS for multiterminal communications

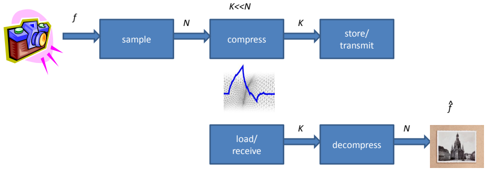
Combinatorial encoder

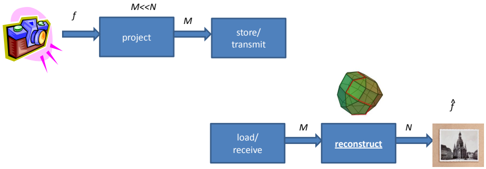
Sparse recovery solver

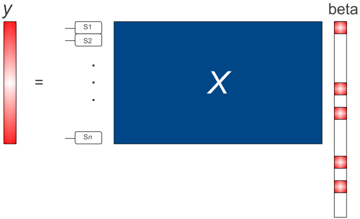
Simulation results

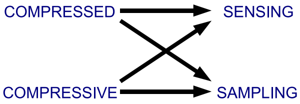
Compressed sensing

- Compressed sensing (Candes, Romberg and Tao 2006; Donoho 2006) combines sampling and compression steps into one - into taking random linear projections.





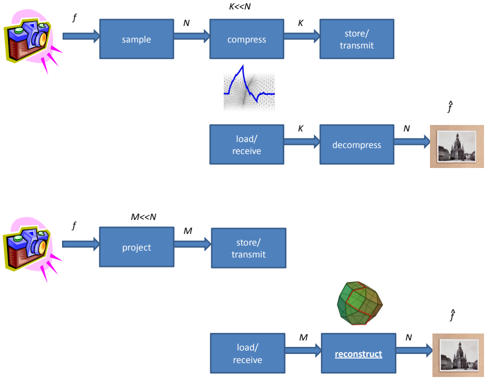


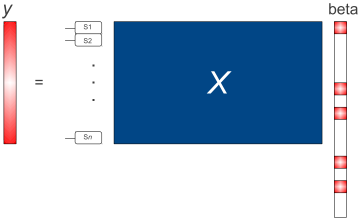


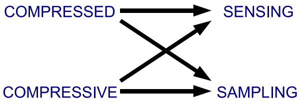
projections $\approx$ number of non-zeros $\times$ log (size of the vector)

Compressed sensing

- Compressed sensing (Candes, Romberg and Tao 2006; Donoho 2006) combines sampling and compression steps into one - into taking random linear projections.



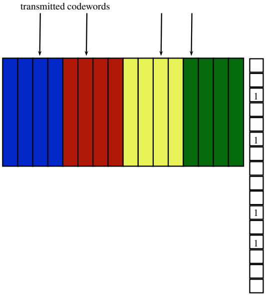


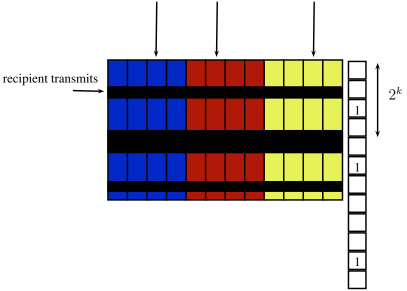


projections $\approx$ number of non-zeros $\times$ log (size of the vector)

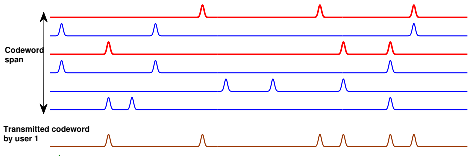
Embracing the additive nature of wireless

- (Zhang and Guo 2010): each node is assigned a (known) dictionary of (sparse) on-off signalling codewords, each codeword corresponding to a single message
- Own transmissions seen as erasures in dictionary
- The dictionary size exponential in the number of bits k in a message - sparse recovery problem of size $2^k N$





Combinations of the codebook elements?

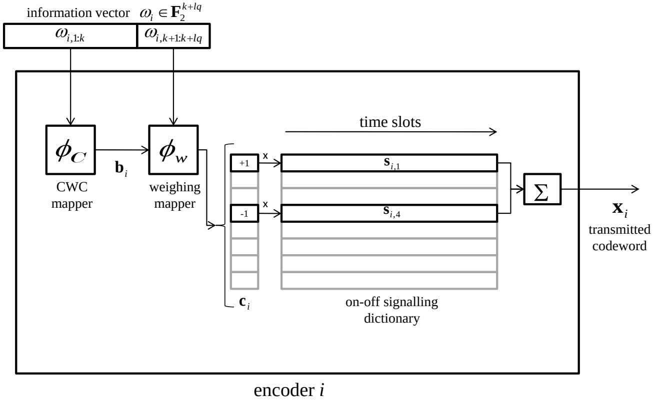


- Idea: Transmit (weighted) sums of a fixed number l of on-off signalling codewords.
- the *choice* of the l-combination of the dictionary elements carries information
- Requires efficient encoding of messages into l-combinations: **constant weight coding** (l out of L codes)
- Need $l \ll L$ for sparse recovery

Reduction of complexity

- (Zhang and Guo, 2010): $k = \log_2 L$ - sparse recovery problem of size $2^k N$
- CCSM: $k \approx \log_2 \binom{L}{l} = \mathcal{O}(L^\alpha \log_2 L)$, for $l = L^\alpha$, $0 < \alpha < 1$ - sparse recovery problem of size$\sim k^{1/\alpha} N$

Encoder



$$\phi_{\mathcal{C}} \; : \; 00000 \;\mapsto\; 010100000$$

$$\phi_{\mathcal{C}} \; : \; 00001 \;\mapsto\; 000101000$$

$$\cdots$$

Outline

CS for multiterminal communications

Combinatorial encoder

Sparse recovery solver

Simulation results

Constant weight coding

- Combinatorial representation of information
- using l-combinations of the set of L elements, i.e., length-L binary vectors with exactly l ones.
    - $l \ll L$ (sparse)

Definition

An (L, l)-constant weight code is a set of length-L binary vectors with Hamming weight l:

$$\mathcal{C} \subseteq \{c \in \mathbb{F}_2^L : \ w_H(c) = l\}.$$

- single-bit error/single-type error detection
- two-out-of-five barcode

0 1 2 3 4 5 6 7 8 9

Constant weight coding

- Set of possible messages: $\mathcal{S} = \{0, 1 \ldots, K-1\}$. Usually $K = 2^k$, and we identify $\mathcal{S} \leftrightarrow \mathbb{F}_2^k$.
- An (L, l)-constant weight code is $\mathcal{C} \subseteq \{c \in \mathbb{F}_2^L : w_H(c) = l\}$.
- Goal: construct a bijective map $\phi_{\mathcal{C}} : \mathcal{S} \to \mathcal{C}$

- Enumeration approaches:
  - (Schalkwijk 1972), (Cover 1973): lexicographic ordering of codewords, requires registers of length $\mathcal{O}(L)$.
  - (Ramabadran 1990): arithmetic coding, computational complexity $\mathcal{O}(L)$.
  - (Knuth 1986): complementation method, computational complexity $\mathcal{O}(L)$ - but much faster in practice. Works only for balanced codes: $l = \lfloor L/2 \rfloor$.

- Can we do better when $l \ll L$?
  - (Tian, Vaishampayan, Sloane 2009): embed both $\mathcal{S}$ and $\mathcal{C}$ into $\mathbb{R}^l$ and establish bijective maps by dissecting certain polytopes in $\mathbb{R}^l$.

Constant weight coding

- Set of possible messages: $\mathcal{S} = \{0, 1 \ldots, K-1\}$. Usually $K = 2^k$, and we identify $\mathcal{S} \leftrightarrow \mathbb{F}_2^k$.
- An (L, l)-constant weight code is $\mathcal{C} \subseteq \{c \in \mathbb{F}_2^L : \ w_H(c) = l\}$.
- Goal: construct a bijective map $\phi_{\mathcal{C}} : \mathcal{S} \to \mathcal{C}$
- Enumeration approaches:
  - (Schalkwijk 1972), (Cover 1973): lexicographic ordering of codewords, requires registers of length $\mathcal{O}(L)$.
  - (Ramabadran 1990): arithmetic coding, computational complexity $\mathcal{O}(L)$.
  - (Knuth 1986): complementation method, computational complexity $\mathcal{O}(L)$ - but much faster in practice. Works only for balanced codes: $l = \lfloor L/2 \rfloor$.
- Can we do better when $l \ll L$?
  - (Tian, Vaishampayan, Sloane 2009): embed both $\mathcal{S}$ and $\mathcal{C}$ into $\mathbb{R}^l$ and establish bijective maps by dissecting certain polytopes in $\mathbb{R}^l$.

Constant weight coding

- Set of possible messages: $\mathcal{S} = \{0, 1 \ldots, K-1\}$. Usually $K = 2^k$, and we identify $\mathcal{S} \leftrightarrow \mathbb{F}_2^k$.
- An (L, l)-constant weight code is $\mathcal{C} \subseteq \{c \in \mathbb{F}_2^L : \; w_H(c) = l\}$.
- Goal: construct a bijective map $\phi_{\mathcal{C}} : \mathcal{S} \to \mathcal{C}$

- Enumeration approaches:
  - (Schalkwijk 1972), (Cover 1973): lexicographic ordering of codewords, requires registers of length $\mathcal{O}(L)$.
  - (Ramabadran 1990): arithmetic coding, computational complexity $\mathcal{O}(L)$.
  - (Knuth 1986): complementation method, computational complexity $\mathcal{O}(L)$ - but much faster in practice. Works only for balanced codes: $l = \lfloor L/2 \rfloor$.

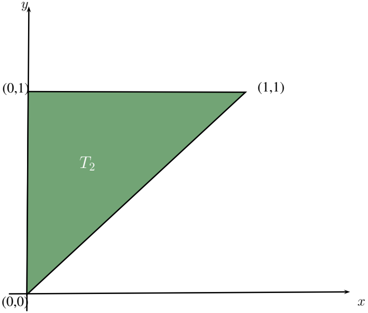
- Can we do better when $l \ll L$?
  - (Tian, Vaishampayan, Sloane 2009): embed both $\mathcal{S}$ and $\mathcal{C}$ into $\mathbb{R}^l$ and establish bijective maps by dissecting certain polytopes in $\mathbb{R}^l$.

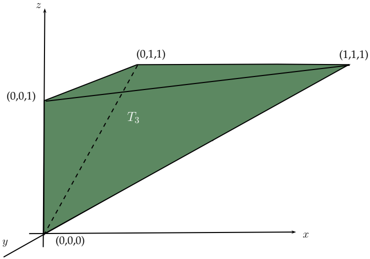
Embedding $\mathcal{C}$ into $\mathbb{R}^l$

- Given a constant-weight codeword c, define $\phi(c) = (\frac{y_1}{L}, \ldots, \frac{y_l}{L}) \in \mathbb{R}^l$, with y_i:= position of the i-th 1 in c.
- for $L = 5$, $l = 2$, $\phi(01010) = (2/5, 4/5)$.
- $\phi(\mathcal{C})$ is a discrete subset of the convex hull T_l of the points:

$$\begin{array}{ccccc}
(0, & 0, & \ldots & 0, & 0) \\
(0, & 0, & \ldots & 0, & 1) \\
\vdots & \vdots & \ldots & \vdots & \vdots \\
(0, & 1, & \ldots & 1, & 1) \\
(1, & 1, & \ldots & 1, & 1)
\end{array}$$

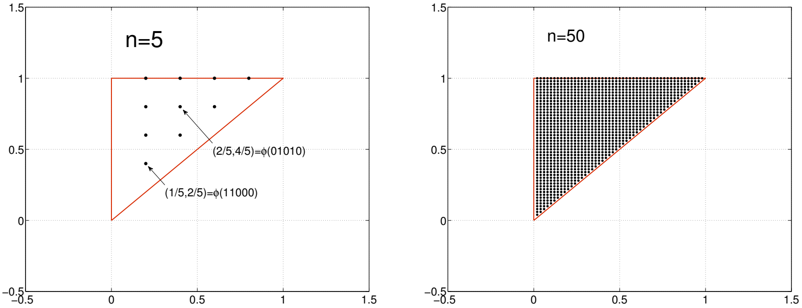
$\mathcal{C}$ embedded in a tetrahedron





- T_2 is the right triangle with vertices $(0,0)$, $(0,1)$, $(1,1)$
- $Vol(T_l) = \frac{1}{l!}$ (the unit cube can be split into $l!$ tetrahedra congruent to T_l)

The case $l = 2$

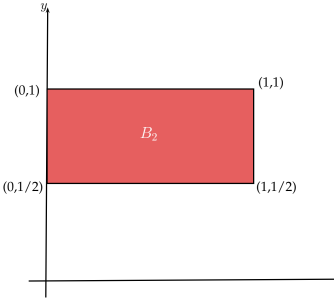


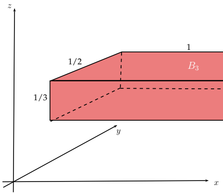
Assume: information is a brick

- Brick $B_l \subset \mathbb{R}^l$ is a hyper-rectangle

$$B_l := [0,1]\times[\frac{1}{2},1]\times[\frac{2}{3},1]\times\cdots\times[\frac{l-1}{l},1]$$

- Represent messages $s \in \{0,1,\ldots,K-1\}$, $K \leq \binom{L}{l}$ as integer l-tuple $(b_1^{(s)}, b_2^{(s)}, \ldots, b_l^{(s)})$, s.t. $(\frac{b_1^{(s)}}{n}, \frac{b_2^{(s)}}{n}, \ldots, \frac{b_l^{(s)}}{n}) \in B_l$: quotient and remainder
- $Vol(B_l) = \frac{1}{l!} = Vol(T_l)$



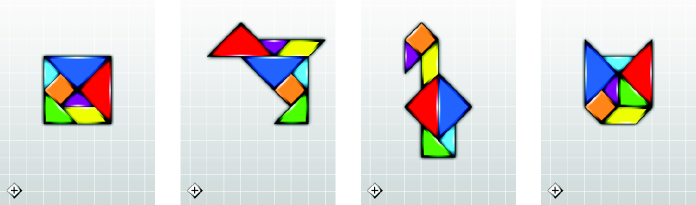


Dissections

Hilbert's third problem:

For any two polyhedra of the same volume, is it possible to dissect one into a finite number of pieces that can be rearranged to give the other?

- In $l = 2$ dimensions (Bolyai-Gerwien 1833): yes



"scissor-equivalence"

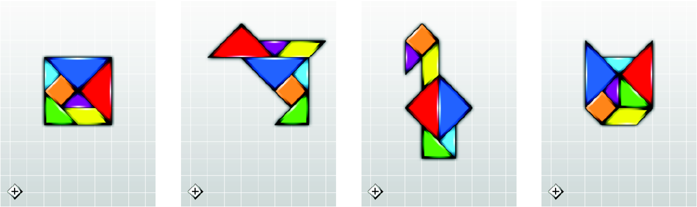
- In $d \geq 3$ dimensions (Dehn, 1902): no - polyhedra must have equal Dehn invariants.

Dissections

Hilbert's third problem:

For any two polyhedra of the same volume, is it possible to dissect one into a finite number of pieces that can be rearranged to give the other?

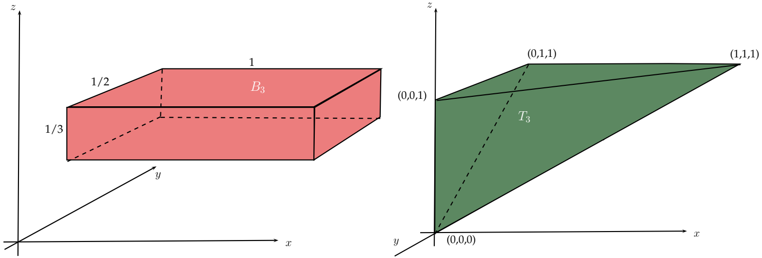
- In $l = 2$ dimensions (Bolyai-Gerwien 1833): yes



"scissor-equivalence"

- In $d \geq 3$ dimensions (Dehn, 1902): no - polyhedra must have equal Dehn invariants.

Encoding



- messages $\mathcal{S} = \{0, 1 \ldots, K-1\} \longleftrightarrow B_l \subset \mathbb{R}^l \underset{\longleftrightarrow}{diss.} T_l \subset \mathbb{R}^l$ $\longleftrightarrow$ constant weight code $\mathcal{C}$
- (Tian, Vaishampayan, Sloane 2009) give an explicit recursive dissection of B_l into T_l computable in $\mathcal{O}(l^2)$

Rate

- CWC rate: $R(\mathcal{C}) = \frac{1}{L}\log_2|\mathcal{C}| \le \frac{1}{L}\log_2\binom{L}{l} = \mathcal{O}(l\frac{\log_2 L}{L}) \to 0$, $L \to \infty$, when $l \ll L$.

$CCSM\ rate \neq CWC\ rate$

- CCSM rate: $\frac{N}{M}\left(\log_2\binom{L}{l} + lq\right)$, where
  - M is the time duration of the waveforms (number of rows in the CS problem + number of own transmissions)
- Typical CS results: it suffices to take M to be the number of non-zeros $\times$ log (size of the vector)
- $M = \mathcal{O}(lN\log_2(LN)) \Rightarrow CCSM\ rate = \mathcal{O}(\frac{\log_2 L}{\log_2 L + \log_2 N})$

Rate

- CWC rate: $R(\mathcal{C}) = \frac{1}{L}\log_2|\mathcal{C}| \leq \frac{1}{L}\log_2\binom{L}{l} = \mathcal{O}(l\frac{\log_2 L}{L}) \to 0$, $L \to \infty$, when $l \ll L$.

CCSM rate ≠ CWC rate

- CCSM rate: $\frac{N}{M}\left(\log_2\binom{L}{l} + lq\right)$, where
  - M is the time duration of the waveforms (number of rows in the CS problem + number of own transmissions)
- Typical CS results: it suffices to take M to be the number of non-zeros $\times$ log (size of the vector)
- $M = \mathcal{O}(lN\log_2(LN)) \Rightarrow$ *CCSM rate* $= \mathcal{O}(\frac{\log_2 L}{\log_2 L + \log_2 N})$

Rate

- CWC rate: $R(\mathcal{C}) = \frac{1}{L}\log_2|\mathcal{C}| \leq \frac{1}{L}\log_2\binom{L}{l} = \mathcal{O}(l\frac{\log_2 L}{L}) \to 0$, $L \to \infty$, when $l \ll L$.

$$\textit{CCSM rate} \neq \textit{CWC rate}$$

- CCSM rate: $\frac{N}{M}\left(\log_2\binom{L}{l} + lq\right)$, where
  - M is the time duration of the waveforms (number of rows in the CS problem + number of own transmissions)

- Typical CS results: it suffices to take M to be the number of non-zeros $\times$ log (size of the vector)
- $M = \mathcal{O}(lN\log_2(LN)) \Rightarrow \textit{CCSM rate} = \mathcal{O}(\frac{\log_2 L}{\log_2 L + \log_2 N})$

Outline

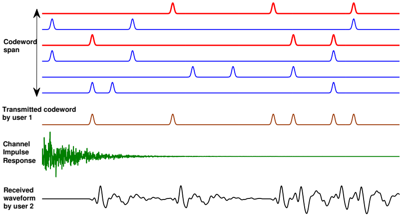
CS for multiterminal communications

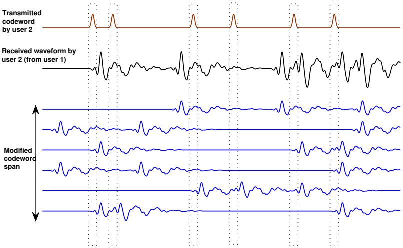
Combinatorial encoder

Sparse recovery solver

Simulation results

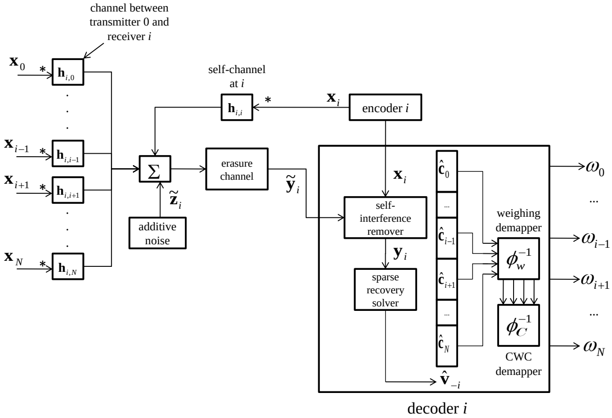
Channel signatures





- Each user convolves codebook elements with the channel impulse response
- Can subtract echoes of its own transmissions (self-interference remover)

Decoder



$$\mathbf{x}_i = \mathbf{S}_i \mathbf{c}_i$$

$$\mathbf{y}_i = \mathbf{E}_i \left(\sum_{j=0}^{N} \mathbf{h}_{i,j} * \mathbf{S}_j \mathbf{c}_j + \tilde{\mathbf{z}}_i \right) - \mathbf{E}_i \left(\mathbf{h}_{i,i} * \mathbf{S}_i \mathbf{c}_i \right) = \mathbf{A}_{-i} \mathbf{v}_{-i} + \mathbf{z}_i$$

Sparse recovery solver

$$\mathbf{y}_i = \mathbf{E}_i \left(\sum_{j=0}^{N} \mathbf{h}_{i,j} * \mathbf{S}_j \mathbf{c}_j + \tilde{\mathbf{z}}_i \right) - \mathbf{E}_i \left(\mathbf{h}_{i,i} * \mathbf{S}_i \mathbf{c}_i \right) = \mathbf{A}_{-i} \mathbf{v}_{-i} + \mathbf{z}_i$$

- User i needs to solve the following problem to detect the desired signal:

$$\begin{aligned} \hat{\mathbf{v}}_{-i} &= \arg\min_{\mathbf{v}_{-i}} \|\mathbf{y}_i - \mathbf{A}_{-i}\mathbf{v}_{-i}\|_2 \\ \text{s.t. } \|\mathbf{c}_j\|_0 &= l, \text{ for all } j \neq i, \end{aligned}$$

  - A non-convex optimisation problem.
  - Exactly Nl out of NL entries in $\mathbf{v}_{-i}$ are non-zero - but sparsity level is: $\frac{l}{L} \ll 1$
  - Can use any of the myriad of CS decoding algorithms.

Sparse recovery solver

$$\mathbf{y}_i = \mathbf{E}_i \left(\sum_{j=0}^{N} \mathbf{h}_{i,j} * \mathbf{S}_j \mathbf{c}_j + \tilde{\mathbf{z}}_i \right) - \mathbf{E}_i \left(\mathbf{h}_{i,i} * \mathbf{S}_i \mathbf{c}_i \right) = \mathbf{A}_{-i} \mathbf{v}_{-i} + \mathbf{z}_i$$

- User i needs to solve the following problem to detect the desired signal:

$$\begin{aligned} \hat{\mathbf{v}}_{-i} &= \arg\min_{\mathbf{v}_{-i}} \|\mathbf{y}_i - \mathbf{A}_{-i}\mathbf{v}_{-i}\|_2 \\ \text{s.t. } \|\mathbf{c}_j\|_0 &= l, \text{ for all } j \neq i, \end{aligned}$$

  - A non-convex optimisation problem.
  - Exactly Nl out of NL entries in $\mathbf{v}_{-i}$ are non-zero - but sparsity level is: $\frac{l}{L} \ll 1$
  - Can use any of the myriad of CS decoding algorithms.

Sparse recovery solver

$$\mathbf{y}_i = \mathbf{E}_i \left(\sum_{j=0}^{N} \mathbf{h}_{i,j} * \mathbf{S}_j \mathbf{c}_j + \tilde{\mathbf{z}}_i \right) - \mathbf{E}_i \left(\mathbf{h}_{i,i} * \mathbf{S}_i \mathbf{c}_i \right) = \mathbf{A}_{-i} \mathbf{v}_{-i} + \mathbf{z}_i$$

- User i needs to solve the following problem to detect the desired signal:

$$\begin{aligned} \hat{\mathbf{v}}_{-i} &= \arg\min_{\mathbf{v}_{-i}} \|\mathbf{y}_i - \mathbf{A}_{-i}\mathbf{v}_{-i}\|_2 \\ \text{s.t. } \|\mathbf{c}_j\|_0 &= l, \text{ for all } j \neq i, \end{aligned}$$

  - A non-convex optimisation problem.
  - Exactly Nl out of NL entries in $\mathbf{v}_{-i}$ are non-zero - but sparsity level is: $\frac{l}{L} \ll 1$
  - Can use any of the myriad of CS decoding algorithms.

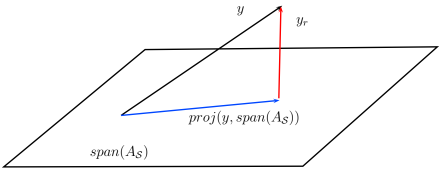
Sparse recovery solver

- Basis Pursuit/LASSO/convex relaxation (Tibshirani 1996), (Chen, Donoho, and Saunders 1998)
- LARS / homotopy (Efron et al, 2004)
- Greedy iterative methods:
  - Compressive Sampling Matching Pursuit - CoSaMP (Needell and Tropp 2009)
  - Subspace Pursuit - SP (Dai and Milenkovic 2009)

Subspace Pursuit

Greedily searches for the support set $\mathcal{S}$ such that y is closest to $span(A_{\mathcal{S}})$.

1. Initialise. Set $\mathcal{S}$ to the Nl columns that maximize $|\langle a_i, y\rangle|$
2. Identify further candidates. Set $\mathcal{S}'$ to the Nl columns that maximize $|\langle a_i, y - proj(y, span(A_{\mathcal{S}})\rangle|$
3. Merge and Prune. Set $\mathcal{S}$ to the Nl columns from $\mathcal{S} \cup \mathcal{S}'$ with largest magnitudes in $A^{+}_{\mathcal{S}\cup\mathcal{S}'}y$
4. Iterate (2)-(3) until the stopping criterion holds.



Group Subspace Pursuit

Sparse vector has additional structure: each of N subvectors has exactly l non-zero entries.

1. Initialise. Set $\mathcal{S}$ to the union of sets of l columns within each group of L that maximize $|\langle a_i, y\rangle|$
2. Identify further candidates. Set $\mathcal{S}'$ to the union of sets of l columns within each group of L that maximize $|\langle a_i, y - proj(y, span(A_{\mathcal{S}}))\rangle|$
3. Merge and Prune. Set $\mathcal{S}$ to the union of sets of l columns within each group of L with largest magnitudes in $A^{+}_{\mathcal{S}\cup\mathcal{S}'}y$
4. Iterate (2)-(3) until the stopping criterion holds.

Outline

CS for multiterminal communications

Combinatorial encoder

Sparse recovery solver

Simulation results

Group Subspace Pursuit (2)

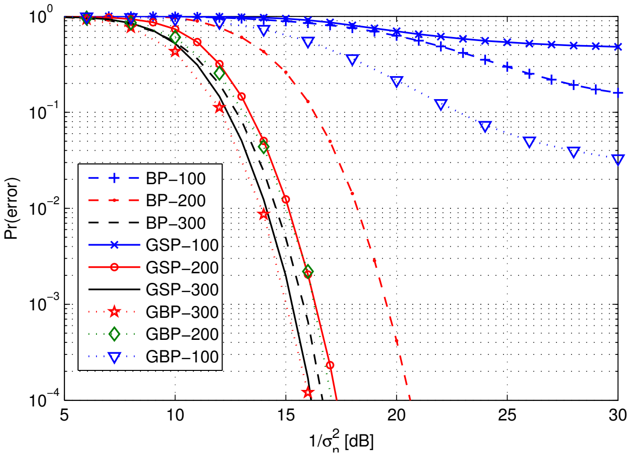


Figure: Performance comparison of Basis Pursuit/Lasso (BP), Group Basis Pursuit (GBP) and Group Subspace Pursuit (GSP). $N = 10$, $l = 4$, $L = 32$

CCSM Performance

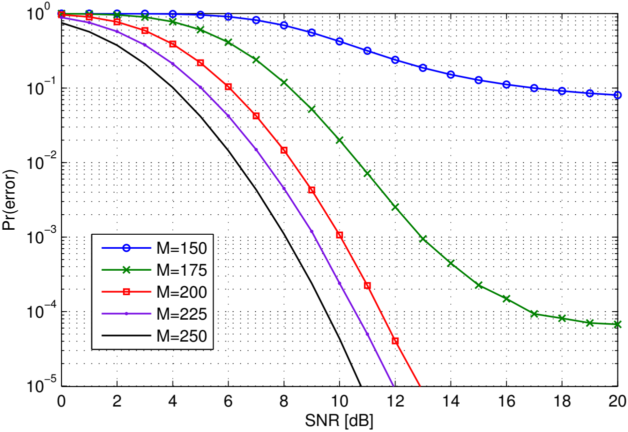
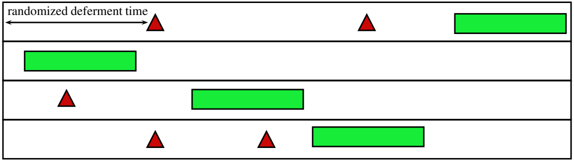


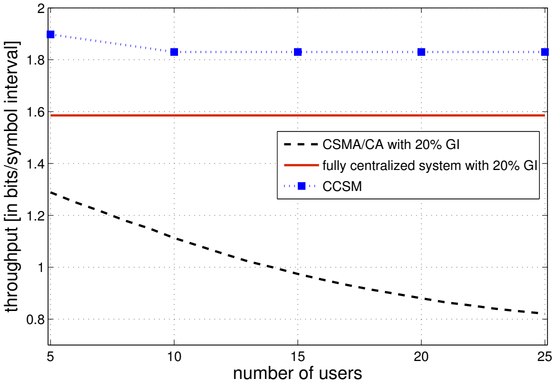
Figure: Performance of the proposed method in terms of message error rate: In this case there are $(N+1) = 5$ users simultaneously broadcasting messages using CCSM with $L = 64$, $l = 12$.

Comparison to CSMA/CA and TDMA (1)

1. TDMA
   - central controlling mechanism
   - time divided equally - no collisions, performance independent of the number of nodes
   - guard interval (cyclic prefix) of 20% slot duration
2. CSMA/CA
   - randomised deferment of transmissions in order to avoid collisions; contention window: 16-1024 symbol intervals
   - no symbol intervals wasted on distributed or short interframe space (DIFS/SIFS), propagation delay, physical or MAC message headers and ACK responses
   - a single message in each transmission queue
   - guard interval (cyclic prefix) of 20% slot duration



Comparison to CSMA/CA and TDMA (2)



CCSM: the minimum number of symbol intervals at which no message errors occurred in at least 100,000 simulation trials

$$L = 64,\ l = 12,\ q = 2 \text{ (QPSK)}$$

$$k = \left\lfloor \log_2 \binom{L}{l} \right\rfloor + lq = 65$$

Concluding remarks

- a novel decentralized modulation and multiplexing method for wireless networks
  - effective time/frequency duplex
  - minimal MAC
  - inherent robustness to time dispersion
- low computational complexity which takes advantage of
  - combinatorial representation of messages
  - sparse recovery detection
- significant throughput improvement in comparison to collision avoidance schemes

Concluding remarks

- a novel decentralized modulation and multiplexing method for wireless networks
  - effective time/frequency duplex
  - minimal MAC
  - inherent robustness to time dispersion
- low computational complexity which takes advantage of
  - combinatorial representation of messages
  - sparse recovery detection
- significant throughput improvement in comparison to collision avoidance schemes

Concluding remarks

- a novel decentralized modulation and multiplexing method for wireless networks
  - effective time/frequency duplex
  - minimal MAC
  - inherent robustness to time dispersion
- low computational complexity which takes advantage of
  - combinatorial representation of messages
  - sparse recovery detection
- significant throughput improvement in comparison to collision avoidance schemes

References

- R. Piechocki and D. Sejdinovic, "Combinatorial Channel Signature Modulation for Wireless ad-hoc Networks" preprint available from: arxiv.org/abs/1201.5608v1 (to appear in *Proc. IEEE Int. Conf. on Communications ICC* 2012).
- L. Zhang and D. Guo, "Wireless Peer-to-Peer Mutual Broadcast via Sparse Recovery" preprint available from: arxiv.org/abs/1101.0294, in Proc. *IEEE Int. Symp. Inform. Theory ISIT* 2011.
- W. Dai and O. Milenkovic, "Subspace pursuit for compressive sensing signal reconstruction," *IEEE Trans. Inform. Theory*, vol. 55, pp. 2230– 2249, 2009.
- C. Tian, V. Vaishampayan and N. J. A. Sloane, "A coding algorithm for constant weight vectors: a geometric approach based on dissections," *IEEE Trans. Inform. Theory*, vol. 55, pp. 1051–1060, 2009.
- G. Bianchi, "Performance Analysis of the IEEE 802.11 Distributed Coordination Function", *IEEE Journal on Selected Areas in Communications*, vol. 18, pp. 535–547, 2000.

Signalling dictionary

- One construction of on-off signalling dictionary:
  - all columns of $\mathbf{S}_i$ have equal number of non-zero entries, set to $\lfloor \frac{M}{L} \rfloor$
  - every two columns in $\mathbf{S}_i$ have disjoint support
  - non-zero entries selected uniformly at random from some discrete constellation
- transmitted codeword $\mathbf{x}_i$ has exactly $l \cdot \lfloor \frac{M}{L} \rfloor$ non-zero entries (on-slots)
- $\tilde{M} = M - l \cdot \lfloor \frac{M}{L} \rfloor$ off-slots used to listen to the incoming signals
- Can also use LDPC / regular Gallager constructions (Baron, Sarvotham, Baraniuk 2010)

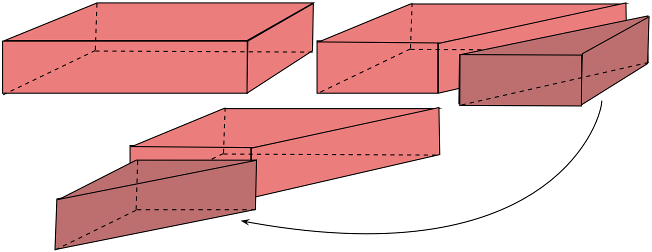
Assume: information is a brick

- Brick $B_l \subset \mathbb{R}^l$ is a hyper-rectangle
  $B_l := [0,1] \times [\frac{1}{2}, 1] \times [\frac{2}{3}, 1] \times \cdots \times [\frac{l-1}{l}, 1]$
- Let $l = 2$, and L even. Then one can uniquely represent message indices $s \in \{0, 1, \ldots, K-1\}$ (where $K \leq \frac{L(L-1)}{2}$) as integer pairs $(b_1^{(s)}, b_2^{(s)})$, where

$$\begin{aligned} 0 &\leq \alpha = & \left\lfloor \frac{s}{L/2} \right\rfloor \leq L-1 \\ 0 &\leq \beta = & s - \frac{L}{2}\alpha \leq \frac{L}{2} - 1 \end{aligned}$$

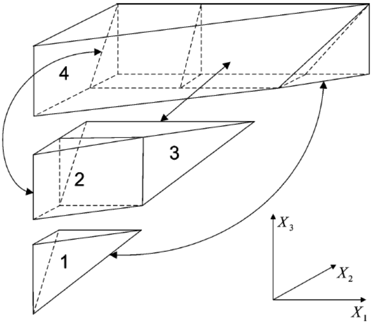
- Define $b_1^{(s)} = \alpha$, $b_2^{(s)} = \beta + \frac{L}{2}$.
- It follows that $\left\{ (\frac{b_1^{(s)}}{L}, \frac{b_2^{(s)}}{L}) \right\}_{s=0}^{K-1} \subset B_l$

Step 1: brick to triangular prism



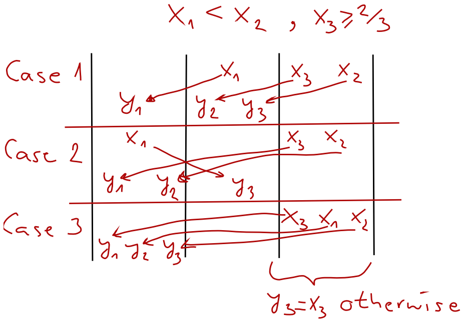
$$B_3 = B_2 \times [2/3, 1] \to T_2 \times [2/3, 1]$$

Step 2: triangular prism to tetrahedron



- inductive dissection for general w:
  1. $B_w = B_{w-1} \times [\frac{w-1}{w}, 1] \to T_{w-1} \times [\frac{w-1}{w}, 1]$
  2. $T_{w-1} \times [\frac{w-1}{w}, 1] \to T_w$

Step 2: triangular prism to tetrahedron



- inductive dissection for general w:
  1. $B_w = B_{w-1} \times [\frac{w-1}{w}, 1] \to T_{w-1} \times [\frac{w-1}{w}, 1]$
  2. $T_{w-1} \times [\frac{w-1}{w}, 1] \to T_w$
- Some loss in rate due to rounding (when n is in the range 100-1000, and $w = \sqrt{n}$, the loss is 1-2 bits/block)