Codes and Card Tricks:
Magic for Adversarial Crowds

Lele Wang

Stanford University and Tel Aviv University

Technion Israel Institute of Technology
April 17, 2016

Joint work with Sihuang Hu and Ofer Shayevitz

Magic tutorial

0000110101001000101111101100111

Magic tutorial

$\underbrace{\boxed{000011010}}_{9}$ 10010001011111101100111

Magic tutorial

$$0\ \underbrace{\boxed{000110101}}_{9}\ 0010001011111011100111$$

Magic tutorial

$$00\underbrace{\boxed{001101010}}_{9}01000101111101100111$$

Magic tutorial

$$000\underbrace{\boxed{011010100}}_{9}1000101111101100111$$

Magic tutorial

$\boxed{000011}$ 01010010001011110110 $\boxed{111}$

Magic tutorial

$\boxed{0000110}$ 1010010001011111011001 $\boxed{11}$

Magic tutorial

$\boxed{00001101}$ 0100100010111110110011 $\boxed{1}$

Magic tutorial

0000110101001000101111101100111

- Card trick: A binary sequence
  - distinct length-9 chunks

Magic tutorial

00001101010010001011111011 00111

- Card trick: A binary sequence
  - distinct length-9 chunks
  - minimum distance 3

Magic tutorial

0000110101001000101111101100111

- Card trick: A binary sequence
  - distinct length-9 chunks
  - minimum distance 3
  - $0 \rightarrow$ a red card; $1 \rightarrow$ a black card

Magic tutorial

00001101010010001011111011001111

- Card trick: A binary sequence
  - distinct length-9 chunks
  - minimum distance 3
  - $0 \to$ a red card; $1 \to$ a black card
  - cutting the deck: cyclic shift of the sequence

Magic tutorial

000011010100100010111110110 0111

- Card trick: A binary sequence
  - distinct length-9 chunks
  - minimum distance 3
  - $0 \rightarrow$ a red card; $1 \rightarrow$ a black card
  - cutting the deck: cyclic shift of the sequence
  - error correction and location

Magic tutorial

0000110101001000101111101100111

- Card trick: A binary sequence
  - distinct length-9 chunks
  - minimum distance 3
  - $0 \to$ a red card; $1 \to$ a black card
  - cutting the deck: cyclic shift of the sequence
  - error correction and location
- Problem: Design a length-n binary sequence
  - distinct length-k chunks
  - minimum distance d

de Bruijn sequence

- Problem: Design a length-n binary sequence
  - distinct length-k chunks
  - minimum distance d

de Bruijn sequence

- Problem: Design a length-n binary sequence
  - distinct length-k chunks
  - minimum distance d

- Special Case: $d = 1, n = 2^k$
  - Order-k de Bruijn sequence (1946)

de Bruijn sequence

- Problem: Design a length-n binary sequence
  - distinct length-k chunks
  - minimum distance d

- Special Case: $d = 1, n = 2^k$
  - Order-k de Bruijn sequence (1946)

$$k = 2 : \qquad 0\ 0\ 1\ 1$$

de Bruijn sequence

- Problem: Design a length-n binary sequence
  - distinct length-k chunks
  - minimum distance d

- Special Case: $d = 1, n = 2^k$
  - Order-k de Bruijn sequence (1946)

$$k = 2 : \qquad 0\ 0\ 1\ 1$$

de Bruijn sequence

- Problem: Design a length-n binary sequence
  - distinct length-k chunks
  - minimum distance d

- Special Case: $d = 1, n = 2^k$
  - Order-k de Bruijn sequence (1946)

$$k = 2 : \qquad 0\ 0\ 1\ 1$$

de Bruijn sequence

- Problem: Design a length-n binary sequence
  - distinct length-k chunks
  - minimum distance d

- Special Case: $d = 1, n = 2^k$
  - Order-k de Bruijn sequence (1946)

$$k = 2 : \qquad 0\ 0\ 1\ 1$$

de Bruijn sequence

- Problem: Design a length-n binary sequence
  - distinct length-k chunks
  - minimum distance d

- Special Case: $d = 1, n = 2^k$
  - Order-k de Bruijn sequence (1946)

$$k = 2 : \qquad 0\ 0\ 1\ 1$$

de Bruijn sequence

- Problem: Design a length-n binary sequence
  - distinct length-k chunks
  - minimum distance d

- Special Case: $d = 1, n = 2^k$
  - Order-k de Bruijn sequence (1946)

$$k = 2 : \qquad 0\ 0\ 1\ 1$$
$$k = 3 : \qquad 0\ 0\ 0\ 1\ 0\ 1\ 1\ 1$$

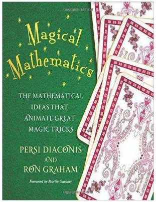
de Bruijn sequence

- Problem: Design a length-n binary sequence
  - distinct length-k chunks
  - minimum distance d

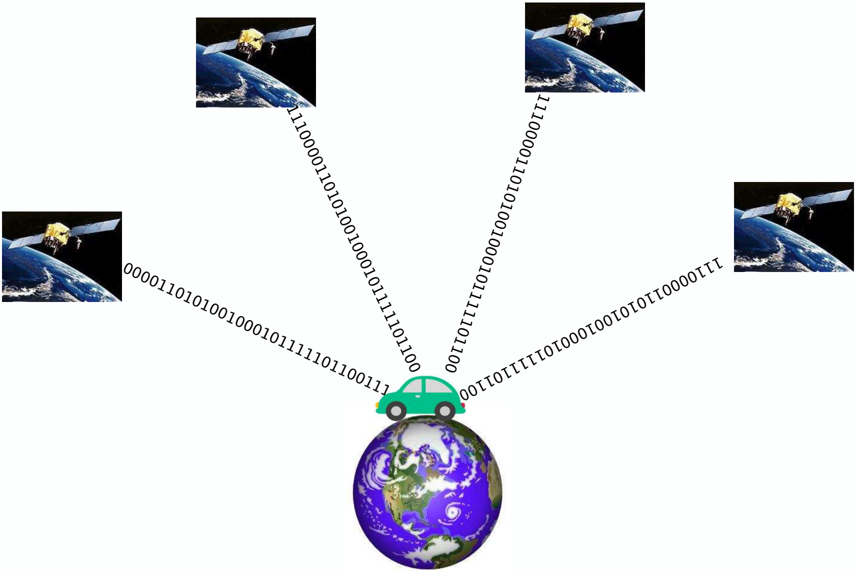
- Special Case: $d = 1, n = 2^k$
  - Order-k de Bruijn sequence (1946)

$$\begin{aligned} k = 2: &\qquad 0\,0\,1\,1 \\ k = 3: &\qquad 0\,0\,0\,1\,0\,1\,1\,1 \end{aligned}$$

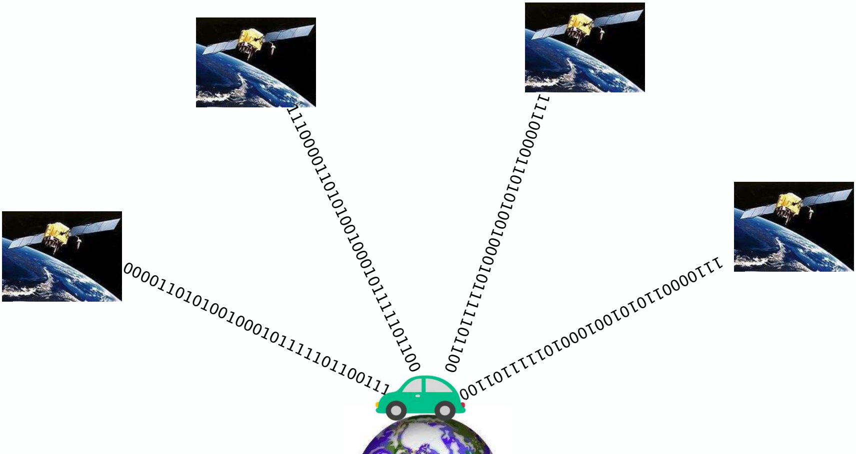
  - Diaconis' mind reader (Diaconis–Graham 2011)
    - ★ Order-5 de Bruijn sequence
    - ★ Top-10 mathematical card trick of the century (Ron Graham)



Motivation: Fast positioning

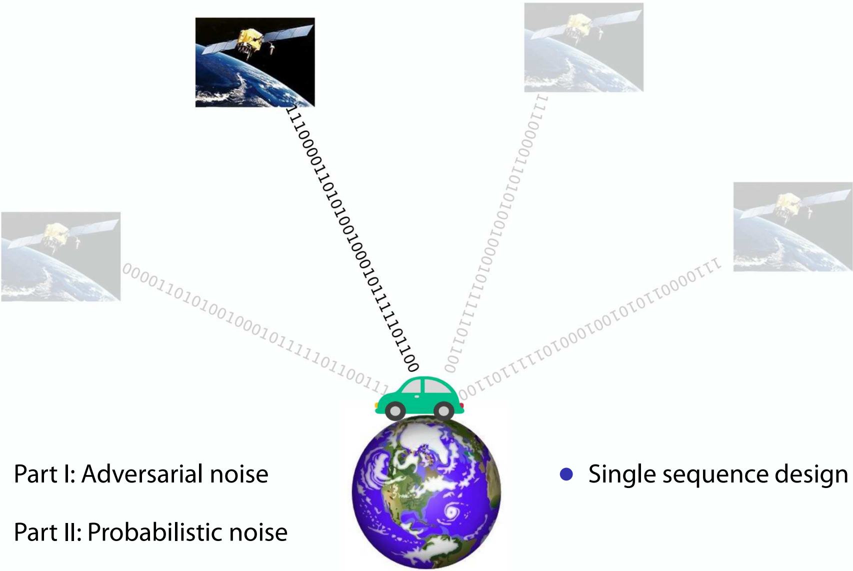


Motivation: Fast positioning



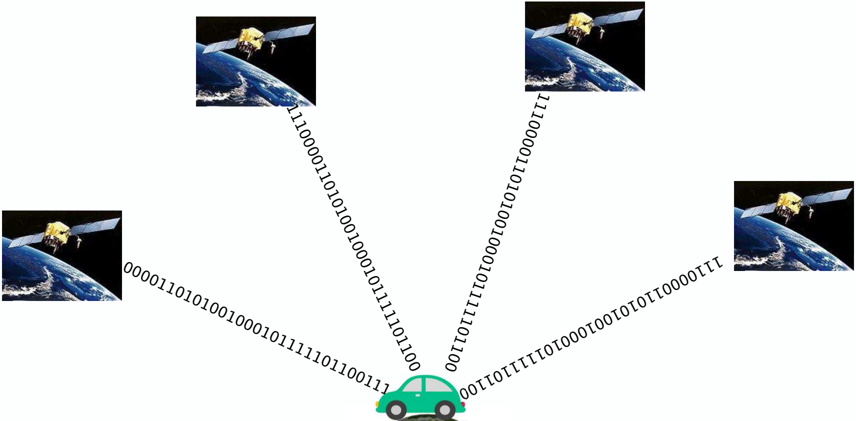
- Part I: Adversarial noise
- Part II: Probabilistic noise

Motivation: Fast positioning



- Part I: Adversarial noise
- Part II: Probabilistic noise
- Single sequence design

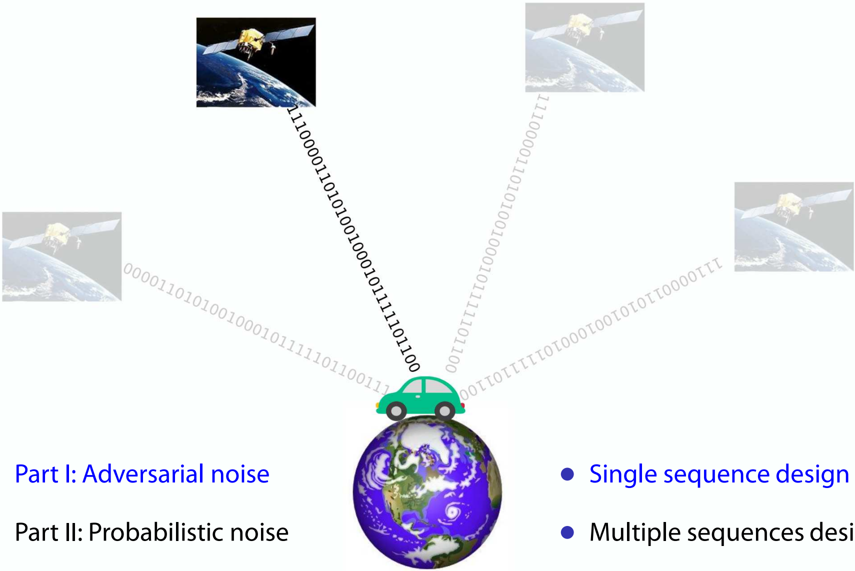
Motivation: Fast positioning



- Part I: Adversarial noise
- Part II: Probabilistic noise

- Single sequence design
- Multiple sequences design

Motivation: Fast positioning

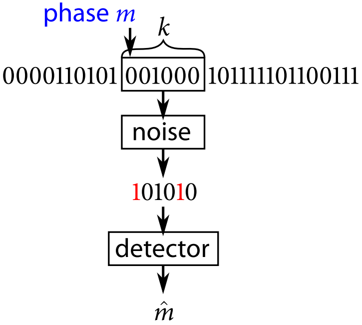


- Part I: Adversarial noise
- Part II: Probabilistic noise

- Single sequence design
- Multiple sequences design

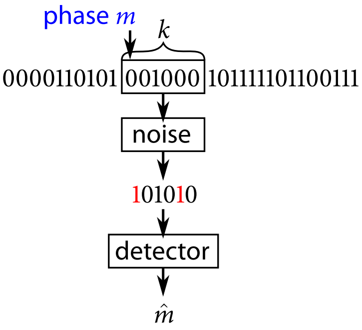
Problem setup

- An (n, k) phase detection scheme
  - a binary sequence x^n
  - a detector $\hat{m} \colon \{0,1\}^k \to [n]$



Problem setup

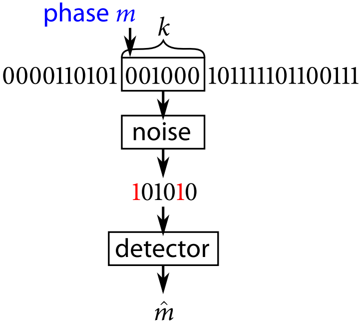
- An (n, k) phase detection scheme
  - a binary sequence x^n
  - a detector $\hat{m}\colon \{0,1\}^k \to [n]$



- Induced codebook $\mathcal{C} = \{$length-k chunks$\}$

Problem setup

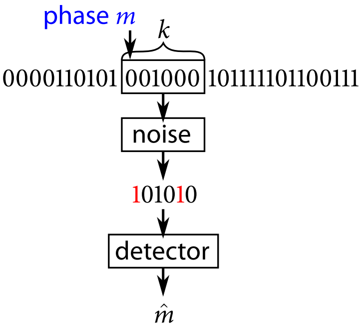
- An (n, k) phase detection scheme
  - a binary sequence x^n
  - a detector $\hat{m}\colon \{0,1\}^k \to [n]$



- Induced codebook $\mathcal{C} = \{$length-k chunks$\}$
- Induced minimum distance d

Problem setup

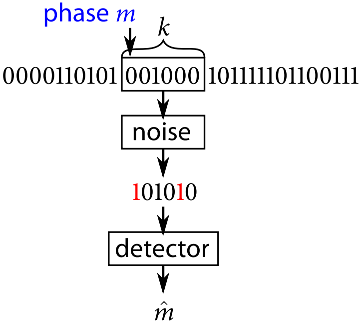
- An (n,k) phase detection scheme
  - a binary sequence x^n
  - a detector $\hat{m}\colon \{0,1\}^k \to [n]$



- Induced codebook $\mathcal{C} = \{$length-k chunks$\}$
- Induced minimum distance d
- For reliable detection, $k \geq \log n$

Problem setup

- An (n,k) phase detection scheme
  - a binary sequence x^n
  - a detector $\hat{m}\colon \{0,1\}^k \to [n]$

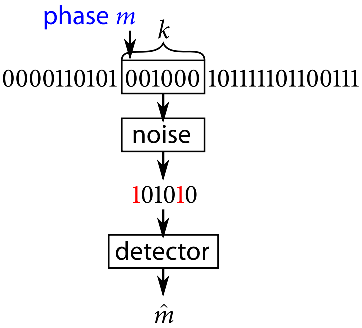


- Induced codebook $\mathcal{C} = \{$length-k chunks$\}$
- Induced minimum distance d
- For reliable detection, $k \geq \log n$
- Rate

$$R \triangleq \frac{\log n}{k}$$

Problem setup

- An (n, k) phase detection scheme
  - a binary sequence x^n
  - a detector $\hat{m} \colon \{0, 1\}^k \to [n]$



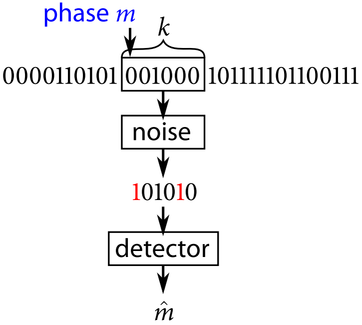
- Induced codebook $\mathcal{C} = \{$length-k chunks$\}$
- Induced minimum distance d
- For reliable detection, $k \geq \log n$
- Rate

$$R \triangleq \frac{\log n}{k}$$

(same as the rate of the induced codebook)

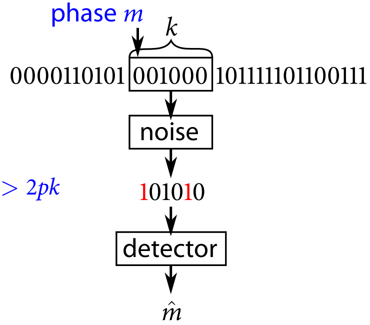
Noise model

- Adversarial noise: Up to pk bits are flipped



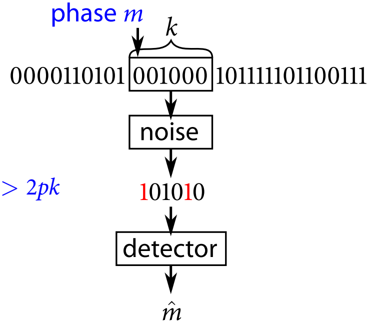
Noise model

- Adversarial noise: Up to pk bits are flipped
  - R achievable if $\exists$ a seq. $(2^{kR}, k)$ schemes with $d > 2pk$

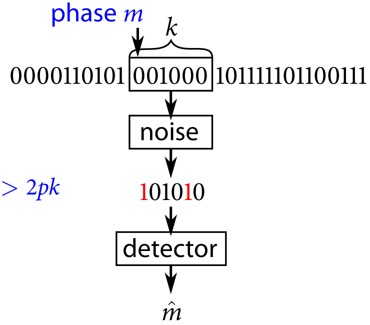


Noise model

- Adversarial noise: Up to pk bits are flipped
  - R achievable if $\exists$ a seq. $(2^{kR}, k)$ schemes with $d > 2pk$
  - Capacity $C_{\mathsf{advs}}(p)$



Noise model



- Adversarial noise: Up to pk bits are flipped
  - R achievable if $\exists$ a seq. $(2^{kR}, k)$ schemes with $d > 2pk$
  - Capacity $C_{\mathsf{advs}}(p)$

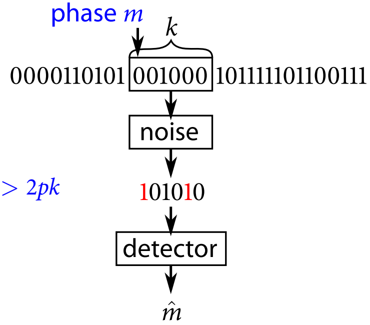
Relation to channel coding

Phase detection scheme

Error-correcting channel code

Noise model

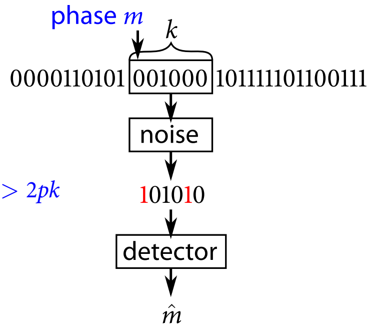
- Adversarial noise: Up to pk bits are flipped
  - R achievable if $\exists$ a seq. $(2^{kR}, k)$ schemes with $d > 2pk$
  - Capacity $C_{\mathsf{advs}}(p)$



Relation to channel coding

Phase detection scheme $\Longrightarrow$ Error-correcting channel code

Noise model



- Adversarial noise: Up to pk bits are flipped
  - R achievable if $\exists$ a seq. $(2^{kR}, k)$ schemes with $d > 2pk$
  - Capacity $C_{\mathrm{advs}}(p)$

Relation to channel coding

Phase detection scheme $\Longrightarrow$ Error-correcting channel code

Phase detection scheme $\overset{?}{\Longleftarrow}$ Error-correcting channel code

Existing results

- Trade-off btw R and d for m-sequences
  - Kumar–Wei 1992
  - Hagita–Matsumoto–Natsu–Ohtsuka 2008

Existing results

- Trade-off btw R and d for m-sequences
  - Kumar–Wei 1992
  - Hagita–Matsumoto–Natsu–Ohtsuka 2008
- Sequence constructions
  - Krishnamachari–Yedavalli 2007
  - Horvath–Herout–Szentandrasi–Zacharias 2013
  - Jorissen–Maesen–Doshi–Bekaert 2014
  - Berkowitz–Kopparty 2016

Existing results

- Trade-off btw R and d for m-sequences
  - Kumar–Wei 1992
  - Hagita–Matsumoto–Natsu–Ohtsuka 2008
- Sequence constructions
  - Krishnamachari–Yedavalli 2007
  - Horvath–Herout–Szentandrasi–Zacharias 2013
  - Jorissen–Maesen–Doshi–Bekaert 2014
  - Berkowitz–Kopparty 2016
- Generalization to 2-D positioning
  - MacWilliams–Sloane 1976
  - Etzion 1988
  - Paterson 1994
  - Bruckstein–Etzion–Giryes–Gordon–Holt 2012

Existing results

- Trade-off btw R and d for m-sequences
  - Kumar–Wei 1992
  - Hagita–Matsumoto–Natsu–Ohtsuka 2008
- Sequence constructions
  - Krishnamachari–Yedavalli 2007
  - Horvath–Herout–Szentandrasi–Zacharias 2013
  - Jorissen–Maesen–Doshi–Bekaert 2014
  - Berkowitz–Kopparty 2016
- Generalization to 2-D positioning
  - MacWilliams–Sloane 1976
  - Etzion 1988
  - Paterson 1994
  - Bruckstein–Etzion–Giryes–Gordon–Holt 2012
- Our focus: 1-D positioning

Existing results

- Trade-off btw R and d for m-sequences
  - Kumar–Wei 1992
  - Hagita–Matsumoto–Natsu–Ohtsuka 2008
- Sequence constructions
  - Krishnamachari–Yedavalli 2007
  - Horvath–Herout–Szentandrasi–Zacharias 2013
  - Jorissen–Maesen–Doshi–Bekaert 2014
  - Berkowitz–Kopparty 2016
- Generalization to 2-D positioning
  - MacWilliams–Sloane 1976
  - Etzion 1988
  - Paterson 1994
  - Bruckstein–Etzion–Giryes–Gordon–Holt 2012
- Our focus: 1-D positioning
  - Trade-off btw R and d in the asymptotic limit (bounds on $C_{\text{advs}}(p)$)

Existing results

- Trade-off btw R and d for m-sequences
  - Kumar–Wei 1992
  - Hagita–Matsumoto–Natsu–Ohtsuka 2008
- Sequence constructions
  - Krishnamachari–Yedavalli 2007
  - Horvath–Herout–Szentandrasi–Zacharias 2013
  - Jorissen–Maesen–Doshi–Bekaert 2014
  - Berkowitz–Kopparty 2016
- Generalization to 2-D positioning
  - MacWilliams–Sloane 1976
  - Etzion 1988
  - Paterson 1994
  - Bruckstein–Etzion–Giryes–Gordon–Holt 2012
- Our focus: 1-D positioning
  - Trade-off btw R and d in the asymptotic limit (bounds on $C_{\mathsf{advs}}(p)$)
  - When can we chain up an error-correcting code?

Fundamental limit

- $C_{\mathsf{advs}}(p) \leq$ any upper bound on adversarial channel codes

Fundamental limit

- $C_{\mathsf{advs}}(p) \le$ any upper bound on adversarial channel codes
- Does it achieve the best known Gilbert–Varshamov (GV) lower bound?

Fundamental limit

- $C_{\mathsf{advs}}(p) \leq$ any upper bound on adversarial channel codes
- Does it achieve the best known Gilbert–Varshamov (GV) lower bound?

Theorem (Achieves the GV bound)

$$C_{\mathsf{advs}}(p) \geq 1 - h(2p)$$

Fundamental limit

- $C_{\mathsf{advs}}(p) \leq$ any upper bound on adversarial channel codes
- Does it achieve the best known Gilbert–Varshamov (GV) lower bound?

Theorem (Achieves the GV bound)

$$C_{\mathsf{advs}}(p) \geq 1 - h(2p)$$

- Proof sketch:
  - Probabilistic method: X^n i.i.d. uniform
  - Challenge: Dependence between overlapping chunks

Fundamental limit

- $C_{\mathsf{advs}}(p) \leq$ any upper bound on adversarial channel codes
- Does it achieve the best known Gilbert–Varshamov (GV) lower bound?

Theorem (Achieves the GV bound)

$$C_{\mathsf{advs}}(p) \geq 1 - h(2p)$$

- Proof sketch:
  - Probabilistic method: X^n i.i.d. uniform
  - Challenge: Dependence between overlapping chunks
  - Solution: Lovász Local Lemma (1975)

Fundamental limit

- $C_{\text{advs}}(p) \leq$ any upper bound on adversarial channel codes
- Does it achieve the best known Gilbert–Varshamov (GV) lower bound?

Theorem (Achieves the GV bound)

$$C_{\text{advs}}(p) \geq 1 - h(2p)$$

- Proof sketch:
  - Probabilistic method: X^n i.i.d. uniform
  - Challenge: Dependence between overlapping chunks
  - Solution: Lovász Local Lemma (1975)
- Good sequence exists, but with vanishingly small probability

Linear phase detection schemes

- When can we chain up an error-correcting code?

Linear phase detection schemes

- When can we chain up an error-correcting code?
- We call a phase detection scheme linear if $\mathcal{C} \cup \{0^k\}$ is a linear code

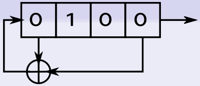
Linear phase detection schemes

- When can we chain up an error-correcting code?
- We call a phase detection scheme linear if $\mathcal{C} \cup \{0^k\}$ is a linear code

Theorem

A phase detection scheme is linear if and only if

- it is generated by a linear feedback shift register
- with a primitive connection polynomial (m-sequence)



$x_j = \sum_{i=1}^{r} a_i x_{j-i}$

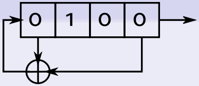
Linear phase detection schemes

- When can we chain up an error-correcting code?
- We call a phase detection scheme linear if $\mathcal{C} \cup \{0^k\}$ is a linear code

Theorem

A phase detection scheme is linear if and only if

- it is generated by a linear feedback shift register
- with a primitive connection polynomial (m-sequence)



$$x_j = \sum_{i=1}^{r} a_i x_{j-i}$$

Corollary

A linear code can be chained up if and only if

- any r consecutive columns are linearly independent
- $\mathbf{g}_j = \sum_{i=1}^{r} a_i \mathbf{g}_{j-i}, \quad r+1 \le j \le k$

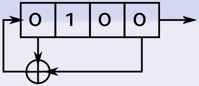
Linear phase detection schemes

- When can we chain up an error-correcting code?
- We call a phase detection scheme linear if $\mathcal{C} \cup \{0^k\}$ is a linear code

Theorem

A phase detection scheme is linear if and only if

- it is generated by a linear feedback shift register
- with a primitive connection polynomial (m-sequence)



$$x_j = \sum_{i=1}^{r} a_i x_{j-i}$$

Corollary

A linear code can be chained up if and only if

- any r consecutive columns are linearly independent
- $\mathbf{g}_j = \sum_{i=1}^{r} a_i \mathbf{g}_{j-i}, \quad r+1 \le j \le k$

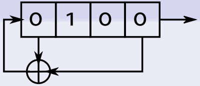
$$G_{r\times k} = \begin{bmatrix} | & | & | & & | & | & | & & | \\ \mathbf{g}_1 & \mathbf{g}_2 & \mathbf{g}_3 & \cdots & \mathbf{g}_r & \mathbf{g}_{r+1} & \mathbf{g}_{r+2} & \cdots & \mathbf{g}_k \\ | & | & | & & | & | & | & & | \end{bmatrix}$$

- When can we chain up an error-correcting code?
- We call a phase detection scheme linear if $\mathcal{C} \cup \{0^k\}$ is a linear code

Theorem

A phase detection scheme is linear if and only if

- it is generated by a linear feedback shift register
- with a primitive connection polynomial (m-sequence)



$x_j = \sum_{i=1}^{r} a_i x_{j-i}$

Corollary

A linear code can be chained up if and only if

- any r consecutive columns are linearly independent
- $\mathbf{g}_j = \sum_{i=1}^{r} a_i \mathbf{g}_{j-i}, \quad r+1 \le j \le k$

$$G_{r\times k} = \left[\boxed{\begin{array}{ccccc} | & | & | & & | \\ \mathbf{g}_1 & \mathbf{g}_2 & \mathbf{g}_3 & \cdots & \mathbf{g}_r \\ | & | & | & & | \end{array}} \begin{array}{cccc} | & | & & | \\ \mathbf{g}_{r+1} & \mathbf{g}_{r+2} & \cdots & \mathbf{g}_k \\ | & | & & | \end{array}\right]$$

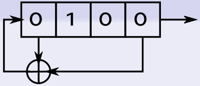
Linear phase detection schemes

- When can we chain up an error-correcting code?
- We call a phase detection scheme linear if $\mathcal{C} \cup \{0^k\}$ is a linear code

Theorem

A phase detection scheme is linear if and only if

- it is generated by a linear feedback shift register
- with a primitive connection polynomial (m-sequence)



$$x_j = \sum_{i=1}^{r} a_i x_{j-i}$$

Corollary

A linear code can be chained up if and only if

- any r consecutive columns are linearly independent
- $\mathbf{g}_j = \sum_{i=1}^{r} a_i \mathbf{g}_{j-i}, \quad r+1 \le j \le k$

$$G_{r\times k} = \begin{bmatrix} | & \boxed{\begin{matrix} | & | & & | & | \\ \mathbf{g}_2 & \mathbf{g}_3 & \cdots & \mathbf{g}_r & \mathbf{g}_{r+1} \\ | & | & & | & | \end{matrix}} & | & & | \\ \mathbf{g}_1 & & \mathbf{g}_{r+2} & \cdots & \mathbf{g}_k \\ | & & | & & | \end{bmatrix}$$

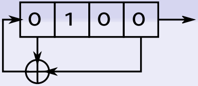
Linear phase detection schemes

- When can we chain up an error-correcting code?
- We call a phase detection scheme linear if $\mathcal{C} \cup \{0^k\}$ is a linear code

Theorem

A phase detection scheme is linear if and only if

- it is generated by a linear feedback shift register
- with a primitive connection polynomial (m-sequence)



$$x_j = \sum_{i=1}^{r} a_i x_{j-i}$$

Corollary

A linear code can be chained up if and only if

- any r consecutive columns are linearly independent
- $\mathbf{g}_j = \sum_{i=1}^{r} a_i \mathbf{g}_{j-i}, \quad r+1 \le j \le k$

$$G_{r\times k} = \begin{bmatrix} | & | & | & & | & | & | & & | \\ \mathbf{g}_1 & \mathbf{g}_2 & \mathbf{g}_3 & \cdots & \mathbf{g}_r & \mathbf{g}_{r+1} & \mathbf{g}_{r+2} & \cdots & \mathbf{g}_k \\ | & | & | & & | & | & | & & | \end{bmatrix}$$

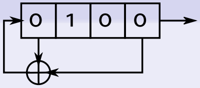
Linear phase detection schemes

- When can we chain up an error-correcting code?
- We call a phase detection scheme linear if $\mathcal{C} \cup \{0^k\}$ is a linear code

Theorem

A phase detection scheme is linear if and only if

- it is generated by a linear feedback shift register
- with a primitive connection polynomial (m-sequence)



$$x_j = \sum_{i=1}^{r} a_i x_{j-i}$$

Corollary

A linear code can be chained up if and only if

- any r consecutive columns are linearly independent
- $\mathbf{g}_j = \sum_{i=1}^{r} a_i \mathbf{g}_{j-i}, \quad r+1 \leq j \leq k$

$$G_{r\times k} = \begin{bmatrix} | & | & | & & | & | & | & & | \\ \mathbf{g}_1 & \mathbf{g}_2 & \mathbf{g}_3 & \cdots & \mathbf{g}_r & \mathbf{g}_{r+1} & \mathbf{g}_{r+2} & \cdots & \mathbf{g}_k \\ | & | & | & & | & | & | & & | \end{bmatrix}$$

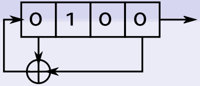
Linear phase detection schemes

- When can we chain up an error-correcting code?
- We call a phase detection scheme linear if $\mathcal{C} \cup \{0^k\}$ is a linear code

Theorem

A phase detection scheme is linear if and only if

- it is generated by a linear feedback shift register
- with a primitive connection polynomial (m-sequence)



$x_j = \sum_{i=1}^r a_i x_{j-i}$

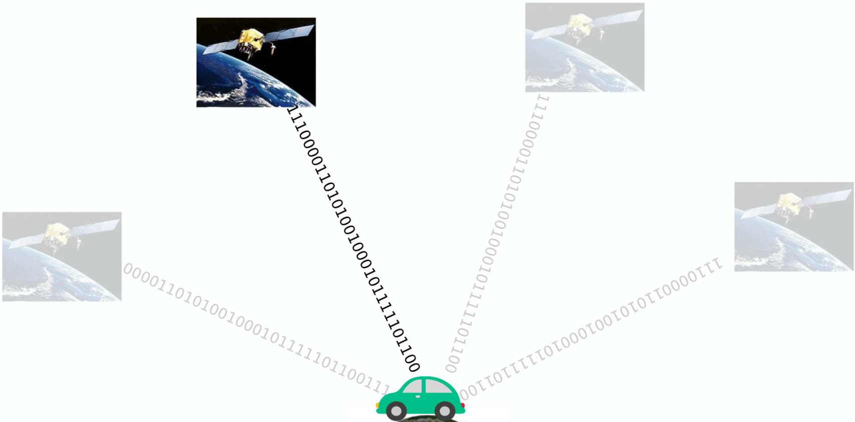
Corollary

A linear code can be chained up if and only if

- any r consecutive columns are linearly independent
- $\mathbf{g}_j = \sum_{i=1}^r a_i \mathbf{g}_{j-i}, \quad r+1 \le j \le k$

$$G_{r\times k} = \begin{bmatrix} 1 & 0 & \cdots & 0 & a_r & | & & | \\ \vdots & \ddots & \ddots & \vdots & \vdots & \mathbf{g}_{r+2} & \cdots & \mathbf{g}_k \\ 0 & \cdots & 1 & 0 & a_2 & & & \\ 0 & \cdots & 0 & 1 & a_1 & | & & | \end{bmatrix}$$

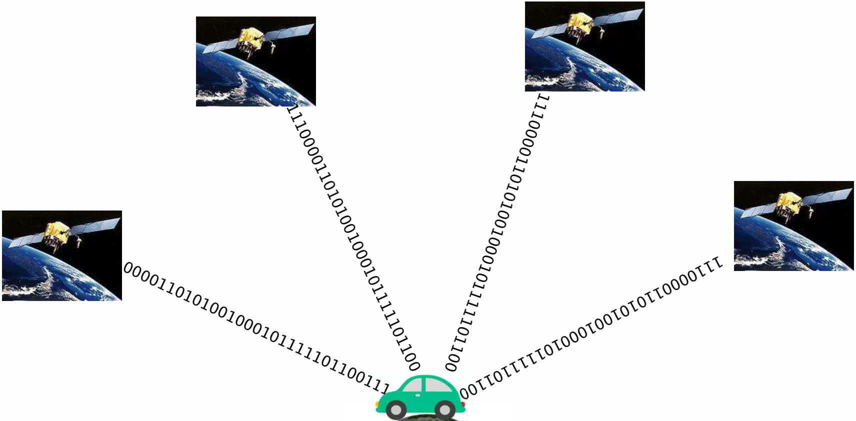
Outline of the talk



- Part I: Adversarial noise
- Part II: Probabilistic noise

- Single sequence design
- Multiple sequences design

Outline of the talk



- Part I: Adversarial noise
- Part II: Probabilistic noise

- Single sequence design
- Multiple sequences design

Multiple access phase detection

phase m_1

$x_1^{n_1}$: $0000110101\,\overbrace{\boxed{0010010}}^{k}\,101111101100111$

phase m_2 $\bigoplus$

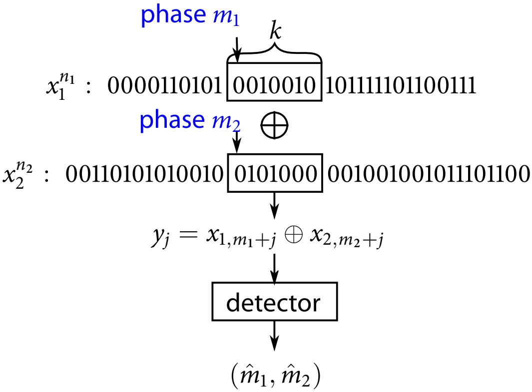
$x_2^{n_2}$: $00110101010010\,\boxed{0101000}\,001001001011101100$

$$y_j = x_{1,m_1+j} \oplus x_{2,m_2+j}$$

$\boxed{\text{detector}}$

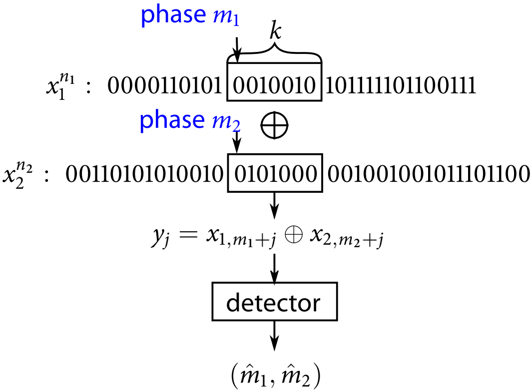
$$(\hat{m}_1, \hat{m}_2)$$

Multiple access phase detection



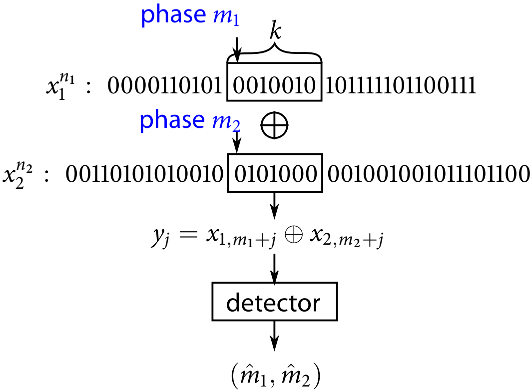
- $\mathcal{C}_1 = \{$ length-k chunks of $x_1^{n_1}\}$, $\mathcal{C}_2 = \{$ length-k chunks of $x_2^{n_2}\}$

Multiple access phase detection



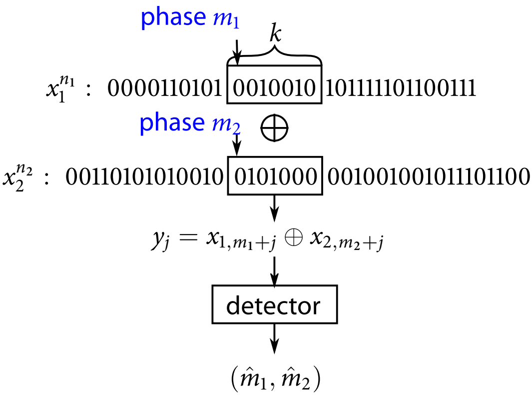
- $\mathcal{C}_1 = \{$ length-k chunks of $x_1^{n_1}\}$, $\mathcal{C}_2 = \{$ length-k chunks of $x_2^{n_2}\}$
- $\mathcal{C}_{\text{sum}} = \{\mathbf{c}_1 \oplus \mathbf{c}_2 : \mathbf{c}_1 \in \mathcal{C}_1, \mathbf{c}_2 \in \mathcal{C}_2\}$

Multiple access phase detection



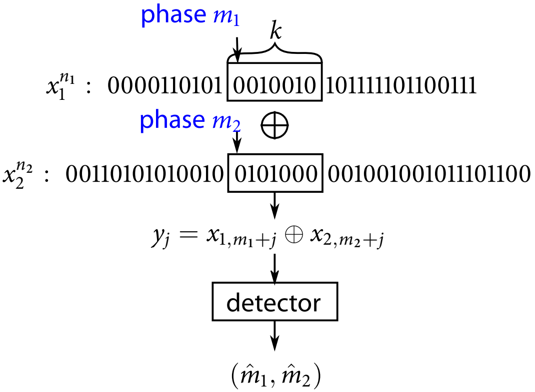
- $\mathcal{C}_1 = \{$ length-k chunks of $x_1^{n_1}\}$, $\mathcal{C}_2 = \{$ length-k chunks of $x_2^{n_2}\}$
- $\mathcal{C}_{\mathsf{sum}} = \{\mathbf{c}_1 \oplus \mathbf{c}_2 : \mathbf{c}_1 \in \mathcal{C}_1, \mathbf{c}_2 \in \mathcal{C}_2\}$
- Rates: $R_1 = \frac{\log n_1}{k}$, $R_2 = \frac{\log n_2}{k}$

Multiple access phase detection



- $\mathcal{C}_1 = \{$ length-k chunks of $x_1^{n_1}\}$, $\mathcal{C}_2 = \{$ length-k chunks of $x_2^{n_2}\}$
- $\mathcal{C}_{\text{sum}} = \{\mathbf{c}_1 \oplus \mathbf{c}_2 : \mathbf{c}_1 \in \mathcal{C}_1, \mathbf{c}_2 \in \mathcal{C}_2\}$
- Rates: $R_1 = \frac{\log n_1}{k}$, $R_2 = \frac{\log n_2}{k}$
- (R_1, R_2) achievable if each $\mathbf{c} \in \mathcal{C}_{\text{sum}}$ is uniquely expressed as $\mathbf{c}_1 \oplus \mathbf{c}_2$

Multiple access phase detection



- $\mathcal{C}_1 = \{$ length-k chunks of $x_1^{n_1}\}$, $\mathcal{C}_2 = \{$ length-k chunks of $x_2^{n_2}\}$
- $\mathcal{C}_{\text{sum}} = \{\mathbf{c}_1 \oplus \mathbf{c}_2 : \mathbf{c}_1 \in \mathcal{C}_1, \mathbf{c}_2 \in \mathcal{C}_2\}$
- Rates: $R_1 = \frac{\log n_1}{k}$, $R_2 = \frac{\log n_2}{k}$
- (R_1, R_2) achievable if each $\mathbf{c} \in \mathcal{C}_{\text{sum}}$ is uniquely expressed as $\mathbf{c}_1 \oplus \mathbf{c}_2$
- Capacity region $\mathscr{C}_{\text{advs}}$

Optimal sequence construction

- Outer bound: $R_1 + R_2 \leq 1$

Optimal sequence construction

- Outer bound: $R_1 + R_2 \leq 1$
- Attempt 1
  - Seq 1: i.i.d. uniform
  - Seq 2: i.i.d. uniform

Optimal sequence construction

- Outer bound: $R_1 + R_2 \leq 1$
- Attempt 1
  - Seq 1: i.i.d. uniform
  - Seq 2: i.i.d. uniform
  - $(R_1, R_2) = (1/3, 1/3)$

Optimal sequence construction

- Outer bound: $R_1 + R_2 \leq 1$
- Attempt 1
  - Seq 1: i.i.d. uniform
  - Seq 2: i.i.d. uniform
  - $(R_1, R_2) = (1/3, 1/3)$
- Attempt 2
  - Seq 1: de Bruijn seq. of weight $\leq pk$ (Sawada–Williams–Wong 2014)
  - Seq 2: phase detection seq. with $d > 2pk$

Optimal sequence construction

- Outer bound: $R_1 + R_2 \leq 1$
- Attempt 1
  - Seq 1: i.i.d. uniform
  - Seq 2: i.i.d. uniform
  - $(R_1, R_2) = (1/3, 1/3)$
- Attempt 2
  - Seq 1: de Bruijn seq. of weight $\leq pk$ (Sawada–Williams–Wong 2014)
  - Seq 2: phase detection seq. with $d > 2pk$
  - $R_1 < h(p), R_2 < 1 - h(2p)$

Optimal sequence construction

- Outer bound: $R_1 + R_2 \leq 1$
- Attempt 1
  - Seq 1: i.i.d. uniform
  - Seq 2: i.i.d. uniform
  - $(R_1, R_2) = (1/3, 1/3)$
- Attempt 2
  - Seq 1: de Bruijn seq. of weight $\leq pk$ (Sawada–Williams–Wong 2014)
  - Seq 2: phase detection seq. with $d > 2pk$
  - $R_1 < h(p), R_2 < 1 - h(2p)$
- Attempt 3
  - Seq 1: m-sequence (linear code $\mathcal{C}_1$)
  - Seq 2: one codeword from each coset of $\mathcal{C}_1$
  - $R_1 + R_2 = 1$

Optimal sequence construction

- Outer bound: $R_1 + R_2 \leq 1$
- Attempt 1
  - Seq 1: i.i.d. uniform
  - Seq 2: i.i.d. uniform
  - $(R_1, R_2) = (1/3, 1/3)$
- Attempt 2
  - Seq 1: de Bruijn seq. of weight $\leq pk$ (Sawada–Williams–Wong 2014)
  - Seq 2: phase detection seq. with $d > 2pk$
  - $R_1 < h(p), R_2 < 1 - h(2p)$
- Attempt 3
  - Seq 1: m-sequence (linear code $\mathcal{C}_1$)
  - Seq 2: one codeword from each coset of $\mathcal{C}_1$ (Lovász local lemma)
  - $R_1 + R_2 = 1$

Optimal sequence construction

- Outer bound: $R_1 + R_2 \leq 1$
- Attempt 1
  - Seq 1: i.i.d. uniform
  - Seq 2: i.i.d. uniform
  - $(R_1, R_2) = (1/3, 1/3)$
- Attempt 2
  - Seq 1: de Bruijn seq. of weight $\leq pk$ (Sawada–Williams–Wong 2014)
  - Seq 2: phase detection seq. with $d > 2pk$
  - $R_1 < h(p), R_2 < 1 - h(2p)$
- Attempt 3
  - Seq 1: m-sequence (linear code $\mathcal{C}_1$)
  - Seq 2: one codeword from each coset of $\mathcal{C}_1$ (Lovász local lemma)
  - $R_1 + R_2 = 1$

Theorem

$\mathscr{C}_{\text{advs}}$ is the set of (R_1, R_2) s.t.

$$R_1 + R_2 \leq 1.$$

Optimal sequence construction

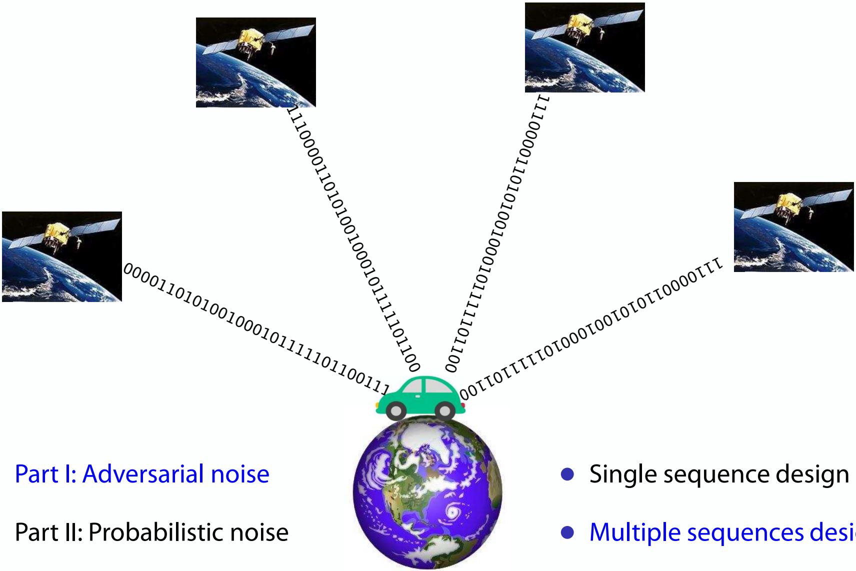
- Outer bound: $R_1 + R_2 \le 1$
- Attempt 1
  - Seq 1: i.i.d. uniform
  - Seq 2: i.i.d. uniform
  - $(R_1, R_2) = (1/3, 1/3)$
- Attempt 2
  - Seq 1: de Bruijn seq. of weight $\le pk$ (Sawada–Williams–Wong 2014)
  - Seq 2: phase detection seq. with $d > 2pk$
  - $R_1 < h(p), R_2 < 1 - h(2p)$
- Attempt 3
  - Seq 1: m-sequence (linear code $\mathcal{C}_1$)
  - Seq 2: one codeword from each coset of $\mathcal{C}_1$ (Lovász local lemma)
  - $R_1 + R_2 = 1$
- Explicit linear construction
  - Seq 1 & 2: m-sequences of two distinct primitive polynomials
  - $k = r_1 + r_2$
  - Extension to L-satellite phase detection

Theorem

$\mathscr{C}_{\text{advs}}$ is the set of (R_1, R_2) s.t.

$$R_1 + R_2 \le 1.$$

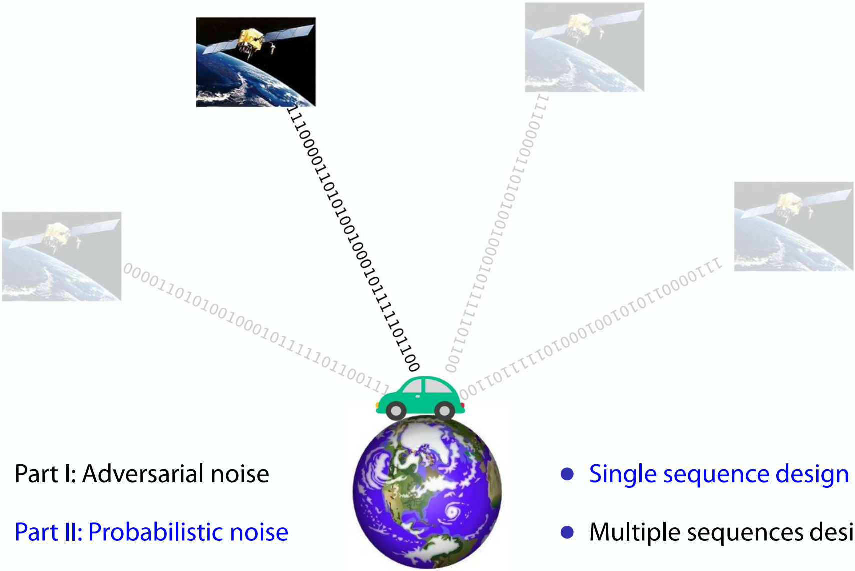
Outline of the talk



- Part I: Adversarial noise
- Part II: Probabilistic noise

- Single sequence design
- Multiple sequences design

Outline of the talk



- Part I: Adversarial noise
- Part II: Probabilistic noise

- Single sequence design
- Multiple sequences design

Probabilistic phase detection

- Probabilistic noise: i.i.d. from a channel $p(y|x)$

Probabilistic phase detection

- Probabilistic noise: i.i.d. from a channel $p(y|x)$
  - R achievable if $\exists$ a seq. of $(2^{kR}, k)$ schemes with

$$\lim_{k \to \infty} \mathsf{P}\{M \neq \hat{M}\} = 0$$

    for $M \sim \mathrm{Unif}[n]$
  - Capacity C_{prob}

Probabilistic phase detection

- Probabilistic noise: i.i.d. from a channel $p(y|x)$
  - R achievable if $\exists$ a seq. of $(2^{kR}, k)$ schemes with
    $$\lim_{k\to\infty} \mathsf{P}\{M \neq \hat{M}\} = 0$$
    for $M \sim \mathrm{Unif}[n]$
  - Capacity C_{prob}

Theorem (Achieves the Shannon capacity)

$$C_{\text{prob}} = \max_{p(x)} I(X; Y)$$

Probabilistic phase detection

- Probabilistic noise: i.i.d. from a channel $p(y|x)$
  - R achievable if $\exists$ a seq. of $(2^{kR}, k)$ schemes with

$$\lim_{k\to\infty} \mathsf{P}\{M \neq \hat{M}\} = 0$$

    for $M \sim \mathrm{Unif}[n]$
  - Capacity C_{prob}

Theorem (Achieves the Shannon capacity)

$$C_{\text{prob}} = \max_{p(x)} I(X;Y)$$

- Proof sketch:
  - Probabilistic method: X^n i.i.d.$\sim \mathrm{Bern}(1/2)$
  - Challenge: Standard packing lemma no longer works

Probabilistic phase detection

- Probabilistic noise: i.i.d. from a channel $p(y|x)$
  - R achievable if $\exists$ a seq. of $(2^{kR}, k)$ schemes with

$$\lim_{k\to\infty} \mathsf{P}\{M \neq \hat{M}\} = 0$$

    for $M \sim \mathrm{Unif}[n]$
  - Capacity C_{prob}

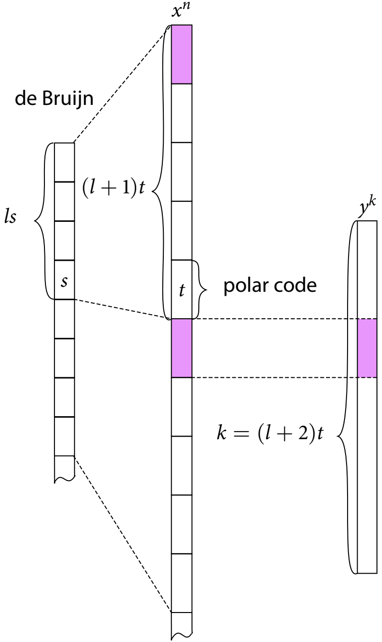
Theorem (Achieves the Shannon capacity)

$$C_{\text{prob}} = \max_{p(x)} I(X;Y)$$

- Proof sketch:
  - Probabilistic method: X^n i.i.d.$\sim \mathrm{Bern}(1/2)$
  - Challenge: Standard packing lemma no longer works
  - Solution: Lemma 24.2 (El Gamal–Kim 2011)

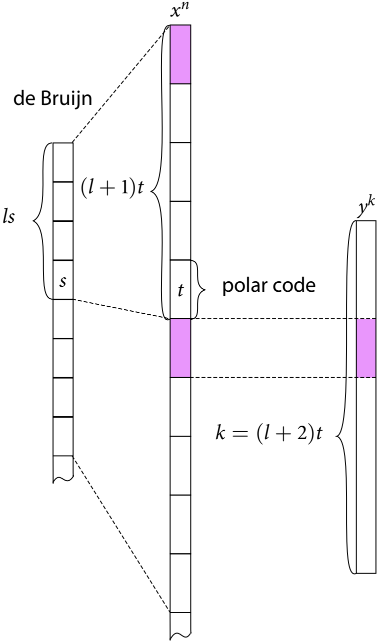
A low-complexity construction

- Capacity achieving
- $O(k \log k)$ complexity



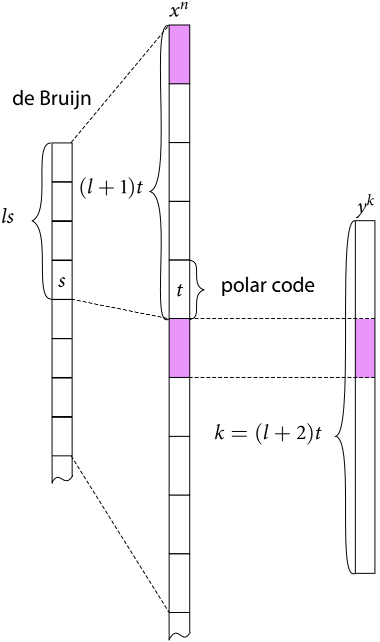
A low-complexity construction

- Capacity achieving
- $O(k \log k)$ complexity
- Main ingredients
  - de Bruijn sequence (Tuliani 2001)
  - polar code (Arıkan 2009)
  - synchronization sequence

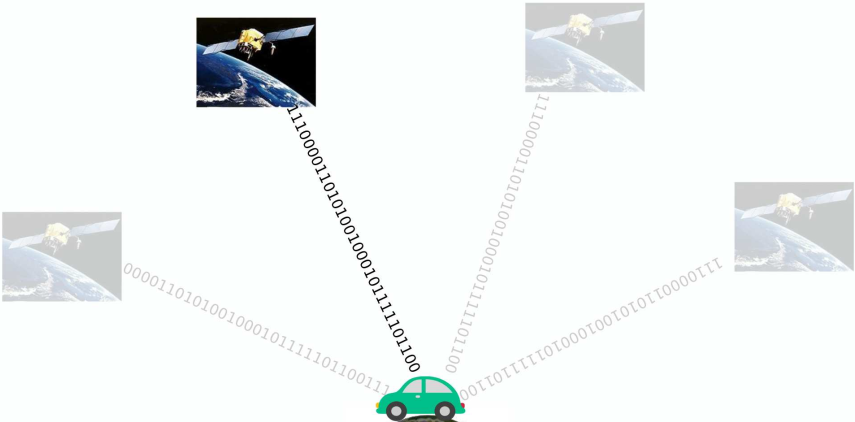


A low-complexity construction

- Capacity achieving
- $O(k \log k)$ complexity
- Main ingredients
  - de Bruijn sequence (Tuliani 2001)
  - polar code (Arıkan 2009)
  - synchronization sequence
- “Equivalence” to channel coding



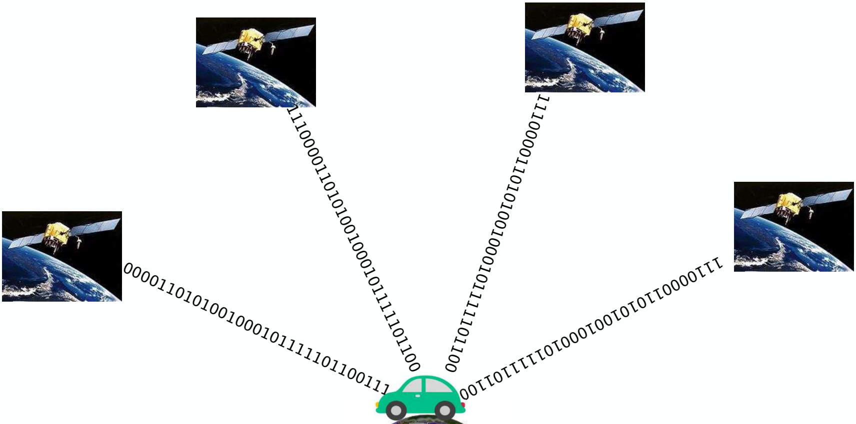
Outline of the talk



- Part I: Adversarial noise
- Part II: Probabilistic noise

- Single sequence design
- Multiple sequences design

Outline of the talk



- Part I: Adversarial noise
- Part II: Probabilistic noise

- Single sequence design
- Multiple sequences design

Problem setup

phase m_1 $\quad$ k

$x_1^{n_1}$: $0000110101\,\boxed{\overbrace{0010010}^{k}}\,101111101100111$

phase m_2

$x_2^{n_2}$: $00110101010010\,\boxed{0101000}\,001001001011101100$

y^k: output of $p(y|x_1, x_2)$

$\downarrow$

$\boxed{\text{detector}}$

$\downarrow$

$(\hat{m}_1, \hat{m}_2)$

- Rates $R_1 = \frac{\log n_1}{k}$, $R_2 = \frac{\log n_1}{k}$
- (R_1, R_2) achievable if for $(M_1, M_2) \sim \mathrm{Unif}([n_1] \times [n_2])$, $\exists$ a seq. of schemes with

$$\lim_{k \to \infty} \mathsf{P}\{(\hat{M}_1, \hat{M}_2) \neq (M_1, M_2)\} = 0$$

- Capacity region $\mathscr{C}_{\text{prob}}$

Fundamental limit

- Existing technique: Gold code in GPS $n_1 = n_2 = k$
  - Good autocorrelation and cross-correlation
  - Minimizes noise sensitivity at the lost of zero rates

Fundamental limit

- Existing technique: Gold code in GPS $n_1 = n_2 = k$
  - Good autocorrelation and cross-correlation
  - Minimizes noise sensitivity at the lost of zero rates
- Our focus: Fast positioning

Theorem

$\mathscr{C}_{\text{prob}}$ is the set of (R_1, R_2) s.t.

$$\begin{aligned} R_1 &< I(X_1; Y | X_2) \\ R_2 &< I(X_2; Y | X_1) \\ R_1 + R_2 &< I(X_1, X_2; Y) \end{aligned}$$

for some $p(x_1)p(x_2)$

Fundamental limit

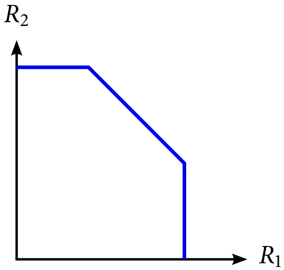
- Existing technique: Gold code in GPS $n_1 = n_2 = k$
  - Good autocorrelation and cross-correlation
  - Minimizes noise sensitivity at the lost of zero rates
- Our focus: Fast positioning

Theorem

$\mathscr{C}_{\text{prob}}$ is the set of (R_1, R_2) s.t.

$$\begin{aligned} R_1 &< I(X_1; Y | X_2) \\ R_2 &< I(X_2; Y | X_1) \\ R_1 + R_2 &< I(X_1, X_2; Y) \end{aligned}$$

for some $p(x_1)p(x_2)$



Fundamental limit

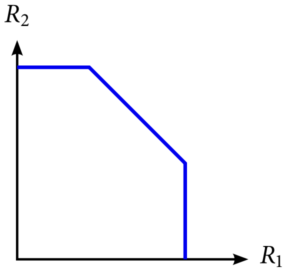
- Existing technique: Gold code in GPS $n_1 = n_2 = k$
  - Good autocorrelation and cross-correlation
  - Minimizes noise sensitivity at the lost of zero rates
- Our focus: Fast positioning

Theorem

$\mathscr{C}_{\text{prob}}$ is the set of (R_1, R_2) s.t.

$$\begin{aligned} R_1 &< I(X_1; Y | X_2) \\ R_2 &< I(X_2; Y | X_1) \\ R_1 + R_2 &< I(X_1, X_2; Y) \end{aligned}$$

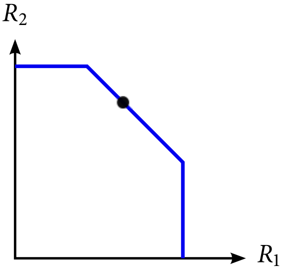
for some $p(x_1)p(x_2)$



- Remarks
  - Strictly smaller than MAC capacity region !
  - Non-convex region (lack of synchronization)

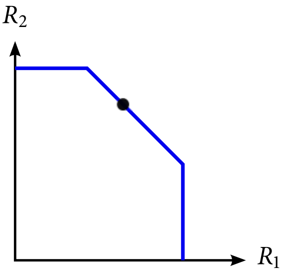
A low-complexity construction

- Achieves arbitrary point in $\mathscr{C}_{\mathsf{prob}}$
- $(k \log k)$ complexity



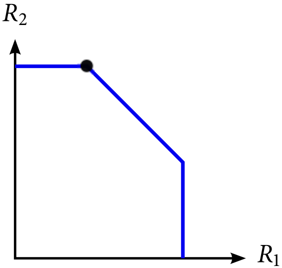
A low-complexity construction

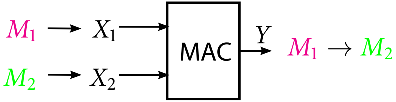
- Achieves arbitrary point in $\mathscr{C}_{\text{prob}}$
- $(k \log k)$ complexity
- Main ingredients
  - Good point-to-point phase detection sequence



A low-complexity construction

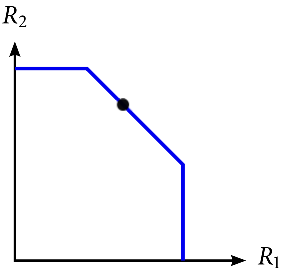
- Achieves arbitrary point in $\mathscr{C}_{\text{prob}}$
- $(k \log k)$ complexity
- Main ingredients
  - Good point-to-point phase detection sequence

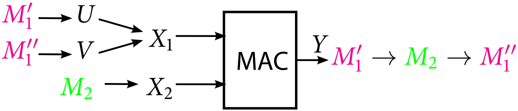




A low-complexity construction

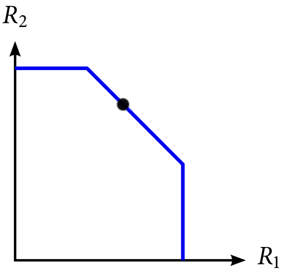
- Achieves arbitrary point in $\mathscr{C}_{\text{prob}}$
- $(k \log k)$ complexity
- Main ingredients
  - Good point-to-point phase detection sequence
  - Rate-splitting (Rimoldi–Urbanke 1996)

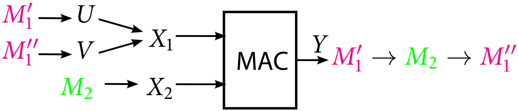




A low-complexity construction

- Achieves arbitrary point in $\mathscr{C}_{\text{prob}}$
- $(k \log k)$ complexity
- Main ingredients
  - Good point-to-point phase detection sequence
  - Rate-splitting (Rimoldi–Urbanke 1996)





  - Symbol-by-symbol mapping $x_{1m} = f(u_i, v_j)$
    with $i = m \pmod{n'}, j = m \pmod{n''}, gcd(n', n'') = 1$

Future research

- Can linear phase detection schemes achieve the GV bound?

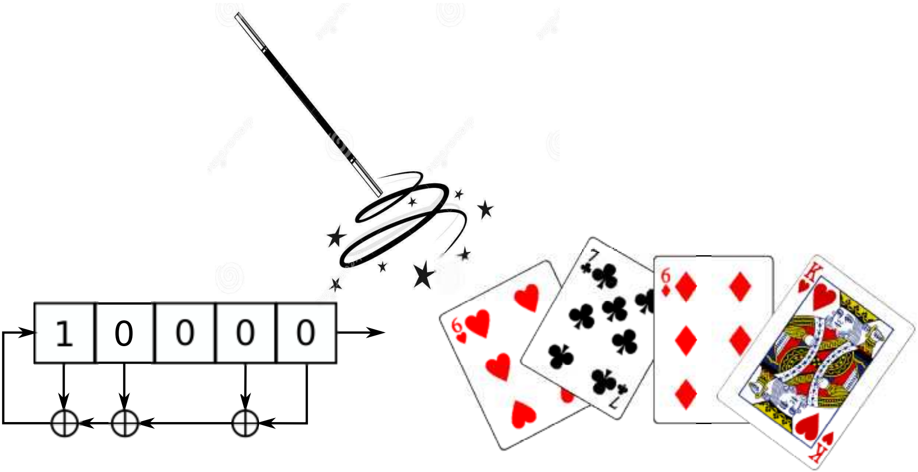
Future research

- Can linear phase detection schemes achieve the GV bound?

- Can we obtain tighter upper bound than coding theory?

Future research

- Can linear phase detection schemes achieve the GV bound?
- Can we obtain tighter upper bound than coding theory?
- Adversarial phase detection for binary adder MAC: $Y = X_1 + X_2$?


1 0 0 0 0