On the Use of Division Algebras for Wireless Communication

Frédérique Oggier
frederique@systems.caltech.edu

California Institute of Technology

AMS meeting, Davidson, March 3rd 2007

Outline

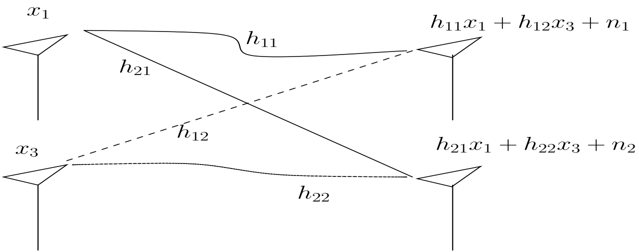
A few wireless coding problems
- Space-Time Coding
- Differential Space-Time Coding
- Distributed Space-Time Coding

Division algebras
- Introducing Division Algebras
- Codewords from Division Algebras

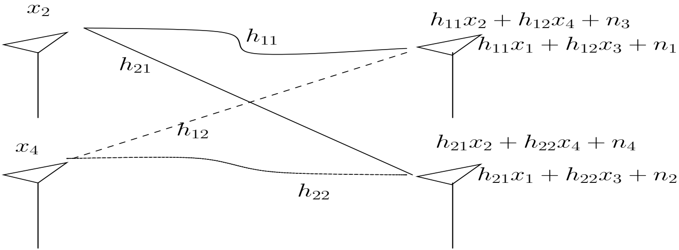
Space-Time Coding

Multiple antenna coding: the model

Multiple antenna coding: the model



Multiple antenna coding: the model



Multiple antenna coding: the coding problem

- We summarize the channel as

$$\mathbf{Y} = \begin{pmatrix} h_{11} & h_{12} \\ h_{21} & h_{22} \end{pmatrix} \underbrace{\begin{pmatrix} x_1 & x_2 \\ x_3 & x_4 \end{pmatrix}}_{\text{space-time codeword}} + \mathbf{W},\ \mathbf{W},\ \mathbf{H} \text{ complex Gaussian}$$

- The goal is the design of the codebook $\mathcal{C}$:

$$\mathcal{C} = \left\{ \mathbf{X} = \begin{pmatrix} x_1 & x_2 \\ x_3 & x_4 \end{pmatrix} | x_1, x_2, x_3, x_4 \in \mathbb{C} \right\}$$

the x_i are functions of the information symbols.

Multiple antenna coding: the coding problem

- We summarize the channel as

$$\mathbf{Y} = \begin{pmatrix} h_{11} & h_{12} \\ h_{21} & h_{22} \end{pmatrix} \underbrace{\begin{pmatrix} x_1 & x_2 \\ x_3 & x_4 \end{pmatrix}}_{\text{space-time codeword}} + \mathbf{W},\ \mathbf{W},\ \mathbf{H} \text{ complex Gaussian}$$

- The goal is the design of the codebook $\mathcal{C}$:

$$\mathcal{C} = \left\{ \mathbf{X} = \begin{pmatrix} x_1 & x_2 \\ x_3 & x_4 \end{pmatrix} | x_1, x_2, x_3, x_4 \in \mathbb{C} \right\}$$

the x_i are functions of the information symbols.

The code design

- The *pairwise probability of error* of sending $\mathbf{X}$ and decoding $\hat{\mathbf{X}} \neq \mathbf{X}$ is upper bounded by

$$P(\mathbf{X} \rightarrow \hat{\mathbf{X}}) \leq \frac{const}{|\det(\mathbf{X} - \hat{\mathbf{X}})|^{2M}},$$

where the receiver knows the channel (*coherent case*).

- Find a family $\mathcal{C}$ of $M \times M$ matrices such that

$$\det(\mathbf{X}_i - \mathbf{X}_j) \neq 0, \ \mathbf{X}_i \neq \mathbf{X}_j \in \mathcal{C},$$

called *fully-diverse*.

The code design

- The *pairwise probability of error* of sending $\mathbf{X}$ and decoding $\hat{\mathbf{X}} \neq \mathbf{X}$ is upper bounded by

$$P(\mathbf{X} \to \hat{\mathbf{X}}) \leq \frac{const}{|\det(\mathbf{X} - \hat{\mathbf{X}})|^{2M}},$$

  where the receiver knows the channel (*coherent case*).
- Find a family $\mathcal{C}$ of $M \times M$ matrices such that

$$\det(\mathbf{X}_i - \mathbf{X}_j) \neq 0, \ \mathbf{X}_i \neq \mathbf{X}_j \in \mathcal{C},$$

  called *fully-diverse*.

The differential noncoherent MIMO channel

- We assume *no channel knowledge*.
- We use *differential unitary space-time modulation*. that is (assuming $\mathbf{S}_0 = \mathbf{I}$)

$$\mathbf{S}_t = \mathbf{X}_{z_t}\mathbf{S}_{t-1},\ t = 1, 2, \ldots,$$

where $z_t \in \{0, \ldots, L-1\}$ is the data to be transmitted, and $\mathcal{C} = \{\mathbf{X}_0, \ldots, \mathbf{X}_{L-1}\}$ the constellation to be designed.

- The matrices $\mathbf{X}$ have to be *unitary*.

The differential noncoherent MIMO channel

- We assume *no channel knowledge*.
- We use *differential unitary space-time modulation*. that is (assuming $\mathbf{S}_0 = \mathbf{I}$)

$$\mathbf{S}_t = \mathbf{X}_{z_t}\mathbf{S}_{t-1},\ t = 1, 2, \ldots,$$

  where $z_t \in \{0, \ldots, L-1\}$ is the data to be transmitted, and $\mathcal{C} = \{\mathbf{X}_0, \ldots, \mathbf{X}_{L-1}\}$ the constellation to be designed.
- The matrices $\mathbf{X}$ have to be *unitary*.

The differential noncoherent MIMO channel

- We assume *no channel knowledge*.
- We use *differential unitary space-time modulation*. that is (assuming $\mathbf{S}_0 = \mathbf{I}$)

$$\mathbf{S}_t = \mathbf{X}_{z_t}\mathbf{S}_{t-1},\ t = 1, 2, \ldots,$$

  where $z_t \in \{0, \ldots, L-1\}$ is the data to be transmitted, and $\mathcal{C} = \{\mathbf{X}_0, \ldots, \mathbf{X}_{L-1}\}$ the constellation to be designed.
- The matrices $\mathbf{X}$ have to be *unitary*.

Decoding and probability of error

- If we assume the channel is roughly constant, we have

$$\begin{aligned}\mathbf{Y}_t &= \mathbf{S}_t\mathbf{H} + \mathbf{W}_t \\ &= \mathbf{X}_{z_t}\mathbf{S}_{t-1}\mathbf{H} + \mathbf{W}_t \\ &= \mathbf{X}_{z_t}(\mathbf{Y}_{t-1} - \mathbf{W}_{t-1}) + \mathbf{W}_t \\ &= \mathbf{X}_{z_t}\mathbf{Y}_{t-1} + \mathbf{W}'_t, \quad \mathbf{H} \text{ does } \textit{not} \text{ appear!}\end{aligned}$$

- The *pairwise probability of error* P_e has the upper bound

$$P_e \leq \left(\frac{1}{2}\right)\left(\frac{8}{\rho}\right)^{MN}\frac{1}{|\det(\mathbf{X}_i - \mathbf{X}_j)|^{2N}}$$

- We need to design *unitary fully-diverse* matrices.

Decoding and probability of error

- If we assume the channel is roughly constant, we have

$$\begin{aligned}\mathbf{Y}_t &= \mathbf{S}_t\mathbf{H} + \mathbf{W}_t \\ &= \mathbf{X}_{z_t}\mathbf{S}_{t-1}\mathbf{H} + \mathbf{W}_t \\ &= \mathbf{X}_{z_t}(\mathbf{Y}_{t-1} - \mathbf{W}_{t-1}) + \mathbf{W}_t \\ &= \mathbf{X}_{z_t}\mathbf{Y}_{t-1} + \mathbf{W}'_t, \quad \mathbf{H} \text{ does } \textit{not} \text{ appear!}\end{aligned}$$

- The *pairwise probability of error* P_e has the upper bound

$$P_e \leq \left(\frac{1}{2}\right)\left(\frac{8}{\rho}\right)^{MN} \frac{1}{|\det(\mathbf{X}_i - \mathbf{X}_j)|^{2N}}$$

- We need to design *unitary fully-diverse* matrices.

Decoding and probability of error

- If we assume the channel is roughly constant, we have

$$\begin{aligned}\mathbf{Y}_t &= \mathbf{S}_t\mathbf{H} + \mathbf{W}_t \\ &= \mathbf{X}_{z_t}\mathbf{S}_{t-1}\mathbf{H} + \mathbf{W}_t \\ &= \mathbf{X}_{z_t}(\mathbf{Y}_{t-1} - \mathbf{W}_{t-1}) + \mathbf{W}_t \\ &= \mathbf{X}_{z_t}\mathbf{Y}_{t-1} + \mathbf{W}'_t, \quad \mathbf{H} \text{ does } \textit{not} \text{ appear!}\end{aligned}$$

- The *pairwise probability of error* P_e has the upper bound

$$P_e \leq \left(\frac{1}{2}\right)\left(\frac{8}{\rho}\right)^{MN} \frac{1}{|\det(\mathbf{X}_i - \mathbf{X}_j)|^{2N}}$$

- We need to design *unitary fully-diverse* matrices.

Decoding and probability of error

- If we assume the channel is roughly constant, we have

$$\begin{aligned}
\mathbf{Y}_t &= \mathbf{S}_t\mathbf{H} + \mathbf{W}_t \\
&= \mathbf{X}_{z_t}\mathbf{S}_{t-1}\mathbf{H} + \mathbf{W}_t \\
&= \mathbf{X}_{z_t}(\mathbf{Y}_{t-1} - \mathbf{W}_{t-1}) + \mathbf{W}_t \\
&= \mathbf{X}_{z_t}\mathbf{Y}_{t-1} + \mathbf{W}'_t, \quad \mathbf{H} \text{ does } \textit{not} \text{ appear!}
\end{aligned}$$

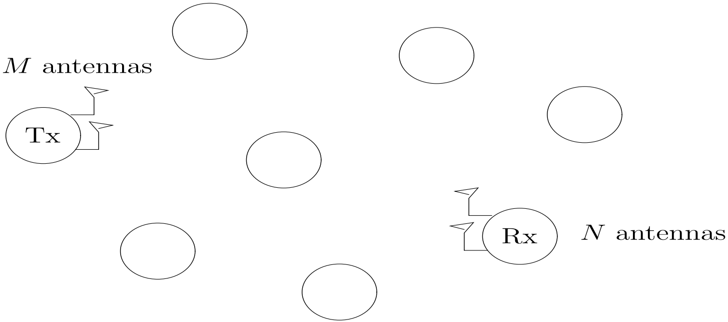
- The *pairwise probability of error* P_e has the upper bound

$$P_e \leq \left(\frac{1}{2}\right)\left(\frac{8}{\rho}\right)^{MN} \frac{1}{|\det(\mathbf{X}_i - \mathbf{X}_j)|^{2N}}$$

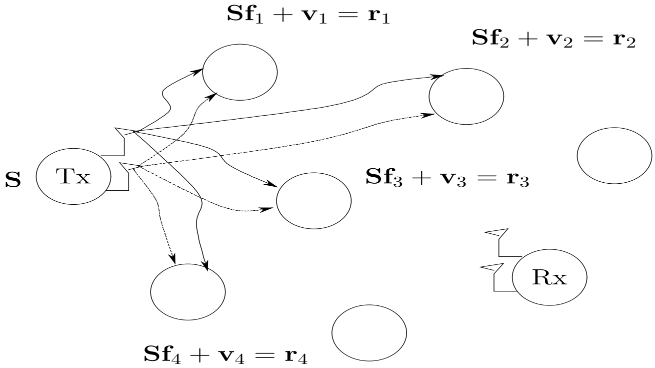
- We need to design *unitary fully-diverse* matrices.

Wireless relay network: model

- A transmitter and a receiver node.
- Relay nodes are small devices with *few* resources.

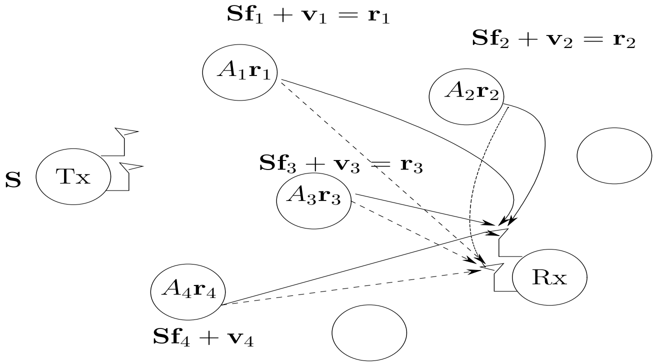


Wireless relay network: phase 1



Wireless relay network: phase 2

- At each node: multiply by a *unitary* matrix.



Channel model

1. At the receiver,

$$\mathbf{y}_n = \sum_{i=1}^{R} g_{in}\mathbf{t}_i + \mathbf{w} = \sum_{i=1}^{R} g_{in}A_i(\mathbf{S}\mathbf{f}_i + \mathbf{v}_i) + \mathbf{w}$$

2. So that finally

$$Y = \begin{bmatrix} \mathbf{y}_1 \\ \vdots \\ \mathbf{y}_n \end{bmatrix} = \underbrace{[A_1\mathbf{S} \cdots A_R\mathbf{S}]}_{\mathbf{X}} \underbrace{\begin{bmatrix} \mathbf{f}_1\mathbf{g}_1 \\ \vdots \\ \mathbf{f}_n\mathbf{g}_n \end{bmatrix}}_{\mathbf{H}} + W$$

3. Each relay encodes a set of columns, so that the encoding is *distributed* among the nodes.

Channel model

1. At the receiver,

$$\mathbf{y}_n = \sum_{i=1}^{R} g_{in}\mathbf{t}_i + \mathbf{w} = \sum_{i=1}^{R} g_{in}A_i(\mathbf{S}\mathbf{f}_i + \mathbf{v}_i) + \mathbf{w}$$

2. So that finally

$$Y = \begin{bmatrix} \mathbf{y}_1 \\ \vdots \\ \mathbf{y}_n \end{bmatrix} = \underbrace{[A_1\mathbf{S} \cdots A_R\mathbf{S}]}_{\mathbf{X}} \underbrace{\begin{bmatrix} \mathbf{f}_1\mathbf{g}_1 \\ \vdots \\ \mathbf{f}_n\mathbf{g}_n \end{bmatrix}}_{\mathbf{H}} + W$$

3. Each relay encodes a set of columns, so that the encoding is *distributed* among the nodes.

The idea behind division algebras

- The difficulty in building $\mathcal{C}$ such that

$$\det(\mathbf{X}_i - \mathbf{X}_j) \neq 0,\ \mathbf{X}_i \neq \mathbf{X}_j \in \mathcal{C},$$

comes from the *non-linearity* of the determinant.

- If $\mathcal{C}$ is taken inside an *algebra* of matrices, the problem simplifies to

$$\det(\mathbf{X}) \neq 0,\ \mathbf{0} \neq \mathbf{X} \in \mathcal{C}.$$

- A *division algebra* is a non-commutative field.

The idea behind division algebras

- The difficulty in building $\mathcal{C}$ such that
$$\det(\mathbf{X}_i - \mathbf{X}_j) \neq 0,\ \mathbf{X}_i \neq \mathbf{X}_j \in \mathcal{C},$$
comes from the *non-linearity* of the determinant.
- If $\mathcal{C}$ is taken inside an *algebra* of matrices, the problem simplifies to
$$\det(\mathbf{X}) \neq 0,\ \mathbf{0} \neq \mathbf{X} \in \mathcal{C}.$$
- A *division algebra* is a non-commutative field.

The idea behind division algebras

- The difficulty in building $\mathcal{C}$ such that
$$\det(\mathbf{X}_i - \mathbf{X}_j) \neq 0,\ \mathbf{X}_i \neq \mathbf{X}_j \in \mathcal{C},$$
comes from the *non-linearity* of the determinant.
- If $\mathcal{C}$ is taken inside an *algebra* of matrices, the problem simplifies to
$$\det(\mathbf{X}) \neq 0,\ \mathbf{0} \neq \mathbf{X} \in \mathcal{C}.$$
- A *division algebra* is a non-commutative field.

An example: cyclic division algebras

- Let $\mathbb{Q}(i) = \{a + ib,\ a, b\ \in \mathbb{Q}\}$.
- Let L be cyclic extension of degree n over $\mathbb{Q}(i)$.
- A *cyclic algebra* $\mathcal{A}$ is defined as follows

$$\mathcal{A} = \{(x_0, x_1, \ldots, x_{n-1}) \mid x_i \in L\}$$

  with basis $\{1, e, \ldots, e^{n-1}\}$ and $e^n = \gamma \in \mathbb{Q}(i)$.
- Think of $i^2 = -1$.
- A *non-commutativity rule*: $\lambda e = e\sigma(\lambda)$, $\sigma : L \longrightarrow L$ the generator of the Galois group of $L/\mathbb{Q}(i)$.

An example: cyclic division algebras

- Let $\mathbb{Q}(i) = \{a + ib,\ a, b\ \in \mathbb{Q}\}$.
- Let L be cyclic extension of degree n over $\mathbb{Q}(i)$.
- A *cyclic algebra* $\mathcal{A}$ is defined as follows

$$\mathcal{A} = \{(x_0, x_1, \ldots, x_{n-1}) \mid x_i \in L\}$$

with basis $\{1, e, \ldots, e^{n-1}\}$ and $e^n = \gamma \in \mathbb{Q}(i)$.
- Think of $i^2 = -1$.
- A *non-commutativity rule*: $\lambda e = e\sigma(\lambda)$, $\sigma : L \longrightarrow L$ the generator of the Galois group of $L/\mathbb{Q}(i)$.

An example: cyclic division algebras

- Let $\mathbb{Q}(i) = \{a + ib,\ a, b\ \in \mathbb{Q}\}$.
- Let L be cyclic extension of degree n over $\mathbb{Q}(i)$.
- A *cyclic algebra* $\mathcal{A}$ is defined as follows

$$\mathcal{A} = \{(x_0, x_1, \ldots, x_{n-1}) \mid x_i \in L\}$$

with basis $\{1, e, \ldots, e^{n-1}\}$ and $e^n = \gamma \in \mathbb{Q}(i)$.

- Think of $i^2 = -1$.
- A *non-commutativity rule*: $\lambda e = e\sigma(\lambda)$, $\sigma : L \longrightarrow L$ the generator of the Galois group of $L/\mathbb{Q}(i)$.

An example: cyclic division algebras

- Let $\mathbb{Q}(i) = \{a + ib,\ a, b\ \in \mathbb{Q}\}$.
- Let L be cyclic extension of degree n over $\mathbb{Q}(i)$.
- A *cyclic algebra* $\mathcal{A}$ is defined as follows

$$\mathcal{A} = \{(x_0, x_1, \dots, x_{n-1}) \mid x_i \in L\}$$

with basis $\{1, e, \dots, e^{n-1}\}$ and $e^n = \gamma \in \mathbb{Q}(i)$.
- Think of $i^2 = -1$.
- A *non-commutativity rule*: $\lambda e = e\sigma(\lambda)$, $\sigma : L \longrightarrow L$ the generator of the Galois group of $L/\mathbb{Q}(i)$.

An example: cyclic division algebras

- Let $\mathbb{Q}(i) = \{a + ib,\ a, b\ \in \mathbb{Q}\}$.
- Let L be cyclic extension of degree n over $\mathbb{Q}(i)$.
- A *cyclic algebra* $\mathcal{A}$ is defined as follows

$$\mathcal{A} = \{(x_0, x_1, \dots, x_{n-1}) \mid x_i \in L\}$$

  with basis $\{1, e, \dots, e^{n-1}\}$ and $e^n = \gamma \in \mathbb{Q}(i)$.
- Think of $i^2 = -1$.
- A *non-commutativity rule*: $\lambda e = e\sigma(\lambda)$, $\sigma : L \to L$ the generator of the Galois group of $L/\mathbb{Q}(i)$.

Cyclic algebras: matrix formulation

1. For $n = 2$, compute the *multiplication* by x of any $y \in \mathcal{A}$:

$$\begin{aligned} xy &= (x_0 + ex_1)(y_0 + ey_1) \\ &= x_0 y_0 + e\sigma(x_0)y_1 + ex_1 y_0 + \gamma\sigma(x_1)y_1 \qquad \lambda e = e\sigma(\lambda) \\ &= [x_0 y_0 + \gamma\sigma(x_1)y_1] + e[\sigma(x_0)y_1 + x_1 y_0] \qquad e^2 = \gamma \end{aligned}$$

2. In the basis $\{1, e\}$, this yields

$$xy = \begin{pmatrix} x_0 & \gamma\sigma(x_1) \\ x_1 & \sigma(x_0) \end{pmatrix} \begin{pmatrix} y_0 \\ y_1 \end{pmatrix}.$$

3. There is thus a correspondence between x and its *multiplication matrix*.

$$x = x_0 + ex_1 \in \mathcal{A} \leftrightarrow \begin{pmatrix} x_0 & \gamma\sigma(x_1) \\ x_1 & \sigma(x_0) \end{pmatrix}.$$

Cyclic algebras: matrix formulation

1. For $n=2$, compute the *multiplication* by x of any $y \in \mathcal{A}$:

$$\begin{aligned} xy &= (x_0+ex_1)(y_0+ey_1) \\ &= x_0y_0 + e\sigma(x_0)y_1 + ex_1y_0 + \gamma\sigma(x_1)y_1 && \lambda e = e\sigma(\lambda) \\ &= [x_0y_0+\gamma\sigma(x_1)y_1] + e[\sigma(x_0)y_1 + x_1y_0] && e^2=\gamma \end{aligned}$$

2. In the basis $\{1, e\}$, this yields

$$xy = \begin{pmatrix} x_0 & \gamma\sigma(x_1) \\ x_1 & \sigma(x_0) \end{pmatrix} \begin{pmatrix} y_0 \\ y_1 \end{pmatrix}.$$

3. There is thus a correspondence between x and its *multiplication matrix*.

$$x = x_0 + ex_1 \in \mathcal{A} \leftrightarrow \begin{pmatrix} x_0 & \gamma\sigma(x_1) \\ x_1 & \sigma(x_0) \end{pmatrix}.$$

Cyclic algebras: matrix formulation

1. For $n = 2$, compute the *multiplication* by x of any $y \in \mathcal{A}$:

$$\begin{aligned} xy &= (x_0 + ex_1)(y_0 + ey_1) \\ &= x_0 y_0 + e\sigma(x_0)y_1 + ex_1 y_0 + \gamma\sigma(x_1)y_1 && \lambda e = e\sigma(\lambda) \\ &= [x_0 y_0 + \gamma\sigma(x_1)y_1] + e[\sigma(x_0)y_1 + x_1 y_0] && e^2 = \gamma \end{aligned}$$

2. In the basis $\{1, e\}$, this yields

$$xy = \begin{pmatrix} x_0 & \gamma\sigma(x_1) \\ x_1 & \sigma(x_0) \end{pmatrix} \begin{pmatrix} y_0 \\ y_1 \end{pmatrix}.$$

3. There is thus a correspondence between x and its *multiplication matrix*.

$$x = x_0 + ex_1 \in \mathcal{A} \leftrightarrow \begin{pmatrix} x_0 & \gamma\sigma(x_1) \\ x_1 & \sigma(x_0) \end{pmatrix}.$$

Cyclic division algebras and encoding

- **Proposition.** If γ and its powers $\gamma^2,\ldots,\gamma^{n-1}$ are not a norm, then the cyclic algebra $\mathcal{A}$ is a *division algebra*.
- In general

$$x \leftrightarrow \begin{pmatrix} x_0 & \gamma\sigma(x_{n-1}) & \gamma\sigma^2(x_{n-2}) & \ldots & \gamma\sigma^{n-1}(x_1) \\ x_1 & \sigma(x_0) & \gamma\sigma^2(x_{n-1}) & \ldots & \gamma\sigma^{n-1}(x_2) \\ \vdots & & \vdots & & \vdots \\ x_{n-1} & \sigma(x_{n-2}) & \sigma^2(x_{n-3}) & \ldots & \sigma^{n-1}(x_0) \end{pmatrix}.$$

- Each $x_i \in L$ *encodes* n information symbols.

Cyclic division algebras and encoding

- **Proposition.** If γ and its powers $\gamma^2, \ldots, \gamma^{n-1}$ are not a norm, then the cyclic algebra $\mathcal{A}$ is a *division algebra*.
- In general

$$x \leftrightarrow \begin{pmatrix} x_0 & \gamma\sigma(x_{n-1}) & \gamma\sigma^2(x_{n-2}) & \ldots & \gamma\sigma^{n-1}(x_1) \\ x_1 & \sigma(x_0) & \gamma\sigma^2(x_{n-1}) & \ldots & \gamma\sigma^{n-1}(x_2) \\ \vdots & & \vdots & & \vdots \\ x_{n-1} & \sigma(x_{n-2}) & \sigma^2(x_{n-3}) & \ldots & \sigma^{n-1}(x_0) \end{pmatrix}.$$

- Each $x_i \in L$ *encodes* n information symbols.

Solutions for the coding problems

Start with a cyclic division algebra, and:

1. For *space-time coding*: use the underlying algebraic properties to optimize the code (for example the discriminant of $L/\mathbb{Q}(i)$).
2. For *differential space-time coding*: endowe the algebra with a suitable *involution*, or use the *Cayley transform*.
3. For *distributed space-time coding*: work in a suitable subfield of L.

Solutions for the coding problems

Start with a cyclic division algebra, and:

1. For *space-time coding*: use the underlying algebraic properties to optimize the code (for example the discriminant of $L/\mathbb{Q}(i)$).
2. For *differential space-time coding*: endowe the algebra with a suitable *involution*, or use the *Cayley transform*.
3. For *distributed space-time coding*: work in a suitable subfield of L.

Solutions for the coding problems

Start with a cyclic division algebra, and:

1. For *space-time coding*: use the underlying algebraic properties to optimize the code (for example the discriminant of $L/\mathbb{Q}(i)$).
2. For *differential space-time coding*: endowe the algebra with a suitable *involution*, or use the *Cayley transform*.
3. For *distributed space-time coding*: work in a suitable subfield of L.

Thank you for your attention!