Decoding of Space-Time Block Codes

Jean-Claude Belfiore

Telecom ParisTech

The Arithmetics of Wireless Communications
Scuola Normale Superiore di Pisa
November, 17$^{\text{th}}$ 2008

Part I

The MIMO Channel

The MIMO Channel

The quasi static fading channel

MIMO System

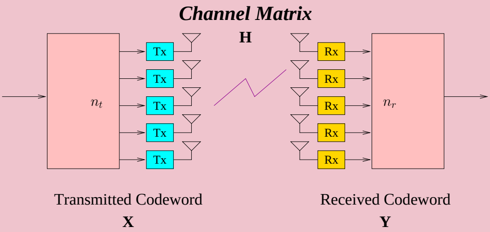


Figure: The Channel Model

The quasi static fading channel

MIMO System

Channel Matrix

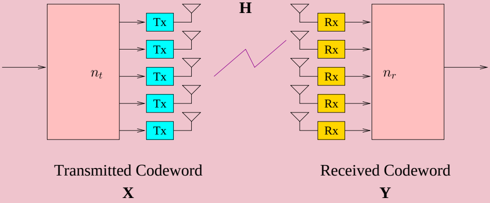


Figure: The Channel Model

- Received signal

$$\boxed{\boldsymbol{Y}_{n_r \times T} = \boldsymbol{H}_{n_r \times n_t} \cdot \boldsymbol{X}_{n_t \times T} + \boldsymbol{W}_{n_r \times T}} \quad (1)$$

  with $\boldsymbol{H}$ perfectly known at the receiver (coherent case).
- $\boldsymbol{H}$ is assumed constant during the transmission of one codeword.

Example of the Golden Code ($n_t = n_r = T = 2$)

Codeword

A codeword $\boldsymbol{X}$ of the Golden code is

$$\boldsymbol{X} = \frac{1}{\sqrt{5}} \cdot \begin{bmatrix} \alpha(s_1 + \theta s_2) & \alpha(s_3 + \theta s_4) \\ i\bar{\alpha}(s_3 + \bar{\theta} s_4) & \bar{\alpha}(s_1 + \bar{\theta} s_2) \end{bmatrix}$$

with $\theta = \frac{1+\sqrt{5}}{2}$, $\bar{\theta} = \frac{1-\sqrt{5}}{2}$, $\alpha = 1 + i - i\theta$, $\overline{\alpha} = 1 + i - i\overline{\theta}$ and s_l, $l = 1 \ldots 4$ are the information symbols carved from a q-QAM.

Example of the Golden Code ($n_t = n_r = T = 2$)

Codeword

A codeword $\boldsymbol{X}$ of the Golden code is

$$\boldsymbol{X} = \frac{1}{\sqrt{5}} \cdot \begin{bmatrix} \alpha(s_1 + \theta s_2) & \alpha(s_3 + \theta s_4) \\ i\bar{\alpha}(s_3 + \bar{\theta} s_4) & \bar{\alpha}(s_1 + \bar{\theta} s_2) \end{bmatrix}$$

with $\theta = \frac{1+\sqrt{5}}{2}$, $\bar{\theta} = \frac{1-\sqrt{5}}{2}$, $\alpha = 1 + i - i\theta$, $\overline{\alpha} = 1 + i - i\overline{\theta}$ and s_l, $l = 1\ldots4$ are the information symbols carved from a q-QAM.

- Vectorization

$$\boldsymbol{y}_{n\cdot T\times 1} = \frac{1}{\sqrt{5}} \cdot \begin{bmatrix} \boldsymbol{H} & 0 \\ 0 & \boldsymbol{H} \end{bmatrix} \cdot \begin{bmatrix} \alpha & \alpha\theta & 0 & 0 \\ 0 & 0 & i\bar{\alpha} & i\bar{\alpha}\bar{\theta} \\ 0 & 0 & \alpha & \alpha\theta \\ \bar{\alpha} & \bar{\alpha}\bar{\theta} & 0 & 0 \end{bmatrix} \begin{bmatrix} s_1 \\ s_2 \\ s_3 \\ s_4 \end{bmatrix} + \boldsymbol{w}$$

$$= \frac{1}{\sqrt{5}} \cdot \boldsymbol{H}_{n\cdot T} \cdot \boldsymbol{\Phi}_{n\cdot T} \cdot \boldsymbol{s}_{n\cdot T\times 1} + \boldsymbol{w}_{n\cdot T\times 1}$$

where $\boldsymbol{\Phi}_{n\cdot T}$ is a unitary matrix.

Example of the Golden code (cont'd)

- Separation of the real and the imaginary parts

$$\begin{aligned} \boldsymbol{y}_{\mathbb{R}} &= \frac{1}{\sqrt{5}} \cdot \begin{bmatrix} \Re(\boldsymbol{H}) & 0 & -\Im(\boldsymbol{H}) & 0 \\ 0 & \Re(\boldsymbol{H}) & 0 & -\Im(\boldsymbol{H}) \\ \Im(\boldsymbol{H}) & 0 & \Re(\boldsymbol{H}) & 0 \\ 0 & \Im(\boldsymbol{H}) & 0 & \Re(\boldsymbol{H}) \end{bmatrix} \cdot \begin{bmatrix} \Re(\boldsymbol{\Phi}) & -\Im(\boldsymbol{\Phi}) \\ \Im(\boldsymbol{\Phi}) & \Re(\boldsymbol{\Phi}) \end{bmatrix} \begin{bmatrix} \Re(\boldsymbol{s}) \\ \Im(\boldsymbol{s}) \end{bmatrix} + \begin{bmatrix} \Re(\boldsymbol{w}) \\ \Im(\boldsymbol{w}) \end{bmatrix} \\ &= \frac{1}{\sqrt{5}} \cdot \boldsymbol{H}_{\mathbb{R}} \cdot \boldsymbol{\Phi}_{\mathbb{R}} \cdot \boldsymbol{s}_{\mathbb{R}} + \boldsymbol{w}_{\mathbb{R}} \end{aligned}$$

Example of the Golden code (cont'd)

- Separation of the real and the imaginary parts

$$\begin{aligned} \boldsymbol{y}_{\mathbb{R}} &= \frac{1}{\sqrt{5}} \cdot \begin{bmatrix} \Re(\boldsymbol{H}) & 0 & -\Im(\boldsymbol{H}) & 0 \\ 0 & \Re(\boldsymbol{H}) & 0 & -\Im(\boldsymbol{H}) \\ \Im(\boldsymbol{H}) & 0 & \Re(\boldsymbol{H}) & 0 \\ 0 & \Im(\boldsymbol{H}) & 0 & \Re(\boldsymbol{H}) \end{bmatrix} \cdot \begin{bmatrix} \Re(\boldsymbol{\Phi}) & -\Im(\boldsymbol{\Phi}) \\ \Im(\boldsymbol{\Phi}) & \Re(\boldsymbol{\Phi}) \end{bmatrix} \begin{bmatrix} \Re(\boldsymbol{s}) \\ \Im(\boldsymbol{s}) \end{bmatrix} + \begin{bmatrix} \Re(\boldsymbol{w}) \\ \Im(\boldsymbol{w}) \end{bmatrix} \\ &= \frac{1}{\sqrt{5}} \cdot \boldsymbol{H}_{\mathbb{R}} \cdot \boldsymbol{\Phi}_{\mathbb{R}} \cdot \boldsymbol{s}_{\mathbb{R}} + \boldsymbol{w}_{\mathbb{R}} \end{aligned}$$

Equivalent Channel

Let $\boldsymbol{M} = \frac{1}{\sqrt{5}} \cdot \boldsymbol{H}_{\mathbb{R}} \cdot \boldsymbol{\Phi}_{\mathbb{R}}$, we get

$$\boxed{\boldsymbol{y}_{\mathbb{R}} = \boldsymbol{M} \cdot \boldsymbol{s}_{\mathbb{R}} + \boldsymbol{w}_{\mathbb{R}}}$$

where vectors $\boldsymbol{y}_{\mathbb{R}}$, $\boldsymbol{s}_{\mathbb{R}}$ and $\boldsymbol{w}_{\mathbb{R}}$ are 8-dimensional vectors and $\boldsymbol{s}_{\mathbb{R}}$ is a vector with integer components.

Example of the Golden code (cont'd)

- Separation of the real and the imaginary parts

$$\begin{aligned} \boldsymbol{y}_{\mathbb{R}} &= \frac{1}{\sqrt{5}} \cdot \begin{bmatrix} \Re(\boldsymbol{H}) & 0 & -\Im(\boldsymbol{H}) & 0 \\ 0 & \Re(\boldsymbol{H}) & 0 & -\Im(\boldsymbol{H}) \\ \Im(\boldsymbol{H}) & 0 & \Re(\boldsymbol{H}) & 0 \\ 0 & \Im(\boldsymbol{H}) & 0 & \Re(\boldsymbol{H}) \end{bmatrix} \cdot \begin{bmatrix} \Re(\boldsymbol{\Phi}) & -\Im(\boldsymbol{\Phi}) \\ \Im(\boldsymbol{\Phi}) & \Re(\boldsymbol{\Phi}) \end{bmatrix} \begin{bmatrix} \Re(\boldsymbol{s}) \\ \Im(\boldsymbol{s}) \end{bmatrix} + \begin{bmatrix} \Re(\boldsymbol{w}) \\ \Im(\boldsymbol{w}) \end{bmatrix} \\ &= \frac{1}{\sqrt{5}} \cdot \boldsymbol{H}_{\mathbb{R}} \cdot \boldsymbol{\Phi}_{\mathbb{R}} \cdot \boldsymbol{s}_{\mathbb{R}} + \boldsymbol{w}_{\mathbb{R}} \end{aligned}$$

Equivalent Channel

Let $\boldsymbol{M} = \frac{1}{\sqrt{5}} \cdot \boldsymbol{H}_{\mathbb{R}} \cdot \boldsymbol{\Phi}_{\mathbb{R}}$, we get

$$\boxed{\boldsymbol{y}_{\mathbb{R}} = \boldsymbol{M} \cdot \boldsymbol{s}_{\mathbb{R}} + \boldsymbol{w}_{\mathbb{R}}}$$

where vectors $\boldsymbol{y}_{\mathbb{R}}$, $\boldsymbol{s}_{\mathbb{R}}$ and $\boldsymbol{w}_{\mathbb{R}}$ are 8-dimensional vectors and $\boldsymbol{s}_{\mathbb{R}}$ is a vector with integer components.

- More generally, the (real) dimension of the vectors is equal to $2 \cdot n_t \cdot T$.

The MIMO Channel

Lattice : Definition

Definition

A ***Euclidean lattice*** is a discrete additive subgroup with rank p, $p \leq n$ of the Euclidean space $\mathbb{R}^n$. We assume $p = n$ in the sequel.

Lattice : Definition

Definition

A ***Euclidean lattice*** is a discrete additive subgroup with rank p, $p \leq n$ of the Euclidean space $\mathbb{R}^n$. We assume $p = n$ in the sequel.

- A lattice $\boldsymbol{\Lambda}$ is a set generated by vectors $\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_n$ of $\mathbb{R}^n$.
- An element $\boldsymbol{v}$ of $\boldsymbol{\Lambda}$ can be written as :

$$\boldsymbol{v} = a_1 \boldsymbol{v}_1 + a_2 \boldsymbol{v}_2 + \ldots + a_n \boldsymbol{v}_n, \; a_1, a_2, \ldots, a_n \in \mathbb{Z}$$

Lattice : Definition

Definition

A ***Euclidean lattice*** is a discrete additive subgroup with rank p, $p \leq n$ of the Euclidean space $\mathbb{R}^n$. We assume $p = n$ in the sequel.

- A lattice $\boldsymbol{\Lambda}$ is a set generated by vectors $\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_n$ of $\mathbb{R}^n$.
- An element $\boldsymbol{v}$ of $\boldsymbol{\Lambda}$ can be written as :

$$\boldsymbol{v} = a_1 \boldsymbol{v}_1 + a_2 \boldsymbol{v}_2 + \ldots + a_n \boldsymbol{v}_n,\ a_1, a_2, \ldots, a_n \in \mathbb{Z}$$

- The lattice $\boldsymbol{\Lambda}$ can be defined as :

$$\boldsymbol{\Lambda} = \left\{ \sum_{i=1}^{n} a_i \boldsymbol{v}_i \mid a_i \in \mathbb{Z} \right\}$$

Lattices : Parameters (1)

- The set of vectors $\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_n$ is a **lattice basis**, with **dimension** n

Definition

Matrix $\boldsymbol{M}$ whose columns are vectors $\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_n$ is a **generator matrix** of the lattice denoted $\Lambda_{\boldsymbol{M}}$.

Lattices : Parameters (1)

- The set of vectors $\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_n$ is a **lattice basis**, with **dimension** n

Definition

Matrix $\boldsymbol{M}$ whose columns are vectors $\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_n$ is a **generator matrix** of the lattice denoted $\Lambda_{\boldsymbol{M}}$.

- Each vector $\mathbf{x} = (x_1, x_2, \ldots, x_n)$ in $\Lambda_{\boldsymbol{M}}$, can be written as,

$$\boldsymbol{x} = \boldsymbol{M} \cdot \boldsymbol{z}$$

where $\boldsymbol{z} = \left(z_1, z_2, \ldots, z_p\right)^\top \in \mathbb{Z}^p$.

- Lattice $\Lambda_{\boldsymbol{M}}$ may be seen as the result of a linear transform applied to lattice $\mathbb{Z}^n$.

Lattices : Properties (2)

- Let $\boldsymbol{Q} \in \mathscr{M}_n(\mathbb{R})$, such that $\boldsymbol{Q} \cdot \boldsymbol{Q}^\top = I_n$ and $\det \boldsymbol{Q} = \pm 1$. $\boldsymbol{Q}$ is an isometry. The two lattices $\Lambda_{\boldsymbol{M}}$ and $\Lambda_{\boldsymbol{Q} \cdot \boldsymbol{M}}$ are **equivalent**.
- Lattice $\Lambda_{\boldsymbol{Q} \cdot \boldsymbol{M}}$ is a rotated version of $\Lambda_{\boldsymbol{M}}$.

Lattices : Properties (2)

- Let $\boldsymbol{Q} \in \mathscr{M}_n(\mathbb{R})$, such that $\boldsymbol{Q} \cdot \boldsymbol{Q}^\top = I_n$ and $\det \boldsymbol{Q} = \pm 1$. $\boldsymbol{Q}$ is an isometry. The two lattices $\Lambda_{\boldsymbol{M}}$ and $\Lambda_{\boldsymbol{Q}\cdot\boldsymbol{M}}$ are **equivalent**.
- Lattice $\Lambda_{\boldsymbol{Q}\cdot\boldsymbol{M}}$ is a rotated version of $\Lambda_{\boldsymbol{M}}$.
- If $\boldsymbol{Q} \in \mathscr{M}_n(\mathbb{Z})$ and $\det \boldsymbol{Q} \neq \pm 1$, then lattice $\Lambda_{\boldsymbol{M}\cdot\boldsymbol{Q}}$ is a **sublattice** of $\Lambda_{\boldsymbol{M}}$.
- A sublattice of $\Lambda_{\boldsymbol{M}}$ is a subgroup of $\Lambda_{\boldsymbol{M}}$.
- An integer lattice is a sublattice of $\mathbb{Z}^n$.

Lattices : Properties (3)

- The generator matrix $\boldsymbol{M}$ describes the lattice $\Lambda_{\boldsymbol{M}}$, but it is not unique. All matrices $\boldsymbol{M} \cdot \boldsymbol{T}$ with $\boldsymbol{T} \in \mathscr{M}_n(\mathbb{Z})$ and $\det \boldsymbol{T} = \pm 1$ are generator matrices of $\Lambda_{\boldsymbol{M}}$. $\boldsymbol{T}$ is called a unimodular matrix.
- We define then invariant parameters.

Lattices : Properties (3)

- The generator matrix $\boldsymbol{M}$ describes the lattice $\Lambda_{\boldsymbol{M}}$, but it is not unique. All matrices $\boldsymbol{M} \cdot \boldsymbol{T}$ with $\boldsymbol{T} \in \mathcal{M}_n(\mathbb{Z})$ and $\det \boldsymbol{T} = \pm 1$ are generator matrices of $\Lambda_{\boldsymbol{M}}$. $\boldsymbol{T}$ is called a unimodular matrix.
- We define then invariant parameters.

Definitions

- The **fundamental parallelotope** of $\Lambda_{\boldsymbol{M}}$ is the region,

$$\mathscr{P} = \left\{\boldsymbol{x} \in \mathbb{R}^n \mid \boldsymbol{x} = a_1 \boldsymbol{v}_1 + a_2 \boldsymbol{v}_2 + \ldots + a_n \boldsymbol{v}_n,\ 0 \le a_i < 1,\ i = 1 \ldots n\right\}$$

- The ***fundamental volume*** is the volume of the fundamental parallelotope. It is denoted $\mathrm{vol}(\Lambda_{\boldsymbol{M}})$.
- $\boldsymbol{G} = \boldsymbol{M}^\top \cdot \boldsymbol{M}$ is the ***Gram matrix*** of the lattice (not invariant).
- The fundamental volume of the lattice is $|\det(\boldsymbol{M})|$, which is $\sqrt{|\det(\boldsymbol{G})|}$ either.

Lattices : Properties (4)

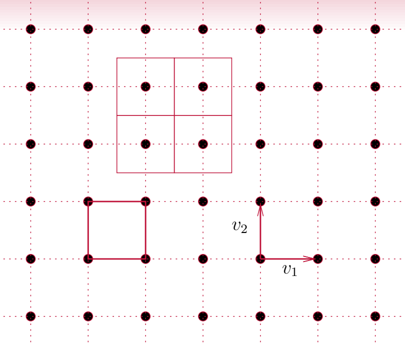
Definition

The ***Voronoï cell*** of a point u belonging to the lattice Λ is the region

$$\mathcal{V}(\boldsymbol{u}) = \left\{\boldsymbol{x} \in \mathbb{R}^n \mid \|\boldsymbol{x}-\boldsymbol{u}\| \leq \left\|\boldsymbol{x}-\boldsymbol{y}\right\|, \boldsymbol{y} \in \Lambda\right\}$$

- Since a lattice is geometrically uniform, all Voronoï cells of a lattice are translated versions of the Voronoï cell of the zero point. This cell is called **Voronoï cell of the lattice**.
- The fundamental volume of a lattice is equal to the volume of its Voronoï cell.

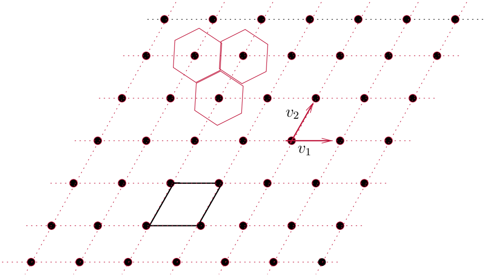
The $\mathbb{Z}^2$ -lattice



$\mathbb{Z}[i]$ lattice

●	Lattice Point
(v_1, v_2)	Lattice Basis
◇	Fundamental Parallelotope
◇	Voronoi region

The A_2 lattice



The A_2 lattice

●	Lattice point
(v_1, v_2)	Lattice basis
◇	Fundamental parallelotope
◇	Voronoi region

Constellations defined from $\mathbb{Z}[j]$

- Perfect STBCs of dimension 3 and 6 use symbols carved from $q-$HEX constellations.
- The lattice representation of a MIMO system using such codes needs some additional procedure. Simply, note that $\mathbb{Z}[j]$ is the hexagonal lattice A_2 with generator matrix.

$$\boldsymbol{B} = \begin{bmatrix} 1 & -0.5 \\ 0 & \frac{\sqrt{3}}{2} \end{bmatrix}$$

$$\begin{aligned} \boldsymbol{y}_{\mathbb{R}} &= \begin{bmatrix} \Re(\boldsymbol{H}) & \cdots & 0 & -\Im(\boldsymbol{H}) & \cdots & 0 \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & \Re(\boldsymbol{H}) & 0 & \cdots & -\Im(\boldsymbol{H}) \\ \Im(\boldsymbol{H}) & \cdots & 0 & \Re(\boldsymbol{H}) & \cdots & 0 \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & \Im(\boldsymbol{H}) & 0 & \cdots & \Re(\boldsymbol{H}) \end{bmatrix} \cdot \begin{bmatrix} 1 & \cdots & 0 & -0.5 & \cdots & 0 \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & 1 & 0 & \cdots & -0.5 \\ 0 & \cdots & 0 & \frac{\sqrt{3}}{2} & \cdots & 0 \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & 0 & 0 & \cdots & \frac{\sqrt{3}}{2} \end{bmatrix} \\ &\cdot \begin{bmatrix} \Re(\boldsymbol{\Phi}) & -\Im(\boldsymbol{\Phi}) \\ \Im(\boldsymbol{\Phi}) & \Re(\boldsymbol{\Phi}) \end{bmatrix} \begin{bmatrix} \Re(\boldsymbol{s}) \\ \Im(\boldsymbol{s}) \end{bmatrix} + \begin{bmatrix} \Re(\boldsymbol{w}) \\ \Im(\boldsymbol{w}) \end{bmatrix} \\ &= \boldsymbol{H}_{\mathbb{R}} \cdot \boldsymbol{B}_{\otimes} \cdot \boldsymbol{\Phi}_{\mathbb{R}} \cdot \boldsymbol{s}_{\mathbb{R}} + \boldsymbol{w}_{\mathbb{R}} \end{aligned}$$

Part II

Lattice Decoding

Lattice Decoding

Introduction

- Traditional constellations (QAM, HEX) are carved from lattices ($\mathbb{Z}^2$, A_2). Labelling and shaping is easier to perform.

Introduction

- Traditional constellations (QAM, HEX) are carved from lattices ($\mathbb{Z}^2$, A_2). Labelling and shaping is easier to perform.
- Another motivation is their decoding which can be derived from lattice decoding algorithms.
- Lattice decoding algorithms are now well-known, let's cite "sphere decoder, Schnorr-Euchner algorithm, sequential decoding,..."

Closest Point

- The **closest point** to $\boldsymbol{y}$ is the lattice point $\hat{\boldsymbol{z}}$ from Λ_M satisfying

$$\boxed{\|\boldsymbol{y}-\hat{\boldsymbol{z}}\|^2 \le \|\boldsymbol{y}-\boldsymbol{z}\|^2} \text{ for all } \boldsymbol{z} \in \Lambda_M$$

- Lattice decoding consists in finding the closest lattice point to $\boldsymbol{y}$.

Closest Point

- The **closest point** to $\boldsymbol{y}$ is the lattice point $\hat{\boldsymbol{z}}$ from Λ_M satisfying

$$\boxed{\|\boldsymbol{y}-\hat{\boldsymbol{z}}\|^2 \leq \|\boldsymbol{y}-\boldsymbol{z}\|^2} \text{ for all } \boldsymbol{z} \in \Lambda_M$$

- Lattice decoding consists in finding the closest lattice point to $\boldsymbol{y}$.
- The main idea of lattice decoders is to search in some well-chosen region
  - Kannan's strategy : the region is a parallelotope
  - Pohst's strategy : the region is a sphere

Closest Point

- The **closest point** to $\boldsymbol{y}$ is the lattice point $\hat{\boldsymbol{z}}$ from Λ_M satisfying

$$\boxed{\|\boldsymbol{y}-\hat{\boldsymbol{z}}\|^2 \leq \|\boldsymbol{y}-\boldsymbol{z}\|^2} \text{ for all } \boldsymbol{z} \in \Lambda_M$$

- Lattice decoding consists in finding the closest lattice point to $\boldsymbol{y}$.
- The main idea of lattice decoders is to search in some well-chosen region
  - Kannan's strategy : the region is a parallelotope
  - Pohst's strategy : the region is a sphere
- Pohst's strategy is the more practical method. Lattice decoders have been inspired by him : **Sphere decoder** and **Schnorr-Euchner** algorithm.

Lattice Decoding

Sphere Decoding (1)

- Decoding consists in searching the lattice point

$$\hat{z} = \arg\min_{z \in \Lambda} \|y - z\|^2$$

which is equivalent to the minimization

$$\min_{w \in y - \Lambda} \|w\|^2$$

- We need to work in the translated lattice $y - \Lambda$.

Sphere Decoding (1)

- Decoding consists in searching the lattice point

$$\hat{\mathbf{z}} = \arg\min_{\mathbf{z}\in\Lambda} \|\mathbf{y}-\mathbf{z}\|^2$$

which is equivalent to the minimization

$$\min_{\mathbf{w}\in\mathbf{y}-\Lambda} \|\mathbf{w}\|^2$$

- We need to work in the translated lattice $\mathbf{y}-\Lambda$.

Change of coordinates

Let's define

$$\begin{aligned}
\mathbf{z} &= \mathbf{M}\cdot\mathbf{u}, \quad \mathbf{u}\in\mathbb{Z}^n \\
\mathbf{y} &= \mathbf{M}\cdot\boldsymbol{\rho}, \quad \boldsymbol{\rho}=(\rho_1,\ldots,\rho_n)^\top\in\mathbb{R}^n \Rightarrow \text{The ZF point} \\
\mathbf{w} &= \mathbf{y}-\mathbf{z}=\mathbf{M}\cdot(\boldsymbol{\rho}-\mathbf{u})=\mathbf{M}\cdot\boldsymbol{\xi}, \quad \boldsymbol{\xi}=(\xi_1,\cdots,\xi_n)^\top\in\mathbb{R}^n
\end{aligned}$$

Sphere Decoding (1)

- Decoding consists in searching the lattice point

$$\hat{\boldsymbol{z}} = \arg\min_{\boldsymbol{z} \in \Lambda} \|\boldsymbol{y} - \boldsymbol{z}\|^2$$

which is equivalent to the minimization

$$\min_{\boldsymbol{w} \in \boldsymbol{y} - \Lambda} \|\boldsymbol{w}\|^2$$

- We need to work in the translated lattice $\boldsymbol{y} - \Lambda$.

Change of coordinates

Let's define

$$\begin{aligned} \boldsymbol{z} &= \boldsymbol{M} \cdot \boldsymbol{u}, \quad \boldsymbol{u} \in \mathbb{Z}^n \\ \boldsymbol{y} &= \boldsymbol{M} \cdot \boldsymbol{\rho}, \quad \boldsymbol{\rho} = (\rho_1, \ldots, \rho_n)^\top \in \mathbb{R}^n \Rightarrow \text{The ZF point} \\ \boldsymbol{w} &= \boldsymbol{y} - \boldsymbol{z} = \boldsymbol{M} \cdot (\boldsymbol{\rho} - \boldsymbol{u}) = \boldsymbol{M} \cdot \boldsymbol{\xi}, \quad \boldsymbol{\xi} = (\xi_1, \cdots, \xi_n)^\top \in \mathbb{R}^n \end{aligned}$$

- Components ξ_i are those of vector $\boldsymbol{u}$ of $\mathbb{Z}^n$ in the new reference.

Sphere Decoder (2)

- The aim is to find the lattice points in the sphere centered on the received signal and of radius $\sqrt{C}$. So,

$$\|\boldsymbol{w}\|^2 \leq C$$

Sphere Decoder (2)

- The aim is to find the lattice points in the sphere centered on the received signal and of radius $\sqrt{C}$. So,

$$\|\boldsymbol{w}\|^2 \leq C$$

$$\|\boldsymbol{w}\|^2 = Q(\boldsymbol{\xi}) = \boldsymbol{\xi}^\top \cdot \boldsymbol{M}^\top \cdot \boldsymbol{M} \cdot \boldsymbol{\xi} = \boldsymbol{\xi}^\top \cdot \boldsymbol{G} \cdot \boldsymbol{\xi} = \sum_{i=1}^{n} \sum_{j=1}^{n} g_{ij} \xi_i \xi_j \leq C$$

Sphere Decoder (2)

- The aim is to find the lattice points in the sphere centered on the received signal and of radius $\sqrt{C}$. So,

$$\|\boldsymbol{w}\|^2 \leq C$$

$$\|\boldsymbol{w}\|^2 = Q(\boldsymbol{\xi}) = \boldsymbol{\xi}^\top \cdot \boldsymbol{M}^\top \cdot \boldsymbol{M} \cdot \boldsymbol{\xi} = \boldsymbol{\xi}^\top \cdot \boldsymbol{G} \cdot \boldsymbol{\xi} = \sum_{i=1}^{n} \sum_{j=1}^{n} g_{ij} \xi_i \xi_j \leq C$$

- The Cholesky factorization of the Gram matrix $\boldsymbol{G} = \boldsymbol{M}^\top \cdot \boldsymbol{M}$, gives $\boldsymbol{G} = \boldsymbol{R} \cdot \boldsymbol{R}^\top$, where $\boldsymbol{R}^\top = (r_{ji})_{i,j=1\ldots n}$ is an upper triangular matrix.

$$\boxed{Q(\boldsymbol{\xi}) = \boldsymbol{\xi}^\top \boldsymbol{R} \cdot \boldsymbol{R}^\top \boldsymbol{\xi} = \left\| \boldsymbol{R}^\top \cdot \boldsymbol{\xi} \right\|^2 = \sum_{i=1}^{n} \left(r_{ii} \xi_i + \sum_{j=i}^{n} r_{ij} \xi_j \right)^2 \leq C}$$

Sphere Decoding (3)

- Let

$$
\begin{aligned}
q_{ii} &= r_{ii}^2,\ i=1,\ldots,n \\
q_{ij} &= \frac{r_{ij}}{r_{ii}},\ i=1,\ldots,n,\ j=i+1,\ldots,n
\end{aligned}
$$

Sphere Decoding (3)

- Let

$$q_{ii} = r_{ii}^2,\ i = 1, \ldots, n$$
$$q_{ij} = \frac{r_{ij}}{r_{ii}},\ i = 1, \ldots, n,\ j = i+1, \ldots, n$$

Ellipsoid

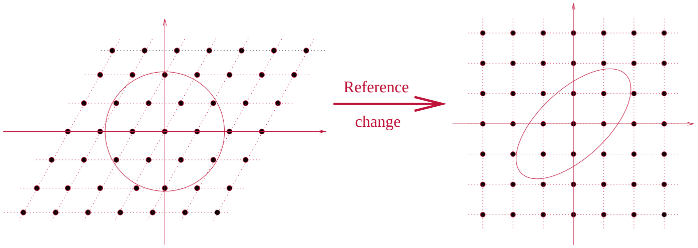
- We get

$$Q(\boldsymbol{\xi}) = \sum_{i=1}^{n} q_{ii} \left(\xi_i + \sum_{j=i+1}^{n} q_{ij} \xi_j \right)^2 \leq C$$

$$Q(\boldsymbol{\xi}) = \sum_{i=1}^{n} q_{ii} U_i^2 \leq C \ \Rightarrow \text{Interior of an ellipsoid}$$

Sphere Decoding (4)

- In the new system defined by $\boldsymbol{\xi}$, the sphere with radius $\sqrt{C}$, centered on the received point, is transformed into an ellipsoid centered on zero and defined by the bilinear form $Q(\boldsymbol{\xi})$.



Sphere Decoding (5)

- In order to determine the ellipsoid boundaries, let do some processing on ξ_n

$$q_{nn}\xi_n^2 \leq C$$

- We have $\xi_n = \rho_n - u_n$

$$\boxed{\left\lceil -\sqrt{\frac{C}{q_{nn}}} + \rho_n \right\rceil \leq u_n \leq \left\lfloor \sqrt{\frac{C}{q_{nn}}} + \rho_n \right\rfloor}$$

where $\lceil x \rceil$ is the smallest integer larger than x and $\lfloor x \rfloor$ is the largest integer smaller than x.

Sphere Decoding (5)

- In order to determine the ellipsoid boundaries, let do some processing on ξ_n

$$q_{nn}\xi_n^2 \leq C$$

- We have $\xi_n = \rho_n - u_n$

$$\boxed{\left\lceil -\sqrt{\frac{C}{q_{nn}}} + \rho_n \right\rceil \leq u_n \leq \left\lfloor \sqrt{\frac{C}{q_{nn}}} + \rho_n \right\rfloor}$$

where $\lceil x \rceil$ is the smallest integer larger than x and $\lfloor x \rfloor$ is the largest integer smaller than x.

- Now the ξ_i, $i = n-1, \ldots, 1$.

$$q_{n-1,n-1}(\xi_{n-1} + q_{n,n-1}\xi_n)^2 + q_{nn}\xi_n^2 \leq C$$

Sphere Decoding (6)

- We get

$$\left\lceil -\sqrt{\frac{C - q_{nn}\xi_n^2}{q_{n-1,n-1}}} + \rho_{n-1} + q_{n-1,n}\xi_n \right\rceil \leq u_{n-1} \leq \left\lfloor \sqrt{\frac{C - q_{nn}\xi_n^2}{q_{n-1,n-1}}} + \rho_{n-1} + q_{n-1,n}\xi_n \right\rfloor$$

Sphere Decoding (6)

- We get

$$\left\lceil -\sqrt{\frac{C-q_{nn}\xi_n^2}{q_{n-1,n-1}}}+\rho_{n-1}+q_{n-1,n}\xi_n \right\rceil \le u_{n-1} \le \left\lfloor \sqrt{\frac{C-q_{nn}\xi_n^2}{q_{n-1,n-1}}}+\rho_{n-1}+q_{n-1,n}\xi_n \right\rfloor$$

- This gives, for the i^{th} component u_i,

$$\left\lceil -\sqrt{\frac{1}{q_{ii}}\left(C-\sum_{l=i+1}^{n} q_{ll}\left(\xi_l+\sum_{j=l+1}^{n} q_{lj}\xi_j\right)^2\right)}+\rho_i+\sum_{j=i+1}^{n} q_{ij}\xi_j \right\rceil \le u_i$$

$$\left\lfloor \sqrt{\frac{1}{q_{ii}}\left(C-\sum_{l=i+1}^{n} q_{ll}\left(\xi_l+\sum_{j=l+1}^{n} q_{lj}\xi_j\right)^2\right)}+\rho_i+\sum_{j=i+1}^{n} q_{ij}\xi_j \right\rfloor \ge u_i$$

Sphere Decoding (7)

- In order to simplify the decoding expressions, we define

$$S_i = \rho_i + \sum_{l=i+1}^{n} q_{il}\xi_l, \; i = 1,\ldots,n$$

$$T_{i-1} = C - \sum_{l=i}^{n} q_{ll}\left(\xi_l + \sum_{j=l+1}^{n} q_{lj}\xi_j\right)^2 = T_i - q_{ii}(S_i - u_i)^2$$

- We get

$$\boxed{b_{\inf,i} = \left\lceil -\sqrt{\frac{T_i}{q_{ii}}} + S_i \right\rceil \le u_i \le \left\lfloor \sqrt{\frac{T_i}{q_{ii}}} + S_i \right\rfloor = b_{\sup,i}}$$

Sphere Decoding (7)

- In order to simplify the decoding expressions, we define

$$
\begin{aligned}
S_i &= \rho_i + \sum_{l=i+1}^{n} q_{il}\xi_l, \; i = 1, \ldots, n \\
T_{i-1} &= C - \sum_{l=i}^{n} q_{ll}\left(\xi_l + \sum_{j=l+1}^{n} q_{lj}\xi_j\right)^2 = T_i - q_{ii}(S_i - u_i)^2
\end{aligned}
$$

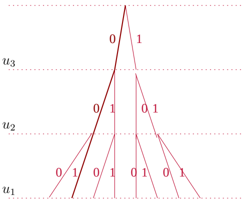
- We get

$$
\boxed{b_{\inf,i} = \left\lceil -\sqrt{\frac{T_i}{q_{ii}}} + S_i \right\rceil \le u_i \le \left\lfloor \sqrt{\frac{T_i}{q_{ii}}} + S_i \right\rfloor = b_{\sup,i}}
$$

- For each component of vector $\boldsymbol{u}$, we define an interval $I_i = [b_{\inf,i}, b_{\sup,i}]$ which contains it.

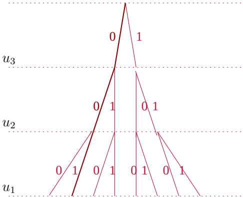
Sphere Decoding (8)

- The Closest point search consists in descending a tree.



Sphere Decoding (8)

- The Closest point search consists in descending a tree.



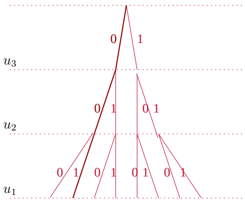
- When a lattice point is found, its squared distance from the received point is given by,

$$\boxed{\hat{d}^2 = C - T_1 + q_{11}(S_1 - u_1)^2}$$

If $\boxed{\hat{d}^2 \le C}$, the point is recorded.

Sphere Decoding (8)

- The Closest point search consists in descending a tree.



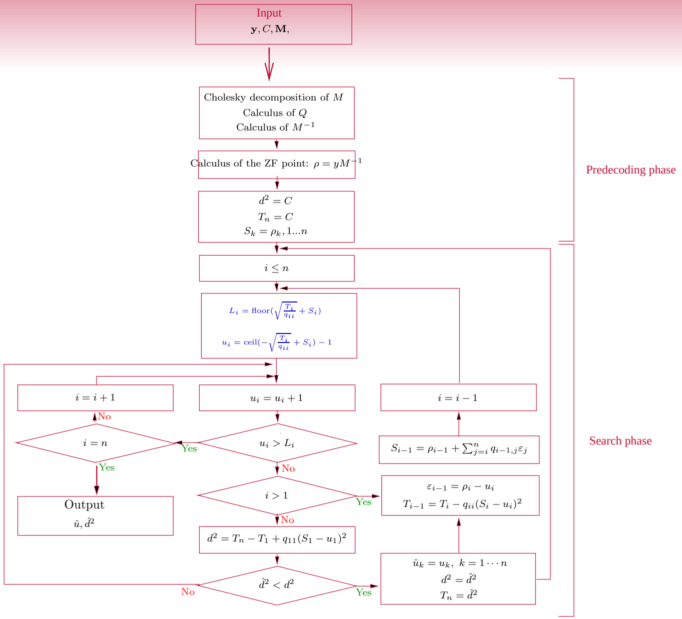
- When a lattice point is found, its squared distance from the received point is given by,

$$\hat{d}^2 = C - T_1 + q_{11}(S_1 - u_1)^2$$

If $\boxed{\hat{d}^2 \leq C}$, the point is recorded.

- The search algorithm makes the sphere radius as well as bounds $b_{\inf,i}$ and $b_{\sup,i}$ for, $i = 1 \cdots n$, release dynamically along the research process when a point is found, i.e. $C \geq \hat{d}^2$.

Flow Chart



Input
$\mathbf{y}, C, \mathbf{M},$
Cholesky decomposition of M
Calculus of Q
Calculus of M^{-1}
Calculus of the ZF point: $\rho = yM^{-1}$
$d^2 = C$
$T_n = C$
$S_k = \rho_k, 1...n$
$i \leq n$
$L_i = \text{floor}(\sqrt{\frac{T_i}{q_{ii}}} + S_i)$
$u_i = \text{ceil}(-\sqrt{\frac{T_i}{q_{ii}}} + S_i) - 1$
$i = i + 1$
$u_i = u_i + 1$
$i = i - 1$
No
$i = n$
Yes
$u_i > L_i$
$S_{i-1} = \rho_{i-1} + \sum_{j=i}^{n} q_{i-1,j}\varepsilon_j$
Yes
No
$i > 1$
Yes
$\varepsilon_{i-1} = \rho_i - u_i$
$T_{i-1} = T_i - q_{ii}(S_i - u_i)^2$
Output
$\hat{u}, \hat{d}^2$
No
$d^2 = T_n - T_1 + q_{11}(S_1 - u_1)^2$
$\hat{d}^2 < d^2$
No
Yes
$\hat{u}_k = u_k,\ k = 1 \cdots n$
$d^2 = \hat{d}^2$
$T_n = \hat{d}^2$
Predecoding phase
Search phase

Choice of the sphere radius

- The radius is a critical parameter for the complexity of the algorithm
    - **A too small radius** : no point inside the sphere
    - **A too large radius** : too many points inside the sphere, which increases the algorithm complexity

Choice of the sphere radius

- The radius is a critical parameter for the complexity of the algorithm
  - **A too small radius** : no point inside the sphere
  - **A too large radius** : too many points inside the sphere, which increases the algorithm complexity
- A good solution is to have the sphere radius equal to the covering radius of the lattice (**too complex**)

Choice of the sphere radius

- The radius is a critical parameter for the complexity of the algorithm
  - **A too small radius** : no point inside the sphere
  - **A too large radius** : too many points inside the sphere, which increases the algorithm complexity
- A good solution is to have the sphere radius equal to the covering radius of the lattice (**too complex**)
- An easier solution is to choose

$$C = \min\left(\min_i\left(\left(\mathrm{diag}\boldsymbol{M}\cdot\boldsymbol{M}^\top\right)_i\right), 2n\sigma^2\right)$$

Decoding of a finite part of a lattice

- **First idea** : add a routine which tests if a candidate point belongs or not to the constellation. Too complex.

Decoding of a finite part of a lattice

- **First idea** : add a routine which tests if a candidate point belongs or not to the constellation. Too complex.
- Decode the constellation with the same complexity as the lattice
- Only visit points inside the constellation

Decoding of a finite part of a lattice

- **First idea** : add a routine which tests if a candidate point belongs or not to the constellation. Too complex.

- Decode the constellation with the same complexity as the lattice
- Only visit points inside the constellation

16 – QAM

redefine intervals

$$I = I_i \cap I_C = \left[\sup(b_{inf,i}, c_{min}), \inf(b_{sup,i}, c_{max})\right]$$

where $I_C = \left[c_{min}, c_{max}\right] = [0,3]$ is the set of the in phase and quadrature components of the constellation.

Lattice Decoding

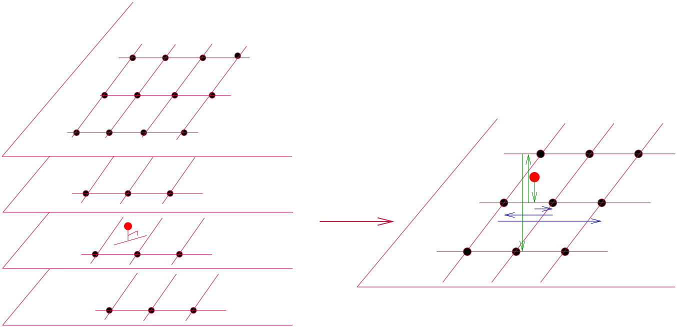
Schnorr-Euchner algorithm (1)

- It is a variant of the Sphere Decoder (SD)
- Same principle than SD applies, that is, search the closest point inside a sphere centered on the received point.

Schnorr-Euchner algorithm (1)

- It is a variant of the Sphere Decoder (SD)
- Same principle than SD applies, that is, search the closest point inside a sphere centered on the received point.
- The main idea of SE is to see the set of $n-$dimensional points ($n-$dimensional lattice) as a superposition of $(n-1)-$dimensional points (in hyperplans).
- The closest point is found by successive projections on hyperplans.
- We need a starting point in the lattice.

Example of a 3 dimensional lattice

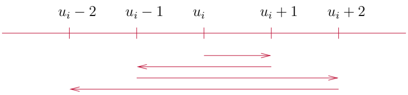


Schnorr-Euchner algorithm (2)

- The starting point is called "**Babai point**". It results from a suboptimal decoding.
- Starting from the Babai point, the algorithm visits the other lattice points inside the sphere centered on the received point, and whose radius is given by the distance between the **Babai point** and the received point.

Schnorr-Euchner algorithm (2)

- The starting point is called "**Babai point**". It results from a suboptimal decoding.
- Starting from the Babai point, the algorithm visits the other lattice points inside the sphere centered on the received point, and whose radius is given by the distance between the **Babai point** and the received point.
- We visit all points inside the sphere, zigzaging around each component of the Babai point



Comparison SD/SE

Similarities

- same principle : search the closest point inside a sphere
- same performance : ML

Comparison SD/SE

Similarities

- same principle : search the closest point inside a sphere
- same performance : ML

Differences

- **Strategies are different**
  - SD : points are visited from the boundary of the sphere towards its center
  - SE : points are visited from the center of the sphere towards its boundaries
- **Sphere radius**
  - SD : needs to initialize the radius
  - SE : no initial radius to choose

Part III

Preprocessing

Preprocessing

A more general problem formulation

Definition

A lattice code $\mathscr{C}(\Lambda, \boldsymbol{t}, \mathscr{S})$ is the set of points of $\Lambda + \boldsymbol{t}$ inside the shaping region $\mathscr{S}$ that is,

$$\mathscr{C}(\Lambda, \boldsymbol{t}, \mathscr{S}) = \{\Lambda + \boldsymbol{t}\} \cap \mathscr{S}$$

- The considered communication model is

$$\boxed{\boldsymbol{y} = \boldsymbol{H} \cdot (\boldsymbol{x} + \boldsymbol{t}) + \boldsymbol{w}}$$

where $\boldsymbol{x} = \boldsymbol{\Phi} \cdot \boldsymbol{u}$, $\boldsymbol{u} \in \mathbb{Z}^m$ and $\boldsymbol{H} \in \mathbb{R}^{n \times m}$.

- $\boldsymbol{\Phi}$ is the precoding matrix.

A more general problem formulation

Definition

A lattice code $\mathscr{C}(\Lambda, \boldsymbol{t}, \mathscr{S})$ is the set of points of $\Lambda + \boldsymbol{t}$ inside the shaping region $\mathscr{S}$ that is,

$$\mathscr{C}(\Lambda, \boldsymbol{t}, \mathscr{S}) = \{\Lambda + \boldsymbol{t}\} \cap \mathscr{S}$$

- The considered communication model is

$$\boxed{\boldsymbol{y} = \boldsymbol{H} \cdot (\boldsymbol{x} + \boldsymbol{t}) + \boldsymbol{w}}$$

  where $\boldsymbol{x} = \boldsymbol{\Phi} \cdot \boldsymbol{u}$, $\boldsymbol{u} \in \mathbb{Z}^m$ and $\boldsymbol{H} \in \mathbb{R}^{n \times m}$.
- $\boldsymbol{\Phi}$ is the precoding matrix.

Decoding problem

Find

$$\boxed{\hat{\boldsymbol{u}} = \arg \min_{\boldsymbol{u} \in \mathscr{U} \subset \mathbb{Z}^m} \left\| \boldsymbol{y} - \boldsymbol{H} \cdot \boldsymbol{t} - \boldsymbol{H} \cdot \boldsymbol{\Phi} \cdot \boldsymbol{u} \right\|^2} \quad (2)$$

Why preprocessing?

- Applications of sphere decoding suffers from two inconveniences
  1. When $\text{rank}(\boldsymbol{H}\cdot\boldsymbol{\Phi}) < m$ or $\boldsymbol{H}\cdot\boldsymbol{\Phi}$ is ill-conditioned the spread of the diagonal elements of $\boldsymbol{H}\cdot\boldsymbol{\Phi}$ is large and the search can be very complex.
  2. Enforcing $\boldsymbol{u}$ is very difficult when constellation $\mathscr{U}$ has a complicated shape
  3. Lattice decoding can solve this problem by searching over $\mathbb{Z}^m$ (instead of $\mathscr{U}$) but it is far from ML in general.

Why preprocessing?

- Applications of sphere decoding suffers from two inconveniences
  1. When $\text{rank}(\boldsymbol{H}\cdot\boldsymbol{\Phi}) < m$ or $\boldsymbol{H}\cdot\boldsymbol{\Phi}$ is ill-conditioned the spread of the diagonal elements of $\boldsymbol{H}\cdot\boldsymbol{\Phi}$ is large and the search can be very complex.
  2. Enforcing $\boldsymbol{u}$ is very difficult when constellation $\mathscr{U}$ has a complicated shape
  3. Lattice decoding can solve this problem by searching over $\mathbb{Z}^m$ (instead of $\mathscr{U}$) but it is far from ML in general.

- **Solution:** Preprocessing!
- In addition, preprocessing $\boldsymbol{H}$ and $\boldsymbol{\Phi}$ can have a great effect on the complexity of the search stage to make the tree more "friendly" (improving the quality of the ZF-DFE).

The preprocessing stage

- ***Left Preprocessing*** ($\rightarrow \times \boldsymbol{H}$): Modifies $\boldsymbol{H}$ and $\boldsymbol{w}$ such that the resulting **CL**osest **P**oint **S**earch (CLPS) is not equivalent to **ML** but has a much better conditioned "channel" matrix and makes lattice decoding near-optimal.

The preprocessing stage

- ***Left Preprocessing*** ($\rightarrow \times \boldsymbol{H}$): Modifies $\boldsymbol{H}$ and $\boldsymbol{w}$ such that the resulting **CL**osest **P**oint **S**earch (CLPS) is not equivalent to **ML** but has a much better conditioned "channel" matrix and makes lattice decoding near-optimal.
- ***Right preprocessing*** ($\boldsymbol{\Phi}\times \leftarrow$): When boundary region is removed, we have the freedom of choosing the lattice basis which is more convenient for the search algorithm.

The preprocessing stage

- ***Left Preprocessing*** ($\rightarrow \times \boldsymbol{H}$): Modifies $\boldsymbol{H}$ and $\boldsymbol{w}$ such that the resulting **CL**osest **P**oint **S**earch (CLPS) is not equivalent to **ML** but has a much better conditioned "channel" matrix and makes lattice decoding near-optimal.
- ***Right preprocessing*** ($\boldsymbol{\Phi} \times \leftarrow$): When boundary region is removed, we have the freedom of choosing the lattice basis which is more convenient for the search algorithm.

Preprocessing

Left preprocessing applied only on the channel matrix; right preprocessing applied on the whole. **Important:** any preprocessing should not destruct the code structure

Preprocessing

The QR decomposition

- QR decomposition applies to $\boldsymbol{H}$.

$$\mathrm{H} = \mathrm{Q} \cdot \mathrm{R}$$

Unitary Matrix (Q) Triangular Matrix (R)

- It can be seen as ZF-DFE with
  - Feedforward matrix $\boldsymbol{Q}$
  - Backward matrix $\boldsymbol{R}$

- When $\boldsymbol{y} = \boldsymbol{H} \cdot \boldsymbol{x} + \boldsymbol{w}$, CLPS is $\boxed{\min_{\boldsymbol{x}} \left\| \boldsymbol{y} - \boldsymbol{H} \cdot \boldsymbol{x} \right\|^2}$ equivalent to $\boxed{\min_{\boldsymbol{X}} \left\| \boldsymbol{Q}^{\dagger} \cdot \boldsymbol{y} - \boldsymbol{R} \cdot \boldsymbol{x} \right\|^2}$.

  Hence, a tree of the channel can be constructed.

The MMSE-DFE (1)

- **MMSE-DFE** outperforms **ZF-DFE** in terms of SINR

$$\tilde{H} \triangleq \begin{bmatrix} H \\ I \end{bmatrix} = \tilde{Q} \cdot R_1$$

The MMSE-DFE (1)

- **MMSE-DFE** outperforms **ZF-DFE** in terms of SINR

$$\boxed{\tilde{\boldsymbol{H}} \triangleq \begin{bmatrix} \boldsymbol{H} \\ \boldsymbol{I} \end{bmatrix} = \tilde{\boldsymbol{Q}} \cdot \boldsymbol{R}_1}$$

- Let $\boldsymbol{Q}_1$ be the upper $n \times m$ part of $\tilde{\boldsymbol{Q}}$. Transformed CLPS is

$$\min_{\boldsymbol{u} \in \mathcal{U}} \left\| \boldsymbol{Q}_1^\dagger \cdot \boldsymbol{r} - \boldsymbol{R}_1 \cdot \boldsymbol{\Phi} \cdot \boldsymbol{u} \right\|^2$$

which is not equivalent to (2) with $\boldsymbol{r} = \boldsymbol{y} - \boldsymbol{H} \cdot \boldsymbol{t}$ since $\boldsymbol{Q}_1$ is not unitary.

The MMSE-DFE (2)

- We have

$$\begin{aligned} \boldsymbol{Q}_1^{\dagger} \cdot \boldsymbol{r} &= \boldsymbol{Q}_1^{\dagger} \cdot \boldsymbol{H} \cdot \boldsymbol{\Phi} \cdot \boldsymbol{u} + \boldsymbol{Q}_1^{\dagger} \cdot \boldsymbol{w} \\ &= \boldsymbol{R}_1 \cdot \boldsymbol{\Phi} \cdot \boldsymbol{u} + \boldsymbol{z} \end{aligned}$$

- The additive noise $\boldsymbol{z} = \boldsymbol{Q}_1^{\dagger} \cdot \boldsymbol{r} - \boldsymbol{R}_1 \cdot \boldsymbol{\Phi} \cdot \boldsymbol{u}$ has a Gaussian component $\boldsymbol{Q}_1^{\dagger} \cdot \boldsymbol{w}$ and a non-Gaussian (signal dependent) component $\left(\boldsymbol{Q}_1^{\dagger} \cdot \boldsymbol{H} - \boldsymbol{R}_1\right) \cdot \boldsymbol{x}$.

The MMSE-DFE (2)

- We have

$$\boldsymbol{Q}_1^\dagger \cdot \boldsymbol{r} = \boldsymbol{Q}_1^\dagger \cdot \boldsymbol{H} \cdot \boldsymbol{\Phi} \cdot \boldsymbol{u} + \boldsymbol{Q}_1^\dagger \cdot \boldsymbol{w}$$
$$= \boldsymbol{R}_1 \cdot \boldsymbol{\Phi} \cdot \boldsymbol{u} + \boldsymbol{z}$$

- The additive noise $\boldsymbol{z} = \boldsymbol{Q}_1^\dagger \cdot \boldsymbol{r} - \boldsymbol{R}_1 \cdot \boldsymbol{\Phi} \cdot \boldsymbol{u}$ has a Gaussian component $\boldsymbol{Q}_1^\dagger \cdot \boldsymbol{w}$ and a non-Gaussian (signal dependent) component $\left(\boldsymbol{Q}_1^\dagger \cdot \boldsymbol{H} - \boldsymbol{R}_1\right) \cdot \boldsymbol{x}$.

The noise is white!!

$$\boxed{\left[\boldsymbol{R}_1 - \boldsymbol{Q}_1^\dagger \cdot \boldsymbol{H}\right]\left[\boldsymbol{R}_1 - \boldsymbol{Q}_1^\dagger \cdot \boldsymbol{H}\right]^\dagger + \boldsymbol{Q}_1^\dagger \cdot \boldsymbol{Q}_1 = \boldsymbol{I}}$$

The MMSE-DFE (3)

- When dimension goes to infinity, then noise W tends to be Gaussian. But why it works even for finite dimensions and finite SNRs is still an **open problem**.

The MMSE-DFE (3)

- When dimension goes to infinity, then noise $\boldsymbol{W}$ tends to be Gaussian. But why it works even for finite dimensions and finite SNRs is still an **open problem**.
- MMSE-DFE followed by optimal search is ML when $\boldsymbol{\Phi} = \boldsymbol{I}$ and the constellation is constant modulus (QPSK)

The MMSE-DFE (3)

- When dimension goes to infinity, then noise $\boldsymbol{W}$ tends to be Gaussian. But why it works even for finite dimensions and finite SNRs is still an **open problem**.
- MMSE-DFE followed by optimal search is ML when $\boldsymbol{\Phi} = \boldsymbol{I}$ and the constellation is constant modulus (QPSK)
- We can solve under-determined linear systems since matrix $\boldsymbol{R}_1$ is always full rank with eigenvalues ≥ 1.

The MMSE-DFE (3)

- When dimension goes to infinity, then noise $\boldsymbol{W}$ tends to be Gaussian. But why it works even for finite dimensions and finite SNRs is still an **open problem**.
- MMSE-DFE followed by optimal search is ML when $\boldsymbol{\Phi} = \boldsymbol{I}$ and the constellation is constant modulus (QPSK)
- We can solve under-determined linear systems since matrix $\boldsymbol{R}_1$ is always full rank with eigenvalues ≥ 1.
- MMSE-DFE atenuates the problem of **boundary control** in the next steps.

Preprocessing

Right preprocessing (1)

- When left preprocessing has been done, we need to QR-decompose matrix

$$\boldsymbol{R}_1 \cdot \Phi = \boldsymbol{Q} \cdot \boldsymbol{R}$$

We want $\boldsymbol{R}$ to be as sparse as possible (e.g. $\boldsymbol{R} \to \boldsymbol{I}$)

Right preprocessing (1)

- When left preprocessing has been done, we need to QR-decompose matrix

$$\boldsymbol{R}_1 \cdot \boldsymbol{\Phi} = \boldsymbol{Q} \cdot \boldsymbol{R}$$

We want $\boldsymbol{R}$ to be as sparse as possible (e.g. $\boldsymbol{R} \to \boldsymbol{I}$)

Problem

Find a unimodular matrix $\boldsymbol{T}$ such that QR decomposition $\boldsymbol{R}_1 \cdot \boldsymbol{\Phi} \cdot \boldsymbol{T}^{-1}$ minimizes the sparsity index of R.

$$\varsigma(\boldsymbol{R}) \triangleq \max_{k=1,2,\ldots,m} \frac{\sum_{i=k+1}^{m} r_{i,j}^2}{r_{i,i}^2}$$

Right preprocessing (1)

- When left preprocessing has been done, we need to QR-decompose matrix

$$\boldsymbol{R}_1 \cdot \boldsymbol{\Phi} = \boldsymbol{Q} \cdot \boldsymbol{R}$$

We want $\boldsymbol{R}$ to be as sparse as possible (e.g. $\boldsymbol{R} \rightarrow \boldsymbol{I}$)

Problem

Find a unimodular matrix $\boldsymbol{T}$ such that QR decomposition $\boldsymbol{R}_1 \cdot \boldsymbol{\Phi} \cdot \boldsymbol{T}^{-1}$ minimizes the sparsity index of R.

$$\varsigma(\boldsymbol{R}) \triangleq \max_{k=1,2,\ldots,m} \frac{\sum_{i=k+1}^{m} r_{i,j}^2}{r_{i,i}^2}$$

- Good approximations to the solutions of this problem exist

Right Preprocessing (2)

- **Lattice reduction:** **L**enstra, **L**enstra and **L**ovász (**LLL**) algorithm (possibly with deep insertion [Schnorr-Euchner]). Find a new lattice basis with reduced vectors $\boldsymbol{H} \cdot \boldsymbol{\Phi} \cdot \boldsymbol{T}_1^{-1}$ (i.e., small norms and/or as orthogonal as possible).
- Column permutation $\boldsymbol{\Pi}$ of $\boldsymbol{H} \cdot \boldsymbol{\Phi} \cdot \boldsymbol{T}_1^{-1}$ such that $\min_i r_{i,i}$ is maximized.
- Right multiply by

$$\boldsymbol{T}^{-1} = \boldsymbol{T}_1^{-1} \cdot \boldsymbol{\Pi}^{-1}$$

- Right multiplication by unimodular matrices does not alter lattice decoding.

Form the tree of the system

- We give $\boldsymbol{H}$ the channel matrix and $\boldsymbol{\Phi}$ the precoding matrix (after vectorization).
- Perform left and right preprocessing
- QR-decompose $\boxed{\boldsymbol{Q}_1^{\dagger} \cdot \boldsymbol{H} \cdot \boldsymbol{\Phi} \cdot \boldsymbol{T}^{-1} = \boldsymbol{Q} \cdot \boldsymbol{R}}$

Form the tree of the system

- We give $\boldsymbol{H}$ the channel matrix and $\boldsymbol{\Phi}$ the precoding matrix (after vectorization).
- Perform left and right preprocessing
- QR-decompose $\boxed{\boldsymbol{Q}_1^{\dagger} \cdot \boldsymbol{H} \cdot \boldsymbol{\Phi} \cdot \boldsymbol{T}^{-1} = \boldsymbol{Q} \cdot \boldsymbol{R}}$

Equivalent System

With convenient notations, we get,

$$\boxed{\begin{pmatrix} y_m \\ \vdots \\ \vdots \\ y_1 \end{pmatrix} = \begin{pmatrix} r_{m,m} & \cdots & \cdots & r_{m,1} \\ 0 & r_{m-1,m-1} & \cdots & r_{m-1,1} \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & r_{1,1} \end{pmatrix} \cdot \begin{pmatrix} x_m \\ \vdots \\ \vdots \\ x_1 \end{pmatrix} + \begin{pmatrix} w_m \\ \vdots \\ \vdots \\ w_1 \end{pmatrix}} \tag{3}$$

Preprocessing

System model

Assumptions

- Rayleigh Flat Fading Channel
- MISO system
- DAST Codes used

System model

Assumptions

- Rayleigh Flat Fading Channel
- MISO system
- DAST Codes used

What are the parameters?

$\boldsymbol{H}$ is a diagonal matrix and $\boldsymbol{\Phi}$ is a unitary transform defined on a number field (rows of $\boldsymbol{\Phi}$ are conjugated).

System model

Assumptions

- Rayleigh Flat Fading Channel
- MISO system
- DAST Codes used

What are the parameters?

$\boldsymbol{H}$ is a diagonal matrix and $\boldsymbol{\Phi}$ is a unitary transform defined on a number field (rows of $\boldsymbol{\Phi}$ are conjugated).

Received signal and aim of this section

$$\boldsymbol{y} = \boldsymbol{H} \cdot \boldsymbol{\Phi} \cdot \boldsymbol{x} + \boldsymbol{n} \tag{4}$$

where

$$\boldsymbol{H} = \operatorname{diag}[h_1, h_2, \ldots, h_n]$$

$\boldsymbol{n}$ is the i.i.d. Gaussian noise and $\boldsymbol{\Phi}$ is a unitary transform bringing modulation diversity to the system. The aim is to design a “not too complex” detector by doing some new lattice reduction.

Assumptions on the unitary transform

Background

- We use $\mathbb{F} = \mathbb{Q}(i)$ as the base field with ring of integer $\mathbb{Z}[i]$ (QAM).
- $\mathbb{K} = \mathbb{F}(\theta)$ is the smallest field containing $\mathbb{F}$ and θ, an element of order n. Its ring of integers is $\mathcal{O}_{\mathbb{K}}$.
- $Gal_{\mathbb{K}/\mathbb{F}}$ is the Galois group of automorphisms on $\mathbb{K}$ with elements denoted $\sigma_i, i = 1, \ldots, n$.

Assumptions on the unitary transform

Background

- We use $\mathbb{F} = \mathbb{Q}(i)$ as the base field with ring of integer $\mathbb{Z}[i]$ (QAM).
- $\mathbb{K} = \mathbb{F}(\theta)$ is the smallest field containing $\mathbb{F}$ and θ, an element of order n. Its ring of integers is $\mathcal{O}_{\mathbb{K}}$.
- $Gal_{\mathbb{K}/\mathbb{F}}$ is the Galois group of automorphisms on $\mathbb{K}$ with elements denoted $\sigma_i, i = 1, \ldots, n$.

The Φ matrix (diversity of modulation)

The structure of $\mathbf{\Phi}$ is the following,

$$\mathbf{\Phi} = \mathbf{\Delta} \cdot \begin{bmatrix} \sigma_1(\omega_1) & \sigma_1(\omega_2) & \cdots & \sigma_1(\omega_n) \\ \sigma_2(\omega_1) & \sigma_2(\omega_2) & \cdots & \sigma_2(\omega_n) \\ \vdots & \vdots & \ddots & \vdots \\ \sigma_n(\omega_1) & \sigma_n(\omega_2) & \cdots & \sigma_n(\omega_n) \end{bmatrix} \tag{5}$$

where $\omega_1, \omega_2, \ldots, \omega_n \in \mathcal{O}_{\mathbb{K}}$ are linearly independent on $\mathbb{F}$. $\mathbf{\Delta}$ is diagonal.

A 2 dimensional example

Background

$\mathbb{K} = \mathbb{F}(\theta)$ with $\theta = \frac{1+\sqrt{5}}{2}$, an element of order 2. Its ring of integers is $\mathcal{O}_{\mathbb{K}} = \mathbb{Z}[i, \frac{1+\sqrt{5}}{2}]$.
Minimal polynomial of θ is $\mu_\theta(X) = X^2 - X - 1$.

- $Gal_{\mathbb{K}/\mathbb{F}}$ is the Galois group of $\mathbb{K}$ with elements $\{1, \sigma\}$ such that

$$\sigma : \theta \longmapsto \bar{\theta} = \frac{1-\sqrt{5}}{2}$$

A 2 dimensional example

Background

$\mathbb{K} = \mathbb{F}(\theta)$ with $\theta = \frac{1+\sqrt{5}}{2}$, an element of order 2. Its ring of integers is $\mathcal{O}_{\mathbb{K}} = \mathbb{Z}[i, \frac{1+\sqrt{5}}{2}]$.
Minimal polynomial of θ is $\mu_\theta(X) = X^2 - X - 1$.

- $Gal_{\mathbb{K}/\mathbb{F}}$ is the Galois group of $\mathbb{K}$ with elements $\{1, \sigma\}$ such that

$$\sigma : \theta \longmapsto \bar{\theta} = \frac{1-\sqrt{5}}{2}$$

The Φ matrix ("Golden Field")

Take $\omega_1 = 1 + i(1-\theta)$ and $\omega_2 = \theta - i$. Then,

$$\mathbf{\Phi} = \frac{1}{\sqrt{5}} \begin{bmatrix} \omega_1 & \omega_2 \\ \sigma(\omega_1) & \sigma(\omega_2) \end{bmatrix} = \frac{1}{\sqrt{5}} \begin{bmatrix} 1 + i(1-\theta) & \theta - i \\ 1 + i(1-\bar{\theta}) & \bar{\theta} - i \end{bmatrix}$$

Matrix representation of an algebraic number

Example 1: $\mathbb{C} \longrightarrow \mathcal{M}_2(\mathbb{R})$

$$z = x + iy \longmapsto \mathbf{T}_z = \begin{pmatrix} x & -y \\ y & x \end{pmatrix}$$

$$N_{\mathbb{C}/\mathbb{R}}(z) = x^2 + y^2 = \det \begin{pmatrix} x & -y \\ y & x \end{pmatrix}$$

Matrix representation of an algebraic number

Example 1: $\mathbb{C} \longrightarrow \mathcal{M}_2(\mathbb{R})$

$$z = x + iy \longmapsto \mathbf{T}_z = \begin{pmatrix} x & -y \\ y & x \end{pmatrix}$$

$$N_{\mathbb{C}/\mathbb{R}}(z) = x^2 + y^2 = \det \begin{pmatrix} x & -y \\ y & x \end{pmatrix}$$

Example 2: $\mathbb{Q}\left(e^{\frac{i\pi}{4}}\right) \longrightarrow \mathcal{M}_2(\mathbb{Q}(i))$

$$z = x + y\theta \longmapsto \mathbf{T}_z = \begin{pmatrix} x & iy \\ y & x \end{pmatrix}$$

$$N_{\mathbb{Q}\left(e^{\frac{i\pi}{4}}\right)/\mathbb{Q}(i)}(z) = x^2 - iy^2 = \det \begin{pmatrix} x & iy \\ y & x \end{pmatrix}$$

Transforming Fadings into a Basis Change (1)

- Matrix H can be expressed as

$$\boldsymbol{H} = \left|\prod_{i=1}^{n} h_i\right|^{\frac{1}{n}} \cdot \operatorname{diag}\left[a_1, a_2, \ldots, a_n\right]$$

with $\left|\prod_{i=1}^{n} a_i\right| = 1$.

Transforming Fadings into a Basis Change (1)

- Matrix H can be expressed as

$$\boldsymbol{H} = \left|\prod_{i=1}^{n} h_i\right|^{\frac{1}{n}} \cdot \mathrm{diag}\left[a_1, a_2, \ldots, a_n\right]$$

with $\left|\prod_{i=1}^{n} a_i\right| = 1$.

- **Assume** that the vector $(|a_1|, |a_2|, \ldots, |a_n|)$ is composed by the magnitudes of the conjugates of some unit u in $\mathcal{O}_{\mathbb{K}}$, i.e., $a_k = e^{i\beta_k}\sigma_k(u), \forall k$ with $\beta_k = \arg a_k - \arg\sigma_k(u)$. The received signal can then be expressed as

$$\boldsymbol{y} = \left|\prod_{i=1}^{n} h_i\right|^{\frac{1}{n}} \cdot \mathrm{diag}\left[e^{i\beta_1}, \ldots, e^{i\beta_n}\right] \cdot \mathrm{diag}\left[\sigma_1(u), \ldots, \sigma_n(u)\right] \cdot \boldsymbol{\Phi} \cdot \boldsymbol{x} + \boldsymbol{n} \qquad (6)$$

Transforming Fadings into a Basis Change (1)

- Matrix H can be expressed as

$$\boldsymbol{H} = \left| \prod_{i=1}^{n} h_i \right|^{\frac{1}{n}} \cdot \operatorname{diag}[a_1, a_2, \ldots, a_n]$$

with $\left| \prod_{i=1}^{n} a_i \right| = 1$.

- **Assume** that the vector $(|a_1|, |a_2|, \ldots, |a_n|)$ is composed by the magnitudes of the conjugates of some unit u in $\mathcal{O}_{\mathbb{K}}$, i.e., $a_k = e^{i\beta_k} \sigma_k(u), \forall k$ with $\beta_k = \arg a_k - \arg \sigma_k(u)$. The received signal can then be expressed as

$$\boldsymbol{y} = \left| \prod_{i=1}^{n} h_i \right|^{\frac{1}{n}} \cdot \operatorname{diag}\left[e^{i\beta_1}, \ldots, e^{i\beta_n}\right] \cdot \operatorname{diag}[\sigma_1(u), \ldots, \sigma_n(u)] \cdot \boldsymbol{\Phi} \cdot \boldsymbol{x} + \boldsymbol{n} \quad (6)$$

- So,

$$\boldsymbol{y} = \left| \prod_{i=1}^{n} h_i \right|^{\frac{1}{n}} \cdot \boldsymbol{\Psi} \cdot \boldsymbol{\Phi} \cdot \boldsymbol{T}_u \cdot \boldsymbol{x} + \boldsymbol{n} \quad (7)$$

with $\boldsymbol{\Psi} = \operatorname{diag}\left[e^{i\beta_1}, e^{i\beta_2}, \ldots, e^{i\beta_n}\right]$ and $\boldsymbol{T}_u$ (**unimodular**) being **the matrix representation** of the unit u.

Transforming Fadings into a Basis Change (2)

- Denote $\boldsymbol{z} = \left|1/\prod_{i=1}^{n} h_i\right|^{\frac{1}{n}} \cdot \boldsymbol{\Phi}^{\dagger} \cdot \boldsymbol{\Psi}^{\dagger} \cdot \boldsymbol{y}$, then

$$\boldsymbol{z} = \boldsymbol{T}_u \cdot \boldsymbol{x} + \boldsymbol{w}$$

where $\boxed{\boldsymbol{w} = \left|1/\prod_{i=1}^{n} h_i\right|^{\frac{1}{n}} \cdot \boldsymbol{\Phi}^{\dagger} \cdot \boldsymbol{\Psi}^{\dagger} \cdot \boldsymbol{n}}$ remains an *i.i.d.* noise vector.

- Now, since $\boxed{|\det \boldsymbol{T}_u| = 1}$, ($u$ is a unit), then a ML lattice decoder is obvious as it is a slicer followed by the product with matrix $\boldsymbol{T}_u^{-1}$.

Transforming Fadings into a Basis Change (2)

- Denote $\boldsymbol{z} = \left|1/\prod_{i=1}^{n} h_i\right|^{\frac{1}{n}} \cdot \boldsymbol{\Phi}^{\dagger} \cdot \boldsymbol{\Psi}^{\dagger} \cdot \boldsymbol{y}$, then

$$\boldsymbol{z} = \boldsymbol{T}_u \cdot \boldsymbol{x} + \boldsymbol{w}$$

where $\boxed{\boldsymbol{w} = \left|1/\prod_{i=1}^{n} h_i\right|^{\frac{1}{n}} \cdot \boldsymbol{\Phi}^{\dagger} \cdot \boldsymbol{\Psi}^{\dagger} \cdot \boldsymbol{n}}$ remains an *i.i.d.* noise vector.

- Now, since $\boxed{|\det \boldsymbol{T}_u| = 1}$, ($u$ is a unit), then a ML lattice decoder is obvious as it is a slicer followed by the product with matrix $\boldsymbol{T}_u^{-1}$.

Approximation

What happens if $(|a_1|, |a_2|, \ldots, |a_n|)$ is not composed by the modules of conjugates of some unit u?

The Logarithmic Lattice

Theorem (Dirichlet)

Let $\mathbb{K}$ be an extension of $\mathbb{Q}$ with signature (r,s) (with degree $r+2s$). Then there exists $r+s-1$ units named "***fundamental units***" $u_1, u_2, \ldots, u_{r+s-1}$ such that any unit u can be expressed as

$$u = \epsilon \cdot \prod_{i=1}^{r+s-1} u_i^{k_i}$$

where ϵ is a complex number with module equal to 1 and $k_i \in \mathbb{Z}$.

The Logarithmic Lattice

Theorem (Dirichlet)

Let $\mathbb{K}$ be an extension of $\mathbb{Q}$ with signature (r,s) (with degree $r+2s$). Then there exists $r+s-1$ units named "***fundamental units***" $u_1, u_2, \ldots, u_{r+s-1}$ such that any unit u can be expressed as

$$u = \epsilon \cdot \prod_{i=1}^{r+s-1} u_i^{k_i}$$

where ϵ is a complex number with module equal to 1 and $k_i \in \mathbb{Z}$.

- Now from a unit u, construct the vector

$$\boldsymbol{u}_{\log} = \left(\log|\sigma_1(u)|, \ldots, \log|\sigma_{r+s}(u)|\right)^\top$$

Then vector $\boldsymbol{u}_{\log}$ lies in a hyperplane with equation

$$\sum_{i=1}^{r+s} x_i = 0$$

The Logarithmic Lattice

Theorem (Dirichlet)

Let $\mathbb{K}$ be an extension of $\mathbb{Q}$ with signature (r, s) (with degree $r + 2s$). Then there exists $r + s - 1$ units named "***fundamental units***" $u_1, u_2, \ldots, u_{r+s-1}$ such that any unit u can be expressed as

$$u = \epsilon \cdot \prod_{i=1}^{r+s-1} u_i^{k_i}$$

where ϵ is a complex number with module equal to 1 and $k_i \in \mathbb{Z}$.

- Now from a unit u, construct the vector
$$\boldsymbol{u}_{\log} = \left(\log|\sigma_1(u)|, \ldots, \log|\sigma_{r+s}(u)|\right)^\top$$
Then vector $\boldsymbol{u}_{\log}$ lies in a hyperplane with equation
$$\sum_{i=1}^{r+s} x_i = 0$$
- All vectors of type $\boldsymbol{u}_{\log}$ are in a lattice named the logarithmic lattice, with generator matrix,
$$\begin{bmatrix} \log|\sigma_1(u_1)| & \log|\sigma_2(u_1)| & \cdots & \log|\sigma_{r+s}(u_1)| \\ \log|\sigma_1(u_2)| & \log|\sigma_2(u_2)| & \cdots & \log|\sigma_{r+s}(u_2)| \\ \vdots & \vdots & \ddots & \vdots \\ \log|\sigma_1(u_{r+s-1})| & \log|\sigma_2(u_{r+s-1})| & \cdots & \log|\sigma_{r+s}(u_{r+s-1})| \end{bmatrix}$$
and fundamental volume R, the regulator.

Golden Field

- Let $\mathbb{K} = \mathbb{Q}(i, \sqrt{5})$ with $\mathbf{\Phi} = \frac{1}{\sqrt{5}} \begin{bmatrix} 1+i(1-\theta) & \theta - i \\ 1+i(1-\bar{\theta}) & \bar{\theta} - i \end{bmatrix}$ and $\theta = \frac{1+\sqrt{5}}{2}$. The logarithmic lattice ($\cong \mathbb{Z}$) has generator matrix $\begin{bmatrix} 0.481 & -0.481 \end{bmatrix}$.

$$\boldsymbol{H} = \begin{pmatrix} h_1 & 0 \\ 0 & h_2 \end{pmatrix}$$

- Assume that fadings are $\boxed{h_1 = 1.271 e^{i\eta_1}}$ and $\boxed{h_2 = 0.071 e^{i\eta_2}}$. We get

$$\boldsymbol{H} = \begin{pmatrix} h_1 & 0 \\ 0 & h_2 \end{pmatrix} = 0.3 \begin{pmatrix} e^{i\eta_1} & 0 \\ 0 & -e^{i\eta_2} \end{pmatrix} \begin{pmatrix} \theta^3 & 0 \\ 0 & \bar{\theta}^3 \end{pmatrix}$$

Golden Field

- Let $\mathbb{K} = \mathbb{Q}(i, \sqrt{5})$ with $\mathbf{\Phi} = \frac{1}{\sqrt{5}} \begin{bmatrix} 1+i(1-\theta) & \theta - i \\ 1+i(1-\bar{\theta}) & \bar{\theta} - i \end{bmatrix}$ and $\theta = \frac{1+\sqrt{5}}{2}$. The logarithmic lattice ($\cong \mathbb{Z}$) has generator matrix $\begin{bmatrix} 0.481 & -0.481 \end{bmatrix}$.

$$\boldsymbol{H} = \begin{pmatrix} h_1 & 0 \\ 0 & h_2 \end{pmatrix}$$

- Assume that fadings are $\boxed{h_1 = 1.271 e^{i\eta_1}}$ and $\boxed{h_2 = 0.071 e^{i\eta_2}}$. We get

$$\boldsymbol{H} = \begin{pmatrix} h_1 & 0 \\ 0 & h_2 \end{pmatrix} = 0.3 \begin{pmatrix} e^{i\eta_1} & 0 \\ 0 & -e^{i\eta_2} \end{pmatrix} \begin{pmatrix} \theta^3 & 0 \\ 0 & \bar{\theta}^3 \end{pmatrix}$$

Reduction

Equations (6) and (7) give

$$\boldsymbol{y} = \boldsymbol{H}\mathbf{\Phi}\boldsymbol{x} + \boldsymbol{n} = 0.3 \begin{pmatrix} e^{i\eta_1} & 0 \\ 0 & -e^{i\eta_2} \end{pmatrix} \begin{pmatrix} \theta^3 & 0 \\ 0 & \bar{\theta}^3 \end{pmatrix} \mathbf{\Phi}\boldsymbol{x} + \boldsymbol{n}$$

$$= 0.3 \begin{pmatrix} e^{i\eta_1} & 0 \\ 0 & -e^{i\eta_2} \end{pmatrix} \mathbf{\Phi} \begin{pmatrix} 1 & 2 \\ 2 & 3 \end{pmatrix} \boldsymbol{x} + \boldsymbol{n}$$

The Reduction

In fact, the true received signal is (after reduction)

$$\boldsymbol{y} = \left| \prod_{i=1}^{n} h_i \right|^{\frac{1}{n}} \cdot \boldsymbol{\Psi\Lambda\Phi T}_u \boldsymbol{x} + \boldsymbol{n}$$

The Reduction

In fact, the true received signal is (after reduction)

$$\boldsymbol{y} = \left| \prod_{i=1}^{n} h_i \right|^{\frac{1}{n}} \cdot \boldsymbol{\Psi \Lambda \Phi T}_u \boldsymbol{x} + \boldsymbol{n}$$

- If $\mathbf{H}$ is proportional to a unitary transform, then $\boldsymbol{\Lambda} = \mathbf{I}$, else, the nearest unit (in the logarithmic lattice) is chosen and $\boldsymbol{\Lambda}$ is a diagonal matrix whose dynamic is bounded (covering radius) and controlled by the logarithmic lattice.

The Reduction

In fact, the true received signal is (after reduction)

$$\boldsymbol{y} = \left| \prod_{i=1}^{n} h_i \right|^{\frac{1}{n}} \cdot \boldsymbol{\Psi \Lambda \Phi} \boldsymbol{T}_u \boldsymbol{x} + \boldsymbol{n}$$

- If $\mathbf{H}$ is proportional to a unitary transform, then $\boldsymbol{\Lambda} = \mathbf{I}$, else, the nearest unit (in the logarithmic lattice) is chosen and $\boldsymbol{\Lambda}$ is a diagonal matrix whose dynamic is bounded (covering radius) and controlled by the logarithmic lattice.
- $\boldsymbol{T}_u$ is the unimodular basis change matrix; it is **the** reduction matrix.

The diversity property (with ZF)

Theorem

The asymptotic ($\gamma \to \infty$) expression of the codeword error probability for the zero forcing detection is

$$P_e(\gamma) \leq O\left(\frac{\log^{n-1}\gamma}{\gamma^n}\right)$$

where n is the dimension

The diversity property (with ZF)

Theorem

The asymptotic ($\gamma \to \infty$) expression of the codeword error probability for the zero forcing detection is

$$P_e(\gamma) \leq O\left(\frac{\log^{n-1}\gamma}{\gamma^n}\right)$$

where n is the dimension

Sketch of the proof

- In fact, $P_e(\gamma) \leq \int_0^\infty p_X(x)e^{-\gamma x}dx$ where $X = \sqrt[n]{\prod_{i=1}^n X_i}$ (by using the covering radius of the logarithmic lattice as an upperbound)
- X_i are i.i.d. random variables with an exponential distribution
- Recursion on n gives the result.

Part IV

Tree Search Strategy

Tree Search Strategy

General Branch and Bound (1)

- We keep notations from (3). We have

$$\boldsymbol{y} = \boldsymbol{R} \cdot \boldsymbol{x} + \boldsymbol{z}$$

  with $\boldsymbol{x} \in \mathbb{Z}^m$.

- The node at level k is denoted $\boldsymbol{x}_1^k = (x_1, x_2, \ldots, x_k)$. Every node is associated to the metric

$$w_k\left(\boldsymbol{x}_1^k\right) = \left|\boldsymbol{y}_k - \sum_{i=1}^{k} r_{k,i} x_i\right|^2$$

- **B**ranch and **B**ound (**BB**) reduces the complexity of tree search by determining if an intermediate node $\boldsymbol{x}_1^k$ has any chance of giving the optimum leaf node, when extended.

General Branch and Bound (2)

- The decision is taken by comparing a cost function (namely $f\left(\boldsymbol{x}_1^k\right)$) against a bounding function t_k.
- BB maintains a list of valid nodes that can be extended, $\mathcal{N}$. BB ends when $\mathcal{N}$ is empty.
- Different BB algorithms differ in their cost functions, bounding functions and the rules to generate and sort the nodes.

General Branch and Bound (2)

- The decision is taken by comparing a cost function (namely $f\left(\boldsymbol{x}_1^k\right)$) against a bounding function t_k.
- BB maintains a list of valid nodes that can be extended, $\mathcal{N}$. BB ends when $\mathcal{N}$ is empty.
- Different BB algorithms differ in their cost functions, bounding functions and the rules to generate and sort the nodes.

A unified framework

BB brings a unified framework for many searching algorithms, considered as special cases. Sphere Decoder, Schnorr-Euchner, Sequential decoding, ...

Tree Search Strategy

10 Branch and Bound
- General Branch and Bound

11 Classification
- Breadth First Search
- Depth First Search
- Best First Search
- The best tradeoff

Breadth First Search (1)

- Bounding function is fixed and cost function is never updated.
- **Pohst Enumeration**: Bounding function is

$$t_k = C_0 \text{ (sphere radius)}.$$

  Cost function is

$$f\left(\boldsymbol{x}_1^k\right) = \sum_{i=1}^{k} w_i\left(\boldsymbol{x}_1^i\right) \leq C_0.$$

- All nodes satisfying $\sum_{i=1}^{k} w_i\left(\boldsymbol{x}_1^i\right) \leq C_0$ are generated before termination.

Breadth First Search (2)

- Generating the child nodes is simplified. For any parent node $\boldsymbol{x}_1^k$, the condition $\sum_{i=1}^{k} w_i\left(\boldsymbol{x}_1^i\right) \leq C_0$ implies for the child nodes that component k of the generated nodes lies in some interval (see SD).
- We can apply some heuristic statistical pruning, e.g. increased radii ($t_k < t_{k+1}$).
- Variants can be found. $M-$ and $T-$algorithms
  - $M-$algorithm only keeps the M best survivors whereas the $T-$algorithm adjusts the bounding function by the best cost function at the level k combined with a predefined threshold (use of the Fano metric, for instance).
- This algorithm remains complex.

Depth First Search

Principle

Order nodes in $\mathcal{N}$ in reverse order of generation. The final bound vector is

$$\boldsymbol{t} = \left[\min\left(t_1, f\left(\mathbf{x}_1^m\right)\right), \min\left(t_2, f\left(\mathbf{x}_1^m\right)\right), \ldots, \min\left(t_m, f\left(\mathbf{x}_1^m\right)\right)\right].$$

Different DFS depends on the way children are generated and on the partial bounds and the ordering of the generated child nodes.

- **Viterbo-Boutros algorithm:** We have $f\left(\boldsymbol{x}_1^k\right) = \sum_{i=1}^k w_i\left(\boldsymbol{x}_1^i\right)$ and for any node $\boldsymbol{x}_1^{k-1}$, its valid children (verifying $\sum_{i=1}^k w_i\left(\boldsymbol{x}_1^i\right) \leq C_0$) are generated lexicographically.
- **Schnorr-Euchner algorithm:** Same properties, but the child nodes are generated *w.r.t.* the accumulated squared distance $\sum_{i=1}^k w_i\left(\boldsymbol{x}_1^i\right)$.

Best First Search (1)

- Sort nodes in $\mathcal{N}$ such that their cost functions are increasing.
- Search can be terminated once a leaf node reaches the top of $\mathcal{N}$.
- Stack algorithm (sequential decoding) is BeFS with cost function

$$\boxed{f\left(\boldsymbol{x}_1^k\right) = \sum_{i=1}^{k+1} w_i\left(\boldsymbol{x}_{1,b}^{k+1}\right) - b(k+1)}$$

where $\boldsymbol{x}_{1,b}^{k+1}$ is the best child of $\boldsymbol{x}_1^k$ not generated yet and $f\left(\boldsymbol{x}_1^m\right) = -\infty$. b is the *bias*.

Theorem

The stack algorithm with $b = 0$ generates the least number of nodes among all optimal tree search algorithms.

Best First Search (2)

- The stack algorithm offers a natural solution for the problem of choosing the initial radius (or radii): $t_k = \infty$.
- The stack allows for a systematic approach for trading-off performance for complexity: $b = 0 \Rightarrow$ Optimal CLPS. $b = \infty \Rightarrow$ MMSE-Babai point decoder.
- In general, for systems with small dimension m, and/or high SNRs/"friendly" channels, one can obtain near-optimal performance with relatively large values of b (i.e., reduced complexity).
- **Disadvantage:** The required memory to maintain the active list $\mathcal{N}$ can be prohibitive.

The best tradeoff

- In general, the choice of the algorithm depends on the dimensions, codes, ...
- For a large variety of MIMO channel, the best tradeoff complexity/performance is given by

Best strategy

Left preprocessing (MMSE-DFE) + **Right** preprocessing (Lattice reduction and reordering), followed by a stack search stage in the lattice.

Conclusion and perspectives

- Sphere decoding has been the first ML decoding algorithm for MIMO (encoded) channels

Conclusion and perspectives

- Sphere decoding has been the first ML decoding algorithm for MIMO (encoded) channels
- Sequential decoding has also been proposed as a near optimal decoder but with much less complexity than SD.

Conclusion and perspectives

- Sphere decoding has been the first ML decoding algorithm for MIMO (encoded) channels
- Sequential decoding has also been proposed as a near optimal decoder but with much less complexity than SD.
- Preprocessing can decrease a lot the searching complexity

Conclusion and perspectives

- Sphere decoding has been the first ML decoding algorithm for MIMO (encoded) channels
- Sequential decoding has also been proposed as a near optimal decoder but with much less complexity than SD.
- Preprocessing can decrease a lot the searching complexity
- Now, do processing at the matrix form level and not at the vector form level.

Thank You