Computing a Minimal Input-State-Output Representation of Convolutional Codes

Joan Josep Climent, Diego Napp and **Verónica Requena**

Coding Theory and Cryptography
A conference in honor of Joachim Rosenthal's 60th birthday

Departament de Matemàtiques,
Universitat d'Alacant

11th july 2022

Outline

Overview

Motivation

- Obtain an algorithm which provide a minimal ISO representation of a k/n-rate convolutional code.
- Apply the previous algorithm to provide a minimal ISO representation of a convolutional product code.


Rosenthal, J., Schumacher, J.M., and York, E.V. (1996).
On behaviors and convolutional codes.
IEEE Transactions on Information Theory, 42(6), 1881–1891.


Rosenthal, J., York, F. V.
BCH convolutional codes (1999)
IEEE Transactions on Information Theory, 45(6), 1833–1844


Climent, J.-J. and Napp, D. and Pinto, R. and Requena, V.(2021).
Minimal State-Space Representation of Convolutional Product Codes.
Mathematics, 9(12).

Motivation

- Obtain an algorithm which provide a minimal ISO representation of a k/n-rate convolutional code.
- Apply the previous algorithm to provide a minimal ISO representation of a convolutional product code.


Rosenthal, J., Schumacher, J.M., and York, E.V. (1996).
On behaviors and convolutional codes.
IEEE Transactions on Information Theory, 42(6), 1881–1891.


Rosenthal, J., York, F. V.
BCH convolutional codes (1999)
IEEE Transactions on Information Theory, 45(6), 1833–1844


Climent, J.-J. and Napp, D. and Pinto, R. and Requena, V.(2021).
Minimal State-Space Representation of Convolutional Product Codes.
Mathematics, 9(12).

Motivation

- Obtain an algorithm which provide a minimal ISO representation of a k/n-rate convolutional code.
- Apply the previous algorithm to provide a minimal ISO representation of a convolutional product code.


Rosenthal, J., Schumacher, J.M., and York, E.V. (1996).
On behaviors and convolutional codes.
IEEE Transactions on Information Theory, 42(6), 1881–1891.


Rosenthal, J., York, F. V.
BCH convolutional codes (1999)
IEEE Transactions on Information Theory, 45(6), 1833–1844


Climent, J.-J. and Napp, D. and Pinto, R. and Requena, V.(2021).
Minimal State-Space Representation of Convolutional Product Codes.
Mathematics, 9(12).

Linear Block codes

Let $\mathbb{F}$ be a finite field and $n \in \mathbb{N}$. A linear block code of length n over $\mathbb{F}$ is a linear subspace $\mathcal{B}$ of $\mathbb{F}^n$. If $\dim \mathcal{B} = k$, then $\mathcal{B}$ is called an $[n, k]$-code and

$$\mathcal{B} = \left\{ \boldsymbol{v} \in \mathbb{F}^n \mid \boldsymbol{v} = \boldsymbol{u}G \text{ with } \boldsymbol{u} \in \mathbb{F}^k \right\}$$

where

- $G \in \mathbb{F}^{k \times n}$ with $\operatorname{rank}(G) = k$ is the **generator matrix** of $\mathcal{B}$,
- $\boldsymbol{u} \in \mathbb{F}^k$ is called **information word**,
- $\boldsymbol{v} \in \mathcal{B}$ is called **codeword**.

Equivalently, an $[n, k]$-code can be described as

$$\mathcal{B} = \left\{ \boldsymbol{v} \in \mathbb{F}^n \mid H\boldsymbol{v}^T = 0 \right\}$$

where $H \in \mathbb{F}^{(n-k) \times n}$ with $\operatorname{rank}(H) = n - k$ is called a **parity-check matrix** of $\mathcal{B}$.

Linear Block codes

Let $\mathbb{F}$ be a finite field and $n \in \mathbb{N}$. A linear block code of length n over $\mathbb{F}$ is a linear subspace $\mathcal{B}$ of $\mathbb{F}^n$. If $\dim \mathcal{B} = k$, then $\mathcal{B}$ is called an $[n, k]$-code and

$$\mathcal{B} = \left\{ \boldsymbol{v} \in \mathbb{F}^n \mid \boldsymbol{v} = \boldsymbol{u}G \text{ with } \boldsymbol{u} \in \mathbb{F}^k \right\}$$

where

- $G \in \mathbb{F}^{k \times n}$ with $\operatorname{rank}(G) = k$ is the **generator matrix** of $\mathcal{B}$,
- $\boldsymbol{u} \in \mathbb{F}^k$ is called **information word**,
- $\boldsymbol{v} \in \mathcal{B}$ is called **codeword**.

Equivalently, an $[n, k]$-code can be described as

$$\mathcal{B} = \left\{ \boldsymbol{v} \in \mathbb{F}^n \mid H\boldsymbol{v}^T = 0 \right\}$$

where $H \in \mathbb{F}^{(n-k) \times n}$ with $\operatorname{rank}(H) = n - k$ is called a **parity-check matrix** of $\mathcal{B}$.

Linear Block codes

Let $\mathbb{F}$ be a finite field and $n \in \mathbb{N}$. A linear block code of length n over $\mathbb{F}$ is a linear subspace $\mathcal{B}$ of $\mathbb{F}^n$. If $\dim \mathcal{B} = k$, then $\mathcal{B}$ is called an $[n, k]$-code and

$$\mathcal{B} = \left\{ \boldsymbol{v} \in \mathbb{F}^n \mid \boldsymbol{v} = \boldsymbol{u}G \text{ with } \boldsymbol{u} \in \mathbb{F}^k \right\}$$

where

- $G \in \mathbb{F}^{k \times n}$ with $\operatorname{rank}(G) = k$ is the **generator matrix** of $\mathcal{B}$,
- $\boldsymbol{u} \in \mathbb{F}^k$ is called **information word**,
- $\boldsymbol{v} \in \mathcal{B}$ is called **codeword**.

Equivalently, an $[n, k]$-code can be described as

$$\mathcal{B} = \left\{ \boldsymbol{v} \in \mathbb{F}^n \mid H\boldsymbol{v}^T = 0 \right\}$$

where $H \in \mathbb{F}^{(n-k) \times n}$ with $\operatorname{rank}(H) = n - k$ is called a **parity-check matrix** of $\mathcal{B}$.

Linear Block codes

Let $\mathbb{F}$ be a finite field and $n \in \mathbb{N}$. A linear block code of length n over $\mathbb{F}$ is a linear subspace $\mathcal{B}$ of $\mathbb{F}^n$. If $\dim \mathcal{B} = k$, then $\mathcal{B}$ is called an $[n, k]$-code and

$$\mathcal{B} = \left\{ \boldsymbol{v} \in \mathbb{F}^n \mid \boldsymbol{v} = \boldsymbol{u}G \text{ with } \boldsymbol{u} \in \mathbb{F}^k \right\}$$

where

- $G \in \mathbb{F}^{k \times n}$ with $\operatorname{rank}(G) = k$ is the **generator matrix** of $\mathcal{B}$,
- $\boldsymbol{u} \in \mathbb{F}^k$ is called **information word**,
- $\boldsymbol{v} \in \mathcal{B}$ is called **codeword**.

Equivalently, an $[n, k]$-code can be described as

$$\mathcal{B} = \left\{ \boldsymbol{v} \in \mathbb{F}^n \mid H\boldsymbol{v}^T = 0 \right\}$$

where $H \in \mathbb{F}^{(n-k) \times n}$ with $\operatorname{rank}(H) = n - k$ is called a **parity-check matrix** of $\mathcal{B}$.

Linear Block codes

Let $\mathbb{F}$ be a finite field and $n \in \mathbb{N}$. A linear block code of length n over $\mathbb{F}$ is a linear subspace $\mathcal{B}$ of $\mathbb{F}^n$. If $\dim \mathcal{B} = k$, then $\mathcal{B}$ is called an $[n, k]$-code and

$$\mathcal{B} = \left\{ \boldsymbol{v} \in \mathbb{F}^n \mid \boldsymbol{v} = \boldsymbol{u}G \text{ with } \boldsymbol{u} \in \mathbb{F}^k \right\}$$

where

- $G \in \mathbb{F}^{k \times n}$ with $\operatorname{rank}(G) = k$ is the **generator matrix** of $\mathcal{B}$,
- $\boldsymbol{u} \in \mathbb{F}^k$ is called **information word**,
- $\boldsymbol{v} \in \mathcal{B}$ is called **codeword**.

Equivalently, an $[n, k]$-code can be described as

$$\mathcal{B} = \left\{ \boldsymbol{v} \in \mathbb{F}^n \mid H\boldsymbol{v}^T = 0 \right\}$$

where $H \in \mathbb{F}^{(n-k) \times n}$ with $\operatorname{rank}(H) = n - k$ is called a **parity-check matrix** of $\mathcal{B}$.

Linear Block Codes

Theorem

Let $G \in \mathbb{F}^{k\times n}$ and $H \in \mathbb{F}^{(n-k)\times n}$ such that $rank(G) = k$ and $rank(H) = n - k$.

1. *Assume that G is a generator matrix of an $[n, k]$-code $\mathcal{B}$. Then, H is a parity-check matrix of $\mathcal{B}$ if and only if $HG^T = 0$.*
2. *Assume that H is a parity-check matrixg of an $[n, k]$-code $\mathcal{B}$. Then, G is a enerator matrix of $\mathcal{B}$ if and only if $HG^T = 0$.*

Linear Block Codes

We say that $G \in \mathbb{F}^{k \times n}$ is a **systematic generator matrix** of $\mathcal{B}$ if $G = [I_k \ A]$ for some $A \in \mathbb{F}^{k \times (n-k)}$.

Theorem

Let $\mathcal{B}$ be an $[n, k]$-code and consider $A \in \mathbb{F}^{k \times (n-k)}$. Then $G = [I_k \ A]$ is a systematic generator matrix of $\mathcal{B}$ if and only if $H = [-A^T \ I_{n-k}]$ is a systematic parity-check matrix of $\mathcal{B}$.

For any non-systematic generator matrix $G \in \mathbb{F}^{k \times n}$, there exists a permutation matrix $P \in \mathbb{F}^{n \times n}$ such that $GP = [I_k \ A]$, for some $A \in \mathbb{F}^{k \times (n-k)}$. Therefore, $H = [-A^T \ I_{n-k}]P^T$ is a non-systematic parity-check matrix.

Linear Block Codes

We say that $G \in \mathbb{F}^{k\times n}$ is a **systematic generator matrix** of $\mathcal{B}$ if $G = [I_k \ A]$ for some $A \in \mathbb{F}^{k\times(n-k)}$.

Theorem

Let $\mathcal{B}$ be an $[n,k]$-code and consider $A \in \mathbb{F}^{k\times(n-k)}$. Then $G = [I_k \ A]$ is a systematic generator matrix of $\mathcal{B}$ if and only if $H = [-A^T \ I_{n-k}]$ is a systematic parity-check matrix of $\mathcal{B}$.

For any non-systematic generator matrix $G \in \mathbb{F}^{k\times n}$, there exists a permutation matrix $P \in \mathbb{F}^{n\times n}$ such that $GP = [I_k \ A]$, for some $A \in \mathbb{F}^{k\times(n-k)}$. Therefore, $H = [-A^T \ I_{n-k}]P^T$ is a non-systematic parity-check matrix.

Linear Block Codes

We say that $G \in \mathbb{F}^{k\times n}$ is a **systematic generator matrix** of $\mathcal{B}$ if $G = [I_k \ A]$ for some $A \in \mathbb{F}^{k\times(n-k)}$.

Theorem

Let $\mathcal{B}$ be an $[n,k]$-code and consider $A \in \mathbb{F}^{k\times(n-k)}$. Then $G = [I_k \ A]$ is a systematic generator matrix of $\mathcal{B}$ if and only if $H = [-A^T \ I_{n-k}]$ is a systematic parity-check matrix of $\mathcal{B}$.

For any non-systematic generator matrix $G \in \mathbb{F}^{k\times n}$, there exists a permutation matrix $P \in \mathbb{F}^{n\times n}$ such that $GP = [I_k \ A]$, for some $A \in \mathbb{F}^{k\times(n-k)}$. Therefore, $H = [-A^T \ I_{n-k}]P^T$ is a non-systematic parity-check matrix.

Overview

Convolutional Codes

Definition

A convolutional code $\mathcal{C}$ is a $\mathbb{F}[z]$-module of $\mathbb{F}[z]^n$.

A matrix $G(z) \in \mathbb{F}[z]^{n \times k}$ whose rows form a basis for $\mathcal{C}$ is called an encoder or generator matrix. If $\mathcal{C}$ has rank k then we say that $\mathcal{C}$ has rate k/n.

$$\mathcal{C} = \mathrm{Im}_{\mathbb{F}[z]} G(z) = \left\{ \boldsymbol{v}(z) = \boldsymbol{u}(z)G(z) \mid \boldsymbol{u}(z) \in \mathbb{F}[z]^k \right\}$$

where $\boldsymbol{u}(z)$ is the **information vector**, and $\boldsymbol{v}(z)$ is the corresponding **codeword**.

Convolutional Codes

Definition

A convolutional code $\mathcal{C}$ is a $\mathbb{F}[z]$-module of $\mathbb{F}[z]^n$.

A matrix $G(z) \in \mathbb{F}[z]^{n \times k}$ whose rows form a basis for $\mathcal{C}$ is called an encoder or generator matrix. If $\mathcal{C}$ has rank k then we say that $\mathcal{C}$ has rate k/n.

$$\mathcal{C} = \mathrm{Im}_{\mathbb{F}[z]} G(z) = \left\{ \boldsymbol{v}(z) = \boldsymbol{u}(z)G(z) \;\mid\; \boldsymbol{u}(z) \in \mathbb{F}[z]^k \right\}$$

where $\boldsymbol{u}(z)$ is the **information vector**, and $\boldsymbol{v}(z)$ is the corresponding **codeword**.

Convolutional Codes

Assume that

$$G(z) = \begin{bmatrix} \boldsymbol{g}_1(z) & \boldsymbol{g}_2(z) & \cdots & \boldsymbol{g}_k(z) \end{bmatrix} \in \mathbb{F}[z]^{n\times k}$$

$$\boldsymbol{g}_j(z) = \boldsymbol{g}_j^{(0)} + \boldsymbol{g}_j^{(1)} z + \cdots + \boldsymbol{g}_j^{(\nu_j-1)} z^{\nu_j-1} + \boldsymbol{g}_j^{(\nu_j)} z^{\nu_j},$$

$$\boldsymbol{g}_j^{(\ell)} \in \mathbb{F}^n \quad j = 1, 2, \ldots, k, \ \ell = 0, 1, \ldots, \nu_j,$$

$$\boldsymbol{g}_j^{(\nu_j)} \neq 0, \ j = 1, 2, \ldots, k,$$

where ν_j is the j-th column degree of $G(z)$, for $j = 1, 2, \ldots, k$.

We can express $G(z) = G_{LD}X(z) + G_{\infty}Y(z)$, where:

Convolutional Codes

We can express $G(z) = G_{LD}X(z) + G_{\infty}Y(z)$, where:

The **high-order coefficient matrix** of $G(z)$ is the matrix

$$G_{\infty} = \left[\boldsymbol{g}_1^{(\nu_1)} \quad \boldsymbol{g}_2^{(\nu_2)} \quad \cdots \quad \boldsymbol{g}_k^{(\nu_k)} \right] \in \mathbb{F}^{n \times k}.$$

We define the **less-degree coefficients matrix**, as follows

$$G_{LD} = \left[\boldsymbol{g}_1^{(0)} \cdots \boldsymbol{g}_1^{(\nu_1 - 1)} \mid \boldsymbol{g}_2^{(0)} \cdots \boldsymbol{g}_2^{(\nu_2 - 1)} \mid \cdots \mid \boldsymbol{g}_k^{(0)} \cdots \boldsymbol{g}_k^{(\nu_k - 1)} \right] \in \mathbb{F}^{n \times \delta}.$$

Convolutional Codes

We can express $G(z) = G_{LD}X(z) + G_{\infty}Y(z)$, where:

The **high-order coefficient matrix** of $G(z)$ is the matrix

$$G_{\infty} = \left[\boldsymbol{g}_1^{(\nu_1)} \quad \boldsymbol{g}_2^{(\nu_2)} \quad \cdots \quad \boldsymbol{g}_k^{(\nu_k)} \right] \in \mathbb{F}^{n \times k}.$$

We define the **less-degree coefficients matrix**, as follows

$$G_{LD} = \left[\boldsymbol{g}_1^{(0)} \cdots \boldsymbol{g}_1^{(\nu_1 - 1)} \;\middle|\; \boldsymbol{g}_2^{(0)} \cdots \boldsymbol{g}_2^{(\nu_2 - 1)} \;\middle|\; \cdots \;\middle|\; \boldsymbol{g}_k^{(0)} \cdots \boldsymbol{g}_k^{(\nu_k - 1)} \right] \in \mathbb{F}^{n \times \delta}.$$

Convolutional Codes

We can express $G(z) = G_{LD}X(z) + G_{\infty}Y(z)$, where:

The **high-order coefficient matrix** of $G(z)$ is the matrix

$$G_{\infty} = \left[\boldsymbol{g}_1^{(\nu_1)} \quad \boldsymbol{g}_2^{(\nu_2)} \quad \cdots \quad \boldsymbol{g}_k^{(\nu_k)} \right] \in \mathbb{F}^{n \times k}.$$

We define the **less-degree coefficients matrix**, as follows

$$G_{LD} = \left[\boldsymbol{g}_1^{(0)} \cdots \boldsymbol{g}_1^{(\nu_1-1)} \;\middle|\; \boldsymbol{g}_2^{(0)} \cdots \boldsymbol{g}_2^{(\nu_2-1)} \;\middle|\; \cdots \;\middle|\; \boldsymbol{g}_k^{(0)} \cdots \boldsymbol{g}_k^{(\nu_k-1)} \right] \in \mathbb{F}^{n \times \delta}.$$

$$X(z) = \begin{bmatrix} X_1(z) & & & \\ & X_2(z) & & \\ & & \ddots & \\ & & & X_k(z) \end{bmatrix}, \; X_j(z) = \begin{bmatrix} 1 \\ z \\ \vdots \\ z^{\nu_j - 1} \end{bmatrix}, \text{ with } j = 1, 2, \ldots, k$$

Convolutional Codes

We can express $G(z) = G_{LD}X(z) + G_{\infty}Y(z)$, where:

The **high-order coefficient matrix** of $G(z)$ is the matrix

$$G_{\infty} = \left[\boldsymbol{g}_1^{(\nu_1)} \quad \boldsymbol{g}_2^{(\nu_2)} \quad \cdots \quad \boldsymbol{g}_k^{(\nu_k)} \right] \in \mathbb{F}^{n \times k}.$$

We define the **less-degree coefficients matrix**, as follows

$$G_{LD} = \left[\boldsymbol{g}_1^{(0)} \cdots \boldsymbol{g}_1^{(\nu_1 - 1)} \;\middle|\; \boldsymbol{g}_2^{(0)} \cdots \boldsymbol{g}_2^{(\nu_2 - 1)} \;\middle|\; \cdots \;\middle|\; \boldsymbol{g}_k^{(0)} \cdots \boldsymbol{g}_k^{(\nu_k - 1)} \right] \in \mathbb{F}^{n \times \delta}.$$

$$Y(z) = \begin{bmatrix} z^{\nu_1} & & & \\ & z^{\nu_2} & & \\ & & \ddots & \\ & & & z^{\nu_k} \end{bmatrix}.$$

Convolutional Code

Theorem

If $\mathcal{C}$ is a rate k/n convolutional code, then $\mathcal{C}$ admits a generator matrix $G(z)$ which is:

- *right-prime*
- *column reduced*

Therefore, $\delta = \sum_{j=1}^{k} \nu_j$ is the **degree** or **complexity** of $\mathcal{C}$.

Overview

Convolutional code as a linear system

A rate k/n convolutional code can be described by a time invariant linear system:

$$\left.\begin{aligned} \boldsymbol{x}_{t+1} &= A\boldsymbol{x}_t + B\boldsymbol{u}_t \\ \boldsymbol{y}_t &= C\boldsymbol{x}_t + D\boldsymbol{u}_t \end{aligned}\right\}, \quad \boldsymbol{v}_t = \begin{bmatrix} \boldsymbol{y}_t \\ \boldsymbol{u}_t \end{bmatrix}, \quad t \geq 0, \quad \boldsymbol{x}_0 = \mathbf{0},$$

where $A \in \mathbb{F}^{m\times m}$, $B \in \mathbb{F}^{m\times k}$, $C \in \mathbb{F}^{(n-k)\times m}$ and $D \in \mathbb{F}^{(n-k)\times k}$.

- $\boldsymbol{x}_t \in \mathbb{F}^m$, **state vector**,
- $\boldsymbol{u}_t \in \mathbb{F}^k$, **input vector**,
- $\boldsymbol{y}_t \in \mathbb{F}^{n-k}$, **output vector (redundancy)**,
- (A, B, C, D) is a **realization** of $G(x)$ of dimension m, known as the **Input-state-output (ISO) representation**.

Convolutional code as a linear system

A rate k/n convolutional code can be described by a time invariant linear system:

$$\left.\begin{aligned} \boldsymbol{x}_{t+1} &= A\boldsymbol{x}_t + B\boldsymbol{u}_t \\ \boldsymbol{y}_t &= C\boldsymbol{x}_t + D\boldsymbol{u}_t \end{aligned}\right\}, \quad \boldsymbol{v}_t = \begin{bmatrix} \boldsymbol{y}_t \\ \boldsymbol{u}_t \end{bmatrix}, \quad t \geq 0, \quad \boldsymbol{x}_0 = \mathbf{0},$$

where $A \in \mathbb{F}^{m\times m}$, $B \in \mathbb{F}^{m\times k}$, $C \in \mathbb{F}^{(n-k)\times m}$ and $D \in \mathbb{F}^{(n-k)\times k}$.

- $\boldsymbol{x}_t \in \mathbb{F}^m$, **state vector**,
- $\boldsymbol{u}_t \in \mathbb{F}^k$, **input vector**,
- $\boldsymbol{v}_t \in \mathbb{F}^n$, **output vector** (codewords).
- (A, B, C, D) is a realization of $G(z)$ of dimension m, known as the **input-state-output (ISO) representation**.

Convolutional code as a linear system

A rate k/n convolutional code can be described by a time invariant linear system:

$$\left.\begin{aligned} \boldsymbol{x}_{t+1} &= A\boldsymbol{x}_t + B\boldsymbol{u}_t \\ \boldsymbol{y}_t &= C\boldsymbol{x}_t + D\boldsymbol{u}_t \end{aligned}\right\}, \quad \boldsymbol{v}_t = \begin{bmatrix} \boldsymbol{y}_t \\ \boldsymbol{u}_t \end{bmatrix}, \quad t \geq 0, \quad \boldsymbol{x}_0 = \boldsymbol{0},$$

where $A \in \mathbb{F}^{m\times m}$, $B \in \mathbb{F}^{m\times k}$, $C \in \mathbb{F}^{(n-k)\times m}$ and $D \in \mathbb{F}^{(n-k)\times k}$.

- $\boldsymbol{x}_t \in \mathbb{F}^m$, **state vector**,
- $\boldsymbol{u}_t \in \mathbb{F}^k$, **input vector**,
- $\boldsymbol{v}_t \in \mathbb{F}^n$, **output vector** (codewords).
- (A, B, C, D) is a realization of $G(z)$ of dimension m, known as the **input-state-output (ISO) representation**.

Convolutional code as a linear system

A rate k/n convolutional code can be described by a time invariant linear system:

$$\left.\begin{aligned} \boldsymbol{x}_{t+1} &= A\boldsymbol{x}_t + B\boldsymbol{u}_t \\ \boldsymbol{y}_t &= C\boldsymbol{x}_t + D\boldsymbol{u}_t \end{aligned}\right\}, \quad \boldsymbol{v}_t = \begin{bmatrix} \boldsymbol{y}_t \\ \boldsymbol{u}_t \end{bmatrix}, \quad t \geq 0, \quad \boldsymbol{x}_0 = \boldsymbol{0},$$

where $A \in \mathbb{F}^{m\times m}$, $B \in \mathbb{F}^{m\times k}$, $C \in \mathbb{F}^{(n-k)\times m}$ and $D \in \mathbb{F}^{(n-k)\times k}$.

- $\boldsymbol{x}_t \in \mathbb{F}^m$, **state vector**,
- $\boldsymbol{u}_t \in \mathbb{F}^k$, **input vector**,
- $\boldsymbol{v}_t \in \mathbb{F}^n$, **output vector** (codewords).
- (A, B, C, D) is a realization of $G(z)$ of dimension m, known as the **input-state-output (ISO) representation**.

Convolutional code as a linear system

A rate k/n convolutional code can be described by a time invariant linear system:

$$\left.\begin{aligned} \boldsymbol{x}_{t+1} &= A\boldsymbol{x}_t + B\boldsymbol{u}_t \\ \boldsymbol{y}_t &= C\boldsymbol{x}_t + D\boldsymbol{u}_t \end{aligned}\right\}, \quad \boldsymbol{v}_t = \begin{bmatrix} \boldsymbol{y}_t \\ \boldsymbol{u}_t \end{bmatrix}, \quad t \geq 0, \quad \boldsymbol{x}_0 = \boldsymbol{0},$$

where $A \in \mathbb{F}^{m\times m}$, $B \in \mathbb{F}^{m\times k}$, $C \in \mathbb{F}^{(n-k)\times m}$ and $D \in \mathbb{F}^{(n-k)\times k}$.

- $\boldsymbol{x}_t \in \mathbb{F}^m$, **state vector**,
- $\boldsymbol{u}_t \in \mathbb{F}^k$, **input vector**,
- $\boldsymbol{v}_t \in \mathbb{F}^n$, **output vector** (codewords).
- (A, B, C, D) is a realization of $G(z)$ of dimension m, known as the **input-state-output (ISO) representation**.

Convolutional code as a linear system

A rate k/n convolutional code can be described by a time invariant linear system:

$$\left.\begin{aligned} \boldsymbol{x}_{t+1} &= A\boldsymbol{x}_t + B\boldsymbol{u}_t \\ \boldsymbol{y}_t &= C\boldsymbol{x}_t + D\boldsymbol{u}_t \end{aligned}\right\}, \quad \boldsymbol{v}_t = \begin{bmatrix} \boldsymbol{y}_t \\ \boldsymbol{u}_t \end{bmatrix}, \quad t \geq 0, \quad \boldsymbol{x}_0 = \boldsymbol{0},$$

where $A \in \mathbb{F}^{m\times m}$, $B \in \mathbb{F}^{m\times k}$, $C \in \mathbb{F}^{(n-k)\times m}$ and $D \in \mathbb{F}^{(n-k)\times k}$.

- $\boldsymbol{x}_t \in \mathbb{F}^m$, **state vector**,
- $\boldsymbol{u}_t \in \mathbb{F}^k$, **input vector**,
- $\boldsymbol{v}_t \in \mathbb{F}^n$, **output vector** (codewords).
- (A, B, C, D) is a realization of $G(z)$ of dimension m, known as the **input-state-output (ISO) representation**.

ISO representation

- A realization (A, B, C, D) of $G(z)$ is **minimal** if $m = \delta$.
- (A, B, C, D) is minimal $\leftrightarrow$ (A, B) is reachable and (A, C) is observable.

(A, B) is called **reachable** if

$$\text{rank} \begin{bmatrix} B & AB & \cdots & A^{\delta-1}B \end{bmatrix} = \delta.$$

(A, C) is **observable** $\leftrightarrow$ (A^T, C^T) is reachable.

ISO representation

- A realization (A, B, C, D) of $G(z)$ is **minimal** if $m = \delta$.
- (A, B, C, D) is minimal $\leftrightarrow$ (A, B) is reachable and (A, C) is observable.

(A, B) is called **reachable** if

$$\text{rank} \begin{bmatrix} B & AB & \cdots & A^{\delta-1}B \end{bmatrix} = \delta.$$

(A, C) is **observable** $\leftrightarrow$ (A^T, C^T) is reachable.

ISO representation

- A realization (A, B, C, D) of $G(z)$ is **minimal** if $m = \delta$.
- (A, B, C, D) is minimal $\leftrightarrow$ (A, B) is reachable and (A, C) is observable.

(A, B) is called **reachable** if

$$\text{rank} \begin{bmatrix} B & AB & \cdots & A^{\delta-1}B \end{bmatrix} = \delta.$$

(A, C) is **observable** $\leftrightarrow$ (A^T, C^T) is reachable.

First order representation

A minimal ISO representation (A, B, C, D) can be obtained from a first order representation (K, L, M).

Rosenthal, J., Schumacher, J.M., and York, E.V. (1996).
On behaviors and convolutional codes.
IEEE Transactions on Information Theory, 42(6), 1881–1891.

First order representation

Theorem (Theorem 3.1 [1])

Assume $\mathcal{C} \subseteq \mathbb{F}[z]^n$ is a rate k/n convolutional code of complexity δ. Then, there exist $(\delta + n - k) \times \delta$ matrices K, L and a $(\delta + n - k) \times n$ matrix M (all defined over $\mathbb{F}$) such that the code $\mathcal{C}$ is described by

$$\mathcal{C} = \left\{ v(z) \in \mathbb{F}[z]^n \mid \exists x(z) \in \mathbb{F}[z]^\delta : zKx(z) + Lx(z) + Mv(z) = 0 \right\}$$

(K, L, M) is known as a **first order representation** *of $\mathcal{C}$. Moreover, it is minimal, if it satisfies:*

1. *K has full column rank,*
2. *$[K \ M]$ has full row rank,*
3. *$[zK + L \mid M]$ is left-prime.*

From First order representation to ISO representation

If $G(z)$ is a generator matrix of $\mathcal{C}$, then (K, L, M) satisfies that

$$[K\ L\ M] \begin{bmatrix} zX(z) \\ X(z) \\ G(z) \end{bmatrix} = 0.$$

Rosenthal's Algorithm

Rosenthal's Algorithm

1. Let $G(z)$ be a generator matrix of a k/n-rate convolutional code.
2. Given $[zX(z)^T, X(z)^T, G(z)^T] \in \mathbb{F}^{(\delta+k)\times(2\delta+n)}$, compute its scalar matrix which describes a linear map $\Phi : \mathbb{F}^{2\delta+k} \rightarrow \mathbb{F}^{\delta+k}$. The vectors of the basis of the Kernel of Φ define a matrix from which they can obtain a minimal first-order representation $(K,\ L,\ M)$.
3. *After a suitable similarity transformation and a permutation (if needed)*, they can obtain

$$[K \mid L \mid M] \sim \left[\begin{array}{c|c|cc} -I & A & 0 & B \\ 0 & C & -I & D \end{array}\right]$$

4. (A, B, C, D) is a minimal ISO representation of $\mathcal{C}$.

Rosenthal's Algorithm

Rosenthal's Algorithm

1. Let $G(z)$ be a generator matrix of a k/n-rate convolutional code.
2. Given $[zX(z)^T, X(z)^T, G(z)^T] \in \mathbb{F}^{(\delta+k)\times(2\delta+n)}$, compute its scalar matrix which describes a linear map $\Phi : \mathbb{F}^{2\delta+k} \rightarrow \mathbb{F}^{\delta+k}$. The vectors of the basis of the Kernel of Φ define a matrix from which they can obtain a minimal first-order representation $(K,\ L,\ M)$.
3. *After a suitable similarity transformation and a permutation (if needed)*, they can obtain

$$[K \mid L \mid M] \sim \left[\begin{array}{c|c|cc} -I & A & 0 & B \\ 0 & C & -I & D \end{array}\right]$$

4. (A, B, C, D) is a minimal ISO representation of $\mathcal{C}$.

Rosenthal's Algorithm

Rosenthal's Algorithm

1. Let $G(z)$ be a generator matrix of a k/n-rate convolutional code.
2. Given $[zX(z)^T, X(z)^T, G(z)^T] \in \mathbb{F}^{(\delta+k)\times(2\delta+n)}$, compute its scalar matrix which describes a linear map $\Phi : \mathbb{F}^{2\delta+k} \rightarrow \mathbb{F}^{\delta+k}$. The vectors of the basis of the Kernel of Φ define a matrix from which they can obtain a minimal first-order representation $(K,\ L,\ M)$.
3. *After a suitable similarity transformation and a permutation (if needed)*, they can obtain

$$[K \mid L \mid M] \sim \left[\begin{array}{c|c|cc} -I & A & 0 & B \\ 0 & C & -I & D \end{array}\right]$$

4. (A, B, C, D) is a minimal ISO representation of $\mathcal{C}$.

Rosenthal's Algorithm

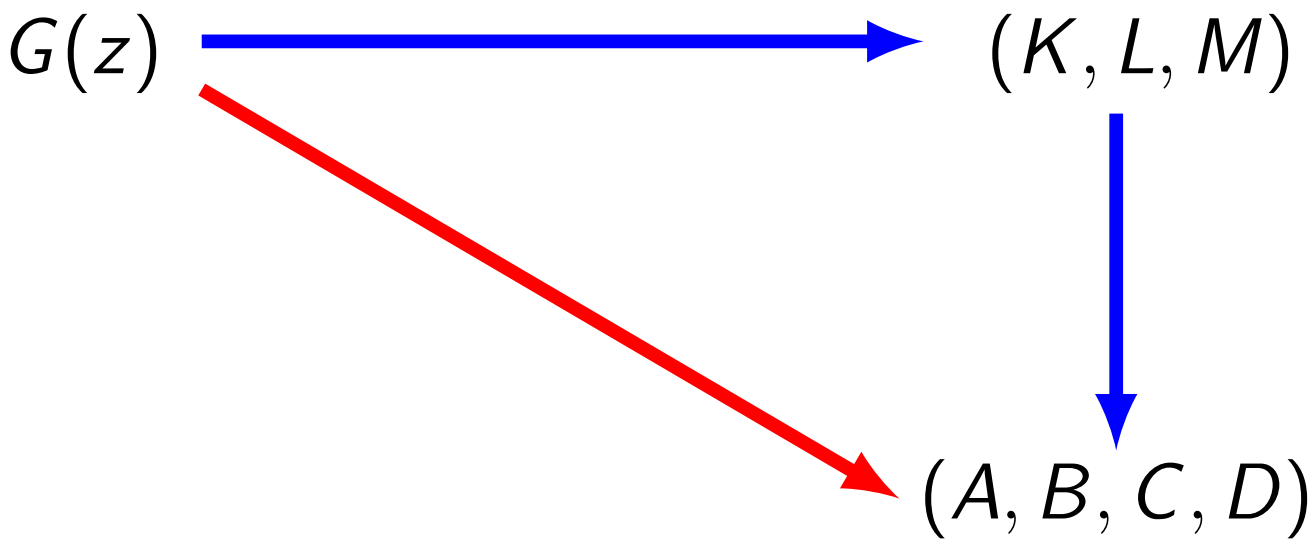
Rosenthal's Algorithm

1. Let $G(z)$ be a generator matrix of a k/n-rate convolutional code.
2. Given $[zX(z)^T, X(z)^T, G(z)^T] \in \mathbb{F}^{(\delta+k)\times(2\delta+n)}$, compute its scalar matrix which describes a linear map $\Phi : \mathbb{F}^{2\delta+k} \rightarrow \mathbb{F}^{\delta+k}$. The vectors of the basis of the Kernel of Φ define a matrix from which they can obtain a minimal first-order representation $(K,\ L,\ M)$.
3. *After a suitable similarity transformation and a permutation (if needed)*, they can obtain

$$[K \mid L \mid M] \sim \left[\begin{array}{c|c|cc} -I & A & 0 & B \\ 0 & C & -I & D \end{array}\right]$$

4. (A, B, C, D) is a minimal ISO representation of $\mathcal{C}$.

From First order representation to ISO representation



Our minimal ISO representation

Aim

We provide an algorithm to obtain a minimal ISO representation (A, B, C, D), using its generator matrix (not (K, L, M)).

Conditions for $G(z)$, generator matrix of a (n, k) convolutional code $\mathcal{C}$:

- $G(z)$ is right-prime.
- $G(z)$ is column reduced.
- Its complexity $\delta = \sum_{i=1}^{k} \nu_i$.
- Assume that the rows which determine the column degrees are in the last k rows of $G(z)$.

Our minimal ISO representation

Aim

We provide an algorithm to obtain a minimal ISO representation (A, B, C, D), using its generator matrix (not (K, L, M)).

Conditions for $G(z)$, generator matrix of a (n, k) convolutional code $\mathcal{C}$:

- $G(z)$ is right-prime.
- $G(z)$ is column reduced .
- Its complexity $\delta = \sum_{j=1}^{k} \nu_j$.
- Assume that the rows which determine the column degrees are in the last k rows of $G(z)$.

Our minimal ISO representation

Aim

We provide an algorithm to obtain a minimal ISO representation (A, B, C, D), using its generator matrix (not (K, L, M)).

Conditions for $G(z)$, generator matrix of a (n, k) convolutional code $\mathcal{C}$:

- $G(z)$ is right-prime.
- $G(z)$ is column reduced .
- Its complexity $\delta = \sum_{j=1}^{k} \nu_j$.
- Assume that the rows which determine the column degrees are in the last k rows of $G(z)$.

Our minimal ISO representation

Aim

We provide an algorithm to obtain a minimal ISO representation (A, B, C, D), using its generator matrix (not (K, L, M)).

Conditions for $G(z)$, generator matrix of a (n, k) convolutional code $\mathcal{C}$:

- $G(z)$ is right-prime.
- $G(z)$ is column reduced .
- Its complexity $\delta = \sum_{j=1}^{k} \nu_j$.
- Assume that the rows which determine the column degrees are in the last k rows of $G(z)$.

Our minimal ISO representation

Aim

We provide an algorithm to obtain a minimal ISO representation (A, B, C, D), using its generator matrix (not (K, L, M)).

Conditions for $G(z)$, generator matrix of a (n, k) convolutional code $\mathcal{C}$:

- $G(z)$ is right-prime.
- $G(z)$ is column reduced .
- Its complexity $\delta = \sum_{j=1}^{k} \nu_j$.
- Assume that the rows which determine the column degrees are in the last k rows of $G(z)$.

ISO representation

$$[K\ L\ M]\begin{bmatrix} zX(z) \\ X(z) \\ G(z) \end{bmatrix} = 0 \longleftrightarrow [K\ L\ M]\,\tilde{G}^T = 0$$

$$\tilde{G} = \left[\begin{array}{cccccccc}
\mathbf{0}^T\ 0 & & & 1\ \mathbf{0}^T & & & & \boldsymbol{g}_1^{(0)T} \\
I_{\nu_1-1}\mathbf{0} & & & \mathbf{0}I_{\nu_1-1} & & & & \bar{\boldsymbol{g}}_1^T \\
\mathbf{0}^T\ 1 & & & 0\ \mathbf{0}^T & & & & \boldsymbol{g}_1^{(\nu_1)T} \\
\hline
 & \mathbf{0}^T\ 0 & & & 1\ \mathbf{0}^T & & & \boldsymbol{g}_2^{(0)T} \\
 & I_{\nu_2-1}\mathbf{0} & & & \mathbf{0}I_{\nu_2-1} & & & \bar{\boldsymbol{g}}_2^T \\
 & \mathbf{0}^T\ 1 & & & 0\ \mathbf{0}^T & & & \boldsymbol{g}_2^{(\nu_2)T} \\
\hline
 & & \ddots & & & \ddots & & \vdots \\
\hline
 & & & \mathbf{0}^T\ 0 & & & 1\ \mathbf{0}^T & \boldsymbol{g}_k^{(0)T} \\
 & & & I_{\nu_k-1}\mathbf{0} & & & \mathbf{0}I_{\nu_k-1} & \bar{\boldsymbol{g}}_k^T \\
 & & & \mathbf{0}^T\ 1 & & & 0\ \mathbf{0}^T & \boldsymbol{g}_k^{(\nu_k)T}
\end{array}\right]$$

with $\bar{\boldsymbol{g}}_j = [\boldsymbol{g}_j^{(1)}\ \boldsymbol{g}_j^{(2)}\ \cdots\ \boldsymbol{g}_j^{(\nu_j-1)}]$, for $j = 1, 2, \ldots, k$

ISO representation

$$[K\ L\ M]\begin{bmatrix} zX(z) \\ X(z) \\ G(z) \end{bmatrix} = 0 \longleftrightarrow [K\ L\ M]\,\tilde{G}^T = 0$$

$$\tilde{G} = \left[\begin{array}{c|c|c|c|c|c|c|c|c} \begin{matrix}\mathbf{0}^T & 0\\ I_{\nu_1-1}\mathbf{0} & \\ \mathbf{0}^T & 1\end{matrix} & & & & \begin{matrix}1 & \mathbf{0}^T\\ \mathbf{0} & I_{\nu_1-1}\\ 0 & \mathbf{0}^T\end{matrix} & & & & \begin{matrix}\boldsymbol{g}_1^{(0)T}\\ \bar{\boldsymbol{g}}_1^T\\ \boldsymbol{g}_1^{(\nu_1)T}\end{matrix} \\ \hline & \begin{matrix}\mathbf{0}^T & 0\\ I_{\nu_2-1}\mathbf{0} & \\ \mathbf{0}^T & 1\end{matrix} & & & & \begin{matrix}1 & \mathbf{0}^T\\ \mathbf{0} & I_{\nu_2-1}\\ 0 & \mathbf{0}^T\end{matrix} & & & \begin{matrix}\boldsymbol{g}_2^{(0)T}\\ \bar{\boldsymbol{g}}_2^T\\ \boldsymbol{g}_2^{(\nu_2)T}\end{matrix} \\ \hline & & \ddots & & & & \ddots & & \vdots \\ \hline & & & \begin{matrix}\mathbf{0}^T & 0\\ I_{\nu_k-1}\mathbf{0} & \\ \mathbf{0}^T & 1\end{matrix} & & & & \begin{matrix}1 & \mathbf{0}^T\\ \mathbf{0} & I_{\nu_k-1}\\ 0 & \mathbf{0}^T\end{matrix} & \begin{matrix}\boldsymbol{g}_k^{(0)T}\\ \bar{\boldsymbol{g}}_k^T\\ \boldsymbol{g}_k^{(\nu_k)T}\end{matrix} \end{array}\right]$$

with $\bar{\boldsymbol{g}}_j = [\boldsymbol{g}_j^{(1)}\ \boldsymbol{g}_j^{(2)}\ \ldots\ \boldsymbol{g}_j^{(\nu_j-1)}]$, for $j = 1, 2, \ldots, k$.

Universitat d'Alacant
Universidad de Alicante

ISO representation

$$[K\ L\ M]\begin{bmatrix} zX(z) \\ X(z) \\ G(z) \end{bmatrix} = 0 \longleftrightarrow [K\ L\ M]\,\tilde{G}^T = 0.$$

$\widetilde{G}$ is a generator matrix of a $[2\delta + n, \delta + k]$ block code $\mathcal{B}$ where $[K\ L\ M]$ is a parity-check of $\mathcal{B}$.

ISO representation

With some row operations in $\widetilde{G}$

$$\widetilde{G} = \left[\begin{array}{c|c|c|c|c|c|c|c|c} \mathbf{0}^T\ 0 & & & & 1\ \mathbf{0}^T & & & & \boldsymbol{g}_1^{(0)^T} \\ I_{\nu_1-1}\mathbf{0} & & & & \mathbf{0}I_{\nu_1-1} & & & & \bar{\boldsymbol{g}}_1^T \\ \mathbf{0}^T\ 1 & & & & 0\ \mathbf{0}^T & & & & \boldsymbol{g}_1^{(\nu_1)^T} \\ \hline & \mathbf{0}^T\ 0 & & & & 1\ \mathbf{0}^T & & & \boldsymbol{g}_2^{(0)^T} \\ & I_{\nu_2-1}\mathbf{0} & & & & \mathbf{0}I_{\nu_2-1} & & & \bar{\boldsymbol{g}}_2^T \\ & \mathbf{0}^T\ 1 & & & & 0\ \mathbf{0}^T & & & \boldsymbol{g}_2^{(\nu_2)^T} \\ \hline & & \ddots & & & & \ddots & & \vdots \\ \hline & & & \mathbf{0}^T\ 0 & & & & 1\ \mathbf{0}^T & \boldsymbol{g}_k^{(0)^T} \\ & & & I_{\nu_k-1}\mathbf{0} & & & & \mathbf{0}I_{\nu_k-1} & \bar{\boldsymbol{g}}_k^T \\ & & & \mathbf{0}^T\ 1 & & & & 0\ \mathbf{0}^T & \boldsymbol{g}_k^{(\nu_k)^T} \end{array}\right]$$

ISO representation

$$\widetilde{\widetilde{G}} = \left[\begin{array}{c|c|c|c|c|c|c|c|c}
\begin{matrix}\mathbf{0}^T & 0\\ I_{\nu_1-1} & \mathbf{0}\end{matrix} & & & & \begin{matrix}1 & \mathbf{0}^T\\ \mathbf{0} & I_{\nu_1-1}\end{matrix} & & & & \begin{matrix}\mathbf{g}_1^{(0)T}\\ \bar{\mathbf{g}}_1^T\end{matrix} \\ \hline
 & \begin{matrix}\mathbf{0}^T & 0\\ I_{\nu_2-1} & \mathbf{0}\end{matrix} & & & & \begin{matrix}1 & \mathbf{0}^T\\ \mathbf{0} & I_{\nu_2-1}\end{matrix} & & & \begin{matrix}\mathbf{g}_2^{(0)T}\\ \bar{\mathbf{g}}_2^T\end{matrix} \\ \hline
 & & \ddots & & & & \ddots & & \vdots \\ \hline
 & & & \begin{matrix}\mathbf{0}^T & 0\\ I_{\nu_k-1} & \mathbf{0}\end{matrix} & & & & \begin{matrix}1 & \mathbf{0}^T\\ \mathbf{0} & I_{\nu_k-1}\end{matrix} & \begin{matrix}\mathbf{g}_k^{(0)T}\\ \bar{\mathbf{g}}_k^T\end{matrix} \\ \hline
\mathbf{0}^T & 1 & & & 0 & \mathbf{0}^T & & & & \mathbf{g}_1^{(\nu_1)T} \\ \hdashline
 & \mathbf{0}^T & 1 & & & 0 & \mathbf{0}^T & & & \mathbf{g}_2^{(\nu_2)T} \\ \hdashline
 & & \ddots & & & & \ddots & & \vdots \\ \hdashline
 & & & \mathbf{0}^T & 1 & & & 0 & \mathbf{0}^T & \mathbf{g}_k^{(\nu_k)T}
\end{array}\right]$$

ISO representation

We have that $[g_1^{(\nu_1)} \; g_2^{(\nu_2)} \cdots g_k^{(\nu_k)}]$ is full rank, then using row operations in $\widetilde{\widetilde{G}}$, we can obtain an equivalent generator matrix

$$G = \left[\begin{array}{cccc} A^T & I_\delta & C^T & 0_{\delta \times k} \\ B^T & 0_{k \times \delta} & D^T & I_k \end{array} \right].$$

For the matrix

$$H = \left[\begin{array}{cccc} I & -A & 0 & -B \\ 0 & -C & I & -D \end{array} \right] \in \mathbb{F}^{(\delta+n-k) \times (2\delta+n)},$$

it follows that $HG^T = 0$. Therefore, as G has full rank, we have that H is the parity check matrix of $\mathcal{B}$. Furthermore,

$$H = [K \mid L \mid M]$$

then, we have that (A, B, C, D) is a minimal ISO representation of a convolutional code $\mathcal{C}$.

ISO representation

We have that $[g_1^{(\nu_1)} \; g_2^{(\nu_2)} \cdots g_k^{(\nu_k)}]$ is full rank, then using row operations in $\widetilde{\widetilde{G}}$, we can obtain an equivalent generator matrix

$$G = \begin{bmatrix} A^T & I_\delta & C^T & 0_{\delta \times k} \\ B^T & 0_{k \times \delta} & D^T & I_k \end{bmatrix}.$$

For the matrix

$$H = \begin{bmatrix} I & -A & 0 & -B \\ 0 & -C & I & -D \end{bmatrix} \in \mathbb{F}^{(\delta+n-k)\times(2\delta+n)},$$

it follows that $HG^T = 0$. Therefore, as G has full rank, we have that H is the parity check matrix of $\mathcal{B}$. Furthermore,

$$H = [K \mid L \mid M]$$

then, we have that (A, B, C, D) is a minimal ISO representation of a convolutional code $\mathcal{C}$.

Algorithm to obtain an ISO representation

Algorithm:

- **Input**: $G(z) \in \mathbb{F}[z]^{n\times k}$. Consider

$$G_{LD} = \begin{bmatrix} U_1 \\ U_2 \end{bmatrix} \quad \text{and} \quad G_{\infty} = \begin{bmatrix} V_1 \\ V_2 \end{bmatrix}$$

where $U_1 \in \mathbb{F}^{(n-k)\times\delta}$, $U_2 \in \mathbb{F}^{k\times\delta}$, $V_1 \in \mathbb{F}^{(n-k)\times k}$, with $V_2 \in \mathbb{F}^{k\times k}$ an invertible matrix.

- **Step 1**: Compute $V_2^{-1} = \begin{bmatrix} \boldsymbol{v}_1 \\ \boldsymbol{v}_2 \\ \vdots \\ \boldsymbol{v}_k \end{bmatrix}$, where $\boldsymbol{v}_j \in \mathbb{F}^{1\times k}$, for $j = 1, 2, \ldots, k$.

Algorithm to obtain an ISO representation

Algorithm:

- **Input**: $G(z) \in \mathbb{F}[z]^{n\times k}$. Consider

$$G_{LD} = \begin{bmatrix} U_1 \\ U_2 \end{bmatrix} \quad \text{and} \quad G_{\infty} = \begin{bmatrix} V_1 \\ V_2 \end{bmatrix}$$

where $U_1 \in \mathbb{F}^{(n-k)\times\delta}$, $U_2 \in \mathbb{F}^{k\times\delta}$, $V_1 \in \mathbb{F}^{(n-k)\times k}$, with $V_2 \in \mathbb{F}^{k\times k}$ an invertible matrix.

- **Step 1**: Compute $V_2^{-1} = \begin{bmatrix} \boldsymbol{v}_1 \\ \boldsymbol{v}_2 \\ \vdots \\ \boldsymbol{v}_k \end{bmatrix}$, where $\boldsymbol{v}_j \in \mathbb{F}^{1\times k}$, for $j = 1, 2, \ldots, k$.

Algorithm to obtain an ISO representation

Algorithm:

- **Step 2**: Compute $D = V_1 V_2^{-1}$.
- **Step 3**: Compute $C = U_1 - DU_2$.
- **Step 4**: Consider the matrix

$$B = \begin{bmatrix} B_1 \\ B_2 \\ \vdots \\ B_k \end{bmatrix} \quad \text{with } B_j = \begin{bmatrix} 0_{(\nu_j - 1) \times k} \\ \boldsymbol{v}_j \end{bmatrix}$$

  for $j = 1, 2, \ldots, k$.

- **Step 5**: Compute

$$\boldsymbol{w}_j = \boldsymbol{v}_j U_2 = \begin{bmatrix} \boldsymbol{v}_j U_2^{(\nu_1)} & \boldsymbol{v}_j U_2^{(\nu_2)} & \ldots & \boldsymbol{v}_j U_2^{(\nu_k)} \end{bmatrix} = \begin{bmatrix} \boldsymbol{w}_j^{(\nu_1)} & \boldsymbol{w}_j^{(\nu_2)} & \ldots & \boldsymbol{w}_j^{(\nu_k)} \end{bmatrix}$$

  for $j = 1, 2, \ldots, k$.

Algorithm to obtain an ISO representation

Algorithm:

- **Step 2**: Compute $D = V_1 V_2^{-1}$.
- **Step 3**: Compute $C = U_1 - DU_2$.
- **Step 4**: Consider the matrix

$$B = \begin{bmatrix} B_1 \\ B_2 \\ \vdots \\ B_k \end{bmatrix} \quad \text{with } B_j = \begin{bmatrix} 0_{(\nu_j - 1) \times k} \\ \boldsymbol{v}_j \end{bmatrix}$$

  for $j = 1, 2, \ldots, k$.

- **Step 5**: Compute

$$\boldsymbol{w}_j = \boldsymbol{v}_j U_2 = \begin{bmatrix} \boldsymbol{v}_j U_2^{(\nu_1)} & \boldsymbol{v}_j U_2^{(\nu_2)} & \ldots & \boldsymbol{v}_j U_2^{(\nu_k)} \end{bmatrix} = \begin{bmatrix} \boldsymbol{w}_j^{(\nu_1)} & \boldsymbol{w}_j^{(\nu_2)} & \ldots & \boldsymbol{w}_j^{(\nu_k)} \end{bmatrix}$$

  for $j = 1, 2, \ldots, k$.

Algorithm to obtain an ISO representation

Algorithm:

- **Step 2**: Compute $D = V_1 V_2^{-1}$.
- **Step 3**: Compute $C = U_1 - DU_2$.
- **Step 4**: Consider the matrix

$$B = \begin{bmatrix} B_1 \\ B_2 \\ \vdots \\ B_k \end{bmatrix} \quad \text{with } B_j = \begin{bmatrix} 0_{(\nu_j - 1) \times k} \\ \boldsymbol{v}_j \end{bmatrix}$$

for $j = 1, 2, \ldots, k$.

- **Step 5**: Compute

$$\boldsymbol{w}_j = \boldsymbol{v}_j U_2 = \left[\boldsymbol{v}_j U_2^{(\nu_1)} \ \boldsymbol{v}_j U_2^{(\nu_2)} \ldots \boldsymbol{v}_j U_2^{(\nu_k)} \right] = \left[\boldsymbol{w}_j^{(\nu_1)} \ \boldsymbol{w}_j^{(\nu_2)} \ldots \boldsymbol{w}_j^{(\nu_k)} \right]$$

for $j = 1, 2, \ldots, k$.

Algorithm to obtain an ISO representation

Algorithm:

- **Step 2**: Compute $D = V_1 V_2^{-1}$.
- **Step 3**: Compute $C = U_1 - DU_2$.
- **Step 4**: Consider the matrix

$$B = \begin{bmatrix} B_1 \\ B_2 \\ \vdots \\ B_k \end{bmatrix} \quad \text{with } B_j = \begin{bmatrix} 0_{(\nu_j - 1) \times k} \\ \boldsymbol{v}_j \end{bmatrix}$$

  for $j = 1, 2, \ldots, k$.

- **Step 5**: Compute

$$\boldsymbol{w}_j = \boldsymbol{v}_j U_2 = \left[\boldsymbol{v}_j U_2^{(\nu_1)} \ \boldsymbol{v}_j U_2^{(\nu_2)} \ldots \boldsymbol{v}_j U_2^{(\nu_k)} \right] = \left[\boldsymbol{w}_j^{(\nu_1)} \ \boldsymbol{w}_j^{(\nu_2)} \ldots \boldsymbol{w}_j^{(\nu_k)} \right]$$

  for $j = 1, 2, \ldots, k$.

Algorithm to obtain an ISO representation

Algorithm:

- **Step 6**: Consider the matrix

$$A = \begin{bmatrix} A_{11} & A_{12} & \cdots & A_{1k} \\ A_{21} & A_{22} & \cdots & A_{2k} \\ \vdots & \vdots & \vdots & \vdots \\ A_{k1} & A_{k2} & \cdots & A_{kk} \end{bmatrix}$$

with

$$A_{jj} = \begin{bmatrix} \mathbf{0} & I_{\nu_j - 1} \\ \boldsymbol{w}_j^{(\nu_j)} & \end{bmatrix}, \quad \text{for } j = 1, 2, \ldots, k$$

$$A_{ij} = \begin{bmatrix} 0_{(\nu_i - 1) \times \nu_j} \\ \boldsymbol{w}_i^{(\nu_j)} \end{bmatrix}, \quad \text{for } j = 1, 2, \ldots, k \quad \text{with } i \neq j.$$

- **Output**: (A, B, C, D) minimal.

Algorithm to obtain an ISO representation

Algorithm:

- **Step 6**: Consider the matrix

$$A = \begin{bmatrix} A_{11} & A_{12} & \cdots & A_{1k} \\ A_{21} & A_{22} & \cdots & A_{2k} \\ \vdots & \vdots & \vdots & \vdots \\ A_{k1} & A_{k2} & \cdots & A_{kk} \end{bmatrix}$$

with

$$A_{jj} = \begin{bmatrix} \mathbf{0} & I_{\nu_j - 1} \\ \multicolumn{2}{c}{\boldsymbol{w}_j^{(\nu_j)}} \end{bmatrix}, \quad \text{for } j = 1, 2, \ldots, k$$

$$A_{ij} = \begin{bmatrix} 0_{(\nu_i - 1) \times \nu_j} \\ \boldsymbol{w}_i^{(\nu_j)} \end{bmatrix}, \quad \text{for } j = 1, 2, \ldots, k \quad \text{with } i \neq j.$$

- **Output**: (A, B, C, D) minimal.

Overview

Minimal ISO representation of Convolutional Product Codes

Main goal:

Let (A_h, B_h, C_h, D_h) and (A_v, B_v, C_v, D_v) be minimal ISO representations of a $\mathcal{C}_h$ horizontal and a $\mathcal{C}_v$ vertical codes. Our aim is to obtain a minimal ISO representations of the convolutional product code from the minimal ISO representations (A_h, B_h, C_h, D_h) and (A_v, B_v, C_v, D_v) without using the generator matrix of the product.


J.-J. Climent, D. Napp, R. Pinto, and V. Requena. (2021).
Minimal state-space representation of convolutional product codes.
Mathematics, 9, Article 1410.


Climent, J.-J., Herranz, M., and Perea, C. (2015).
Input-state-output representation of convolutional product codes.
Proceedings of the 4th International Castle Meeting on Coding Theory and Applications (4ICMCTA), CIM Series in Mathematical Sciences, 3(6), 107–114.

Minimal ISO representation of Convolutional Product Codes

Main goal:

Let (A_h, B_h, C_h, D_h) and (A_v, B_v, C_v, D_v) be minimal ISO representations of a $\mathcal{C}_h$ horizontal and a $\mathcal{C}_v$ vertical codes. Our aim is to obtain a minimal ISO representations of the convolutional product code from the minimal ISO representations (A_h, B_h, C_h, D_h) and (A_v, B_v, C_v, D_v) without using the generator matrix of the product.

J.-J. Climent, D. Napp, R. Pinto, and V. Requena. (2021).
Minimal state-space representation of convolutional product codes.
Mathematics, 9, Article 1410.

Climent, J.-J., Herranz, M., and Perea, C. (2015).
Input-state-output representation of convolutional product codes.
Proceedings of the 4th International Castle Meeting on Coding Theory and Applications (4ICMCTA), CIM Series in Mathematical Sciences, 3(6), 107–114.

Minimal ISO representation of Convolutional Product Codes

Main goal:

Let (A_h, B_h, C_h, D_h) and (A_v, B_v, C_v, D_v) be minimal ISO representations of a $\mathcal{C}_h$ horizontal and a $\mathcal{C}_v$ vertical codes. Our aim is to obtain a minimal ISO representations of the convolutional product code from the minimal ISO representations (A_h, B_h, C_h, D_h) and (A_v, B_v, C_v, D_v) without using the generator matrix of the product.


J.-J. Climent, D. Napp, R. Pinto, and V. Requena. (2021).
Minimal state-space representation of convolutional product codes.
Mathematics, 9, Article 1410.


Climent, J.-J., Herranz, M., and Perea, C. (2015).
Input-state-output representation of convolutional product codes.
Proceedings of the 4th International Castle Meeting on Coding Theory and Applications (4ICMCTA), CIM Series in Mathematical Sciences, 3(6), 107–114.

ISO representations of Convolutional Product Codes

Theorem (Theorem 4 in Climent2015)

Let (A_h, B_h, C_h, D_h) and (A_v, B_v, C_v, D_v) be minimal ISO representations of a $\mathcal{C}_h$ horizontal and a $\mathcal{C}_v$ vertical codes. Then,

$$A = \begin{bmatrix} I_{n_h-k_h} \otimes A_v & C_h \otimes B_v & 0 \\ 0 & A_h \otimes I_{k_v} & 0 \\ 0 & 0 & I_{k_h} \otimes A_v \end{bmatrix} B = \begin{bmatrix} D_h \otimes B_v \\ B_h \otimes I_{k_v} \\ I_{k_h} \otimes B_v \end{bmatrix}$$

$$C = \begin{bmatrix} I_{n_h-k_h} \otimes C_v & C_h \otimes D_v & 0 \\ 0 & C_h \otimes I_{k_v} & 0 \\ 0 & 0 & I_{k_h} \otimes C_v \end{bmatrix} D = \begin{bmatrix} D_h \otimes D_v \\ D_h \otimes I_{k_v} \\ I_{k_h} \otimes D_v \end{bmatrix}$$

is an ISO representation of the convolutional product code $\mathcal{C} = \mathcal{C}_h \otimes \mathcal{C}_v$ of rate $k_h k_v / n_h n_v$.

ISO representations of Convolutional Product Codes

Theorem (Theorem 4 in Climent2015)

Let (A_h, B_h, C_h, D_h) and (A_v, B_v, C_v, D_v) be minimal ISO representations of a $\mathcal{C}_h$ horizontal and a $\mathcal{C}_v$ vertical codes. Then,

$$A = \begin{bmatrix} A_h \otimes I_{n_v - k_v} & B_h \otimes C_v & 0 \\ 0 & I_{k_h} \otimes A_v & 0 \\ 0 & 0 & A_h \otimes I_{k_v} \end{bmatrix} B = \begin{bmatrix} B_h \otimes D_v \\ I_{k_h} \otimes B_v \\ B_h \otimes I_{k_v} \end{bmatrix}$$

$$C = \begin{bmatrix} C_h \otimes I_{n_v - k_v} & D_h \otimes C_v & 0 \\ 0 & I_{k_h} \otimes C_v & 0 \\ 0 & 0 & C_h \otimes I_{k_v} \end{bmatrix} D = \begin{bmatrix} D_h \otimes D_v \\ I_{k_h} \otimes D_v \\ D_h \otimes I_{k_v} \end{bmatrix}$$

is an ISO representation of the convolutional product code $\mathcal{C} = \mathcal{C}_h \otimes \mathcal{C}_v$ of rate $k_h k_v / n_h n_v$.

A minimal ISO representation of a Convolutional Product Code

Let (A_h, B_h, C_h, D_h) and (A_v, B_v, C_v, D_v) be minimal ISO representations of a $\mathcal{C}_h$ horizontal and a $\mathcal{C}_v$ vertical convolutional codes. Assume that $G_h = G_{LD}^{(h)} X_h(z) + G_{\infty}^{(h)} Y_h(z)$ and $G_v = G_{LD}^{(v)} X_v(z) + G_{\infty}^{(v)} Y_v(z)$ are their generator matrices.

$$
\begin{aligned}
G(z) =& G_h(z) \otimes G_v(z) \\
=& \left[G_{LD}^{(h)} \otimes G_{LD}^{(v)} G_{LD}^{(h)} \otimes G_{\infty}^{(v)} G_{\infty}^{(h)} \otimes G_{LD}^{(h)} \right] \begin{bmatrix} X_h(z) \otimes X_v(z) \\ X_h(z) \otimes Y_v(z) \\ Y_h(z) \otimes X_v(z) \end{bmatrix} \\
&+ G_{\infty}^{(h)} \otimes G_{\infty}^{(v)} \left(Y_h(z) \otimes Y_v(z) \right) \\
=& \widetilde{G}_{LD} \widetilde{X}(z) + G_{\infty} Y(z)
\end{aligned}
$$

$$
\widetilde{G}_{LD} \widetilde{X}(z) >> G_{LD} X(z)
$$

$$
\widetilde{G}_{LD} P^{-1} = \begin{bmatrix} G_{LD} & \bar{G}_{LD} \end{bmatrix} \quad P \widetilde{X}(z) = \begin{bmatrix} X(z) \\ 0 \end{bmatrix}
$$

A minimal ISO representation of a Convolutional Product Code

Let (A_h, B_h, C_h, D_h) and (A_v, B_v, C_v, D_v) be minimal ISO representations of a $\mathcal{C}_h$ horizontal and a $\mathcal{C}_v$ vertical convolutional codes. Assume that $G_h = G_{LD}^{(h)} X_h(z) + G_{\infty}^{(h)} Y_h(z)$ and $G_v = G_{LD}^{(v)} X_v(z) + G_{\infty}^{(v)} Y_v(z)$ are their generator matrices.

$$
\begin{aligned}
G(z) =& G_h(z) \otimes G_v(z) \\
=& \left[G_{LD}^{(h)} \otimes G_{LD}^{(v)} G_{LD}^{(h)} \otimes G_{\infty}^{(v)} G_{\infty}^{(h)} \otimes G_{LD}^{(h)} \right] \begin{bmatrix} X_h(z) \otimes X_v(z) \\ X_h(z) \otimes Y_v(z) \\ Y_h(z) \otimes X_v(z) \end{bmatrix} \\
&+ G_{\infty}^{(h)} \otimes G_{\infty}^{(v)} \left(Y_h(z) \otimes Y_v(z) \right) \\
=& \widetilde{G}_{LD} \widetilde{X}(z) + G_{\infty} Y(z)
\end{aligned}
$$

$$\widetilde{G}_{LD} \widetilde{X}(z) >> G_{LD} X(z)$$

$$\widetilde{G}_{LD} P^{-1} = \begin{bmatrix} G_{LD} & \bar{G}_{LD} \end{bmatrix} \quad P \widetilde{X}(z) = \begin{bmatrix} X(z) \\ 0 \end{bmatrix}$$

A minimal ISO representation of a Convolutional Product Code

Let (A_h, B_h, C_h, D_h) and (A_v, B_v, C_v, D_v) be minimal ISO representations of a $\mathcal{C}_h$ horizontal and a $\mathcal{C}_v$ vertical convolutional codes. Assume that $G_h = G_{LD}^{(h)} X_h(z) + G_{\infty}^{(h)} Y_h(z)$ and $G_v = G_{LD}^{(v)} X_v(z) + G_{\infty}^{(v)} Y_v(z)$ are their generator matrices.

$$
\begin{aligned}
G(z) =& G_h(z) \otimes G_v(z) \\
=& \left[G_{LD}^{(h)} \otimes G_{LD}^{(v)} G_{LD}^{(h)} \otimes G_{\infty}^{(v)} G_{\infty}^{(h)} \otimes G_{LD}^{(h)} \right] \begin{bmatrix} X_h(z) \otimes X_v(z) \\ X_h(z) \otimes Y_v(z) \\ Y_h(z) \otimes X_v(z) \end{bmatrix} \\
& + G_{\infty}^{(h)} \otimes G_{\infty}^{(v)} \left(Y_h(z) \otimes Y_v(z) \right) \\
=& \widetilde{G}_{LD} \widetilde{X}(z) + G_{\infty} Y(z)
\end{aligned}
$$

$$
\widetilde{G}_{LD} \widetilde{X}(z) >> G_{LD} X(z)
$$

$$
\widetilde{G}_{LD} P^{-1} = \begin{bmatrix} G_{LD} & \bar{G}_{LD} \end{bmatrix} \quad P \widetilde{X}(z) = \begin{bmatrix} X(z) \\ 0 \end{bmatrix}
$$

Summary of the basic ideas

- We provide a method to obtain a minimal ISO representation (A, B, C, D), using the generator matrix of a convolutional code $\mathcal{C}$, without the need to compute the first-order representation (K, L, M).
- We will use this algorithm to apply in convolutional product podes to obtain a minimal ISO representation from minimal ISO representations of a vertical and a horizontal convolutional code.

Summary of the basic ideas

- We provide a method to obtain a minimal ISO representation (A, B, C, D), using the generator matrix of a convolutional code $\mathcal{C}$, without the need to compute the first-order representation (K, L, M).
- We will use this algorithm to apply in convolutional product podes to obtain a minimal ISO representation from minimal ISO representations of a vertical and a horizontal convolutional code.


Rosenthal, J., Schumacher, J.M., and York, E.V. (1996).
On behaviors and convolutional codes.
IEEE Transactions on Information Theory, 42(6), 1881–1891.


Rosenthal, J., York, F. V.
BCH convolutional codes (1999)
IEEE Transactions on Information Theory, 45(6), 1833–1844


Climent, J.-J. and Napp, D. and Pinto, R. and Requena, V.(2021).
Minimal State-Space Representation of Convolutional Product Codes.
Mathematics, 9(12).


Climent, J.-J., Herranz, M., and Perea, C. (2015).
Input-state-output representation of convolutional product codes.
Proceedings of the 4th International Castle Meeting on Coding Theory and Applications (4ICMCTA), CIM Series in Mathematical Sciences, 3(6), 107–114.

Thank you